A Bayesian approach to constrained single- and multi-objective optimization

Paul Féliot, Julien Bect, Emmanuel Vazquez

A Bayesian approach to constrained single- and multi-objective optimization

Paul FELIOT · Julien BECT · Emmanuel VAZQUEZ

Abstract This article addresses the problem of derivative-free (single- or multi-objective) optimization subject to multiple inequality constraints. Both the objective and constraint functions are assumed to be smooth, non-linear and expensive to evaluate. As a consequence, the number of evaluations that can be used to carry out the optimization is very limited, as in complex industrial design optimization problems. The method we propose to overcome this difficulty has its roots in both the Bayesian and the multi-objective optimization literatures. More specifically, an extended domination rule is used to handle the constraints and a corresponding Bayesian expected hyper-volume improvement sampling criterion is proposed. This new criterion extends existing Bayesian sampling criteria to the multi-objective constrained case, and makes it possible to start the algorithm without an initial feasible point. The calculation and optimization of the criterion are performed using Sequential Monte Carlo techniques. In particular, an algorithm similar to the subset simulation method, which is well known in the field of structural reliability, is used to estimate the expected hyper-volume improvement criterion. The method, which we call BMOO (for Bayesian Multi-Objective Optimization), is compared to state-of-the-art algorithms for single-objective and multi-objective constrained optimization problems.

Keywords Bayesian optimization · Kriging · Gaussian process · Multi-Objective · Sequential Monte Carlo · Subset simulation

1 Introduction

This article addresses the problem of derivative-free multi-objective optimization of real-valued functions subject to multiple inequality constraints. The problem consists in finding an approximation of the set

$$\Gamma = \{x \in \mathbb{X} : c(x) \leq 0 \text{ and } \nexists\, x' \in \mathbb{X} \text{ such that } \ f(x') \prec f(x)\} \tag{1}$$

where $\mathbb{X} \subset \mathbb{R}^d$ is the search domain, $c = (c_i)_{1 \leq i \leq q}$ is a vector of constraint functions ($c_i : \mathbb{X} \to \mathbb{R}$), $c(x) \leq 0$ means that $c_i(x) \leq 0$ for all $1 \leq i \leq q$, $f = (f_j)_{1 \leq j \leq p}$ is a vector of objective functions to be minimized ($f_j : \mathbb{X} \to \mathbb{R}$), and $\prec$ denotes the Pareto domination rule (see, e.g., Fonseca and Fleming, 1998). Both the objective functions f_j and the constraint functions c_i are assumed to be continuous. The search domain $\mathbb{X}$ is assumed to be compact—typically, $\mathbb{X}$ is a hyper-rectangle defined by bound constraints. Moreover, the objective and constraint functions are regarded as black boxes and, in particular, we assume that no gradient information is available. Finally, the objective and the

Paul Feliot (Ph.D. student) – E-mail: paul.feliot@irt-systemx.fr
Institut de Recherche Technologique SystemX, Palaiseau, France.
Julien Bect & Emmanuel Vazquez – E-mail: firstname.lastname@centralesupelec.fr
Laboratoire des Signaux et Systèmes, Gif-sur-Yvette, France

constraint functions are assumed to be expensive to evaluate, which arises for instance when the values $f(x)$ and $c(x)$, for a given $x \in \mathbb{X}$, correspond to the outputs of a computationally expensive computer program. In this setting, the emphasis is on building optimization algorithms that perform well under a very limited budget of evaluations (e.g., a few hundred evaluations).

We adopt a *Bayesian approach* to this optimization problem. The essence of Bayesian optimization is to choose a prior model for the expensive-to-evaluate function(s) involved in the optimization problem—usually a Gaussian process model (Santner et al., 2003; Williams and Rasmussen, 2006) for tractability—and then to select the evaluation points sequentially in order to obtain a small average error between the approximation obtained by the optimization algorithm and the optimal solution, under the selected prior. See, e.g., Kushner (1964); Mockus (1975); Mockus et al. (1978); Archetti and Betrò (1979) and Mockus (1994) for some of the earliest references in the field. During several decades, Bayesian optimization research has focused on the case of single-objective bound-constrained optimization: the Expected Improvement (EI) criterion (Mockus et al., 1978; Jones et al., 1998) has emerged in this case as one of the most popular criteria for selecting evaluation points. More recently, the EI criterion has been extended to handle constraints (Schonlau et al., 1998; Sasena et al., 2002; Gramacy and Lee, 2011; Parr et al., 2012; Gramacy et al., to appear) and to address bound-constrained multi-objective problems (Emmerich et al., 2006; Jeong et al., 2006; Wagner et al., 2010; Svenson and Santner, 2010). The more general case of a non-linearly constrained multi-objective problem has only been addressed very recently by Shimoyama et al. (2013). In their paper, Shimoyama et al. consider three different Bayesian criteria for bound-constrained multi-objective optimization and study the effect of multiplying these criteria by a probability of feasibility in order to handle the constraints. This heuristic has the merit of simplicity but suffers from limitations, in particular when the constraints are difficult to satisfy and no feasible solution is known from the start.

In this article, we propose a new Bayesian optimization algorithm for multi-objective non-linearly constrained problems. Our contribution is twofold. First, we propose a new sampling criterion, which can handle both multiple objectives and multiple non-linear constraints. This criterion corresponds to a one-step look-ahead Bayesian strategy, using the *dominated hyper-volume as an utility function* (following in this respect Emmerich et al., 2006). The dominated hyper-volume is defined using an *extended domination rule*, which handles both the objectives and constraints jointly (in the spirit of Fonseca and Fleming, 1998; Ray et al., 2001; Oyama et al., 2007). This makes it possible to deal with problems where no feasible solution is known at the beginning of the optimization procedure. Several criteria from the literature are recovered as special cases. The second part of our contribution resides in the numerical methods employed to compute and optimize the sampling criterion. Indeed, this criterion takes the form of an integral over the space of constraints and objectives, for which no analytical expression is available in the general case. Besides, it must be optimized at each iteration of the algorithm to determine the next evaluation point. In order to compute the integral, we use an algorithm similar to the *subset simulation method* (Au and Beck, 2001; Cérou et al., 2012), which is well known in the field of structural reliability and rare event estimation. For the optimization of the criterion, we resort to *Sequential Monte Carlo* (SMC) techniques (see Del Moral et al., 2006; Liu, 2008, and references therein), following earlier work by Benassi et al. (2012) for single-objective bound-constrained problems. The resulting algorithm is called BMOO (for Bayesian multi-objective optimization).

The structure of the article is as follows. In Section 2, we recall the framework of Bayesian optimization based on the expected improvement sampling criterion, starting with the unconstrained single-objective setting. Section 3 presents our new sampling criterion for constrained multi-objective optimization. The calculation and the optimization of the criterion are discussed in Section 4. Section 5 presents experimental results. An illustration on a two-dimensional toy problem is proposed for visualization purpose. Then, the performances of the method are compared to those of reference methods on both single-objective and multi-objective constrained optimization problems from the literature. Finally, future work is discussed in Section 6.

2 Background literature

2.1 Expected Improvement

Consider the single-objective unconstrained optimization problem

$$x^\star = \operatorname{argmin}_{x\in\mathbb{X}} f(x)\,,$$

where f is a continuous real-valued function defined over $\mathbb{X}\subset\mathbb{R}^d$. Our objective is to find an approximation of $x^\star$ using a sequence of evaluation points $X_1,\, X_2,\, \ldots \in \mathbb{X}$. Because the choice of a new evaluation point X_{n+1} at iteration n depends on the evaluation results of f at $X_1,\, \ldots,\, X_n$, the construction of an optimization strategy $\underline{X} : f \mapsto (X_1,\, X_2,\, X_3\ldots)$ is a sequential decision problem.

The Bayesian approach to this decision problem originates from the early work of Kushner (1964) and Mockus et al. (1978). Assume that a loss function $\varepsilon_n(\underline{X}, f)$ has been chosen to measure the performance of the strategy $\underline{X}$ on f after n evaluations. For instance, Mockus et al. consider the loss function

$$\varepsilon_n(\underline{X}, f) = m_n - m\,, \tag{2}$$

with $m_n = f(X_1)\wedge\cdots\wedge f(X_n)$ and $m = \min_{x\in\mathbb{X}} f(x)$. Then, a good strategy in the Bayesian sense is a strategy that achieves, on average, a small value of $\varepsilon_n(\underline{X}, f)$ when n increases, where the average is taken with respect to a stochastic process model ξ (defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P}_0)$, with parameter in $\mathbb{X}$) for the function f. In other words, the Bayesian approach assumes that $f = \xi(\omega, \cdot)$ for some $\omega\in\Omega$. The probability distribution of ξ represents *prior* knowledge on the function f—before actual evaluations are performed. The reader is referred to (Vazquez and Bect, 2014) for a discussion of other possible loss functions in the context of Bayesian optimization.

Observing that the Bayes-optimal strategy for a budget of N evaluations is intractable for N greater than a few units, Mockus et al. (1978) proposed to use a one-step look-ahead strategy (also known as a myopic strategy): given $n < N$ evaluation results, the next evaluation point X_{n+1} is chosen in order to minimize the conditional expectation of the future loss $\varepsilon_{n+1}(\underline{X}, \xi)$ given available evaluation results:

$$X_{n+1} = \operatorname{argmin}_{x\in\mathbb{X}} \mathbb{E}_n\big(\varepsilon_{n+1}(\underline{X}, \xi) \mid X_{n+1} = x\big)\,, \tag{3}$$

where $\mathbb{E}_n$ stands for the conditional expectation with respect to $X_1,\, \xi(X_1),\, \ldots,\, X_n,\, \xi(X_n)$. Most of the work produced in the field of Bayesian optimization since then has been focusing, as the present paper will, on one-step look-ahead (or similar) strategies; the reader is referred to Ginsbouger and Le Riche (2009) and Benassi (2013) for discussions about two-step look-ahead strategies.

When (2) is used as a loss function, the right-hand side of (3) can be rewritten as

$$\begin{aligned}\operatorname{argmin} \mathbb{E}_n\big(\varepsilon_{n+1}(\underline{X}, \xi) \mid X_{n+1} = x\big) &= \operatorname{argmin} \mathbb{E}_n\left(m_{n+1} \;\middle|\; X_{n+1} = x\right)\\ &= \operatorname{argmax} \mathbb{E}_n\big((m_n - \xi(x))_+\big)\,,\end{aligned} \tag{4}$$

with $z_+ = \max(z,\, 0)$. The function

$$\rho_n(x) : x \mapsto \mathbb{E}_n\big((m_n - \xi(x))_+\big) \tag{5}$$

is called the *Expected Improvement* (EI) criterion. It was first introduced by Mockus et al. (1978), and later popularized through the EGO algorithm (Schonlau et al., 1998; Jones et al., 1998). If we assume that ξ is a Gaussian process with known mean and covariance functions, then $\rho_n(x)$ has a closed-form expression:

$$\rho_n(x) \;=\; \gamma\left(\, m_n - \widehat{\xi}_n(x),\, \sigma_n^2(x)\,\right), \tag{6}$$

where

$$\gamma(z, s) \;=\; \begin{cases} \sqrt{s}\,\varphi\left(\frac{z}{\sqrt{s}}\right) + z\,\Phi\left(\frac{z}{\sqrt{s}}\right) & \text{if } s > 0,\\ \max(z, 0) & \text{if } s = 0,\end{cases}$$

with Φ standing for the normal cumulative distribution function, $\varphi = \Phi'$ for the normal probability density function, $\widehat{\xi}_n(x) = \mathbb{E}_n\left(\xi(x)\right)$ for the kriging predictor at x (posterior mean of $\xi(x)$ after n evaluations) and $\sigma_n^2(x)$ for the kriging variance at x (posterior variance of $\xi(x)$ after n evaluations). See, e.g., the books of Stein (1999), Santner et al. (2003), and Williams and Rasmussen (2006) for more information on Gaussian process models and kriging (also known as Gaussian process interpolation).

Finally, observe that the one-step look-ahead strategy (3) requires to solve an auxiliary global optimization problem on $\mathbb{X}$ for each new evaluation point to be selected. The objective function ρ_n is rather inexpensive to evaluate when ξ is a Gaussian process, using (6), but it is typically severely multi-modal. A simple method to optimize ρ_n consists in choosing a fixed finite set of points that covers $\mathbb{X}$ reasonably well and perform a discrete search. Recently, *sequential Monte Carlo* techniques (see Del Moral et al., 2006; Liu, 2008, and references therein) have been shown to be a valuable tool for this task (Benassi et al., 2012). A review of other approaches is provided in the PhD thesis of Benassi (2013, Section 4.2).

2.2 EI-based multi-objective optimization without constraints

We now turn to the case of unconstrained multi-objective optimization. Under this framework, we consider a set of objective functions $f_j : \mathbb{X} \to \mathbb{R}$, $j = 1, \ldots, p$, to be minimized, and the objective is to build an approximation of the *Pareto front* and of the set of corresponding solutions

$$\Gamma = \left\{x \in \mathbb{X} : \nexists\, x' \in \mathbb{X} \text{ such that } \; f(x') \prec f(x)\right\}, \tag{7}$$

where $\prec$ stands for the Pareto domination rule defined by

$$y = (y_1, \ldots, y_p) \prec z = (z_1, \ldots, z_p) \Longleftrightarrow \begin{cases} \forall i \leq p, \;\; y_i \leq z_i \,, \\ \exists j \leq p, \;\; y_j < z_j \,. \end{cases} \tag{8}$$

Given evaluation results $f(X_1) = (f_1(X_1), \ldots, f_p(X_1))$, …, $f(X_n) = (f_1(X_n), \ldots, f_p(X_n))$, define

$$H_n = \left\{y \in \mathbb{B}; \exists i \leq n,\, f(X_i) \prec y\right\}, \tag{9}$$

where $\mathbb{B} \subset \mathbb{R}^p$ is a set of the form $\mathbb{B} = \left\{y \in \mathbb{R}^p;\, y \leq y^{\text{upp}}\right\}$ for some $y^{\text{upp}} \in \mathbb{R}^p$, which is introduced to ensure that the volume of H_n is finite. H_n is the subset of $\mathbb{B}$ whose points are dominated by the evaluations.

A natural idea to extend the EI sampling criterion (5) is to consider the *improvement* yielded by a new evaluation result $f(X_{n+1}) = (f_1(X_{n+1}), \ldots, f_p(X_{n+1}))$, where the improvement is defined as the increase of the volume of the dominated region:

$$I_n\left(X_{n+1}\right) = \left|H_{n+1} \setminus H_n\right| = \left|H_{n+1}\right| - \left|H_n\right|, \tag{10}$$

where $|\cdot|$ denotes the usual (Lebesgue) volume in $\mathbb{R}^p$ (see Figure 1). Given a vector-valued Gaussian random process model $\xi = (\xi_1, \ldots, \xi_p)$ of $f = (f_1, \ldots, f_p)$, defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P}_0)$, a multi-objective EI criterion can then be derived as

$$\begin{aligned} \rho_n(x) &= \mathbb{E}_n\left(I_n(x)\right) \\ &= \mathbb{E}_n\left(\int_{\mathbb{B}\setminus H_n} \mathbb{1}_{\xi(x) \prec y} \,\mathrm{d}y\right) \\ &= \int_{\mathbb{B}\setminus H_n} \mathbb{E}_n\left(\mathbb{1}_{\xi(x) \prec y}\right) \,\mathrm{d}y \\ &= \int_{\mathbb{B}\setminus H_n} \mathbb{P}_n\left(\xi(x) \prec y\right) \,\mathrm{d}y \,, \end{aligned} \tag{11}$$

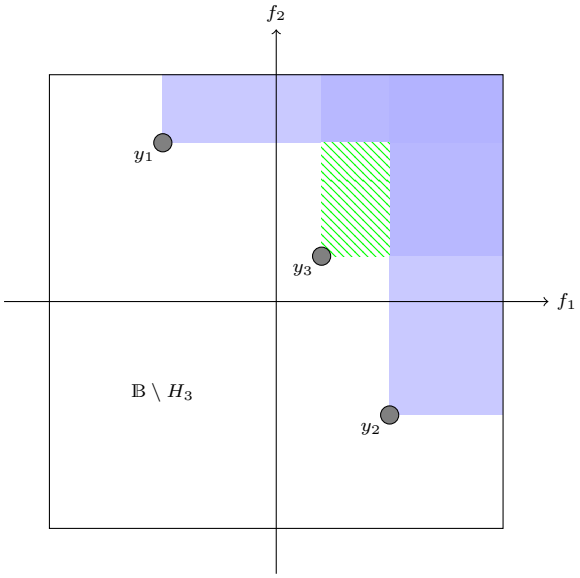


Fig. 1: Example of an improvement of the dominated region. The regions dominated by y_1 and y_2 are represented in shaded areas, with darker shades indicating overlapping regions. The hatched area corresponds to the improvement of the dominated region resulting from the observation of y_3.

where $\mathbb{P}_n$ stands for the probability $\mathbb{P}_0$ conditioned on $X_1, \xi(X_1), \ldots, X_n, \xi(X_n)$. The multi-objective sampling criterion (11), also called Expected Hyper-Volume Improvement (EHVI), has been studied by Emmerich and coworkers (Emmerich, 2005; Emmerich et al., 2006; Emmerich and Klinkenberg, 2008; Wagner et al., 2010; Hupkens et al., 2014). Exact and approximate implementations of the EHVI criterion are available, together with other Gaussian process-based criteria for unconstrained multi-objective optimization, in the GPareto package (Binois and Picheny, 2015) for R.

Remark 1 An alternative approach based on the use of a penalty function has been proposed by Knowles (see Knowles, 2006; Knowles and Hughes, 2005) under the name ParEGO. This algorithm was shown to be almost systematically outperformed by a dominated hypervolume-based algorithm called SMS-EGO (see Ponweiser et al. (2008)) on a small collection of test problems, but an in-depth comparison between dominated hypervolume-based approaches (including the algorithm proposed in this paper) and aggregation-based techniques remains to be done, and is left for future work.

Remark 2 The multi-objective sampling criterion (11) reduces to the usual EI criterion (5) in the single-objective case (assuming that $f(X_i) \leq y^{\text{upp}}$ for at least one $i \leq n$).

Under the assumption that the components ξ_i of ξ are mutually independent[1], $\mathbb{P}_n\left(\xi(x) \prec y\right)$ can be expressed in closed form: for all $x \in \mathbb{X}$ and $y \in \mathbb{B} \setminus H_n$,

$$\mathbb{P}_n\left(\xi(x) \prec y\right) = \prod_{i=1}^{p} \Phi\left(\frac{y_i - \widehat{\xi}_{i,n}(x)}{\sigma_{i,n}(x)}\right), \tag{12}$$

where $\widehat{\xi}_{i,n}(x)$ and $\sigma^2_{i,n}(x)$ denote respectively the kriging mean and the kriging variance at x for the i^{th} component of ξ.

[1] This is the most common modeling assumption in the Bayesian optimization literature, when several objective functions, and possibly also several constraint functions, have to be dealt with. See the VIPER algorithm of Williams et al. (2010) for an example of an algorithm based on correlated Gaussian processes.

The integration of (12) over $\mathbb{B} \setminus H_n$, in the expression (11) of the multi-objective EI criterion, is a non-trivial problem. Several authors (Emmerich and Klinkenberg, 2008; Bader and Zitzler, 2011; Hupkens et al., 2014) have proposed decomposition methods to carry out this computation, where the integration domain $\mathbb{B} \setminus H_n$ is partitioned into hyper-rectangles, over which the integral can be computed analytically. The computational complexity of these methods, however, increases exponentially with the number of objectives, which makes the approach impractical in problems with more than a few objective functions. The method proposed in this work also encounters this type of integration problem, but takes a different route to solve it (using SMC techniques; see Section 4). Our approach will make it possible to deal with more objective functions.

2.3 EI-based single-objective optimization with constraints

In this section, we consider optimization problems with a single objective and several constraints.

$$\begin{cases} \min_{x \in \mathbb{X}} f(x)\,, \\ c(x) \leq 0\,, \end{cases} \tag{13}$$

where $c = (c_1, \ldots, c_q\}$ is a vector of continuous constraints. The set $C = \{x \in \mathbb{X};\, c(x) \leq 0\}$ is called the *feasible domain*. If it is assumed that at least one evaluation has been made in C, it is natural to define a notion of improvement with respect to the best objective value $m_n = \min\{f(x);\, x \in \{X_1, \ldots, X_n\} \cap C\}$:

$$\begin{aligned} I_n(X_{n+1}) &= \mathbb{1}_{c(X_{n+1}) \leq 0} \cdot \big(m_n - f(X_{n+1})\big)_+ \\ &= \begin{cases} m_n - f(X_{n+1}) & \text{if } X_{n+1} \in C \text{ and } f(X_{n+1}) < m_n, \\ 0 & \text{otherwise}\,. \end{cases} \end{aligned} \tag{14}$$

In other words, a new observation makes an improvement if it is feasible and improves upon the best past value (Schonlau et al., 1998).

If f is modeled by a random process ξ_{o} and c is modeled by a vector-valued random process $\xi_{\mathrm{c}} = (\xi_{\mathrm{c},1}, \ldots, \xi_{\mathrm{c},q})$ independent of ξ_{o}, then an EI sampling criterion can be written as

$$\begin{aligned} \rho_n(x) &= \mathbb{E}_n\left(\mathbb{1}_{\xi_{\mathrm{c}}(x) \leq 0} \cdot \big(m_n - \xi_{\mathrm{o}}(x)\big)_+\right) \\ &= \mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0)\, \mathbb{E}_n\big((m_n - \xi_{\mathrm{o}}(x))_+\big)\,. \end{aligned} \tag{15}$$

The term $\mathbb{E}_n\big((m_n - \xi_{\mathrm{o}}(x))_+\big)$ can be computed analytically as in Section 2.1. If we further assume that the components of ξ_{c} are Gaussian and independent, then

$$\mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0) = \prod_{j=1}^{q} \Phi\left(-\frac{\widehat{\xi}_{\mathrm{c},\,j,\,n}(x)}{\sigma_{\mathrm{c},\,j,\,n}(x)}\right) \tag{16}$$

where $\widehat{\xi}_{\mathrm{c},\,j,\,n}(x)$ and $\sigma^2_{\mathrm{c},\,j,\,n}(x)$ stand respectively for the kriging predictor and kriging variance of $\xi_{\mathrm{c},\,j}$ at x. The sampling criterion (15)–(16) is extensively discussed, and compared with other Gaussian-process based constraint handling methods, in the PhD thesis of Sasena (2002). More generally, sampling criteria for constrained optimization problems have been reviewed recently by Parr et al. (2012).

Remark 3 Gramacy et al. (to appear) have recently proposed another approach to Gaussian process-based constrained optimization, based on the *augmented Lagrangian* approach of Conn et al. (1991). As indicated in Remark 1, an in-depth comparison between dominated hypervolume-based approaches and aggregation-based techniques is left for future work.

Remark 4 The improvement (14) and the corresponding EI criterion (15) are well defined under the assumption that at least one feasible point has already been obtained, which is an important limitation in practice. We will propose, in Section 3, a generalization of the EI criterion which is well defined even without a feasible point available. Other Gaussian process-based approaches that can be used without an initial feasible point include the method by Gramacy et al. (to appear) based on the augmented Lagrangian approach (see Remark 3) and several recent methods (Picheny, 2014; Gelbart, 2015; Hernández-Lobato et al., 2015) based on Stepwise Uncertainty Reduction (SUR) strategies (Villemonteix et al., 2009; Bect et al., 2012; Chevalier et al., 2014).

3 An EI criterion for constrained multi-objective optimization

3.1 Extended domination rule

In a constrained multi-objective optimization setting, we propose to handle the constraints using an extended Pareto domination rule that takes both objectives and constraints into account, in the spirit of Fonseca and Fleming (1998), Ray et al. (2001) and Oyama et al. (2007). For ease of presentation, denote by $\mathbb{Y}_{\mathrm{o}} = \mathbb{R}^p$ and $\mathbb{Y}_{\mathrm{c}} = \mathbb{R}^q$ the objective and constraint spaces respectively, and let $\mathbb{Y} = \mathbb{Y}_{\mathrm{o}} \times \mathbb{Y}_{\mathrm{c}}$.

We shall say that $y_1 \in \mathbb{Y}$ dominates $y_2 \in \mathbb{Y}$, which will be written as $y_1 \lhd y_2$, if $\psi(y_1) \prec \psi(y_2)$, where $\prec$ is the usual Pareto domination rule recalled in Section 2.2 and, denoting by $\overline{\mathbb{R}}$ the extended real line,

$$\begin{aligned} \psi : \mathbb{Y}_{\mathrm{o}} \times \mathbb{Y}_{\mathrm{c}} &\to \overline{\mathbb{R}}^p \times \mathbb{R}^q \\ (y_{\mathrm{o}},\, y_{\mathrm{c}}) &\mapsto \begin{cases} (y_{\mathrm{o}},\, 0) & \text{if } y_{\mathrm{c}} \leq 0, \\ \big(+\infty,\, \max(y_{\mathrm{c}}, 0)\big) & \text{otherwise.} \end{cases} \end{aligned} \tag{17}$$

The extended domination rule (17) has the following properties:

(i) For unconstrained problems ($q = 0$, $\mathbb{Y}_{\mathrm{c}} = \emptyset$), the extended domination rule boils down to the Pareto domination rule on $\mathbb{Y} = \mathbb{Y}_{\mathrm{o}}$.
(ii) Feasible solutions (corresponding to $y_{\mathrm{c}} \leq 0$) are compared using the Pareto domination rule applied in the objective space (in other words, using the Pareto domination rule with respect to the objective values y_{o}).
(iii) Non-feasible solutions (corresponding to $y_{\mathrm{c}} \not\leq 0$) are compared using the Pareto domination rule applied to the vector of constraint violations.
(iv) Feasible solutions always dominate non-feasible solutions.

These properties are illustrated on Figure 2.

3.2 A new EI criterion

The extended domination rule presented above makes it possible to define a notion of expected hypervolume improvement as in Section 2.2 for the *constrained* multi-objective setting. Given evaluation results $(f(X_1), c(X_1))$, …, $(f(X_n), c(X_n))$, define

$$H_n = \{y \in \mathbb{B};\ \exists i \leq n,\ (f(X_i),\, c(X_i)) \lhd y\}$$

with $\mathbb{B} = \mathbb{B}_{\mathrm{o}} \times \mathbb{B}_{\mathrm{c}}$, where $\mathbb{B}_{\mathrm{o}} \subset \mathbb{Y}_{\mathrm{o}}$ and $\mathbb{B}_{\mathrm{c}} \subset \mathbb{Y}_{\mathrm{c}}$ are two bounded hyper-rectangles that are introduced to ensure, as in Section 2.2, that $|H_n| < +\infty$ (see Appendix A). Then, define the improvement yielded by a new evaluation $(f(X_{n+1}),\, c(X_{n+1}))$ by

$$I_n\left(X_{n+1}\right) = |H_{n+1} \setminus H_n| = |H_{n+1}| - |H_n|$$

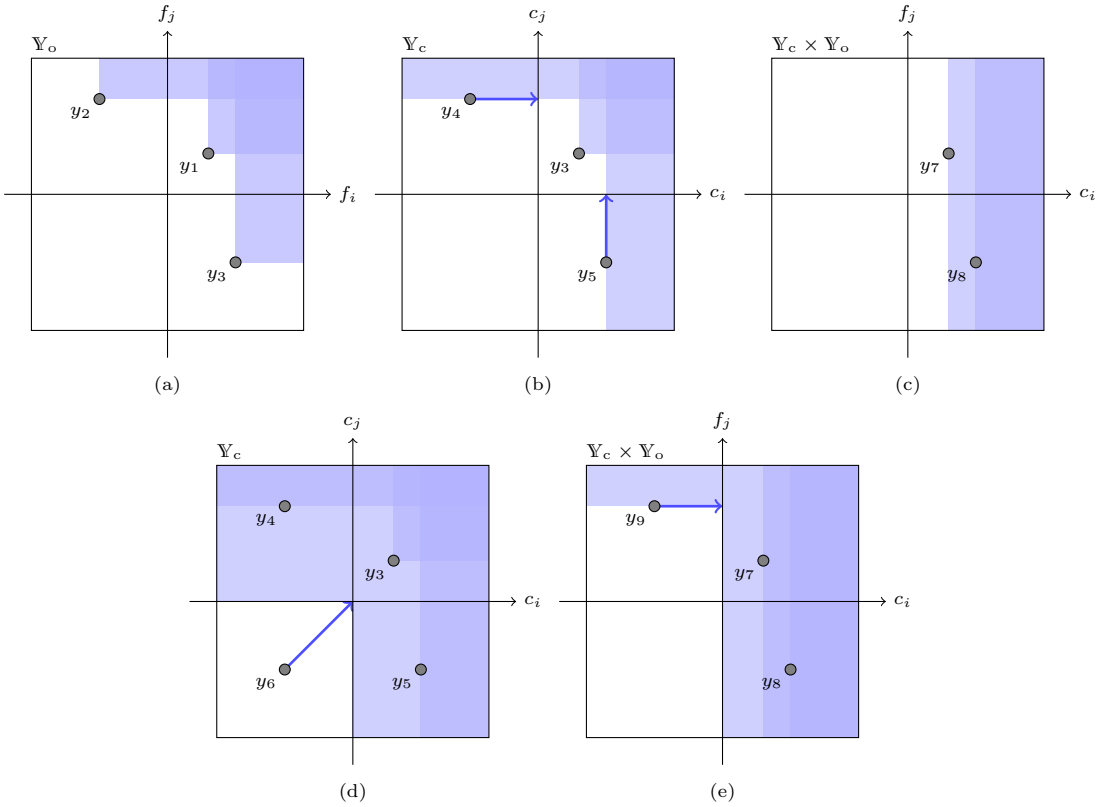


Fig. 2: Illustration of the extended domination rule in different cases. The region dominated by each point is represented by a shaded area. Darker regions indicate overlapping regions. (a) Feasible solutions are compared with respect to their objective values using the usual domination rule in the objective space—see properties (i) and (ii). (b–c) Non-feasible solutions are compared using the Pareto domination rule applied to the vectors of constraint violations according to property (iii). Note that y_4 dominates points having a higher value of c_j regardless of the corresponding value of c_i, and, likewise, y_5 dominates points with higher values of c_i. (d–e) Feasible solutions always dominate non-feasible solutions: y_6 is feasible and hence dominates y_3, y_4 and y_5; y_9 is feasible and dominates both y_7 and y_8 as stated in (iv).

as in Section 2.2. In order to get a meaningful concept of improvement both before and after the first feasible point has been found, we assume without loss of generality that $(0, \ldots, 0) \in \mathbb{R}^q$ is in the interior of $\mathbb{B}_c$.

If (f, c) is modeled by a vector-valued random process $\xi = (\xi_o, \xi_c)$, with $\xi_o = (\xi_{o,1}, \ldots, \xi_{o,p})$ and $\xi_c = (\xi_{c,1}, , \ldots \xi_{c,q})$, then the expected improvement for the constrained multi-objective optimization problem may be written as

$$\rho_n(x) = \mathbb{E}_n\big((I_n(x)\big) = \mathbb{E}_n\left(\int_{G_n} \mathbb{1}_{\xi(x) \lhd y} \, \mathrm{d}y\right) = \int_{G_n} \mathbb{P}_n(\xi(x) \lhd y) \, \mathrm{d}y \,, \tag{18}$$

where $G_n = \mathbb{B} \setminus H_n$ is the set of all non-dominated points in $\mathbb{B}$.

As in Section 2.2, under the assumption that the components of ξ are mutually independent and Gaussian, $\mathbb{P}_n\left(\xi(x) \lhd y\right)$ can be expressed in closed form: for all $x \in \mathbb{X}$ and $y = (y_{\mathrm{o}},\, y_{\mathrm{c}}) \in G_n$,

$$\mathbb{P}_n(\xi(x) \lhd y) = \begin{cases} \left(\displaystyle\prod_{i=1}^{p} \Phi\left(\frac{y_{\mathrm{o},\,i} - \widehat{\xi}_{\mathrm{o},\,i,\,n}(x)}{\sigma_{\mathrm{o},\,i,\,n}(x)}\right)\right)\left(\displaystyle\prod_{j=1}^{q} \Phi\left(-\frac{\widehat{\xi}_{\mathrm{c},\,j,\,n}(x)}{\sigma_{\mathrm{c},\,j,\,n}(x)}\right)\right) & \text{if } y_{\mathrm{c}} \leq 0\,, \\ \displaystyle\prod_{j=1}^{q} \Phi\left(\frac{\max(y_{\mathrm{c},\,j},\,0) - \widehat{\xi}_{\mathrm{c},\,j,\,n}(x)}{\sigma_{\mathrm{c},\,j,\,n}(x)}\right) & \text{otherwise}\,. \end{cases} \tag{19}$$

The EI-based constrained multi-objective optimization procedure may be written as (3). In practice, the computation of each new evaluation point requires to solve two numerical problems: a) the computation of the integral in (18); b) the optimization of ρ_n in the procedure (3). These problems will be addressed in Section 4.

Remark 5 When there are no constraints ($q = 0$, $\mathbb{Y}_{\mathrm{c}} = \emptyset$), the extended domination rule $\lhd$ corresponds to the usual Pareto domination rule $\prec$. In this case, the sampling criterion (18) simplifies to

$$\rho_n(x) = \int_{\mathbb{B}_{\mathrm{o}} \setminus H_{n,\mathrm{o}}} \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right)\, \mathrm{d}y_{\mathrm{o}}, \tag{20}$$

with

$$H_{n,\mathrm{o}} = \left\{y_{\mathrm{o}} \in \mathbb{B}_{\mathrm{o}};\ \exists i \leq n,\ f(X_i) \prec y_{\mathrm{o}}\right\}.$$

Denote by $y_{\mathrm{o}}^{\mathrm{low}}$, $y_{\mathrm{o}}^{\mathrm{upp}} \in \mathbb{Y}_{\mathrm{o}}$ the lower and upper corners of the hyper-rectangle $\mathbb{B}_{\mathrm{o}}$. Then, the sampling criterion (20) is equivalent to the multi-objective EI criterion presented in Section 2.2 in the limit $y_{\mathrm{o}}^{\mathrm{low}} \to -\infty$. If, moreover, the problem has only one objective function, then the criterion (18) boils down to the original Expected Improvement criterion as soon as the best evaluation dominates $y_{\mathrm{o}}^{\mathrm{upp}}$ (see Remark 2).

3.3 Decomposition of the expected improvement: feasible and unfeasible components

Assume now that there is at least one constraint ($q \geq 1$). Then, the expected improvement $\rho_n(x)$ can be decomposed as

$$\rho_n(x) = \rho_n^{\mathrm{feas}}(x) + \rho_n^{\mathrm{unf}}(x), \tag{21}$$

by splitting the integration domain in the right-hand side of (18) in two parts: $\rho_n^{\mathrm{feas}}(x)$ corresponds to the integral on $(G_n) \cap \{y^{\mathrm{c}} \leq 0\}$, while $\rho_n^{\mathrm{unf}}(x)$ corresponds to the integral on $(G_n) \cap \{y^{\mathrm{c}} \not\leq 0\}$.

More explicit expressions will now be given for both terms. First,

$$\begin{aligned} \rho_n^{\mathrm{unf}}(x) &= \int_{G_n \cap \{y^{\mathrm{c}} \not\leq 0\}} \mathbb{P}_n\left((\xi_{\mathrm{o}}(x), \xi_{\mathrm{c}}(x)) \lhd (y_{\mathrm{o}}, y_{\mathrm{c}})\right)\, \mathrm{d}(y_{\mathrm{o}}, y_{\mathrm{c}}) \\ &= |\mathbb{B}_{\mathrm{o}}| \cdot \int_{\mathbb{B}_{\mathrm{c}} \setminus H_{n,\mathrm{c}}} \mathbb{P}_n\left(\xi_{\mathrm{c}}^{+}(x) \prec y_{\mathrm{c}}^{+}\right)\, \mathbb{1}_{y_{\mathrm{c}} \not\leq 0}\, \mathrm{d}y_{\mathrm{c}} \end{aligned} \tag{22}$$

where $y_{\mathrm{c}}^{+} = \max\left(y_{\mathrm{c}}, 0\right)$ and

$$H_{n,\mathrm{c}} = \left\{y_{\mathrm{c}} \in \mathbb{B}_{\mathrm{c}} \mid \exists i \leq n,\ \xi_{\mathrm{c}}^{+}(X_i) \prec y_{\mathrm{c}}^{+}\right\}.$$

Let now $\mathbb{B}_{\mathrm{c}}^{-} = \mathbb{B}_{\mathrm{c}} \cap \left]-\infty, 0\right]^q$ denote the feasible subset of $\mathbb{B}_{\mathrm{c}}$. Then, assuming that ξ_{c} and ξ_{o} are independent,

$$\begin{aligned} \rho_n^{\mathrm{feas}}(x) &= \int_{G_n \cap \{y^{\mathrm{c}} \leq 0\}} \mathbb{P}_n\left((\xi_{\mathrm{o}}(x), \xi_{\mathrm{c}}(x)) \lhd (y_{\mathrm{o}}, y_{\mathrm{c}})\right)\, \mathrm{d}(y_{\mathrm{o}}, y_{\mathrm{c}}) \\ &= \left|\mathbb{B}_{\mathrm{c}}^{-}\right| \cdot \mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0) \cdot \int_{\mathbb{B}_{\mathrm{o}} \setminus H_{n,\mathrm{o}}} \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right)\, \mathrm{d}y_{\mathrm{o}}\,, \end{aligned} \tag{23}$$

where

$$H_{n,\mathrm{o}} = \{y_\mathrm{o} \in \mathbb{B}_\mathrm{o} \mid \exists i \leq n,\, \xi_\mathrm{c}(X_i) \leq 0 \text{ and } \xi_\mathrm{o}(X_i) \prec y_\mathrm{o}\}\ .$$

Remark 6 The set $\mathbb{B}_\mathrm{c} \setminus H_{n,\mathrm{c}}$ becomes empty as soon as a feasible point has been evaluated. As a consequence, the component ρ^{unf} of the expected improvement vanishes and therefore, according to (23),

$$\rho_n(x) \propto \mathbb{P}_n(\xi_\mathrm{c}(x) \leq 0) \cdot \int_{\mathbb{B}_\mathrm{o} \setminus H_{n,\mathrm{o}}} \mathbb{P}_n\left(\xi_\mathrm{o}(x) \prec y_\mathrm{o}\right) \mathrm{d}y_\mathrm{o}\,.$$

In other words, up to a multiplicative constant, the expected improvement becomes equal in this case to the product of the probability of feasibility with a modified EHVI criterion in the objective space, where only feasible points are used to define the dominated region. In particular, in the case of constrained single-objective optimization, we recover in the limit $y_\mathrm{o}^{\mathrm{low}} \to -\infty$ the criterion of Schonlau et al. (already discussed in Section 2.3) as soon as the best evaluation dominates $y_\mathrm{o}^{\mathrm{upp}}$.

Remark 7 In our numerical experiments, estimates of the range of the objective and constraint functions will be used to define the hyper-rectangles $\mathbb{B}_\mathrm{o}$ and $\mathbb{B}_\mathrm{c}$, respectively (see Appendix B for a more detailed description of the adaptive algorithm that is used). Another natural choice for the $\mathbb{B}_\mathrm{o}$ would have been to use (an estimate of) the range of the objective functions *restricted to the feasible subset* $C \subset \mathbb{X}$ for $\mathbb{B}_\mathrm{o}$. Further investigation of this idea is left for future work.

4 Sequential Monte Carlo techniques to compute and optimize the expected improvement

4.1 Computation of the expected improvement

Since the dimension of $\mathbb{Y}$ is likely to be high in practical problems (say, $p + q \geq 5$), the integration of $y \mapsto \mathbb{P}_n(\xi(x) \lhd y)$ over G_n cannot be carried out using decomposition methods (Emmerich and Klinkenberg, 2008; Bader and Zitzler, 2011; Hupkens et al., 2014) because, as mentioned in Section 2.2, the computational complexity of these methods increases exponentially with the dimension of $\mathbb{Y}$.

To address this difficulty, we propose to use a Monte Carlo approximation of the integral (18):

$$\rho_n(x) \approx \frac{1}{m} \sum_{k=1}^{m} \mathbb{P}_n(\xi(x) \lhd y_{n,k}), \tag{24}$$

where $\mathcal{Y}_n = (y_{n,k})_{1 \leq k \leq m}$ is a set of *particles* distributed according to the uniform density $\pi_n^{\mathbb{Y}} \propto \mathbb{1}_{G_n}$ on G_n. In principle, sampling uniformly over G_n could be achieved using an accept-reject method (see, e.g., Robert and Casella, 2013), by sampling uniformly over $\mathbb{B}$ and discarding points in H_n (Bader and Zitzler, 2011). However, when the dimension of $\mathbb{Y}$ is high, G_n will probably have a small volume with respect to that of $\mathbb{B}$. Then, the acceptance rate becomes small and the cost of generating a uniform sample on G_n becomes prohibitive. (As an example, consider an optimization problem with $q = 20$ constraints. If $B_\mathrm{c} = [-v/2, +v/2]^q$, then the volume of the feasible region is $2^{20} \approx 10^6$ times smaller than that of $\mathbb{B}_\mathrm{c}$.)

In this work, we use a variant of the technique called *subset simulation* (Au and Beck, 2001; Cérou et al., 2012) to achieve uniform sampling over G_n. The subset simulation method is a well-known method in the field of structural reliability and rare event estimation, which is used to estimate the volume of small sets by Monte Carlo sampling.

Denote by $\Pi_0^{\mathbb{Y}}$ the uniform distribution over $\mathbb{B}$ and assume that the probability $\Pi_0^{\mathbb{Y}}(G_n)$ becomes small when n increases, so that sampling G_n using an accept-reject method is impractical. Observe that the sets G_n, $n = 1, 2, \ldots$ form a nested sequence of subsets of $\mathbb{B}$ (hence the name *subset simulation*):

$$\mathbb{B} \supset G_1 \supset G_2 \supset \cdots\ . \tag{25}$$

Algorithm 1: Remove-Resample-Move procedure to construct $\mathcal{Y}_n$

if $n = 0$ **then**
  Generate m independent and uniformly distributed particles over $G_0 = \mathbb{B}$.
else
  Remove: Set $\mathcal{Y}_n^0 = \mathcal{Y}_{n-1} \cap G_n$ and $m_0 = \left|\mathcal{Y}_n^0\right|$.
  Resample: Set $\mathcal{Y}_n^1 = \mathcal{Y}_n^0 \cup \{\tilde{y}_{n,1}, \ldots, \tilde{y}_{n,m-m_0}\}$, where $\tilde{y}_{n,1}, \ldots, \tilde{y}_{n,m-m_0}$ are independent and uniformly distributed on $\mathcal{Y}_n^0$. (Each $\tilde{y}_{n,k}$ is a replicate of a particle from $\mathcal{Y}_n^0$.)
  Move: Move the particles using a Metropolis-Hastings algorithm (see, e.g, Robert and Casella, 2013) which targets the uniform distribution over G_{n+1}. The resulting set of particles is $\mathcal{Y}_{n+1}$.

Algorithm 2: Modified procedure to construct $\mathcal{Y}_n$

Notation: Given a set A in $\mathbb{Y}$, denote by Pareto(A) the set of points of A that are not dominated by any other point of A

if $n = 0$ **then**
  Generate m independent and uniformly distributed particles over $G_0 = \mathbb{B}$.
else
  Set $\mathcal{P}_{n-1} = \text{Pareto}\left(\{\xi(X_1), \ldots, \xi(X_{n-1})\}\right)$.
  Set $\mathcal{P}_n = \text{Pareto}\left(\{\xi(X_1), \ldots, \xi(X_n)\}\right) = \text{Pareto}\left(\mathcal{P}_{n-1} \cup \{\xi(X_n)\}\right)$.
  Construct $\mathcal{Y}_n$ using the adaptive multi-level splitting procedure described in Algorithm 3, with $\mathcal{Y}_{n-1}$, $\mathcal{P}_{n-1}$, $\mathcal{P}_n$ and $\mathbb{B}$ as inputs.

Denote by $\Pi_n^{\mathbb{Y}}$ the uniform distribution on G_n, which has the probability density function $\pi_n^{\mathbb{Y}}$ defined above. Since the addition of a single new evaluation, at iteration $n+1$, is likely to yield only a small modification of the set G_n, the probability

$$\Pi_n^{\mathbb{Y}}(G_{n+1}) = \int_{G_{n+1}} \pi_n^{\mathbb{Y}}(y)\,\mathrm{d}y \;=\; \frac{\Pi_0^{\mathbb{Y}}(G_{n+1})}{\Pi_0^{\mathbb{Y}}(G_n)}$$

is likely to be high. Then, supposing that a set of particles $\mathcal{Y}_n = (y_{n,k})_{1\le k\le m}$ uniformly distributed on G_n is already available, one obtains a sample $\mathcal{Y}_{n+1}$ uniformly distributed over G_{n+1} using the *Remove-Resample-Move* procedure described in Algorithm 1. (All the random variables generated in Algorithm 1 are independent of ξ conditionally on X_1, $\xi(X_i)$, …, X_{n+1}, $\xi(X_{n+1})$.)

Algorithm 1 obviously requires that at least one particle from $\mathcal{Y}_n$, which belongs by construction to G_n, also belongs to G_{n+1}; otherwise, the set of surviving particles, referred to in the second step of the algorithm, will be empty. More generally, Algorithm 1 will typically fail to produce a good sample from $\Pi_{n+1}^{\mathbb{Y}}$ if the number of surviving particles is small, which happens with high probability if $\Pi_n^{\mathbb{Y}}(G_{n+1})$ is small—indeed, the expected number of particles of $\mathcal{Y}_n$ in a given[2] set $A \subset \mathbb{B}$ is

$$\mathbb{E}_n\Big(N(A;\,\mathcal{Y}_n)\Big) = \mathbb{E}_n\left(\sum_{k=1}^{m} \mathbb{1}_A(y_{n,k})\right) = m \cdot \Pi_n^{\mathbb{Y}}(A)\,, \tag{26}$$

where $N(A;\,\mathcal{Y})$ denotes the number of particles of $\mathcal{Y}$ in A. This situation occurs, for instance, when a new evaluation point brings a large improvement $G_n \setminus G_{n+1} = H_{n+1} \setminus H_n$.

When the number of surviving particles is smaller than a prescribed fraction ν of the population size, that is, when $N(G_{n+1};\,\mathcal{Y}_n) < m\nu$, intermediate subsets are inserted in the decreasing sequence (25) to ensure that the volume of the subsets does not decrease too fast. The corrected version of Algorithm 1 is described in Algorithms 2 and 3. The method used in Algorithm 3 to construct the intermediate subsets is illustrated on Figures 3 and 4.

[2] Equation (26) does not hold exactly for $A = G_{n+1}$ since, conditionally on X_1, $\xi(X_i)$, …, X_n, $\xi(X_n)$, the set G_{n+1} is a *random* set and is not independent of $\mathcal{Y}_n$. Indeed, G_{n+1} depends on $\xi(X_{n+1})$ and X_{n+1} is chosen by minimization of the approximate expected improvement, which in turn is computed using $\mathcal{Y}_n$.

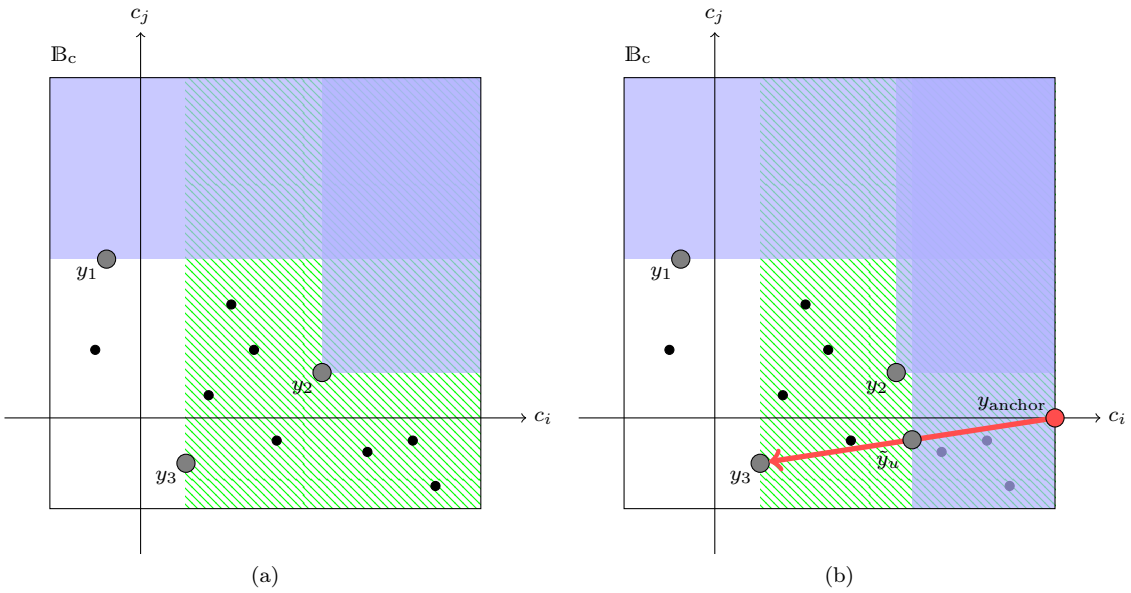


Fig. 3: Illustration of the steps 10 → 12 and 13 → 15 of Algorithm 3. The objective is to build a uniform sample $\mathcal{Y}_3$ on G_3 from $\mathcal{Y}_2$. The initial Pareto front $\mathcal{P}_0$ is determined by evaluations results $y_1 = (f(X_1), c(X_1))$ and $y_2 = (f(X_2), c(X_2))$. $\mathcal{P}_T$ corresponds to the Pareto front determined by $\mathcal{P}_0 \cup \{y_3\}$, with $y3 = (f(X_3), c(X_3))$. At the end of steps 1–9, y_3 is not in $\mathcal{P}$ because the number of surviving particles in $\mathcal{Y}_2$ is too small: in (a), there is only one particle (black dot) in G_3 (white region). Thus, intermediate subsets are needed. The main idea here is to build a *continuous* path between $\mathcal{P}_0$ and $\mathcal{P}_T$, which is illustrated in (b). Here, we pick $y^\star = y_3$ and since y_3 is not feasible, $q^\star < q$. Then, we set an anchor point y_{anchor} on the edge of $\mathbb{B}$, as described in step 14, and we build an intermediate Pareto front $\widetilde{\mathcal{P}}_u$ determined by y_1, y_2 and $\tilde{y}_u$, where $\tilde{y}_u$ lies on the segment (y_{anchor}–y_3). The intermediate Pareto front $\widetilde{\mathcal{P}}_u$ is chosen in such a way that the number of killed particles (grey dots) is not too large.

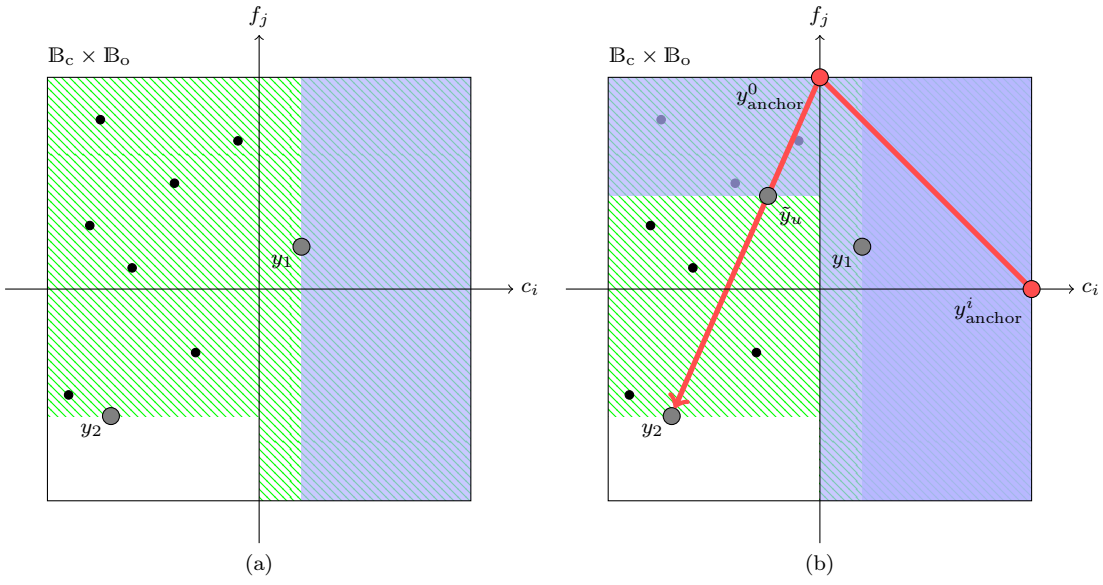


Fig. 4: Illustration of the steps 10 → 12 and 16 → 20 of Algorithm 3. The setting is the same as that described in Figure 3, except that the new observation (y_2 in this case) is feasible. Hence, $q^\star = q$. As above, the main idea is to construct a continuous path between $\mathcal{P}_0$ and $\mathcal{P}_T$, represented by the broken red line.

Algorithm 3: Adaptive multi-level splitting in the $\mathbb{Y}$-domain

Notations: Given a set A in $\mathbb{Y}$, denote by
- $\text{Pareto}(A)$ the set of points of A that are not dominated by any other point of A,
- $G(A) := \mathbb{B} \setminus \{y \in \mathbb{B};\ \exists y' \in A \text{ such that } y' \lhd y\}$ the region of $\mathbb{B}$ not dominated by A.

Inputs: $\mathcal{Y}_0$, $\mathcal{P}_0$, $\mathcal{P}^\star$ and $\mathbb{B}$ such that

- $\mathcal{P}_0 = \text{Pareto}\,(\mathcal{P}_0)$, i.e., no point in $\mathcal{P}_0$ is dominated by another point in $\mathcal{P}_0$, and similarly $\mathcal{P}^\star = \text{Pareto}\,(\mathcal{P}^\star)$,
- $G\,(\mathcal{P}^\star) \subset G\,(\mathcal{P}_0)$,
- $\mathcal{Y}_0 = \big(y_{0,k}\big)_{1\leq k\leq m} \in \mathbb{Y}^m$ is uniformly distributed on $G\,(\mathcal{P}_0)$. Note that $\mathcal{Y}_0$ may contain replicated values.
- $y_{\text{o}}^{\text{low}}, y_{\text{o}}^{\text{upp}}, y_{\text{c}}^{\text{low}}$ and $y_{\text{c}}^{\text{upp}}$ such that $\mathbb{B}_{\text{o}} = \big\{y \in \mathbb{Y}_{\text{o}};\ y_{\text{o}}^{\text{low}} \leq y \leq y_{\text{o}}^{\text{upp}}\big\}$, $\mathbb{B}_{\text{c}} = \big\{y \in \mathbb{Y}_{\text{c}};\ y_{\text{c}}^{\text{low}} \leq y \leq y_{\text{c}}^{\text{upp}}\big\}$, and $\mathbb{B} = \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$ contains $\mathcal{P}_0$ and $\mathcal{P}^\star$.

Output: A set of particles $\mathcal{Y}_T = \big(y_{T,k}\big)_{1\leq k\leq m} \in \mathbb{Y}^m$ uniformly distributed on $G(\mathcal{P}_T)$.

$t \leftarrow 0$
while $\mathcal{P}_t \neq \mathcal{P}^\star$ **do**
  Initialize: $\mathcal{P} \leftarrow \mathcal{P}_t$.
  $\mathcal{P}$ is the front that we will build upon. First we try to add the points of $\mathcal{P}^\star$ into $\mathcal{P}$:
  for $y \in \mathcal{P}^\star$ **do**
    $\mathcal{P}_{\text{try}} \leftarrow \text{Pareto}\,(\mathcal{P} \cup \{y\})$
    Compute the number $N(G(\mathcal{P}_{\text{try}}); \mathcal{Y}_t)$ of particles of $\mathcal{Y}_t$ in $G(\mathcal{P}_{\text{try}})$
    if $N(G(\mathcal{P}_{\text{try}}); \mathcal{Y}_t) \geq \nu m$ **then**
      $\mathcal{P} \leftarrow \mathcal{P}_{\text{try}}$
  At the end of this first step, either $\mathcal{P} = \mathcal{P}^\star$ or $\mathcal{P}^\star \setminus \mathcal{P}$ contains points that cannot be added without killing a large number of particles, in which case we insert intermediate fronts.
  if $(\mathcal{P}^\star \setminus \mathcal{P}) \neq \emptyset$ **then**
    Randomly choose a point $y^\star = (y_{\text{o}}^\star, y_{\text{c}}^\star) \in (\mathcal{P}^\star \setminus \mathcal{P})$ toward which we will try to augment the front $\mathcal{P}$.
    Count the number $q^\star$ of constraints satisfied by $y^\star$.
    if $q^\star < q$ **then**
      $y_{\text{anchor}} \leftarrow (y_{\text{o}}^{\text{upp}}, y_{\text{c}}) \in \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$, where $y_{\text{c},j} = y_{\text{c},j}^{\text{upp}}$ if $y_{\text{c},j}^\star > 0$ and zero otherwise, $1 \leq j \leq q$.
      Find $\widetilde{\mathcal{P}}_u$ such that $N(G(\widetilde{\mathcal{P}}_u); \mathcal{Y}_t) \approx \nu m$ using a dichotomy on $u \in [0,1]$, where $\widetilde{\mathcal{P}}_u = \text{Pareto}(\mathcal{P} \cup \{y_{\text{anchor}} + u(y^\star - y_{\text{anchor}})\})$.
    else
      $y_{\text{anchor}}^0 \leftarrow (y_{\text{o}}^{\text{upp}}, 0) \in \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$
      $y_{\text{anchor}}^k \leftarrow (y_{\text{o}}^{\text{upp}}, y_{\text{c}}^k) \in \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$, where $y_{\text{c},j}^k = y_{\text{c},j}^{\text{upp}}$ if $j = k$ and zero otherwise, for $1 \leq j \leq q$ and $1 \leq k \leq q$.
      if $N(G(\{y_{\text{anchor}}^0\}); \mathcal{Y}_t) \geq \nu m$ **then**
        Find $\widetilde{\mathcal{P}}_u$ such that $N(G(\widetilde{\mathcal{P}}_u); \mathcal{Y}_t) \approx \nu m$ using a dichotomy on $u \in [0,1]$, where $\widetilde{\mathcal{P}}_u = \text{Pareto}(\mathcal{P} \cup \{y_{\text{anchor}}^0 + u(y^\star - y_{\text{anchor}}^0)\})$.
      else
        Find $\widetilde{\mathcal{P}}_u$ such that $N(G(\widetilde{\mathcal{P}}_u); \mathcal{Y}_t) \approx \nu m$ using a dichotomy on $u \in [0,1]$, where $\widetilde{\mathcal{P}}_u = \text{Pareto}(\mathcal{P} \cup \{y_{\text{anchor}}^1 + u(y_{\text{anchor}}^0 - y_{\text{anchor}}^1)\} \cup \cdots \cup \{y_{\text{anchor}}^q + u(y_{\text{anchor}}^0 - y_{\text{anchor}}^q)\})$.
    $\mathcal{P} \leftarrow \widetilde{\mathcal{P}}_u$
  $\mathcal{P}_{t+1} \leftarrow \mathcal{P}$
  Generate $\mathcal{Y}_{t+1} = \big(y_{t+1,k}\big)_{1\leq k\leq m}$ uniformly distributed on $G\,(\mathcal{P}_{t+1})$ using the "Remove-Resample-Move" steps described in Algorithm 1.
  $t \leftarrow t + 1$

Remark 8 The algorithms presented in this section provide a general numerical method for the approximate computation of the expected improvement criterion, that can be used with multiple objectives, multiples constraints and possibly correlated Gaussian process models. When the objectives and constraints are independent, the decomposition introduced in Section 3.3 makes it possible to compute two integrals over spaces of lower dimension (over $\mathbb{B}_{\text{c}} \setminus H_{n,\text{c}}$ and $\mathbb{B}_{\text{o}} \setminus H_{n,\text{o}}$, respectively) instead of one integral over $G_n = \mathbb{B} \setminus H_n$. In fact, only one of the two integrals actually needs to be approximated numerically: indeed, the term ρ^{feas} of the decomposition can be calculated in closed form prior to finding

feasible solutions, and the term ρ^{unf} vanishes once a feasible observation has been made. We have taken advantage of this observation for all the numerical results presented in Section 5.

4.2 Maximization of the sampling criterion

The optimization of the sampling criterion (18) is a difficult problem in itself because, even under the unconstrained single-objective setting, the EI criterion is very often highly multi-modal. Our proposal is to conduct a discrete search on a small set of good candidates provided at each iteration by a Sequential Monte Carlo algorithm, in the spirit of Benassi et al. (2012), Li et al. (2012), Li (2012) and Benassi (2013).

The key of such an algorithm is the choice of a suitable sequence $\left(\pi_n^{\mathbb{X}}\right)_{n\geq 0}$ of probability density functions on $\mathbb{X}$, which will be the targets of the SMC algorithm. Desirable but antagonistic properties for this sequence of densities are stability—$\pi_{n+1}^{\mathbb{X}}$ should not differ too much from $\pi_n^{\mathbb{X}}$—and concentration of the probability mass in regions corresponding to high values of the expected improvement. We propose, following Benassi et al. (2012), to consider the sequence defined by

$$\begin{cases} \pi_n^{\mathbb{X}}(x) \propto 1 & \text{if } n = 0, \\ \pi_n^{\mathbb{X}}(x) \propto \mathbb{P}_n(\xi(x) \in G_n) & \text{for } n = 1, 2, \dots \end{cases}$$

In other words, we start from the uniform distribution on $\mathbb{X}$ and then we use the probability of improvement $x \mapsto \mathbb{P}_n(\xi(x) \in G_n)$ as an un-normalized probability density function.

A procedure similar to that described in Algorithms 1 is used to generate particles distributed from the target densities $\pi_n^{\mathbb{X}}$. At each step n of the algorithm, our objective is to construct a set of weighted particles

$$\mathcal{X}_n = (x_{n,k}, w_{n,k})_{k=1}^m \in (\mathbb{X} \times \mathbb{R})^m \tag{27}$$

such that the empirical distribution $\sum_k w_{n,k}\delta_{x_{n,k}}$ (where δ_x denotes the Dirac measure at $x \in \mathbb{X}$) is a good approximation, for m large enough, of the target distribution with density $\pi_n^{\mathbb{X}}$. The main difference with respect to Section 4.1 is the introduction of weighted particles, which makes it possible to deal with non-uniform target distributions. When a new sample is observed at step n, the weights of the particles are updated to fit the new density $\pi_{n+1}^{\mathbb{X}}$:

$$w_{n+1,k}^0 \propto \frac{\pi_{n+1}^{\mathbb{X}}(x_{n,k})}{\pi_n^{\mathbb{X}}(x_{n,k})}\, w_{n,k}. \tag{28}$$

The weighted sample $\mathcal{X}_{n+1}^0 = (x_{n,k}, w_{n+1,k}^0)_{1\leq k\leq m}$ is then distributed from $\pi_{n+1}^{\mathbb{X}}$. Since the densities $\pi_0, \pi_1, \dots$ become more and more concentrated as more information is obtained about the functions f and c, the regions of high values for $\pi_{n+1}^{\mathbb{X}}$ become different from the regions of high values for $\pi_n^{\mathbb{X}}$. Consequently, the weights of some particles degenerate to zero, indicating that those particles are no longer good candidates for the optimization. Then, the corresponding particles are killed, and the particles with non-degenerated weights are replicated to keep the size of the population constant. All particles are then moved randomly using an MCMC transition kernel targeting $\pi_{n+1}^{\mathbb{X}}$, in order to restore some diversity. The corresponding procedure, which is very similar to that described in Algorithm 1, is summarized in Algorithm 4.

When the densities $\pi_n^{\mathbb{X}}$ and $\pi_{n+1}^{\mathbb{X}}$ are too far apart, it may happen that the number of particles with non-degenerated weights is very small and that the empirical distribution $\sum_k w_{n,k}\delta_{x_{n+1,k}}$ is not a good approximation of $\pi_{n+1}^{\mathbb{X}}$. This is similar to the problem explained in Section 4.1, except that in the case of non uniform target densities, we use the Effective Sample Size (ESS) to detect degeneracy (see, e.g., Del Moral et al., 2006), instead of simply counting the surviving particles. When the ESS falls below a prescribed fraction of the population size, we insert intermediate densities, in a similar way to what was described in Section 4.1. The intermediate densities are of the form $\tilde{\pi}_u(x) \propto \mathbb{P}_n(\xi(x) \in \tilde{G}_u)$, with $G_{n+1} \subset \tilde{G}_u \subset G_n$. The corresponding modification of Algorithm 4 is straightforward. It is very similar to the procedure described in Algorithms 2 and 3 and is not repeated here for the sake of brevity.

Algorithm 4: Reweight-Resample-Move procedure to construct $\mathcal{X}_n$

if $n = 0$ **then**
  Set $\mathcal{X}_0 = \left(x_{0,k}, \frac{1}{m}\right)_{1 \leq k \leq m}$ with $x_{0,1}, \ldots, x_{0,m}$ independent and uniformly distributed on $\mathbb{X}$.
else
  Reweight $\mathcal{X}_{n-1}$ according to Equation (28) to obtain $\mathcal{X}_n^0$.
  Resample with a residual resampling scheme (see, e.g., Douc and Cappé, 2005) to obtain a set of particles $\mathcal{X}_n^1 = \left(x_{n,k}^1, \frac{1}{m}\right)_{1 \leq k \leq m}$.
  Move the particles with an MCMC transition kernel to obtain $\mathcal{X}_n = \left(x_{n,k}, \frac{1}{m}\right)_{1 \leq k \leq m}$.

Remark 9 A closed form expression of the probability of improvement is available in the single-objective case, as soon as one feasible point has been found. When no closed form expression is available, we estimate the probability of improvement using a Monte Carlo approximation: $1/N \sum_{k=1}^{N} \mathbb{1}_{G_n}(Z_k)$, where $(Z_k)_{1 \leq k \leq N}$ is an N-sample of Gaussian vectors, distributed as $\xi(x)$ under $\mathbb{P}_n$. A rigorous justification for the use of such an unbiased estimator inside a Metropolis-Hastings transition kernel (see the *Move* step of Algorithm 4) is provided by Andrieu and Roberts (2009).

Remark 10 It sometimes happens that a new evaluation result—say, the n-th evaluation result—changes the posterior so dramatically that the ESS falls below the threshold νm (see Algorithm 3) for the *current* region G_{n-1}. When that happens, we simply restart the Sequential Monte Carlo procedure using a sequence of transitions from $\mathcal{P}_0 = \varnothing$ to $\mathcal{P}^\star = \text{Pareto}\left(\xi(X_1), \ldots, \xi(X_n)\right)$.

5 Experiments

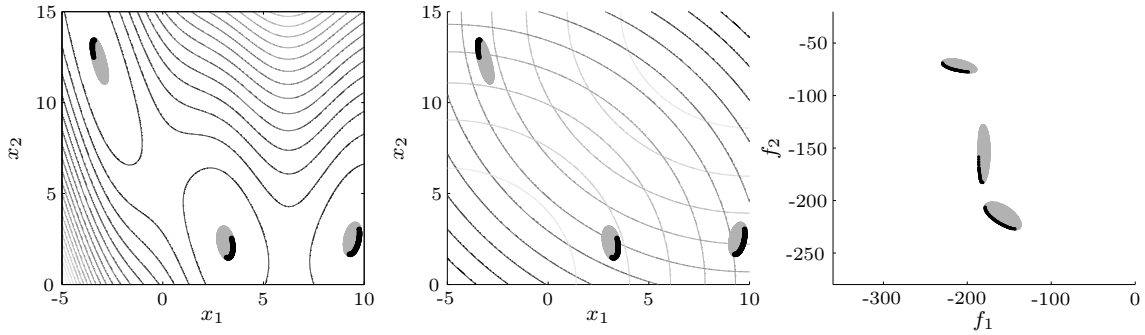
This section presents numerical experiments for both single- and multi-objective optimization problems. In the case of multi-objective optimization, we first run the algorithm on a simple two-dimensional toy problem that makes it possible to illustrate the behavior of the algorithm. In particular, we illustrate the operation of the SMC procedures for the calculation and the optimization of the sampling criterion. Then, we provide a comparison of our algorithm with other methods for solving constrained single-objective optimization problems and constrained multi-objective problems.

5.1 Settings

The BMOO algorithm has been implemented in the Matlab/Octave programming langage, using the STK toolbox (Bect et al., 2015) for the Gaussian process modeling part. All simulations results have been obtained with Matlab R2014b.

In all our experiments, the algorithm is initialized using a *maximin Latin hypercube design* consisting of $N_{\text{init}} = 3d$ evaluations. This is an arbitrary rule of thumb. A dedicated discussion about the size of initial designs can be found in Loeppky et al. (2009). The objective and constraint functions are modeled using independent Gaussian processes, with a constant but unknown mean function, and a Matérn covariance function with regularity parameter $\nu = 5/2$ (these settings are described, for instance, in Bect et al., 2012). The variance parameter σ^2 and the range parameters θ_i, $1 \leq i \leq d$, of the covariance functions are (re-)estimated at each iteration using a maximum a posteriori (MAP) estimator. Besides, we assume that the observations are slightly noisy to improve the conditioning of the covariance matrices, as is usually done in kriging implementations.

The computation of the expected improvement is carried out using the sequential Monte Carlo method described in Section 4.1, with $m = 1000$ particles. Taking advantage of Remark 8, the integration is



(a) Constraint function (b) Objective functions (c) Objective space

Fig. 5: (a) represents contour lines of the constraint function, and (b) corresponds to contour lines of the two objective functions. The three gray areas correspond to the feasible region on (a) and (b), and the image of the feasible region by the objective functions on (c). Thick dark curves correspond to the set of feasible and non-dominated solutions on (a) and (b). On (c), thick dark curves correspond to the Pareto front.

performed only on either the constraint space (prior to finding a feasible point) or the objective space (once a feasible point has been found). The parameter ν of Algorithm 3 is set to 0.2. For the mono-objective benchmark of Section 5.3, the analytical expression of the criterion, i.e., Schonlau et al.'s criterion, is used as soon as one feasible point has been found (see Remark 6).

For the optimization of the sampling criterion, the resampling step is done with a residual resampling scheme (Douc and Cappé, 2005), and we use an adaptive anisotropic Gaussian random walk Metropolis-Hastings algorithm to move the particles (Chib and Greenberg, 1995).

The bounding hyper-rectangles $\mathbb{B}_{\mathrm{o}}$ and $\mathbb{B}_{\mathrm{c}}$ are determined using the adaptive procedure described in Appendix B with $\lambda_{\mathrm{o}} = \lambda_{\mathrm{c}} = 5$.

5.2 Illustration on a constrained multi-objective problem

The proposed method is illustrated in this section on a two-dimensional two-objective toy problem, which allows for easy visualization. The optimization problem is as follows:

$$\begin{array}{ll} \text{minimize} & f_1 \text{ and } f_2\,, \\ \text{subject to} & c(x) \leq 0 \text{ and } x = (x_1, x_2) \in [-5, 10] \times [0, 15]\,, \end{array}$$

where

$$\begin{cases} f_1 : (x_1, x_2) \mapsto -(x_1 - 10)^2 - (x_2 - 15)^2, \\ \\ f_2 : (x_1, x_2) \mapsto -(x_1 + 5)^2 - x_2^2, \\ \\ c : (x_1, x_2) \;\mapsto \left(x_2 - \dfrac{5.1}{4\pi^2} x_1^2 + \dfrac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \dfrac{1}{8\pi} \right) \cos(x_1) + 9. \end{cases}$$

The set of solutions to that problem is represented on Figure 5. The feasible subset consists of three disconnected regions of relatively small size compared to that of the search space. The solution Pareto front consists of three corresponding disconnected fronts in the space of objectives. (The visualization is achieved by evaluating the objectives and constraints on a fine grid, which would not be affordable in the case of truly expensive-to-evaluate functions.)

The behavior of BMOO is presented in Figure 6. The algorithm is initialized with $5d = 10$ function evaluations. Figure 6 shows that the algorithm correctly samples the three feasible regions, and achieves

Pbm	d	q	$\Gamma(\%)$	Best	Target
g1	13	9	$4 \cdot 10^{-4}$	-15	-14.85
g3mod	20	1	10^{-4}	-0.693	-0.33
g5mod	4	5	$8.7 \cdot 10^{-2}$	5126.2	5150
g6	2	2	$6.6 \cdot 10^{-3}$	-6961.8	-6800
g7	10	8	10^{-4}	24.3	25
g8	2	2	0.86	-0.0958	-0.09
g9	7	4	0.52	680.6	1000
g10	8	6	$7 \cdot 10^{-4}$	7049.4	8000
g13mod	5	3	4.5	0.0035	0.005
g16	5	38	$1.3 \cdot 10^{-2}$	-1.916	-1.8
g18	9	13	$2 \cdot 10^{-10}$	-0.866	-0.8
g19	15	5	$3.4 \cdot 10^{-3}$	32.66	40
g24	2	2	44.3	-5.5080	-5
SR7	7	11	$9.3 \cdot 10^{-2}$	2994.4	2995
PVD4	4	3	$5.6 \cdot 10^{-1}$	5804.3	6000
WB4	4	6	$5.6 \cdot 10^{-2}$	2.3813	2.5

Table 1: Main features of the mono-objective problems of our first benchmark.

good covering of the solution Pareto front after only a few iterations. Note that no feasible solution is given at the beginning of the procedure and that the algorithm finds one after 10 iterations.

5.3 Mono-objective optimization benchmark

The first benchmark that we use to assess the performance of BMOO consists of a set of sixteen constrained single-objective test problems proposed by Regis (2014). Table 1 summarizes the main features of these problems. The input dimension d varies from 2 to 20, and the number q of constraints from 1 to 38. The problems may have linear or non-linear constraints but this information is not used by the algorithms that we use in our comparisons (all functions are regarded as black boxes). Column $\Gamma(\%)$ gives the ratio in percents of the volume of feasible region C to the volume of the search space $\mathbb{X}$. This ratio has been estimated using Monte Carlo sampling and gives an indication on the difficulty of the problem for finding a feasible point. Note that problems g1, g3mod, g6, g7, g10, g19 and in particular problem g18 have very small feasible regions. The last two columns correspond respectively to the best known feasible objective value and to target values for the optimization. The target values are the ones used in the work of Regis (2014).

BMOO is compared to two classes of algorithms. The first class consists of four local optimization algorithms: the COBYLA algorithm of Powell (1994), using the implementation proposed by Johnson (2012), and three algorithms implemented in the Matlab function `fmincon`[3], namely, an interior-point algorithm, an active-set algorithm and an SQP algorithm. Local optimization methods are known to perform well on a limited budget provided that good starting points are chosen. We think that they are relevant competitors in our context. The second class of algorithms are those proposed by Regis (2014), which are state-of-the-art—to the best of our knowledge—algorithms for constrained optimization under a limited budget of evaluations.

Each algorithm is run 30 times on each problem of the benchmark. Note that we use a random starting point uniformly distributed inside the search domain for local search algorithms, and a random initial design for BMOO, as described in Section 5.1. For the local search algorithms the maximum number of evaluations is set to two hundred times the dimension d of the problem. Concerning the algorithms proposed by Regis (2014), we simply reproduce here the figures presented by the author and we refer the reader to the original article for more details about the settings. The results are presented in Tables 2 and 3. In both tables, a solution is considered as feasible when there is no constraint violation larger than 10^{-5}.

[3] Optimization toolbox v7.1, MATLAB R2014b

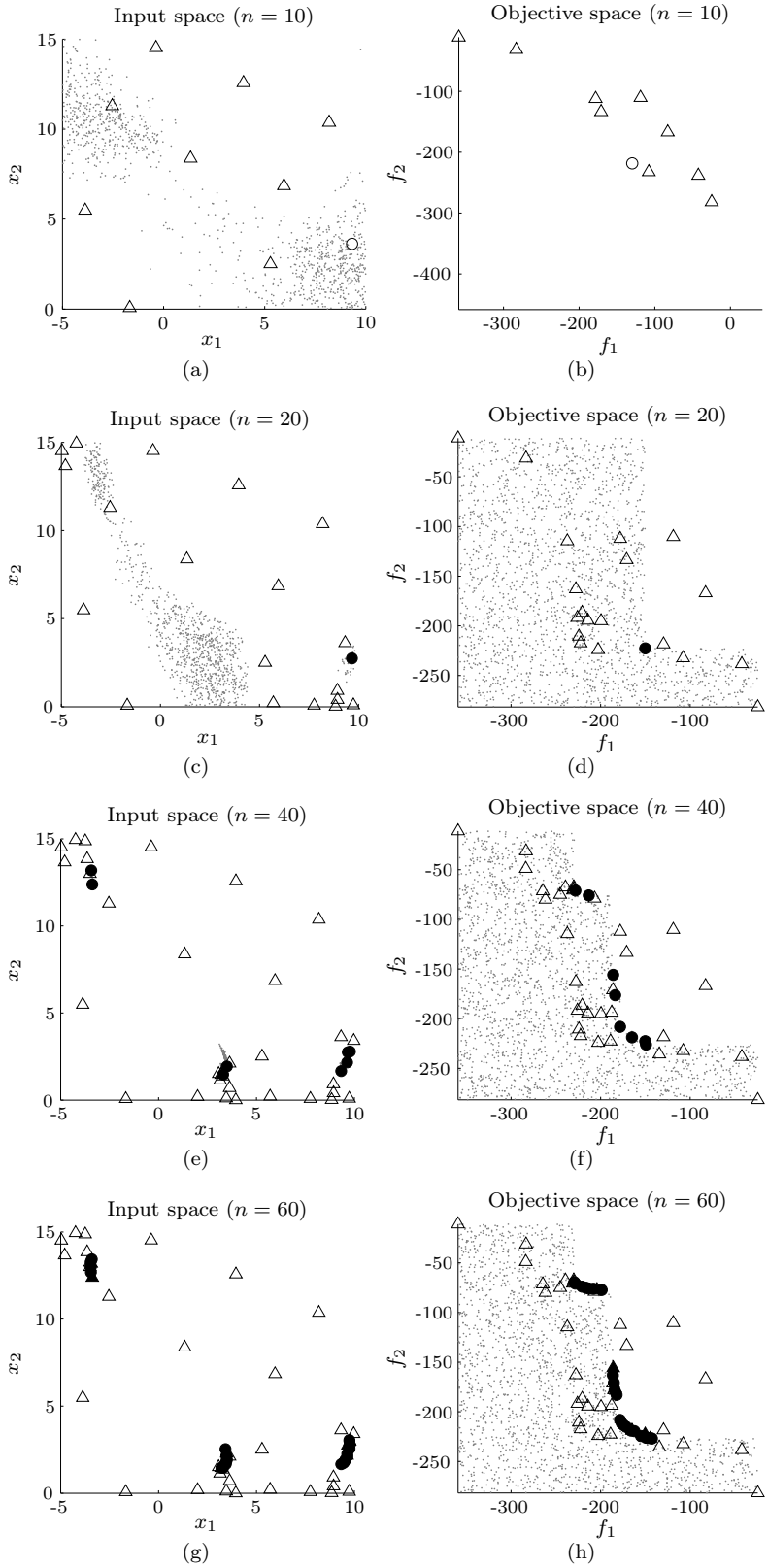


Fig. 6: Convergence of the algorithm after $n = 10, 20, 40$ and 60 evaluations. The left column shows the input space $\mathbb{X}$, whereas the right one shows the objective space $\mathbb{B}_o$. Dominated observations are represented by triangles (filled or empty), and non-dominated ones by circles (or disks). The symbols are filled for feasible points and empty otherwise. On the left column, the small dots represent the particles used for the optimization of the expected improvement (see Section 4.2). On the right column, the small dots represent the particles used for the computation of the expected improvement (see Section 4.1). Note in particular that they appear only when a feasible point has been observed: before that, the term ρ_n^{feas} (see Section 3.3) can be computed analytically.

In Table 2, we measure the performance for finding feasible solutions. For local algorithms and Regis' algorithms, only the results of the best scoring algorithm are reported in the table. Full results can be found in Appendix C. For local algorithms, the first column indicates the best scoring algorithm: Cob for the COBYLA algorithm, IP for the interior-point algorithm, AS for the active-set algorithm and SQP for the SQP algorithm. Similarly, for the algorithms proposed by Regis (2014), the first column indicates the best scoring algorithm: CG for COBRA-Global, CL for COBRA-Local and Ext for Extended-ConstrLMSRBF. The second column gives the number of successful runs—a run being successful when at least one feasible solution has been found. The third column gives the number of function evaluations that were required to find the first feasible point, averaged over the successful runs. The corresponding standard deviation is given in parentheses.

Table 3 presents convergence results. Again, for local algorithms and for those proposed by Regis (2014), the first column indicates the best scoring algorithm. The next columns give successively the number of successful runs (a run being considered successful when a feasible solution with objective value below the target value of Table 1 has been found), the average number—over successful runs—of evaluations that were required to reach the target value, and the corresponding standard deviation (in parentheses). The reader is referred to Appendix C for the full results.

BMOO achieves very good results on most test problems. It very often comes close to the best algorithm in each of the two classes of competitors, and sometimes significantly outperforms both of them—see, in particular, the results for g1, g6, g7, g9, g16 and WB4 in Table 3. However, BMOO stalls on test problems g3mod, g10, g18 and PVD4. We were able to identify the causes of theses problems and to propose remedies, which are presented in the following paragraphs. It can also be observed that BMOO is sometimes slower than the best algorithm of Regis (2014) to find a first feasible point. In almost all cases (except for g10, g18 and PVD4, which are discussed separately below), this is easily explained by the size of the initial design which is $N_{\text{init}} = 3d$ in our experiments (see Section 5.1). Further work on this issue is required to make it possible to start BMOO with a much smaller set of evaluations.

Regarding g3mod, g10 and PVD4, the difficulty lies in the presence of functions, among the objective or the constraints, which are not adequately modeled using a Gaussian process with a stationary covariance function. However, in all three cases, we can see in Table 4 that the performances of BMOO are greatly improved if a transformation of the form $f \to f^\lambda$, $\lambda > 0$, is applied to the functions that cause the problem (see Appendix D for more details). Thus, we think that the theoretical foundations of BMOO are not being questioned by these tests problems, but further work is needed on the Gaussian process models for a proper treatment of these cases. In light of the results of our experiments, one promising direction would be to consider models of the form ξ^λ, where ξ is a Gaussian process and λ is a parameter to be estimated from the evaluation results (see, e.g., Box and Cox, 1964; Snelson et al., 2004).

Regarding the g18 test problem, the difficulty stems from our choice of a sampling density derived from the probability of improvement for optimizing the expected improvement. When the number of constraints is high ($q = 13$ for the g18 test problem) and no feasible point has yet been found, the expected number of particles in the feasible regions C is typically very small with this choice of density. Consequently, there is a high probability that none of the particles produced by the SMC algorithm are good candidates for the optimization of the expected improvement. To illustrate this phenomenon, consider the following idealized setting. Suppose that $q = d$, $\mathbb{X} = [-1/2, 1/2]^q$ and $c_j : x = (x_1, \ldots, x_q) \mapsto |x_j| - \frac{\varepsilon}{2}$, $j = 1, \ldots, q$, for some $\varepsilon \in (0; 1]$. Thus, the feasible domain is $C = [-\varepsilon/2, \varepsilon/2]^q$ and the volume of the subset of $\mathbb{X}$ where exactly k constraints are satisfied is

$$V_k \approx \binom{q}{k} \varepsilon^k \, (1 - \varepsilon)^{q-k} .$$

Assume moreover that the Gaussian process models are almost perfect, i.e.,

$$\mathbb{P}_n \left(\xi_{\mathrm{c},j}(x) \leq 0 \right) \approx \begin{cases} 1, & \text{if } c_j(x) \leq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

for $j = 1, \ldots, q$. Further assume $n = 1$ with $X_1 = \left(\frac{1}{2}, \ldots, \frac{1}{2}\right)$ and observe that $\xi(X_1)$ is dominated by $\xi(x)$ for any $x \in \mathbb{X}$ (except at the corners) so that the probability of improvement $\mathbb{P}_n \left(\xi(x) \in G_1 \right)$ is

Pbm	Local (best among 4)			Regis (best among 3)			BMOO	
g1	IP	30	128.4 (27.8)	CG	30	**15.0** (0)	30	44.2 (1.9)
g3mod	IP	30	342.3 (66.3)	Ext	30	**31.2** (0.3)	30	63.1 (0.6)
g5mod	AS	30	35.0 (5.5)	CL	30	**6.4** (0.1)	30	13.0 (1.2)
g6	AS	30	29.7 (5.0)	CL	30	**10.9** (0.3)	30	**9.7** (0.7)
g7	SQP	30	107.6 (9.3)	CG	30	47.5 (4.6)	30	**38.8** (3.3)
g8	IP	30	12.1 (7.7)	CL	30	**6.5** (0.2)	30	**7.0** (0.2)
g9	IP	30	170.9 (42.9)	CG	30	**21.5** (1.9)	30	**21.8** (5.1)
g10	SQP	25	144.6 (132.3)	CG	30	**22.8** (1.5)	30	71.5 (28.1)
g13mod	IP	30	21.4 (17.1)	Ext	30	**8.6** (0.7)	30	10.5 (5.6)
g16	Cob	27	31.5 (20.4)	Ext	30	**19.6** (1.8)	30	**21.7** (7.3)
g18	SQP	30	**101.9** (19.8)	CL	30	**108.6** (6.5)	0	- (-)
g19	SQP	30	**19.7** (6.1)	CL	30	**16.5** (0.5)	30	46.4 (3.0)
g24	IP	30	4.0 (3.5)	CG	30	**1.3** (0.1)	30	2.6 (1.6)
SR7	SQP	30	27.1 (3.6)	CG	30	**9.5** (0.1)	30	22.0 (0.2)
WB4	SQP	30	76.6 (21.9)	CL	30	37.4 (5.9)	30	**19.1** (5.8)
PVD4	SQP	26	**7.6** (4.8)	CG	30	**7.9** (0.4)	30	15.7 (5.7)

Table 2: Number of evaluations to find a first feasible point. In bold, the good results in terms of average number of evaluations. We consider the results to be good if more than 20 runs where successful and the average number of evaluations is at most 20% above the best result. See Tables 10 and 12 in Appendix C for more detailed results.

Pbm	Local (best among 4)			Regis (best among 3)			BMOO	
g1	IP	20	349.7 (57.0)	CG	30	125.2 (15.3)	30	**57.7** (2.6)
g3mod	IP	30	356.9 (65.1)	Ext	30	**141.7** (8.6)	0	- (-)
g5mod	AS	30	35.8 (4.3)	CL	30	**12.9** (0.5)	30	16.3 (0.6)
g6	AS	30	29.7 (5.0)	CL	30	53.6 (14.0)	30	**13.3** (0.8)
g7	SQP	30	107.6 (9.3)	CG	30	99.8 (5.7)	30	**55.8** (2.8)
g8	IP	18	59.3 (87.0)	CL	30	**30.3** (2.8)	30	**26.3** (10.4)
g9	IP	30	179.3 (42.0)	CG	30	176.4 (26.3)	30	**61.6** (14.3)
g10	SQP	18	658.3 (316.7)	CG	29	**193.7** (-)	0	- (-)
g13mod	IP	25	**122.5** (70.3)	Ext	30	**146.4** (29.2)	30	180.3 (84.6)
g16	Cob	27	60.0 (65.2)	Ext	30	38.4 (3.6)	30	**30.3** (12.3)
g18	SQP	21	**97.5** (23.8)	CL	24	195.9 (-)	0	- (-)
g19	SQP	30	**61.3** (12.4)	CL	30	698.5 (75.3)	30	133.3 (6.2)
g24	IP	16	10.4 (5.3)	CG	30	**9.0** (0)	30	**9.9** (1.0)
SR7	SQP	30	**27.1** (3.6)	CG	30	**33.5** (1.6)	30	**29.3** (0.7)
WB4	SQP	30	78.3 (18.0)	CL	30	164.6 (12.2)	30	**44.5** (13.3)
PVD4	SQP	23	**54.7** (27.5)	CG	30	155.4 (38.2)	2	151.0 (21.2)

Table 3: Number of evaluations to reach specified target. In bold, the good results in terms of average number of evaluations. We consider the results to be good if more than 20 runs where successful and the average number of evaluations is at most 20% above the best result. See Tables 11 and 13 in Appendix C for more detailed results.

close to one everywhere on $\mathbb{X}$. As a consequence, the sampling density $\pi_1^{\mathbb{X}}$ that we use to optimize the expected improvement is (approximately) uniform on $\mathbb{X}$ and the expected number of particles satisfying exactly k constraints is $m\,V_k$. In particular, if q is large, the particles thus tend to concentrate in regions where $k \approx q\varepsilon$, and the expected number $m\,V_q$ of particles in C is small. To compensate for the decrease of V_k, when k is close to q, we suggest using the following modified sampling density:

$$\pi_n^{\mathbb{X}} \propto \mathbb{E}_n\left(K(x)!\,\mathbb{1}_{\xi(x)\in G_n}\right),$$

where $K(x)$ is the number of constraints satisfied by ξ at x. Table 5 shows the promising results obtained with this modified density on g18. Further investigations on this particular issue are left for future work.

5.4 Multi-objective optimization benchmark

The second benchmark consists of a set of eight constrained multi-objective test problems from Chafekar et al. (2003) and Deb et al. (2002). The main features of these problems are given in Table 6. The input

Pbm	Feasible		Target	
modified-g3mod	30	63.3 (0.8)	30	151.8 (12.2)
modified-g10	30	48.4 (8.0)	30	63.1 (10.4)
modified-PVD4	30	12.9 (1.6)	30	32.9 (13.2)

Table 4: Number of evaluations to find a first feasible point and to reach the target on transformed versions of the g3mod, g10 and PDV4 problems, using the BMOO algorithm.

Pbm	Feasible		Target	
g18	30	75.5 (11.5)	30	83.6 (9.1)

Table 5: Number of evaluations to find a first feasible point and to reach the target using a modified probability density function for the criterion optimization.

Pbm	d	q	p	$\Gamma(\%)$	V	y_o^{ref}
BNH	2	2	2	93,6	5249	[140; 50]
SRN	2	2	2	16,1	31820	[200; 50]
TNK	2	2	2	5,1	0,6466	[1,2; 1,2]
OSY	6	6	2	3,2	16169	[0; 80]
TwoBarTruss	3	1	2	86,3	4495	[0,06; 10^5]
WeldedBeam	4	4	2	45,5	0,4228	[50; 0,01]
CONSTR	2	2	2	52,5	3,8152	[1; 9]
WATER	3	7	5	92	0,5138	[1; 1; 1; 1,6; 3,2]

Table 6: Main features of the multi-objective problems in our benchmark.

dimension d varies from two to six, and the number q of constraints from one to seven. All problems have two objective functions, except the WATER test problem, which has five. As in Table 1, column $\Gamma(\%)$ gives an estimate of the ratio (in percents) of the volume of the feasible region to that of the search space. Column V gives the volume of the region dominated by the Pareto front[4], measured using a reference point y_o^{ref}, whose coordinates are specified in the last column.

To the best of our knowledge, state-of-the-art methods to solve multi-objective optimization problems in the presence of nonlinear constraints are based on genetic or evolutionary approaches. The most popular algorithms are probably NSGA2 (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001). Such algorithms, however, are not primarily designed to work on a limited budget of function evaluations. Note, that there are some methods that try to combine genetic/evolutionary approaches and surrogate modeling techniques (see, e.g., Emmerich et al., 2006; Jin, 2011, and references therein) to save evaluations. Nevertheless, we can safely say that the field of constrained multi-objective optimization on a limited budget of evaluations remains rather unexplored. Thus, in this article, we shall not focus on the rate of convergence, but will study instead the capacity of BMOO to build a good approximation of the Pareto front on a limited budget of evaluations.

To this end, we use the `gamultiobj` algorithm of Matlab[5], which implements a version of the NSGA2 algorithm, as a reference. For the BNH, SRN, TNK, OSY, TwoBarTruss and WeldedBeam problems, the `gamultiobj` algorithm is run with a population of size 100, over 100 generations. For the WATER problem the population size is 200. This gives reference Pareto fronts, as shown in Figures 8, 9 and 10. Then, we run BMOO with 100 function evaluations on the BNH, SRN, TNK, OSY, TwoBarTruss and WeldedBeam problems, and 200 function evaluations for the WATER problem, and compare the resulting Pareto fronts to the references.

On the BNH, SRN and CONSTR problems, our results compare with those achieved by `gamultiobj`. On the TNK problem, which has a disconnected front, `gamultiobj` fails to cover the entire Pareto

[4] This volume is the best value ever observed in all our runs of both BMOO and `gamultiobj`. Therefore, it might be slightly under-estimated.

[5] Global Optimization toolbox v3.3, MATLAB R2014b

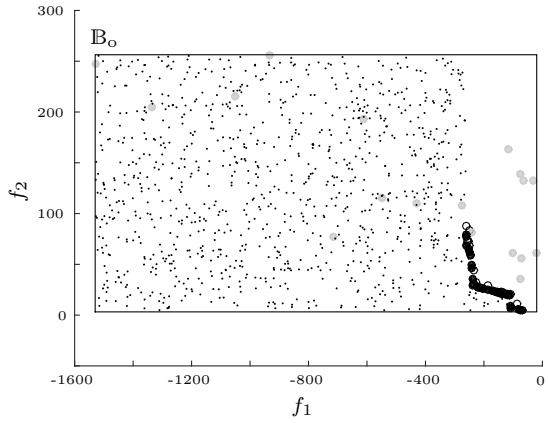


Fig. 7: An illustration, in the objective domain $\mathbb{Y}_o$, of BMOO running on the OSY test problem. The small dots are the particles used for the computation of the expected improvement. They are uniformly distributed on the non-dominated subset of $\mathbb{B}_o$. Dark disks indicate the non-dominated solutions found so far, light gray disks indicate the dominated ones.

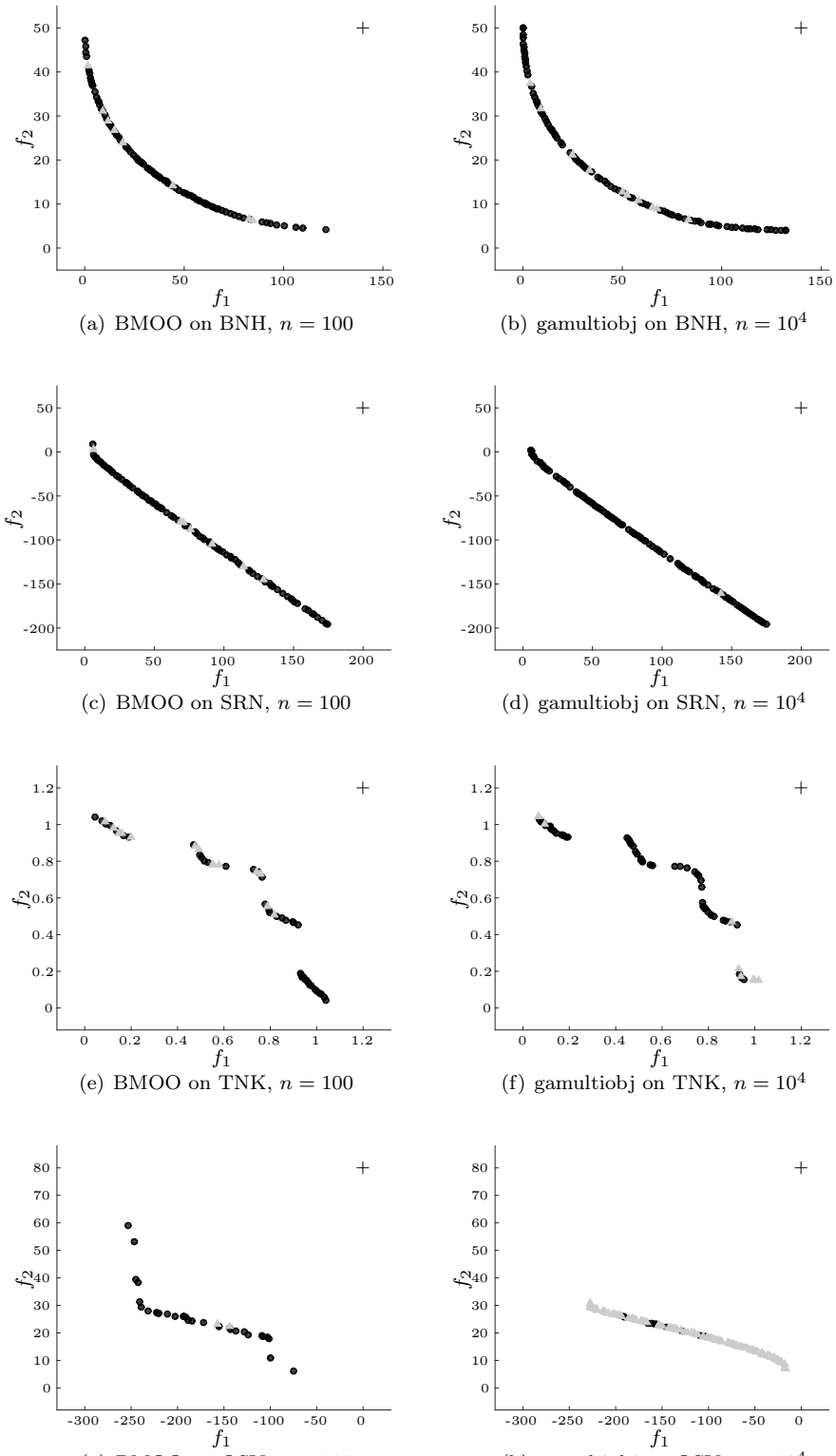
front while BMOO achieves a better spread. On the OSY test problem, both algorithms fail to produce a good set of solutions. For the BMOO algorithm, this is due to the choice of a uniform sampling density for the criterion calculation. Indeed, Figure 7 reveals that most particles do not effectively participate to the criterion calculation because they are unreachable, which leads to a poor estimation of the expected improvement value. Note that restricting the bounding box $\mathbb{B}_o$ to feasible values of the objectives would suffice to improve the quality of the expected improvement approximation in this case. On the TwoBarTruss and WeldedBeam problems, BMOO fails to approximate the Pareto front accurately. Further investigations revealed that this is due to Gaussian process modeling problems, similar to those encountered on the g3mod, g10 and PVD4 test problems (see Section 5.3). On the WATER problem, which has five objective functions, both algorithms correctly approximate the set of solutions. However, Figure 10 reveals that the solutions found by BMOO dominate those found by `gamultiobj`.

In Tables 7, 8 and 9, we also assess the number of function evaluations required to dominate respectively 90%, 95% and 99% of the reference volume V (as defined in Table 6). The results support the conclusions of the graphical analysis and show that on most problems, BMOO is able to provide a good approximation of the set of solutions with a limited number of function evaluations.

6 Conclusions and future work

In this article, a new Bayesian optimization approach has been proposed to solve multi-objective optimization problems in the presence of non-linear constraints. The constraints are handled using an extended domination rule and a new expected improvement formulation is proposed. In particular, the new formulation makes it possible to deal with problems where no feasible solution is available from the start. Several criteria from the literature are recovered as special cases.

The computation and the optimization of the new expected improvement criterion are carried out using sequential Monte Carlo sampling techniques. Indeed, the criterion takes the form of an integral over the space of objectives and constraints, for which no closed-form expression is known. Besides, the sampling criterion may be highly multi-modal, as is well known in the special case of unconstrained single-objective optimization, which makes it difficult to optimize. Our sampling techniques, which borrow ideas from the literature of structural reliability for estimating the probability of rare events, can be viewed as a contribution in itself.



(a) BMOO on BNH, $n = 100$

(b) gamultiobj on BNH, $n = 10^4$

(c) BMOO on SRN, $n = 100$

(d) gamultiobj on SRN, $n = 10^4$

(e) BMOO on TNK, $n = 100$

(f) gamultiobj on TNK, $n = 10^4$

(g) BMOO on OSY, $n = 100$

(h) gamultiobj on OSY, $n = 10^4$

Fig. 8: Solution fronts obtained with BMOO (left column) and gamultiobj (right column) on BNH (first row), SRN (second row), TNK (third row) and OSY (fourth row). For each algorithm, the light gray triangles represent non-dominated solutions that are dominated by solutions found by the other algorithm. Black disks represent solutions that are not dominated by any solution found by any of the two algorithms.

Pbm	gamultiobj		BMOO	
BNH	30	10^4 (0)	30	8.5 (0.6)
SRN	30	10^4 (0)	30	16.7 (0.9)
TNK	30	10^4 (0)	30	41.9 (26.7)
OSY	8	10^4 (0)	30	29.0 (1.7)
TwoBarTruss	30	10^4 (0)	22	90.9 (62.0)
WeldedBeam	30	10^4 (0)	28	146.5 (41.1)
CONSTR	30	10^4 (0)	30	12.4 (1.0)
WATER	0	10^4 (0)	30	48.3 (3.6)

Table 7: Results achieved by gamultiobj and BMOO on the problems of Table 6 when the target is set to 90% of the volume V. The first column contains the number of successful runs over 30 runs. The second column contains the number of function evaluations, averaged over the successful runs, with the corresponding standard deviation (in parentheses).

Pbm	gamultiobj		Our algorithm	
BNH	30	10^4 (0)	30	12.70 (0.7)
SRN	30	10^4 (0)	30	22.4 (1.0)
TNK	28	10^4 (0)	30	49.3 (26.3)
OSY	30	10^4 (0)	30	38.2 (3.4)
TwoBarTruss	30	10^4 (0)	1	234 (-)
WeldedBeam	3	10^4 (0)	2	212 (33.9)
CONSTR	30	10^4 (0)	30	19.2 (1.4)
WATER	0	10^4 (0)	30	80.7 (5.6)

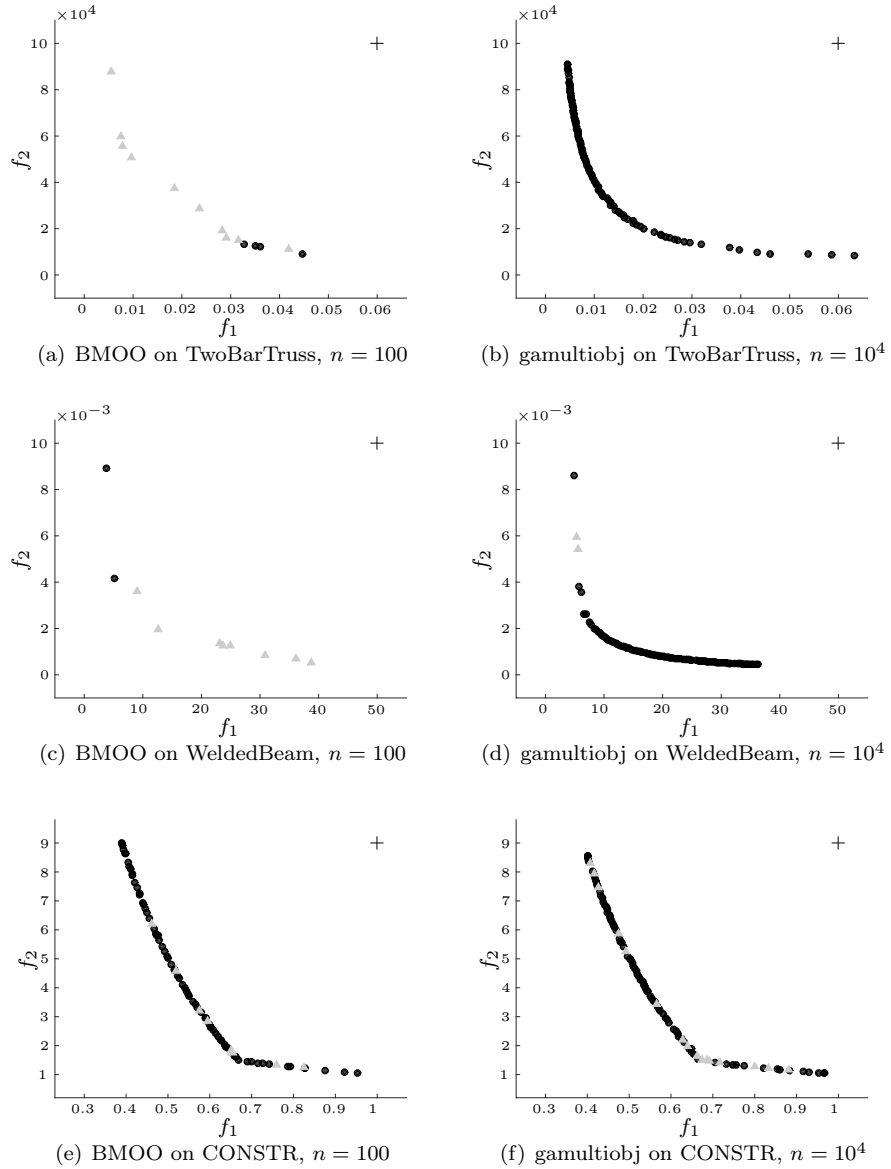
Table 8: Results achieved by gamultiobj and BMOO on the problems of Table 6 when the target is set to 95% of the volume V. See Table 7 for more information.

Pbm	gamultiobj		Our algorithm	
BNH	30	10^4 (0)	30	34.6 (1.3)
SRN	29	10^4 (0)	30	52.6 (4.1)
TNK	13	10^4 (0)	30	77.1 (22.4)
OSY	0	10^4 (0)	13	119.8 (53.0)
TwoBarTruss	22	10^4 (0)	0	> 250 (-)
WeldedBeam	30	10^4 (0)	0	> 250 (-)
CONSTR	14	10^4 (0)	30	83.5 (5.9)
WATER	0	10^4 (0)	30	139.1 (8.0)

Table 9: Results achieved by gamultiobj and BMOO on the problems of Table 6 when the target is set to 99% of the volume V. See Table 7 for more information.

We show that the resulting algorithm, which we call BMOO, achieves good results on a set of single-objective constrained test problems, with respect to state-of-the-art algorithms. In particular, BMOO is able to effectively find feasible solutions, even when the feasible region is very small compared to the size of the search space and when the number of constraints is high. In the case of multi-objective optimization with non-linear constraints, we show that BMOO is able to yield good approximations of the Pareto front on small budgets of evaluations.

Several questions are left open for future work. Regarding the BMOO algorithm itself (as opposed to the stochastic process models on which BMOO relies), our numerical results have revealed that the main limitations of the current implementation are related to the choices of sampling densities that were made, both in the input domain —as demonstrated by our poor results on the g18 test problem—and in the output domain—as shown on the OSY case. Suggestions of possible directions for the improvement of these choices were made in the paper, that will be the object of future investigations. Another direction for future research consists in working more finely on the random process models that we use in BMOO for the objective and constraint functions. Indeed, stationary Gaussian processes have proved to be a poor choice for some of the functions considered in the benchmarks, for instance in the PVD4 problem. Several ideas have been proposed in the literature—warped Gaussian processes (Snelson et al., 2004), non-stationary Gaussian processes (see Toal and Keane, 2012, and references therein), deep Gaussian



(a) BMOO on TwoBarTruss, $n = 100$
(b) gamultiobj on TwoBarTruss, $n = 10^4$
(c) BMOO on WeldedBeam, $n = 100$
(d) gamultiobj on WeldedBeam, $n = 10^4$
(e) BMOO on CONSTR, $n = 100$
(f) gamultiobj on CONSTR, $n = 10^4$

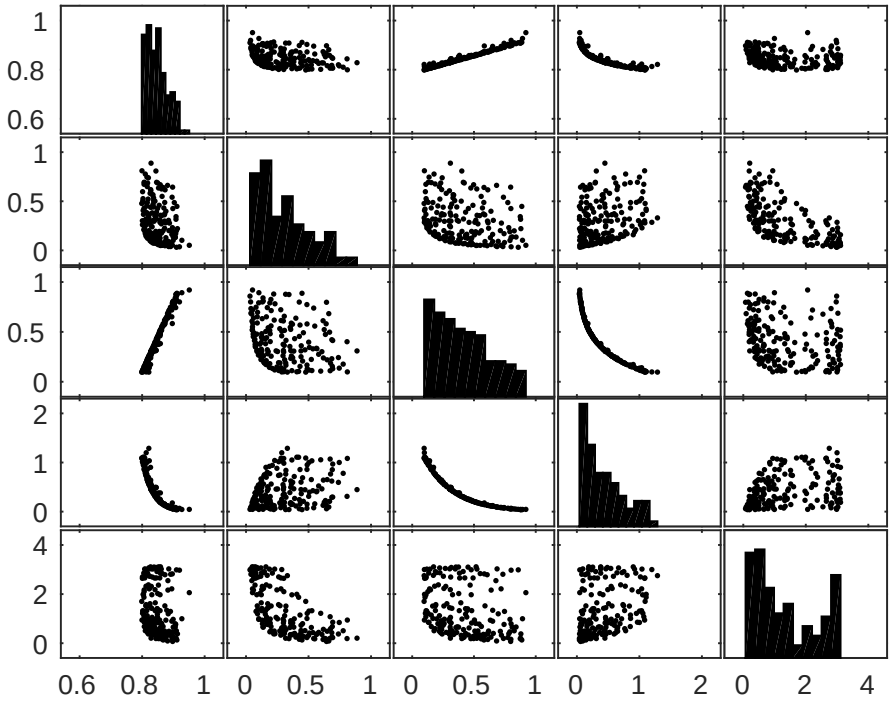
Fig. 9: Solution fronts obtained with BMOO (left column) and gamultiobj (right column) on TwoBarTruss (first row), WeldedBeam (second row) and CONSTR (third row). See Figure 8 for more information.

processes (Damianou and Lawrence, 2013), etc.—that could provide interesting directions regarding this issue.

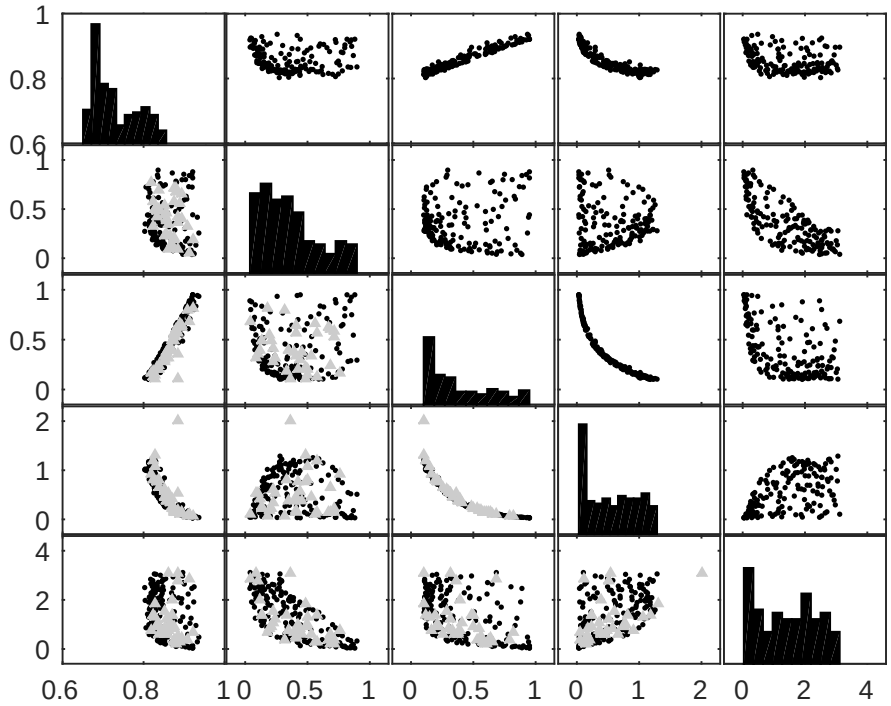
Acknowledgements This research work has been carried out within the Technological Research Institute SystemX, using public funds from the French Programme *Investissements d'Avenir*.

References

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, pages 697–725, 2009.



(a) BMOO on WATER, $n = 200$



(b) gamultiobj on WATER, $n = 2.10^4$

Fig. 10: Pareto optimal solutions obtained with (a) BMOO and (b) gamultiobj on the WATER test problem. See Figure 8 for more information.

F. Archetti and B. Betrò. A probabilistic algorithm for global optimization. *CALCOLO*, 16(3):335–343, 1979.

S.-K. Au and J. L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

J. Bect, E. Vazquez, et al. STK: a Small (Matlab/Octave) Toolbox for Kriging. Release 2.3, 2015. URL http://kriging.sourceforge.net.

R. Benassi. *Nouvel algorithme d'optimisation bayésien utilisant une approche Monte-Carlo séquentielle.* PhD thesis, Supélec, 2013.

R. Benassi, J. Bect, and E. Vazquez. Bayesian optimization using sequential Monte Carlo. In *Learning and Intelligent Optimization. 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers*, volume 7219 of *Lecture Notes in Computer Science*, pages 339–342. Springer, 2012.

M. Binois and V. Picheny. *GPareto: Gaussian Processes for Pareto Front Estimation and Optimization*, 2015. URL http://CRAN.R-project.org/package=GPareto. R package version 1.0.1.

G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22(3):795–808, 2012.

D. Chafekar, J. Xuan, and K. Rasheed. Constrained multi-objective optimization using steady state genetic algorithms. In *Genetic and Evolutionary Computation-GECCO 2003*, pages 813–824. Springer, 2003.

C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4), 2014.

S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

A. R. Conn, N. I. M. Gould, and P. Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28 (2):545–572, 1991.

A. Damianou and N. Lawrence. Deep gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 207–215, 2013.

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

P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

M. Emmerich. *Single- and multiobjective evolutionary design optimization assisted by Gaussian random field metamodels.* PhD thesis, Technical University Dortmund, 2005.

M. Emmerich and J. W. Klinkenberg. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Rapport technique, Leiden University*, 2008.

M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans*, 28(1):26–37, 1998.

M. A. Gelbart. *Constrained Bayesian Optimization and Applications.* PhD thesis, Harvard University, Graduate School of Arts and Sciences, 2015.

D. Ginsbouger and R. Le Riche. Towards Gaussian process-based optimization with finite time horizon. In *Invited talk at the 6th Autumn Symposium of the "Statistical Modelling" Research Training Group*, November 21st 2009.

R. Gramacy and H. Lee. Optimization under unknown constraints. In *Bayesian Statistics 9. Proceedings of the Ninth Valencia International Meeting*, pages 229–256. Oxford University Press, 2011.

R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. H. Lee, P. Ranjan, G. Wells, and S. M. Wild. Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics*, to appear.

J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37*, 2015.

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

S. Jeong, Y. Minemura, and S. Obayashi. Optimization of combustion chamber for diesel engine using kriging model. *Journal of Fluid Science and Technology*, 1(2):138–146, 2006.

Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

S. G. Johnson. The nlopt nonlinear-optimization package (version 2.3). *URL http://ab-initio. mit. edu/nlopt*, 2012.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *Evolutionary Computation, IEEE Transactions on*, 10(1):50–66, 2006.

J. Knowles and E. J. Hughes. Multiobjective optimization on a budget of 250 evaluations. In *Evolutionary Multi-Criterion Optimization*, pages 176–190. Springer, 2005.

H. J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Fluids Engineering*, 86(1):97–106, 1964.

L. Li. *Sequential Design of Experiments to Estimate a Probability of Failure*. PhD thesis, Supélec, 2012.

L. Li, J. Bect, and E. Vazquez. Bayesian Subset Simulation: a kriging-based subset simulation algorithm for the estimation of small probabilities of failure. In *Proceedings of PSAM 11 & ESREL 2012, 25-29 June 2012, Helsinki, Finland*. IAPSAM, 2012.

J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2008.

J. L. Loeppky, J. Sacks, and W. J. Welch. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4), 2009.

J. Mockus. On Bayesian methods of optimization. In *Towards Global Optimization*, pages 166–181. North-Holland, 1975.

J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

J. Mockus, V. Tiesis, and A. Žilinskas. The application of Bayesian methods for seeking the extremum. In L. C. W. Dixon and Gábor. P. Szegö, editors, *Towards Global Optimization*, volume 2, pages 117–129, North Holland, New York, 1978.

A. Oyama, K. Shimoyama, and K. Fujii. New constraint-handling method for multi-objective and multi-constraint evolutionary optimization. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 50(167):56–62, 2007.

J. M. Parr, A. J. Keane, A. I. J. Forrester, and C. M. E. Holden. Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization*, 44(10):1147–1166, 2012.

V. Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS), 2014, Reykjavik, Iceland.*, volume 33, pages 787–795. JMLR: W&CP, 2014.

W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *Parallel Problem Solving from Nature (PPSN X)*, volume 5199 of *Lecture Notes in Computer Science*, pages 784–794. 2008.

M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.

T. Ray, K. Tai, and K. C. Seow. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, 33(4):399–424, 2001.

R. G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, 46(2):218–243, 2014.

C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, 2013.

T. J. Santner, B. J. Williams, and W. Notz. *The design and analysis of computer experiments*. Springer Science & Business Media, 2003.

M. J. Sasena. *Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations*. PhD thesis, University of Michigan, 2002.

M. J. Sasena, P. Papalambros, and P. Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, 34(3):263–278, 2002.

M. Schonlau, W. J. Welch, and D. R. Jones. Global versus local search in constrained optimization of computer models. In *New Developments and Applications in Experimental Design: Selected Proceedings of a 1997 Joint AMS-IMS-SIAM Summer Conference*, volume 34 of *IMS Lecture Notes-Monographs Series*, pages 11–25. Institute of Mathematical Statistics, 1998.

K. Shimoyama, K. Sato, S. Jeong, and S. Obayashi. Updating kriging surrogate models based on the hypervolume indicator in multi-objective optimization. *Journal of Mechanical Design*, 135(9):094503, 2013.

E. Snelson, C. E. Rasmussen, and Z. Ghahramani. Warped gaussian processes. *Advances in neural information processing systems*, 16:337–344, 2004.

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.

J. D. Svenson and T. J. Santner. Multiobjective optimization of expensive black-box functions via expected maximin improvement. Technical report, Tech. rep., 43210, Ohio University, Columbus, Ohio, 2010.

D. J. J. Toal and A. J. Keane. Non-stationary kriging for design optimization. *Engineering Optimization*, 44(6):741–765, 2012.

E. Vazquez and J. Bect. A new integral loss function for Bayesian optimization. *arXiv preprint arXiv:1408.4622*, 2014. URL http://arxiv.org/abs/1408.4622.

J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.

T. Wagner, M. Emmerich, A. Deutz, and W. Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *Parallel Problem Solving from Nature, PPSN XI. 11th International Conference, Krakov, Poland, September 11-15, 2010, Proceedings, Part I*, volume 6238 of *Lecture Notes in Computer Science*, pages 718–727. Springer, 2010.

B. J. Williams, T. J. Santner, W. I. Notz, and J. S. Lehman. Sequential design of computer experiments for constrained optimization. In T. Kneib and G. Tutz, editors, *Statistical Modelling and Regression Structures*, pages 449–472. Physica-Verlag HD, 2010.

C. K. I. Williams and C. Rasmussen. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving Strength Pareto Evolutionary Algorithm, 2001.

A On the bounded hyper-rectangles $\mathbb{B}_o$ and $\mathbb{B}_c$

We have assumed in Section 3 that $\mathbb{B}_o$ and $\mathbb{B}_c$ are *bounded* hyper-rectangles; that is, sets of the form

$$\mathbb{B}_o = \{y \in \mathbb{Y}_o;\ y_o^{low} \leq y \leq y_o^{upp}\},$$
$$\mathbb{B}_c = \{y \in \mathbb{Y}_c;\ y_c^{low} \leq y \leq y_c^{upp}\},$$

for some $y_o^{low}, y_o^{upp} \in \mathbb{Y}_o$ and $y_c^{low}, y_c^{upp} \in \mathbb{Y}_c$, with the additional assumption that $y_{c,j}^{low} < 0 < y_{c,j}^{upp}$ for all $j \leq q$. Remember that upper bounds only where required in the unconstrained case discussed in Section 2.2. To shed some light on the role of these lower and upper bounds, let us now compute the improvement $I_1(X_1) = |H_1|$ brought by a single evaluation.

If X_1 is not feasible, then

$$|H_1| = |\mathbb{B}_o| \cdot \prod_{j=1}^{q} \left(y_{c,j}^{upp} - y_{c,j}^{low}\right)^{\gamma_j} \left(y_{c,j}^{upp} - \xi_{c,j}(X_1)\right)^{1-\gamma_j} \tag{30}$$

where $\gamma_j = \mathbb{1}_{\xi_{c,j}(X_1)\leq 0}$. It is clear from the right-hand of (30) that both $\mathbb{B}_o$ and $\mathbb{B}_c$ has to be bounded if we want $|H_1| < +\infty$ for any $\gamma = (\gamma_1, , \ldots, \gamma_q) \in \{0,1\}^q$. Note, however, that only the volume of $\mathbb{B}_o$ actually matters in this expression, not the actual values of y_o^{low} and y_o^{upp}. Equation (30) also reveals that the improvement is a discontinuous function of the observations: indeed, the j^{th} term in the product jumps from $y_{c,j}^{upp}$ to $y_{c,j}^{upp} - y_{c,j}^{low} > y_{c,j}^{upp}$ when $\xi_{c,j}(X_1)$ goes from 0^+ to 0. The increment $-y_{c,j}^{low}$ can be thought of as a reward associated to finding a point which is feasible with respect to the j^{th} constraint.

The value of $|H_1|$ when X_1 is feasible is

$$|H_1| = |\mathbb{B}_o| \cdot \left(|\mathbb{B}_c| - |\mathbb{B}_c^-|\right) + \prod_{j\leq p} \left(\min\left(\xi_{o,j}(X_1), y_{o,j}^{upp}\right) - \max\left(\xi_{o,j}(X_1), y_{o,j}^{low}\right)\right) \cdot |\mathbb{B}_c^-|, \tag{31}$$

where $\left|\mathbb{B}_c^-\right| = \prod_{j=1}^{q} \left|y_{c,j}^{low}\right|$ is the volume of the feasible subset of $\mathbb{B}_c$, $\mathbb{B}_c^- = \mathbb{B}_c \cap]-\infty; 0]^q$. The first term in the right-hand side of (31) is the improvement associated to the domination of the entire unfeasible subset of $\mathbb{B} = \mathbb{B}_o \times \mathbb{B}_c$; the second term measures the improvement in the space of objective values.

B An adaptive procedure to set $\mathbb{B}_o$ and $\mathbb{B}_c$

This section describes the adaptive numerical procedure that is used, in our numerical experiments, to define the hyper-rectangles $\mathbb{B}_o$ and $\mathbb{B}_c$. As said in Section 3.3, these hyper-rectangles are defined using estimates of the range of the objective and constraint functions, respectively. To this end, we will use the available evaluations results, together with posterior quantiles provided by our Gaussian process models on the set of candidate points $\mathcal{X}_n$ (defined in Section 4.2).

More precisely, assume that n evaluation results $\xi(X_i)$, $1 \leq i \leq n$, are available. Then, we define the corners of $\mathbb{B}_o$ by

$$\begin{cases} y_{o,i,n}^{low} = \min\left(\min_{i\leq n} \xi_{o,i}(X_i),\ \min_{x\in\mathcal{X}_n} \widehat{\xi}_{o,\,i,\,n}(x) - \lambda_o \sigma_{o,\,i,\,n}(x)\right), \\ y_{o,i,n}^{upp} = \max\left(\max_{i\leq n} \xi_{o,i}(X_i),\ \max_{x\in\mathcal{X}_n} \widehat{\xi}_{o,\,i,\,n}(x) + \lambda_o \sigma_{o,\,i,\,n}(x)\right), \end{cases} \tag{32}$$

for $1 \leq i \leq p$, and the corners of $\mathbb{B}_c$ by

$$\begin{cases} y_{c,j,n}^{low} = \min\left(0,\ \min_{i\leq n} \xi_{c,j}(X_i),\ \min_{x\in\mathcal{X}_n} \widehat{\xi}_{c,\,j,\,n}(x) - \lambda_c \sigma_{c,\,j,\,n}(x)\right), \\ y_{c,j,n}^{upp} = \max\left(0,\ \max_{i\leq n} \xi_{c,j}(X_i),\ \max_{x\in\mathcal{X}_n} \widehat{\xi}_{c,\,j,\,n}(x) + \lambda_c \sigma_{c,\,j,\,n}(x)\right), \end{cases} \tag{33}$$

for $1 \leq j \leq q$, where λ_o and λ_c are positive numbers.

C Mono-objective benchmark result tables

In Section 5.3, only the best results for both the "Local" and the "Regis" groups of algorithms were shown. In this Appendix, we present the full results. Tables 10 and 11, and Tables 12 and 13 present respectively the results obtained with the local optimization algorithms and the results obtained by Regis (2014) on the single-objective benchmark test problems (see Table 1). Table 10 and Table 11 show the performances for finding feasible solutions and for reaching the targets specified in Table 1 for the COBYLA, Active-Set, Interior-Point and SQP algorithms. Similarly, Table 12 and Table 13 show the performances for finding feasible solutions and for reaching the targets for the COBRA-Local, COBRA-Global and Extended-ConstrLMSRBF algorithms of Regis (2014).

D Modified g3mod, g10 and PVD4 test problems

We detail here the modified formulations of the g3mod, g10 and PVD4 problems that were used in Section 5.3 to overcome the modeling problems with BMOO. Our modifications are shown in boldface. The rationale of the modifications is to smooth local jumps.

Pbm	COBYLA		active-set		interior-point		sqp	
g1	30	52.3 (102.3)	30	15.0 (0.0)	30	128.4 (27.8)	30	15.0 (0.0)
g3mod	28	386.1 (645.8)	30	643.2 (248.9)	30	342.3 (66.3)	30	794.3 (53.7)
g5mod	22	30.7 (23.0)	30	35.0 (5.5)	30	41.3 (16.9)	30	38.5 (10.5)
g6	26	39.7 (12.7)	30	29.7 (5.0)	30	99.7 (14.3)	30	32.6 (5.4)
g7	28	162.4 (175.7)	30	109.4 (11.2)	30	146.0 (18.1)	30	107.6 (9.3)
g8	28	53.3 (77.1)	28	17.6 (5.0)	30	12.1 (7.7)	30	19.6 (8.5)
g9	25	95.2 (104.7)	30	313.7 (84.4)	30	170.9 (42.9)	30	194.5 (60.2)
g10	2	14.5 (3.5)	9	53.6 (41.9)	12	469.8 (393.8)	25	144.6 (132.3)
g13mod	30	53.9 (68.8)	30	74.0 (59.5)	30	21.4 (17.1)	30	69.4 (62.4)
g16	27	31.5 (20.4)	30	38.0 (15.0)	22	100.9 (160.3)	30	40.7 (17.1)
g18	26	345.0 (275.7)	30	114.5 (41.5)	30	70.3 (22.2)	30	101.9 (19.8)
g19	19	31.4 (19.5)	30	21.8 (7.5)	30	291.3 (57.9)	30	19.7 (6.1)
g24	30	7.7 (10.2)	30	5.2 (5.3)	30	4.0 (3.5)	30	5.1 (5.2)
SR7	29	30.0 (50.1)	30	27.5 (3.9)	30	78.6 (23.1)	30	27.1 (3.6)
WB4	27	71.8 (82.5)	30	125.7 (71.0)	30	93.5 (48.9)	30	76.6 (21.9)
PVD4	12	50.8 (70.2)	3	51.3 (27.7)	30	59.1 (43.5)	26	7.6 (4.8)

Table 10: Number of evaluations to find a first feasible point for the COBYLA, Active-Set, Interior-Point and SQP local optimization algorithms.

Pbm	COBYLA		active-set		interior-point		sqp	
g1	7	212.9 (225.8)	6	22.0 (7.7)	20	349.7 (57.0)	6	22.0 (7.7)
g3mod	16	1312.3 (1123.6)	24	760.5 (79.8)	30	356.9 (65.1)	30	794.3 (53.7)
g5mod	22	53.4 (20.3)	30	35.8 (4.3)	30	54.8 (11.7)	30	41.8 (7.5)
g6	26	41.0 (11.1)	30	29.7 (5.0)	30	99.7 (14.3)	30	32.6 (5.4)
g7	20	495.5 (461.3)	30	109.4 (11.2)	30	147.2 (18.2)	30	107.6 (9.3)
g8	4	79.5 (84.6)	2	30.5 (2.1)	18	59.3 (87.0)	4	55.8 (27.0)
g9	22	144.9 (143.7)	30	334.5 (84.0)	30	179.3 (42.0)	30	194.5 (60.2)
g10	0	- (-)	0	- (-)	0	- (-)	18	658.3 (316.7)
g13mod	23	191.9 (209.7)	24	153.9 (46.6)	25	122.5 (70.3)	22	147.6 (75.1)
g16	27	60.0 (65.2)	14	85.1 (41.1)	13	400.0 (242.1)	30	152.2 (53.2)
g18	14	383.0 (389.3)	21	101.0 (30.2)	21	149.1 (39.4)	21	97.5 (23.8)
g19	16	912.1 (685.8)	30	61.3 (12.4)	30	335.5 (65.4)	30	61.3 (12.4)
g24	18	17.5 (8.9)	17	14.7 (3.9)	16	10.4 (5.3)	17	16.4 (5.3)
SR7	28	62.5 (52.1)	30	27.5 (3.9)	30	80.2 (22.1)	30	27.1 (3.6)
WB4	24	247.1 (176.2)	29	162.0 (73.1)	30	168.2 (94.4)	30	78.3 (18.0)
PVD4	2	58.0 (35.4)	3	54.0 (25.1)	26	146.7 (115.2)	23	54.7 (27.5)

Table 11: Number of evaluations to reach the target for the COBYLA, Active-Set, Interior-Point and SQP local optimization algorithms.

Pbm	COBRA-Local		COBRA-Global		Extended-ConstrLMSRBF	
g1	30	15.0 (0.0)	30	15.0 (0.0)	30	19.1 (0.4)
g3mod	30	23.5 (0.2)	30	23.5 (0.2)	30	31.2 (0.3)
g5mod	30	6.4 (0.1)	30	6.4 (0.1)	30	9.6 (0.3)
g6	30	10.9 (0.3)	30	10.9 (0.3)	30	11.9 (0.2)
g7	30	47.5 (4.6)	30	47.5 (4.7)	30	39.8 (2.9)
g8	30	6.5 (0.2)	30	6.5 (0.2)	30	5.2 (0.2)
g9	30	21.5 (1.9)	30	21.5 (1.9)	30	23.1 (2.3)
g10	30	22.8 (1.5)	30	22.8 (1.5)	30	51.1 (6.5)
g13mod	30	9.4 (0.8)	30	9.4 (0.8)	30	8.6 (0.7)
g16	30	14.7 (2.4)	30	14.7 (2.4)	30	19.6 (1.8)
g18	30	108.6 (6.5)	30	108.6 (6.5)	30	122.0 (5.6)
g19	30	16.5 (0.5)	30	16.5 (0.5)	30	20.8 (0.8)
g24	30	1.3 (0.1)	30	1.3 (0.1)	30	1.3 (0.1)
SR7	30	9.5 (0.1)	30	9.5 (0.1)	30	12.4 (0.4)
WB4	30	37.4 (5.9)	30	37.4 (5.9)	30	25.0 (4.1)
PVD4	30	7.9 (0.4)	30	7.9 (0.4)	30	10.4 (0.7)

Table 12: Number of evaluations to find a first feasible point for the COBRA-Local, COBRA-Global and Extended-ConstrLMSRBF optimization algorithms. These results are taken from (Regis, 2014).

Pbm	COBRA-Local		COBRA-Global		Extended-ConstrLMSRBF	
g1	7	387.8 (-)	30	125.2 (15.3)	0	$>$ 500 (-)
g3mod	6	451.1 (-)	6	440.0 (-)	30	141.7 (8.6)
g5mod	30	12.9 (0.5)	30	16.6 (1.8)	30	40.3 (1.4)
g6	30	53.6 (14.0)	30	62.5 (10.5)	26	101.2 (-)
g7	30	199.5 (20.7)	30	99.8 (5.7)	30	264.5 (34.2)
g8	30	30.3 (2.8)	30	31.2 (2.5)	30	46.2 (6.2)
g9	28	275.5 (-)	30	176.4 (26.3)	29	294.0 (-)
g10	30	276.4 (43.6)	29	193.7 (-)	24	394.3 (-)
g13mod	30	221.7 (35.6)	30	169.0 (19.1)	30	146.4 (29.2)
g16	30	38.8 (9.3)	30	46.3 (13.5)	30	38.4 (3.6)
g18	24	195.9 (-)	23	212.8 (-)	21	276.0 (-)
g19	30	698.5 (75.3)	30	850.9 (70.6)	0	$>$ 1000 (-)
g24	30	9.0 (0.0)	30	9.0 (0.0)	30	91.9 (6.0)
SR7	30	35.0 (2.7)	30	33.5 (1.6)	0	$>$ 500 (-)
WB4	30	164.6 (12.2)	30	202.0 (13.0)	30	238.6 (20.0)
PVD4	28	212.2 (-)	30	155.4 (38.2)	29	263.5 (-)

Table 13: Number of evaluations to reach the target for the COBRA-Local, COBRA-Global and Extended-ConstrLMSRBF optimization algorithms. These results are taken from (Regis, 2014).

- modified-g3mod problem

$$\begin{cases} f(x) = -\text{plog}((\sqrt{d})^d \prod_{i=1}^{d} x_i)^{\mathbf{0.1}} \\ c(x) = (\sum_{i=1}^{d} x_i^2) - 1 \end{cases}$$

- modified-g10 problem

$$\begin{cases} f(x) = x_1 + x_2 + x_3 \\ c_1(x) = 0.0025(x_4 + x_6) - 1 \\ c_2(x) = 0.0025(x_5 + x_7 - x_4) - 1 \\ c_3(x) = 0.01(x_8 - x_5) - 1 \\ c_4(x) = \text{plog}(100x_1 - x_1x_6 + 833.33252x_4 - 83333.333)^{\mathbf{7}} \\ c_5(x) = \text{plog}(x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5)^{\mathbf{7}} \\ c_6(x) = \text{plog}(x_3x_5 - x_3x_8 - 2500x_5 + 1250000)^{\mathbf{7}} \end{cases}$$

- modified-PVD4 problem

$$\begin{cases} f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\ c_1(x) = -x_1 + 0.0193x_3 \\ c_2(x) = -x_2 + 0.00954x_3 \\ c_3(x) = \text{plog}(-\pi x_3^2x_4 - 4/3\pi x_3^3 + 1296000)^{\mathbf{7}} \end{cases}$$

Note that the above defined problems make use of the plog function defined below (see Regis (2014)).

$$\text{plog}(x) = \begin{cases} \log(1+x) & \text{if } x \geq 0 \\ -\log(1-x) & \text{otherwise} \end{cases}$$