Une approche Bayésienne pour l'optimisation multi-objectif sous contraintes

Paul Feliot

Une approche Bayésienne pour l'optimisation multi-objectif sous contraintes

Thèse de doctorat de l'université Paris Saclay préparée à CentraleSupélec

École doctorale n° 580

Sciences et technologies de l'information et de la communication – STIC

Spécialité de doctorat: Traitement du signal

Thèse présentée et soutenue à Gif-sur-Yvette, le 12 Juillet 2017, par

Paul FELIOT

Composition du jury

M.	Patrice Aknin	IRT SystemX	Éxaminateur
Mme	Anne Auger	INRIA, Ecole Polytechnique	Éxaminateur
M.	Julien Bect	CentraleSupélec	Encadrant
M.	Sébastien Da Veiga	Safran Tech	Éxaminateur
M.	David Ginsbourger	Université de Bern	Rapporteur
M.	Serge Gratton	ENSEEIHT, CERFACS	Président
M.	Luc Pronzato	CNRS, UNSA	Rapporteur
M.	Emmanuel Vazquez	CentraleSupélec	Directeur de thèse

NNT Number: 2017SACLC045

Remerciements

Je remercie en premier lieu mes superviseurs Julien Bect et Emmanuel Vazquez, pour leur implication et pour leurs conseils toujours avisés. Je ne saurais dire ici tout ce que vous m'avez apporté. Je me contenterais donc de vous exprimer ma plus profonde gratitude et j'espère que l'avenir nous permettra de travailler à nouveau ensembles car j'ai encore beaucoup à apprendre de vous.

Je remercie ensuite mes collègues et amis de l'IRT SystemX, trop nombreux pour les nommer. J'ai beaucoup apprécié ces trois années passées ensembles et je vous souhaite toute la réussite que vous méritez. Notre amitié perdurera j'en suis certain.

Je remercie aussi les différents collaborateurs avec qui j'ai eu l'occasion de travailler; Nabil R., Abdelkader O., Sébastien D. V., Christian R., Emmanuel C., Caroline S., Amin E. B., Yves T. et Sophie F. J'ai véritablement apprécié travailler avec vous et je vous souhaite le meilleur pour la suite.

Enfin, je souhaite remercier chaleureusement ma famille, et en particulier ma mère et ma compagne, pour leur amour et leur soutien au quotidien.

Résumé

Ces travaux de thèse portent sur l'optimisation multi-objectif de fonctions à valeurs réelles sous contraintes d'inégalités. En particulier, nous nous intéressons à des problèmes pour lesquels les fonctions objectifs et contraintes sont évaluées au moyen d'un programme informatique nécessitant potentiellement plusieurs heures de calcul pour retourner un résultat. Dans ce cadre, il est souhaitable de résoudre le problème d'optimisation en utilisant le moins possible d'appels au code de calcul. Par ailleurs, nous nous intéressons à des problèmes d'optimisation potentiellement fortement contraints, c'est à dire des problèmes pour lesquels satisfaire simultanément l'ensemble des contraintes est difficile. Ce type de problème est caractéristique des problèmes d'optimisation de systèmes complexes et met en défaut de nombreux algorithmes d'optimisation.

Nous proposons dans cette thèse un algorithme d'optimisation Bayésienne baptisé BMOO. Cet algorithme encode un nouveau critère d'amélioration espérée spécifiquement développé afin d'être applicable à des problèmes fortement contraints et/ou avec de nombreux objectifs. Ce critère s'appuie sur une fonction de perte mesurant le volume de l'espace dominé par les observations courantes, ce dernier étant défini au moyen d'une règle de domination étendue permettant de comparer des solutions potentielles à la fois selon les valeurs des objectifs et des contraintes qui leurs sont associées. Le critère ainsi défini généralise plusieurs critères classiques d'amélioration espérée issus de la littérature au cas de l'optimisation multi-objectif sous contraintes d'inégalités.

Ce critère prend la forme d'une intégrale sur l'espace joint des objectifs et des contraintes qui n'est pas calculable analytiquement dans le cas général. Par ailleurs, il doit être maximisé à chaque itération de l'algorithme afin de sélectionner le prochain point d'évaluation; maximisation qui est connue pour être difficile car les critères d'amélioration espérée ont tendance à être multimodaux. Afin de pallier ces difficultés, nous proposons dans cette thèse des algorithmes de Monte-Carlo séquentiel dans la lignée de travaux précédemment réalisés par Benassi (2013) dans le cas de l'optimisation globale sans contraintes. En particulier, nous proposons une densité L_2–optimale pour le calcul du nouveau critère pour un ensemble de points candidats, et une densité dédiée à l'optimisation du critère pour des problèmes fortement contraints.

Quatre extensions de l'algorithme sont par ailleurs proposées, ces dernières pouvant être vues comme des contributions indépendantes. Tout d'abord, BMOO est généralisé à des problèmes définis sur des espaces de recherche non-hypercubiques, définis par exemple par une fonction d'appartenance ou par des contraintes peu coûteuses à évaluer, ainsi qu'à des problèmes ayant des contraintes cachées. Ces dernières apparaissent, par exemple, lorsque le code de calcul utilisé pour évaluer les fonctions du problème ne permet pas d'obtenir un résultat pour certaines régions

de l'espace de recherche. Par ailleurs, afin de tirer avantage des moyens de calcul parallèle lorsque ceux-ci sont disponibles, une version multi-point de l'algorithme est proposée. Enfin, un critère d'amélioration espérée permettant d'orienter la recherche de solutions optimales vers des régions choisies par l'utilisateur est finalement proposé. Ce critère permet à l'expert métier d'influencer le processus d'optimisation afin d'obtenir des solutions plus pertinentes.

L'algorithme proposé obtient de meilleurs résultats que des algorithmes d'optimisation "état de l'art" sur des problèmes d'optimisation à la fois mono- et multi-objectifs issus de la littérature. Nous montrons qu'il peut être appliqué avec de bons résultats et une bonne répétabilité sur un large ensemble de problèmes. En particulier, l'algorithme permet de résoudre des problèmes fortement contraints et/ou faisant état de nombreux objectifs, ce qui était l'objectif initial.

BMOO est également appliqué avec succès à quatre problèmes représentatifs des types de problèmes d'optimisation rencontrés dans l'industrie. Il est appliqué au dimensionnement du système de régulation d'air d'un avion commercial (collaboration avec Airbus Group Innovation), au dimensionnement de la chaîne de traction d'un véhicule électrique (collaboration avec Renault), au paramétrage optimal d'un contrôleur de ligne de visée (collaboration avec Safran Electronics & Defense), ainsi qu'au dimensionnement d'une aube de soufflante de turbomachine (collaboration avec Safran Aircraft Engines et Cénaéro). Il est montré en particulier que les extensions sus-mentionnées sont pertinentes au regard de ce type de problèmes d'optimisation.

Certaines limitations intrinsèques rendent cependant BMOO inefficace sur certains types de problèmes d'optimisation, qui sont illustrés dans cette thèse. Tout d'abord, BMOO n'est pas adapté à la résolution de problèmes ayant des fonctions non-stationnaires. En effet, l'algorithme utilise des modèles de processus Gaussiens et la stationnarité des objectifs et des contraintes est une des hypothèses de modélisation qui sont faites. Lorsque celle-ci n'est pas respectée, les modèles ne permettent pas une optimisation efficace. Nous montrons cependant que dans certains cas, l'utilisation de transformations simples permet de rendre stationnaires certaines fonctions qui ne le sont pas à l'origine, et donc d'utiliser BMOO de manière efficace. Par ailleurs, l'algorithme utilise l'hypervolume de la région dominée comme fonction de perte. Or, l'hypervolume a tendance à favoriser certaines régions du front de Pareto davantage que d'autres, en fonction de sa courbure. De ce fait, BMOO est sujet à un biais intrinsèque et il peut arriver que sur certains problèmes, typiquement des problèmes pour lesquels le front de Pareto a des concavités, la distribution des solutions obtenues par l'algorithme ne représente pas de manière satisfaisante certaines régions du front.

Contents

Chapter 1

Introduction

1.1 Context

1.1.1 Industrial design of complex systems

The object of this thesis is the optimal design of complex systems. As an introductory example, consider the design of a commercial aircraft turbomachine. A turbomachine is a complex system made of several interacting subsystems. The main components of a typical turbomachine are represented on Figure 1.1.

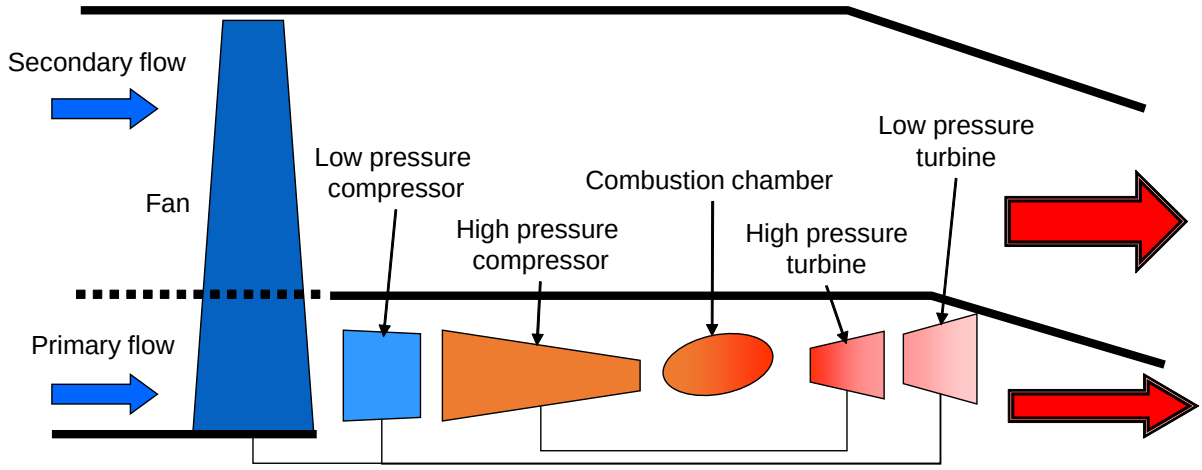


Figure 1.1: Global architecture of a turbomachine.

When designing such a system, a manufacturer has to make several design choices. What should be the shape of the combustion chamber? How many compressor stages are required to achieve a given level of performance? What materials should the fan blades be composed of? What is the inner blade radius of the first stage of the high pressure turbine? Etc. Those choices are often made using past experience in designing similar systems and performance assessment studies. An established practice to assess the performances of a given design, is to rely on numerical models of the physical system. This is in general less costly and less time-consuming than prototyping. Besides, using numerical models makes it possible to consider far more candidate designs.

A common approach to cast a design problem into a mathematical framework is to formulate the decision-maker wishes in terms of objectives and constraints. In the turbomachine example, objectives for the design of the combustion chamber could be to minimize fuel consumption or to maximize the mixing of fuel and air inside the chamber. One could also try to do both simultaneously. Constraints could be to keep the temperature and pressure inside the chamber below some threshold value to avoid damaging the casing. Naturally, those threshold values may depend on the design of the casing itself.

Within this framework, a notion of optimal design can be introduced: a design is considered optimal if it respects all the constraints and achieves an optimal trade-off between the objectives. From a mathematical point of view, the problem consists in finding an approximation of the set

$$\Gamma = \{x \in \mathbb{X} : c(x) \leq 0 \text{ and } \nexists\, x' \in \mathbb{X} \text{ s.t. } c(x') \leq 0 \text{ and } f(x') \prec f(x)\}, \tag{1.1}$$

where $\mathbb{X}$ is a design domain, $c = (c_i)_{1 \leq i \leq q}$ is a vector of constraint functions, $f = (f_j)_{1 \leq j \leq p}$ is a vector of objective functions to be minimized, and $\prec$ is a partial order relation. The elements of Γ correspond to design solutions that both respect the constraints and achieve optimal trade-off between the objectives, as formulated by the decision-maker[1].

In the setting that we consider, for a given design $x \in \mathbb{X}$, the values $f(x)$ and $c(x)$ in (1.1) correspond to the outputs of a numerical model that may involve the resolution of partial differential equations, meshing steps or large matrix inversions. The affordable number of evaluations of f and c is therefore limited by the computational cost. When it is high, finding Γ is a difficult problem.

1.1.2 A brief literature review of continuous optimization

In the literature, several algorithms have been proposed for solving the optimization problem (1.1). For the sake of clarity, we limit the scope of our review to the continuous optimization of deterministic functions, i.e. we consider problems where $\mathbb{X}$ is a subset of $\mathbb{R}^d$, d being the number of design variables, and for which the vectors $f(x)$ and $c(x)$ for some $x \in \mathbb{X}$ are deterministic (as opposed to stochastic, see e.g. Fu (2002); Tekin and Sabuncuoglu (2004); Kleijnen (2008) and references therein). Moreover, we do not consider optimization methods that require assumptions on the structure of the functions of the problem, such as convexity or linearity for example, and we do not consider optimization problems with equality constraints. These give rise to a specific literature that falls out of the scope of this thesis. See, e.g., the book of Bonnans et al. (2006) for a broader discussion on continuous optimization.

Local and global optimization

Optimization problems with only one objective function fall into the category of the single-objective optimization problems. This is probably the most documented category and the first that was addressed in the literature. The solution to a single-objective problem is often a single point called the *global optimizer*.

Single-objective problems can be solved using local and global optimization algorithms. Given a starting point, local optimization algorithms perform a local search and hopefully converge to a local optimum of the objective function. Algorithms in this class usually have a good convergence rate and require few objective function evaluations. In this category, we find first and second order gradient-based optimizers such as the *method of steepest descent*, the *conjugate gradient method*, the *modified Newton's method* or the *quasi-Newton method*, and derivative-free optimizers such as the *Direct Search* algorithm of Hooke and Jeeves (1961), the *Trust-Region* algorithm of Powell (1964), the *Simplex* algorithm of Nelder and Mead (1965) or the *Generalized Pattern Search* algorithm of Torczon (1997). For more details on these approaches, the reader is referred to the book of Nocedal and Wright (2006) and references therein.

Global optimization algorithms on the other hand seek a global optimum of the objective function. They are often population-based and/or introduce some randomness in the optimiza-

[1] Note that different formulations of the optimization problem can lead to different optimal solutions.

tion process to escape local optima. Among this category of optimization algorithms we find, for example, the *Simulated Annealing* algorithm of Kirkpatrick et al. (1983), the *Hill Climbing* algorithm of Russell et al. (2003), the *DIRECT* algorithm of Jones et al. (1993), the *Multilevel Coordinate Search* algorithm of Huyer and Neumaier (1999), several genetic and evolutionary algorithms (see e.g. Back (1996)), random search algorithms (see e.g. Zhigljavsky (2012)), Bayesian optimization algorithms, such as the *EGO* algorithm of Jones et al. (1998) or the *IAGO* algorithm of Villemonteix et al. (2009), and surrogate-based optimization algorithms such as the *COBRA* algorithms of Regis (2014). Local optimization algorithms can also be made global by running them several times with different starting points (multi-start approach). For more details on global optimization methods, the reader is referred to the books of Torn and Zilinskas (1989); Weise (2009); Zhigljavsky (2012) and Nocedal and Wright (2006).

Multi-objective optimization

Optimization problems with more than one objective are called multi-objective optimization problems. The term many-objective optimization problems is also used to refer to multi-objective problems with more than two objectives. Unlike single-objective problems, the solution to a multi-objective optimization problem is often a set of optimal solutions called a Pareto front.

In the literature, a distinction is made between algorithms that look for a single solution on the Pareto front and algorithms that build an approximation of the Pareto front. For both categories, a survey of approaches is provided by Marler and Arora (2004). See also the books of Miettinen (2012) and Collette and Siarry (2013) for more in-depth discussions on multi-objective optimization.

The most popular algorithms for approximating Pareto fronts are probably genetic and evolutionary algorithms. Since they are population-based, they are well-suited to approximating a set of solutions. A comprehensive review of genetic and evolutionary multi-objective optimization algorithms is provided by Coello (2000) and Coello et al. (2002).

In the Bayesian optimization literature, algorithms for approximating Pareto fronts have been proposed by Knowles (2006); Svenson (2011); Keane (2006); Hernández-Lobato et al. (2015) and Emmerich et al. (2006), among others. Compared to genetic and evolutionary approaches, these approaches usually require fewer objective functions evaluations, which makes them particularly interesting in our context.

Constraint handling

Most of the above cited algorithms can be extended to handle constrained optimization problems, i.e. problems with at least one constraints. The most popular approach is to penalize the objective function(s) by a quantity related to the constraints violation. Lagrangian formulations for example fall in this category.

Among the class of local optimization algorithms that can handle constraints, let us cite, for example, the *Sequential Quadratic Programming* algorithm of Han (1977), the *COBYLA* algorithm of Powell (1994), the *Interior-Point* algorithm of Byrd et al. (1999) or the *Mesh Adaptive*

Direct Search algorithm of Audet and Dennis Jr (2006). For genetic and evolutionary single-objective optimization algorithms, a comprehensive review of constraint-handling techniques is provided by Mezura-Montes and Coello (2011). More generally, see the book of Nocedal and Wright (2006) for a review of constraint-handling approaches in single-objective optimization.

As regards constrained multi-objective optimization, most of the recent literature comes from the genetic and evolutionary communities. Popular algorithms for solving constrained multi-objective problems are the *NSGA2* algorithm of Deb et al. (2002) or the *SPEA2* algorithm of Zitzler et al. (2002). For more details about this class of approaches, the reader is referred to the book of Deb (2001). In the Bayesian optimization literature, constrained multi-objective optimization algorithms have been proposed by Emmerich et al. (2006); Garrido-Merchán and Hernández-Lobato (2016).

Gradient-based optimization

When gradient information is available, which happens for example when adjoint solvers are used (see e.g. Giles and Pierce (2000)), it is often advantageous to use it to guide the search for optimal solutions. In particular when the number d of variables is large, gradient information can prove invaluable to focus the search in the right direction and solve the problem using few functions evaluations.

Note that in the case where gradients are not given, they can still be estimated (see e.g. Nocedal and Wright (2006)). However, gradient approximation methods usually scale unfavourably with the dimension of the problem (d evaluations are required to estimate a gradient using finite differences), which often renders them impractical when d is large (say $d > 10$) and the functions of the problem are expensive to evaluate.

Population-based optimization

Population-based algorithms such as random search algorithms (see e.g. Zhigljavsky (2012)), genetic and evolutionary algorithms (see e.g. Coello (2000); Coello et al. (2002)) or Estimation of distribution algorithms (see e.g. Hauschild and Pelikan (2011)) also form an important subclass of the derivative-free optimization algorithms which has gained in popularity over the last decades.

One advantage of population-based algorithms is that they are often robust to difficult "landscapes" (involving for example discontinuities, irregularities or multiple modes) and high dimensional input spaces (see, e.g., Hansen and Kern (2004)). In a sense, it can be said that they compensate for the lack of information about the gradients and structure of the functions of the problem by using statistics (see, e.g., Bäck (1996)). This usually comes at the expense of many functions evaluations though.

Model-based optimization

Many derivative-free optimization algorithms are model-based, in the sense that they rely on a mathematical model to guide the search for optimal solutions. For example, it can be a statistical model as in Bayesian optimization algorithms (see, e.g., Mockus (2012); Locatelli (1997); Jones

et al. (1998)), a local approximation model as in the *COBYLA* algorithm of Powell (1994), or a global approximation model, as in surrogate-based optimization algorithms (see e.g. Wang and Shan (2007); Koziel et al. (2011); Queipo et al. (2005); Booker et al. (1999); Regis (2016)).

Algorithms of this class usually require few functions evaluations. However, some degree of smoothness from the functions of the problem is often necessary and they do not usually scale favourably with the dimension. As such, their use remains limited to a certain class of problems.

1.2 Background literature

1.2.1 Bayesian optimization

For this thesis, the choice was made to take a Bayesian approach to the optimization problem (1.1). Historically, this approach was introduced by Kushner (1964) and developed by Mockus (1975), Mockus et al. (1978), Archetti and Betrò (1979) and Mockus (2012). It was later made popular by Jones et al. (1998) who proposed the EGO algorithm, which is one of the most efficient existing algorithms for solving global optimization problems with a small number of function evaluations.

To present the Bayesian approach to optimization, it is useful to recall the Bayes rule, which states that given a statistical model where ξ is a quantity of interest and $\mathcal{I}$ represents available information about ξ, the posterior probability of ξ knowing the information $\mathcal{I}$ is proportional to the likelihood of the information $\mathcal{I}$ assuming ξ times the prior probability that is placed on ξ:

$$p(\xi|\mathcal{I}) \propto p(\mathcal{I}|\xi)\, p(\xi). \tag{1.2}$$

In a Bayesian optimization setting, it is assumed that the functions of the problem are sample paths of a vector valued random process ξ. Then, $p(\xi)$ represents a priori knowledge about these functions, such as regularity for example. Usually, stationary Gaussian process priors are used because of their flexibility and because they yield good results in practice (see e.g. Williams and Rasmussen (2006)). The information $\mathcal{I}$ is made of the past observations of ξ. In a sequential optimization procedure, assuming that a set $\mathbf{X}_n = (X_1, \ldots, X_n) \in \mathbb{X}^n$ of n observations have been made at time n, then $\mathcal{I} = \mathcal{I}_n$ is the information $\mathbf{Y}_n = \xi(\mathbf{X}_n)$, where $\mathbf{Y}_n = (Y_1, \ldots, Y_n) \in \mathbb{R}^{p+q}$ is the vector of the observed values. Under this framework, $p(\xi|\mathcal{I}_n)$ is the posterior distribution of ξ, conditional on the past observations. An illustration of this approach is proposed in Figure 1.2.

In the Bayesian optimization literature, various criteria have been proposed to select the evaluation points $(X_1, X_2, \ldots)$. In this thesis, the choice was made to focus on the expected improvement (EI) sampling criterion (see, e.g., Jones et al. (1998)) but other approaches based on *stepwise uncertainty reduction* (see e.g. Villemonteix et al. (2009); Bect et al. (2012); Chevalier et al. (2014a); Picheny (2014b); Hennig and Schuler (2012); Hernández-Lobato et al. (2015)) constitute alternative directions that may have been taken.

Consider the global optimization setting where the objective is to find the minimum m of a real valued function $f : \mathbb{X} \to \mathbb{R}$. The quality of an optimization strategy $\underline{X} : f \mapsto (X_1, X_2, \ldots)$

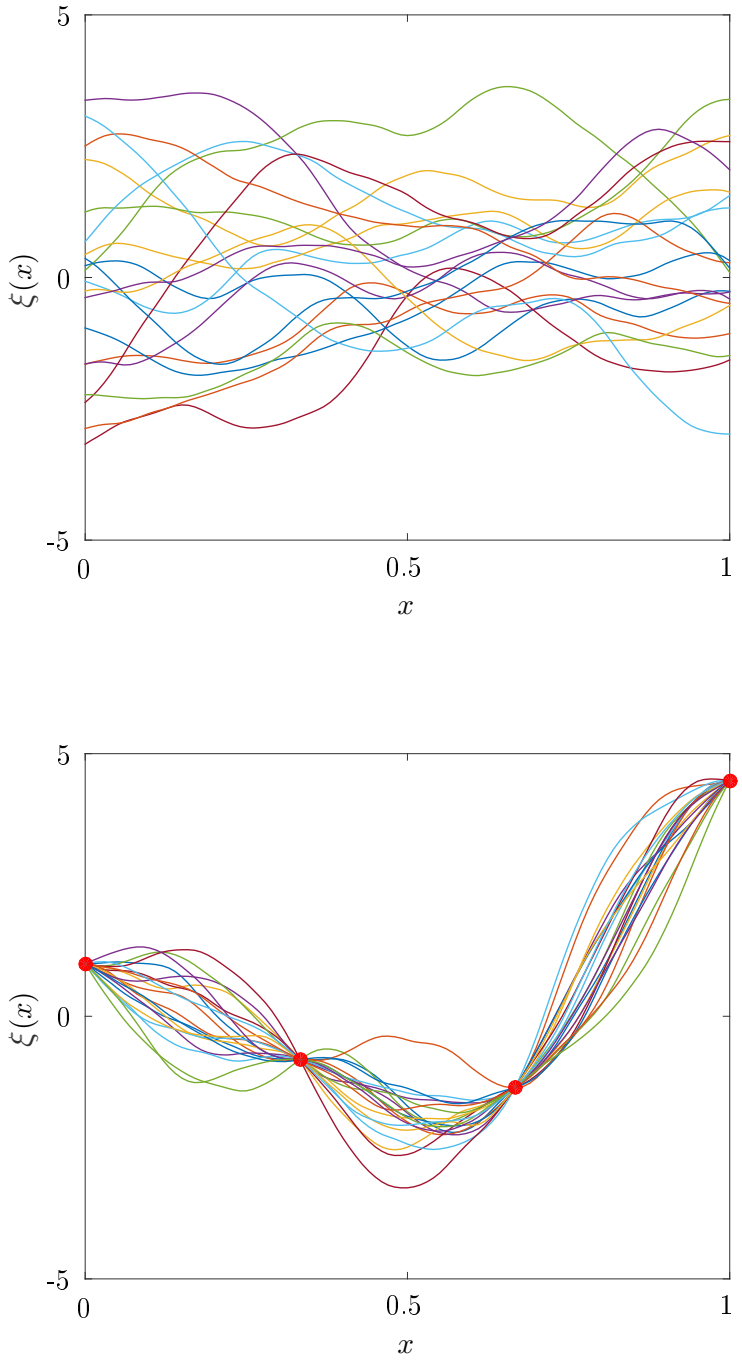


Figure 1.2: Realizations of ξ under a Gaussian process prior distribution (top). Conditional realizations of ξ when four observations have been made (bottom).

for f after n evaluations, can be measured using the loss function

$$\varepsilon_n(\underline{X}, f) = m_n - m\,, \tag{1.3}$$

where $m_n = f(X_1) \wedge \cdots \wedge f(X_n)$ is the best solution that has been observed after n evaluations. Using the Bayesian formalism, the improvement brought by the observation of a new point $x \in \mathbb{X}$ at time n can be measured by the reduction of the loss:

$$I_n(x) = \varepsilon_n(\underline{X}, f) - \varepsilon_{n+1}(\underline{X}, f) = m_n - m_n \wedge \xi(x) = (m_n - \xi(x))_+. \tag{1.4}$$

Note that since $\xi(x)$ is a random variable, the improvement (1.4) is a random quantity. Then, a one-step lookahead optimal choice for the next evaluation point X_{n+1} is to take the point that maximizes the conditional expectation of the improvement I_n:

$$X_{n+1} = \operatorname{argmax}_{x \in \mathbb{X}} \mathbb{E}_n \left(I_n(x) \right)\,, \tag{1.5}$$

where $\mathbb{E}_n$ stands for the conditional expectation with respect to $\mathbf{Y}_n = f(\mathbf{X}_n)$. In the following, we shall denote $\rho_n(x) = \mathbb{E}_n \left(I_n(x) \right)$, $x \in \mathbb{X}$. See Figure 1.3 for an illustration of the operation of this optimization procedure.

The sampling criterion (1.5) is called the expected improvement. In the Bayesian optimization literature, it has been extended to constrained single-objective problems by Schonlau et al. (1998) and to multi-objective problems by Emmerich et al. (2006), among others[2]. The state-of-the-art approach to handle constraints in Bayesian optimization consists in multiplying the expected improvement by the posterior probability of jointly satisfying the constraints, as will be discussed in more details in Section 2.2.3 of this manuscript. This approach however, is not suitable for highly constrained problems, where finding a feasible solution is a challenge in itself.

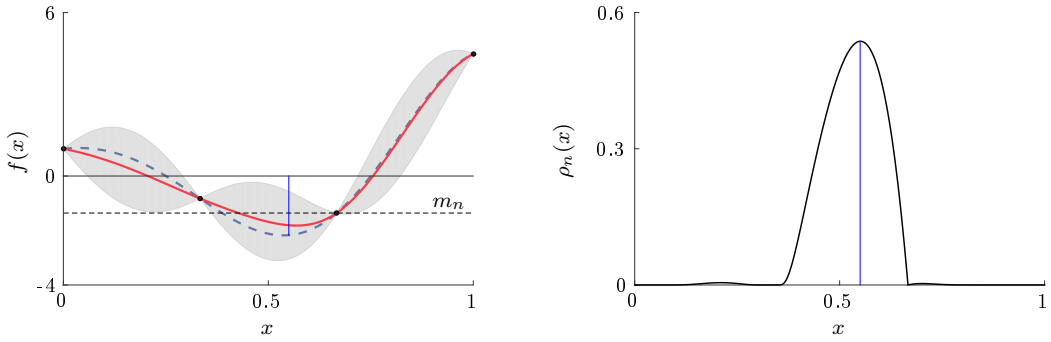
Moreover, note that choosing X_{n+1} using (1.5) requires to solve an auxiliary optimization problem. The EI is cheap to evaluate but it is known to be highly multi-modal (see Figure 1.3), which makes solving this problem difficult in some cases. In the global optimization context, a review of approaches that have been proposed to solve this problem can be found in the PhD thesis of Benassi (2013).

In this thesis we address both difficulties. To handle highly-constrained problems, we propose an extension of the expected improvement criterion. For solving the optimization problem (1.5) we propose dedicated sequential Monte-Carlo techniques, following in this respect Benassi et al. (2012).

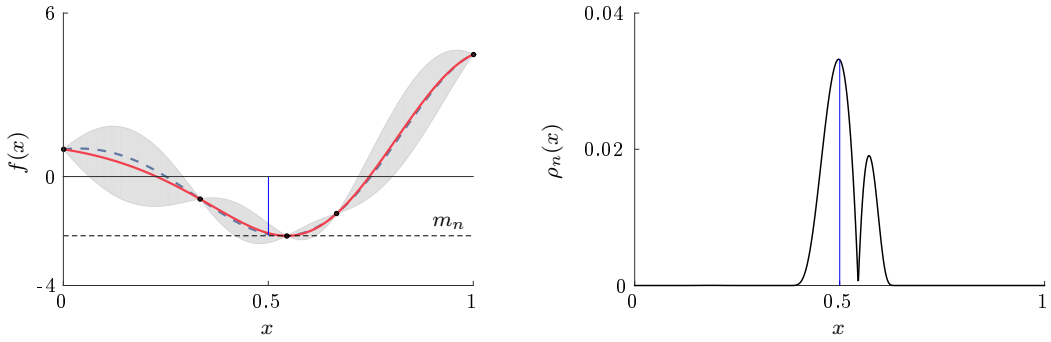
1.2.2 Previous work on similar topics

This thesis work is a continuation of previous work initiated by Julien Bect and Emmanuel Vazquez, supervisors of this thesis, on the coupling between Gaussian process models and se-

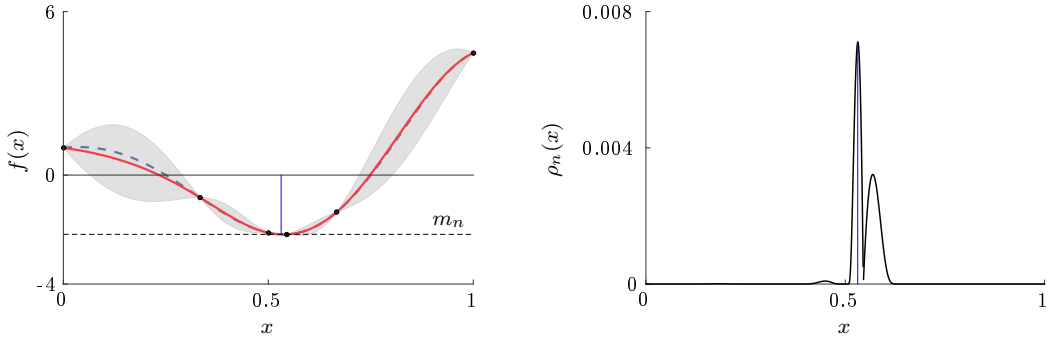
[2] See Remark 1 in Section 2.2.2



(a) Iteration 1 (b) Iteration 1



(c) Iteration 2 (d) Iteration 2



(e) Iteration 3 (f) Iteration 3

Figure 1.3: Bayesian optimization using the EI sampling criterion. On the left column, the function to be minimized is represented as a dashed blue line, the posterior mean of ξ is shown in red and the shaded region corresponds to a 95% confidence interval of the posterior distribution. The observations are shown as black disks and the current best observed value is shown as a black dashed line. On the right column, the values of the EI function are shown as a black curve. On both columns, the location of the maximizer of the EI (i.e. the next iterate) is shown with a blue vertical line.

quential Monte-Carlo (SMC) techniques[3].

In the PhD thesis of Li (2012), a Bayesian approach to the estimation of small probabilities of failure is developed. The proposed approach is an adaptation of the *Subset Simulation* algorithm of Au and Beck (2001), an SMC algorithm for computing small probabilities of failure, to the case where the functions of the problem are expensive to evaluate, and are modeled using Gaussian processes.

In the PhD thesis of Benassi (2013), a fully Bayesian approach to global optimization is proposed and sequential Monte-Carlo techniques inspired from the Subset Simulation algorithm are used for optimizing the EI criterion[4]. In this thesis, we go a step farther and propose an extension of the approach to the case of constrained multi-objective optimization.

1.2.3 Illustration

As the name suggests, sequential Monte-Carlo techniques are sequential sampling techniques (see, e.g., Del Moral et al. (2006)). Given a sequence of distributions $(\pi_n)_{n\geq 1}$ defined on $\mathbb{X}$, they can be used to iteratively draw weighted samples $(\mathcal{X}_n)_{n\geq 1}$, where $\mathcal{X}_n = (x_{n,i}, w_{n,i})_{1\leq i\leq m} \in \mathbb{X}^m \times [0,1]^m$ is approximately distributed from π_n, i.e. the empirical distribution $\tilde{\pi}_n = \sum_{1\leq i\leq m} w_{n,i}\delta_{x_{n,i}}$ is an approximation of π_n.

In the Bayesian global optimization setting where the objective is to minimize a function $f : \mathbb{X} \to \mathbb{R}$ modeled by a Gaussian process ξ, Benassi et al. (2012) define the density π_n, $n \geq 1$ as:

$$\pi_n(x) \propto \mathbb{P}_n(\xi(x) \leq m_n), \tag{1.6}$$

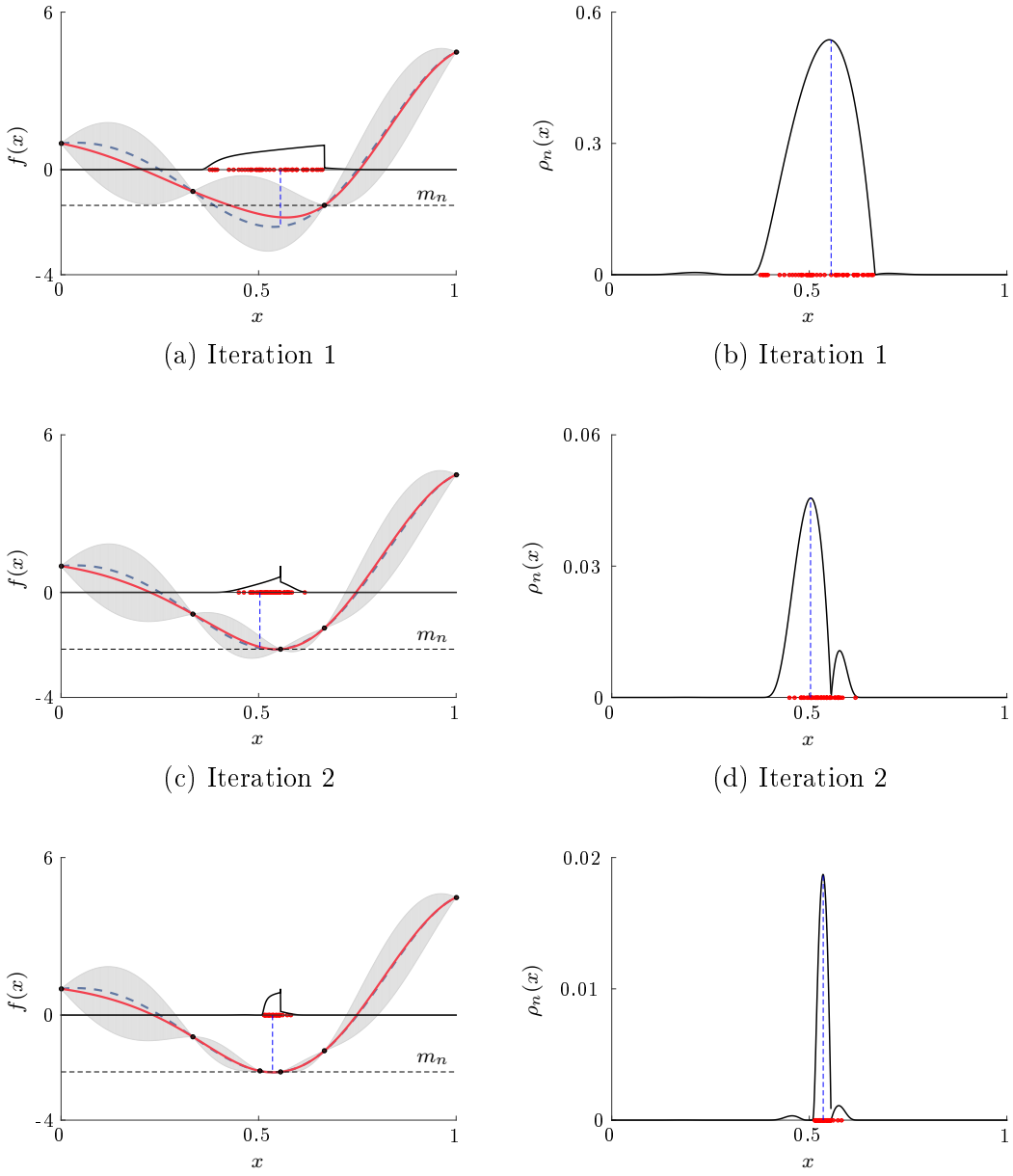
where $\mathbb{P}_n$ denotes the conditional probability knowing the past observations and m_n is the current best solution as in Section 1.2.1. In other words, π_n is chosen proportional to the posterior probability of improving upon the current best solution. Then, using SMC, a weighted sample distributed from π_n can be obtained and the resulting particles $(x_{n,i})_{1\leq i\leq m}$ can be used as candidates for the optimization of the EI criterion:

$$X_{n+1} = \text{argmax}_{1\leq i\leq m}\, \rho_n(x_{n,i})\,. \tag{1.7}$$

The operation of this procedure is illustrated in Figure 1.4. Note in particular how the density of particles follows the concentration of the EI from one iteration to the other.

[3] This coupling has also been studied by Dubourg et al. (2011) in the context of reliability based design optimization.

[4] In the thesis work of Benassi (2013), the EI criterion that is considered is not exactly the one that is introduced in Section 1.2.1 but the ideas that are used for optimizing the criterion can be generalized to other definitions of the EI.



(a) Iteration 1 (b) Iteration 1

(c) Iteration 2 (d) Iteration 2

(e) Iteration 3 (f) Iteration 3

Figure 1.4: Illustration of the SMC procedure for optimizing the EI criterion. The particles are shown as red dots. They are distributed from a density proportional to the probability of improvement, which is shown as a black curve in the figures of the left column. The maximizer of the EI among the particles is shown with a dashed blue line. See Figure 1.3 for more information.

1.3 About this thesis work

1.3.1 Main contributions and outline of the manuscript

The main contribution of this thesis is the proposal of an algorithm for solving constrained multi-objective optimization problems in the case where the functions of the problem are expensive to evaluate. In particular, our focus is on heavily constrained problems, i.e. problems for which finding a feasible solution is difficult in itself, and on many-objective problems.

The proposed algorithm, which we call BMOO, implements a Bayesian approach and is detailed in Chapter 2. This chapter is a reproduction of Feliot et al. (2017) with a few modifications. It is structured as follows. In Section 2.2, we recall the framework of Bayesian optimization based on the expected improvement criterion and discuss some of its extensions to constrained optimization and to multi-objective optimization. Then, we introduce a new EI formulation in Section 2.3. This new formulation is a generalization of the expected hypervolume improvement (EHVI) criterion of Emmerich et al. (2006) and is adapted to both the search of feasible solutions and to the constrained optimization of multiple objectives. For the computation and optimization of the criterion, we propose dedicated sequential Monte-Carlo algorithms. These are detailed respectively in Sections 2.4.1 and 2.4.2. They have applications outside of the framework of the BMOO algorithm and can be viewed as contributions of independent interest. Then, we present experimental results in Section 2.5. The BMOO algorithm is shown to compare favourably with state-of-the-art algorithms for solving constrained single- and multi-objective optimization problems under a limited budget of function evaluations. Conclusions and perspectives for future work are discussed in Section 2.6.

In Chapter 3, we propose improvements and extensions of the algorithm. In Sections 3.2 and 3.3, the computation and optimization of the criterion are revisited and novel sampling densities to be used in the sequential Monte-Carlo samplers are proposed. These new densities make it possible to improve the performances of the BMOO algorithm. Then, in Section 3.4, the algorithm is tested on many-objective problems with up to eight objective functions. Finally, in Section 3.5, we propose extensions of the algorithm. BMOO is extended to handle problems defined on non-hypercubic design spaces (i.e. design spaces defined by bound constraints, a cheap-to-evaluate indicator function and/or cheap-to-evaluate constraints) and to problems having hidden constraints (due to numerical simulation failures for example). Also, to take advantage of parallel computation facilities when available, a batch version of the algorithm is proposed. Finally, the new EI criterion is extended to include user preferences into the search for Pareto optimal solutions.

In Chapter 4, we present applications of the algorithm to real-life design optimization problems. BMOO is applied to the design of a commercial aircraft environment control system (Feliot et al., 2016), to the design of an electric vehicle power-train, to the tuning of a line of sight controller and to the design of a turbo-machine fan blade.

To conclude, in Chapter 5, we make a summary of the manuscript and discuss perspectives for future work.

1.3.2 Publications and communications

2017

Journal paper

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

2016

Conference paper

P. Feliot, Y. Le Guennec, J. Bect, and E. Vazquez. *Design of a commercial aircraft environment control system using Bayesian optimization techniques.* In Engineering Optimization (ENGOPT 2016), Iguassu, Brazil, 2016.

Communications

P. Feliot, J. Bect, and E. Vazquez. *Bayesian multi-objective optimization: Application to the design of an electric vehicle powertrain.* Forum incertitudes CEA/DAM. Méthodes de quantification des incertitudes, Bruyères-le-Châtel, October 2016.

P. Feliot, J. Bect, and E. Vazquez. *Bayesian multi-objective optimization with constraints: Application to the design of a commercial aircraft environment control system.* GdR MASCOT-NUM working meeting: Dealing with stochastics in optimization problems, Institut Henri Poincaré (IHP), Paris, May 2016.

P. Feliot., J. Bect and E. Vazquez. *BMOO: a Bayesian multi-objective optimization algorithm.* MascotNum Annual Conference, Toulouse, France, March 2016.

2015

Conference paper

P. Feliot, J. Bect, and E. Vazquez. *A Bayesian approach to constrained multi-objective optimization.* In International Conference on Learning and Intelligent Optimization, pages 256–261. Springer, 2015a.

2015

Communications

P. Feliot, J. Bect, and E. Vazquez. *A Bayesian approach to constrained multi-objective optimization of expensive-to-evaluate functions.* In World Congress on Global Optimization (WCGO 2015), Gainesville (Florida), United States, 2015b.

P. Feliot, J. Bect, and E. Vazquez. *A Bayesian approach to constrained multi-objective optimization.* Journées annuelles du GdR MASCOT NUM, Saint-Etienne, France, April 2015. (Poster)

P. Feliot, J. Bect, and E. Vazquez. *A Bayesian approach to constrained multi-objective optimization.* In Sequential Monte Carlo workshop (SMC 2015), Paris, France, August 2015. (Poster)

2014

Communication

P. Feliot, J. Bect, and E. Vazquez. *A Bayesian subset simulation approach to constrained global optimization of expensive-to-evaluate black-box functions.* In PGMO-COPI 14, Palaiseau, France, October 2014.

Chapter 2

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

2.1 Introduction

In this thesis, we address 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

$$\Gamma = \{x \in \mathbb{X} : c(x) \leq 0 \text{ and } \nexists\, x' \in \mathbb{X} \text{ s.t. } c(x') \leq 0 \text{ and } f(x') \prec f(x)\} \tag{2.1}$$

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 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 (2012) for some of the earliest references in the field. Bayesian optimization research was first 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. Later, the EI criterion has been extended to handle constraints (Schonlau et al., 1998; Sasena et al., 2002; Gramacy and Lee, 2011; Gelbart et al., 2014; Gramacy et al., 2016) and to address bound-constrained multi-objective problems (Emmerich et al., 2006; Jeong et al., 2006; Wagner et al., 2010; Svenson and Santner, 2010).

With this chapter, our contribution is twofold. The first part of the contribution is the proposition of a new sampling criterion that handles multiple objectives and non-linear constraints simultaneously. This criterion corresponds to a one-step look-ahead Bayesian strategy, using the dominated hyper-volume as a utility function (following in this respect Emmerich et al., 2006). More specifically, the dominated hyper-volume is defined using an extended domination rule, which handles objectives and constraints in a unified way (in the spirit of Fonseca and Fleming, 1998; Ray et al., 2001; Oyama et al., 2007). This new criterion is naturally adapted to the search of a feasible point when none is available, and several criteria from the literature—the EI

criterion and some of its constrained/multi-objective extensions—are recovered as special cases when at least one feasible point is known. The second part of the contribution lies 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 a well known Sequential Monte Carlo (SMC) technique (see Del Moral et al., 2006; Liu, 2001, and references therein) from the field of structural reliability and rare event estimation. For the optimization of the criterion, we resort to an SMC method as well, 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).

This chapter is based on Feliot et al. (2017). Its structure is as follows. In Section 2.2, we recall the framework of Bayesian optimization based on the expected improvement sampling criterion, starting with the unconstrained single-objective setting. Section 2.3 presents our new sampling criterion for constrained multi-objective optimization. The calculation and the optimization of the criterion are discussed in Section 2.4. Section 2.5 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- and multi-objective constrained optimization problems from the literature. Finally, future work is discussed in Section 2.6.

2.2 Background literature

2.2.1 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 the classical loss function

$$\varepsilon_n(\underline{X}, f) = m_n - m\,, \tag{2.2}$$

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 about 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:

$$X_{n+1} \;=\; \operatorname{argmin}_{x \in \mathbb{X}} \mathbb{E}_n\big(\varepsilon_{n+1}(\underline{X}, \xi) \mid X_{n+1} = x\big)\,, \tag{2.3}$$

where $\mathbb{E}_n$ stands for the conditional expectation with respect to X_1, $\xi(X_1)$, …, 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[1]; the reader is referred to Ginsbourger and Le Riche (2010) and Benassi (2013) for discussions about two-step look-ahead strategies.

When (2.2) is used as a loss function, the right-hand side of (2.3) can be rewritten as

$$\begin{aligned} \operatorname{argmin} \mathbb{E}_n\big(\varepsilon_{n+1}(\underline{X}, \xi) \mid X_{n+1} = x\big) &= \operatorname{argmin} \mathbb{E}_n\left(m_{n+1} \,\middle|\, X_{n+1} = x\right) \\ &= \operatorname{argmax} \mathbb{E}_n\big((m_n - \xi(x))_+\big)\,, \end{aligned} \tag{2.4}$$

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

$$\rho_n(x) : x \mapsto \mathbb{E}_n\big((m_n - \xi(x))_+\big) \tag{2.5}$$

is called the Expected Improvement (EI) criterion (Schonlau et al., 1998; Jones et al., 1998). When ξ is a Gaussian process with known mean and covariance functions, $\rho_n(x)$ has a closed-form expression:

$$\rho_n(x) \;=\; \gamma\left(m_n - \widehat{\xi}_n(x),\, \sigma_n^2(x)\right), \tag{2.6}$$

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, $\widehat{\xi}_n(x) = \mathbb{E}_n(\xi(x))$ for the kriging predictor at x (the posterior mean of $\xi(x)$ after n evaluations) and $\sigma_n^2(x)$ for the kriging variance at x (the posterior variance of $\xi(x)$ after

[1] Mockus (2012, Section 2.5) heuristically introduces a modification of (2.3) to compensate for the fact that subsequent evaluation results are not taken into account in the myopic strategy and thus enforce a more global exploration of the search domain. In this work, we consider a purely myopic strategy as in Jones et al. (1998).

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 (2.3) 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 (2.6), 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 then performing a discrete search. Recently, sequential Monte Carlo techniques (see Del Moral et al., 2006; Liu, 2001, 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).

2.2.2 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

$$\Gamma = \{x \in \mathbb{X} : \nexists\, x' \in \mathbb{X} \text{ such that } \; f(x') \prec f(x)\}\,, \tag{2.7}$$

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

$$y = (y_1, \ldots, y_p) \prec z = (z_1, \ldots, z_p) \Longleftrightarrow \begin{cases} \forall i \leq p, & y_i \leq z_i\,, \\ \exists j \leq p, & y_j < z_j\,. \end{cases} \tag{2.8}$$

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

$$H_n = \{y \in \mathbb{B}; \exists i \leq n,\, f(X_i) \prec y\}\,, \tag{2.9}$$

where $\mathbb{B} \subset \mathbb{R}^p$ is a set of the form $\mathbb{B} = \{y \in \mathbb{R}^p;\, y \leq y^{\text{upp}}\}$ 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 (2.5) to the multi-objective case, is to use the volume of the non-dominated region as loss function:

$$\varepsilon_n(\underline{X}, f) = |H \setminus H_n|\,,$$

where $H = \{y \in \mathbb{B}; \exists x \in \mathbb{X}, f(x) \prec y\}$ and $|\cdot|$ denotes the usual (Lebesgue) volume in $\mathbb{R}^p$.

The improvement yielded by a new evaluation result $f(X_{n+1}) = (f_1(X_{n+1}), \ldots, f_p(X_{n+1}))$ is then the increase of the volume of the dominated region (see Figure 2.1):

$$I_n\left(X_{n+1}\right) = |H \setminus H_n| - |H \setminus H_{n+1}| = |H_{n+1} \setminus H_n| = |H_{n+1}| - |H_n|\,, \tag{2.10}$$

since $H_n \subset H_{n+1} \subset H$. 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

$$\begin{aligned} \rho_n(x) &= \mathbb{E}_n\left(I_n(x)\right) \\ &= \mathbb{E}_n\left(\int_{\mathbb{B}\setminus H_n} \mathbb{1}_{\xi(x)\prec y}\,\mathrm{d}y\right) \\ &= \int_{\mathbb{B}\setminus H_n} \mathbb{E}_n\left(\mathbb{1}_{\xi(x)\prec y}\right)\,\mathrm{d}y \\ &= \int_{\mathbb{B}\setminus H_n} \mathbb{P}_n\left(\xi(x)\prec y\right)\,\mathrm{d}y\,, \end{aligned} \tag{2.11}$$

where $\mathbb{P}_n$ stands for the probability $\mathbb{P}_0$ conditioned on X_1, $\xi(X_1)$, …, X_n, $\xi(X_n)$. The multi-objective sampling criterion (2.11), also called Expected Hyper-Volume Improvement (EHVI), has been proposed by Emmerich and coworkers (Emmerich, 2005; Emmerich et al., 2006; Emmerich and Klinkenberg, 2008).

Remark 1 *A variety of alternative approaches have been proposed to extend the EI criterion to the multi-objective case, which can be roughly classified into aggregation-based techniques (Knowles, 2006; Knowles and Hughes, 2005; Zhang et al., 2010) and domination-based techniques (see e.g. Jeong and Obayashi, 2005; Keane, 2006; Ponweiser et al., 2008; Bautista, 2009; Svenson and Santner, 2010; Wagner et al., 2010). We consider these approaches are heuristic extensions of the EI criterion, in the sense that none of them emerges from a proper Bayesian formulation (i.e., a myopic strategy associated to some well-identified loss function). A detailed description of these approaches is out of the scope of this thesis. The reader is referred to Wagner et al. (2010), Couckuyt et al. (2014) and Horn et al. (2015) for some comparisons and discussions. See also Picheny (2014b) and Hernández-Lobato et al. (2015) for other approaches not directly related to the concept of expected improvement.*

Remark 2 *The multi-objective sampling criterion* (2.11) *reduces to the usual EI criterion* (2.5) *in the single-objective case (assuming that $f(X_i) \leq y^{\mathrm{upp}}$ for at least one $i \leq n$).*

Under the assumption that the components ξ_i of ξ are mutually independent[2], $\mathbb{P}_n\left(\xi(x) \prec y\right)$ can be expressed in closed form: for all $x \in \mathbb{X}$ and $y \in \mathbb{B} \setminus H_n$,

$$\mathbb{P}_n\left(\xi(x) \prec y\right) = \prod_{i=1}^{p} \Phi\left(\frac{y_i - \widehat{\xi}_{i,n}(x)}{\sigma_{i,n}(x)}\right), \tag{2.12}$$

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

[2] 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.

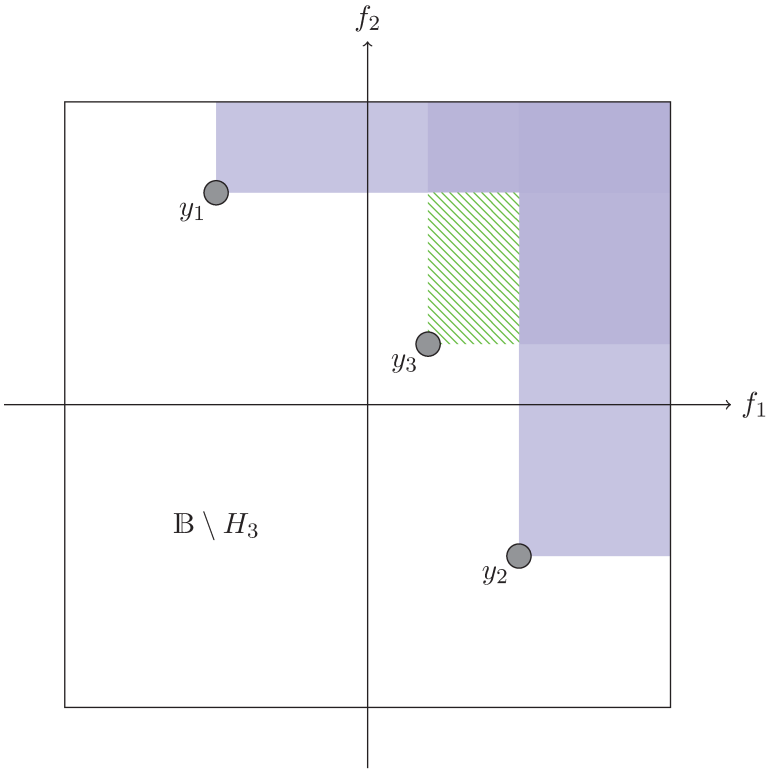


Figure 2.1: Example of an improvement of the dominated region. The regions dominated by y_1 and y_2 are represented in shaded areas, with darker shades indicating overlapping regions. The hatched area corresponds to the improvement of the dominated region resulting from the observation of y_3.

The integration of (2.12) over $\mathbb{B} \setminus H_n$, in the expression (2.11) 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; Couckuyt 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[3], 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 2.4). Our approach will make it possible to deal with more objective functions.

Remark 3 *Exact and approximate implementations of the EHVI criterion are available, together with other Gaussian-process-based criteria for bound-constrained multi-objective optimization, in the Matlab/Octave toolbox STK (Bect et al., 2016b) and in the R packages GPareto (Binois and Picheny, 2015) and mlrMBO (Horn et al., 2015). Note that several approaches discussed in Remark 1 maintain an affordable computational cost when the number of objectives grows, and therefore constitute possible alternatives to the SMC technique proposed in this paper for many-objective box-constrained problems.*

2.2.3 EI-based optimization with constraints

In this section, we discuss extensions of the expected improvement criterion for single- and multi-objective constrained optimization.

Consider first the case of problems with a single objective and several constraints:

$$\begin{cases} \min_{x \in \mathbb{X}} f(x)\,, \\ c(x) \leq 0\,, \end{cases} \tag{2.13}$$

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 observed objective value $m_n = \min\{f(x);\, x \in \{X_1, \ldots, X_n\} \cap C\}$:

$$\begin{aligned} I_n(X_{n+1}) &= m_n - m_{n+1} \\ &= \mathbb{1}_{c(X_{n+1}) \leq 0} \cdot \big(m_n - f(X_{n+1})\big)_+ \\ &= \begin{cases} m_n - f(X_{n+1}) & \text{if } X_{n+1} \in C \text{ and } f(X_{n+1}) < m_n, \\ 0 & \text{otherwise}\,. \end{cases} \end{aligned} \tag{2.14}$$

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

[3] See, e.g., Beume (2009), Hupkens et al. (2014), Couckuyt et al. (2014) and references therein for decomposition algorithms and complexity results.

follows from taking the expectation:

$$\rho_n(x) = \mathbb{E}_n\left(\mathbb{1}_{\xi_c(x)\leq 0} \cdot \big(m_n - \xi_o(x)\big)_+\right) . \tag{2.15}$$

If f is modeled by a random process ξ_o and c is modeled by a vector-valued random process $\xi_c = (\xi_{c,1}, \ldots, \xi_{c,q})$ independent of ξ_o, then the sampling criterion (2.15) simplifies to Schonlau et al.'s criterion:

$$\rho_n(x) = \mathbb{P}_n(\xi_c(x) \leq 0)\ \mathbb{E}_n\big((m_n - \xi_o(x))_+\big) . \tag{2.16}$$

In other words, the expected improvement is equal in this case to the product of the unconstrained expected improvement, with respect to m_n, with the probability of feasibility. The sampling criterion (2.16) 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 by Parr et al. (2012) and Gelbart (2015).

In the general case of constrained multi-objective problems, the aim is to build an approximation of Γ defined by (2.1). If it is assumed that an observation has been made in the feasible set C, a reasoning similar to that used in the single-objective case can be made to formulate an extension of the EI (2.11):

$$\rho_n(x) = \mathbb{E}_n\left(|H_{n+1}| - |H_n|\right), \tag{2.17}$$

where

$$H_n = \{y \in \mathbb{B}; \exists i \leq n,\ X_i \in C \text{ and } f(X_i) \prec y\} \tag{2.18}$$

is the subset of $\mathbb{B}$, defined as in Section 2.2.2, whose points are dominated by feasible evaluations. When ξ_o and ξ_c are assumed independent, (2.17) boils down to the product of a modified EHVI criterion, where only feasible points are considered[4], and the probability of feasibility, as suggested by Emmerich et al. (2006) and Shimoyama et al. (2013b):

$$\rho_n(x) = \mathbb{P}_n\left(\xi_c(x) \leq 0\right) \int_{\mathbb{B}\setminus H_n} \mathbb{P}_n\left(\xi_o(x) \prec y\right) \mathrm{d}y. \tag{2.19}$$

Observe that the sampling criterion (2.17) is the one-step look-ahead criterion associated to the loss function $\varepsilon_n(\underline{X}, f) = -|H_n|$, where H_n is defined by (2.18). This loss function remains constant as long as no feasible point has been found and, therefore, is not an appropriate measure of loss for heavily constrained problems where finding feasible points is sometimes the main difficulty[5]. From a practical point of view, not all unfeasible points should be considered equivalent: a point that does not satisfy a constraint by a small amount has probably more value than one that does not satisfy the constraint by a large amount, and should therefore

[4] Note that this modified EHVI criterion remains well defined even when $H_n = \emptyset$, owing to the introduction of an upper bound y^{upp} in the definition of $\mathbb{B}$. Its single-objective counterpart introduced earlier (see Equation (2.15)), however, was only well defined under the assumption that at least one feasible point is known. Introducing an upper bound y^{upp} is of course also possible in the single-objective case.

[5] The same remark holds for the variant (see, e.g., Gelbart et al., 2014) which consists in using the probability of feasibility as a sampling criterion when no feasible point is available. This is indeed equivalent to using the loss function $\varepsilon_n(\underline{X}, f) = -\mathbb{1}_{\exists i \leq n, X_i \in C}$ in the search for feasible points.

make the loss smaller. Section 2.3 will present a generalization of the expected improvement for constrained problems, relying on a new loss function that encodes this preference among unfeasible solutions.

Remark 4 *Other Gaussian-process-based approaches that can be used to handle constraints include the method by Gramacy et al. (2016), based on the* augmented Lagrangian *approach of Conn et al. (1991), and several recent methods (Picheny, 2014a; Gelbart, 2015; Hernández-Lobato et al., 2015, 2016a) based on stepwise uncertainty reduction strategies (see, e.g., Villemonteix et al., 2009; Bect et al., 2012; Chevalier et al., 2014a, for more information on this topic).*

Remark 5 *The term* $\mathbb{E}_n\big((m_n - \xi_{\mathrm{o}}(x))_+\big)$ *in* (2.16) *can be computed analytically as in Section 2.2.1, and the computation of the integral in* (2.19) *has been discussed in Section 2.2.2. If it is further assumed that the components of* ξ_{c} *are Gaussian and independent, then the probability of feasibility can be written as*

$$\mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0) = \prod_{j=1}^{q} \Phi\left(-\frac{\widehat{\xi}_{\mathrm{c},\,j,\,n}(x)}{\sigma_{\mathrm{c},\,j,\,n}(x)}\right) \tag{2.20}$$

where $\widehat{\xi}_{\mathrm{c},\,j,\,n}(x)$ *and* $\sigma^2_{\mathrm{c},\,j,\,n}(x)$ *stand respectively for the kriging predictor and the kriging variance of* $\xi_{\mathrm{c},\,j}$ *at* x*.*

2.3 An EI criterion for constrained multi-objective optimization

2.3.1 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 2.2.2 and, denoting by $\overline{\mathbb{R}}$ the extended real line,

$$\begin{array}{rccl} \psi: & \mathbb{Y}_{\mathrm{o}} \times \mathbb{Y}_{\mathrm{c}} & \to & \overline{\mathbb{R}}^p \times \mathbb{R}^q \\ & (y_{\mathrm{o}},\, y_{\mathrm{c}}) & \mapsto & \begin{cases} (y_{\mathrm{o}},\, 0) & \text{if } y_{\mathrm{c}} \leq 0, \\ \big(+\infty,\, \max(y_{\mathrm{c}}, 0)\big) & \text{otherwise.} \end{cases} \end{array} \tag{2.21}$$

The extended domination rule (2.21) 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 $\mathbb{Y} = \mathbb{Y}_{\mathrm{o}}$.

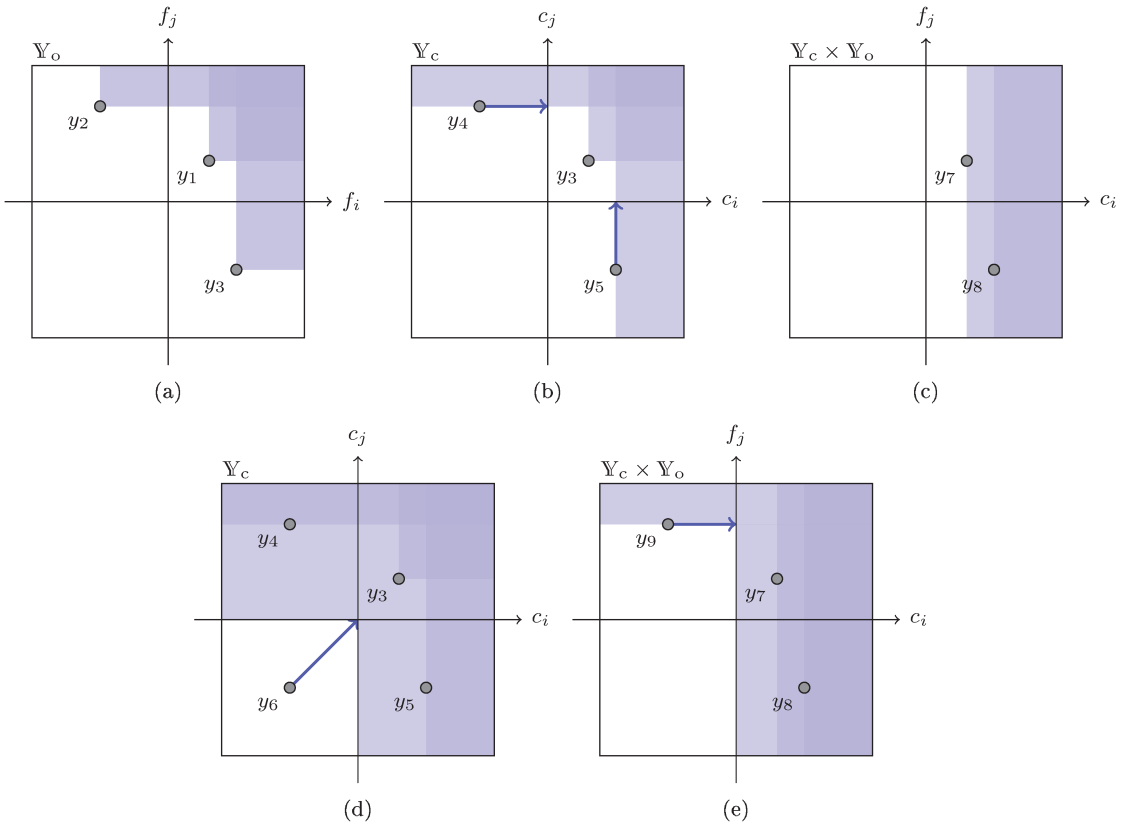


Figure 2.2: Illustration of the extended domination rule in different cases. The region dominated by each point is represented by a shaded area. Darker regions indicate overlapping regions. (a) Feasible solutions are compared with respect to their objective values using the usual domination rule in the 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).

(ii) Feasible solutions (corresponding to $y_c \leq 0$) 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 $y_c \not\leq 0$) 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 2.2.

2.3.2 A new EI criterion

The extended domination rule presented above makes it possible to define a notion of expected hyper-volume improvement as in Section 2.2.2 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 $\mathbb{B} = \mathbb{B}_{\rm o} \times \mathbb{B}_{\rm c}$, where $\mathbb{B}_{\rm o} \subset \mathbb{Y}_{\rm o}$ and $\mathbb{B}_{\rm c} \subset \mathbb{Y}_{\rm c}$ are two bounded hyper-rectangles that are introduced to ensure, as in Section 2.2.2, that $|H_n| < +\infty$ (see Appendix 2.7.1 and subfigures (c) and (e) of Figure 2.2). Then, define the improvement yielded by a new evaluation $(f(X_{n+1}),\, c(X_{n+1}))$ by

$$I_n\left(X_{n+1}\right) = |H_{n+1} \setminus H_n| \,= |H_{n+1}| - |H_n| \tag{2.22}$$

as in Section 2.2.2. 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 $\mathbb{B}_{\rm c}$.

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

$$\rho_n(x) = \mathbb{E}_n\big((I_n(x)\big) = \mathbb{E}_n\left(\int_{G_n} \mathbb{1}_{\xi(x) \lhd y}\, {\rm d}y\right) = \int_{G_n} \mathbb{P}_n(\xi(x) \lhd y)\, {\rm d}y\,, \tag{2.23}$$

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

As in Section 2.2.2, 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 $y = (y_{\rm o},\, y_{\rm c}) \in G_n$,

$$\mathbb{P}_n(\xi(x) \lhd y) = \begin{cases} \displaystyle\left(\prod_{i=1}^{p} \Phi\left(\frac{y_{{\rm o},\,i} - \widehat{\xi}_{{\rm o},\,i,\,n}(x)}{\sigma_{{\rm o},\,i,\,n}(x)}\right)\right)\left(\prod_{j=1}^{q} \Phi\left(-\frac{\widehat{\xi}_{{\rm c},\,j,\,n}(x)}{\sigma_{{\rm c},\,j,\,n}(x)}\right)\right) & \text{if } y_{\rm c} \leq 0\,, \\ \displaystyle\prod_{j=1}^{q} \Phi\left(\frac{\max(y_{{\rm c},\,j},\, 0) - \widehat{\xi}_{{\rm c},\,j,\,n}(x)}{\sigma_{{\rm c},\,j,\,n}(x)}\right) & \text{otherwise}\,. \end{cases} \tag{2.24}$$

The EI-based constrained multi-objective optimization procedure may be written as (2.3). In practice, the computation of each new evaluation point requires to solve two numerical problems: a) the computation of the integral in (2.23); b) the optimization of ρ_n in the procedure (2.3). These problems will be addressed in Section 2.4.

Remark 6 *When there are no constraints ($q = 0$, $\mathbb{Y}_{\rm c} = \emptyset$), the extended domination rule $\lhd$ corresponds to the usual Pareto domination rule $\prec$. In this case, the sampling criterion* (2.23) *simplifies to*

$$\rho_n(x) = \int_{\mathbb{B}_{\rm o} \setminus H_{n,{\rm o}}} \mathbb{P}_n\left(\xi_{\rm o}(x) \prec y_{\rm o}\right)\, {\rm d}y_{\rm o}, \tag{2.25}$$

with

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

Denote by $y_{\rm o}^{low}$, $y_{\rm o}^{upp} \in \mathbb{Y}_{\rm o}$ the lower and upper corners of the hyper-rectangle $\mathbb{B}_{\rm o}$. Then, the sam-

pling criterion (2.25) *is equivalent to the multi-objective EI criterion presented in Section 2.2.2 in the limit* $y_{\mathrm{o}}^{low} \to -\infty$. *If, moreover, the problem has only one objective function, then the criterion* (2.23) *boils down to the original expected improvement criterion as soon as the best evaluation dominates* y_{o}^{upp} *(see Remark 2).*

2.3.3 Decomposition of the expected improvement: feasible and unfeasible components

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

$$\rho_n(x) = \rho_n^{\mathrm{feas}}(x) + \rho_n^{\mathrm{unf}}(x), \tag{2.26}$$

by splitting the integration domain in the right-hand side of (2.23) 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,

$$\begin{aligned} \rho_n^{\mathrm{unf}}(x) &= \int_{G_n \cap \{y_{\mathrm{c}} \not\leq 0\}} \mathbb{P}_n\left((\xi_{\mathrm{o}}(x), \xi_{\mathrm{c}}(x)) \lhd (y_{\mathrm{o}}, y_{\mathrm{c}})\right) \mathrm{d}(y_{\mathrm{o}}, y_{\mathrm{c}}) \\ &= |\mathbb{B}_{\mathrm{o}}| \cdot \int_{\mathbb{B}_{\mathrm{c}} \setminus H_{n,\mathrm{c}}} \mathbb{P}_n\left(\xi_{\mathrm{c}}^+(x) \prec y_{\mathrm{c}}^+\right) \mathbb{1}_{y_{\mathrm{c}} \not\leq 0} \, \mathrm{d}y_{\mathrm{c}} \end{aligned} \tag{2.27}$$

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

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

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

$$\begin{aligned} \rho_n^{\mathrm{feas}}(x) &= \int_{G_n \cap \{y^c \leq 0\}} \mathbb{P}_n\left((\xi_{\mathrm{o}}(x), \xi_{\mathrm{c}}(x)) \lhd (y_{\mathrm{o}}, y_{\mathrm{c}})\right) \mathrm{d}(y_{\mathrm{o}}, y_{\mathrm{c}}) \\ &= \left|\mathbb{B}_{\mathrm{c}}^-\right| \cdot \mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0) \cdot \int_{\mathbb{B}_{\mathrm{o}} \setminus H_{n,\mathrm{o}}} \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}}\,, \end{aligned} \tag{2.28}$$

where

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

Remark 7 *The set* $\mathbb{B}_{\mathrm{c}} \setminus H_{n,\mathrm{c}}$ *is 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* (2.28),

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

In other words, up to a multiplicative constant, the expected improvement is 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 constrained single-objective problems, the criterion of Schonlau et al. (see Section 2.2.3) is recovered as the limit case $y_{\mathrm{o}}^{low} \to -\infty$, *as soon as the best evaluation dominates* y_{o}^{upp}.

Remark 8 *In the numerical experiments of Section 2.5,* $\mathbb{B}_{\mathrm{o}}$ *and* $\mathbb{B}_{\mathrm{c}}$ *are defined using estimates of the range of the objective and constraint functions (see Appendix 2.7.2). Another natural choice for* $\mathbb{B}_{\mathrm{o}}$ *would have been to use (an estimate of) the range of the objective functions restricted to the feasible subset* $C \subset \mathbb{X}$*. Further investigation of this idea is left for future work.*

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

2.4.1 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 2.2.2, 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 (2.23):

$$\rho_n(x) \approx \frac{1}{m} \sum_{k=1}^{m} \mathbb{P}_n(\xi(x) \lhd y_{n,k}), \tag{2.29}$$

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, 2004), 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 $\mathbb{B}_{\mathrm{c}} = [-v/2, +v/2]^q$ for some $v > 0$, then the volume of $\mathbb{B}_{\mathrm{c}}^-$ 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):

$$\mathbb{B} \supset G_1 \supset G_2 \supset \cdots . \tag{2.30}$$

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

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, 2004) which targets the uniform distribution over G_n. The resulting set of particles is $\mathcal{Y}_n$.

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 $\mathcal{Y}_n = (y_{n,k})_{1\leq k\leq m}$ 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[6].

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[7] set $A \subset \mathbb{B}$ is

$$\mathbb{E}_n\Big(N(A;\,\mathcal{Y}_n)\Big) = \mathbb{E}_n\left(\sum_{k=1}^{m} \mathbb{1}_A(y_{n,k})\right) = m \cdot \Pi_n^{\mathbb{Y}}(A)\,, \tag{2.31}$$

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 (2.30) to ensure that the volume of the subsets does not decrease too fast. The corrected version of Algorithm 1 is described in Algorithms 2, 3 and 4. The method used in Algorithm 4 to construct the intermediate subsets is illustrated on Figures 2.3 and 2.4.

Remark 9 *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*

[6] All the random variables generated in Algorithm 1 are independent of ξ conditionally on X_1, $\xi(X_1)$, …, X_{n+1}, $\xi(X_{n+1})$.

[7] Equation (2.31) does not hold exactly for $A = G_{n+1}$ since, conditionally on X_1, $\xi(X_1)$, …, 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 $\mathcal{Y}_n$.

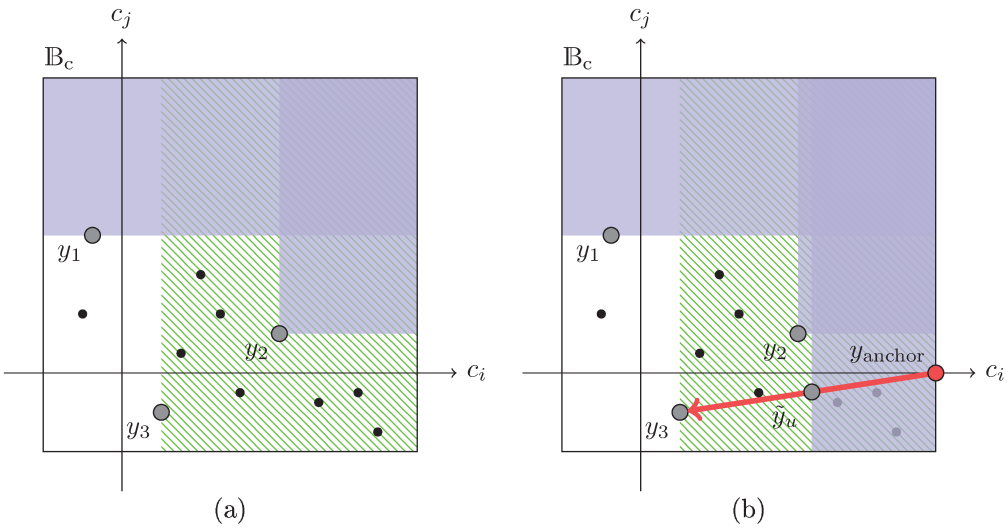


Figure 2.3: Illustration of the steps $1 \to 2$ and $3 \to 5$ of Algorithm 4. The objective is to build a uniform sample $\mathcal{Y}_3$ on G_3 from $\mathcal{Y}_2$. The initial Pareto front $\mathcal{P}_0$ is determined by evaluation results $y_1 = (f(X_1), c(X_1))$ and $y_2 = (f(X_2), c(X_2))$. $\mathcal{P}_T$ 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 of Algorithm 3, 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 $\mathcal{P}$ and $\mathcal{P}^\star$, 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 4, 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 $(y_{\text{anchor}}$–$y_3)$. 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.

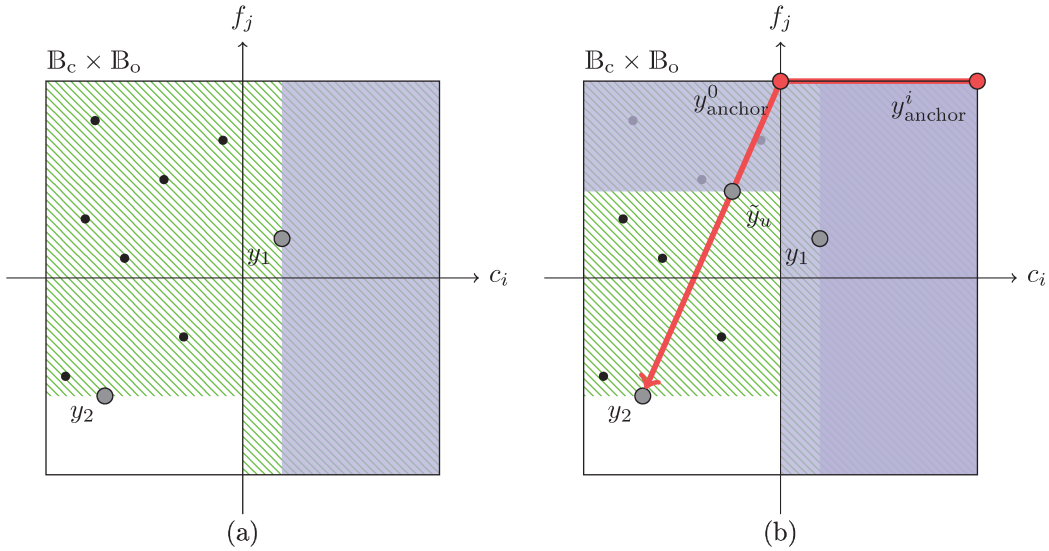


Figure 2.4: Illustration of the steps $1 \to 2$ and $6 \to 10$ of Algorithm 4. The setting is the same as that described in Figure 2.3, except that the new observation (y_2 in this case) is feasible. Hence, $q^\star = q$. As above, the main idea is to construct a continuous path between $\mathcal{P}$ and $\mathcal{P}^\star$, as illustrated in (b).

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.

objectives, multiples constraints and possibly correlated Gaussian process models. When the objectives and constraints are independent, the decomposition introduced in Section 2.3.3 makes it possible to compute two integrals over spaces of lower dimension (over $\mathbb{B}_{\mathrm{c}} \setminus H_{n,\mathrm{c}}$ and $\mathbb{B}_{\mathrm{o}} \setminus H_{n,\mathrm{o}}$, 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 2.5.

2.4.2 Maximization of the sampling criterion

The optimization of the sampling criterion (2.23) is a difficult problem in itself because, even under the unconstrained single-objective setting, the EI criterion is very often highly multimodal. 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

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

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 Algorithm 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

$$\mathcal{X}_n = (x_{n,k}, w_{n,k})_{k=1}^m \in (\mathbb{X} \times \mathbb{R})^m \tag{2.32}$$

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

Notations: 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,
- $G(A) := \mathbb{B} \setminus \{y \in \mathbb{B};\ \exists y' \in A \text{ such that } y' \lhd y\}$ the region of $\mathbb{B}$ not dominated by A.

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

- $G(\mathcal{P}^\star) \subset G(\mathcal{P}_0)$, with $\mathcal{P}_0 = \text{Pareto}(\mathcal{P}_0)$ and $\mathcal{P}^\star = \text{Pareto}(\mathcal{P}^\star)$,
- $\mathcal{Y}_0 = (y_{0,k})_{1 \leq k \leq m} \in \mathbb{Y}^m$ 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 $\mathbb{B}_{\text{o}} = \left\{y \in \mathbb{Y}_{\text{o}};\ y_{\text{o}}^{\text{low}} \leq y \leq y_{\text{o}}^{\text{upp}}\right\}$, $\mathbb{B}_{\text{c}} = \left\{y \in \mathbb{Y}_{\text{c}};\ y_{\text{c}}^{\text{low}} \leq y \leq y_{\text{c}}^{\text{upp}}\right\}$, 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 $\mathcal{Y}_t = (y_{t,k})_{1 \leq k \leq m} \in \mathbb{Y}^m$ uniformly distributed on $G(\mathcal{P}^\star)$.

$t \leftarrow 0$
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**
    $\mathcal{P} \leftarrow \widetilde{\mathcal{P}}_u$ with $\widetilde{\mathcal{P}}_u = \text{chooseNextFront}(\mathcal{Y}_t, \mathcal{P}, \mathcal{P}^\star, \mathbb{B})$
  $\mathcal{P}_{t+1} \leftarrow \mathcal{P}$
  Generate $\mathcal{Y}_{t+1} = (y_{t+1,k})_{1 \leq k \leq m}$ uniformly distributed on $G(\mathcal{P}_{t+1})$ using the "Remove-Resample-Move" steps described in Algorithm 1.
  $t \leftarrow t + 1$

Algorithm 4: chooseNextFront

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

- $G\left(\mathcal{P}^\star\right) \subset G\left(\mathcal{P}\right)$, with $\mathcal{P} = \text{Pareto}\left(\mathcal{P}\right)$ and $\mathcal{P}^\star = \text{Pareto}\left(\mathcal{P}^\star\right)$,
- $\mathcal{Y}_t = \left(y_{t,k}\right)_{1 \leq k \leq m} \in \mathbb{Y}^m$ is uniformly distributed on $G\left(\mathcal{P}_t\right)$ (see Algorithm 3). Note that $\mathcal{Y}_t$ 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 $\mathbb{B}_{\text{o}} = \left\{y \in \mathbb{Y}_{\text{o}};\ y_{\text{o}}^{\text{low}} \leq y \leq y_{\text{o}}^{\text{upp}}\right\}$, $\mathbb{B}_{\text{c}} = \left\{y \in \mathbb{Y}_{\text{c}};\ y_{\text{c}}^{\text{low}} \leq y \leq y_{\text{c}}^{\text{upp}}\right\}$, and $\mathbb{B} = \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$ contains $\mathcal{P}$ and $\mathcal{P}^\star$.

Output: An intermediate front $\widetilde{\mathcal{P}}_u$ such that $N(G(\widetilde{\mathcal{P}}_u); \mathcal{Y}_t) \approx \nu m$.

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
        $$\widetilde{\mathcal{P}}_u = \text{Pareto}(\mathcal{P} \cup \{y_{\text{anchor}}^1 + u(y_{\text{anchor}}^0 - y_{\text{anchor}}^1)\} \cup \cdots \cup \{y_{\text{anchor}}^q + u(y_{\text{anchor}}^0 - y_{\text{anchor}}^q)\}).$$

Algorithm 5: 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 (2.33) 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 $\mathcal{X}_n = \left(x_{n,k}, \frac{1}{m}\right)_{1 \leq k \leq m}$.

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 2.4.1 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 $\pi_{n+1}^{\mathbb{X}}$:

$$w_{n+1,k}^0 \propto \frac{\pi_{n+1}^{\mathbb{X}}(x_{n,k})}{\pi_n^{\mathbb{X}}(x_{n,k})} \, w_{n,k}. \tag{2.33}$$

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 $\pi_0, \pi_1, \ldots$ 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 5.

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 $\sum_k w_{n+1,k} \, \delta_{x_{n,k}}$ is not a good approximation of $\pi_{n+1}^{\mathbb{X}}$. This is similar to the problem explained in Section 2.4.1, 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[8]. 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 2.4.1. 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 5 is straightforward. It is very similar to the procedure described in Algorithms 2, 3 and 4 and is not repeated here for the sake of brevity.

[8] For a weighted sample of size n, the ESS is defined as the number of *random* samples that produces Monte-Carlo estimates with a variance equal to that of the weighted sample. As such, it can be viewed as a measure of the degeneracy of a weighted sample.

Remark 10 *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 5) is provided by Andrieu and Roberts (2009).*

Remark 11 *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 the target front* $\mathcal{P}^\star$ *(notation introduced in Algorithm 3).*

Remark 12 *For the sake of clarity, the number of particles used in the SMC approximation has been denoted by* m *both in Section 2.4.1 and in Section 2.4.2. Note that the two sample sizes are, actually, not tied to each other. We will denote them respectively by* $m_{\mathcal{X}}$ *and* $m_{\mathcal{Y}}$ *in the following sections.*

2.5 Experiments

2.5.1 Settings

The BMOO algorithm has been written in the Matlab/Octave programming language, using the Small Toolbox for Kriging (STK) (Bect et al., 2016b) for the Gaussian process modeling part. All simulation results have been obtained using Matlab R2014b.

In all our experiments except the illustration of Section 2.5.2, the algorithm is initialized with 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.

In Sections 2.5.3 and 2.5.4, the computation of the expected improvement is carried out using the SMC method described in Section 2.4.1. Taking advantage of Remark 9, the integration is performed only on the constraint space (prior to finding a feasible point) or the objective space (once a feasible point is found). In the case of single-objective problems (Section 2.5.3), we perform exact calculation using (2.16) once a feasible point has been observed. The parameter ν of Algorithm 3 is set to 0.2 and we take $m = m_{\mathcal{Y}} = 1000$ particles. The bounding hyper-

rectangles $\mathbb{B}_o$ and $\mathbb{B}_c$ are determined using the adaptive procedure described in Appendix 2.7.2 with $\lambda_o = \lambda_c = 5$.

For the optimization of the sampling criterion, we use the SMC method of Section 2.4.2, with $m = m_{\mathcal{X}} = 1000$ particles, residual resampling (Douc and Cappé, 2005), and an adaptive anisotropic Gaussian random walk Metropolis-Hastings algorithm to move the particles (Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009). When the probability of improvement cannot be written under closed-form, a Monte Carlo approximation (see Remark 10) with $N = 100$ simulations is used.

2.5.2 Illustration on a constrained multi-objective problem

In this section, the proposed method is illustrated 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

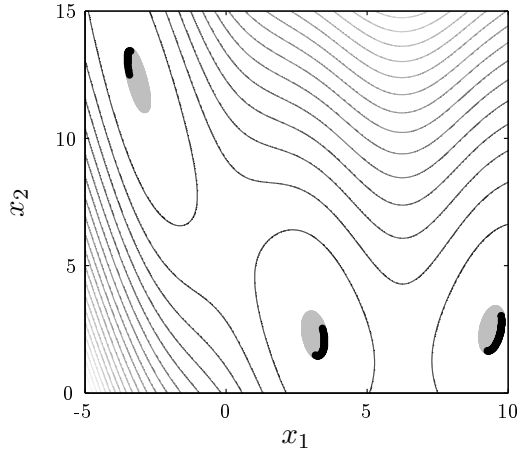
$$\left\{ \begin{array}{rcl} f_1 : (x_1, x_2) & \mapsto & -(x_1 - 10)^2 - (x_2 - 15)^2, \\ \\ f_2 : (x_1, x_2) & \mapsto & -(x_1 + 5)^2 - x_2^2, \\ \\ c : (x_1, x_2) & \mapsto & \left(x_2 - \dfrac{5.1}{4\pi^2} x_1^2 + \dfrac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \dfrac{1}{8\pi} \right) \cos(x_1) +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

The set of solutions to that problem is represented on Figure 2.5. 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.)

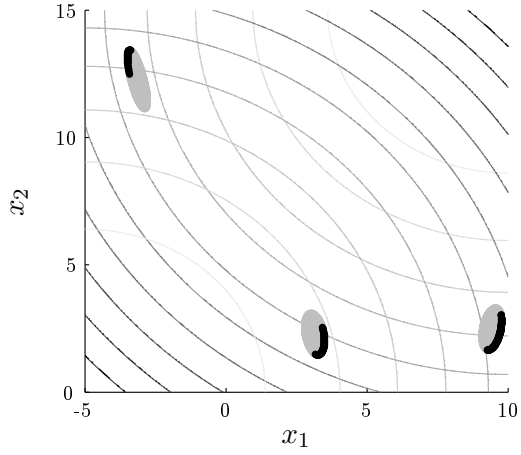
The behavior of BMOO is presented in Figure 2.6. The algorithm is initialized with $5d = 10$ function evaluations. Figure 2.6 shows that the algorithm correctly samples the three feasible regions, and achieves good covering of the solution Pareto front after only a few iterations. Note that no feasible solution is given at the beginning of the procedure and that the algorithm finds one after 10 iterations.

2.5.3 Mono-objective optimization benchmark

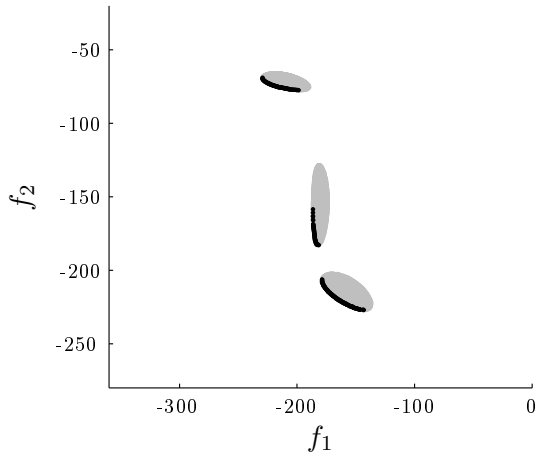
The first benchmark that we use to assess the performance of BMOO consists of a set of sixteen constrained single-objective test problems proposed by Regis (2014). Table 2.1 summarizes the main features of these problems. The input dimension d varies from 2 to 20, and the number q of constraints from 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



(a) Constraint function



(b) Objective functions



(c) Objective space

Figure 2.5: Figure (a) represents contour lines of the constraint function, and Figure (b) corresponds to contour lines of the two objective functions. The three gray areas correspond to the feasible region on Figures (a) and (b), and to the image of the feasible region by the objective functions on Figure (c). Thick dark curves correspond to the set of feasible and non-dominated solutions on Figures (a) and (b). On Figure (c), thick dark curves correspond to the Pareto front.

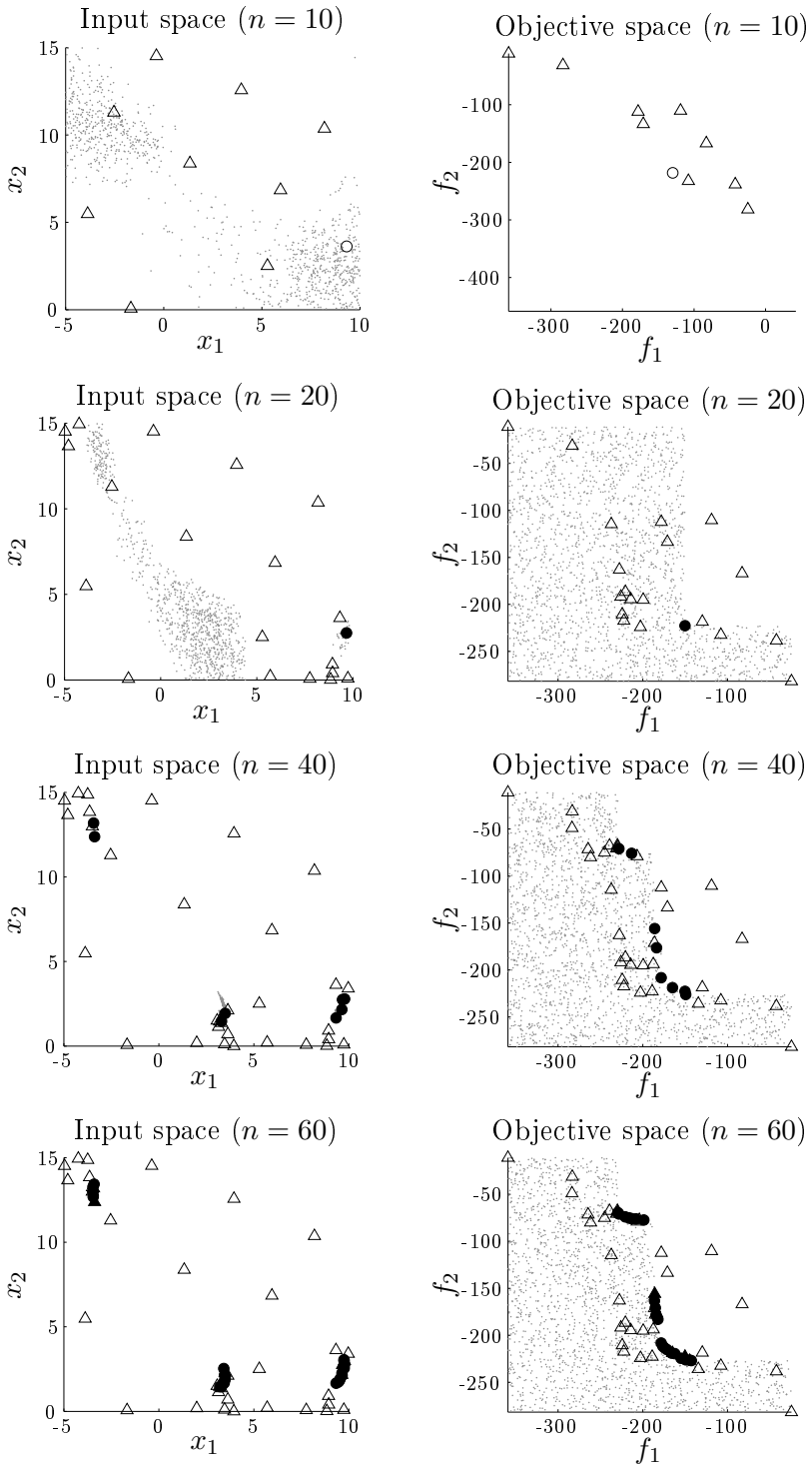


Figure 2.6: Convergence of the algorithm after $n = 10$, 20, 40 and 60 evaluations. The left column shows the input space $\mathbb{X}$, whereas the right one shows the objective space $\mathbb{B}_o$. Dominated observations are represented by triangles (filled or empty), and non-dominated ones by circles (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 2.4.2). Notice their progressive concentration in regions where improvements can be made. On the right column, the small dots represent the particles used for the computation of the expected improvement (see Section 2.4.1). Note in particular that they appear only when a feasible point has been observed: before that, the term ρ_n^{feas} (see Section 2.3.3) can be computed analytically.

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

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[9]. 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`[10], 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 2.5.1. For the local search algorithms

[9]The introduction of target values makes it possible to assess the convergence of the algorithms. The optimization can be stopped once a feasible observation with an objective value below the target value has been made.

[10]Optimization toolbox v7.1, MATLAB R2014b

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 the results presented by the author; the reader is referred to the original article for more details about the settings. Results are presented in Tables 2.2 and 2.3. In both tables, a solution is considered as feasible when there is no constraint violation larger than 10^{-5}.

In Table 2.2, 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 are presented in Appendix 2.7.4. 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 2.3 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 2.1 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 2.7.4 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 2.3. 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 2.5.1). 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, as we can see in Table 2.4, the performances of BMOO are greatly improved in all three cases if a transformation of the form $f \to f^\lambda$ (for $\lambda > 0$) is applied to the functions that cause the problem (see Appendix 2.7.3 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

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 2.2: 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 2.12 and 2.14 in Appendix 2.7.4 for more detailed results. Dash symbols are used when a value cannot be calculated.

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 2.3: Number of evaluations to reach specified target. See Table 2.2 for conventions. See Tables 2.13 and 2.15 in Appendix 2.7.4 for more detailed results.

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 region 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.,

$$\mathbb{P}_n \left(\xi_{\mathrm{c},j}(x) \leq 0 \right) \approx \begin{cases} 1, & \text{if } c_j(x) \leq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{2.34}$$

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 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 2.5 shows the promising results obtained with this modified density on g18. Further investigations on this particular issue can be found in Section 3.2.

2.5.4 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 2.6. The input 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 2.1, 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

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 2.4: 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 2.5: Number of evaluations to find a first feasible point and to reach the target using a modified probability density function for the criterion optimization.

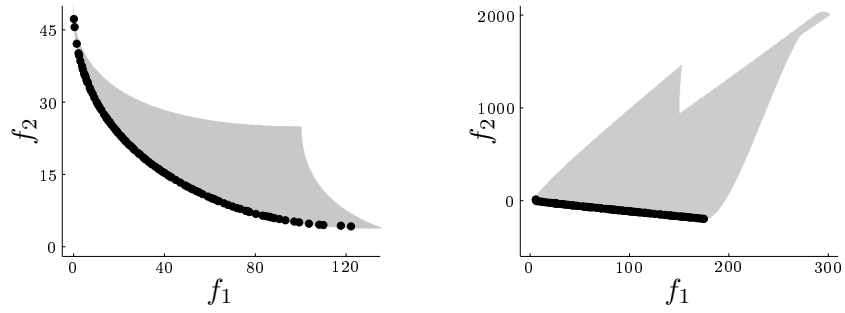
Pbm	d	q	p	$\Gamma(\%)$	V	y_o^{ref}
BNH	2	2	2	93,6	5249	[140; 50]
SRN	2	2	2	16,1	31820	[200; 50]
TNK	2	2	2	5,1	0,6466	[1,2; 1,2]
OSY	6	6	2	3,2	16169	[0; 80]
TwoBarTruss	3	1	2	86,3	4495	[0,06; 10^5]
WeldedBeam	4	4	2	45,5	0,4228	[50; 0,01]
CONSTR	2	2	2	52,5	3,8152	[1; 9]
WATER	3	7	5	92	0,5138	[1; 1; 1;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

Table 2.6: Main features of the multi-objective problems in our benchmark.

region dominated by the Pareto front[11], measured using a reference point y_o^{ref}, whose coordinates are specified in the last column. As an illustration, the result of one run of BMOO is shown on Figures 2.7 and 2.8, for each test problem.

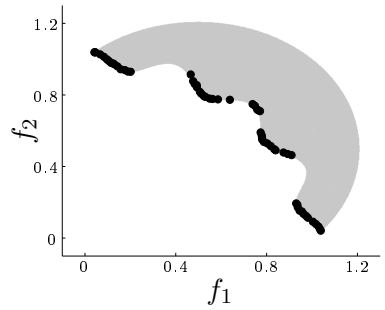
To the best of our knowledge, published state-of-the-art methods to solve multi-objective optimization problems in the presence of non-linear constraints are based on genetic or evolutionary approaches. The most popular algorithms are probably NSGA2 (Deb et al., 2002) and SPEA2 (Zitzler et al., 2002). Such algorithms, however, are not primarily designed to work on a limited budget of function evaluations. Some methods that combine genetic/evolutionary approaches and surrogate modeling techniques have been proposed in the literature (see, e.g., Emmerich et al., 2006; Jin, 2011, and references therein), but a quantitative comparison with these methods would necessitate to develop proper implementations, which is out of the scope of this work. In this section, we shall limit ourselves to emphasizing advantages and limitations of the proposed approach. Since the ability of the BMOO algorithm to find feasible solutions has already been demonstrated in Section 2.5.3, we will focus here on the other contributions of the chapter: the SMC methods for the computation and optimization of the expected improvement sampling criterion.

[11] This volume has been obtained using massive runs of the `gamultiobj` algorithm of Matlab. It might be slightly under-estimated.

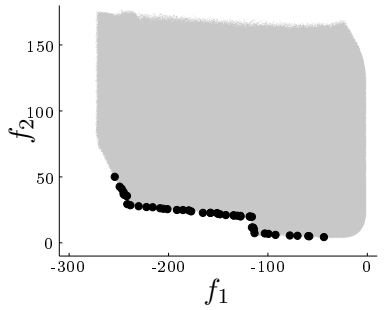


(a) BNH

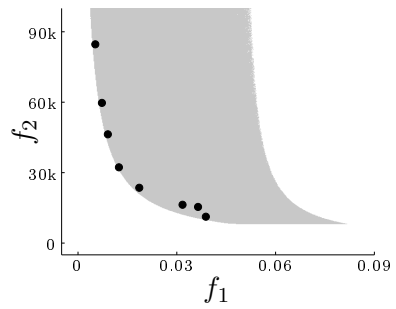
(b) SRN



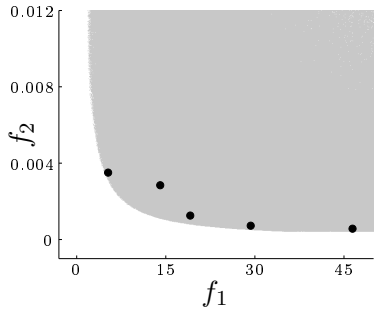
(c) TNK



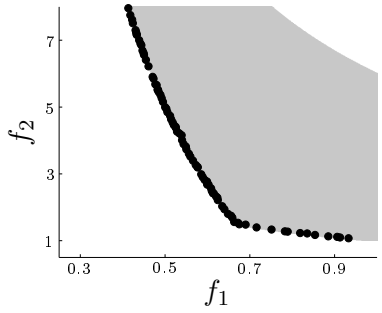
(d) OSY



(e) TwoBarTruss



(f) WeldedBeam



(g) CONSTR

Figure 2.7: Result of one run of the BMOO algorithm on the bi-objective problems of Table 2.6, with $n = 100$ evaluations. Black dots represent non-dominated solutions. For bi-objective problems, the set of feasible objective values is shown in gray. On the subfigure corresponding to the WeldedBeam problem, a zoom has been made to improve visualization.

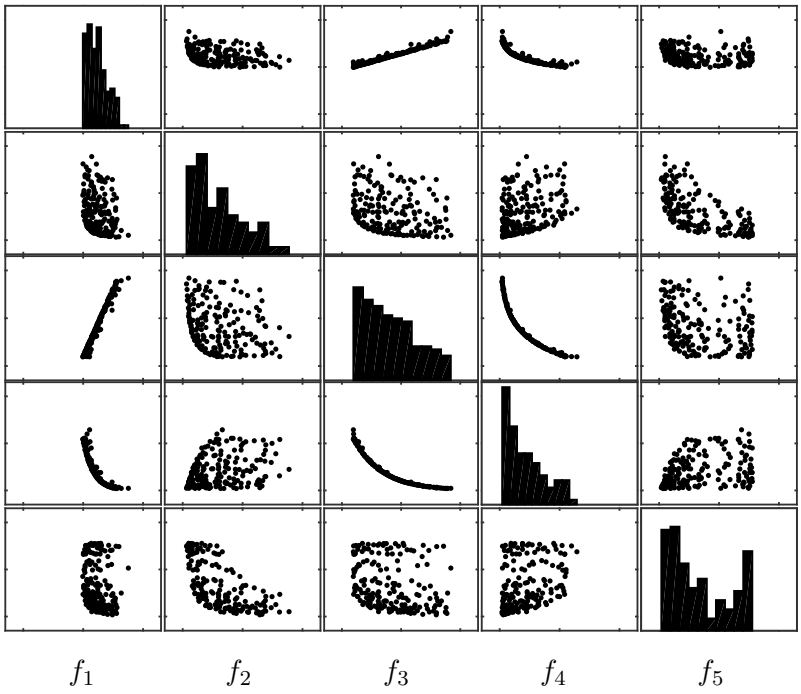


Figure 2.8: Result of one run of the BMOO algorithm on the WATER problem of Table 2.6, with $n = 200$ evaluations. Black dots represent non-dominated solutions.

First, we demonstrate the effectiveness of the proposed SMC algorithm for optimizing expected improvement based criteria. We compare our SMC approach (see Section 2.4.2) with the approach used by Couckuyt et al. (2014), that we shall call MCSQP (for Monte-Carlo Sequential Quadratic Programming). This approach consists in selecting the best point out of a population of candidates uniformly distributed on the search space $\mathbb{X}$, and then running an SQP algorithm starting from this point. In our experiments, the number of candidates is chosen equal to the population size $m_{\mathcal{X}} = 1000$ of the SMC method.

Table 2.7 presents experimental results obtained with the extended EHVI criterion proposed in Section 2.3 as a sampling criterion. As a preliminary remark, observe that the finest target precision is systematically reached by our SMC method in all but three test cases (OSY, TwoBarTruss and WeldedBeam). The OSY case will be discussed below. On the TwoBarTruss and WeldedBeam problems, we found out that the poor performances are due to Gaussian process modeling issues, similar to those encountered earlier on the g3mod, g10 and PVD4 test problems (see Section 2.5.3). The results on these problems are thus left out of the analyses in the following, but will motivate future work on the models, as concluded in Section 2.5.3. Regarding the optimization of the criteria, the results show that our SMC approach compares very favorably with MCSQP. More specifically, we note a drop of performance of the MCSQP method compared with the SMC approach as we try to converge more finely toward the Pareto front (see, in particular, column "Target 99%" of Table 2.7, but this is also visible in the other columns as well for most of the test cases). Because of its sequential nature, the SMC approach is able to track much more efficiently the concentration of the sampling criterion in the search domain, and thus

makes it possible to reach higher accuracy.

Tables 2.8 and 2.9 provide additional results obtained when performing the same study with respectively the EMMI and WCPI criteria[12] (see Svenson and Santner, 2010; Keane, 2006, respectively). These criteria are not primarily designed to address constrained problems, but they can easily be extended to handle constraints by calculating them using only feasible values of the objectives, and then multiplying them by the probability of satisfying the constraints (as explained in Section 2.2.3). When no feasible solution is available at the start of the optimization procedure, we use the probability of feasibility as a sampling criterion, as advised by Gelbart et al. (2014). The conclusions drawn from Table 2.7 for the extended EHVI criterion carry through to the results presented in Tables 2.8–2.9. It shows that the SMC algorithm proposed in Section 2.4.2 can be viewed as a contribution of independent interest for optimizing improvement-based sampling criteria.

Next we study the influence on the convergence of the algorithm of the number $m = m_{\mathcal{Y}}$ of particles used in Algorithm 1 for approximating the expected improvement value. In Tables 2.10 and 2.11 we compare the number of evaluations required to dominate successively 90%, 95% and 99% of the volume V of Table 2.6 when using different numbers of particles. As expected, the overall performances of the algorithm deteriorate when the number $m_{\mathcal{Y}}$ of particles used to approximate the expected improvement decreases. However, the algorithm maintains satisfactory convergence properties even with a small number of particles. For reference, we have also included results obtained by choosing the evaluation point randomly in the set of candidate points. Notice that these results are always much worse than those obtained using the sampling criterion with $m_{\mathcal{Y}} = 200$. This shows that not all candidate points are equally good, and thus confirms that the sampling criterion, even with a rather small value of $m_{\mathcal{Y}}$, is effectively discriminating between good and bad candidate points.

We observe poor performances of the BMOO algorithm on the OSY test problem, regardless of the number of particles that are used to estimate the expected improvement. Figure 2.9 reveals that this is due to the choice of a uniform sampling density on $\mathbb{B}_o \setminus H_n$ as the target density of the SMC algorithm used for the approximate computation of the criterion. Indeed, most of the particles do not effectively participate to the approximation of the integral, since they lie outside the set of feasible objective values (see Figure 2.7(d)). Further work is required on this topic to propose a better sampling density, that would concentrate on objective values that are likely to be feasible (instead of covering uniformly the entire non-dominated region $\mathbb{B}_o \setminus H_n$).

In practice, for problems with a small number of objectives, and especially for bi-objective problems, we do not recommend the use of our SMC algorithm for the (approximate) computation of the EHVI criterion since exact and efficient domain-decomposition-based algorithms are available (see Hupkens et al., 2014; Couckuyt et al., 2014, and references therein). An in-depth study of the quality of the approximation provided by our SMC method, and a comparison with exact methods, is therefore needed before more precise recommandations can be made.

[12] An implementation of the EMMI criterion is available in the STK. An implementation of the WCPI sampling crtiterion for bi-objective problems is distributed alongside with Forrester et al.'s book (Forrester et al., 2008).

Problem	optimizer	Target 90%		Target 95%		Target 99%	
BNH	SMC	30	8.5 (0.6)	30	12.7 (0.7)	30	34.6 (1.3)
	MCSQP	30	8.4 (0.6)	30	12.8 (0.7)	30	38.9 (2.2)
SRN	SMC	30	16.7 (0.9)	30	22.4 (1.0)	30	52.6 (4.1)
	MCSQP	30	20.5 (2.4)	30	35.6 (5.9)	0	> 250 (-)
TNK	SMC	30	35.5 (2.6)	30	44.1 (2.5)	30	71.1 (4.0)
	MCSQP	30	43.5 (4.6)	30	71.6 (11.3)	0	> 250 (-)
OSY	SMC	30	29.0 (1.7)	30	38.2 (3.4)	13	119.8 (53.0)
	MCSQP	0	> 250 (-)	0	> 250 (-)	0	> 250 (-)
TwoBarTruss	SMC	22	90.9 (62.0)	1	234 (-)	0	> 250 (-)
	MCSQP	26	88.7 (68.4)	2	162.0 (29.7)	0	> 250 (-)
WeldedBeam	SMC	28	146.5 (41.1)	2	212 (33.9)	0	> 250 (-)
	MCSQP	26	171.3 (46.9)	1	229.0 (-)	0	> 250 (-)
CONSTR	SMC	30	12.4 (1.0)	30	19.2 (1.4)	30	83.5 (5.9)
	MCSQP	30	13.8 (1.4)	30	26.3 (3.3)	0	> 250 (-)
WATER	SMC	30	48.3 (3.6)	30	80.7 (5.6)	30	139.1 (8.0)
	MCSQP	30	53.5 (4.8)	30	88.7 (7.5)	30	164.3 (9.6)

Table 2.7: Results achieved when using either SMC or MCSQP for the optimization of the extended EHVI, on the problems of Table 2.6. We measure the number of function evaluations for dominating successively 90%, 95% and 99% of the volume V. For each target, 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). Dash symbols are used when a value cannot be calculated.

Problem	optimizer	Target 90%		Target 95%		Target 99%	
BNH	SMC	30	9.8 (1.1)	30	15.9 (1.5)	30	41.2 (2.8)
	MCSQP	30	9.5 (0.7)	30	15.4 (1.4)	30	42.6 (2.4)
SRN	SMC	30	15.5 (1.2)	30	21.0 (1.4)	30	48.3 (2.8)
	MCSQP	30	18.6 (1.8)	30	29.1 (2.7)	30	90.9 (9.0)
TNK	SMC	30	47.7 (3.5)	30	61.8 (4.4)	30	100.2 (5.4)
	MCSQP	30	60.6 (8.2)	30	94.3 (13.2)	5	224.2 (15.0)
OSY	SMC	30	32.3 (2.9)	30	41.9 (3.9)	25	73.6 (20.8)
	MCSQP	0	> 250 (-)	0	> 250 (-)	0	> 250 (-)
TwoBarTruss	SMC	28	116.5 (48.5)	3	199.0 (24.1)	0	> 250 (-)
	MCSQP	26	130.9 (63.9)	1	174.0 (-)	0	> 250 (-)
WeldedBeam	SMC	16	156.6 (50.5)	4	177.0 (40.5)	0	> 250 (-)
	MCSQP	9	161.9 (60.1)	3	156.0 (35.8)	0	> 250 (-)
CONSTR	SMC	30	22.1 (2.5)	30	33.8 (3.0)	30	100.9 (8.6)
	MCSQP	30	18.4 (2.1)	30	30.9 (3.1)	30	154.8 (9.0)
WATER	SMC	30	60.4 (6.5)	30	93.4 (8.8)	30	153.9 (9.0)
	MCSQP	30	68.2 (8.1)	30	103.9 (11.3)	30	172.7 (13.7)

Table 2.8: Results achieved when using the EMMI criterion. See Table 2.7 for more information.

Problem	optimizer	Target 90%		Target 95%		Target 99%	
BNH	SMC	30	20.9 (8.9)	30	43.4 (7.6)	30	132.4 (15.4)
	MCSQP	30	18.7 (8.2)	30	49.0 (14.2)	30	176.1 (29.1)
SRN	SMC	30	39.1 (6.0)	30	57.53 (7.5)	30	154.9 (12.8)
	MCSQP	20	154.5 (62.1)	1	248.0 (-)	0	> 250 (-)
TNK	SMC	30	53.3 (6.8)	30	68.3 (6.9)	30	120.8 (13.7)
	MCSQP	0	> 250 (-)	0	> 250 (-)	0	> 250 (-)
OSY	SMC	30	39.7 (5.7)	29	61.5 (22.0)	14	123.0 (41.9)
	MCSQP	0	> 250 (-)	0	> 250 (-)	0	> 250 (-)
TwoBarTruss	SMC	29	70.1 (40.3)	8	180.4 (40.0)	0	> 250 (-)
	MCSQP	29	69.6 (47.3)	11	185.2 (53.0)	0	> 250 (-)
WeldedBeam	SMC	0	> 250 (-)	0	> 250 (-)	0	> 250 (-)
	MCSQP	0	> 250 (-)	0	> 250 (-)	0	> 250 (-)
CONSTR	SMC	30	40.0 (5.6)	30	60.4 (7.8)	30	212.1 (15.6)
	MCSQP	30	42.2 (16.0)	26	150.7 (42.8)	0	> 250 (-)

Table 2.9: Results achieved when using the WCPI criterion. See Table 2.7 for more information.

Problem	EHVI	Target 90%		Target 95%		Target 99%	
BNH	SMC ($m_{\mathcal{Y}} = 5000$)	30	8.3 (0.7)	30	12.5 (0.5)	30	32.8 (1.0)
	SMC ($m_{\mathcal{Y}} = 1000$)	30	8.5 (0.6)	30	12.7 (0.7)	30	34.6 (1.3)
	SMC ($m_{\mathcal{Y}} = 200$)	30	8.8 (0.6)	30	13.1 (0.7)	30	39.2 (2.0)
	random	30	12.8 (2.7)	30	29.6 (6.0)	30	106.8 (13.2)
SRN	SMC ($m_{\mathcal{Y}} = 5000$)	30	16.3 (1.0)	30	21.6 (1.1)	30	47.3 (2.1)
	SMC ($m_{\mathcal{Y}} = 1000$)	30	16.7 (0.9)	30	22.4 (1.0)	30	52.6 (4.1)
	SMC ($m_{\mathcal{Y}} = 200$)	30	16.6 (1.3)	30	23.0 (1.9)	30	60.9 (6.9)
	random	30	30.6 (5.2)	30	51.1 (8.2)	30	146.2 (13.2)
TNK	SMC ($m_{\mathcal{Y}} = 5000$)	30	36.2 (4.4)	30	43.4 (3.6)	30	65.1 (3.1)
	SMC ($m_{\mathcal{Y}} = 1000$)	30	35.5 (2.6)	30	44.1 (2.5)	30	71.1 (4.0)
	SMC ($m_{\mathcal{Y}} = 200$)	30	37.7 (4.1)	30	48.4 (5.0)	30	87.3 (5.9)
	random	30	64.0 (10.3)	30	94.2 (12.4)	29	193.3 (27.4)

Table 2.10: Results achieved on the BNH, SRN and TNK problems of Table 2.6 when using successively $m_{\mathcal{Y}} = 200, 1000$ and 5000 particles for the approximate computation of the extended EHVI criterion. For reference, results obtained by selecting the evaluation point randomly in the pool of candidates points are provided ("random" rows). See Table 2.7 for more information.

Problem	EHVI	Target 90%		Target 95%		Target 99%	
OSY	SMC ($m_{\mathcal{Y}} = 5000$)	30	28.6 (2.0)	30	36.0 (2.8)	22	82.5 (33.5)
	SMC ($m_{\mathcal{Y}} = 1000$)	30	29.0 (1.7)	30	38.2 (3.4)	13	119.8 (53.0)
	SMC ($m_{\mathcal{Y}} = 200$)	30	32.4 (3.1)	29	49 (16.0)	5	164.8 (54.6)
	random	30	140.2 (21.0)	25	203.4 (21.4)	0	> 250 (-)
CONSTR	SMC ($m_{\mathcal{Y}} = 5000$)	30	12.2 (0.7)	30	18.0 (1.0)	30	68.8 (4.7)
	SMC ($m_{\mathcal{Y}} = 1000$)	30	12.4 (1.0)	30	19.2 (1.4)	30	83.5 (5.9)
	SMC ($m_{\mathcal{Y}} = 200$)	30	12.9 (1.2)	30	21.0 (1.6)	30	109.2 (10.7)
	random	30	31.1 (6.6)	30	58.1 (8.5)	18	235.1 (11.0)
WATER	SMC ($m_{\mathcal{Y}} = 5000$)	30	45.8 (4.0)	30	75.3 (6.2)	30	127 (8.2)
	SMC ($m_{\mathcal{Y}} = 1000$)	30	48.3 (3.6)	30	80.7 (5.6)	30	139.1 (8.0)
	SMC ($m_{\mathcal{Y}} = 200$)	30	52.5 (4.5)	30	88.6 (6.0)	30	154.8 (8.8)
	random	14	223.2 (15.4)	0	> 250 (-)	0	> 250 (-)

Table 2.11: Results achieved on the OSY, CONSTR and WATER problems of Table 2.6 when using successively $m_{\mathcal{Y}} = 200, 1000$ and 5000 particles for the approximate computation of the extended EHVI criterion. For reference, results obtained by selecting the evaluation point randomly in the pool of candidates points are provided ("random" rows). See Table 2.7 for more information.

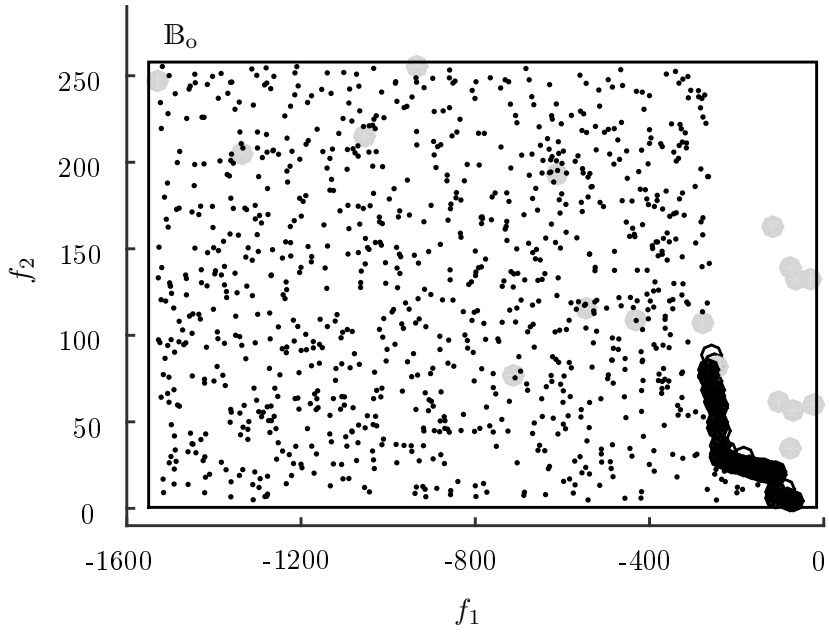


Figure 2.9: An illustration, in the objective domain $\mathbb{Y}_o$, of BMOO running on the OSY test problem. The small dots are the particles used for the computation of the expected improvement. They are uniformly distributed on the non-dominated subset of $\mathbb{B}_o$. Dark disks indicate the non-dominated solutions found so far, light gray disks indicate the dominated ones and empty black circles are used to indicate the non-feasible solutions.

2.6 Conclusions and future work

In this chapter, a new Bayesian optimization approach is proposed to solve multi-objective optimization problems with 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 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. The proposed sampling techniques borrow ideas from the literature of structural reliability for estimating the probability of rare events, and can be viewed as a contribution in itself.

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[13].

[13]BMOO ranked 2nd at the EMO'2017 Real-world Problems Track BBComp competition (see

Several questions are left open for future work. First, our numerical studies reveal that the choice of sampling densities in the input domain (as demonstrated by unsatisfactory results on the g18 test problem) and in the output domain (as shown on the OSY case) could be improved. Suggestions for improvement are proposed in the chapter and will be the object of future investigations. Second, an in-depth study of the quality of the approximation provided by our SMC method, and a comparison with exact methods, is needed before recommandations can be made on when to switch between exact and approximate calculation of the expected improvement, and how to select the sample size—possibly in an adaptive manner—used for the SMC approximation. Last, the choice of the random processes used for modeling objective and constraint functions deserves more attention. Stationary Gaussian process models have been found to lack flexibility on some single- and multi-objective cases (g3mod, g10, PVD4, TwoBarTruss and WeldedBeam). Several types of models proposed in the literature—warped Gaussian processes (Snelson et al., 2004), non-stationary Gaussian processes (Toal and Keane, 2012, see), deep Gaussian processes (Damianou and Lawrence, 2013), etc.—provide interesting directions regarding this issue.

2.7 Additional Material

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

We have assumed in Section 2.3 that $\mathbb{B}_o$ and $\mathbb{B}_c$ are *bounded* hyper-rectangles; that is, sets of the form

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

for some $y_o^{\text{low}}, y_o^{\text{upp}} \in \mathbb{Y}_o$ and y_c^{low}, $y_c^{\text{upp}} \in \mathbb{Y}_c$, with the additional assumption that $y_{c,j}^{\text{low}} < 0 < y_{c,j}^{\text{upp}}$ for all $j \leq q$. Remember that upper bounds only where required in the unconstrained case discussed in Section 2.2.2. 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

$$|H_1| = |\mathbb{B}_o| \cdot \prod_{j=1}^{q} \left(y_{c,j}^{\text{upp}} - y_{c,j}^{\text{low}}\right)^{\gamma_j} \left(y_{c,j}^{\text{upp}} - \xi_{c,j}(X_1)\right)^{1-\gamma_j} \tag{2.35}$$

where $\gamma_j = \mathbb{1}_{\xi_{c,j}(X_1)\leq 0}$. It is clear from the right-hand side of (2.35) that both $\mathbb{B}_o$ and $\mathbb{B}_c$ have 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 (2.35) also reveals that the improvement is a discontinuous function of the observations: indeed, the j^{th} term in the product jumps from $y_{c,j}^{\text{upp}}$ to $y_{c,j}^{\text{upp}} - y_{c,j}^{\text{low}} > y_{c,j}^{\text{upp}}$ when $\xi_{c,j}(X_1)$ goes from 0^+ to 0. The increment $-y_{c,j}^{\text{low}}$ can be thought of as a reward associated to finding a point

https://bbomp.ini.rub.de/results/BBComp2017EMO/summary.html).

which is feasible with respect to the j^{th} constraint.

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

$$|H_1| = |\mathbb{B}_{\text{o}}| \cdot \left(|\mathbb{B}_{\text{c}}| - \left|\mathbb{B}_{\text{c}}^{-}\right|\right) + \prod_{j \leq p} \left(\min\left(\xi_{\text{o},j}(X_1), y_{\text{o},j}^{\text{upp}}\right) - \max\left(\xi_{\text{o},j}(X_1), y_{\text{o},j}^{\text{low}}\right)\right) \cdot \left|\mathbb{B}_{\text{c}}^{-}\right|, \quad (2.36)$$

where $|\mathbb{B}_{\text{c}}^{-}| = \prod_{j=1}^{q} \left|y_{\text{c},j}^{\text{low}}\right|$ is the volume of the feasible subset of $\mathbb{B}_{\text{c}}$, $\mathbb{B}_{\text{c}}^{-} = \mathbb{B}_{\text{c}} \cap]-\infty; 0]^q$. The first term in the right-hand side of (2.36) is the improvement associated to the domination of the entire unfeasible subset of $\mathbb{B} = \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$; the second term measures the improvement in the space of objective values.

2.7.2 An adaptive procedure to set $\mathbb{B}_{\text{o}}$ and $\mathbb{B}_{\text{c}}$

This section describes the adaptive numerical procedure that is used, in the numerical experiments of Section 2.5, to define the hyper-rectangles $\mathbb{B}_{\text{o}}$ and $\mathbb{B}_{\text{c}}$. As said in Section 2.3.3, 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 2.4.2).

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

$$\left\{ \begin{array}{rcl} y_{\text{o},i,n}^{\text{low}} & = & \min\left(\min_{i \leq n} \xi_{\text{o},i}(X_i),\ \min_{x \in \mathcal{X}_n} \widehat{\xi}_{\text{o},\,i,\,n}(x) - \lambda_{\text{o}} \sigma_{\text{o},\,i,\,n}(x)\right), \\ y_{\text{o},i,n}^{\text{upp}} & = & \max\left(\max_{i \leq n} \xi_{\text{o},i}(X_i),\ \max_{x \in \mathcal{X}_n} \widehat{\xi}_{\text{o},\,i,\,n}(x) + \lambda_{\text{o}} \sigma_{\text{o},\,i,\,n}(x)\right), \end{array} \right. \quad (2.37)$$

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

$$\left\{ \begin{array}{rcl} y_{\text{c},j,n}^{\text{low}} & = & \min\left(0,\ \min_{i \leq n} \xi_{\text{c},j}(X_i),\ \min_{x \in \mathcal{X}_n} \widehat{\xi}_{\text{c},\,j,\,n}(x) - \lambda_{\text{c}} \sigma_{\text{c},\,j,\,n}(x)\right), \\ y_{\text{c},j,n}^{\text{upp}} & = & \max\left(0,\ \max_{i \leq n} \xi_{\text{c},j}(X_i),\ \max_{x \in \mathcal{X}_n} \widehat{\xi}_{\text{c},\,j,\,n}(x) + \lambda_{\text{c}} \sigma_{\text{c},\,j,\,n}(x)\right), \end{array} \right. \quad (2.38)$$

for $1 \leq j \leq q$, where λ_{o} and λ_{c} are positive numbers.

2.7.3 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 2.5.3 to overcome the modeling problems with BMOO. Our modifications are shown in boldface. The rationale of the modifications is to smooth local jumps.

- modified-g3mod problem

$$\left\{ \begin{array}{rcl} f(x) & = & -\text{plog}((\sqrt{d})^d \prod_{i=1}^{d} x_i)^{\mathbf{0.1}} \\ c(x) & = & (\sum_{i=1}^{d} x_i^2) - 1 \end{array} \right.$$

- 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)).

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

2.7.4 Mono-objective benchmark result tables

In Section 2.5.3, 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 2.12 and 2.13, and Tables 2.14 and 2.15 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 2.1). Table 2.12 and Table 2.13 show the performances for finding feasible solutions and for reaching the targets specified in Table 2.1 for the COBYLA, Active-Set, Interior-Point and SQP algorithms. Similarly, Table 2.14 and Table 2.15 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).

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 2.12: Number of evaluations to find a first feasible point for the COBYLA, Active-Set, Interior-Point and SQP local optimization algorithms. See Table 2.2 for conventions.

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 2.13: Number of evaluations to reach the target for the COBYLA, Active-Set, Interior-Point and SQP local optimization algorithms. See Table 2.2 for conventions.

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 2.14: 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).See Table 2.2 for conventions.

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 2.15: 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). See Table 2.2 for conventions.

Chapter 3

Improvements and extensions of the BMOO algorithm

3.1 Introduction

This chapter addresses some of the perspectives raised in Chapter 2. It is organized as follows.

In Section 3.2, we address the limitations regarding the optimization of the new EI criterion. In Chapter 2 it was empirically demonstrated that the use of the probability of improvement for optimizing the new criterion yields unsatisfactory results on problems having many constraints. Here, we design a test problem to validate our hypotheses regarding the causes of this issue. We show that the performances of BMOO are improved when a more suitable density is used for optimizing the EI criterion.

In Section 3.3 we discuss in more details the computation of the criterion. It was demonstrated on the OSY problem in Section 2.5.4 that the uniform density may not be efficient for approximating the criterion. In this section, we introduce a novel sampling density which has better approximation performances than the uniform density and discuss the choice of the sample size to be used in the SMC procedure. A comparison between approximate and exact computation of the criterion is made and recommendations are given at the end of the section.

In Section 3.4, we assess the performances of the BMOO algorithm on many-objective optimization problems. First, a test problem for which the curvature of the Pareto front and the numbers of objectives can be controlled is introduced. Then, BMOO is compared with a strategy where the hypervolume indicator is maximized sequentially on this problem. The influence of the choice of the reference point in the definition of the hypervolume is discussed and the adaptive procedure to set this reference point is revisited.

Finally, in Section 3.5, we discuss miscellaneous extensions of the algorithm. BMOO is extended to handle non-hypercubic design spaces and hidden constraints. A multi-point strategy for batch sequential optimization is proposed and the incorporation of user preferences is discussed.

3.2 Efficient optimization of the EI criterion

3.2.1 Introduction

In this thesis, we study the possibility of optimizing EI-based criteria using sequential Monte-Carlo techniques. The idea is to exploit the stability in time of the interesting regions during the optimization process (Benassi et al., 2012). At time n, our objective is to obtain a weighted sample $\mathcal{X}_n$ from a density of interest $\pi_n^{\mathbb{X}}$, that concentrates in regions of high EI value and stays "stable" in time. These are antagonist objectives and the choice of a density $\pi_n^{\mathbb{X}}$ that makes a suitable trade-off between them is crucial to the performances of the optimization algorithm.

In the work of Benassi et al. (2012), the probability of improvement is used as an unnormalized probability density function, and good results are obtained in the single-objective case: this is the path that we followed in Chapter 2. However, the numerical experiments of Section 2.5.3 revealed issues with this choice of density when dealing with problems having numerous constraints, as observed on the g18 problem. An alternative density was proposed to better concentrate the particles in regions of high EI values, with promising results.

In this section, we further study alternative densities to optimize the EI criterion. In particular, we focus on the optimization of the criterion before feasible solutions are known, i.e. in the phase where the points are compared using the Pareto domination rule applied to the vector of constraint violations (see Section 2.3.1). The section is organized as follows. After having presented a test problem crafted to exhibit small feasibility regions and a large number of constraints, we propose three novel sampling densities. A comparison of performances is made and one particular density is shown to outperform the others. This density, that we call PICPI, achieves very good results both on the test problem and on the problems of Section 2.5.3. Besides, it can be computed in closed form when independence between the functions of the problem is assumed, which makes it computationally efficient.

3.2.2 The YUCCA test problem

We define the following single-objective test problem:

$$
\begin{array}{l}
\text{YUCCA } (d, \kappa): \\
\quad [-1,1]^d \quad \to \quad \mathbb{R} \\
x = (x_1, \ldots, x_d) \quad \mapsto \quad \left\{ \begin{array}{lcll} f(x) & = & \sum_{1 \leq i \leq d} \left(x_i - x_i^{\text{opt}} \right)^2 & , \\ c_{2i-1}(x) & = & \sin\left(x_i - x_i^{\text{opt}} - \epsilon \right) & , \quad 1 \leq i \leq d\,, \\ c_{2i}(x) & = & \sin\left(x_i^{\text{opt}} - x_i + \epsilon \right) & , \quad 1 \leq i \leq d\,, \end{array} \right.
\end{array}
$$

where $\epsilon = 10^{-\kappa}$ and $x_i^{\text{opt}} = -1 + \frac{2i-1}{2d}$, $1 \leq i \leq d$. The parameter d (the dimension of the search space) also controls the number of constraints of the problem since $q = 2d$. The parameter κ controls the size of the excursion set of the constraints below zero. For this problem, the feasible region is an hypercube of side length 2ϵ, centred on x^{opt}, which is the global optimum of the optimization problem[1]. The ratio between the volume of the feasible set and the volume of the design space is $r = \frac{(2\epsilon)^d}{2^d} = \epsilon^d$. It becomes small when κ and d increase. An illustration of the YUCCA test problem with $\kappa \in \{1, 2\}$ and $d = 2$ is presented in Figure 3.1.

The constraints of the problem only depend on one variable at a time and they are easily modeled using Gaussian processes. Besides, the minimization of the objective is straightforward. The difficulty of the problem lies in the simultaneous satisfaction of all constraints when d and κ increase. Indeed, the constraints are pairwise antagonist and satisfying simultaneously c_{2i-1} and c_{2i}, for a given $i \in [\![1, d]\!]$, is difficult. As an example, consider the problem with $\kappa = 2$ and $d = 20$. Then, the ratio between the volume of the feasible region and the volume of the design space is $r = 10^{-40}$ and one cannot expect to find a feasible solution without an efficient optimization procedure.

[1] Here it is implicitly assumed that $\epsilon \leq \frac{1}{2d}$. Otherwise, the feasible region is located near the border of the domain.

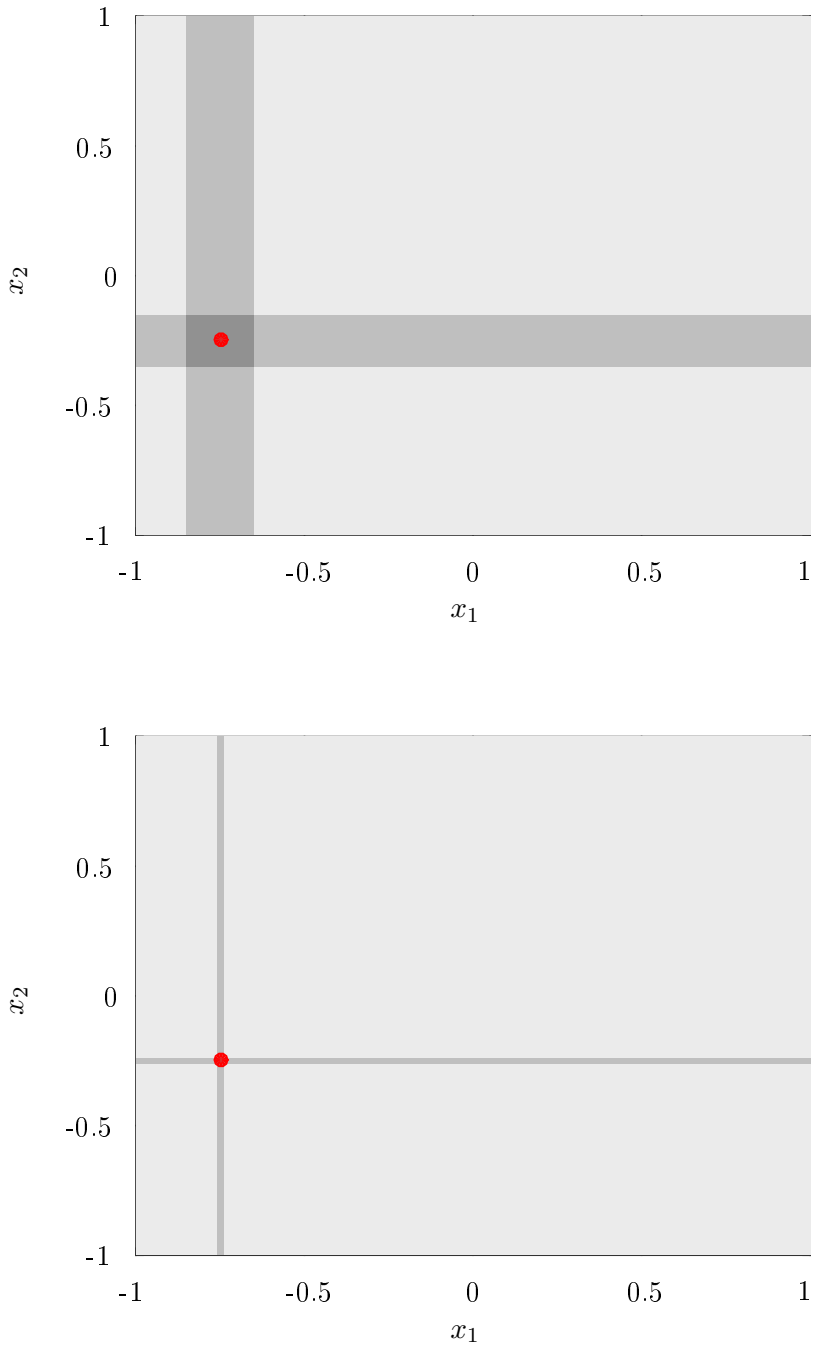


Figure 3.1: Illustration of the YUCCA test problem with $d = 2$ and $\kappa = 1$ (top) or $\kappa = 2$ (bottom). The regions where constraints are satisfied are represented in shades of gray. Darker shades of gray indicate overlapping regions. The optimum is represented as a red disk.

3.2.3 Failure of the probability of improvement sampling density on the YUCCA test problem

Here we explain the causes of the failure of the probability of improvement on the YUCCA test problem.

Consider a set of N points $X_k = (X_{k,1}, \ldots, X_{k,d})$, $k \in [\![1, N]\!]$, uniformly distributed on $\mathbb{X}$. For every point X_k, $k \in [\![1, N]\!]$, only constraints c_{2i-1} and c_{2i}, $i \in [\![1, d]\!]$, depend on the value of $x_{k,i}$. Thus, the simultaneous satisfaction of the pairs of constraints (c_{2i-1}, c_{2i}) and (c_{2j-1}, c_{2j}) when $i \neq j$ are two independent events. Assume for simplicity that $\epsilon \leq \frac{1}{2d}$ so that the feasible region is not near the boundary of the domain. Since $x_{k,i}$ is uniformly distributed on the interval $[-1, 1]$, the probability that both c_{2i-1} and c_{2i} are satisfied is equal to 2ϵ and the probability that only one of the two is satisfied is equal to $1 - 2\epsilon$ (any $x \in [-1, 1]^d$ satisfies at least d constraints, see Figure 3.1). The probability p_l that x_k satisfies exactly l constraints is thus

$$\left\{ \begin{array}{rcll} p_l & = & 0 & \text{if } l < d, \\ p_l & = & \binom{d}{2d-l}(2\epsilon)^{l-d}(1-2\epsilon)^{2d-l} & \text{if } d \leq l \leq 2d\,. \end{array} \right. \tag{3.1}$$

Observe in particular that $p_d = (1 - 2\epsilon)^d$, which is large if ϵ is small. Consequently, if ϵ is small, it is most likely that x_k satisfies exactly d constraints. As an example, consider the problem with $\kappa = 3$. Then $\epsilon = 10^{-3}$ and $p_d > 0.9$ for all values of d inferior to 50.

For the YUCCA test problem, the number of possible combinations of d constraints is equal to 2^d (for all $i \in [\![1, d]\!]$, choose either c_{2i-1} or c_{2i}), which is large when d is large. As a consequence, as long as $N \ll 2^d$, then with high probability, all the points X_k, $k \in [\![1, N]\!]$, satisfy a different combination of d constraints when ϵ is small and d is large[2]. Furthermore, in that case, with high probability, any new point X_{N+1} uniformly drawn on $\mathbb{X}$ will satisfy a combination of d constraints that is different from what was previously observed, i.e. it will be non-dominated. In other words, when ϵ is small, d is large and $N \ll 2^d$, then with high probability, any new point X_{N+1} uniformly drawn on $\mathbb{X}$ makes an improvement. Assuming the models are almost perfect (so that $c_i(x) \leq 0 \Rightarrow \xi_{c,i}(x) \leq 0$), the probability of improvement is therefore close to uniform on $\mathbb{X}$.

Consider now the hypervolume improvement $I(x)$ yielded by the observation of a point $x \in \mathbb{X}$ that satisfies exactly d constraints. Assuming that $\mathbb{B}_c = [-1, 1]^{2d}$, which is a reasonable choice for the YUCCA problem, then the ratio between $I(x)$ and the volume of $\mathbb{B}_c$, i.e. the volume dominated by a feasible solution is given by

$$\frac{I(x)}{|\mathbb{B}_c|} \leq \frac{\mathbb{1}_{c(x) \in \mathbb{B}_c}}{2^{2d}} \prod_{i=1}^{2d} \left(2 \times \mathbb{1}_{c_i(x) \leq 0} + (1 - c_i(x))\mathbb{1}_{c_i(x) > 0}\right) \leq \frac{1}{2^d}\,, \tag{3.2}$$

which is small when d is large. In other words, the improvement yielded by points drawn from the probability of improvement is likely to be small when d is large and ϵ is small, and the probability of improvement is not suitable for optimizing the EI in that case.

[2]Due to the way x^{opt} is defined, not all combinations have the same probability of occurence. To simplify the analysis, we neglect this effect.

Remark 13 *Note that estimating the improvement (3.2) using a sample uniformly distributed on* $\mathbb{B}_c \setminus H_{n,c}$ *as proposed in Section 2.4.1 would require a large number of particles. In Section 3.3, a density will be proposed that can be used to compute small improvements with a reasonable number of particles.*

3.2.4 Novel sampling densities

On the YUCCA test problem, the EI criterion is likely to be high only in regions where a large number of constraints are simultaneously satisfied. Therefore we need a density that concentrates in regions where the probability of jointly satisfying the constraints is high. In this section, we consider the following three densities defined on $\mathbb{X}$:

$$\begin{cases} \text{PIEK1}(x) & \propto & \text{PI}_n^{\text{o}}(x) \cdot \mathbb{E}_n \left(K(x)! \, \mathbb{1}_{\xi_c(x) \in \mathbb{B}_c \setminus H_{n,c}} \right), \\ \text{PIEK2}(x) & \propto & \text{PI}_n^{\text{o}}(x) \cdot \mathbb{E}_n \left(K(x)! \, e^{-\tau \sum_{i=1}^{q} \xi_{c,i}(x)_+} \, \mathbb{1}_{\xi_c(x) \in \mathbb{B}_c \setminus H_{n,c}} \right), \\ \text{PICPI}(x) & \propto & \text{PI}_n^{\text{o}}(x) \cdot \prod_{i=1}^{q} \mathbb{P}_n \left(\xi_{c,i}(x) \leq r_i \right), \end{cases} \tag{3.3}$$

where $\text{PI}_n^{\text{o}}(x) = \mathbb{P}_n \left(\xi_o(x) \in \mathbb{B}_o \setminus H_{n,o} \right)$ denotes the probability of improvement with respect to the objectives, $K(x) = \# \left\{ i \in [\![1, d]\!], \xi_{c,i}(x) \leq 0 \right\}$ denotes the number of constraints satisfied by $\xi_c(x)$ as in Section 2.4.2, and $r_i = \min_{\{X_1, \ldots, X_n\}} c_i(X)_+$, $1 \leq i \leq q$.

The PIEK1 density corresponds to the density introduced in Section 2.4.2. It is a form of expected improvement where the term $K(x)!$ is used to concentrate the density in regions where more constraints are simultaneously satisfied. It achieved good results on the g18 problem (see Section 2.5.3). On the YUCCA test problem, the probability that x satisfies more than d constraints is small when ϵ is small ($1 - p_d \approx 2d\epsilon$ in that case, see (3.1)), because the regions where constraints are simultaneously satisfied are small bands of width 2ϵ. The PIEK1 density is based on a discontinuous improvement function and on this problem, it is close to uniform on $\mathbb{X}$, except in the small regions of width 2ϵ where it will have high plateaus. This is a "landscape" that is very difficult to sample using SMC, because there is no slope toward the regions of high density.

The PIEK2 density is similar to PIEK1 but it is smoother. The parameter τ in its expression allows to control the slope of the density. A sharp slope, i.e. a large τ, will concentrate the density mass in regions of low constraint violation whereas a smooth slope, i.e. a small τ, will make the concentration of the density more progressive. Sampling the PIEK2 density with SMC is therefore easier than it is with the PIEK1 density.

The PICPI density is based on the product of the probabilities of improvement on each constraint. It can be seen as a form of expected improvement where the improvement is one when we improve upon all the constraints simultaneously and zero otherwise. As such, this is a very "optimistic" density. As will be seen in Section 3.2.5, sampling the PICPI density is simpler than sampling the PIEK1 and PIEK2 densities because it is possible to define a sequence of easy-to-sample densities that converges to the PICPI density.

Prior to finding feasible solutions, $H_{n,\mathrm{o}}$ is empty and the term $\mathrm{PI}_n^{\mathrm{o}}(x)$ in the expressions of the PIEK1, PIEK2 and PICPI densities can be computed in closed form when independence between the objectives is assumed:

$$\begin{aligned}\mathrm{PI}_n^{\mathrm{o}}(x) &= \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \in \mathbb{B}_{\mathrm{o}}\right), \\ &= \prod_{i=1}^{p} \Phi\left(\frac{y_{\mathrm{o},i}^{\mathrm{upp}} - \widehat{\xi}_{\mathrm{o},i,n}(x)}{\sigma_{\mathrm{o},i,n}(x)}\right),\end{aligned}$$

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

Once feasible solutions are known, $H_{n,\mathrm{o}}$ is no longer empty and $\mathrm{PI}_n^{\mathrm{o}}(x)$ has to be estimated. The Monte-Carlo approximation technique refered to in Remark 10 of Section 2.4.2 can be used in this case. Some authors refer to this Monte-Carlo approximation technique as the Sample Average Approximation (SAA) technique (see, e.g., Svenson (2011)). In the following, we will use this terminology as well.

There exist no closed forms of the expectations in the expressions of the PIEK1 and PIEK2 densities. Again, they can be computed approximately using the SAA technique. The product term in the expression of the PICPI density on the other hand, can be computed in closed form:

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

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

Remark 14 *Observe that all the densities considered in this section default to the probability of improvement as soon as a feasible solution has been observed. Our focus here is on optimizing the EI criterion prior to finding feasible solutions. There is no indication in our experiments that the probability of improvement is not suitable for optimizing the EI once a feasible solution has been identified (see experiments of Section 3.2.6).*

3.2.5 Sampling procedure

In Section 2.4.2, the sampling procedure to construct the successive $(\mathcal{X}_n)_{n\geq 1}$ from the PI density has not been detailed because it is very similar to what is described in Algorithms 2, 3 and 4. It consists of two ingredients. First, an initialization procedure is required to produce $\mathcal{X}_0$ from π_0. Then, a specific strategy for making transitions between two successive densities π_n and π_{n+1} is necessary because it may happen that the two densities are too different to produce $\mathcal{X}_{n+1}$ from $\mathcal{X}_n$ in one step using Algorithm 5.

For initialization, we used a sample $\mathcal{X}_0$ independently and uniformly distributed on $\mathbb{X}$ in Algorithm 5. This was possible for the PI density because $\pi_0(x) \propto \mathbb{P}_n\left(\xi(x) \in \mathbb{B}\right)$ is close to one for every $x \in \mathbb{X}$ when y^{upp} is set high enough (it is equal to one in the limit $y^{\mathrm{upp}} \to +\infty$), i.e. π_0 is almost uniform. Then, to make transitions, a procedure based on intermediate fronts

of non-dominated points was proposed in Algorithms 3 and 4.

When the PIEK1, PIEK2 and PICPI densities are used, this sampling procedure has to be adapted. Indeed, π_0 is not in general uniform on $\mathbb{X}$ when the PIEK1 and PIEK2 densities are used, and transitions based on intermediate fronts of non-dominated points are not suitable for the PICPI density.

A more general initialization procedure that makes it possible to handle non-uniform initial densities π_0 is proposed in Algorithm 6. It implements a tempering procedure to construct $\mathcal{X}_0$ from π_0 starting from a sample uniformly distributed on $\mathbb{X}$.

For making transitions with the PIEK1 and PIEK2 densities, Algorithms 3 and 4 can be used as is. The structure of the PICPI density is different. It is not defined with respect to fronts of non-dominated points as the other densities. Only one reference point $r = (r_1, \ldots, r_q) \in \mathbb{R}_+^q$ is necessary to define the PICPI density (see (3.3)). A procedure to make transitions in that case is provided in Algorithm 7.

Algorithm 6: Tempering procedure to construct a sample $\mathcal{X} = \mathcal{X}_0$ approximately distributed from a target density $\pi = \pi_0$.

Notations: Let $\mathcal{X} = (x_k, w_k)_{1 \leq k \leq m}$ be a weighted sample distributed from a density π and let $w^\star = (w_k^\star)_{1 \leq k \leq m}$ be the normalized update weights for the cloud $\mathcal{X}$ and a density $\pi^\star$ (see Equation (2.33)). We shall denote $ESS(\mathcal{X}, \pi^\star)$ the effective sample size of $\mathcal{X}$ with respect to density $\pi^\star$:

$$ESS(\mathcal{X}, \pi^\star) = \left(\sum_{1 \leq k \leq m} (w_k^\star)^2 \right)^{-1}$$

Input: A target initial density π, a sample size m and a threshold value ν.

Output: A weighted sample $\mathcal{X}$ of size m drawn from π.

```
α_0 ← 0
Draw X_{α_0} of size m uniformly distributed on 𝕏.
while α_0 < 1 do
    if ESS(X_{α_0}, π) ≥ νm then
        α_0 ← 1
    else
        Find α such that ESS(X_{α_0}, π^α) ≈ νm using a dichotomy on α ∈ [α_0, 1].
        α_0 ← α
    Draw a sample X_{α_0} distributed from π^{α_0} using steps 4 → 6 of
      Algorithm 5 (Reweight-Resample-Move steps).
X ← X_{α_0}
```

3.2.6 Numerical experiments

We make two successive benchmarks. First, the performances of BMOO using the PI, PIEK1, PIEK2 and PICPI densities are compared on the YUCCA test problem for different values of d

Algorithm 7: Procedure to construct $\mathcal{X}_n$, $n \geq 1$ when the PICPI density is used.

Notations: Given $x \in \mathbb{X}$ and $r = (r_1, \ldots, r_q) \in \mathbb{R}_+^q$, denote by $\pi_r(x)$ the PICPI density at location x for a reference r:

$$\pi_r(x) \propto \mathrm{PI}_n^{\mathrm{o}}(x) \cdot \prod_{i=1}^{q} \mathbb{P}_n \left(\xi_{\mathrm{c},i}(x) \leq r_i \right).$$

Inputs: A sample size m, a threshold value ν, and $\mathcal{X}_0$, r_0 and $r^\star$ such that

- $r_0 = (r_{0,1}, \ldots, r_{0,q})$ and $r^\star = (r_1^\star, \ldots, r_q^\star)$ with $r_i^\star \leq r_{0,i}$, for all $i \in [\![1, q]\!]$.
- $\mathcal{X}_0 = (x_{0,k})_{1 \leq k \leq m} \in \mathbb{X}^m$ is distributed from π_{r_0}. Note that $\mathcal{X}_0$ may contain replicated values.

Output: A sample $\mathcal{X}^\star$ distributed from $\pi_{r^\star}$.

```
t ← 0
while t < 1 do
    if ESS(X_t, π_{r*}) ≥ νm then
        t ← 1
    else
        Find u ∈ [t, 1] such that ESS(X_t, π_{r_u}) ≈ νm using dichotomy, where
          r_u = r_t + u(r* − r_t).
        t ← u
    Draw a sample X_t distributed from π_{r_t} using steps 4 → 6 of
      Algorithm 5 (Reweight-Resample-Move steps).
X* ← X_t
```

and κ. The results of this first benchmark are presented in Tables 3.1, 3.2 and 3.3. The PICPI density is shown to outperform the other three densities on this problem. To see if its good results generalize to other problems, we redo the experiments of Section 2.5.3 with BMOO using the PICPI density. The results are compared with the results obtained with the PI density in Section 2.5.3, to the results of Regis (2014) and to results obtained with BMOO using the expected improvement as a sampling density for optimizing the criterion once a feasible solution has been found.

In Table 3.1, we set $\kappa = 1$ and d varies from $d = 2$ to $d = 100$. In tables 3.2 and 3.3, κ is set successively to $\kappa = 3$ and $\kappa = 5$ and d varies from $d = 2$ to $d = 50$. We do not run the algorithm with $d = 100$ for $\kappa > 1$ because then the ratio r between the volume of the feasible region and that of the design space is too small (Note that when $\kappa = 5$ and $d = 50$, $r = 10^{-250}$ already). The algorithm is run with the settings described in Section 2.5.1, except for the parameter ν which is set to $\nu = 0.05$. When the dimension of the problem augments, the ratio between the volume of the feasible region and the volume of $\mathbb{X}$ decreases very fast and the SMC algorithm has to make many transitions between two iterations of the algorithm. Taking $\nu = 0.05$ instead of $\nu = 0.2$ permits to speed up the algorithm by decreasing the number of SMC transitions.

In our experiments, we stop the algorithm as soon as a feasible point is found and measure the number of samples that were required. The runs are repeated 30 times each with different initial design of experiments and with a limiting number of affordable functions evaluation of $N_{max} = 200$. For reference, results obtained using the four local optimization algorithms of Section 2.5.3 are shown in the tables. These are run 30 times each with different starting points and with a limiting number of affordable functions evaluation of $N_{max} = 1000$. Only the results of the best scoring algorithm are shown in the tables; see Section 3.6.1 for more detailed results.

For each density, the first column corresponds to the number of successful runs (a run is considered successful if a feasible solution has been found). A value of 30 thus indicates that all runs were successful in finding a feasible solution. The second column contains the average number of function evaluations that were required to find a feasible solution, where the average is taken over successful runs only. The corresponding standard deviation is given in parenthesis. For the local optimization algorithms, the name of the best scoring algorithm is given in the first column (Cob stands for the COBYLA algorithm, IP stands for the interior-point algorithm, AS stands for the active-set algorithm and SQP stands for the SQP algorithm). The informations in the other two columns are the same as for the densities[3].

The PICPI density clearly outperforms the other densities on the YUCCA problem. The algorithm is able to solve the problem very efficiently up to $\kappa = 5$ and $d = 50$ when it is used. When the other densities are used, the algorithm stalls for $\kappa > 1$ (see Tables 3.2 and 3.3). Looking at the results of Table 3.1 where $\kappa = 1$, we observe a clear hierarchy between the four densities. The PI density works up to $d = 5$, the PIEK1 density works up to $d = 10$ and the PIEK2 density works up to $d = 20$. The PICPI density on the other hand works up to $d = 100$. The introduction of a slope in the PIEK2 density seems to help but it is not sufficient to achieve

[3] Some of the local optimization algorithms used in this study allow some tolerance on the constraints violation. In the tables, a solution is considered as feasible when there is no constraint violation larger than 10^{-5}. The parameters of the local optimization algorithms are set accordingly.

d	N_{init}	Local (best among 4)			PI		PIEK1		PIEK2		PICPI	
2	10	SQP	30	**7.2** (1.6)	30	10.7 (1.3)	30	10.6 (1.4)	30	10.3 (1.9)	30	10.9 (0.4)
5	20	SQP	28	**16.2** (3.0)	30	26.8 (2.7)	30	21.2 (0.7)	30	21.0 (0.0)	30	21.0 (0.0)
10	30	SQP	25	**31.4** (5.8)	0	- (-)	30	58.0 (17.9)	30	31.5 (0.9)	30	**31.0** (0.0)
20	40	SQP	16	70.6 (12.6)	0	- (-)	2	134.5 (79.9)	30	81.2 (18.1)	30	**41.0** (0.0)
50	50	SQP	11	191.1 (23.8)	0	- (-)	0	- (-)	0	- (-)	30	**51.0** (0.0)
100	60	SQP	1	405.0 (0.0)	0	- (-)	0	- (-)	0	- (-)	30	**65.4** (2.4)

Table 3.1: Results obtained by the local optimization algorithms and by BMOO using the PI, PIEK1, PIEK2 and PICPI densities on the YUCCA test problem with $\kappa = 1$. 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. Dash symbols are used when a value can not be calculated.

d	N_{init}	Local (best among 4)			PI		PIEK1		PIEK2		PICPI	
2	10	SQP	27	**9.4** (2.4)	30	16.8 (1.3)	30	15.4 (1.5)	30	13.8 (1.0)	30	**11.0** (0.0)
5	20	SQP	24	**21.8** (3.1)	0	- (-)	0	- (-)	0	- (-)	30	**21.0** (0.0)
10	30	SQP	19	40.9 (5.5)	0	- (-)	0	- (-)	0	- (-)	30	**31.4** (0.5)
20	40	SQP	17	85.0 (0.0)	0	- (-)	0	- (-)	0	- (-)	30	**42.0** (0.0)
50	50	SQP	3	205.0 (0.0)	0	- (-)	0	- (-)	0	- (-)	30	**56.2** (7.6)

Table 3.2: Results obtained by the local optimization algorithms and by BMOO using the PI, PIEK1, PIEK2 and PICPI densities on the YUCCA test problem with $\kappa = 3$. See Table 3.1 for conventions.

d	N_{init}	Local (best among 4)			PI		PIEK1		PIEK2		PICPI	
2	10	SQP	27	**10.1** (1.6)	30	39.8 (4.5)	30	35.1 (4.3)	30	30.6 (3.2)	30	**12.0** (0.0)
5	20	SQP	21	**21.6** (3.6)	0	- (-)	0	- (-)	0	- (-)	30	**22.4** (0.5)
10	30	SQP	20	42.2 (4.9)	0	- (-)	0	- (-)	0	- (-)	30	**32.9** (0.3)
20	40	SQP	10	80.8 (8.9)	0	- (-)	0	- (-)	0	- (-)	30	**43.0** (0.0)
50	50	SQP	3	205.0 (0.0)	0	- (-)	0	- (-)	0	- (-)	10	65.4 (37.8)

Table 3.3: Results obtained by the local optimization algorithms and by BMOO using the PI, PIEK1, PIEK2 and PICPI densities on the YUCCA test problem with $\kappa = 5$. See Table 3.1 for conventions.

good optimization performances[4].

Among the four local optimization algorithms, the SQP algorithm achieves the best results on the YUCCA test problem for all values of κ and d considered in this study. It achieves good results up to $d = 5$ but is outperformed by BMOO using the PICPI density for higher values of d. In particular, its performances strongly depend on the starting point, with many unsuccessful runs.

To see if the goods results obtained with BMOO using the PICPI density generalize to other problems, we redo the experiments of Section 2.5.3. The results are given in Tables 3.4 and 3.5. In the tables, BMOO using the PICPI density is compared to the results presented in Section 2.5.3 using the PI density, to the results of Regis (2014) and to results obtained by BMOO using the following density:

$$\text{EICPI}(x) \propto \mathbb{E}_n\left((y_n - \xi_{\text{o}}(x))_+\right) \prod_{i=1}^{q} \mathbb{P}_n\left(\xi_{\text{c},i}(x) \leq r_i\right), \tag{3.4}$$

where $y_n = \min\left(\{\xi_{\text{o}}(X_i) \text{ s.t. } \xi_{\text{c}}(X_i) \leq 0, 1 \leq i \leq n\} \cup \{y_{\text{o}}^{\text{upp}}\}\right)$ is either the upper corner of $\mathbb{B}_{\text{o}}$ when no feasible solution is knwon or the current best feasible objective value and, as for the PICPI denisty, $r_i = \min_{\{X_1,\ldots,X_n\}} c_i(X)_+$, $1 \leq i \leq q$.

The EICPI density is similar to the PICPI density except that it is proportional to the expected improvement once a feasible solution is known. Indeed, on single-objective problems, once a feasible solution is known, the EI criterion can be computed in closed form with a low computational cost. Therefore, one may wonder if using the expected improvement as a sampling density for optimizing the EI criterion would yield better results than the probability of improvement.

Remark 15 *In the unconstrained single-objective setting, Benassi (2013) observed in Section 4.3.1 of his thesis that the probability of improvement favours regions close to the current minimum, whereas the expected improvement tends to be more explorative and, hence, more "unstable"*[5]*. In a constrained setting, this argument may not hold anymore. Indeed, when the feasible region is small, the EI is likely to favour local (feasible) solutions as well. In a multi-objective setting, the question arises anew because the concepts of exploitation and exploration take a different form (see, e.g., Bosman and Thierens (2003)). We do not, in this thesis, advance any argument other than implementation complexity (see Section 3.3) to motivate our choice of using the probability of improvement instead of the expected improvement for optimizing the EI on multi-objective problems.*

Several observations can be made based on the results of this experiment. First, it can be seen that the good performances of the PICPI density generalize to the problems of this benchmark. On the problems where we had good results with the PI density, the PICPI density does just as

[4] In our experiments, the parameter τ that controls the slope of the PIEK2 density is set to $\tau = 5$.

[5] The "unstability" of the sampling density is not necessarily an issue for BMOO because it has a restart capability (see Remark 11 in Section 2.4.2). However, the accumulation of restarts increases the computing time of the algorithm.

Pbm	N_{init}	PICPI		EICPI		PI		Regis (best among 3)		
g1	39	30	40.4 (0.5)	30	40.5 (0.5)	30	44.2 (1.9)	CG	30	**15.0** (0)
g5mod	12	30	13.0 (0.0)	30	13.0 (0.3)	30	13.0 (1.2)	CL	30	**6.4** (0.1)
g6	6	30	**9.4** (0.7)	30	**9.2** (0.7)	30	**9.7** (0.7)	CL	30	**10.9** (0.3)
g7	30	30	**31.0** (0.0)	30	**31.0** (0.0)	30	38.8 (3.3)	CG	30	47.5 (4.6)
g8	6	30	**7.1** (0.4)	30	**6.7** (1.6)	30	**7.0** (0.2)	CL	30	**6.5** (0.2)
g9	21	30	**23.1** (1.7)	30	**22.9** (4.3)	30	**21.8** (5.1)	CG	30	**21.5** (1.9)
g13mod	15	30	11.5 (5.4)	30	12.0 (5.6)	30	10.5 (5.6)	Ext	30	**8.6** (0.7)
g16	15	30	**20.8** (4.3)	30	**21.8** (5.3)	30	**21.7** (7.3)	Ext	30	**19.6** (1.8)
g18	27	30	**32.0** (1.7)	30	**32.7** (1.7)	0	- (-)	CL	30	108.6 (6.5) (-)
g19	45	30	46.0 (0.0)	30	46.0 (0.0)	30	46.4 (3.0)	CL	30	**16.5** (0.5)
g24	6	30	2.2 (1.5)	30	2.1 (1.3)	30	2.6 (1.6)	CG	30	**1.3** (0.1)
SR7	21	30	21.9 (0.4)	30	21.2 (3.4)	30	22.0 (0.2)	CG	30	**9.5** (0.1)
WB4	12	30	**17.5** (4.9)	30	**18.2** (7.0)	30	**19.1** (5.8)	CL	30	37.4 (5.9)

Table 3.4: Number of evaluations to find a first feasible point. See Table 3.1 for conventions.

Pbm	N_{init}	PICPI		EICPI		PI		Regis (best among 3)		
g1	39	30	**55.1** (2.0)	30	**54.8** (2.0)	30	**57.7** (2.6)	CG	30	125.2 (15.3)
g5mod	12	30	**15.5** (0.5)	30	**15.2** (0.4)	30	16.3 (0.6)	CL	30	**12.9** (0.5)
g6	6	30	**13.5** (0.7)	30	**13.2** (0.8)	30	**13.3** (0.8)	CL	30	53.6 (14.0)
g7	30	30	**50.4** (2.0)	30	**50.7** (1.6)	30	**55.8** (2.8)	CG	30	99.8 (5.7)
g8	6	30	**27.3** (7.8)	30	**26.1** (7.3)	30	**26.3** (10.4)	CL	30	**30.3** (2.8)
g9	21	30	**59.5** (11.9)	30	**63.4** (16.3)	30	**61.6** (14.3)	CG	30	176.4 (26.3)
g13mod	15	26	**133.9** (38.8)	24	**128.7** (41.4)	30	180.3 (84.6)	Ext	30	**146.4** (29.2)
g16	15	30	**32.1** (9.9)	30	**33.3** (9.0)	30	**30.3** (12.3)	Ext	30	38.4 (3.6)
g18	27	30	**58.3** (4.9)	30	**59.3** (3.9)	0	- (-)	CL	24	195.9 (-)
g19	45	30	**131.5** (5.4)	30	**132.4** (5.4)	30	**133.3** (6.2)	CL	30	698.5 (75.3)
g24	6	30	**9.4** (1.5)	30	**9.8** (1.1)	30	**9.9** (1.0)	CG	30	**9.0** (0)
SR7	21	30	**28.4** (0.8)	30	**27.8** (0.6)	30	**29.3** (0.7)	CG	30	**33.5** (1.6)
WB4	12	30	**51.4** (24.9)	30	**54.5** (17.5)	30	**44.5** (13.3)	CL	30	164.6 (12.2)

Table 3.5: Number of evaluations to reach specified target (see Table 2.1). See Table 3.1 for conventions.

well and on the g18 problem, it does much better. In particular, BMOO using the PICPI density obtains results consistently better or as good as the best results of Regis (2014) (see Remark 16). Note that since the PICPI density can be computed in closed form prior to finding feasible solutions, we also have a gain in terms of computing time. Regarding the comparison between the PICPI and the EICPI densities, we do not observe any noticeable difference in performances on the problems of this benchmark.

Remark 16 *We do not consider the g3mod, g10 and PVD4 test problems in this benchmark. Recall from Section 2.5.3 that BMOO failed on these problems because of modeling issues. Since nothing was done to improve the algorithm on this particular aspect, it cannot be expected to perform any better.*

Remark 17 *Note the mitigated results obtained on the g13mod problem. The algorithm either solves the problem in about 130 evaluations or stalls. This happens because the target is set very close to the true optimum for this problem (see Table 2.1). A more in depth study of the behaviour of the algorithm close to convergence would be required to better understand this phenomenon.*

3.2.7 Conclusions

The main result of this section is the proposal of a novel sampling density for optimizing the EI criterion on problems with many constraints. This new density, which we call the PICPI density, makes it possible to improve the performances of the BMOO algorithm on the problems of Section 2.5.3. As an additional benefit, it can be computed in closed form prior to finding feasible solutions and a simplified sampling procedure can be used.

We believe that there is room for further improvement of the proposed strategy for optimizing the EI criterion. In this thesis work, we use a simple algorithm to perform the move step of Algorithm 5 (see Section 2.5.1). While it is sufficient in most situations, it is not computationally efficient and better performances could probably be achieved using a more suitable algorithm.

Additionally, in this section, we discuss the pertinence of using a sampling density proportional to the probability of improvement instead of the expected improvement for optimizing the EI criterion in the single-objective case. No evidence emerges and a more in-depth comparison between the two strategies would be required to better understand the consequences of this choice.

3.3 Efficient computation of the EHVI criterion

3.3.1 Introduction

In this section, we study in more details the computation method of the EI criterion presented in Section 2.4.1.

As briefly discussed in Section 2.3.3, the computational effort can be reduced if independence between the objectives and constraints is assumed. In that case, the EI criterion (2.23) can be

decomposed as a sum of two terms by splitting the integration domain in two parts:

$$\rho_n(x) = \rho_n^{\text{feas}}(x) + \rho_n^{\text{unf}}(x), \text{ for any } x \in \mathbb{X},$$

where the term ρ_n^{feas} corresponds to the integral on $G_n \cap \{y_{\text{c}} \leq 0\}$, and the term ρ_n^{unf} corresponds to the integral on $G_n \cap \{y_{\text{c}} \not\leq 0\}$, where $G_n = \mathbb{B} \setminus H_n$ denotes the set of all non-dominated points in $\mathbb{B}$ as in Section 2.3.3. In other words, the term ρ_n^{feas} corresponds to the contribution on the feasible region of $\mathbb{B} = \mathbb{B}_{\text{o}} \times \mathbb{B}_{\text{c}}$ and the term ρ_n^{unf} corresponds to the contribution on the unfeasible region of $\mathbb{B}$:

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

and

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

Using this decomposition, two successive phases of the optimization process can be distinguished. Prior to finding a feasible solution, the set $H_{n,\text{o}}$ is empty and the terms $\rho_n^{\text{feas}}(x)$ and $\rho_n^{\text{unf}}(x)$ can be rewritten in the following form:

$$\left\{\begin{array}{rcl}
\rho_n^{\text{feas}}(x) &=& \left|\mathbb{B}_{\text{c}}^-\right| \cdot \mathbb{P}_n(\xi_{\text{c}}(x) \leq 0) \cdot \displaystyle\int_{\mathbb{B}_{\text{o}}} \mathbb{P}_n\left(\xi_{\text{o}}(x) \prec y_{\text{o}}\right)\, \mathrm{d}y_{\text{o}}\,, \\
\rho_n^{\text{unf}}(x) &=& |\mathbb{B}_{\text{o}}| \cdot \displaystyle\int_{\mathbb{B}_{\text{c}} \setminus H_{n,\text{c}}} \mathbb{P}_n\left(\xi_{\text{c}}^+(x) \prec y_{\text{c}}^+\right)\, \mathbb{1}_{y_{\text{c}} \not\leq 0}\, \mathrm{d}y_{\text{c}}.
\end{array}\right. \tag{3.5}$$

The term $\rho_n^{\text{feas}}(x)$ can be computed in closed form for all $x \in \mathbb{X}$ when independence between the functions of the problem is assumed. It is the product of the volume of $\mathbb{B}_{\text{c}}^-$, which is known, with the probability of feasibility and with an integral over $\mathbb{B}_{\text{o}}$. The probability of feasibility can be computed in closed form when independence between the constraints is assumed (see Remark 5) and the integral over $\mathbb{B}_{\text{o}}$ can be computed using the following formula when

independence between the objectives is assumed:

$$\begin{aligned}
\int_{\mathbb{B}_o} \mathbb{P}_n\left(\xi_o(x) \prec y_o\right) \mathrm{d}y_o &= \prod_{i=1}^{p} \int_{y_{o,i,n}^{low}}^{y_{o,i,n}^{upp}} \mathbb{P}_n\left(\xi_{o,i}(x) \leq y\right) \mathrm{d}y \\
&= \prod_{i=1}^{p} \mathbb{E}_n\left(\int_{y_{o,i,n}^{low}}^{y_{o,i,n}^{upp}} \mathbb{1}_{(\xi_{o,i}(x) \leq y)} \, \mathrm{d}y\right) \\
&= \prod_{i=1}^{p} \mathbb{E}_n\left(\left(y_{o,i,n}^{upp} - \max\left(\xi_{o,i}(x), y_{o,i,n}^{low}\right)\right)_+\right) \\
&= \prod_{i=1}^{p} \left[\mathbb{E}_n\left(\left(y_{o,i,n}^{upp} - \xi_{o,i}(x)\right)_+\right)\right. \\
&\qquad \left. -\mathbb{E}_n\left(\left(y_{o,i,n}^{low} - \xi_{o,i}(x)\right)_+\right)\right]
\end{aligned} \tag{3.6}$$

Remark 18 *Notice the terms corresponding to the expected improvement with respect to the lower bounding values $y_{o,i,n}^{low}$ in (3.6). These are a consequence of introducing lower bounding values in the definition of the EI. They are necessary in the first phase of the optimization process because the volume of $\mathbb{B}_o$ appears in the expression of ρ_n^{unf}.*

The term $\rho_n^{\mathrm{unf}}(x)$ on the other hand has to approximated because there exists no exact computation method for this term. Developing a domain decomposition method in the spirit of what is done for the exact computation of the EHVI criterion would be possible. However, in most applications, the number of constraints is likely to be high and such a method would probably be too costly to be practical.

Remark 19 *Note that in the beginning of the optimization process, the term $\mathbb{P}_n(\xi_c(x) \leq 0)$ in the expression of $\rho_n^{feas}(x)$ is likely to be small when the problem is highly constrained. The optimization process is then mostly driven by $\rho_n^{unf}(x)$, which depends only on the constraints. Once progress has been made on the resolution of the constraints and we are close to finding a feasible solution, the term $\rho_n^{feas}(x)$ then permits to discriminate between potentially feasible solutions (i.e. solutions for which $\mathbb{P}_n(\xi_c(x) \leq 0)$ is close to 1), by considering the improvement with respect to the upper corner of $\mathbb{B}_o$, thus rewarding more generously candidates with good objectives values. This observation holds for reasonable $\mathbb{B}_o$ and $\mathbb{B}_c$, the choice of which can influence the optimization process though.*

Once a feasible solution has been observed, the second phase of the optimization process corresponds to minimizing the objectives. In that situation, the set $\mathbb{B}_c \setminus H_{n,c}$ is empty and the terms $\rho_n^{\mathrm{feas}}(x)$ and $\rho_n^{\mathrm{unf}}(x)$ take the following form:

$$\begin{cases}
\rho_n^{\mathrm{feas}}(x) &= \left|\mathbb{B}_c^-\right| \cdot \mathbb{P}_n(\xi_c(x) \leq 0) \cdot \displaystyle\int_{\mathbb{B}_o \setminus H_{n,o}} \mathbb{P}_n\left(\xi_o(x) \prec y_o\right) \mathrm{d}y_o \,, \\
\rho_n^{\mathrm{unf}}(x) &= 0 \,.
\end{cases} \tag{3.7}$$

The term $\rho_n^{\mathrm{unf}}(x)$ vanishes and the expected improvement is equal to the product of a constant with the probability of satisfying the constraints and with a modified EHVI criterion defined using

Phase I	$\rho_n^{\text{feas}} = \left\lvert\mathbb{B}_{\text{c}}^-\right\rvert \, \mathbb{P}_n(\xi_{\text{c}}(x) \leq 0) \int_{\mathbb{B}_{\text{o}}} \mathbb{P}_n\left(\xi_{\text{o}}(x) \prec y_{\text{o}}\right) \mathrm{d}y_{\text{o}}$	Exact
	$\rho_n^{\text{unf}} = \lvert\mathbb{B}_{\text{o}}\rvert \int_{\mathbb{B}_{\text{c}} \setminus H_{n,\text{c}}} \mathbb{P}_n\left(\xi_{\text{c}}^+(x) \prec y_{\text{c}}^+\right) \mathbb{1}_{y_{\text{c}} \not\preceq 0} \, \mathrm{d}y_{\text{c}}$	Approx.
Phase II	$\rho_n^{\text{feas}}(x) = \left\lvert\mathbb{B}_{\text{c}}^-\right\rvert \, \mathbb{P}_n(\xi_{\text{c}}(x) \leq 0) \int_{\mathbb{B}_{\text{o}} \setminus H_{n,\text{o}}} \mathbb{P}_n\left(\xi_{\text{o}}(x) \prec y_{\text{o}}\right) \mathrm{d}y_{\text{o}}$	Exact or approx.
	$\rho_n^{\text{unf}}(x) = 0$	Exact

Table 3.6: Formulas used in the computation of $\rho_n^{\text{feas}}(x)$, $x \in \mathbb{X}$. Phase I corresponds to a situation where no feasible solution is known. Phase II corresponds to a situation where a feasible solution is known. During Phase II, the term $\rho_n^{\text{feas}}(x)$ can be computed exactly using the procedure described in Section 3.3.3 or approximated using the proposed SMC procedure.

feasible values of the objectives only. Note that the constant term $\left\lvert\mathbb{B}_{\text{c}}^-\right\rvert$ does not influence the location of the maximum of the EI. Therefore it doesn't affect the optimization process.

The computation of the integral in the expression of $\rho_n^{\text{feas}}(x)$ can be done exactly using an EHVI computation method (see Section 2.4.1), or approximately by using either the SMC technique proposed in Section 2.4.1 or the SAA technique (see Section 3.2.4). See Table 3.6 for a summary of the above mentioned results.

Remark 20 *Note that different techniques can be used for the computation of the integral in ρ_n^{feas} once a feasible observation has been made and for the computation of the integral in ρ_n^{unf} prior to observing a feasible solution. The first can be done exactly and the second can be approximated. Besides, one can switch from exact computation to approximate computation or the opposite depending, for example, on the number of non-dominated solutions and the dimension of the problem.*

In this section, we address questions related to the computation of the new EI criterion. First, we introduce in Section 3.3.2 a novel sampling density specifically designed for the approximate computation of the EI criterion for a set of candidate solutions. The new density, which we call the L_2^{opt} density, does not suffer from the limitations of the uniform density observed in Section 2.5.4 on the OSY test problem. Then, we discuss in Section 3.3.3 the computational complexity of the exact method and compare it with the complexity of the approximate method. It is shown that for problems with more than five objectives, it rapidly becomes impractical to use the exact computation method when the number of non-dominated observations augments. Finally, we discuss the choice of the number of particles used in the approximation method in Section 3.3.4. A simple control strategy based on the variance of estimation is proposed and illustrated on a toy example.

3.3.2 The L_2^{opt} density

Let $\mathcal{X}_n = (x_{n,k})_{1 \leq k \leq m_{\mathbb{X}}} \in \mathbb{X}^{m_{\mathbb{X}}}$ be a set of $m_{\mathbb{X}}$ particles for which we want to compute the value of the EI criterion. As recalled in the introduction, the problem reduces to the computation of

integrals of the form

$$I_{n,k} = \int_{G_n} \mathbb{P}_n \left(\xi(x_{n,k}) \prec y\right) \mathrm{d}y \,, 1 \leq k \leq m_{\mathbb{X}}, \tag{3.8}$$

where $\xi = \xi_{\mathrm{o}}$ or ξ_{c}^{+} and $G_n = \mathbb{B}_{\mathrm{o}} \backslash H_{n,\mathrm{o}}$ or $\mathbb{B}_{\mathrm{c}} \backslash H_{n,\mathrm{c}}$, depending on which phase of the optimization process is considered. In this work, we consider approximations of the integral in (3.8) of the form

$$\widehat{I}_{n,k}^{\pi_n} = \frac{1}{m_{\mathbb{Y}}} \sum_{i=1}^{m_{\mathbb{Y}}} \frac{\mathbb{P}_n \left(\xi(x_{n,k}) \prec y_{n,i}\right)}{\pi_n(y_{n,i})} \,, 1 \leq k \leq m_{\mathbb{X}}, \tag{3.9}$$

where $\mathcal{Y}_n = (y_{n,i})_{1 \leq i \leq m_{\mathbb{Y}}} \in G_n^{m_{\mathbb{Y}}}$ is a set of $m_{\mathbb{Y}}$ particles distributed from a density π_n on G_n. Under the assumption that the $(y_{n,i})_{1 \leq i \leq m_{\mathbb{Y}}}$ are identically and independently distributed from π_n, the importance sampling estimator (3.9) of (3.8) is unbiased[6]. However, the quality of the approximation depends on the choice of the sampling density π_n.

In Section 2.4.1, a uniform density was used for simplicity but limitations were identified. Indeed, it may happen that a large proportion of the particles $(y_{n,i})_{1 \leq i \leq m_{\mathbb{Y}}}$ have a very small probability of being dominated by the $(\xi(x_{n,k}))_{1 \leq k \leq m_{\mathbb{X}}}$ when the uniform density is used (see the results on the OSY problem in Figure (2.9)). This happens, for example, close to convergence when the improvements that can be expected from observations at the $(x_{n,k})_{1 \leq k \leq m_{\mathbb{X}}}$ are small, or when the lower bounding values y^{low} are set too low. These two situations are illustrated in Figure 3.2.

In this section, we introduce a new sampling density that is more concentrated in the regions of importance for computing the integral (3.8). First, consider a single candidate solution $x \in \mathbb{X}$ for which we want to compute an estimator $\widehat{I}_n^{\pi_n}(x)$ of the form (3.9) of the integral $I_n(x) = \int_{G_n} \mathbb{P}_n \left(\xi(x) \prec y\right) \mathrm{d}y$ using the density π_n. The variance of the estimator $\widehat{I}_n^{\pi_n}(x)$, when the $(y_{n,i})_{1 \leq i \leq m_{\mathbb{Y}}}$ are identically and independently distributed from π_n, is

$$\mathbb{E}\left(\left(\widehat{I}_n^{\pi_n}(x) - I_n(x)\right)^2\right) = \frac{1}{m_{\mathbb{Y}}} \left(\int_{G_n} \frac{\mathbb{P}_n \left(\xi(x) \prec y\right)^2}{\pi_n(y)^2} \pi_n(y) \,\mathrm{d}y - I_n(x)^2\right). \tag{3.10}$$

Then, an optimal choice $\pi_{n,x}^{\star}$ for π_n, i.e. a choice that minimizes the variance of estimation (3.10), is

$$\pi_{n,x}^{\star} \propto \mathbb{P}_n \left(\xi(x) \prec y\right), \tag{3.11}$$

which is a well-known result stemming from the Cauchy-Schwarz inequality.

Consider now the simultaneous estimation of all $I_{n,k}$, $1 \leq k \leq m_{\mathbb{X}}$, using a common set of particles $(y_{n,i})_{1 \leq i \leq m_{\mathbb{Y}}}$. Naturally, the above mentioned result can not be used in this case because the regions of high density differ from one $x_{n,k}$ to the other. As an alternative, we propose to

[6] In practice, $\mathcal{Y}_n$ is obtained using SMC and the $(y_{n,i})_{1 \leq i \leq m_{\mathbb{Y}}}$ are not identically and independently distributed from π_n.

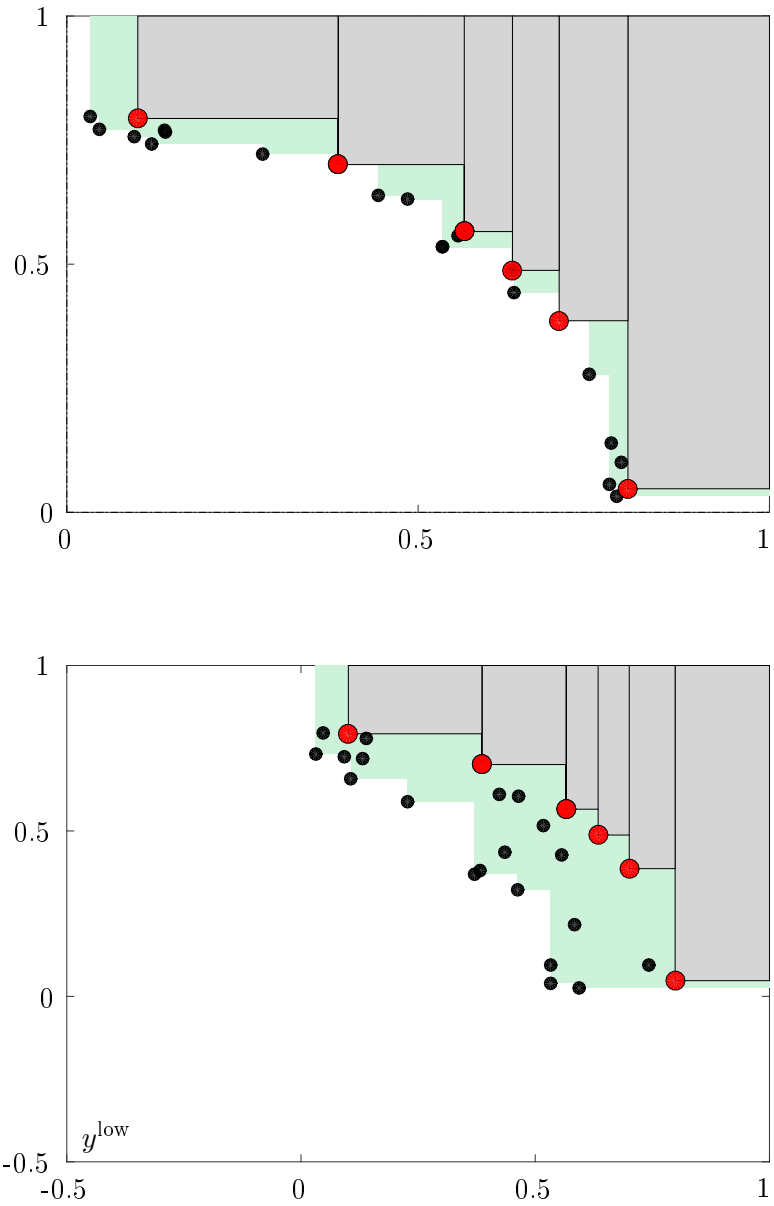


Figure 3.2: Examples of situations where the uniform density is likely to fail to produce good estimates of the EI criterion. In the figures, the non-dominated observations are represented as red disks and the dominated set H_n is represented as gray rectangles. The green area corresponds to the region dominated by the $(\widehat{\xi}(x_{n,k}))_{1\leq k\leq m_{\mathbb{X}}}$, which are shown as black dots. In the top figure, only small improvements can be expected from the $(\widehat{\xi}(x_{n,k}))_{1\leq k\leq m_{\mathbb{X}}}$ and in the bottom figure, $y^{\text{low}} = (-0.5, -0.5)$ is set too low. For the approximation (3.9) to be of good quality, it is necessary that a significant population of the particles $(y_{n,i})_{1\leq i\leq m_{\mathbb{Y}}}$ be in the green area.

minimize the sum of squared approximation errors:

$$\begin{aligned}
&\mathbb{E}\left(\left\|\widehat{I}^{\pi}_{n,k} - I_{n,k}\right\|_2^2\right) \\
&= \mathbb{E}\left(\sum_{k=1}^{m_{\mathbb{X}}}\left(\widehat{I}^{\pi}_{n,k} - I_{n,k}\right)^2\right), \\
&= \frac{1}{m_{\mathbb{Y}}}\sum_{k=1}^{m_{\mathbb{X}}}\left(\int_{G_n}\frac{\mathbb{P}_n\left(\xi(x_{n,k}) \prec y\right)^2}{\pi(y)^2}\pi(y)\,\mathrm{d}y - I_{n,k}^2\right), \\
&= \frac{1}{m_{\mathbb{Y}}}\left(\int_{G_n}\frac{\sum_{k=1}^{m_{\mathbb{X}}}\mathbb{P}_n\left(\xi(x_{n,k}) \prec y\right)^2}{\pi(y)^2}\pi(y)\,\mathrm{d}y - \sum_{k=1}^{m_{\mathbb{X}}} I_{n,k}^2\right),
\end{aligned} \tag{3.12}$$

which leads, using the same argument as above, to the definition of the following optimal density on $\mathbb{Y}$:

$$L_2^{\mathrm{opt}}(y) \propto \sqrt{\sum_{k=1}^{m_{\mathbb{X}}}\mathbb{P}_n\left(\xi(x_{n,k}) \prec y\right)^2 \cdot \mathbb{1}_G}\,. \tag{3.13}$$

The L_2^{opt} density concentrates in regions likely to be dominated by the $(\xi(x_{n,k}))_{1\leq k\leq m_{\mathbb{X}}}$, thus alleviating the issue raised in Section 2.5.4 and recalled earlier in this section. The use of this new density on the problem introduced in Section 2.5.2 is illustrated in Figure 3.3. Observe in particular the concentration of the density when the number of evaluations increases, and its independence to the lower bounding values y^{low}.

Remark 21 *The L_2^{opt} density is not in general uniform when $G = \mathbb{B}_{\mathrm{o}}$ or $\mathbb{B}_{\mathrm{c}}$ because of its dependence on the $(x_{k,n})_{1\leq k\leq m_{\mathbb{X}}}$. In the sampling procedure, an initialization step similar to that described in Algorithm 6 for the optimization of the criterion is thus required. Then, to make transitions, Algorithms 2, 3 and 4 can be adapted to non-uniform densities using the effective sample size as it is done for the criterion optimization.*

The right-hand side of (3.13) can be computed in closed form for a given set $\mathcal{X}_n$ when independence between the components of ξ is assumed. However, in practice, the computation of the L_2^{opt} density is not cheap enough to be used in an SMC procedure, because it requires a large number of evaluations of the normal cumulative distribution function. To alleviate this issue, we propose to use the following approximation of the normal cumulative distribution function in the definition of the L_2^{opt} density:

$$\Phi(y) \approx \begin{cases}
0 & \text{if } y \leq -2.6 \\
0.01 & \text{if } -2.6 < y < -2.2 \\
0.5 - 0.1y(4.4 - y) & \text{if } -2.2 \leq y \leq 0 \\
0.5 + 0.1y(4.4 - y) & \text{if } 0 \leq y \leq 2.2 \\
0.99 & \text{if } 2.2 < y < 2.6 \\
1 & \text{otherwise}
\end{cases} \tag{3.14}$$

This approximation makes it possible to lower the computational cost of the procedure without affecting too much its performances (see Figure 3.4).

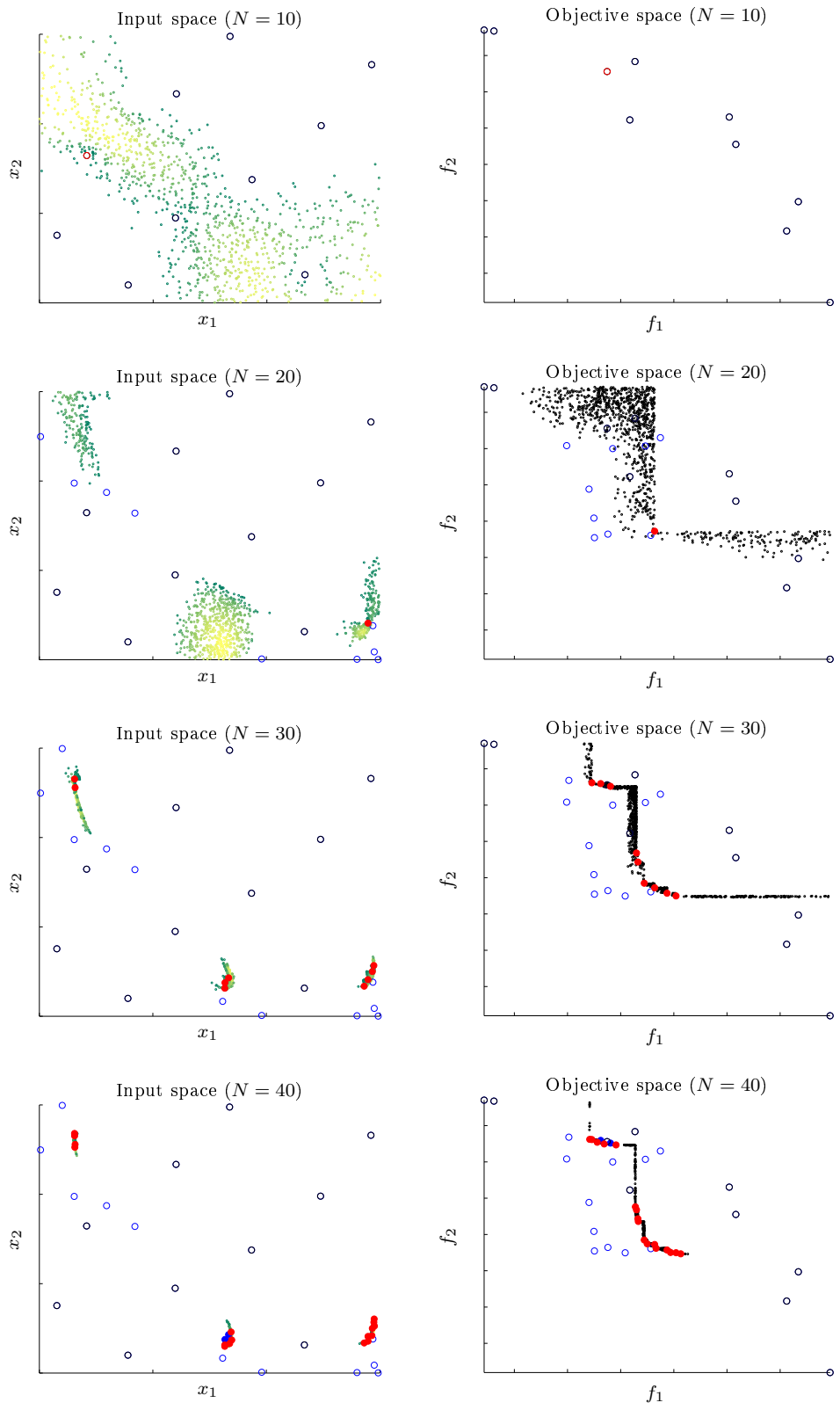


Figure 3.3: Illustration of the use of the L_2^{opt} density on the problem introduced in Section 2.5.2. Non-feasible observations are shown as circles and feasible ones as disks. The red color is used for non-dominated observations and the blue color is used for dominated ones. On the left column (input space), the particles used for optimizing the EI are shown as small dots, the color of which indicates the EI value. On the right column, the particles used for computing the EI are shown as black dots. They are distributed from the L_2^{opt} density.

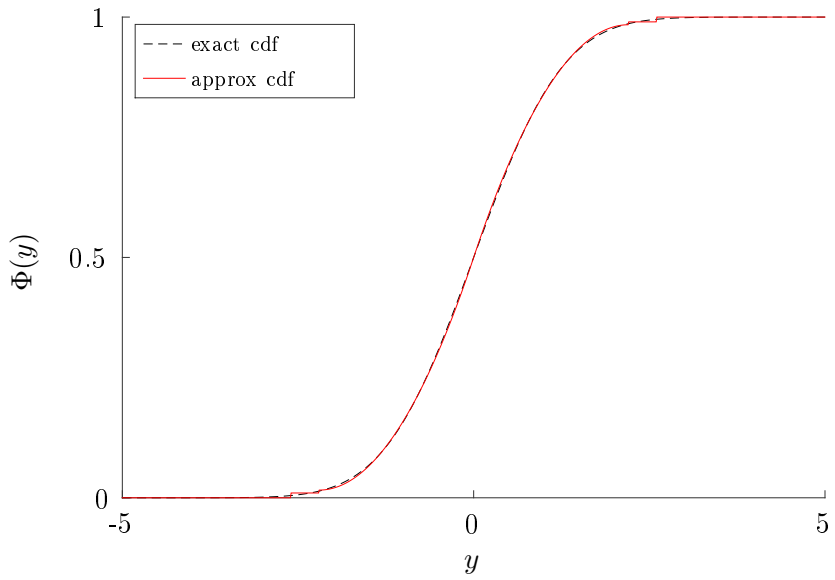


Figure 3.4: Comparison between the exact normal cumulative distribution function and the approximation (3.14).

In Figures 3.5 and 3.6, we show the results of experiments using the L_2^{opt} density and the uniform density in the situations illustrated in Figure 3.2. In Figure 3.5, we compute the estimates provided by the approximation method when the number $m_{\mathbb{Y}}$ of particles increases, and compare them to their exact values. For this experiment, we consider a bi-objective problem with $f_1(x) = x_1$ and $f_2(x) = x_2$ and we take $\mathbb{X} = \mathbb{B}_o = [0,1]^2$. The front of non-dominated points is made of 50 points randomly distributed on the first quadrant of a circle of radius $r = 0.8$ and we consider one thousand candidates distributed between the first quadrants of two circles of radii 0.75 and 0.8, as illustrated in Figure 3.2. The GP models are built using 20 points randomly distributed on $\mathbb{X}$. As expected, the uniform density fails to provide good estimates in this case, and better results are obtained with the L_2^{opt} density. For both densities, the quality of the prediction increases when the number of particles used in the approximation increases.

Then in Figure 3.6, we show the results of experiments when y^{low} is set too low and the number of objectives augments. For this experiment, $y^{\text{low}} = (-0.5, \ldots, -0.5)$ and the approximation is made using $m_{\mathbb{Y}} = 1000$ particles. We consider problems with p objectives $f_i(x) = x_i$, $1 \leq i \leq p$ and we take $\mathbb{X} = [0,1]^p$. The front of non-dominated points is made of 20 points randomly distributed on the first quadrant of an hypershpere of radius $r = 0.8$ and we consider one thousand candidates distributed between the first quadrants of two hyperspheres of radii 0.5 and 0.8, as illustrated in Figure 3.2. The GP models are built using $10p$ points randomly distributed on $\mathbb{X}$. Again, as expected, the uniform density fails to provide good estimates in this case, in particular when the number of objectives increases. Better results are obtained with the L_2^{opt} density, which is not sensitive to the choice of y^{low}. Note in particular that the quality of the approximation is maintained even when the number of objectives increases.

Remark 22 *In this section, it is implicitly assumed that π_n can be computed exactly. However, it*

is defined using an unnormalised probability density function, the normalizing constant of which is unknown. In our experiments, we use an approximation of the normalizing constant. The details about the approximation procedure can be found in Appendix 3.6.3.

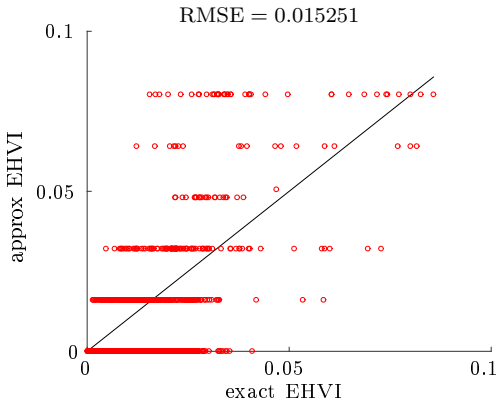
3.3.3 Complexity of the exact and approximate computation methods

The exact computation of the EHVI criterion for an arbitrary number of objectives is a challenging problem because the complexity of the integration domain rapidly grows when the number of objectives is greater than three. This problem has been addressed by Emmerich and Klinkenberg (2008); Emmerich et al. (2011); Couckuyt et al. (2014); Hupkens et al. (2015) and Emerich et al. (2016).

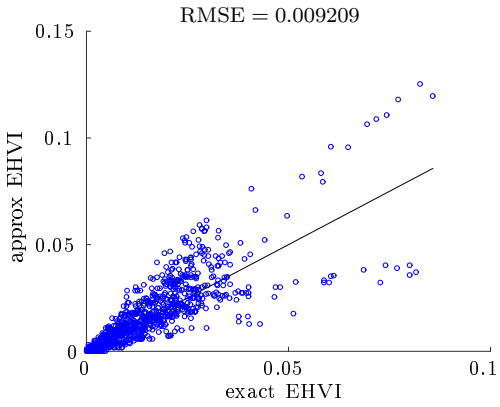
The method usually consists in a two-stage procedure (see, e.g., Couckuyt et al. (2014)). The dominated region is first decomposed into a set of (possibly overlapping) cells of rectangular shape over which the integration can be performed analytically. Then, the computation is done for the different candidate solutions by summing the contributions of each cell. Domain decomposition methods usually apply to the dominated set, whereas the EHVI criterion is defined on the non-dominated region. Therefore, in this section, we consider the following transformation of the integral in the expression of the EHVI:

$$\begin{aligned}
&\int_{\mathbb{B}_{\mathrm{o}} \setminus H_{n,\mathrm{o}}} \mathbb{P}_n \left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}} \\
&= \int_{\mathbb{B}_{\mathrm{o}}} \mathbb{P}_n \left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}} - \int_{H_{n,\mathrm{o}}} \mathbb{P}_n \left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}} \\
&= \prod_{i=1}^{p} \left[\mathbb{E}_n \left(\left(y_{\mathrm{o},i}^{\mathrm{upp}} - \xi_{\mathrm{o},i}(x) \right)_+ \right) - \mathbb{E}_n \left(\left(y_{\mathrm{o},i}^{\mathrm{low}} - \xi_{\mathrm{o},i}(x) \right)_+ \right) \right] - \int_{H_{n,\mathrm{o}}} \mathbb{P}_n \left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}} \,.
\end{aligned} \tag{3.15}$$

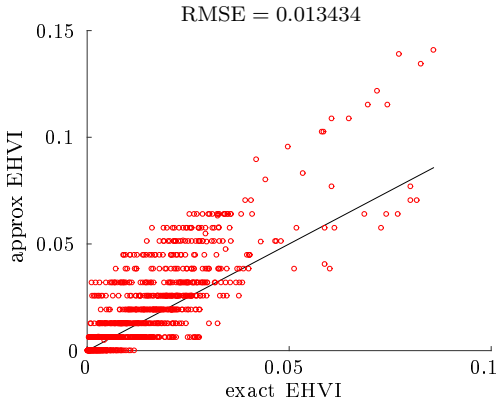
Let $D = (s_k, C_k)_{1 \leq k \leq K}$ be a signed decomposition of $H_{n,\mathrm{o}}$ composed of K cells with $C_k = [l_{k,1}, u_{k,1}] \times \ldots \times [l_{k,p}, u_{k,p}]$ and $s_k \in \{-1, 1\}$, $1 \leq k \leq K$ such that $\sum_k s_k \mathbb{1}_{C_k} = \mathbb{1}_{H_{n,\mathrm{o}}}$. Such a decomposition can be obtained using a modified hypervolume computation algorithm as explained in Couckuyt et al. (2014). We do not detail this particular point in this work. The interested reader is referred to the work of Lacour et al. (2017) and references therein for an up-to-date review of existing hypervolume computation algorithms. The EHVI criterion can then be computed exactly for any $x \in \mathbb{X}$ by summing the (possibly negative) contributions of



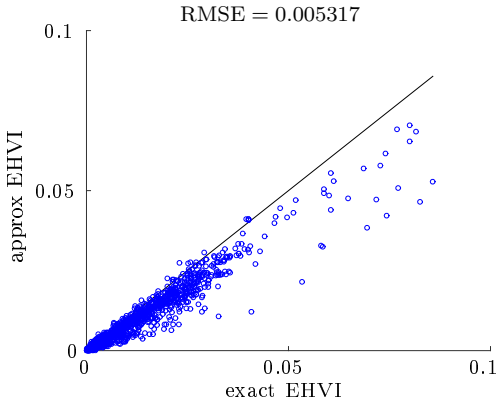
(a) $m_{\mathbb{Y}} = 200$



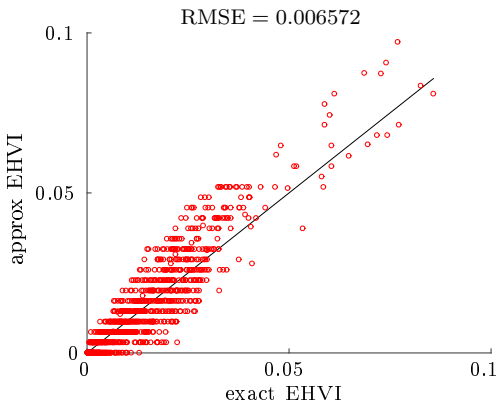
(b) $m_{\mathbb{Y}} = 200$



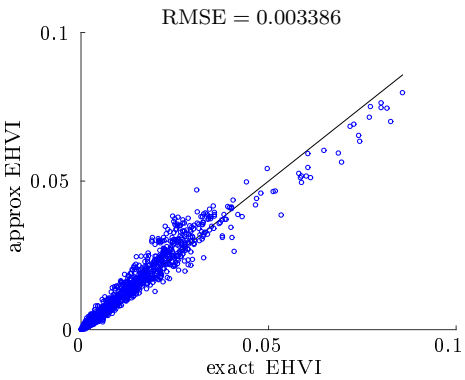
(c) $m_{\mathbb{Y}} = 500$



(d) $m_{\mathbb{Y}} = 500$

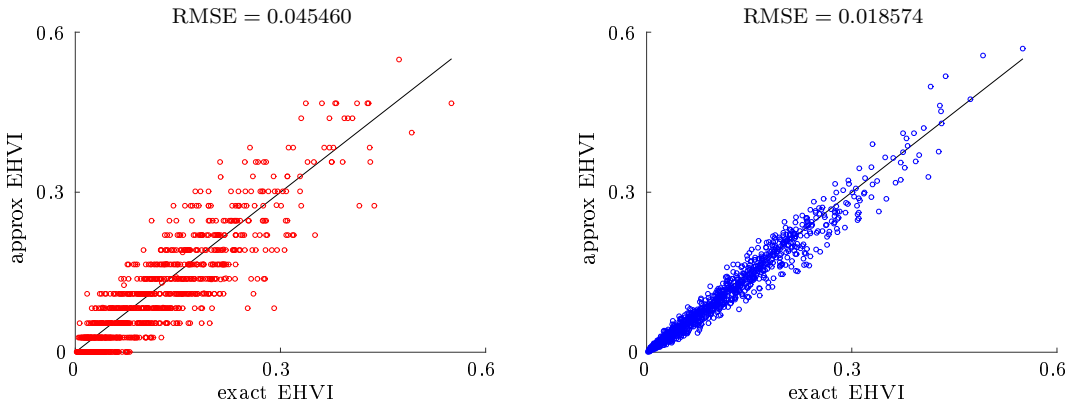


(e) $m_{\mathbb{Y}} = 1000$



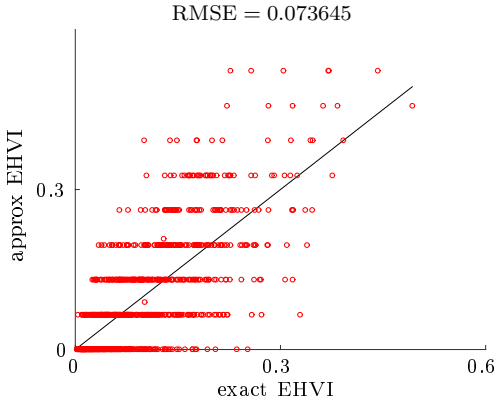
(f) $m_{\mathbb{Y}} = 1000$

Figure 3.5: Comparison of accuracy between the estimates provided by the SMC procedure when the uniform density is used (left) and when the L_2^{opt} density is used (right).

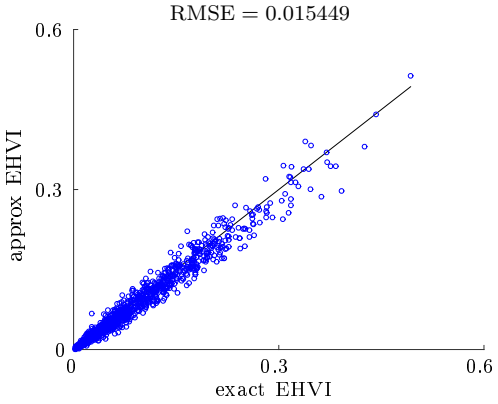


(a) $p = 4$

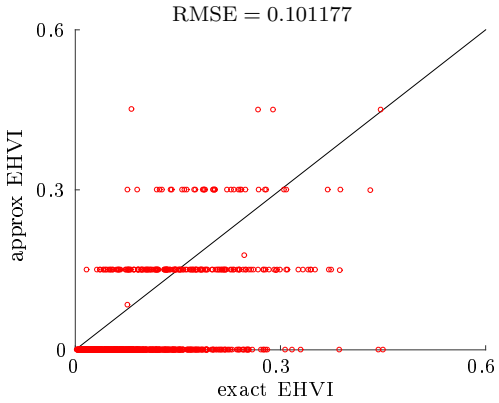
(b) $p = 4$



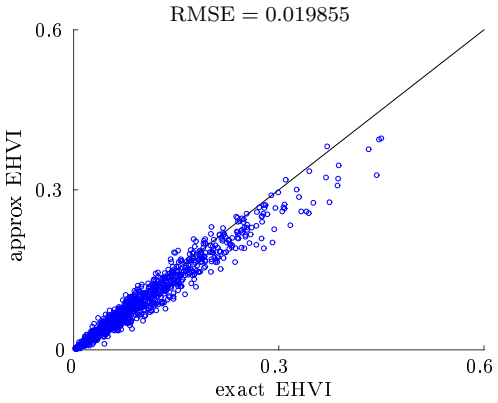
(c) $p = 6$



(d) $p = 6$



(e) $p = 8$



(f) $p = 8$

Figure 3.6: Comparison of accuracy between the estimates provided by the SMC procedure when the uniform density is used (left) and when the L_2^{opt} density is used (right).

each cell:

$$\begin{aligned}
\int_{H_{n,\mathrm{o}}} \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}} &= \sum_{k=1}^{K} s_k \int_{C_k} \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}} \\
&= \sum_{k=1}^{K} s_k \prod_{i=1}^{p} \int_{l_{k,i}}^{u_{k,i}} \mathbb{P}_n\left(\xi_{\mathrm{o},i}(x) \leq y\right) \mathrm{d}y \\
&= \sum_{k=1}^{K} s_k \prod_{i=1}^{p} \mathbb{E}_n\left(\int_{l_{k,i}}^{u_{k,i}} \mathbb{1}_{\left(\xi_{\mathrm{o},i}(x) \leq y\right)} \mathrm{d}y\right) \\
&= \sum_{k=1}^{K} s_k \prod_{i=1}^{p} \left[\mathbb{E}_n\left(\left(u_{k,i} - \xi_{\mathrm{o},i}(x)\right)_+\right) - \mathbb{E}_n\left(\left(l_{k,i} - \xi_{\mathrm{o},i}(x)\right)_+\right)\right].
\end{aligned} \tag{3.16}$$

The exact computation of the EHVI for a given candidate solution $x \in \mathbb{X}$ thus requires $N_{\text{exact}} = 2p(K+1)$ calls to the γ function defined in Section 2.2.1, which means that N_{exact} evaluations of the normal cumulative distribution function have to be performed for every candidate solution. This is actually responsible for most of the time required for computing the EHVI. Indeed, obtaining the decomposition D is fast when an efficient algorithm is used (see Figure 3.8). Besides, it has to be done only once at each iteration of the algorithm, and solely if the last evaluation brought an improvement. The number K of cells in the decomposition however, can be quite large, especially when the number of objectives and the number of non-dominated points are high, which causes both memory storage and computing time issues.

In Figure 3.8 we show the typical values of K for different number of objectives and non-dominated points, when the points are randomly distributed on the first quadrant of an hypersphere of radius 0.8 and $\mathbb{B}_{\mathrm{o}} = [0,1]^p$, as illustrated in Figure 3.7 (procedure detailed in Section 3.4.2). The algorithm used to compute the decomposition is an adaptation of the WFG algorithm of While et al. (2012), available in the Matlab/Octave STK toolbox of Bect et al. (2016b). Each experiment is repeated 30 times with different random seeds. As expected, the number of cells in the decomposition increases rapidly when the number of non-dominated points and the number of objectives augment. The time required to compute the decomposition remains reasonable in the ranges considered in the experiment.

The proposed approximation method also consists in a two-stage procedure. First, an SMC algorithm has to be run to obtain a cloud $\mathcal{Y}_n$ of $m_{\mathbb{Y}}$ particles distributed from a density $\pi_n^{\mathbb{Y}}$. Then, the approximation is carried out using (3.9), which requires $N_{\text{approx}} = m_{\mathbb{Y}} p$ evaluations of the normal cumulative distribution function for each candidate solution. Sampling $\mathcal{Y}_n$ takes a few seconds at most when the uniform density is used. It is more time consuming when the L_2^{opt} density is used, because each evaluation of the density requires $N = m_{\mathbb{X}} p$ evaluations of the normal cumulative distribution function (see (3.13)). In practice, we use the approximation (3.14) to lower the computational cost of the procedure.

When the optimization of the criterion is performed using the SMC procedure of Section 2.4.2, the EHVI has to be computed exactly $m_{\mathbb{X}}$ times at each iteration to select the best candidate solution. The total cost of using the exact computation method is thus approximately $N_{\text{tot}}^{\text{exact}} = 2pm_{\mathbb{X}}(K+1)$ if we neglect the time required to compute the decomposition. In comparison, the

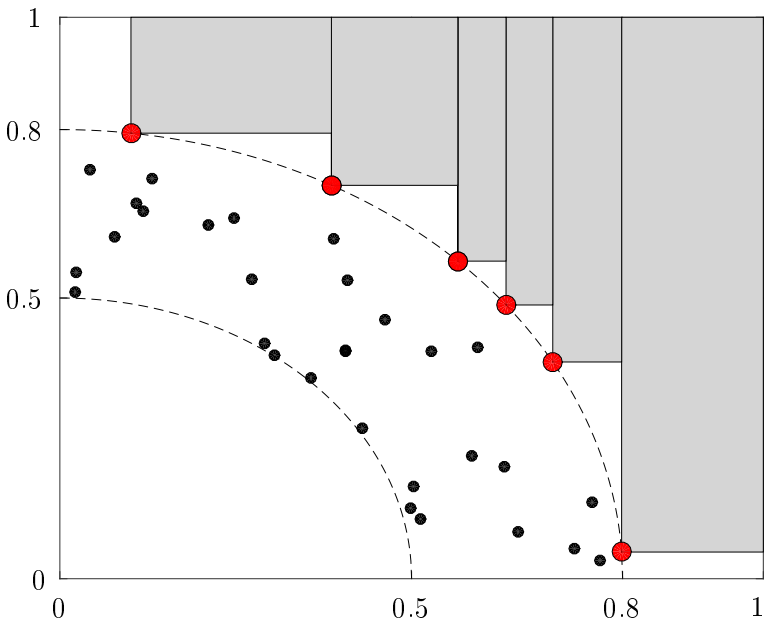


Figure 3.7: Illustration of the settings of the experiments of Figure 3.8 and 3.9 when $p = 2$. The non-dominated points are drawn randomly on the first quadrant of an hypershere of radius 0.8 and are represented as red disks. The decomposition of the dominated region is shown as gray faced rectangles. The black dots are the candidates used in the experiments of Figure 3.9. They are randomly distributed between the first quadrants of two hypershperes of radii 0.5 and 0.8 (dashed lines).

total cost of using the approximation method is approximately $N_{\text{tot}}^{\text{approx}} = m_{\mathbb{X}} m_{\mathbb{Y}} p$ if we neglect the sampling time, which is much lower when K is large.

In Figure 3.9 we show the typical time required to compute the EHVI for a population of one thousand candidates when the exact and approximate methods are used. The times that are reported include both stages of the computation for both methods. For this experiment, we consider a simple problem where $\mathbb{X} = [0, 1]^p$ and $f_i(x) = x_i$ for $i \in [\![1, p]\!]$. The GP models are built using Latin-hypercube designs of $N = 10p$ experiments and the candidates are drawn randomly between the first quadrants of two hyperspheres of radii 0.5 and 0.8. The non-dominated points are randomly distributed on the first quadrant of the hypershpere of radius 0.8 and we take $\mathbb{B}_{\text{o}} = [0, 1]^p$ as previously (see Figure 3.7). The number p of objectives varies from 4 to 6 and the number $m_{\mathbb{Y}}$ of particles used in the approximate computation methods is successively 200, 500, 1000 and 2000. For each pair $(p, m_{\mathbb{Y}})$, we repeat the experiment 30 times with different random seeds.

Note the rapid increase of the exact method computation time when the number of objectives and non-dominated points augment. In practice it is impractical for problems with more than five objectives. As regards the computation time of the approximate method, we note a strong dependence to the density that is used. When the uniform density is used, the computation is almost instantaneous which is a quality that is expected from Monte-Carlo approximation methods. When the L_2^{opt} density is used on the other hand, the computing time is affordable but not negligible. More work would be required on the L_2^{opt} density to make it as computationally

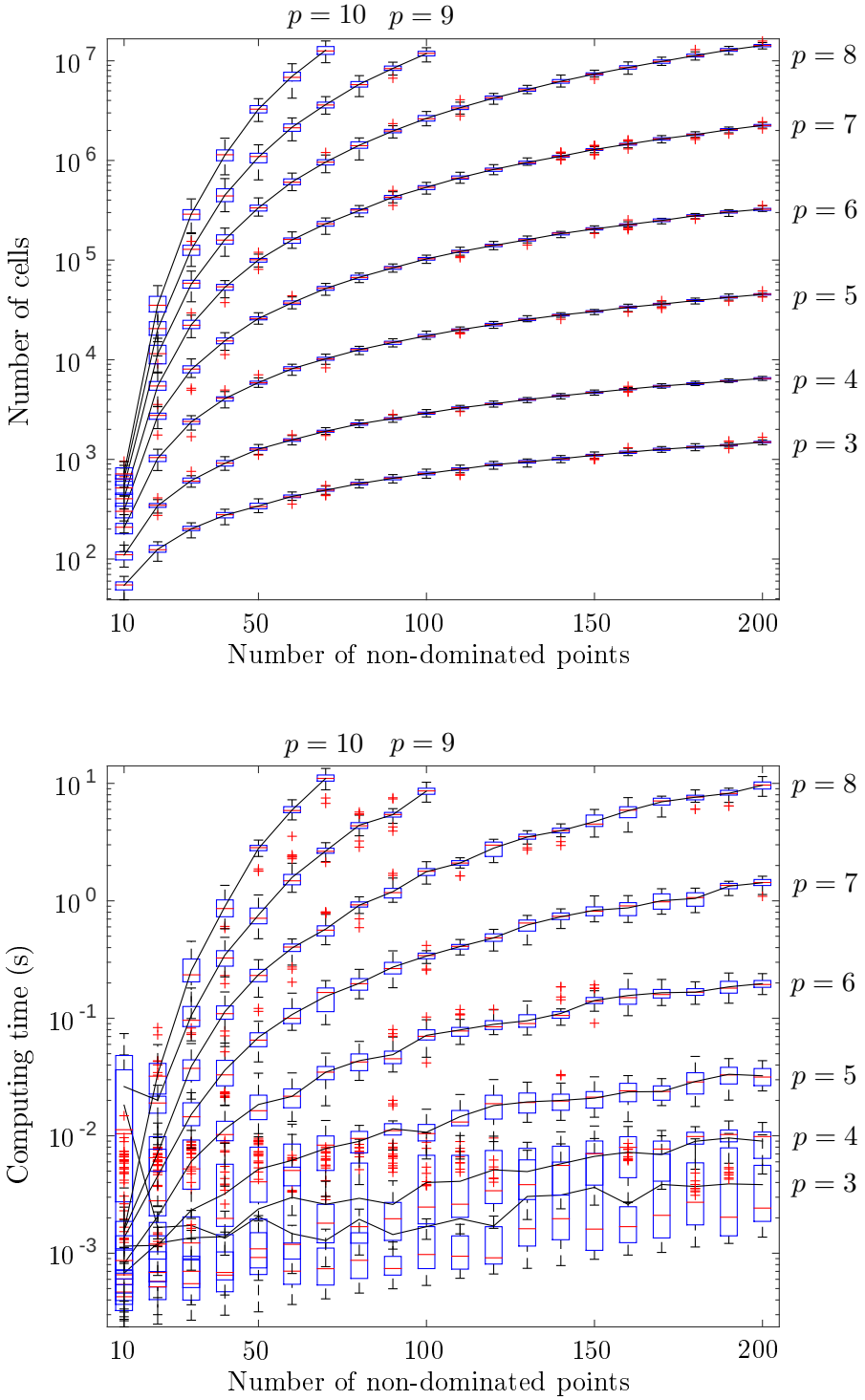


Figure 3.8: Number K of cells in the decomposition of the dominated region (top) and computing time required to compute the decomposition (bottom) when the non-dominated points are drawn randomly on the first quadrant of an hypersphere of increasing dimension (see Figure 3.7). The boxplots represent the distributions obtained over 30 repetitions with different random seeds (the mean is shown as a black line).

efficient as the uniform density and this shall motivate future work on this aspect of the BMOO algorithm.

Remark 23 *For the approximate computation method, we start the procedure from scratch, i.e. with $\mathcal{Y}_0$ uniformly distributed on $\mathbb{B}_{\mathrm{o}}$. Note that in this setting, the time required to compute the approximate EHVI is overestimated because the algorithm has to make more transitions to reach the target density during the sampling phase than it would in a sequential optimization scenario.*

3.3.4 Toward a better control of the sample size

Looking at the results of Figure 3.5, and in particular the results with $m_{\mathbb{Y}} = 200$, it can be seen that the choice of the number of particles in the approximate EI computation method has a strong influence on the quality of the approximation. In our experiments, we often use one thousand particles and we obtain good results with this choice.

In fact, the problem is not so much to obtain a good approximation of the EI for all candidate points, but rather to identify with a good confidence its maximizer. In practice, very close points may have very similar EIs and a large sample size would be required to distinguish them. It seams more reasonable then, to only look for a good candidate point, i.e. a point for which the EI value is likely to be high. In this section, we propose a simple quantile based approach to this purpose.

As before, let $(x_{n,k})_{1\leq k\leq m_{\mathbb{X}}} \in \mathbb{X}^{m_{\mathbb{X}}}$ be a set of points at which we want to estimate the value of the EI criterion, and denote $(\widehat{\rho}_{n,k})_{1\leq k\leq m_{\mathbb{X}}}$ the vector of the estimates produced by the proposed sequential Monte Carlo procedure, i.e. $\widehat{\rho}_{n,k} \approx \rho_n(x_{n,k})$, $k \in [\![1, m_{\mathbb{X}}]\!]$. Under an idealized setting where the successive clouds of particles $(\mathcal{Y}_k)_{0\leq k\leq n}$ are independent and where, for every $k \in [\![0, n]\!]$, the particles $(y_{k,i})_{1\leq i\leq m_{\mathbb{Y}}}$ are independently and identically distributed from a density π_k, it is possible to produce a closed form formulation of the covariance structure of $(\widehat{\rho}_{n,k})_{1\leq k\leq m_{\mathbb{X}}}$. In Appendix 3.6.3, we show that in this setting

$$\frac{\mathrm{Cov}\left(\widehat{\rho}_{n,i}\,,\,\widehat{\rho}_{n,j}\right)}{\rho_n(x_{n,i})\rho_n(x_{n,j})} \approx A_n^2 \cdot \left(\widehat{\Lambda}_n^{i,j}(x_i, x_j) + \left(1 + \widehat{\Lambda}_n^{i,j}(x_i, x_j)\right) \cdot \widehat{\Delta}_n^2\right), \tag{3.17}$$

where $\widehat{\Delta}_n$ is computed recursively using (3.57), and $\widehat{\Lambda}_n^{i,j}(x_i, x_j)$ is computed using (3.63). The term A_n is equal to $|\mathbb{B}_{\mathrm{o}}|$ in the first phase of the optimization process, i.e. prior to finding a feasible solution, and to $|\mathbb{B}_{\mathrm{c}}^-| \cdot \mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0)$ in the second phase of the optimization process, i.e. once a feasible solution is known.

In the idealized setting mentionned above, the proposed SMC approximation method can be viewed as a multi-level version of the ideal adaptive algorithm studied by Cérou et al. (2012). In their work, they show that in this setting, the distribution of the estimators is asymptotically normally distributed with a bias that is negligible compared to its standard deviation. Motivated by these results, we shall assume in the following that $(\widehat{\rho}_{n,k})_{1\leq k\leq m_{\mathbb{X}}}$ is distributed from a multivariate Gaussian distribution Ψ_n with mean $\mu_n = (\rho_n(x_{n,k}))_{1\leq k\leq m_{\mathbb{X}}}$ and a covariance matrix $\Sigma_n \in \mathcal{M}_{m_{\mathbb{X}}\times m_{\mathbb{X}}}$ computed from (3.17).

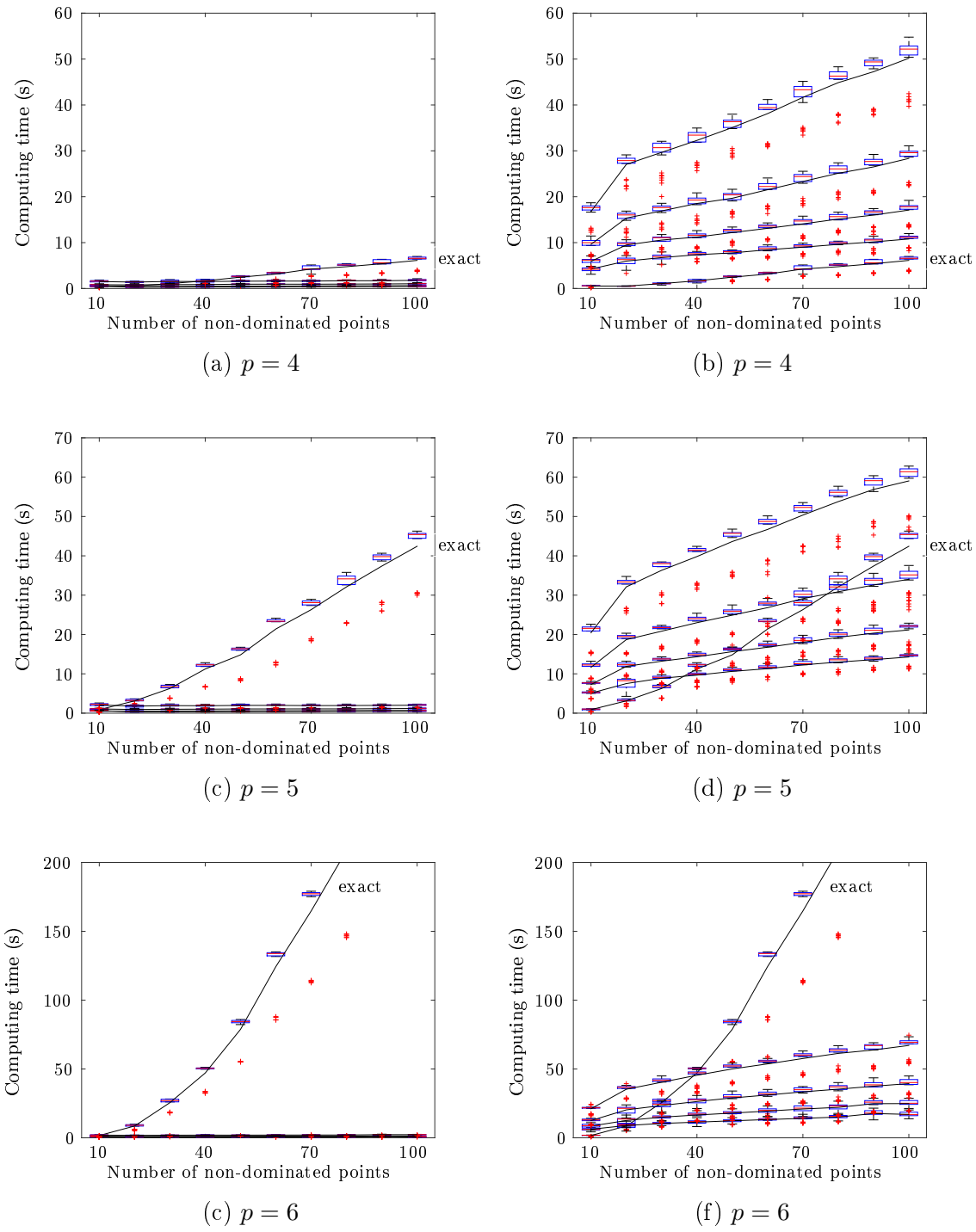


Figure 3.9: Time required for the computation of the EHVI for one thousand candidates (settings illustrated in Figure 3.7). On each subfigure, the results of the exact method are compared with the results of the approximate method using successively (from bottom to top) $m_Y = 200$, 500, 1000 and 2000 particles. The density used in the approximate method is either the uniform density (left column) or the L_2^{opt} density (right column). The boxplots represent the distributions obtained over 30 repetitions with different random seeds (the mean is shown as a black line).

Let $x_n^\star$ denote the maximizer of $\widehat{\rho}_n$ among the $(x_{n,k})_{1\leq k\leq m_{\mathbb{X}}}$ and $\alpha \in [0,1]$, and consider the quantile q_α of level α of the empirical distribution of the $(\widehat{\rho}_{n,k})_{1\leq k\leq m_{\mathbb{X}}}$. We shall say that $m_{\mathbb{Y}}$ is sufficiently large to accept $x_n^\star$ with a confidence of level α if $\widehat{\rho}_n(x_n^\star) - 3\,\mathrm{Std}(\widehat{\rho}_n(x_n^\star)) \geq q_\alpha$. In other words, we decide to accept $x_n^\star$ only if the probability that its EI value is larger than q_α is greater than α. Otherwise, we decide that the sample size is not sufficiently large, i.e. the variance is not small enough, to conclude that $x_n^\star$ is indeed a good point. When this happens, we propose to simply restart the SMC procedure with a larger number of particles.

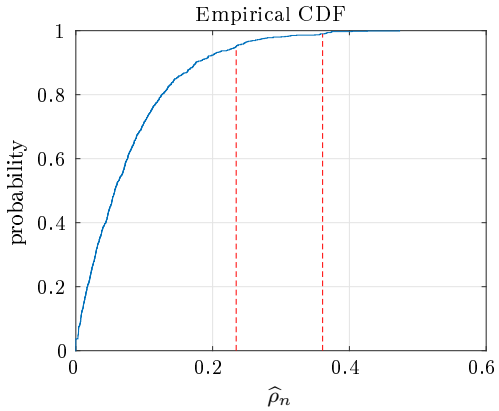
As a proof of concept, the operation of the proposed approach is illustrated in Figure 3.10. The experiment is made using the same settings as in Figure 3.7 with $p = 6$ and we consider the cases where $\alpha = 0.95$ and $\alpha = 0.99$. It can be seen that for both values of α, the method refuses $x_n^\star$ when $m_{\mathbb{Y}} = 200$ because the estimate cannot be trusted. When $m_{\mathbb{Y}} = 500$, $x_n^\star$ is accepted when the threshold it set to $q_{0.95}$ but refused when it is set to $q_{0.99}$. It is accepted in both cases when $m_{\mathbb{Y}} = 1000$.

3.3.5 Conclusions

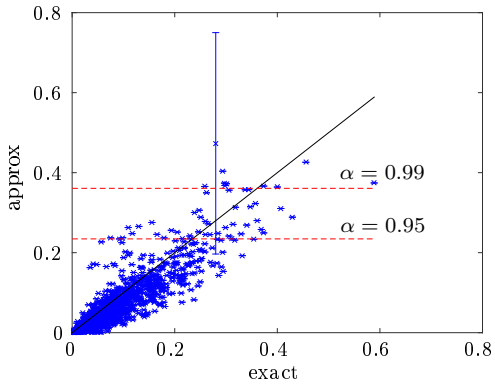
In this section, we discuss the computation of the new EI criterion in the case where independence between the functions of the problem is assumed. First, a novel sampling density, called the L_2^{opt} density, to be used in the sequential Monte-Carlo approximation procedure proposed in Section 2.4.1, is introduced. It is shown that the new density does not suffer from the limitations of the uniform density observed in Section 2.5.4. Then, an empirical study of the computational complexity of the exact computation method is made and it is shown that it is not practical for problems with more than five objectives. Finally, a simple online strategy to control the quality of the approximation is proposed.

Unlike the uniform density, the L_2^{opt} density takes the information about the points at which the criterion is to be computed into account in its definition. This permits to achieve very good approximation performances in most situations. However, it is not cheap-to-evaluate enough to be used as is within a sequential Monte Carlo algorithm and some approximations are necessary in its definition to make it computationally efficient. That being said, the time required to compute the criterion for a set of candidate solutions when this density is used remains in the order of the minute, which is negligible in the case where the functions of the problem are truly expensive to evaluate. Still, we believe that there is room for improvement of its computational efficiency. In particular, more work is required on the sampling process to lower the cost of the approximation procedure.

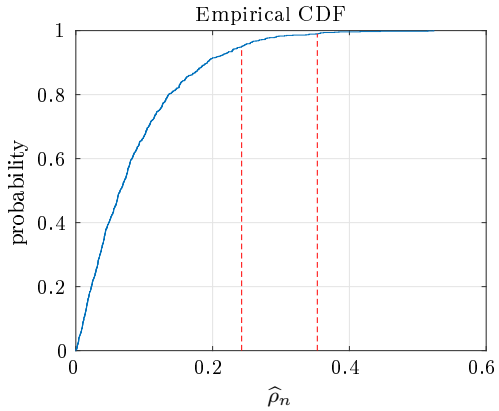
To control the quality of the approximation, a simple approach using the variance of the estimation at the maximizer of the criterion is proposed. However, more information could probably be extracted from the covariance between the approximations for all candidates. This could motivate future work on a strategy to determine adaptively the number of particles that should be used in the approximation procedure.



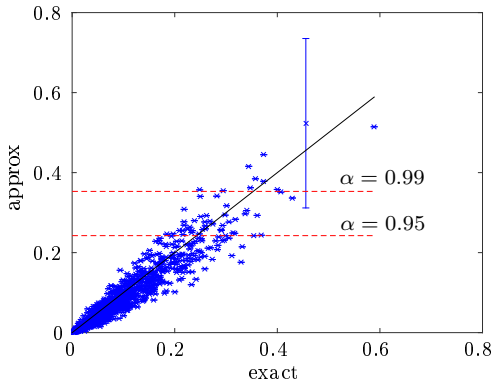
(a) $m_{\mathbb{Y}} = 200$



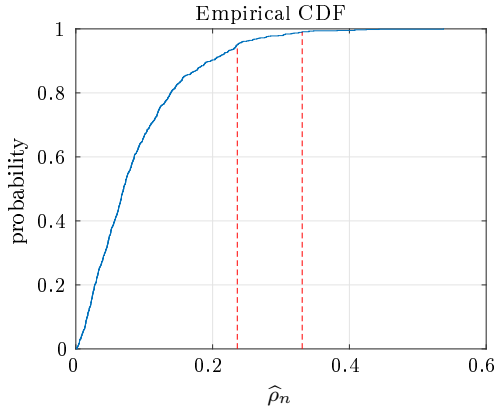
(b) $m_{\mathbb{Y}} = 200$



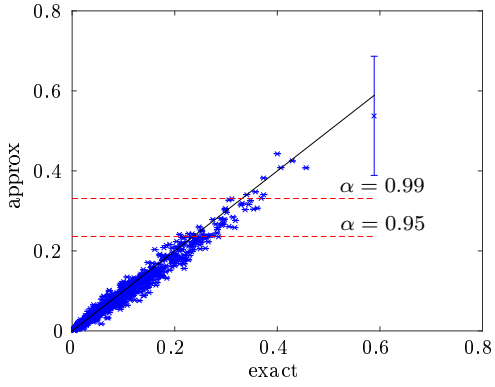
(c) $m_{\mathbb{Y}} = 500$



(d) $m_{\mathbb{Y}} = 500$



(e) $m_{\mathbb{Y}} = 1000$



(f) $m_{\mathbb{Y}} = 1000$

Figure 3.10: Illustration of the operation of the approach proposed for controling the quality of the estimation of the EI.

3.4 BMOO for Bayesian Many-Objective Optimization

3.4.1 Introduction

In this section, we assess BMOO on many-objective problems. Such problems often emerge in engineering design optimization (see, e.g., Fleming et al. (2005)). However, they are often reformulated in a less complex form (functions that should be objectives are aggregated or formulated as constraints for example) because high dimensional objective spaces are difficult to handle (see, e.g., Ishibuchi et al. (2008)) and because the exploitation of many-objective optimization results is not straightforward.

Most of the documentation on many-objective optimization can be found in the evolutionary multi-objective (EMO) literature. An up-to-date review of approaches that have been proposed in this setting can be found in the PhD thesis of Li (2015). The reader is also referred to the works of Wagner et al. (2007); Ishibuchi et al. (2008); Bader and Zitzler (2011) and Yang et al. (2013) for more details about this class of approaches. In the Bayesian literature, the many-objective optimization problem has been studied, e.g., by Shimoyama et al. (2013a); Couckuyt et al. (2014) and Luo et al. (2015).

The section is organized as follows. In Section 3.4.2, we introduce the FICUS problem. This problem is a configurable test problem for which it is possible to control the number of objectives and the curvature of the Pareto front. Also, the Pareto front for this problem is known explicitly. A closed form expression of the hypervolume it dominates is known and independent samples can be drawn on its surface with a simple procedure. The FICUS test problem is thus convenient for evaluating the performances of many-objective optimization algorithms. In Section 3.4.3, the FICUS problem is used to empirically study the distribution of solutions obtained by sequentially maximizing the hypervolume indicator (see, e.g., Auger et al. (2009c)). The influence of the choice of the reference point and the influence of the curvature of the Pareto front are discussed. Then, experimental results obtained by the BMOO algorithm on the FICUS test problem are presented in Section 3.4.4. Conclusions are drawn in Section 3.4.5.

3.4.2 The FICUS test problem

We define the following test problem:

$$\begin{array}{rccl} \text{FICUS } (p,r,c) & : & [0,1]^p & \to & \mathbb{R} \\ & & x=(x_1,\ldots,x_p) & \mapsto & \left\{ \begin{array}{lcl} f_1(x) & = & x_1\,, \\ \ldots, & & \\ f_p(x) & = & x_p\,, \\ c(x) & = & r^c - \sum_{i=1}^p x_i^c\,. \end{array} \right. \end{array}$$

The features of the FICUS problem are represented on Figure 3.11 for $p = 2$ and for different values of r and c. The Pareto front for this problem is the first quadrant of an hypersphere of dimension p and radius r, in the norm L_c. It is concave for $c > 1$, linear for $c = 1$ and convex for $c < 1$.

This problem is interesting in particular for two reasons. First, the volume $V_p^c(r)$ of the region dominated by the Pareto front of the FICUS (p, r, c) can be computed in closed form:

$$V_p^c(r) = \prod_{i=1}^{p} R_i - \frac{\Gamma\left(1+\frac{1}{c}\right)^p}{\Gamma\left(1+\frac{p}{c}\right)} r^p \,, \tag{3.18}$$

where $R = (R_1\,,\,\ldots\,,\,R_p)$ is the reference point and Γ denotes the Gamma function[7]. In practice, the volume $V_p^c(r)$ can be used as a reference to assess the performance of an optimization strategy with respect to the hypervolume indicator.

Second, it is possible to draw samples distributed on the Pareto front of the FICUS problem. Let $(Z_i)_{1\leq i\leq p}$ be random variables independently and identically distributed from a centred reduced normal distribution. Then,

$$S = \left(\frac{r|Z_1|}{(\sum_{i=1}^{p} Z_i^c)^{\frac{1}{c}}}, \ldots, \frac{r|Z_p|}{(\sum_{i=1}^{p} Z_i^c)^{\frac{1}{c}}} \right), \tag{3.19}$$

is distributed on the Pareto front of the FICUS (p, r, c) problem[8]. Let then $(S_i)_{1\leq i\leq m}$ be m independent copies of S. The $(S_i)_{1\leq i\leq m}$ can be used to optimize the hypervolume indicator or to evaluate the quality of Pareto approximation sets.

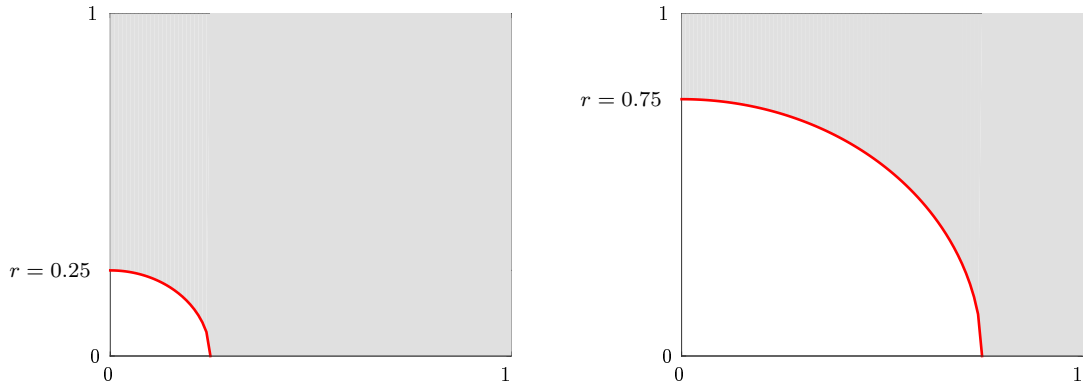
3.4.3 Empirical study of the hypervolume

The hypervolume indicator has several desirable properties. First, it is the only known Pareto-compliant unary indicator (see Zitzler et al. (2003)). Second, it was shown by Fleischer (2003) that the hypervolume dominated by a set of μ points, $\mu > 0$, is maximal only if all μ points are Pareto-optimal. Third, the distribution on the Pareto front of points resulting from the maximization of the hypervolume is unchanged by a linear scaling of the objectives. As such, it has become very popular both as a measure to determine the quality of a Pareto approximation set (see, e.g., Zitzler and Thiele (1998); Laumanns et al. (1999); Knowles and Corne (2002)) and to design efficient multi-objective optimization algorithms (see, e.g., Beume et al. (2007); Emmerich et al. (2005); Knowles et al. (2003); Zitzler and Künzli (2004); Igel et al. (2007)). In particular, hypervolume-based optimization algorithms have been shown to outperform other algorithms on many-objective problems (see, e.g., Wagner et al. (2007); Brockhoff et al. (2008)).

In the EMO literature, optimal μ-distributions, i.e. sets of μ points that maximize the hypervolume indicator (the term can be generalized to other indicators as well), have been studied for the bi-objective case by Auger et al. (2009c); Friedrich et al. (2009); Bringmann and Friedrich (2010) and for three objectives by Auger et al. (2010). It was shown by Auger et al. (2009c) in the bi-objective case and for continuous fronts, that the distribution of solutions along the front is asymptotically proportional to the square root of the negative of the first derivative of the front. In the same paper, it is shown that for some fronts, it is not possible to include the extreme points in the μ-optimal distribution, regardless of the choice of the reference point

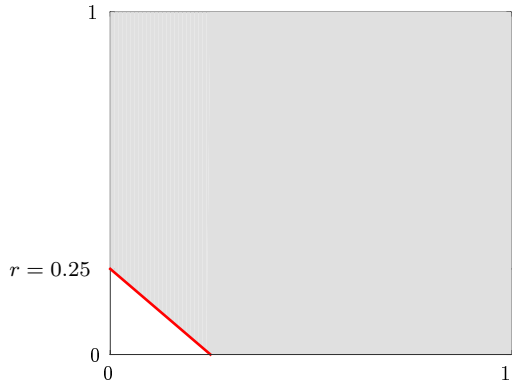
[7] For simplicity, it is assumed that $R_i > r$ for all $i \in [\![1, p]\!]$.

[8] The distribution is uniform for $c = 2$ but this result does not extend to $c \neq 2$.

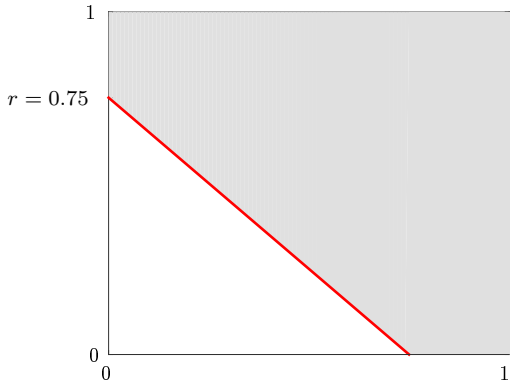


(a) $c = 2$

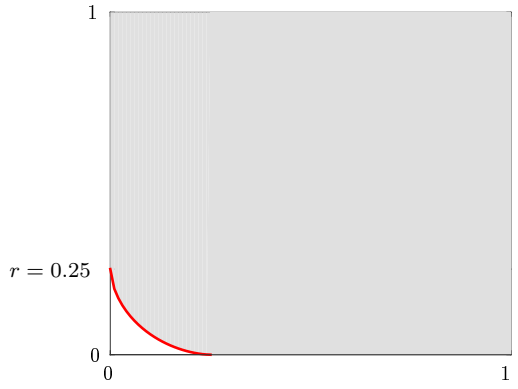
(b) $c = 2$



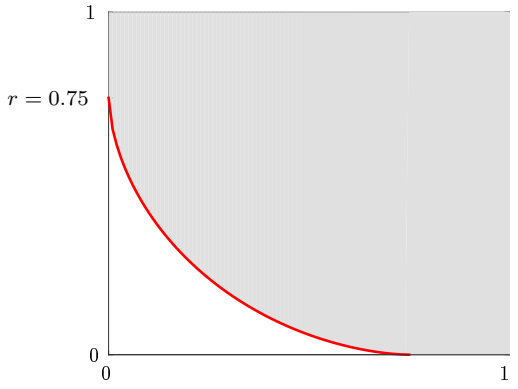
(c) $c = 1$



(d) $c = 1$



(e) $c = 0.6$



(f) $c = 0.6$

Figure 3.11: Illustration of the features of the FICUS test problem for $p = 2$ and for different values of r and c. The grey area represents feasible objectives values and the Pareto front is represented in red.

used to define the hypervolume. This last statement is shown to extend to problems with three objectives by Auger et al. (2009c).

To the best of our knowledge, such theoretical results on the distribution of solutions obtained by *sequentially* maximizing the hypervolume (one point at a time) are not available, and there is no guarantee that the aforementioned results extend to this case. In general, algorithms maximizing the hypervolume achieve well-spread Pareto front approximation sets (see, e.g., Knowles et al. (2003); Emmerich et al. (2005)). However, Zitzler and Thiele (1998) report on the tendency of the hypervolume to favor convex regions over concave ones and Deb et al. (2005) point out the bias of the hypervolume toward boundary solutions on some problems.

For the FICUS problem, a sequential (approximate) maximization of the hypervolume indicator can be achieved using a large number of samples independently distributed on the Pareto front as said in Section 3.4.2. For better performances, in our experiments, we use Sobol sequences to build the samples and manually add the extreme points of the front. Pareto approximation sets obtained using this procedure over 50 iterations for the FICUS problem with $p = 3$, $r = 0.5$ and different values for c and for the reference point R are shown in Figure 3.12.

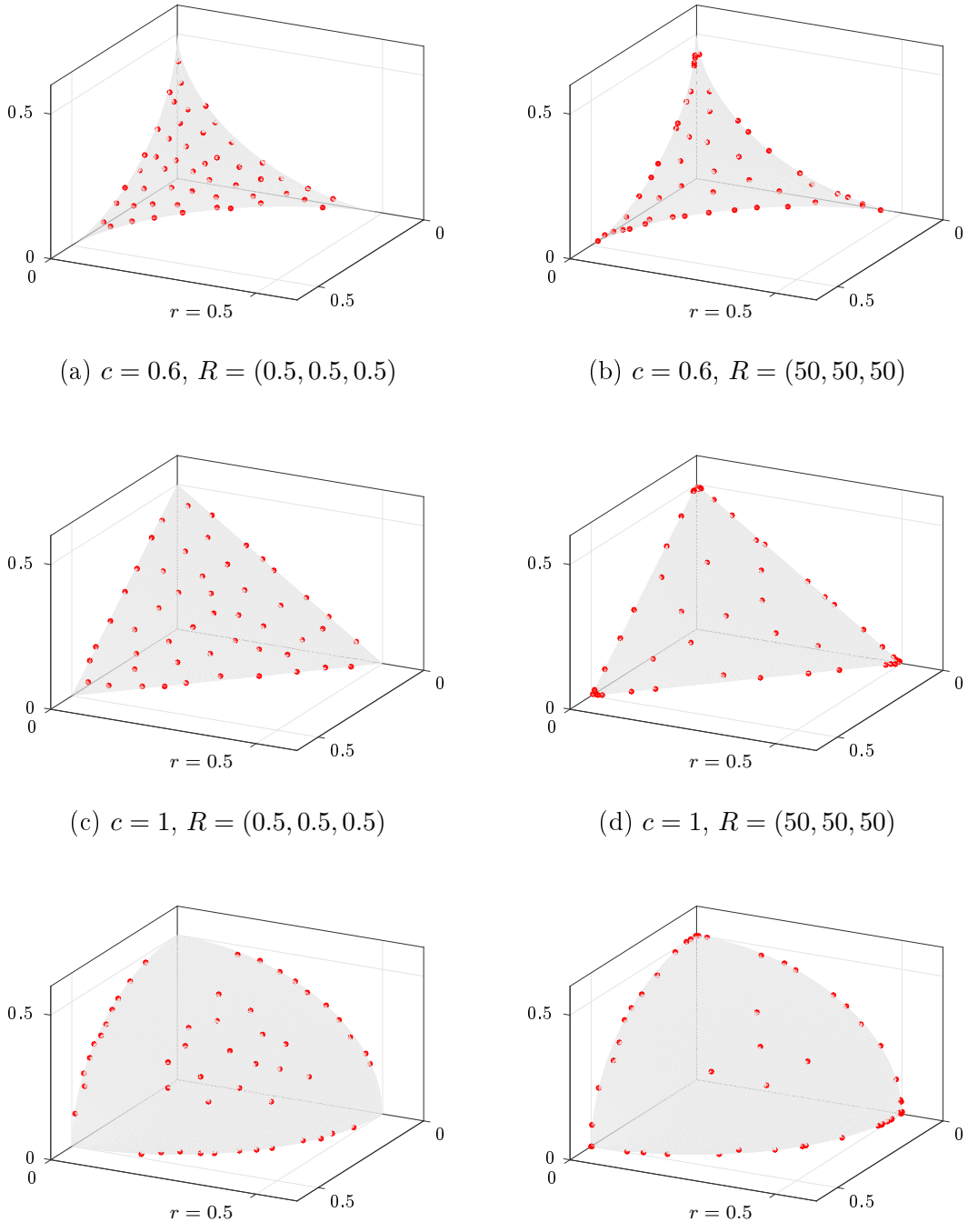
Several observations can be made based on the results of this experiment. First, we note that, as reported by Zitzler and Thiele (1998), the spread of the distribution is not satifactory on concave problems (corresponding to the subfigures (e) and (f) with $c = 2$). Large regions between the boundaries and the center of the Pareto front are not represented in the approximation set. This observation holds independently of the choice of the reference R. As regards the results when $c = 0.6$ (convex front) and $c = 1$ (linear front), we obtain a better spread of solutions. We observe that when the reference point is chosen as the nadir point, i.e. $R = (r, r, r)$ for the FICUS problem, the extreme points of the Pareto front are not contained in the Pareto approximation set and that when it is set far from the front, the distribution tends to concentrate near the boundaries of the front. This is in line with the observations of Deb et al. (2005) about the bias of the hypervolume toward boundary solutions.

In higher dimension, assessing the quality of a Pareto approximation set is difficult and a variety of metrics have been proposed to measure different quality aspects. The reader is referred to Jiang et al. (2014) for a review of such metrics. In this work, we consider the following three metrics.

$$\left\{\begin{array}{lcl} M_1(Y_n) & = & \dfrac{|H_{n,\mathrm{o}}|}{V_p^c(r)}, \\ M_2(Y_n) & = & \displaystyle\sum_{1\leq i\leq p} \min_{1\leq k\leq n} d(y_k, f_i^{max}), \\ M_3(Y_n) & = & 2 \displaystyle\max_{1\leq i\leq m} \min_{1\leq k\leq n} d(S_i, y_k), \end{array}\right. \tag{3.20}$$

where $Y_n = (y_1, \ldots, y_n) \in \mathbb{Y}_{\mathrm{o}}^n$ denotes a set of n observations, d denotes the Euclidean distance, $f_i^{max} \in \mathbb{Y}_{\mathrm{o}}^n$ denotes the extreme point of the front in the direction of the objective $i \in [\![1, p]\!]$ and $(S_i)_{1\leq i\leq m}$ is a large sample distributed on the Pareto front and augmented with the extreme solutions[9]. The metric M_1 measures the convergence with respect to the hypervolume indicator, using $R = (1\,, \ldots\,, 1)$ as a reference. It should tend to one when n augments. The metric M_2

[9] In our experiments, we take $m = 10000 + p$ and S is obtained using the procedure described in Section 3.4.2.



(a) $c = 0.6$, $R = (0.5, 0.5, 0.5)$

(b) $c = 0.6$, $R = (50, 50, 50)$

(c) $c = 1$, $R = (0.5, 0.5, 0.5)$

(d) $c = 1$, $R = (50, 50, 50)$

(e) $c = 2$, $R = (0.5, 0.5, 0.5)$

(f) $c = 2$, $R = (50, 50, 50)$

Figure 3.12: Pareto approximation sets obtained by sequentially maximizing the hypervolume over 50 iterations for the FICUS problem with $p = 3$, $r = 0.5$ when different values for c and for the reference point R are used. The grey region represents the Pareto front. The red dots are the iterative maximizers of the hypervolume.

is used to assess whether the extreme points of the front are well represented by the Pareto approximation set Y_n. The optimal value for this metric is zero, which indicates that all extreme solutions are contained in the approximation set. Last, the metric M_3 measures the approximate diameter of the largest ball centred on the Pareto front that can be inserted between elements of Y_n. It should be as small as possible.

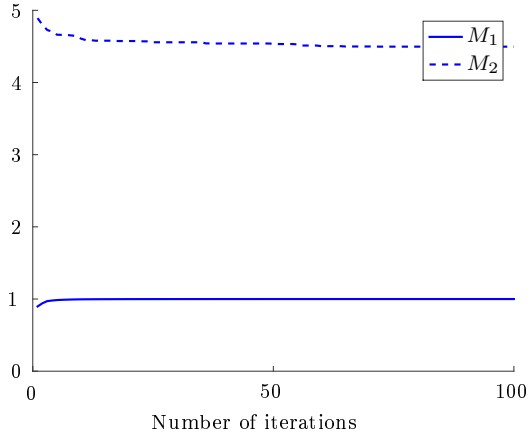
Experimental results obtained by sequential maximization of the hypervolume over 100 iterations for $p = 10$ are presented in Figures 3.13, 3.14 and 3.15. Different values for c and R are used and the approximation sets are evaluated using the metrics M_1, M_2 and M_3. Additional results for $p = 6$ and $p = 8$ can be found in the additional material of this chapter, in Section 3.6.4.

As a reference for the M_3 metric, we consider a strategy where Y_n is initialized with the extreme solutions and enriched sequentially by taking $Y_{n+1} = \text{argmax}_{(S_i)_{1\leq i\leq m}} \min_{1\leq k\leq n} d(S_i, y_k)$. In other words, Y_{n+1} is chosen as the point in $(S_i)_{1\leq i\leq m}$ that is farther from the points in Y_n. This strategy corresponds to a one-step lookahead minimization of the M_3 metric. The results obtained with this strategy are shown as a dashed line in Figures 3.13, 3.14 and 3.15.

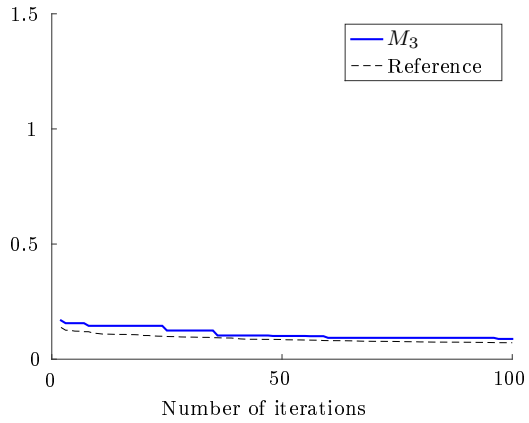
First, we look at the left columns in Figures 3.13, 3.14 and 3.15. For all considered values of c and R, we observe a rapid convergence of the M_1 metric (hypervolume). Regarding the results for the M_2 metric, we note a dependence to the choice of the reference R and, to a lesser extent, to the curvature of the front. When R is set at the nadir point (subfigure (a) in Figures 3.13, 3.14 and 3.15), the M_2 metric does not converges to zero. This indicates that the extreme solutions of the front are not well represented in the Pareto approximation set. When R is set away from the nadir point (subfigures (c) and (e) in Figures 3.13, 3.14 and 3.15), the metric converges to zero. In particular when $R = (1, \dots, 1)$ (subfigure (c)), the extreme solutions are the first points selected by the hypervolume. When R is set far from the Pareto front (subfigure (e)), the extreme solutions are among the first selected points when the front is concave but take more time to be selected when the front is more convex.

The results for the M_3 metric (right column in the figures) are close to those obtained by the reference strategy, which indicates that maximizing the hypervolume indicator sequentially yields a good coverage of the Pareto front on the FICUS problem. Contrary to our expectations, we do not observe the phenomenon observed in Figure 3.12 when $c = 2$ (empty regions between the boundaries and the center of the Pareto front).

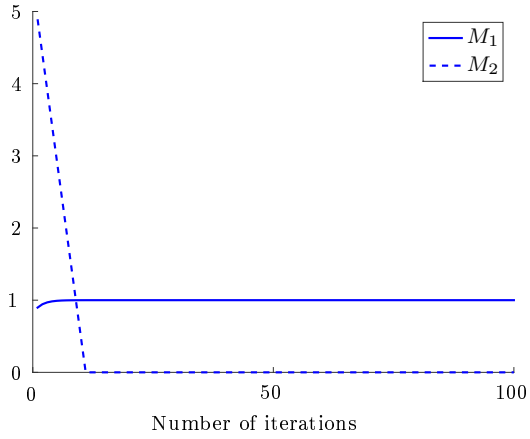
Remark 24 *Observe that the results for the M_3 metric are slightly better when $R = (0.5, \dots, 0.5)$ or $R = (1, \dots, 1)$ than when $R = (50, \dots, 50)$, and that the results for the M_2 metric are better when $R = (1, \dots, 1)$. In practice, this means that it is better to chose a reference a little away from the Pareto front but not too far, to favour both the extreme solutions and a good coverage of the front. On some problems though, it cannot be said in advance what will be the range of variation of the objectives along the Pareto front. In particular, when dealing with constrained problems, the objectives values corresponding to feasible solutions can be severely restricted (see the results of Section 4.3). The adaptive procedure proposed in Section 2.7.2 to set $\mathbb{B}_o$ is not suitable on such problems, because it does not aim at setting the reference close to the Pareto front. In Section 3.6.2, we propose a procedure that makes use of the information provided by the models to sequentially adapt the reference to keep it close to the Pareto approximation set.*



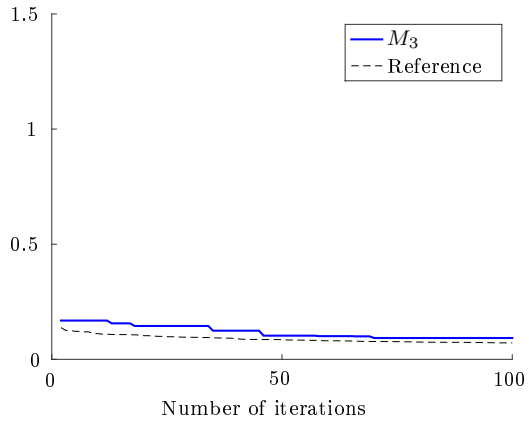
(a) $R = (0.5, 0.5, 0.5)$



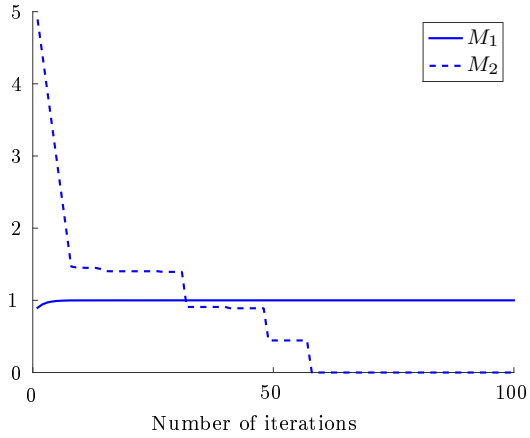
(b) $R = (0.5, 0.5, 0.5)$



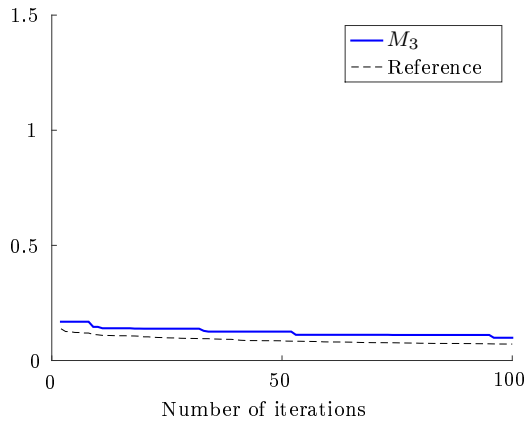
(c) $R = (1, 1, 1)$



(d) $R = (1, 1, 1)$

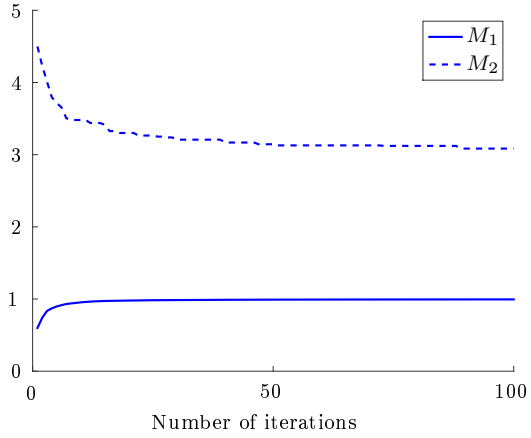


(e) $R = (50, 50, 50)$

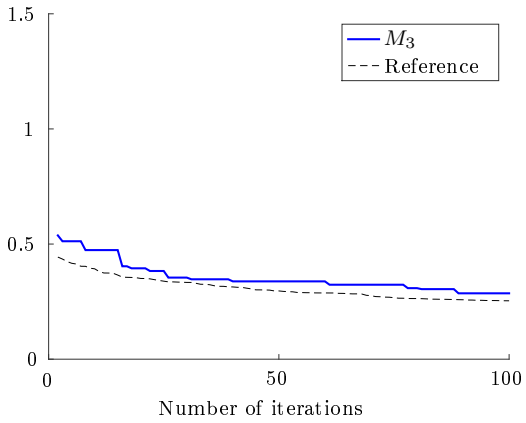


(f) $R = (50, 50, 50)$

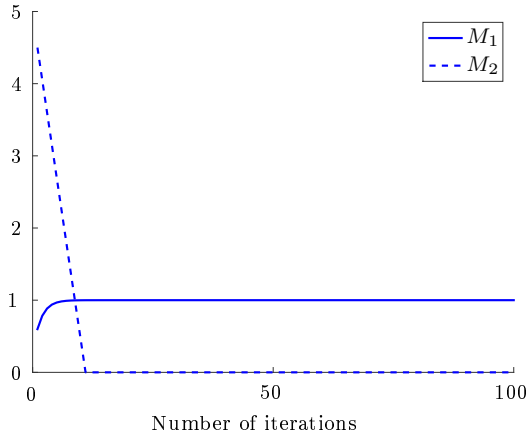
Figure 3.13: Results obtained on the FICUS $(10, 0.5, 0.6)$ problem (convex front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



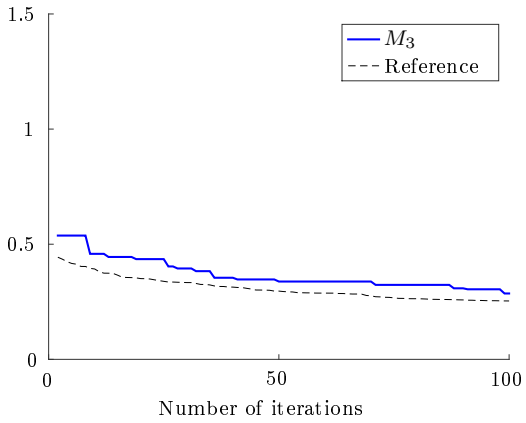
(a) $R = (0.5, 0.5, 0.5)$



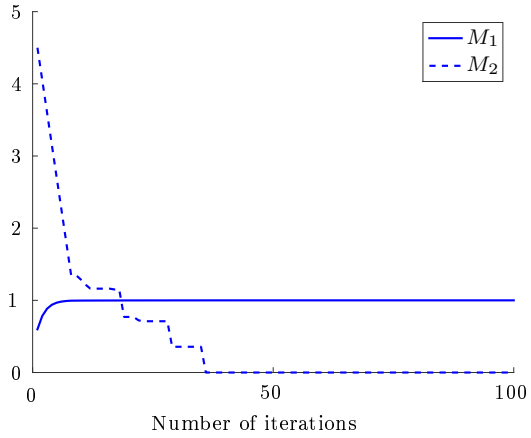
(b) $R = (0.5, 0.5, 0.5)$



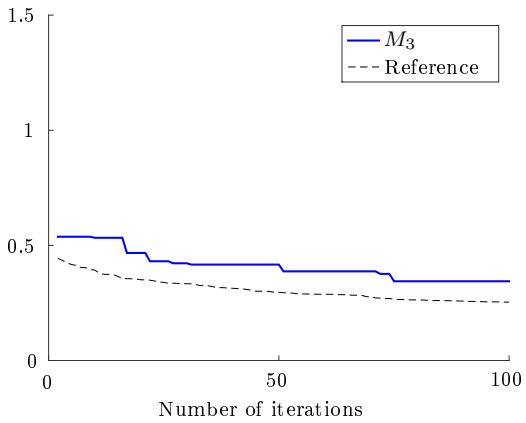
(c) $R = (1, 1, 1)$



(d) $R = (1, 1, 1)$

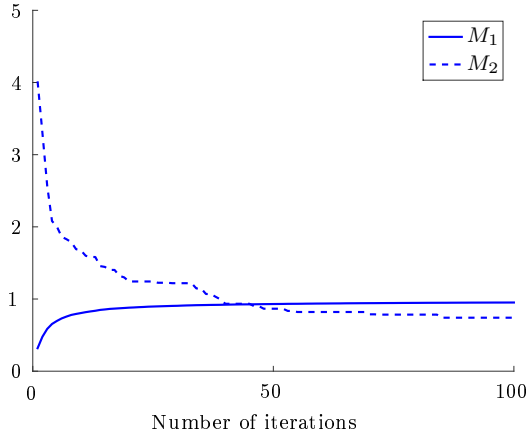


(e) $R = (50, 50, 50)$

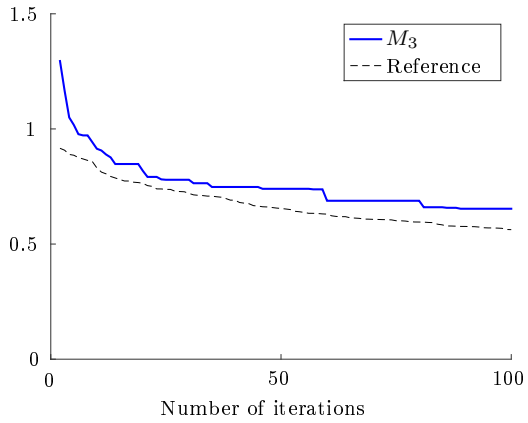


(f) $R = (50, 50, 50)$

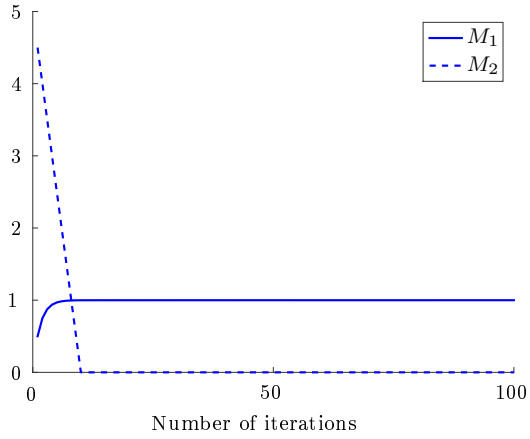
Figure 3.14: Results obtained on the FICUS $(10, 0.5, 1)$ problem (linear front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



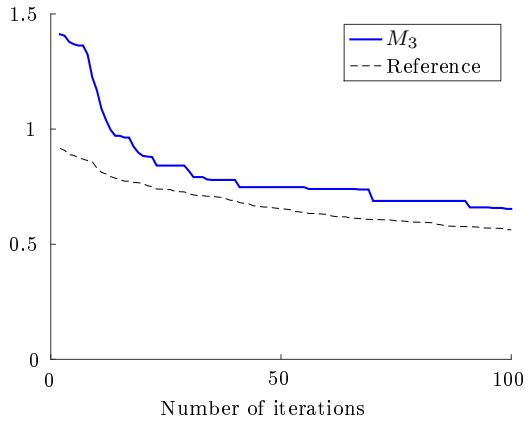
(a) $R = (0.5, 0.5, 0.5)$



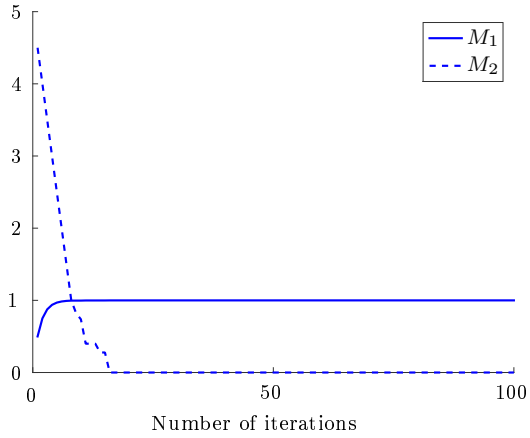
(b) $R = (0.5, 0.5, 0.5)$



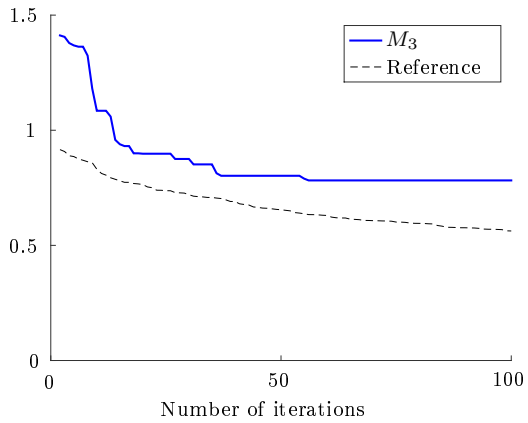
(c) $R = (1, 1, 1)$



(d) $R = (1, 1, 1)$



(e) $R = (50, 50, 50)$



(f) $R = (50, 50, 50)$

Figure 3.15: Results obtained on the FICUS $(10, 0.5, 2)$ problem (concave front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.

3.4.4 Numerical experiments

In Section 3.4.3, we study a strategy which consists in using a large number of samples distributed on the Pareto front for maximizing the hypervolume indicator. This strategy can be seen as an idealized version of the BMOO algorithm where the GP models are perfect, the criterion is computed exactly and its maximization is made almost perfectly. As such, the results presented in Figures 3.13, 3.14 and 3.15 and in Section 3.6.4 constitute a reference that can be used to assess the performances of BMOO on the FICUS problem.

In Figures 3.16, 3.17 and 3.18 we show the results of experiments made with BMOO on the FICUS problem when the number of objectives is $p = 6$, the radius is $r = 0.5$ and $c \in \{0.6, 1, 2\}$. Additional results for $p = 8$ can be found in Section 3.6.4[10]. For this experiment, the algorithm is initialized with $N_{init} = 30$ experiments and run over 100 iterations. For the computation of the EI criterion, we use the L_2^{opt} density (see Section 3.3) with $m_{\mathbb{Y}} = 1000$ particles. To set $\mathbb{B}_o$, we use the procedure detailed in Section 3.6.2 with $\gamma = 0$ and $\gamma = 1$. For the optimization of the criterion, we use the PICPI density (see Section 3.2) with $m_{\mathbb{X}} = 1000$ particles. For every value of c, the experiment is repeated 30 times with different random seeds to account for the randomness of the algorithm.

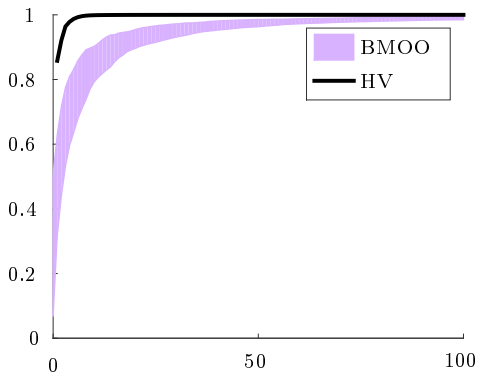
BMOO achieves satisfying results on the FICUS problem. In particular, the results for the M_3 metric are close to those of the strategy where the hypervolume is maximized sequentially, which indicates that BMOO is able to find solutions close to the Pareto front and well distributed. However, BMOO is not able to capture the extreme points of the front on this problem, as can be observed through the M_2 metric. This observation holds for both values of the parameter γ considered in this study. Therefore it does not seem related to the choice of $\mathbb{B}_o$. Regarding the convergence with respect to the M_1 metric, observe that about 100 iterations are necessary for $p = 6$ and about 200 for $p = 8$. The convergence is much slower than the reference strategy and this is partly a consequence of the extreme solutions not being captured by the algorithm. Our belief is that the SMC procedure that we use for optimizing the criterion is not proposing candidates near the boundaries of the domain, where the extreme solutions can be found. It could also be due to imprecisions of the models in those regions.

3.4.5 Conclusions

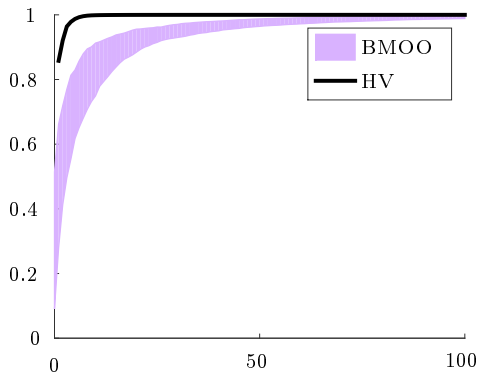
The gist of this section is to evaluate the performances of BMOO on many-objective problems. In particular, we are interested in the quality of the Pareto approximation sets obtained by the algorithm. To this purpose, we introduce the FICUS problem. This problem is a configurable test problem for which it is possible to control the number of objectives functions and the curvature of the Pareto front. Moreover, a closed form expression of the hypervolume dominated by its Pareto front is known, and it is possible to draw samples on it with a simple sampling procedure.

First, the FICUS problem is used to empirically study the characteristics of distributions of Pareto-optimal solutions obtained by sequential maximization of the hypervolume indicator.

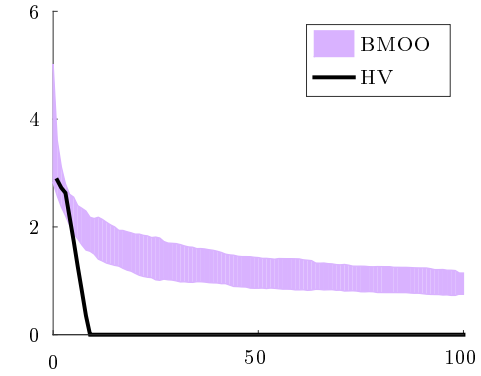
[10] We do not provide results for $p = 10$ because BMOO requires about 300 functions evaluations and the computing time becomes prohibitive for the M_1 metric and for the strategy where the hypervolume is maximized sequentially (see Figure 3.8).



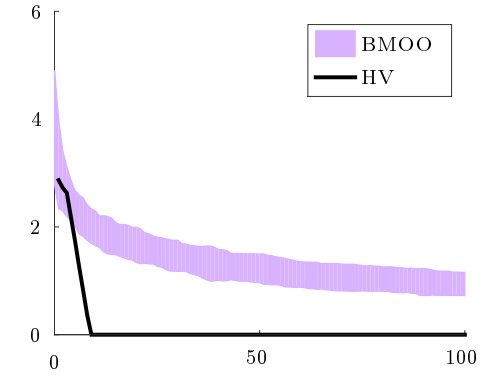
(a) M_1 , $\gamma = 0$



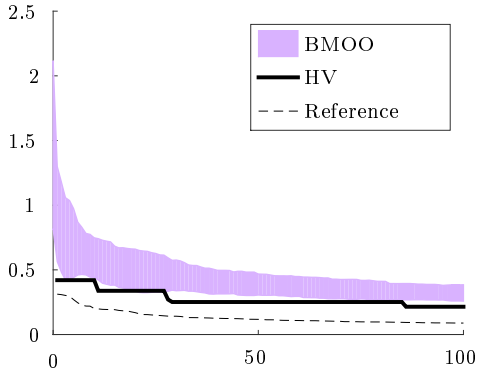
(b) M_1 , $\gamma = 1$



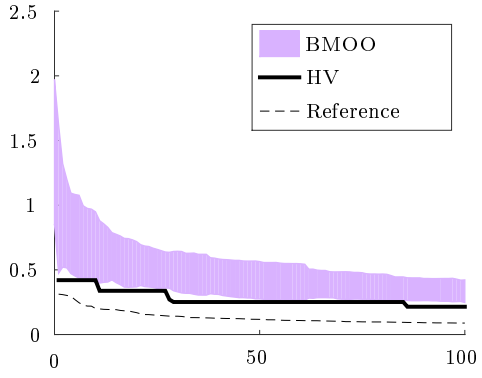
(c) M_2 , $\gamma = 0$



(d) M_2 , $\gamma = 1$

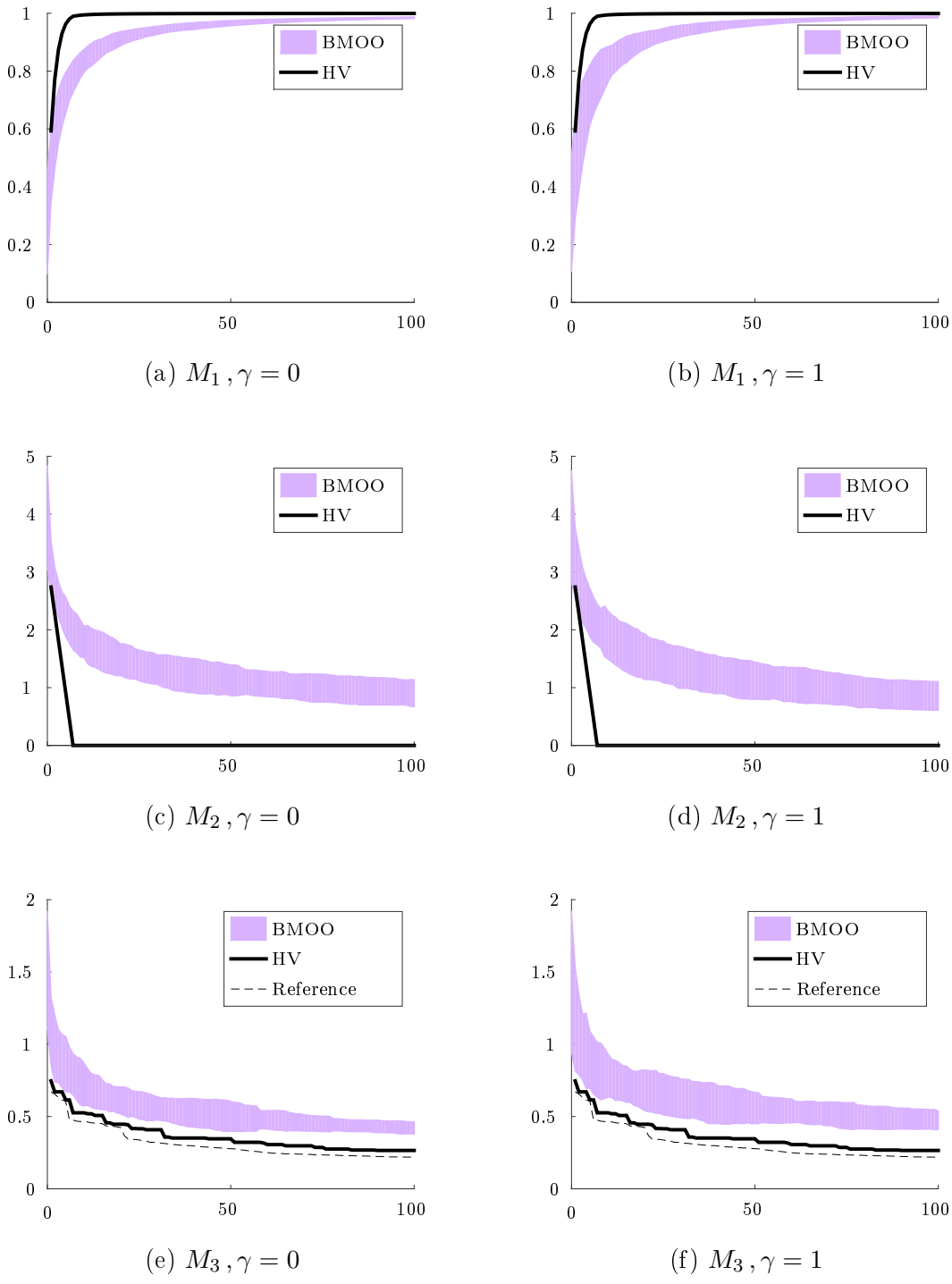


(e) M_3 , $\gamma = 0$



(f) M_3 , $\gamma = 1$

Figure 3.16: Results obtained by the BMOO algorithm on the FICUS $(6, 0.5, 0.6)$ problem (convex front) when the procedure of Section 3.6.2 to set $\mathbb{B}_o$ is used with $\gamma = 0$ (left column) or $\gamma = 1$ (right column). The reference for the M_3 metric is shown as a dashed line and the shaded region corresponds to a 95% confidence interval empirically computed from 30 runs of BMOO.

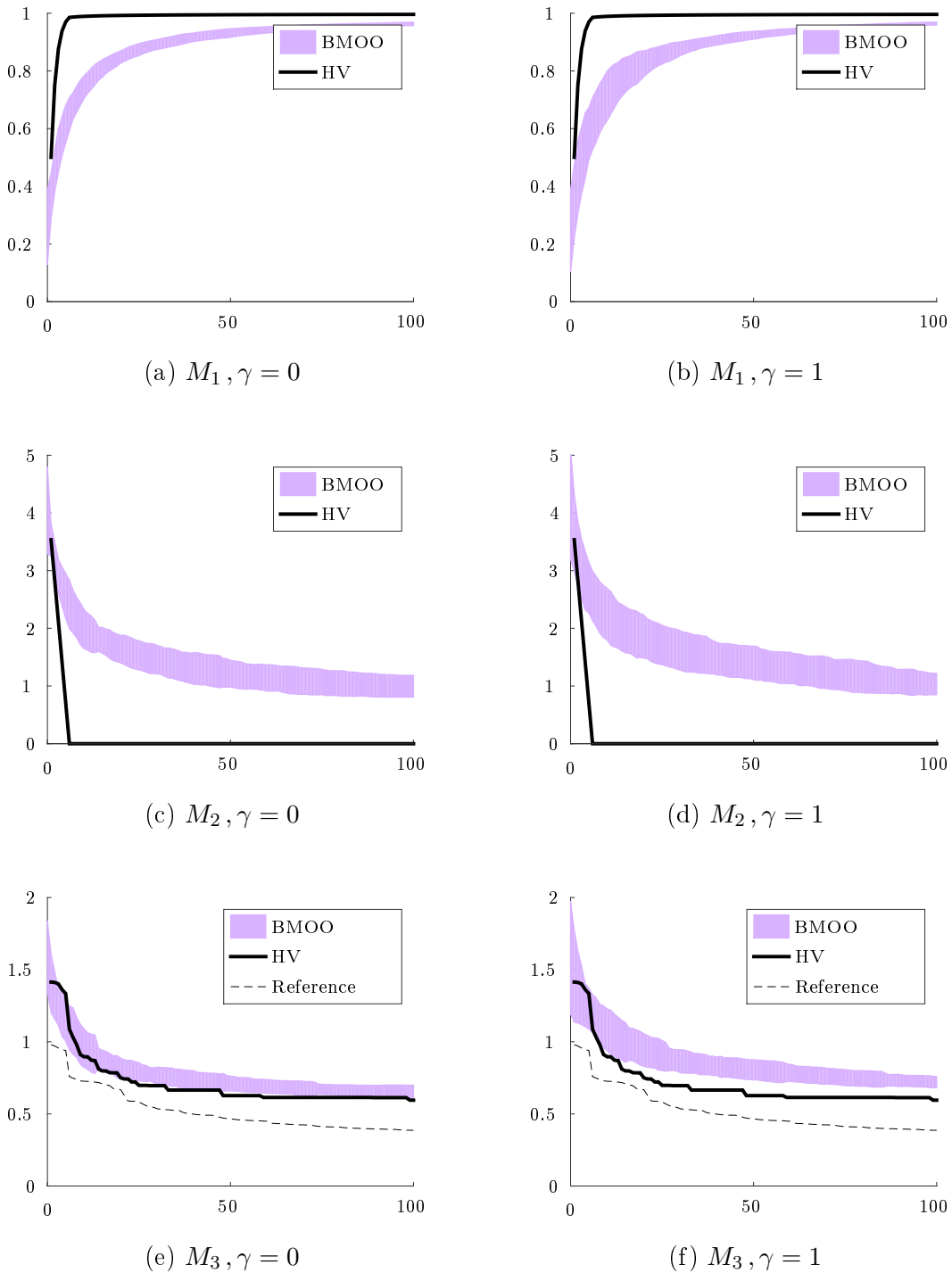


(a) M_1 , $\gamma = 0$ (b) M_1 , $\gamma = 1$

(c) M_2 , $\gamma = 0$ (d) M_2 , $\gamma = 1$

(e) M_3 , $\gamma = 0$ (f) M_3 , $\gamma = 1$

Figure 3.17: Results obtained by the BMOO algorithm on the FICUS $(6, 0.5, 1)$ problem (linear front) when the procedure of Section 3.6.2 to set $\mathbb{B}_o$ is used with $\gamma = 0$ (left column) or $\gamma = 1$ (right column). The reference for the M_3 metric is shown as a dashed line and the shaded region corresponds to a 95% confidence interval empirically computed from 30 runs of BMOO.



(a) M_1 , $\gamma = 0$

(b) M_1 , $\gamma = 1$

(c) M_2 , $\gamma = 0$

(d) M_2 , $\gamma = 1$

(e) M_3 , $\gamma = 0$

(f) M_3 , $\gamma = 1$

Figure 3.18: Results obtained by the BMOO algorithm on the FICUS $(6, 0.5, 2)$ problem (concave front) when the procedure of Section 3.6.2 to set $\mathbb{B}_o$ is used with $\gamma = 0$ (left column) or $\gamma = 1$ (right column). The reference for the M_3 metric is shown as a dashed line and the shaded region corresponds to a 95% confidence interval empirically computed from 30 runs of BMOO.

Three evaluation metrics are considered. They measure respectively the convergence with respect to the hypervolume indicator, the membership of the extreme solutions of the Pareto front to a Pareto approximation set and the presence of "holes" in a Pareto approximation set distribution. It is shown that for problems with convex and linear fronts, satisfactory distributions are obtained when the hypervolume indicator is maximized sequentially. For problems with three objectives and a concave front, it is observed that maximizing the hypervolume does not yield good Pareto approximation sets. Our experiments do not permit to conclude that this behaviour also exists in higher dimensions.

As regards the choice of the reference point, it is shown that it mostly influences the representation of extreme solutions in the Pareto approximation set. When it is chosen as the nadir point, these are not included in the approximation set. When it is chosen at a "reasonable" distance from the Pareto front, the extreme solutions are the first points selected by the hypervolume. When it is chosen far from the Pareto front, the extreme solutions are eventually selected by the hypervolume indicator but not necessarily at the begining of the optimization process.

The results obtained by sequential maximization of the hypervolume indicator are then used as a reference to evaluate the performances of the BMOO algorithm on the FICUS problem. Indeed, this procedure can be viewed as an idealized version of BMOO where the GP models are perfect, the criterion is computed exactly and its optimization is made almost perfectly. It is shown that for the FICUS problem with up to eight objectives, BMOO achieves results relatively close to this ideal reference. Its convergence with respect to the hypervolume indicator is slower but effective and the distributions of the solutions found by the algorithm seem close to those obtained by the reference (as measured by the size of the "holes" in the distributions). However, BMOO is not able to correctly capture the extreme solutions for the FICUS problems. It could be due to imprecisions of the models near the boundaries of the domain or it could be that the SMC procedure used for optimizing the criterion fails to propose candidates near the boundaries of the domain. More work is required to better understand this phenomenom.

3.5 Extensions of the BMOO algorithm

3.5.1 Introduction

In this section, we propose extensions of the BMOO algorithm that are necessary to address the problems of Chapter 4, along with other useful extensions which come as a by-product of the algorithm. The section is organized as follows. In Section 3.5.2, we extend BMOO to address problems defined on non-hypercubic design spaces and to problems having cheap-to-evaluate constraints. In Section 3.5.3 we study the case where f and/or c cannot be computed for some $x \in \mathbb{X}$. This happens for example when a computer program fails to return a result for some combinations of the design parameters. Then we propose a multi-point version of the algorithm in Section 3.5.4. Finally, in Section 3.5.5 we propose an extension of the EHVI criterion that makes it possible to encode user preferences into the sampling criterion.

Throughout the section, the proposed extensions are illustrated on simple test problems; see Chapter 4 for applications to real-life problems.

3.5.2 Non-hypercubic design spaces

In this section we extend the BMOO algorithm to address the following problem:

$$\begin{array}{ll} \text{minimize} & f(x), \\ \text{subject to} & x \in \mathbb{X} = \{x \in \mathbb{H}\,;\, \mathbb{1}_{\mathcal{S}}(x) = 1 \text{ and } \ e(x) \leq 0\}\,, \\ & c(x) \leq 0, \end{array} \tag{3.21}$$

where $f = (f_1, \ldots, f_p)$ and $c = (c_1, \ldots, c_q)$ are vectors of expensive to evaluate objectives and constraints as in Section 2.1, $\mathbb{H} \subset \mathbb{R}^d$ is an hypercube defined by bound constraints, $e = (e_1, \ldots, e_s)$ is a vector of real valued cheap-to-evaluate[11] constraint functions defined on $\mathbb{H}$ ($e_i : \mathbb{H} \to \mathbb{R}$) and $\mathcal{S}$ is a subset of $\mathbb{H}$ which is not given explicitly, in the sense that the membership of x to $\mathcal{S}$ can only be assessed by means of a membership function $\mathbb{1}_{\mathcal{S}}$. In the following, we assume that the membership function is also cheap to evaluate. The case where $\mathbb{1}_{\mathcal{S}}$ is expensive to evaluate will be addressed in Section 3.5.3 with a different approach. An example of such problem can be found in Section 4.2.

In this work, we propose to view the constraints $(e_i)_{1 \leq i \leq s}$ and the membership function $\mathbb{1}_{\mathcal{S}}$ as restrictions of the design space. In this setting, the objectives $(f_i)_{1 \leq i \leq p}$ and the constraints $(c_j)_{1 \leq j \leq q}$ may indifferently be or not be observable outside of $\mathbb{X}$. In practice, this requires two adaptations of the BMOO algorithm. First, a dedicated procedure is required to build an initial design of experiment (DOE) on $\mathbb{X}$. Secondly, it is necessary to prevent the algorithm from going outside of $\mathbb{X}$ in the subsequent iterations.

Usually, the initial DOE is chosen to be both space filling and well spread in all dimensions. So far, we have been using pseudo-maximin Latin hypercube designs because they achieve a good trade-off between these two properties and the computing time required to chose the points of the DOE. However, it is not possible in the general case to build a Latin hypercube design on a non-hypercubic design space. Besides, this would not be desirable anyway. As an example[12], consider the case where $\mathbb{X} = \left\{(x_1, x_2) \in [0,1]^2\,;\, x_1 - x_2 \leq 0\right\}$. Then the only possible Latin hypercube design is a set of points falling close to the diagonal $x_1 = x_2$, which, of course, is not desirable because it is not a space filling design.

In the literature, maximin designs on non-hypercubic sets have been studied by e.g. Stinstra et al. (2003); Auffray et al. (2012) and Chen et al. (2014). The methods that have been proposed usually involve complex optimization steps and developing such methods falls out of the scope of the present work.

We propose a simpler approach based on a three-step procedure. The first step consists in building a large sample uniformly distributed on $\{x \in \mathbb{H}\,;\, e(x) \leq 0\}$. This can be achieved using SMC by considering a sequence of transition densities of the form $\mathbb{1}_{\{x \in \mathbb{H}\,;\, e_i(x) \leq \alpha, 1 \leq i \leq s\}}$, where $\alpha \in \mathbb{R}^+$ is chosen sequentially in order to keep a significant population of particles between two iterations (this is very similar to Algorithm 7, that is used to make transitions with the PICPI

[11] Typically, we consider a function to be cheap to evaluate if its evaluation takes in the order of the millisecond, or less. More generally, a function may be considered as cheap to evaluate when no restriction is placed on the number of affordable evaluations of this function.

[12] This example is taken from Auffray et al. (2012).

density). Then, the population is pruned to keep only particles falling in $\mathcal{S}$. The resulting sample is uniformly distributed on $\mathbb{X} = \{x \in \mathbb{H}\,;\, e(x) \leq 0, \mathbb{1}_{\mathcal{S}}(x) = 1\}$. Note that for this procedure to be effective, it is necessary that the volume of $\mathbb{X}$ be not too small compared to the volume of $\{x \in \mathbb{H}\,;\, e(x) \leq 0\}$. Finally, the population is pruned again by sequentially removing the samples that are closest to their neighbours until the required sample size is reached. The resulting set is a pseudo-maximin DOE on the domain $\mathbb{X}$.

Staying within the limits of the $\mathbb{X}$ domain is straightforward when the density $\pi_n^{\mathbb{X}}$ is defined on $\mathbb{X}$ because then, the sequential Monte-Carlo procedure used for optimizing the EI criterion never proposes candidates outside of $\mathbb{X}$. In practice though, $\mathbb{X}$ is not known explicitly and membership to $\mathbb{X}$ has to be checked within the SMC algorithm[13]. Besides, an initialization procedure starting from a sample uniformly distributed on $\mathbb{H}$ is required to construct $\mathcal{X}_0$ from $\pi_0^{\mathbb{X}}$ defined on $\mathbb{X}$. As mentioned above, it is possible to build a sample (of size $m_{\mathbb{X}}$) uniformly distributed on $\{x \in \mathbb{H}\,;\, e(x) \leq 0\}$. Then, Algorithm 6 can be used to build $\mathcal{X}_0$ from $\pi_0^{\mathbb{X}}$. Note that, again, for this procedure to be effective, it is necessary that the volume of $\mathbb{X}$ be not too small compared to the volume of $\{x \in \mathbb{H}\,;\, e(x) \leq 0\}$.

To illustrate the proposed approach, we consider an unconstrained single-objective optimization problem formulated as (3.21) with

$$\begin{cases} \mathbb{H} = [-5, 10] \times [0, 15], \\ \mathcal{S} = \{(x_1, x_2) \in \mathbb{H}\,;\, x_1^2 + (x_2 - 15)^2 \leq 25\}, \\ f : (x_1, x_2) \mapsto \left(x_2 - \dfrac{5.1}{4\pi^2}x_1^2 + \dfrac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \dfrac{1}{8\pi}\right)\cos(x_1) + 9, \\ e : (x_1, x_2) \mapsto x_1 - x_2. \end{cases} \tag{3.22}$$

The features of this problem and the operation of the BMOO algorithm are illustrated respectively in Figures 3.19 and 3.20. For this experiment, the initial DOE is a pseudo-maximim design of 10 experiments chosen using the three-step procedure detailed above (Iteration 0 in Figure 3.20) and the algorithm is run over 10 iterations.

3.5.3 Hidden constraints management

We now turn to the case where the membership function $\mathbb{1}_{\mathcal{S}}$ is expensive to evaluate. This happens, for example, when a time-consuming computer program is used to evaluate the objectives and/or the constraints, and fails to return a result for some combinations of the design parameters which are not known beforehand.

When the objectives and constraints are modelled using random processes as it is the case for BMOO, regions that can not be observed are problematic because the variance of the processes can not be decreased in those regions. Therefore, if nothing is done, the EI criterion is likely to become significant in those regions at some point during the optimization process. In the Bayesian literature, this problem has been addressed in the unconstrained single-objective setting

[13] Note that since both e and $\mathbb{1}_{\mathcal{S}}$ are cheap to evaluate, this is not an issue to check the membership of the particles to $\mathbb{X}$ several times within an SMC algorithm.

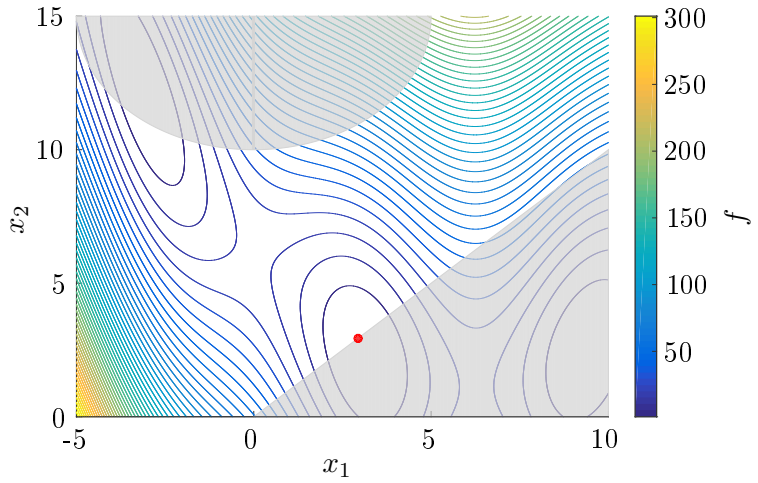


Figure 3.19: Illustration of the features of the optimization problem defined by (3.21) and (3.22). The contours of the objective function are shown as colored lines and the set $\mathbb{H} \setminus \mathbb{X}$ is shaded. The optimum of the objective function on $\mathbb{X}$ is shown as a red disk.

as a missing data problem by Roy (2006) and as a problem having hidden constraints by Lee et al. (2011).

In Roy (2006), it is proposed to impute well-chosen values at missing data locations to artificially decrease the variance of the processes and thus prevent the algorithm to return in the regions that can not be observed. In our experiments, we found this approach difficult to extend to problems having possibly several objectives and several constraints.

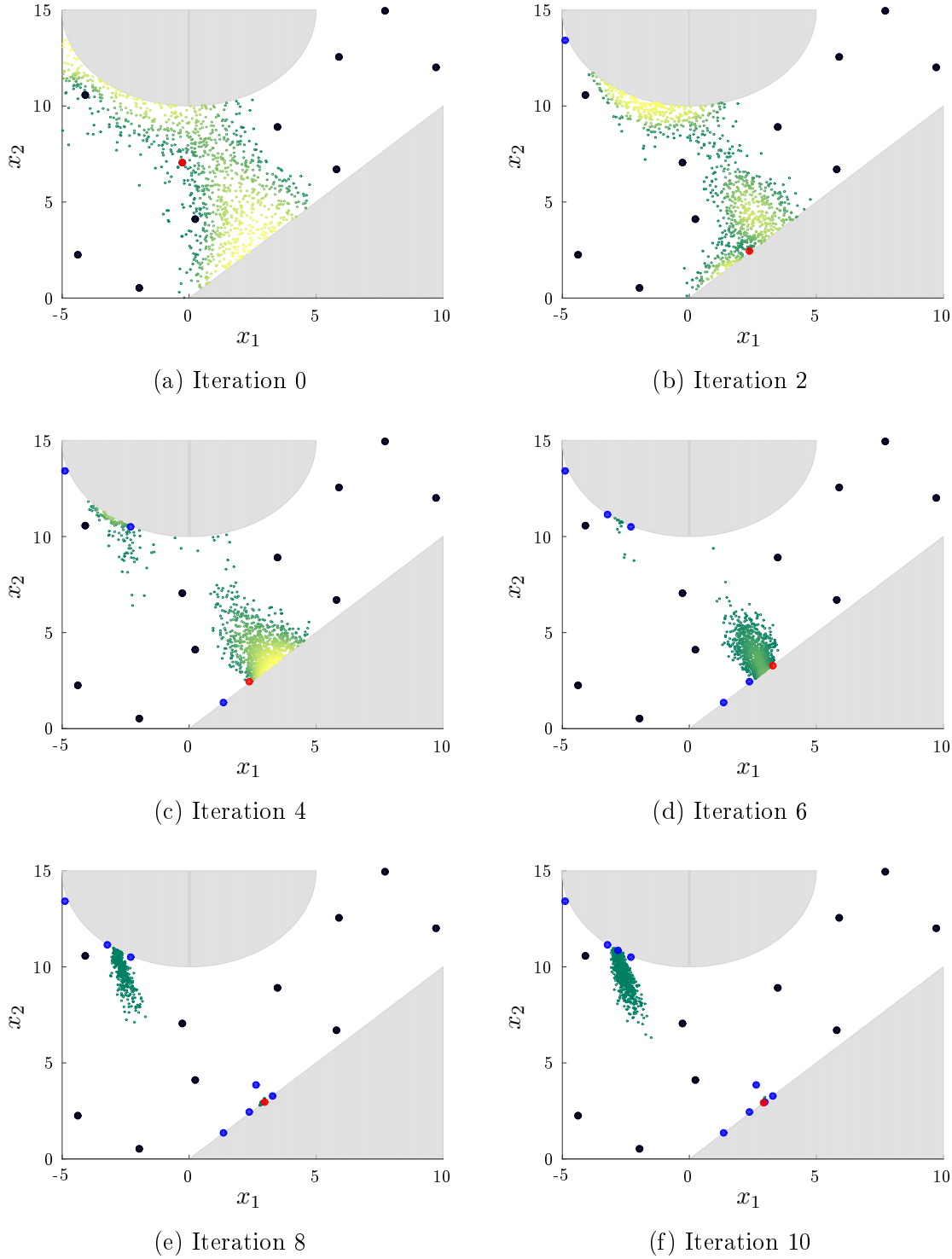
Lee et al. (2011) propose a different approach based on a statistical model: A soft classifier of the observed/non-observed data is used to produce for every point of the design space $x \in \mathbb{X}$ a "probability of observability" $p_n(x \in \mathcal{S})$. The authors then propose to multiply the expected improvement by this probability to prevent it from rewarding regions that can not be observed:

$$\tilde{\rho}_n(x) = \rho_n(x) \cdot p_n(x \in \mathcal{S})\,. \tag{3.23}$$

In their work, a Random Forest classifier is used but the approach can be extended to other kinds of soft classifiers such as support vector machines, nearest-neighbours classifiers, neural networks, etc.

The adaptation of this approach to the constrained multi-objective case is straightforward. In our context, it corresponds to multiplying both the EI function and the density $\pi_n^{\mathbb{X}}$ used to optimize the EI by p_n. In this work, we use a simple k-nearest-neighbours classifier to produce this probability. Observed samples are given the label one and non-observed samples are given the label zero. Then, the probability of observability at a location $x \in \{x' \in \mathbb{H}\,;\, e(x') \leq 0\}$ is computed as the mean of the labels of the k nearest neighbours of x, as given by the L_2 norm. This model has the advantage of simplicity and does not require any assumption on the shape of the set $\mathcal{S}$. Besides, there was no evidence in our experiments that a more elaborate statistical model is required.

An illustration of the operation of this procedure is provided in Figure 3.21 for the optimiza-



(a) Iteration 0

(b) Iteration 2

(c) Iteration 4

(d) Iteration 6

(e) Iteration 8

(f) Iteration 10

Figure 3.20: Illustration of a run of BMOO on the optimization problem defined by (3.22) and (3.21). Black disks represent the initial DOE and blue disks represent subsequent evaluation points. The current best evaluation point in shown as a red disk. See Figure 3.19 for more details.

tion problem defined by (3.21) with

$$\begin{cases} \mathbb{H} = [-5, 10] \times [0, 15]\,, \\ \mathcal{S} = \{(x_1, x_2) \in \mathbb{H}\,;\, x_1^2 + (x_2 - 15)^2 \leq 25\,,\, x_2 - x_1 \leq 0\}\,, \\ f : (x_1, x_2) \mapsto \left(x_2 - \dfrac{5.1}{4\pi^2} x_1^2 + \dfrac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \dfrac{1}{8\pi}\right)\cos(x_1) + 9\,. \end{cases} \tag{3.24}$$

For this experiment, we take $k = 5$, the initial DOE is a pseudo maximin set of 20 experiments, and the algorithm is run over 20 iterations. Note that unlike this test problem, it is unlikely in practice that the optimum lies at the boundary of the non-observable domain.

3.5.4 Batch sequential multi-objective optimization

Throughout this manuscript, we have been using the number of function evaluations as a measure of the cost of an optimization process. However, when parallel computation facilities offer the possibility of evaluating several design solutions simultaneously, i.e. within the same time lap, it can be very advantageous to take advantage of it to lower this cost.

In this section, we propose a batch version of the algorithm that relies on a multi-points version of the EI criterion inspired from the q-EI criterion of Ginsbourger et al. (2010b). The problem consists in selecting at each iteration of the algorithm a batch of q points to be evaluated simultaneously. We consider a synchronous approach where the evaluations of the q selected points are made in parallel and take approximately the same time (see Remark 26).

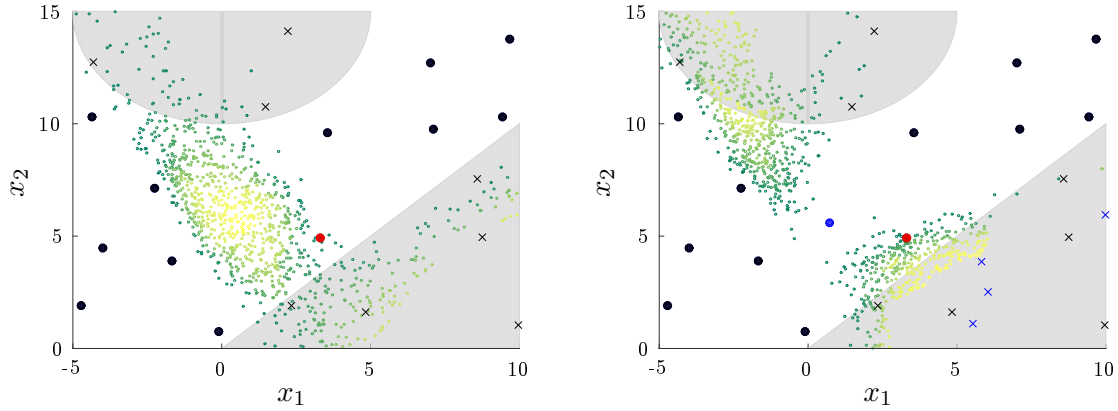
Denote $\boldsymbol{X}_n = (X_1, \ldots, X_n)$ the set of observations available at time $n \geq 1$ and $\boldsymbol{Y}_n = \xi(\boldsymbol{X}_n)$. In the unconstrained single-objective context, Ginsbourger et al. (2010b) define the q-EI criterion as

$$\rho_{q,n}(x_1, \ldots, x_q) = \mathbb{E}_n\left(\left(m_n - \min\left(\xi(x_1), \ldots, \xi(x_q)\right)\right)_+\right), \tag{3.25}$$

where $m_n = \xi(X_1) \wedge \cdots \wedge \xi(X_n)$ is the current best solution as in Section 2.2.1 and $\mathbb{E}_n$ stands for the expectation conditional on the information $(\boldsymbol{X}_n, \boldsymbol{Y}_n)$ available at time n. The strategy to choose the batch of experiments then naturally consists in selecting the q points that maximize the q-EI criterion.

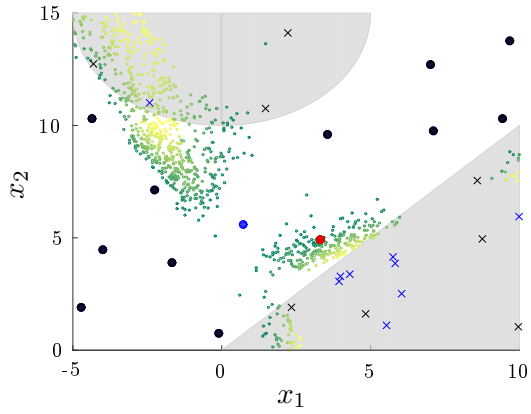
$$(X_{n+1}, \ldots, X_{n+q}) = \operatorname{argmax}_{(x_1,\ldots,x_q)\in\mathbb{X}^q} \rho_{q,n}(x_1, \ldots, x_q). \tag{3.26}$$

The criterion (3.25) is appealing because it corresponds to the Bayes-optimal one-step lookahead strategy for the loss function (2.2). However, it is difficult to use in practice for two reasons. First, its computation is not simple. A closed form formula is provided by Chevalier and Ginsbourger (2013) but it becomes computationally intensive for large values of q and is therefore limited to small batch sizes (say $q < 10$). For larger batch sizes, approximate computation procedures have been proposed by Ginsbourger et al. (2010b) and Marmin et al. (2016) and can be used with a moderate computational cost. Secondly, solving the auxiliary optimization problem (3.26) is challenging because the size of the search space is $d \times q$, which can be quite large even for moderate batch sizes when d is already rather large. To solve this problem, Frazier

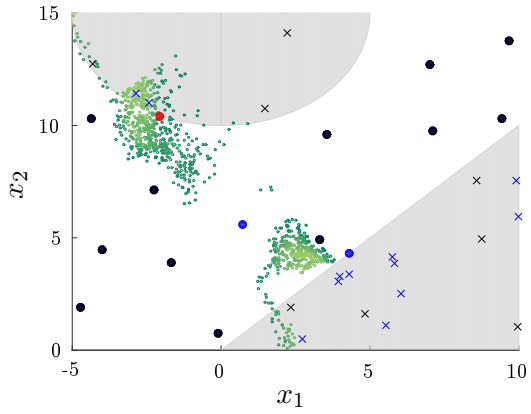


(a) Iteration 0

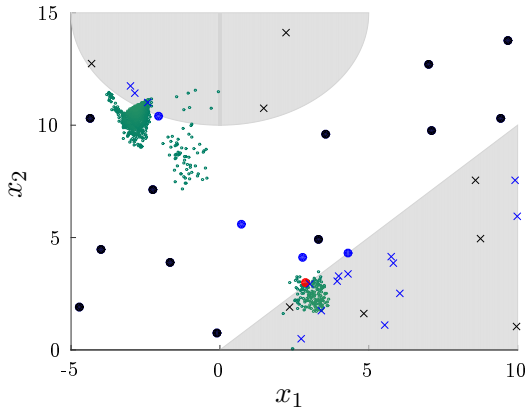
(b) Iteration 5



(c) Iteration 10



(d) Iteration 15



(e) Iteration 20

Figure 3.21: Illustration of a run of BMOO on the optimization problem defined by (3.21) and (3.24). Black disks represent the initial DOE and blue disks represent subsequent evaluation points. The current best evaluation point in shown as a red disk. The non-observable domain is shaded in gray and failed evaluations are shown as crosses.

and Clark (2012); Wang et al. (2016a) and Marmin et al. (2016) propose multi-start stochastic gradient algorithms relying on approximate gradient computation methods. Note that a closed form expression of the gradient of the q-EI criterion has been developed by Marmin et al. (2015) but again, the approach is limited to small batch sizes.

To circumvent these difficulties, Ginsbourger et al. (2010b) propose to use a sequential approximation of (3.26) where the X_{n+i}, $i \in [\![1, q]\!]$, are chosen one after the other using the following procedure.

$$\left\{ \begin{array}{rcl} X_{n+1} & = & \underset{x \in \mathbb{X}}{\operatorname{argmax}}\, \mathbb{E}_n \left((m_n - \xi(x))_+ \right), \\ X_{n+i} & = & \underset{x \in \mathbb{X}}{\operatorname{argmax}}\, \mathbb{E}_n \left((\min(m_n, \xi(X_{n+1}), \ldots, \xi(X_{n+i-1})) - \xi(x))_+ \,|\, X_{n+1}, \right. \\ & & \left. \qquad \ldots, X_{n+i-1} \right), \quad 2 \leq i \leq q. \end{array} \right. \tag{3.27}$$

Observing that (3.27) still poses computational difficulties, the authors further simplify the procedure as

$$\left\{ \begin{array}{rcl} X_{n+1} & = & \underset{x \in \mathbb{X}}{\operatorname{argmax}}\, \mathbb{E}_n \left((m_n - \xi(x))_+ \right), \\ X_{n+i} & = & \underset{x \in \mathbb{X}}{\operatorname{argmax}}\, \mathbb{E}_n \left((\min(m_n, \xi(X_{n+1}), \ldots, \xi(X_{n+i-1})) - \xi(x))_+ \,|\, \xi(X_{n+1}) = y_1, \right. \\ & & \left. \qquad \ldots, \xi(X_{n+i-1}) = y_{i-1} \right), \quad 2 \leq i \leq q, \end{array} \right. \tag{3.28}$$

where the $(y_i)_{1 \leq i \leq q} \in \mathbb{R}^q$ are values imputed to the successive $(\xi(X_{n+i}))_{1 \leq i \leq q}$. The authors then propose two strategies for setting the $(y_i)_{1 \leq i \leq q}$. The Constant Liar (CL) strategy consists in lying with the same value $y_i = y_n \in \mathbb{R}$, $i \in [\![1, q]\!]$, each time. For examples they study the cases where $y_n = \min(\boldsymbol{Y}_n)$, $y_n = \text{mean}(\boldsymbol{Y}_n)$ and $y_n = \max(\boldsymbol{Y}_n)$. The Kriging Believer (KB) strategy consists in setting y_i as the kriging mean at X_{n+i}: $y_i = \widehat{\xi}(X_{n+i})$, $i \in [\![1, q]\!]$. Note that with this strategy, the computation problem reduces to the computation of the EI function and that the optimization problem reduces to q optimizations in dimension d.

To the best of our knowledge, the problem of selecting batches of experiments in a Bayesian multi-objective context has only been addressed by Zhang et al. (2010). This is somewhat surprising because, unlike the single-objective case where a single best point will eventually emerge from the observation of q new points, in a multi-objective context, the newly observed points can potentially all simultaneously improve upon the current front of non-dominated observations if they do not dominate each other. Hence, it makes a lot of sense to consider batch strategies in a multi-objective context and it seems plausible that a good batch strategy could outperform the sequential strategy in terms of number of iterations, which is not so evident in the single-objective case.

A natural extension of (3.25) to a constrained multi-objective setting where the improvement is measured using the hypervolume measure defined using the domination rule (2.21) is to consider the following criterion.

$$\rho_{q,n}(x_1, \ldots, x_q) = \mathbb{E}_n \left((|H((\boldsymbol{Y}_n, \xi(x_1), \ldots, \xi(x_q)))| - |H_n|)_+ \right), \tag{3.29}$$

Algorithm 8: Sequential procedure to select a batch of q experiments for evaluation.

$i \leftarrow 1$
$X_{n+1} \leftarrow \text{argmax}_{x \in \mathcal{X}_n} \rho_n(x)$
while $i < q$ **do**
  $i \leftarrow i + 1$
  Generate $(Y_1, \ldots, Y_{i-1})$ from $(\xi(X_{n+1}), \ldots, \xi(X_{n+i-1}))$ conditional on $\boldsymbol{Y}_n = (f(\boldsymbol{X}_n), c(\boldsymbol{X}_n))$.
  $\boldsymbol{Y}_{n,i} \leftarrow (\boldsymbol{Y}_n, Y_1, \ldots Y_{i-1})$
  $\boldsymbol{X}_{n,i} \leftarrow (\boldsymbol{X}_n, X_{n+1}, \ldots X_{n+i-1})$
  $X_{n+i} \leftarrow \text{argmax}_{x \in \mathcal{X}_n} \mathbb{E}\left((|H((\boldsymbol{Y}_{n,i}, \xi(x)))| - |H(Y_{n,i})|)_+ | \xi(\boldsymbol{X}_{n,i}) = \boldsymbol{Y}_{n,i} \right)$

where $H(\boldsymbol{Y}) = \{z \in \mathbb{B}; \exists y \in \boldsymbol{Y} \text{ s.t. } y \prec z\}$ is the subset of $\mathbb{B}$ made of the points dominated by the elements of a vector $\boldsymbol{Y}$ of elements of $\mathbb{B}$. Observe in particular that for $q = 1$, the criterion (3.29) corresponds to the expected improvement (2.23).

As in the single-objective case, the computation and optimization of the criterion (3.29) are difficult problems and an in depth study of these two aspects falls out of the scope of this thesis work. We limit ourselves to proposing a sequential procedure in the spirit of the KB strategy of Ginsbourger et al. (2010b). The proposed procedure is summarized in Algorithm 8. Note that unlike the KB strategy, in the proposed algorithm, we use conditional realisations of ξ to produce the $(y_i)_{1 \leq i \leq q}$ that are imputed to the model. As such, (3.28) is an unbiased estimator of (3.27). In Algorithm 8, a single realisation of ξ is used at each iteration but of course, it it is possible to use more draws to improve the robustness of the approach.

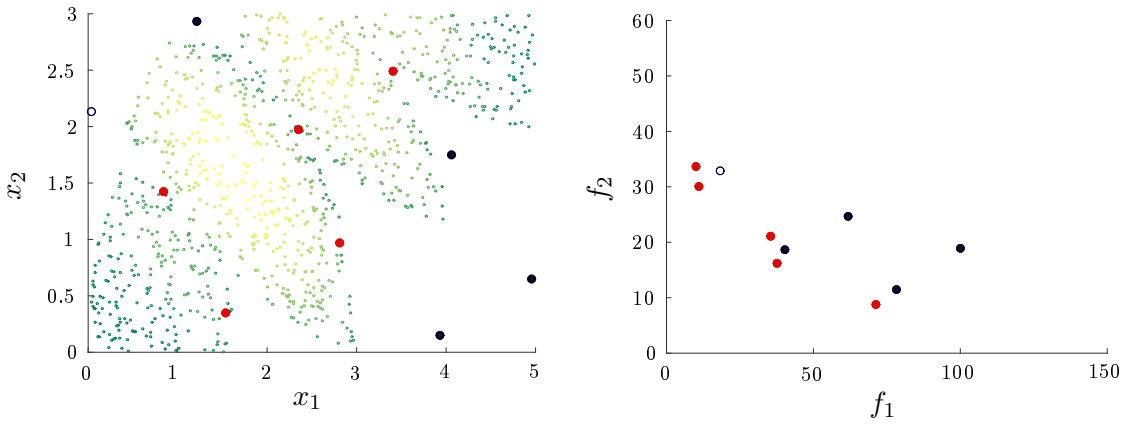
In Figures 3.22 and 3.23, we show an illustration of one run of the BMOO algorithm on the BNH problem introduced in Section 2.5.4. Batches of $q = 10$ experiments selected using Algorithm 8 are evaluated simultaneously at each iteration. The algorithm is initialized using a pseudo-maximin latin hypercube design of 10 experiments (Iteration 0 of Figure 3.22) and is run over 5 iterations. Note how most of the points in the batches contribute to the Pareto front discovery. For comparison, the same experiment but with $q = 25$ is shown in Figure 3.24.

Remark 25 *See Azimi et al. (2010); Viana and Haftka (2010); Zhang et al. (2010); Desautels et al. (2014); Chevalier et al. (2014b); Shah and Ghahramani (2015); González et al. (2015); Guerra (2016); Wu and Frazier (2016); Kathuria et al. (2016); Habib et al. (2016); Li et al. (2016b) and Daxberger and Low (2017) for alternative approaches to select batches of experiments not directly related to the concept of expected improvement.*

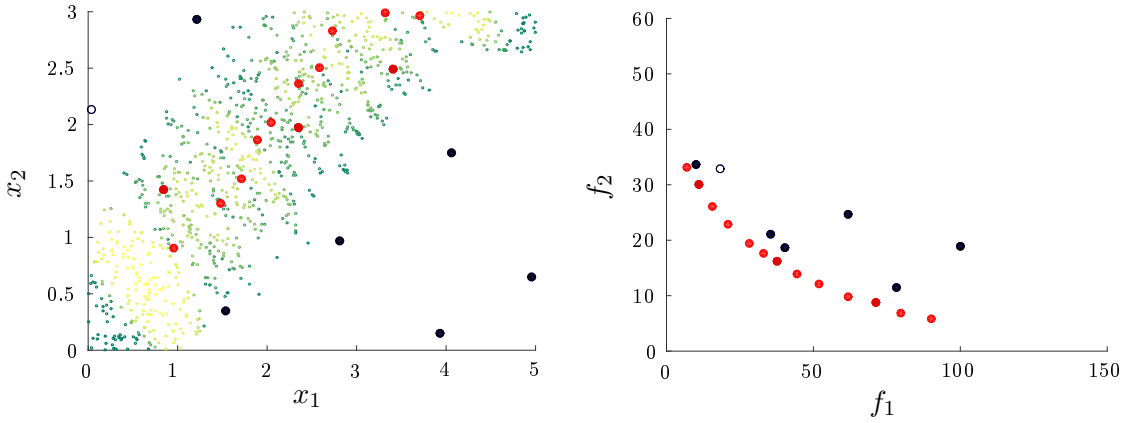
Remark 26 *In this work, we consider synchronous batches of experiments. See the works of Ginsbourger et al. (2010a); Janusevskis et al. (2012); Girdziusas et al. (2012) and Le Riche et al. (2012) for asynchronous batch selection approaches.*

3.5.5 User preferences in multi-objective optimization

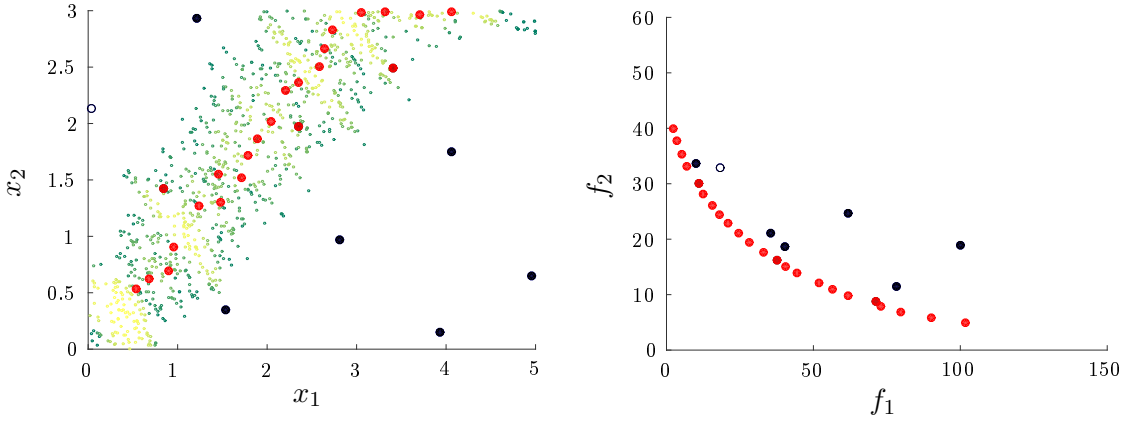
Sometimes, the end-user is able to say in advance what part of the Pareto front is more interesting for his particular application. For example, he might be interested more in extreme solutions



(a) Iteration 0 (b) Iteration 0

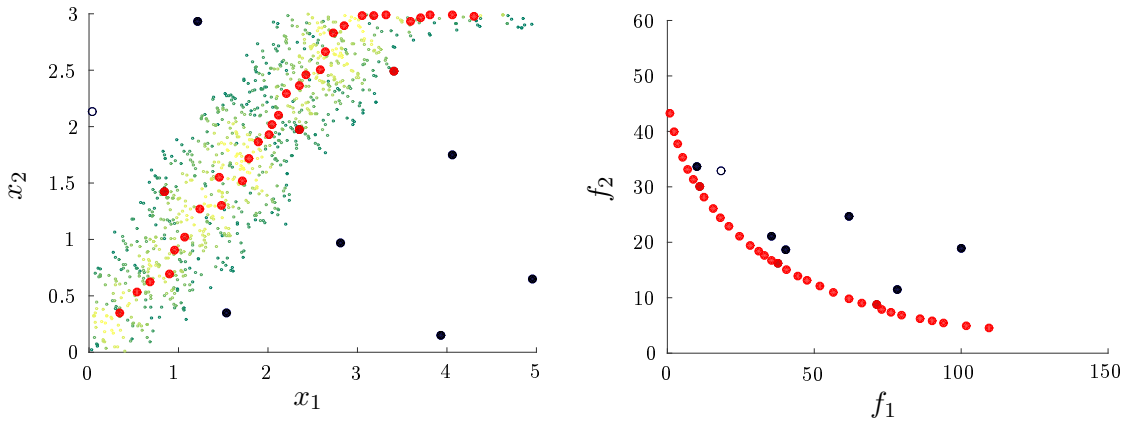


(c) Iteration 1 (d) Iteration 1

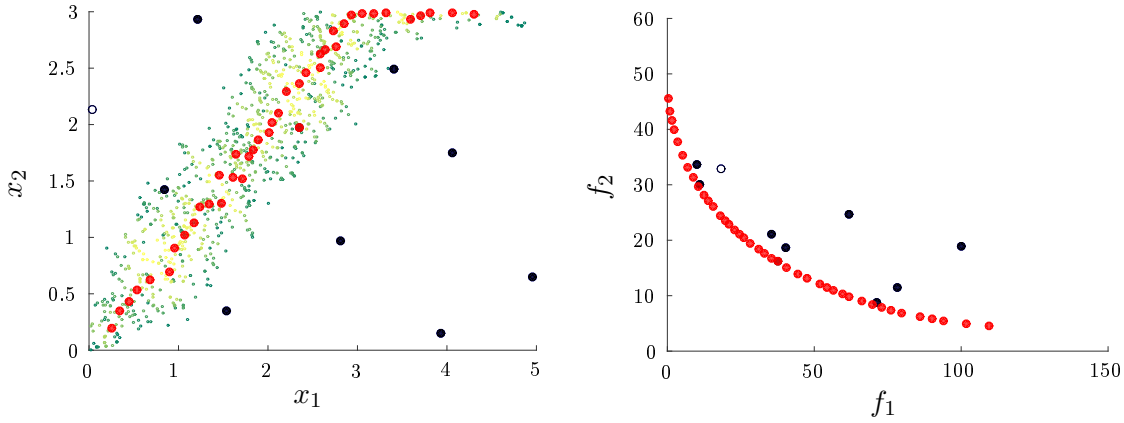


(e) Iteration 2 (f) Iteration 2

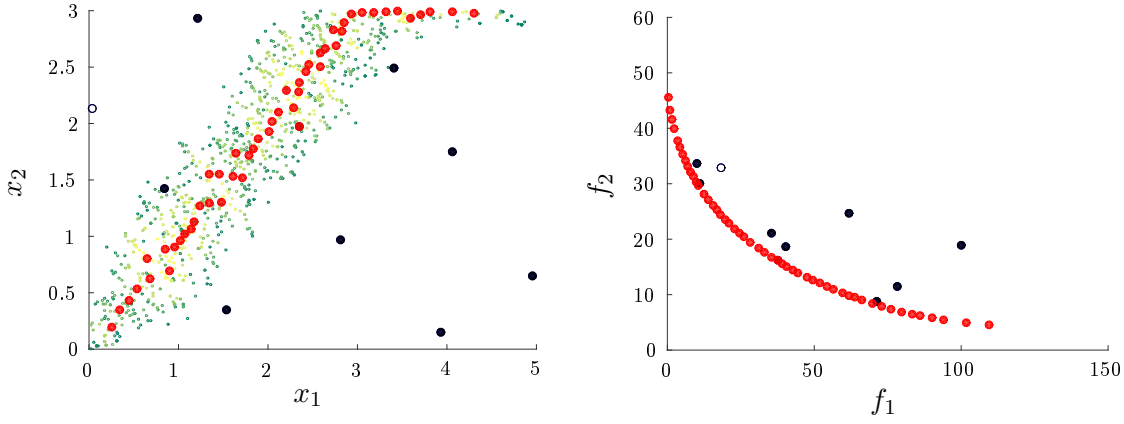
Figure 3.22: Illustration of one run of the BMOO algorithm on the BNH problem. Batches of $q = 10$ experiments are made at each iteration. Non-dominated solutions are shown as red disks and dominated ones as black disks.



(a) Iteration 3 (b) Iteration 3

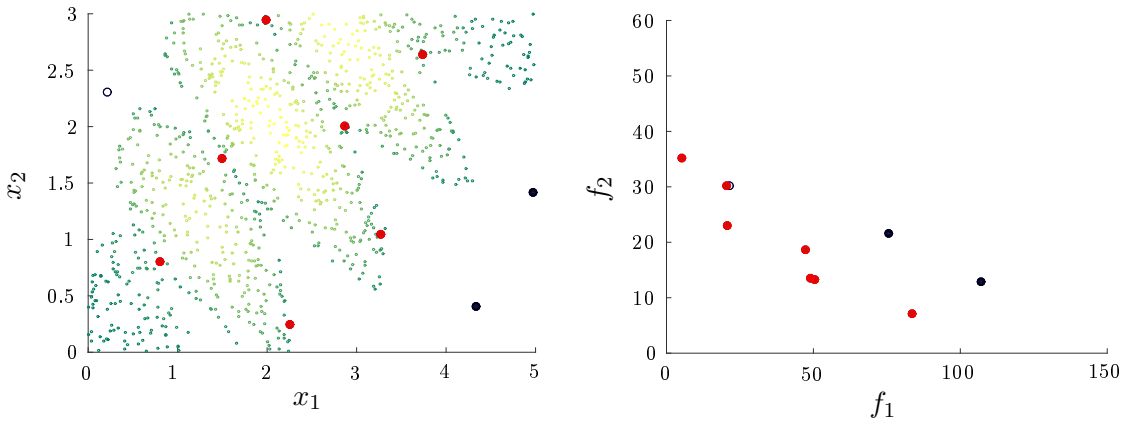


(c) Iteration 4 (d) Iteration 4

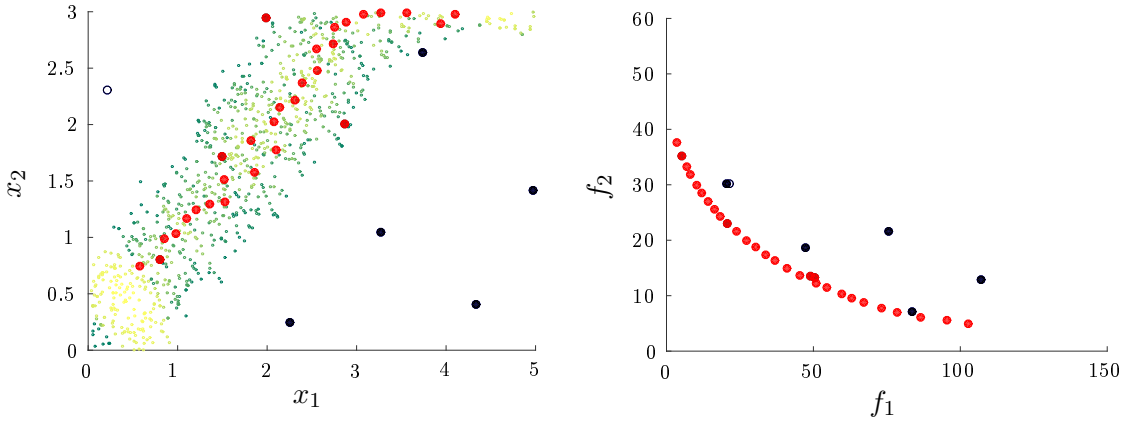


(e) Iteration 5 (f) Iteration 5

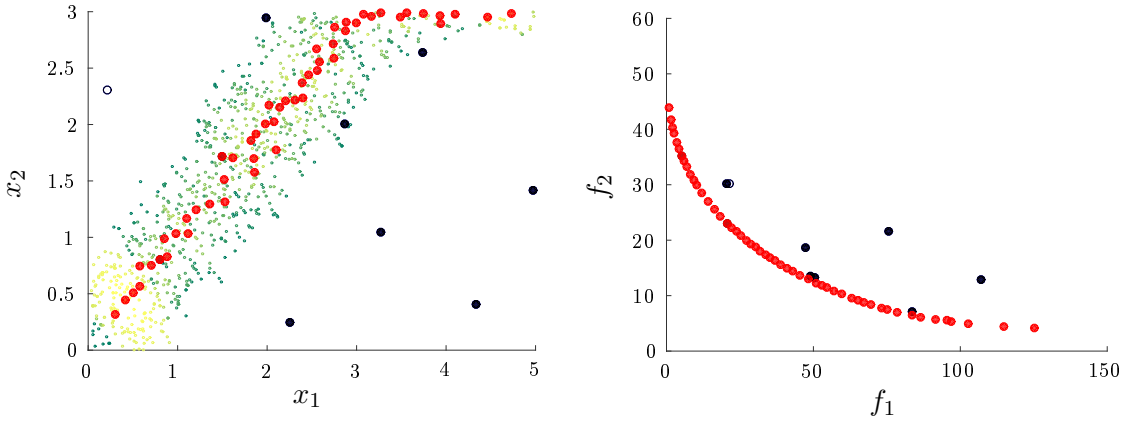
Figure 3.23: Illustration of one run of the BMOO algorithm on the BNH problem. Batches of $q = 10$ experiments are made at each iteration. Non-dominated solutions are shown as red disks and dominated ones as black disks.



(a) Iteration 0 (b) Iteration 0



(c) Iteration 1 (d) Iteration 1



(e) Iteration 2 (f) Iteration 2

Figure 3.24: Illustration of one run of the BMOO algorithm on the BNH problem. Batches of $q = 25$ experiments are made at each iteration. Non-dominated solutions are shown as red disks and dominated ones as black disks.

or prefer the improvement of one objective over the improvement of another. To encode this preference, in this section, we propose the expected weighted hypervolume improvement (EWHI) criterion. This criterion is an extension of the weighted hypervolume indicator (WHI) introduced by Zitzler et al. (2007) to a Bayesian setting where the functions of the problem are modeled by Gaussian processes.

The hypervolume indicator naturally introduces an implicit preference toward certain regions of the Pareto front[14], as discussed in Section 3.4 (see Figure 3.12). Based on this observation, Zitzler et al. (2007) proposed to introduce a weight function in the definition of the hypervolume indicator to orient this preference[15]:

$$I_n^{\omega} = \int_{H_n} \omega(y)\mathrm{d}y, \tag{3.30}$$

where H_n is the dominated subset of $\mathbb{B}$, as defined in the unconstrained case in Section 2.2.2, and $\omega : \mathbb{R}^p \to \mathbb{R}^+$ is a positive weight function. In this setting, the value $\omega(y)$ for some $y \in \mathbb{R}^p$ can be seen as a reward for dominating y.

Remark 27 *Observe that, up to a constant, the usual hypervolume indicator is recovered with the weight function $\omega = \mathbb{1}_{\mathbb{B}}$. The WHI is thus a generalization of the hypervolume indicator.*

Later, Emmerich et al. (2014) showed that for weight functions possessing the bounded improper integral (BI) property (see Definition 1), the introduction of a reference point in the definition of the WHI is not required.

Definition 1 *A positive weight function $\omega : \mathbb{R}^m \to \mathbb{R}^+$ is said to possess the bounded improper integral property if for all $\alpha \in \mathbb{R}^m$*

$$\int_{y \geq \alpha} \omega(y)dy < +\infty. \tag{3.31}$$

The generalization of (3.30) to the constrained multi-objective case is straightforward when the extended domination rule (2.21) is used. It corresponds to taking H_n and $\mathbb{B}$ as in Section 2.3.2 and to consider weight functions defined on $\mathbb{R}^{p+q}$. Then, to get rid of the reference points y^{low} and y^{upp}, it is required that ω be a measure on $\mathbb{R}^{p+q}$[16]. Based on these remarks, we define the EWHI criterion in the constrained multi-objective setting as:

$$\rho_n(x) = \int_{H_n^c} \omega(y) \cdot \mathbb{P}_n\left(\xi(x) \lhd y\right) \mathrm{d}y, \tag{3.32}$$

where $H_n = \{y \in \mathbb{R}^{p+q};\ \exists i \leq n,\ (f(X_i),\, c(X_i)) \lhd y\}$ and $\omega : \mathbb{R}^{p+q} \to \mathbb{R}^+$ is a measure on $\mathbb{R}^{p+q}$.

[14] In the bi-objective case and for continuous fronts, Auger et al. (2009c) made explicit this preference and showed that it is related both to the choice of the reference point and to the shape of the Pareto front.

[15] In the original definition of the WHI, the authors introduce additional terms to weight the axis. In this work, one of our objective is to get rid of the reference point, as proposed by Emmerich et al. (2014). Therefore we do not consider these terms.

[16] Note that in the constrained setting, lower bounding values are required and the BI property is not sufficient to insure that the integral is well defined in the absence of reference points.

Observe that this definition of the EI corresponds to the loss function

$$\varepsilon_n(\underline{X}, (f,c)) = \mu(H \setminus H_n), \tag{3.33}$$

where $H = \{y \in \mathbb{R}^{p+q};\ \exists x \in \mathbb{X},\ (f(x),\, c(x)) \lhd y\}$ is the region of $\mathbb{R}^{p+q}$ that is dominated by the Pareto front, $\underline{X} = (X_1, X_2, \ldots)$ is an optimization strategy for (f,c) and μ is the measure with density ω with respect to the Lebesgue measure.

If it is assumed that the objectives and constraints are independent and that ω can be put under the product form:

$$\begin{array}{rccl} \omega & : & \mathbb{R}^p \times \mathbb{R}^q & \to & \mathbb{R}^+ \\ & & (y_{\mathrm{o}}, y_{\mathrm{c}}) & \mapsto & \omega_{\mathrm{o}}(y_{\mathrm{o}}) \cdot \omega_{\mathrm{c}}(y_{\mathrm{c}})\,, \end{array} \tag{3.34}$$

where ω_{o} is a measure on $\mathbb{R}^p$ and ω_{c} is a measure on $\mathbb{R}^q$, then a decomposition similar to that introduced in Section 2.3.3 is possible:

$$\rho_n(x) = \rho_n^{\mathrm{feas}}(x) + \rho_n^{\mathrm{unf}}(x), \tag{3.35}$$

where

$$\left\{ \begin{array}{rcl} \rho_n^{\mathrm{feas}}(x) & = & \Omega_{\mathrm{c}}^- \cdot \mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0) \cdot \displaystyle\int_{\mathbb{R}^p} \omega_{\mathrm{o}}(y_{\mathrm{o}})\, \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right)\, \mathrm{d}y_{\mathrm{o}}\,, \\ \rho_n^{\mathrm{unf}}(x) & = & \Omega_{\mathrm{o}} \cdot \displaystyle\int_{H_{n,\mathrm{c}}^c} \omega_{\mathrm{c}}(y_{\mathrm{c}})\, \mathbb{P}_n\left(\xi_{\mathrm{c}}(x) \prec y_{\mathrm{c}}^+\right)\, \mathbb{1}_{y_{\mathrm{c}} \not\preceq 0}\, \mathrm{d}y_{\mathrm{c}}, \end{array} \right. \tag{3.36}$$

prior to finding a feasible solution and

$$\left\{ \begin{array}{rcl} \rho_n^{\mathrm{feas}}(x) & = & \Omega_{\mathrm{c}}^- \cdot \mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0) \cdot \displaystyle\int_{H_{n,\mathrm{o}}^c} \omega_{\mathrm{o}}(y_{\mathrm{o}})\, \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right)\, \mathrm{d}y_{\mathrm{o}}\,, \\ \rho_n^{\mathrm{unf}}(x) & = & 0, \end{array} \right. \tag{3.37}$$

once a feasible solution is known, where $\Omega_{\mathrm{o}} = \int_{\mathbb{R}^p} \omega_{\mathrm{o}}$ and $\Omega_{\mathrm{c}}^- = \int_{\mathbb{R}^q \setminus [-\infty,0]^q} \omega_{\mathrm{c}}$.

The extension of the SMC procedure proposed in Section 2.4.1 to the approximate computation of the integrals in (3.36) and (3.37) is straightforward. It consists in introducing the respective weight functions in the estimator (3.9). Note that in this case, prior to finding a feasible solution, both ρ_n^{unf} and ρ_n^{feas} are computed approximately.

As regards the sampling density $\pi_n^{\mathbb{Y}}$ to be used in the SMC procedure, a development similar to that used in Section 3.3.2 for the L_2^{opt} density leads to the definition of the following optimal density:

$$\pi_n^{\mathbb{Y}}(y) \propto \sqrt{\sum_{k=1}^{m_{\mathbb{X}}} \omega(y)^2\, \mathbb{P}_n\left(\xi(x_{k,n}) \prec y\right)^2} \cdot \mathbb{1}_{G_n}, \tag{3.38}$$

where $\xi = \xi_{\mathrm{o}}$ or ξ_{c}, $\omega = \omega_{\mathrm{o}}$ or ω_{c} and $G_n = H_{n,\mathrm{o}}^c$ or $H_{n,\mathrm{c}}^c$, depending on which phase of the optimization process is considered, and the $(x_{n,k})_{1 \leq k \leq m}$ are candidates for the optimization of the criterion (see Section 3.2).

In a bi-objective context, Zitzler et al. (2007) propose three weight functions that encode

respectively preference for extremal solutions, preference for one of the objective functions and bias toward a reference point (see also Auger et al. (2009a)). The authors point out that it is possible to define ω as a combination of these three weight functions to combine their effects. Based on these results, in Figure 3.25, we illustrate the operation of the EWHI criterion on the BNH problem of Section 2.5.4 for the two following weight functions defined on $\mathbb{R}^p \times \mathbb{R}^q$, with $p = 2$ and $q = 1$:

$$\begin{cases} \omega_1(y_o, y_c) &= \frac{1}{15} e^{-\frac{y_{o,1}}{15}} \cdot \frac{\mathbb{1}_{[0,150]}(y_{o,1})}{150} \cdot \frac{\mathbb{1}_{[0,60]}(y_{o,2})}{60} \cdot \mathbb{1}_{\mathbb{R}}(y_c), \\ \omega_2(y_o, y_c) &= \frac{1}{2} \left(\varphi\left(y_o, \mu_1, C\right) + \varphi\left(y_o, \mu_2, C\right)\right) \cdot \mathbb{1}_{\mathbb{R}}(y_c), \end{cases} \tag{3.39}$$

where $\varphi(y, \mu, C)$ denotes the Gaussian probability density function with mean μ and covariance matrix C, evaluated at y. The ω_1 weight function is based on an exponential distribution and encodes preference for the minimization of the first objective. The ω_2 weight function is a sum of two bivariate Gaussian distributions and encodes preference for improving upon two reference points μ_1 and μ_2[17]. For this experiment, BMOO is initialized with a pseudo-maximin design of $N = 10$ experiments and is iterated over 20 iterations and we take $\mu_1 = (80, 20)$, $\mu_2 = (30, 40)$ and $C = RS(RS)^T$, where

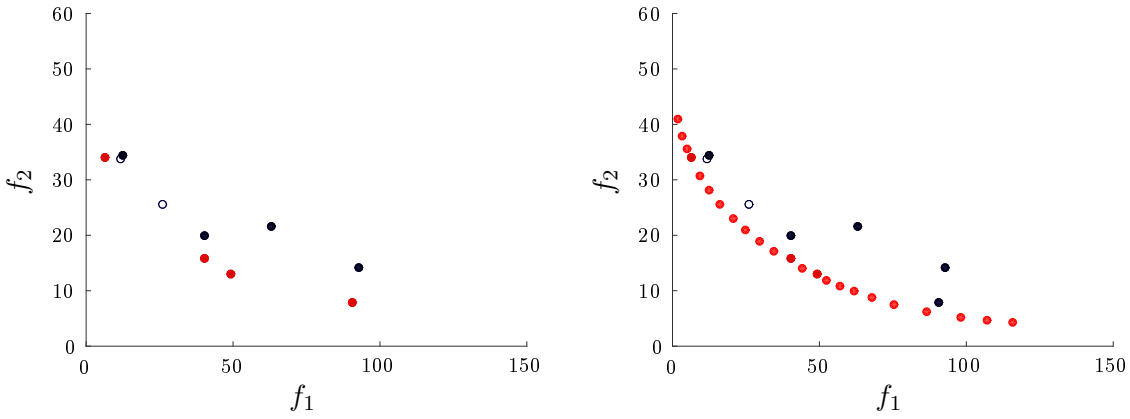
$$R = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} 20 & 0 \\ 0 & 3 \end{bmatrix}. \tag{3.40}$$

As expected, the resulting distributions of solutions are concentrated in the regions of the Pareto front that dominate the regions of high weight values.

Remark 28 *In the EMO literature, an up-to-date review of approaches that have been proposed to incorporate user preferences into the optimization scheme is provided by Li et al. (2016a) and theoretical results on optimal μ-distributions (see Section 3.4.3) for the weighted hypervolume indicator are provided by Auger et al. (2009b).*

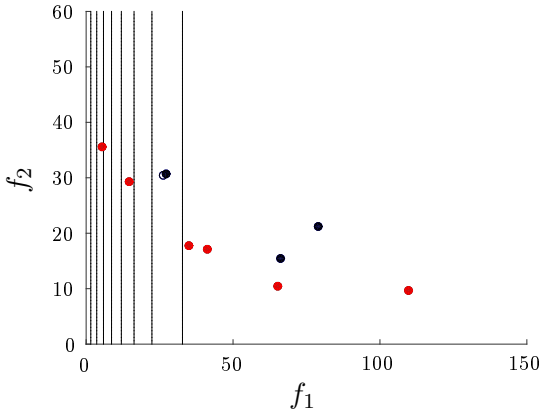
Remark 29 *An alternative approach based on desirability functions (see Harrington (1965)) is proposed by Wagner and Trautmann (2010). In their approach, each objective is mapped to a (nonlinear) desirability function that is comprised between 0 and 1, a value of 1 indicating full satisfaction of the end-user. The authors then propose the DF-SMS-EMOA algorithm, which extends the SMS-EMOA algorithm of Beume et al. (2007) to this setting. This approach was shown by Emmerich et al. (2014) to be a special case of the WHI approach. Furthermore the authors make explicit the relationship between desirability functions and weighted hypervolume indicator for desirability functions of the Harrington type and of the Derringer-Suich type (Suich and Derringer, 1977) and show how they can be used to define weight functions. By extension, these can be used to define weight functions for the EWHI criterion as well.*

[17] Note that in this example, we take ω_c uniform over the entire $\mathbb{R}$ domain. In that case, ω_c is not a measure. However, observe that on problems where a feasible solution is known from the initial DOE, which is the case for the BNH problem, the EWHI is equal to the term ρ_n^{feas} in (3.37). In that case, we omit the term Ω_c^- because it does not influence the maximizer of the EWHI.

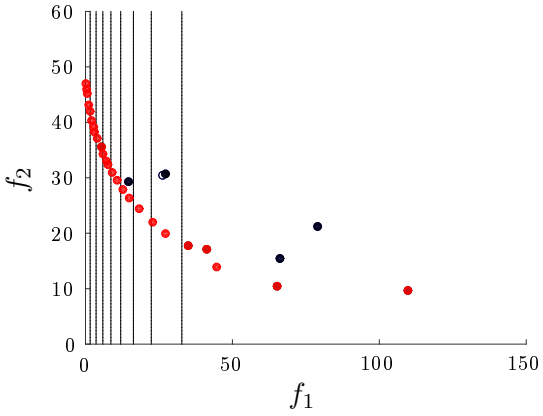


(a) EHVI ($N = 10$)

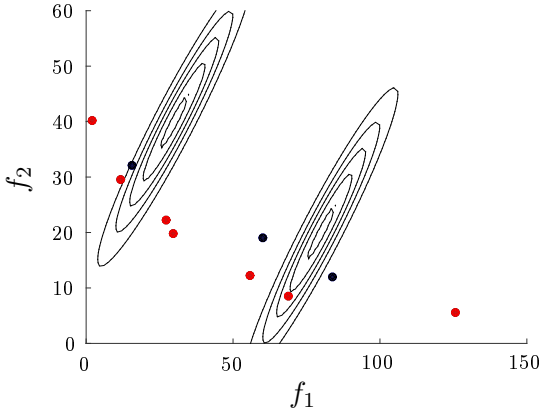
(b) EHVI ($N = 30$)



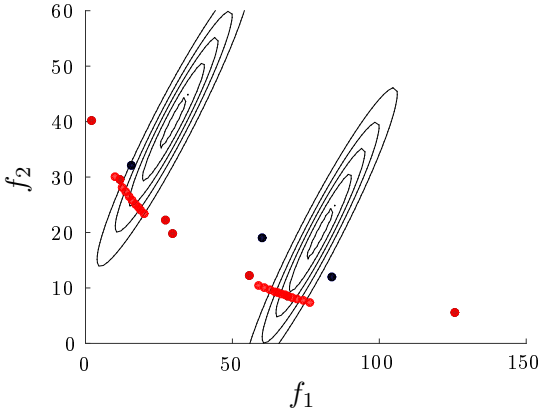
(c) EWHI with ω_1 ($N = 10$)



(d) EWHI with ω_1 ($N = 30$)



(e) EWHI with ω_2 ($N = 10$)



(f) EWHI with ω_2 ($N = 30$)

Figure 3.25: Distributions obtained after 20 iterations of the BMOO algorithm on the BNH problem when the weight functions ω_1 and ω_2 are used. The results obtained using the EHVI are shown for reference. The contours of the weight functions are represented as black lines and the non-dominated solutions as red disks. Black disks indicate dominated solutions.

Remark 30 *In a Bayesian setting, Yang et al. (2016a,b) define the truncated expected hypervolume improvement. This criterion can be used on bi-objective problems to encode preference toward a domain of the objective space defined by bound constraints of the form* $(y_1, y_2) \in [A_1, B_1] \times [A_2, B_2]$. *Note that a similar preference can be encoded using the EWHI criterion by taking* $\omega_{\mathrm{o}}(y_1, y_2) \propto \mathbb{1}_{[A_1,B_1]}(y_1) \cdot \mathbb{1}_{[A_2,B_2]}(y_2)$.

3.6 Additional material

3.6.1 Local optimization algorithms results

Tables 3.7, 3.8 and 3.9 present the results obtained by the Cobyla, Active-Set, Interior-Point and SQP local optimization algorithms on the YUCCA test problem with $\kappa = 1$, $\kappa = 3$ and $\kappa = 5$. In Section 3.2, only the results of the best scoring algorithm are given in the tables. Here, we present the full results. The algorithms are run with their default values, except for the maximum number of functions evaluations, which is set to $N_{max} = 1000$, and the tolerance to constraint violation, which is set to 10^{-5} on each constraint. See Section 3.2 for more details.

3.6.2 Correction of the adaptive procedure to set $\mathbb{B}_{\mathrm{o}}$

A preliminary adaptive procedure to set $\mathbb{B}_{\mathrm{o}}$ has been introduced in Section 2.7.2. In this procedure, for every objective f_i, $1 \leq i \leq p$, the upper bounding value $y_{\mathrm{o},i}^{\mathrm{upp}}$ is set as an approximation of $\max_{\mathbb{X}} f_i$. However, in Section 3.4, it is empirically shown that for better performances, it is advisable to set the upper bounding values $y_{\mathrm{o}}^{\mathrm{upp}}$ higher than the nadir point but not too far from the Pareto front (see Remark 24 in Section 3.4.3). This procedure is therefore not suitable for problems where the maximum of f_i over the entire $\mathbb{X}$ domain is large compared to its maximum value on the Pareto front. This happens, for example, when the maximum of f_i over the feasible set is small compared to its maximum over $\mathbb{X}$. Such a situation occurs, for the FICUS problem, when the parameter r is set small compared to one, as illustrated in Figure 3.26. Here, we propose a new adaptive procedure to set $\mathbb{B}_{\mathrm{o}}$ that better adapts to this potential difficulty.

Assume that a set of n evaluation results $\xi(X_i)$, $1 \leq i \leq n$, are available and recall from Section 3.3 that, assuming independence between the objectives and constraints, prior to finding a feasible solution, the EI at some point $x \in \mathbb{X}$ is a sum of two terms:

$$\begin{aligned} \rho_n(x) &= \rho_n^{\mathrm{unf}}(x) + \rho_n^{\mathrm{feas}}(x)\,, \\ &\propto \frac{1}{\left|\mathbb{B}_{\mathrm{c}}^{-}\right|} \int_{\mathbb{B}_{\mathrm{c}} \setminus H_{n,\mathrm{c}}} \mathbb{P}_n\left(\xi_{\mathrm{c}}^{+}(x) \prec y_{\mathrm{c}}^{+}\right) \mathbb{1}_{y_{\mathrm{c}} \not\preceq 0}\, \mathrm{d}y_{\mathrm{c}} + \frac{\mathbb{P}_n(\xi_{\mathrm{c}}(x) \leq 0)}{|\mathbb{B}_{\mathrm{o}}|} \int_{\mathbb{B}_{\mathrm{o}}} \mathbb{P}_n\left(\xi_{\mathrm{o}}(x) \prec y_{\mathrm{o}}\right) \mathrm{d}y_{\mathrm{o}}\,. \end{aligned}$$

The EI value of points that have a small probability of feasibility is not strongly influenced by $\mathbb{B}_{\mathrm{o}}$ because then $\rho_n^{\mathrm{feas}}(x)$ is likely to ba small compared to $\rho_n^{\mathrm{unf}}(x)$. For points with a significant probability of feasibility on the other hand, the term $\rho_n^{\mathrm{feas}}(x)$ can be seen as a reward that is larger when the volume between $\min(\max(\xi_{\mathrm{o}}(x)\,,\, y_{\mathrm{o}}^{\mathrm{low}})\,,\, y_{\mathrm{o}}^{\mathrm{upp}})$ and $y_{\mathrm{o}}^{\mathrm{upp}}$ is larger. To avoid truncating

d	COBYLA		active-set		interior-point		SQP	
2	28	**8.7** (5.5)	27	**7.7** (2.1)	30	11.0 (6.4)	30	**7.2** (1.6)
5	25	21.9 (10.7)	27	20.6 (7.7)	28	24.9 (24.7)	28	**16.2** (3.0)
10	17	80.6 (88.4)	17	37.4 (7.7)	25	50.0 (18.4)	25	**31.4** (5.8)
20	14	168.2 (84.3)	10	77.0 (14.8)	19	108.2 (42.6)	16	70.6 (12.6)
50	4	593.8 (191.9)	5	298.4 (154.8)	14	491.6 (234.2)	11	191.1 (23.8)
100	0	- (-)	0	- (-)	0	- (-)	1	405.0 (0.0)

Table 3.7: Results obtained using the Cobyla, Active-Set, Interior-Point and SQP algorithms on the YUCCA test problem with $\kappa = 1$. 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 presented in the table. Dash symbols are used when a value cannot be calculated.

d	COBYLA		active-set		interior-point		SQP	
2	27	17.2 (5.2)	27	**10.1** (2.7)	27	20.6 (4.0)	27	**9.4** (2.4)
5	19	51.7 (52.2)	17	25.2 (6.8)	25	52.1 (11.0)	24	**21.8** (3.1)
10	15	72.6 (30.4)	14	56.4 (54.5)	22	109.5 (40.9)	19	40.9 (5.5)
20	8	242.2 (208.5)	6	148.3 (63.8)	17	223.3 (37.4)	17	85.0 (0.0)
50	1	227.0 (0.0)	0	- (-)	4	752.0 (174.9)	3	205.0 (0.0)

Table 3.8: Results obtained using the Cobyla, Active-Set, Interior-Point and SQP algorithms on the YUCCA test problem with $\kappa = 3$. See Table 3.7 for conventions.

d	COBYLA		active-set		interior-point		SQP	
2	24	23.8 (6.0)	22	**12.0** (5.6)	28	30.9 (7.3)	27	**10.1** (1.6)
5	16	54.1 (26.9)	13	24.0 (6.0)	25	67.4 (18.2)	21	**21.6** (3.6)
10	15	91.1 (31.4)	8	47.9 (12.8)	23	120.0 (26.5)	20	**42.2** (4.9)
20	1	102.0 (0.0)	2	116.5 (44.5)	12	261.7 (26.5)	10	80.8 (8.9)
50	0	- (-)	0	- (-)	5	752.8 (117.5)	3	205.0 (0.0)

Table 3.9: Results obtained using the Cobyla, Active-Set, Interior-Point and SQP algorithms on the YUCCA test problem with $\kappa = 5$. See Table 3.7 for conventions.

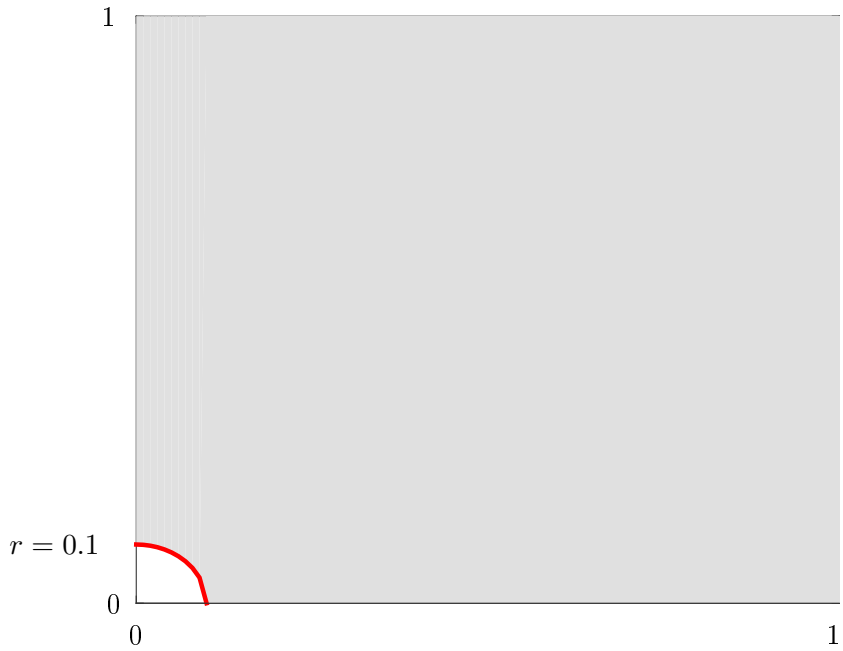


Figure 3.26: Illustration of a situation where the procedure of Section 2.7.2 to set $\mathbb{B}_{\mathrm{o}}$ is not suitable because the size of the Pareto front is small compared to the size of the domain of variation of the objectives (gray region). The illustration is based on the FICUS $(2, 0.1, 2)$ problem.

this volume, in that case, we propose to set y^{upp} and y^{low} as follows:

$$\left\{ \begin{array}{rcll} y^{\mathrm{low}}_{\mathrm{o},i,n} & = & \min\left(\min_{i\leq n} \xi_{\mathrm{o},i}(X_i),\ \min_{x\in\mathcal{X}_n} \widehat{\xi}_{\mathrm{o},\,i,\,n}(x) - \lambda_{\mathrm{o}}\sigma_{\mathrm{o},\,i,\,n}(x)\right), & 1 \leq i \leq p\,, \\ y^{\mathrm{upp}}_{\mathrm{o},i,n} & = & \max\left(\max_{i\leq n} \xi_{\mathrm{o},i}(X_i),\ \max_{x\in\mathcal{X}_n} \widehat{\xi}_{\mathrm{o},\,i,\,n}(x) + \lambda_{\mathrm{o}}\sigma_{\mathrm{o},\,i,\,n}(x),\right), & 1 \leq i \leq p\,, \end{array} \right. \tag{3.41}$$

where λ_{o} is a positive number as in Section 2.7.2 and $\mathcal{X}_n = (x_{n,i})_{1\leq i\leq m}$ is a cloud of particles approximately distributed from a density π_n on $\mathbb{X}$. In other words, $\mathbb{B}_{\mathrm{o}}$ is taken in that case as an hyper-rectangle likely to contains both the past and future observations.

Once a feasible solution has been found, it is possible to adapt $\mathbb{B}_{\mathrm{o}}$ to keep it close to the set of feasible and non-dominated solutions. To this purpose, we consider the two following approximations of the Pareto set:

$$\left\{ \begin{array}{rcl} P_{\max} & = & \mathrm{Pareto}\left(\{\xi_{\mathrm{o}}(X_j)\,;\, \xi_{\mathrm{c}}(X_j) \leq 0, 1 \leq j \leq n\} \cup \left\{\widehat{\xi}_{\mathrm{o},n}(x_{n,i}) + \lambda_{\mathrm{o}}\sigma_{\mathrm{o},\,n}(x_{n,i}), 1 \leq i \leq m\right\}\right), \\ P_{\min} & = & \mathrm{Pareto}\left(\{\xi_{\mathrm{o}}(X_j)\,;\, \xi_{\mathrm{c}}(X_j) \leq 0, 1 \leq j \leq n\} \cup \left\{\widehat{\xi}_{\mathrm{o},n}(x_{n,i}) - \lambda_{\mathrm{o}}\sigma_{\mathrm{o},\,n}(x_{n,i}), 1 \leq i \leq m\right\}\right). \end{array} \right. \tag{3.42}$$

The set $P_{\max}$ can be seen as a pessimistic approximation of the Pareto front and the set $P_{\min}$ as an optimistic one. Then, for every objective function f_i, $i \in [\![1, p]\!]$, we define upper and lower bounding values

$$\left\{ \begin{array}{rcl} f_{i,\max} & = & \max P_{i,\max}, \\ f_{i,\min} & = & \min P_{i,\min}, \end{array} \right. \tag{3.43}$$

where $P_{i,\max}$ and $P_{i,\min}$ denote the vectors made of the ith components of $P_{\max}$ and $P_{\min}$ respectively. The upper and lower bounds $y_{\mathrm{o},i,n}^{\mathrm{upp}}$ and $y_{\mathrm{o},i,n}^{\mathrm{low}}$, $i \in [\![1,p]\!]$ can then be set as

$$\left\{ \begin{array}{rcl} y_{\mathrm{o},i,n}^{\mathrm{upp}} & = & f_{i,\max} + \gamma^{\mathrm{upp}} \left(f_{i,\max} - f_{i,\min} \right), \\ y_{\mathrm{o},i,n}^{\mathrm{low}} & = & f_{i,\min} - \gamma^{\mathrm{low}} \left(f_{i,\max} - f_{i,\min} \right), \end{array} \right. \tag{3.44}$$

where γ^{upp} and γ^{low} are scaling parameters. In our experiments, we often set $\gamma^{\mathrm{upp}} = \gamma^{\mathrm{low}} = 0$, which corresponds to choosing $y_{\mathrm{o},i,n}^{\mathrm{upp}} = f_{i,\max}$ and $y_{\mathrm{o},i,n}^{\mathrm{low}} = f_{i,\min}$.

Remark 31 *As regards the choice of $\mathbb{B}_{\mathrm{c}}$, there is no indication in our experiments that the adaptive procedure proposed in Section 2.7.2 is not suitable.*

3.6.3 Variance of the EI estimator

Here, we derive formulas that can be used to estimate the covariance between the estimators of the EI obtained using the proposed sequential Monte Carlo approach. To simplify the analysis, we consider the following form of the estimator (3.9):

$$I_n(x) \approx \frac{\Gamma_n}{m} \sum_{i=1}^{m} \frac{\mathbb{P}_n \left(\xi(x) \prec y_{n,i} \right)}{\gamma_n(y_{n,i})}, \tag{3.45}$$

where $x \in \mathbb{X}$ is a candidate point and the $(y_{n,i})_{1 \leq i \leq m} \in G_n^m$ are particles approximately distributed from a density $\pi_n \propto \gamma_n$ on G_n, where G_n denotes the non-dominated set. With these notations,

$$\Gamma_n = \int_{G_n} \gamma_n(y) \mathrm{d}y, \tag{3.46}$$

is the normalizing constant for the unnormalized probability density function γ_n. In this work, γ_n can be, for example, equal to $\mathbb{1}_{G_n}$ when the uniform density is used or to (3.13) when the L_2^{opt} density is used but the results of this section extend to other densities as well.

In the setting that we consider, the $(y_{n,i})_{1 \leq i \leq m}$ are obtained using SMC by considering a sequence of densities $(\pi_{n,t})_{0 \leq t \leq T}$ with $\pi_{n,0}$ uniform and $\pi_{n,T} = \pi_n$. The sequence $(\pi_{n,t})_{0 \leq t \leq T}$ comprises both initialization steps and intermediate transitions (see, e.g., Remark 21 in Section 3.3). In this setting, it is convenient to introduce the notations $(\gamma_{n,t})_{0 \leq t \leq T}$ and $(\Gamma_{n,t})_{0 \leq t \leq T}$ to refer to the associated sequences of unnormalized probability density functions and normalizing constants. In this framework, $\Gamma_n = \Gamma_{n,T}$ and $\gamma_n = \gamma_{n,T}$.

No closed form expressions of the normalizing constants $(\Gamma_{n,t})_{1 \leq t \leq T}$ are known in the general case. In practice, they have to be approximated. Observe that, for $t \in [\![1,T]\!]$

$$\begin{array}{rcl} \Gamma_{n,t} & = & \displaystyle\int_{G_n} \gamma_{n,t}(y) \mathrm{d}y \\ & = & \displaystyle\int_{G_n} \frac{\gamma_{n,t}(y)}{\pi_{n,t-1}(y)} \pi_{n,t-1}(y) \mathrm{d}y \\ & = & \Gamma_{n,t-1} \displaystyle\int_{G_n} \frac{\gamma_{n,t}(y)}{\gamma_{n,t-1}(y)} \pi_{n,t-1}(y) \mathrm{d}y \,. \end{array} \tag{3.47}$$

For all $t \in [\![1, T]\!]$, an approximation of $\Gamma_{n,t}$, $t > 0$, can thus be produced using the following recursion formula:

$$\widehat{\Gamma}_{n,t} = \widehat{\Gamma}_{n,t-1} \left(\frac{1}{m} \sum_{i=1}^{m} \frac{\gamma_{n,t}(y_{n,t-1,i})}{\gamma_{n,t-1}(y_{n,t-1,i})} \right), \tag{3.48}$$

where the $(y_{n,t-1,i})_{1 \leq i \leq m}$ are particles distributed from $\pi_{n,t-1}$ (see, e.g., Bect et al. (2016a)). The estimator that we actually consider in the approximate computation of the integral in the EI expression is then

$$\begin{aligned} \widehat{I}_n(x) &= \frac{\widehat{\Gamma}_n}{m} \sum_{i=1}^{m} \frac{\mathbb{P}_n\left(\xi(x) \prec y_{n,i}\right)}{\gamma_n(y_{n,i})} \\ &= \left(\frac{1}{m} \sum_{i=1}^{m} \frac{\mathbb{P}_n\left(\xi(x) \prec y_{n,i}\right)}{\gamma_n(y_{n,i})} \right) \cdot \prod_{t=1}^{T} \left(\frac{1}{m} \sum_{i=1}^{m} \frac{\gamma_{n,t}(y_{n,t-1,i})}{\gamma_{n,t-1}(y_{n,t-1,i})} \right) \Gamma_0, \end{aligned} \tag{3.49}$$

where $\Gamma_0 = |\mathbb{B}_{\mathrm{o}}|$ or $|\mathbb{B}_{\mathrm{c}} \setminus \mathbb{B}_{\mathrm{c}}^{-}|$ since $\pi_{n,0}$ is uniform in the settings that we consider.

To the best of our knowledge, without further assumptions, there exists no closed form expression for the variance of such an estimator (asymptotic results exist in some particular cases). To go farther, a common practice in the literature on sequential Monte Carlo is to make the following assumptions (see, e.g, Cérou et al. (2012)):

(i) The samples $(\mathcal{Y}_{n,t})_{0 \leq t \leq T}$ are independent.

(ii) For all $t \in [\![0, T]\!]$, the particles $(y_{n,t,i})_{1 \leq i \leq m}$ are independently and identically distributed from $\pi_{n,t}$.

To simplify the analysis, we introduce the following notations:

$$\widehat{\theta}_{n,t} = \frac{1}{m} \sum_{i=1}^{m} \frac{\gamma_{n,t}(y_{n,t-1,i})}{\gamma_{n,t-1}(y_{n,t-1,i})}, \; t \in [\![1, T]\!], \tag{3.50}$$

and

$$\widehat{\alpha}_n(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{\mathbb{P}_n\left(\xi(x) \prec y_{n,i}\right)}{\gamma_n(y_{n,i})}. \tag{3.51}$$

Observe that, under (ii), $\widehat{\theta}_{n,t}$ is an unbiased estimator of $\theta_{n,t} = \frac{\Gamma_{n,t}}{\Gamma_{n,t-1}}$, $1 \leq t \leq T$ (see Equation (3.47)), and that $\widehat{\alpha}_n(x)$ is an unbiased estimator of $\alpha_n(x) = \frac{I_n(x)}{\Gamma_n}$ (see Equation (3.45)). Moreover, under (i), the $(\widehat{\theta}_{n,t})_{1 \leq t \leq T}$ and $\widehat{\alpha}_n(x)$ are all independent. Then denote, for all $t \in [\![1, T]\!]$,

$$\widehat{\Theta}_{n,t} = \Gamma_0 \prod_{u=1}^{t} \widehat{\theta}_{n,u}, \tag{3.52}$$

and observe that, under both (i) and (ii), $\widehat{\Theta}_{n,t}$, $t \in [\![1, T]\!]$, is the product of unbiased and independent estimators. As such, $\widehat{\Theta}_{n,t}$ is an unbiased estimator of $\Theta_{n,t} = \Gamma_0 \prod_{u=1}^{t} \theta_{n,u} = \Gamma_{n,t}$.

Moreover, $\widehat{\theta}_{n,\,t+1}$ and $\widehat{\Theta}_{n,\,t}$ are independent and, for all $t \in [\![1, T]\!]$,

$$\begin{aligned}
\mathrm{Var}\left(\widehat{\Theta}_{n,\,t}\right) &= \mathrm{Var}\left(\widehat{\theta}_{n,\,t} \cdot \widehat{\Theta}_{n,\,t-1}\right) \\
&= \mathbb{E}\left(\widehat{\theta}^2_{n,\,t} \cdot \widehat{\Theta}^2_{n,\,t-1}\right) - \mathbb{E}\left(\widehat{\theta}_{n,\,t} \cdot \widehat{\Theta}_{n,\,t-1}\right)^2 \\
&= \mathbb{E}\left(\widehat{\theta}^2_{n,\,t}\right) \cdot \mathbb{E}\left(\widehat{\Theta}^2_{n,\,t-1}\right) - \mathbb{E}\left(\widehat{\theta}_{n,\,t}\right)^2 \cdot \mathbb{E}\left(\widehat{\Theta}_{n,\,t-1}\right)^2 \\
&= \left(\mathrm{Var}\left(\widehat{\theta}_{n,\,t}\right) + \theta^2_{n,\,t}\right) \cdot \left(\mathrm{Var}\left(\widehat{\Theta}_{n,\,t-1}\right) + \Theta^2_{n,\,t-1}\right) - \theta^2_{n,\,t} \cdot \Theta^2_{n,\,t-1} \\
&= \mathrm{Var}\left(\widehat{\theta}_{n,\,t}\right) \cdot \mathrm{Var}\left(\widehat{\Theta}_{n,\,t-1}\right) + \theta^2_{n,\,t} \cdot \mathrm{Var}\left(\widehat{\Theta}_{n,\,t-1}\right) + \Theta^2_{n,\,t-1} \cdot \mathrm{Var}\left(\widehat{\theta}_{n,\,t}\right) .
\end{aligned}$$

For all $t \in [\![1, T]\!]$, the variance of $\widehat{\Theta}_{n,\,t}$ can thus be expressed in an elegant form using the coefficients of variation of $\Theta_{n,\,t}$, $\Theta_{n,\,t-1}$ and $\theta_{n,\,t}$:

$$\Delta^2_{n,\,t} = \delta^2_{n,\,t} + \left(1 + \delta^2_{n,\,t}\right) \cdot \Delta^2_{n,\,t-1}, \tag{3.53}$$

where $\Delta^2_{n,\,t} = \frac{\mathrm{Var}\left(\widehat{\Theta}_{n,\,t}\right)}{\Theta^2_{n,\,t}}$ and $\delta^2_{n,\,t} = \frac{\mathrm{Var}\left(\widehat{\theta}_{n,\,t}\right)}{\theta^2_{n,\,t}}$.

Applying the same treatment to $\widehat{I}_n(x) = \widehat{\alpha}_n(x) \cdot \widehat{\Theta}_{n,\,T}$, we obtain

$$\frac{\mathrm{Var}\left(\widehat{I}_n(x)\right)}{I_n(x)^2} = \Lambda_n(x)^2 + \left(1 + \Lambda_n(x)^2\right) \cdot \Delta^2_{n,\,T}, \tag{3.54}$$

where $\Lambda_n(x)$ denotes the coefficient of variation of $\widehat{\alpha}_n(x)$.

Estimators for the terms $\delta^2_{n,\,t}$ and $\Delta^2_{n,\,t}$ in Equation (3.53) and for the term $\Lambda_n(x)^2$ in Equation (3.54) can be derived when (ii) is assumed. Observe that, in this case,

$$\begin{aligned}
\delta^2_{n,\,t} &= \frac{\mathrm{Var}\left(\widehat{\theta}_{n,\,t}\right)}{\theta^2_{n,\,t}} \\
&= \frac{1}{m} \frac{\mathrm{Var}\left(\frac{\gamma_{n,\,t}(y_{n,\,t-1,\,1})}{\gamma_{n,\,t-1}(y_{n,\,t-1,\,1})}\right)}{\mathbb{E}\left(\frac{\gamma_{n,\,t}(y_{n,\,t-1,\,1})}{\gamma_{n,\,t-1}(y_{n,\,t-1,\,1})}\right)^2},
\end{aligned} \tag{3.55}$$

and an estimator of $\delta^2_{n,\,t}$, $t \in [\![1, T]\!]$, is

$$\widehat{\delta}^2_{n,\,t} = \frac{\left(\sum_{i=1}^m \frac{\gamma_{n,\,t}(y_{n,\,t-1,\,i})^2}{\gamma_{n,\,t-1}(y_{n,\,t-1,\,i})^2}\right)}{\left(\sum_{i=1}^m \frac{\gamma_{n,\,t}(y_{n,\,t-1,\,i})}{\gamma_{n,\,t-1}(y_{n,\,t-1,\,i})}\right)^2} - \frac{1}{m}. \tag{3.56}$$

This estimator can then be plugged in Equation (3.53) to obtain an estimator of $\Delta^2_{n,\,t}$:

$$\widehat{\Delta}^2_{n,\,t} = \widehat{\delta}^2_{n,\,t} + \left(1 + \widehat{\delta}^2_{n,\,t}\right) \cdot \widehat{\Delta}^2_{n,\,t-1}. \tag{3.57}$$

Using a similar reasoning, under (ii), an estimator of $\Lambda_n(x)^2$ is

$$\widehat{\Lambda}_n(x)^2 = \frac{\left(\sum_{i=1}^m \frac{\mathbb{P}_n(\xi(x)\prec y_{n,\,i})^2}{\gamma_n(y_{n,\,i})^2}\right)}{\left(\sum_{i=1}^m \frac{\mathbb{P}_n(\xi(x)\prec y_{n,\,i})}{\gamma_n(y_{n,\,i})}\right)^2} - \frac{1}{m}. \tag{3.58}$$

To summarize, the variance of the estimator (3.49) can be estimated using

$$\mathrm{Var}\left(\widehat{I}_n(x)\right) \approx \left(\frac{\widehat{\Gamma}_{n,\,T}}{m}\sum_{i=1}^m \frac{\mathbb{P}_n\left(\xi(x)\prec y_{n,\,i}\right)}{\gamma_n(y_{n,\,i})}\right)^2 \cdot \left(\widehat{\Lambda}_n(x)^2 + \left(1+\widehat{\Lambda}_n(x)^2\right)\cdot \widehat{\Delta}^2_{n,\,T}\right), \tag{3.59}$$

where $\widehat{\Gamma}_{n,\,T}$ and $\widehat{\Delta}^2_{n,\,T}$ are obtained recursively using (3.48) and (3.57), and $\widehat{\Lambda}_n(x)^2$ is computed using (3.58).

Let us now consider a pair $(x_1, x_2) \in \mathbb{X}^2$. The covariance between $\widehat{I}_n(x_1)$ and $\widehat{I}_n(x_2)$ is given by

$$\begin{aligned}
&\mathrm{Cov}\left(\widehat{I}_n(x_1), \widehat{I}_n(x_2)\right)\\
&= \mathbb{E}\left(\widehat{\Gamma}_n^2 \cdot \widehat{\alpha}_n(x_1)\cdot\widehat{\alpha}_n(x_2)\right) - I_n(x_1)\cdot I_n(x_2)\\
&= \mathbb{E}\left(\widehat{\Gamma}_n^2\right)\cdot \mathbb{E}\left(\widehat{\alpha}_n(x_1)\cdot\widehat{\alpha}_n(x_2)\right) - I_n(x_1)\cdot I_n(x_2)\\
&= \left(\mathrm{Var}\left(\widehat{\Gamma}_n\right) + \Gamma_n^2\right)\cdot\left(\mathrm{Cov}\left(\widehat{\alpha}_n(x_1)\,,\,\widehat{\alpha}_n(x_2)\right) + \alpha_n(x_1)\cdot\alpha_n(x_2)\right) - I_n(x_1)\cdot I_n(x_2),
\end{aligned}$$

which, again, can be formulated in a more elegant form as

$$\frac{\mathrm{Cov}\left(\widehat{I}_n(x_1), \widehat{I}_n(x_2)\right)}{I_n(x_1)\cdot I_n(x_2)} = \Lambda_n^{1,2}(x_1,x_2) + \left(\Lambda_n^{1,2}(x_1,x_2)+1\right)\cdot\Delta^2_{n,\,T}, \tag{3.60}$$

where

$$\Lambda_n^{1,2}(x_1,x_2) = \frac{\mathrm{Cov}\left(\widehat{\alpha}_n(x_1)\,,\,\widehat{\alpha}_n(x_2)\right)}{\alpha_n(x_1)\cdot\alpha_n(x_2)}. \tag{3.61}$$

As previously, observe that under the assumption (ii),

$$\Lambda_n^{1,2}(x_1,x_2) = \frac{1}{m}\cdot\frac{\mathrm{Cov}\left(\frac{\mathbb{P}_n(\xi(x_1)\prec y_{n,\,1})}{\gamma_n(y_{n,\,1})}, \frac{\mathbb{P}_n(\xi(x_2)\prec y_{n,\,1})}{\gamma_n(y_{n,\,1})}\right)}{\mathbb{E}\left(\mathbb{P}_n\left(\xi(x_1)\prec y_{n,1}\right)\right)\cdot\mathbb{E}\left(\mathbb{P}_n\left(\xi(x_2)\prec y_{n,1}\right)\right)}. \tag{3.62}$$

An estimator of $\Lambda_n^{1,2}(x_1,x_2)$ can thus be written as

$$\widehat{\Lambda}_n^{1,2}(x_1,x_2) = \frac{\left(\sum_{i=1}^m \frac{\mathbb{P}_n(\xi(x_1)\prec y_{n,\,i})\cdot\mathbb{P}_n(\xi(x_2)\prec y_{n,\,i})}{\gamma_n(y_{n,\,i})^2}\right)}{\left(\sum_{i=1}^m \frac{\mathbb{P}_n(\xi(x_1)\prec y_{n,\,i})}{\gamma_n(y_{n,\,i})}\right)\left(\sum_{j=1}^m \frac{\mathbb{P}_n(\xi(x_2)\prec y_{n,\,j})}{\gamma_n(y_{n,\,j})}\right)} - \frac{1}{m}. \tag{3.63}$$

Therefore, the covariance between $\widehat{I}_n(x_1)$ and $\widehat{I}_n(x_2)$ can be approached using

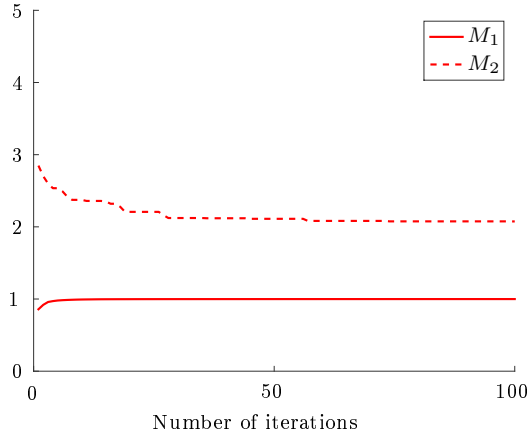
$$\begin{aligned}\mathrm{Cov}\left(\widehat{I}_n(x_1)\,,\,\widehat{I}_n(x_2)\right) \;\approx\; & \frac{\widehat{\Gamma}^2_{n,T}}{m^2}\left(\sum_{i=1}^{m}\frac{\mathbb{P}_n\left(\xi(x_1)\prec y_{n,i}\right)}{\gamma_n(y_{n,i})}\right)\left(\sum_{i=1}^{m}\frac{\mathbb{P}_n\left(\xi(x_2)\prec y_{n,i}\right)}{\gamma_n(y_{n,i})}\right) \\ & \left(\widehat{\Lambda}^{1,2}_n(x_1,x_2)+\left(1+\widehat{\Lambda}^{1,2}_n(x_1,x_2)\right)\cdot\widehat{\Delta}^2_{n,T}\right),\end{aligned} \tag{3.64}$$

where $\widehat{\Gamma}_{n,T}$ and $\widehat{\Delta}^2_{n,T}$ are obtained recursively using respectively (3.48) and (3.57) as before, and $\widehat{\Lambda}^{1,2}_n(x_1,x_2)$ is computed using (3.63).

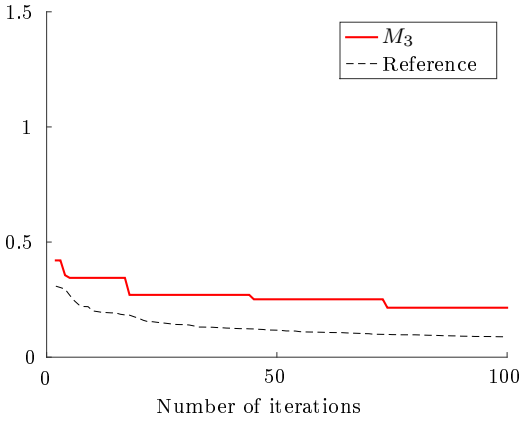
Notice that we plug the estimates $\widehat{I}_n$ of I_n in the expressions (3.59) of the variance and (3.64) of the covariance. When the estimates are of poor quality, this can lead to very unreliable results. The results of this section should thus be used with caution.

3.6.4 Experimental results for $p = 6$ and $p = 8$

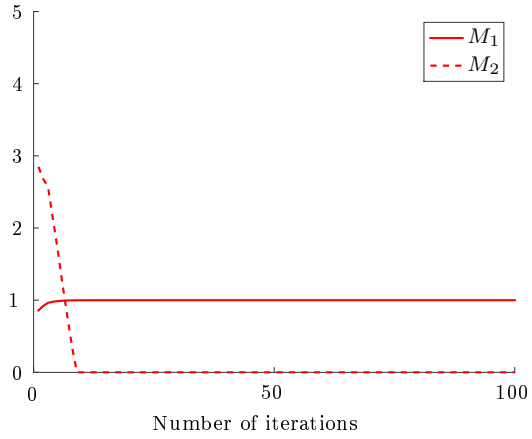
This section contains additional results that go along with Section 3.4. Results obtained when sequentially maximizing the hypervolume indicator on the FICUS problem with $r = 0.5$ and $c \in \{0.6, 1, 2\}$ are shown in Figures 3.27, 3.28 and 3.29 for $p = 6$ and in Figures 3.30, 3.31 and 3.32 for $p = 8$. Results obtained with BMOO on the FICUS problem with $r = 0.5$, $p = 8$ and $c \in \{0.6, 1, 2\}$ are shown in Figures 3.33 and 3.34.



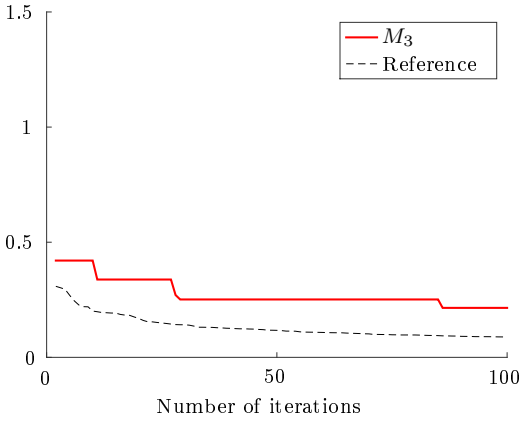
(a) $R = (0.5, 0.5, 0.5)$



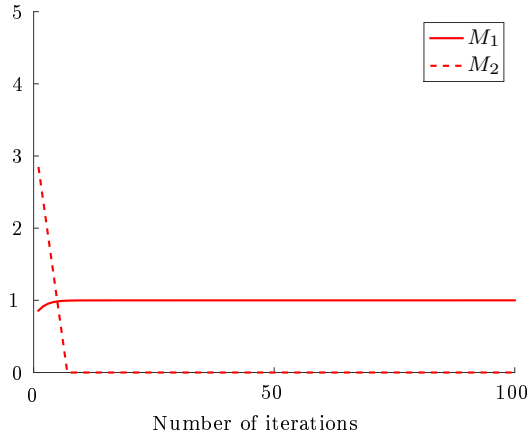
(b) $R = (0.5, 0.5, 0.5)$



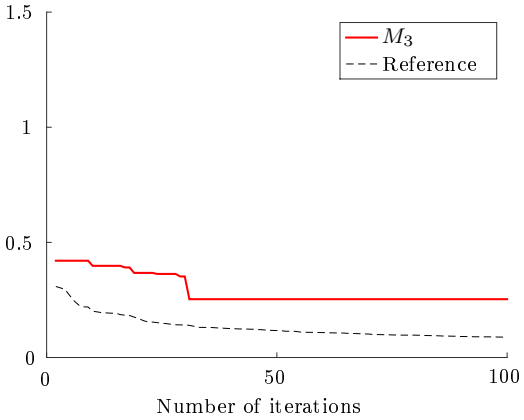
(c) $R = (1, 1, 1)$



(d) $R = (1, 1, 1)$

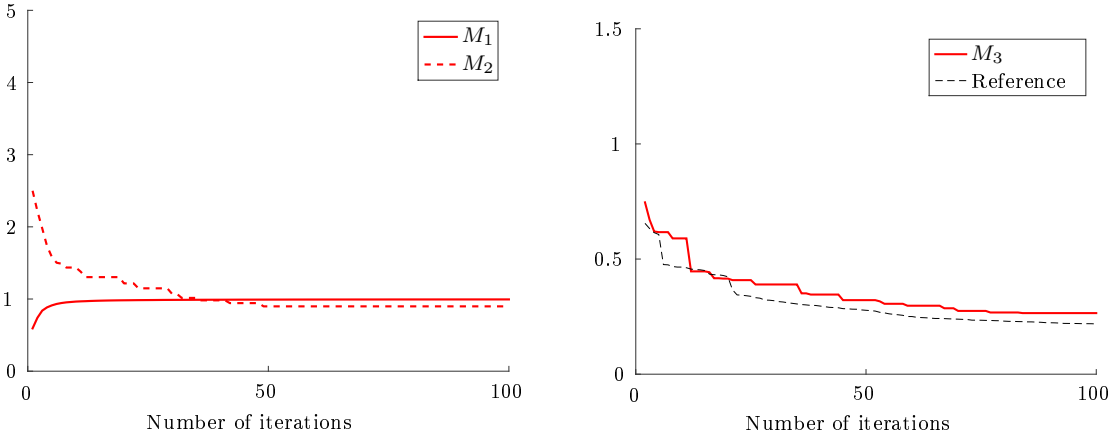


(e) $R = (50, 50, 50)$



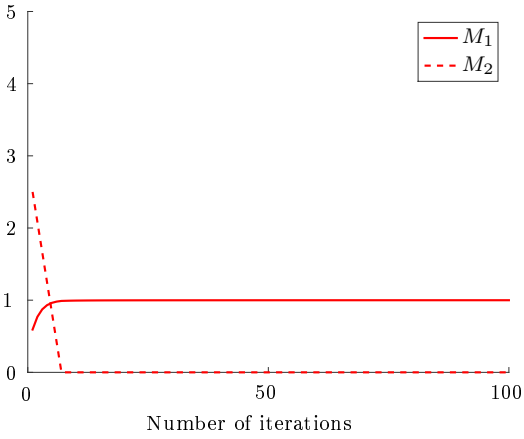
(f) $R = (50, 50, 50)$

Figure 3.27: Results obtained on the FICUS $(6, 0.5, 0.6)$ problem (convex front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.

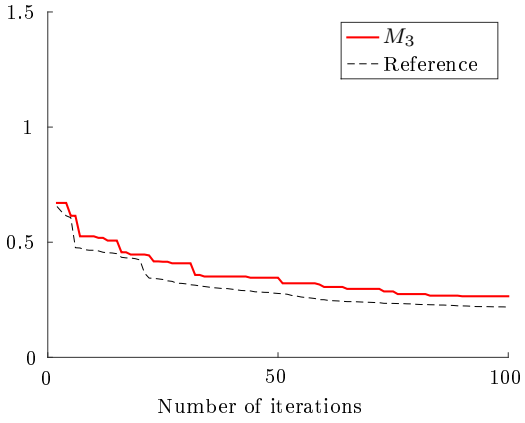


(a) $R = (0.5, 0.5, 0.5)$

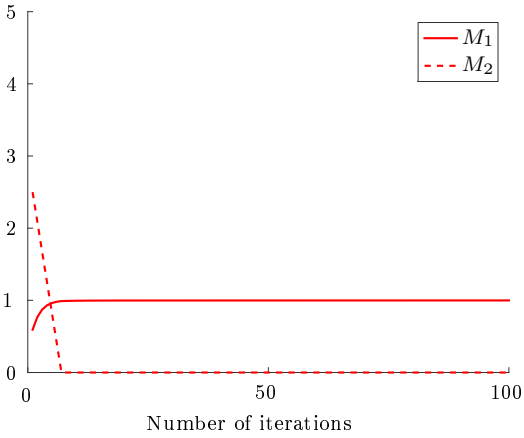
(b) $R = (0.5, 0.5, 0.5)$



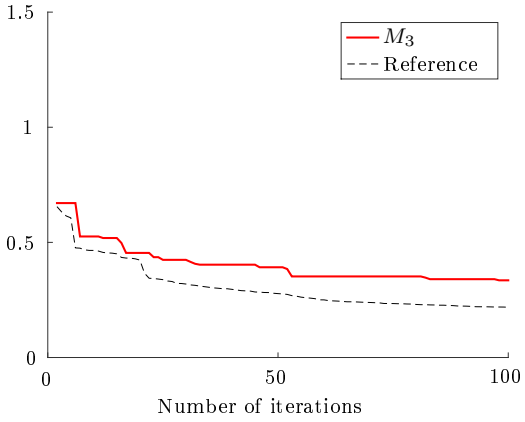
(c) $R = (1, 1, 1)$



(d) $R = (1, 1, 1)$

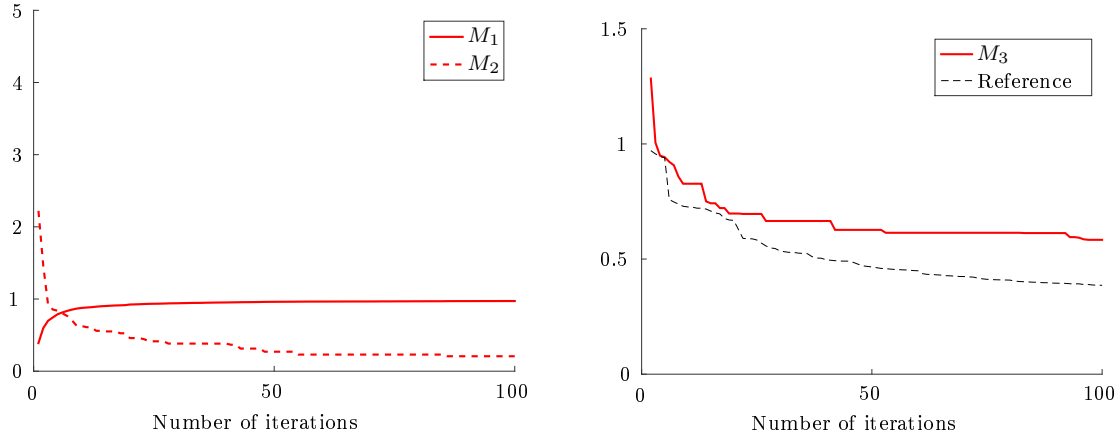


(e) $R = (50, 50, 50)$



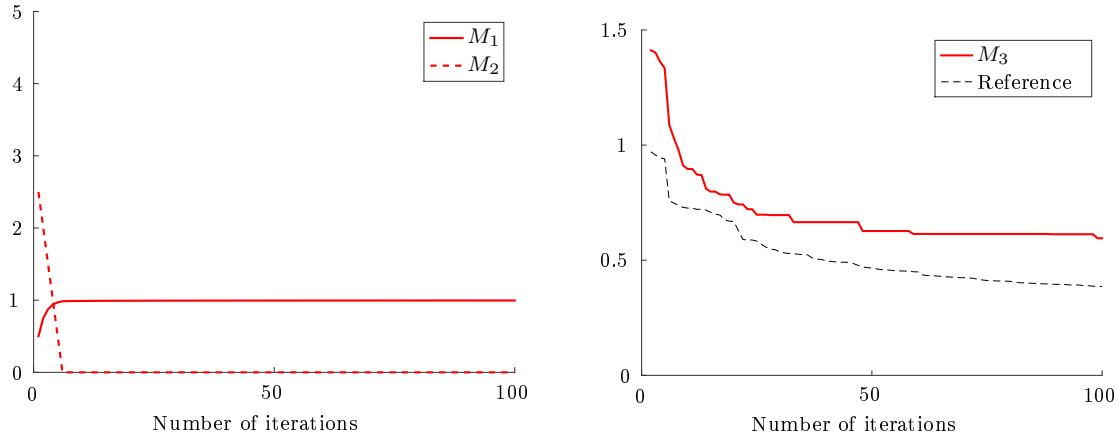
(f) $R = (50, 50, 50)$

Figure 3.28: Results obtained on the FICUS $(6, 0.5, 1)$ problem (linear front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



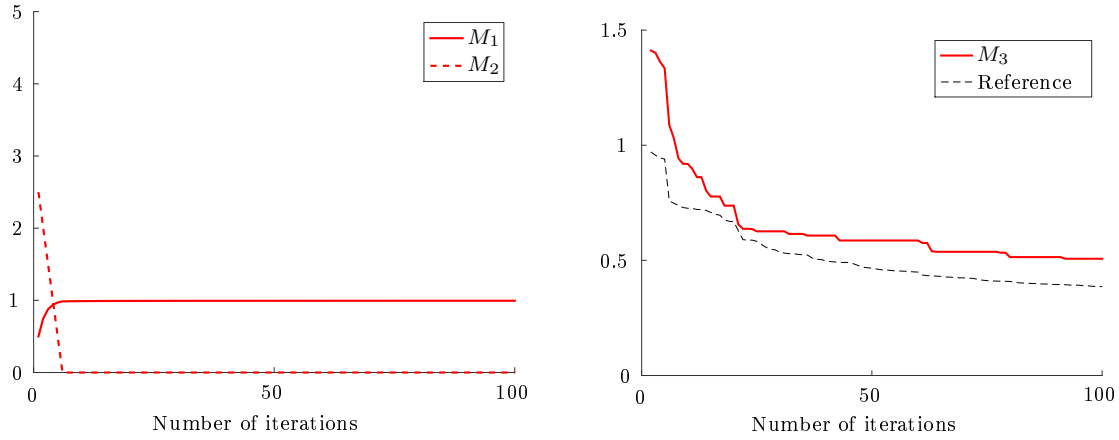
(a) $R = (0.5, 0.5, 0.5)$

(b) $R = (0.5, 0.5, 0.5)$



(c) $R = (1, 1, 1)$

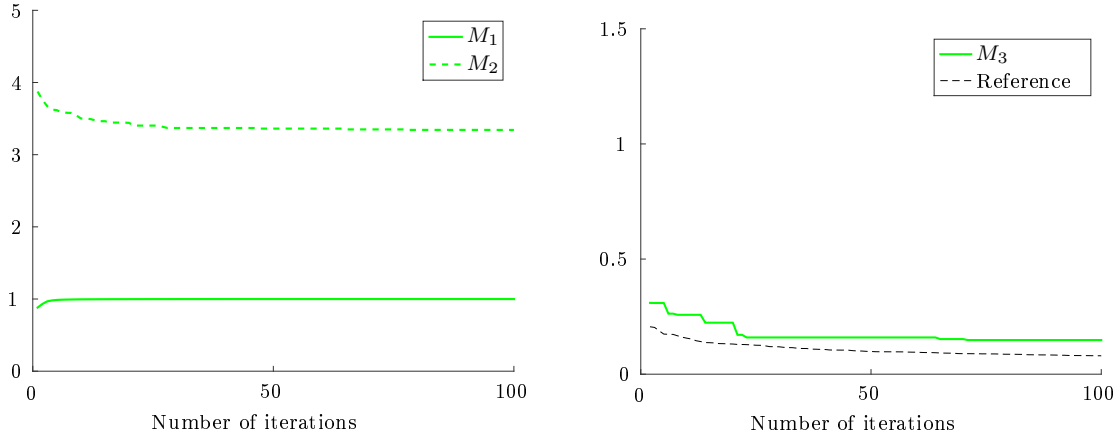
(d) $R = (1, 1, 1)$



(e) $R = (50, 50, 50)$

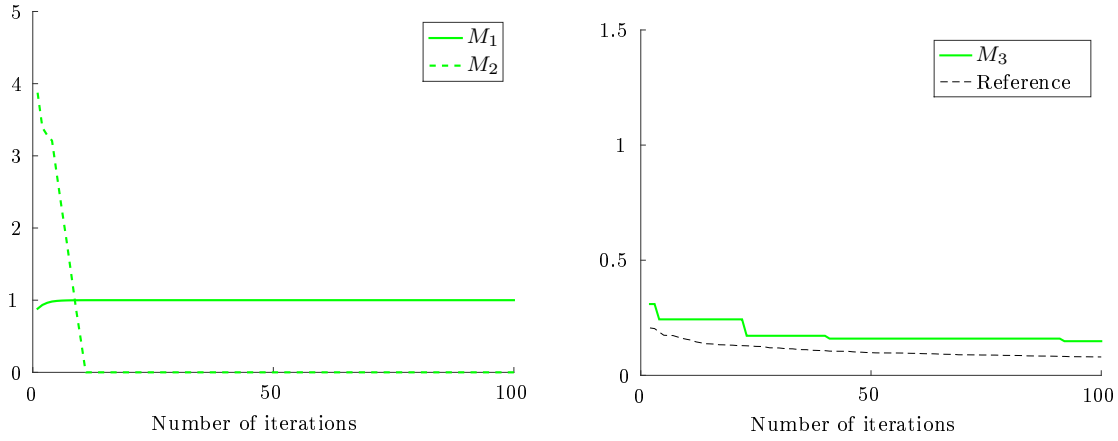
(f) $R = (50, 50, 50)$

Figure 3.29: Results obtained on the FICUS $(6, 0.5, 2)$ problem (concave front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



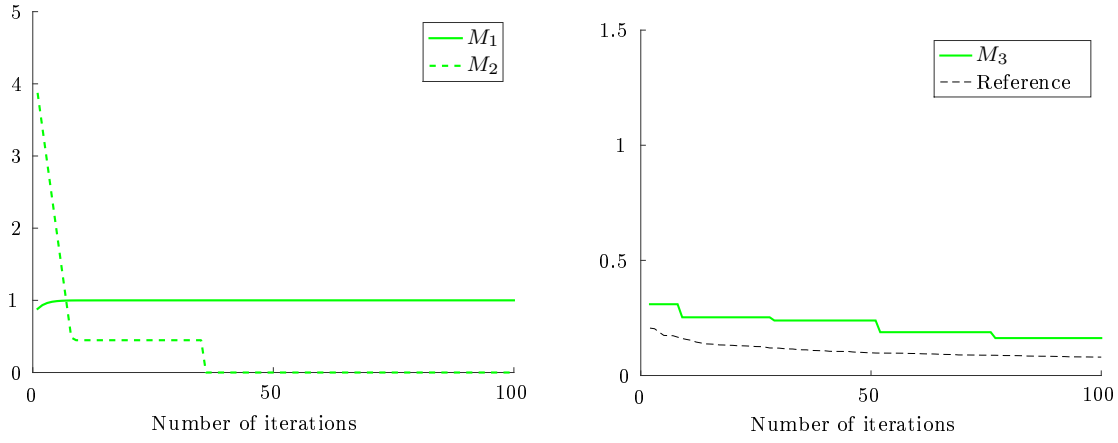
(a) $R = (0.5, 0.5, 0.5)$

(b) $R = (0.5, 0.5, 0.5)$



(c) $R = (1, 1, 1)$

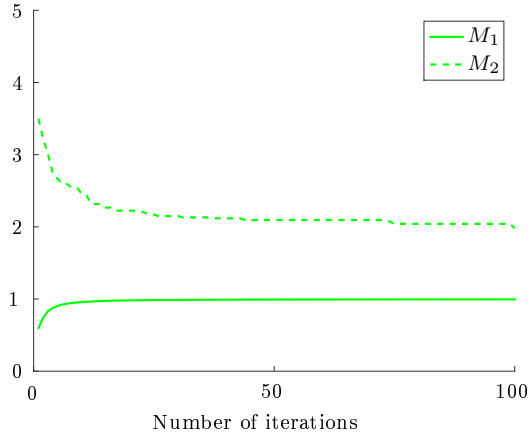
(d) $R = (1, 1, 1)$



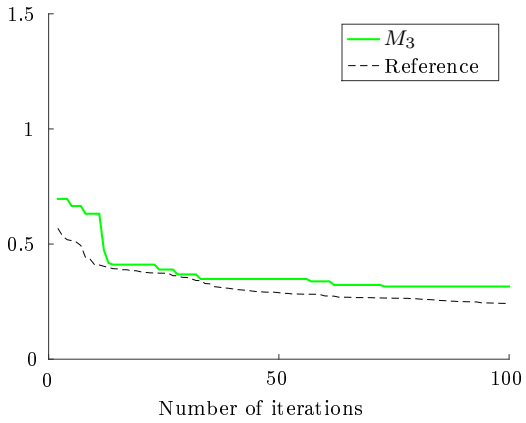
(e) $R = (50, 50, 50)$

(f) $R = (50, 50, 50)$

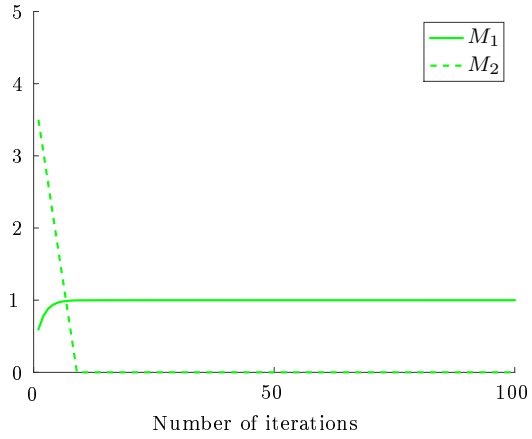
Figure 3.30: Results obtained on the FICUS $(8, 0.5, 0.6)$ problem (convex front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



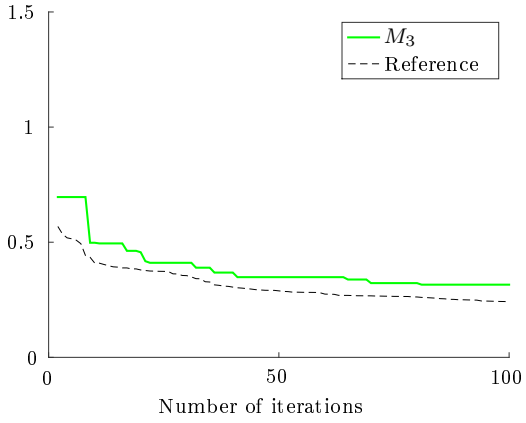
(a) $R = (0.5, 0.5, 0.5)$



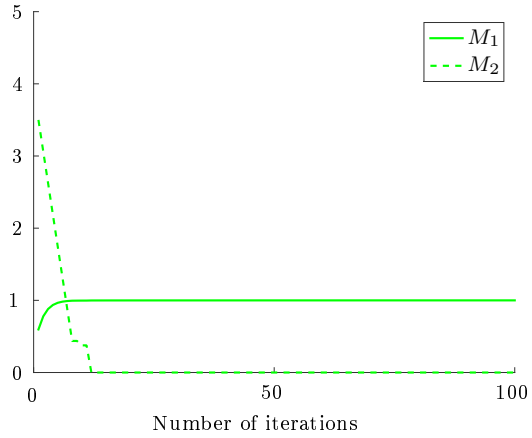
(b) $R = (0.5, 0.5, 0.5)$



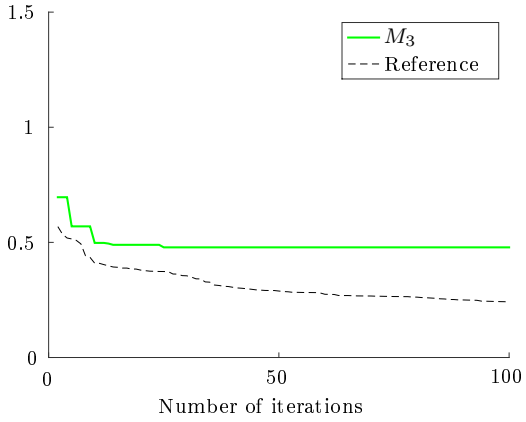
(c) $R = (1, 1, 1)$



(d) $R = (1, 1, 1)$

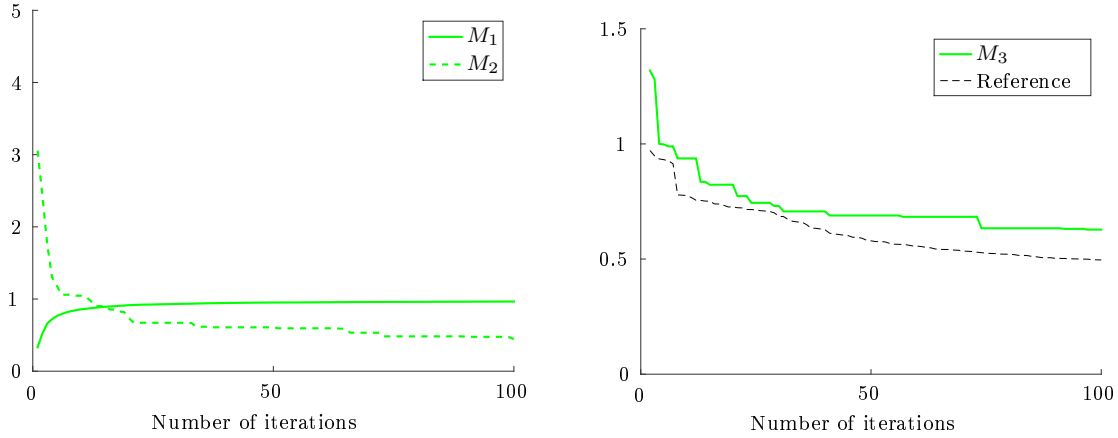


(e) $R = (50, 50, 50)$



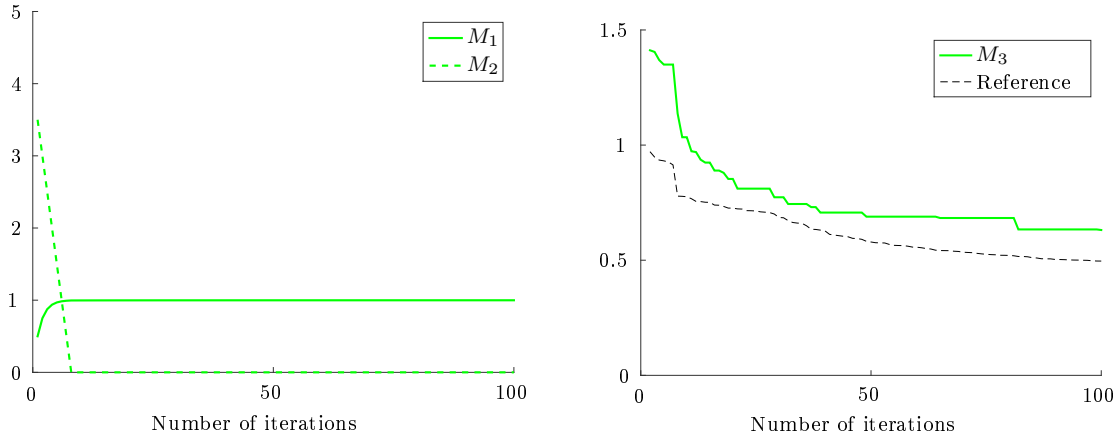
(f) $R = (50, 50, 50)$

Figure 3.31: Results obtained on the FICUS $(8, 0.5, 1)$ problem (linear front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



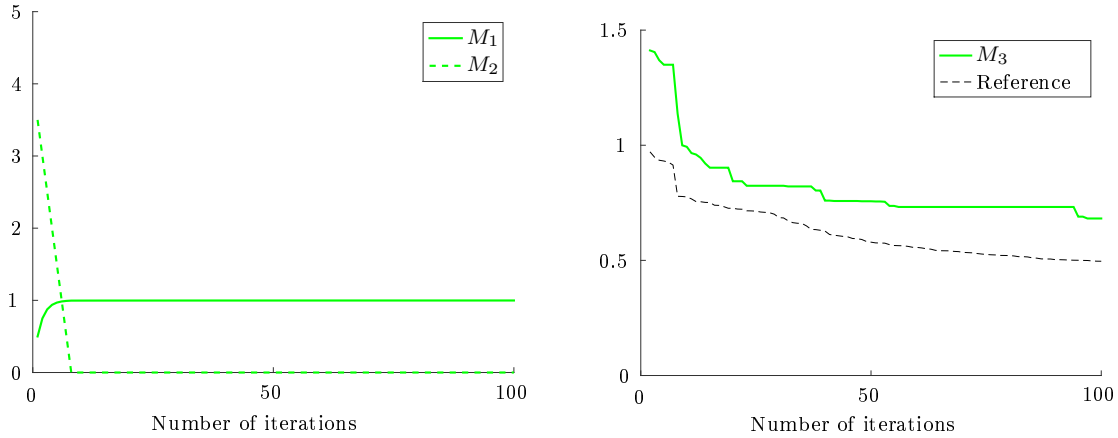
(a) $R = (0.5, 0.5, 0.5)$

(b) $R = (0.5, 0.5, 0.5)$



(c) $R = (1, 1, 1)$

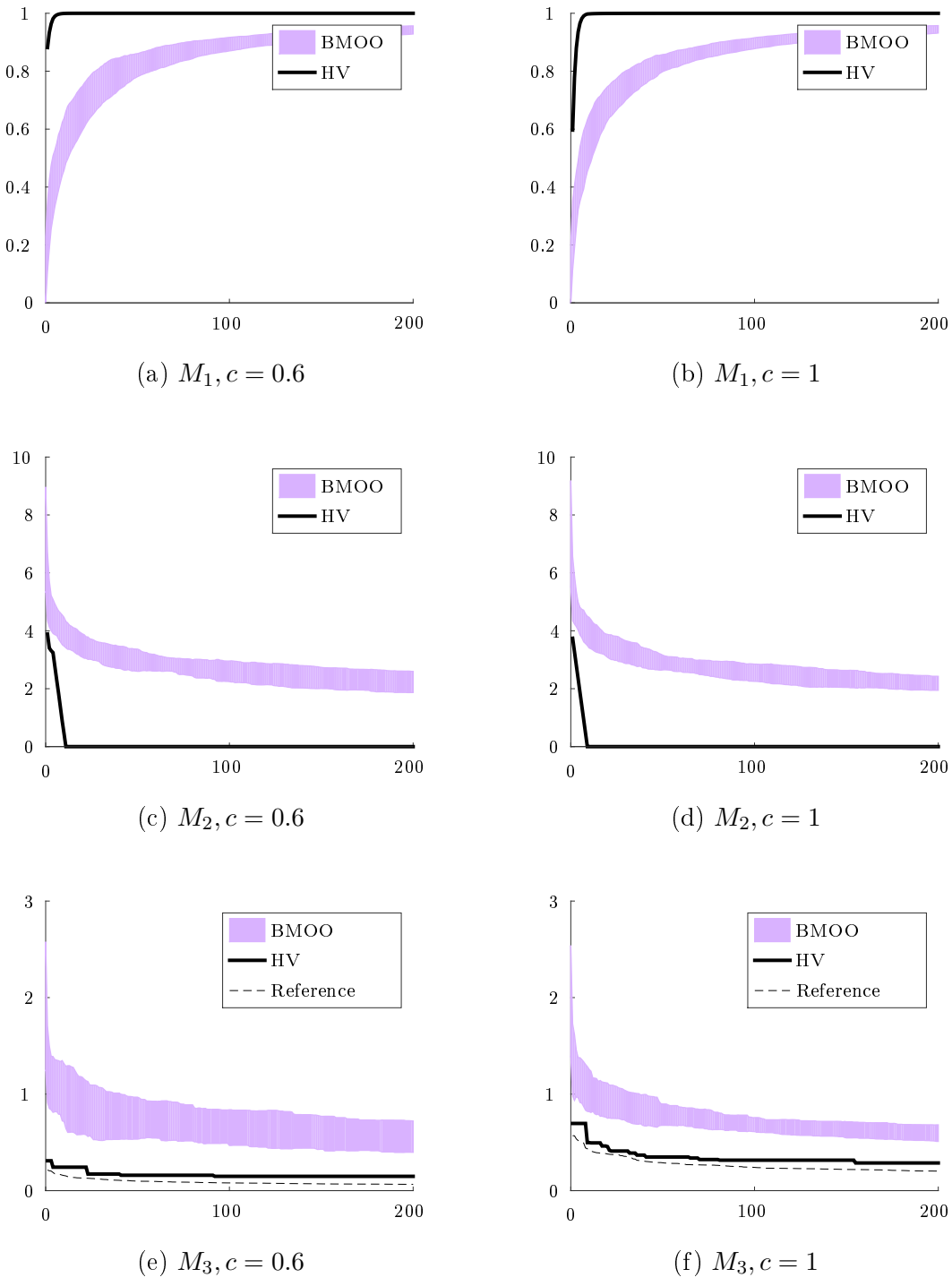
(d) $R = (1, 1, 1)$



(e) $R = (50, 50, 50)$

(f) $R = (50, 50, 50)$

Figure 3.32: Results obtained on the FICUS $(8, 0.5, 2)$ problem (concave front) by the optimization strategy where the hypervolume indicator is maximized sequentially using a large number of samples uniformly distributed on the Pareto front. The reference for the M_3 metric is shown as a dashed line.



(a) $M_1, c = 0.6$

(b) $M_1, c = 1$

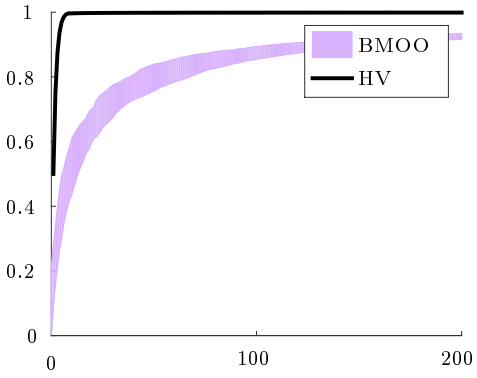
(c) $M_2, c = 0.6$

(d) $M_2, c = 1$

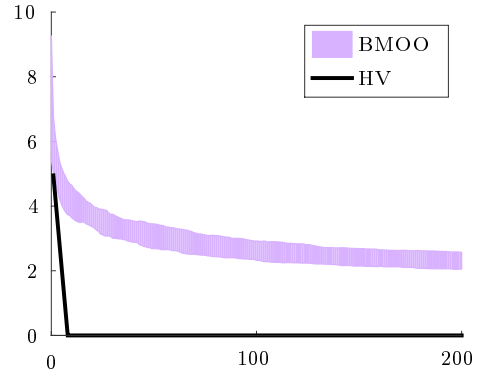
(e) $M_3, c = 0.6$

(f) $M_3, c = 1$

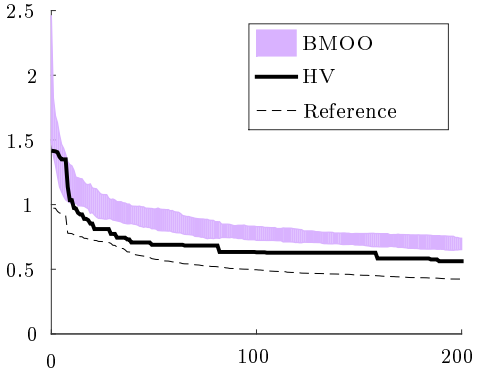
Figure 3.33: Results obtained by the BMOO algorithm on the FICUS $(8, 0.5, 0.6)$ problem (convex front, left column) and on the FICUS $(6, 0.5, 1)$ problem (linear front, right column). The reference for the M_3 metric is shown as a dashed line and the shaded region corresponds to a 95% confidence interval empirically computed from 30 runs of BMOO.



(a) $M_1, c = 2$



(b) $M_2, c = 2$



(c) $M_3, c = 2$

Figure 3.34: Results obtained by the BMOO algorithm on the FICUS $(8, 0.5, 2)$ problem (concave front). The reference for the M_3 metric is shown as a dashed line and the shaded region corresponds to a 95% confidence interval empirically computed from 30 runs of BMOO.

Chapter 4

Applications

4.1 Introduction

In this chapter, we present four real-life design optimization studies led using the BMOO algorithm and its extensions presented in Section 3.5.

The first study is on the design optimization of a commercial aircraft environment control system and is presented in Section 4.2. It is a reproduction of Feliot et al. (2016) with little modifications. This study was led within the framework of the Technological Research Institute SystemX (IRT SystemX), in collaboration with Airbus Group Innovation. The considered optimization problem has $d = 18$ design variables, $p = 2$ objectives and $q = 15$ constraints. It features a non-hypercubic design space defined by cheap-to-evaluate constraints and has hidden constraints.

The second study presented in this chapter is on the design of an electric vehicle powertrain and is presented in Section 4.3. It was also led within the framework of the IRT SystemX and is the fruit of a collaboration with Renault. The considered optimization problem has $d = 33$ design variables, $p = 2$ objectives and it features a non-hypercubic design space. For this study, the simulator implements numerical solving of Maxwell's equations using a finite elements method, which makes the functions of the problem expensive to evaluate. Moreover, it involves equality constraints. To handle these, a relaxation method is used to transform them into pairs of inequality constraints. As a consequence, the simultaneous satisfaction of all constraints is difficult for this problem because the feasible region is small.

The third study is on the tuning of a line of sight controller. It can be found in Section 4.4. This study was led at CentraleSupélec in collaboration with Sophie Frasnedo (CentraleSupélec, Safran Electronics & Denfense) and is based on her thesis work (Frasnedo et al., 2015a,b; Frasnedo, 2016). The optimization problem has $d = 7$ design variables, $q = 15$ constraints, and it features a non-hypercubic design space and hidden constraints. For this study, we consider two optimization problems: one with three objective functions and one with five objective functions. Also, we compare results obtained when the points are chosen one at a time and results obtained using batches of 50 experiments on a seven-objective formulation.

Finally, the fourth application is on the design of a turbomachine fan blade and is presented in Section 4.5. This study was led in collaboration with Cénaéro, and Safran Aircraft Engines. The problem is multi-physic. It involves mechanical, acoustic and aerodynamic computations. The underlying simulation chain relies on several softwares and takes approximately four hours to return a result. In this setting, it is highly desirable to propose design solutions using few functions evaluations. The formulation of the optimization that is considered in this study has $d = 26$ design variables, $p = 3$ objectives, $q = 9$ constraints and it features hidden constraints.

For all applications presented in this chapter except that of Section 4.2, the PICPI density is used for optimizing the expected improvement, as recommended in Section 3.2. For the computation of the criterion, as recommended in Section 3.3, we use the sequential Monte-Carlo approximation procedure with the L_2^{opt} density prior to finding feasible solutions and exact computation afterwards when the problem has less than five objectives. To compute the EI for the seven-objective formulation, the SMC procedure using the L_2^{opt} density is used during the

whole optimization process. When the problem has cheap-to-evaluate constraints, they are seen as restrictions of the design space, as discussed in Section 3.5.2. To handle hidden constraints, the procedure of Section 3.5.3 is used. Otherwise, the settings of the algorithm are as described in Section 2.5.1.

4.2 Design of a commercial aircraft environment control system

4.2.1 Introduction

The purpose of the environment control system (ECS) of a commercial aircraft is to provide a certain level of comfort to the passengers by regulating the temperature and pressure of the air injected into the cabin. The system is based on an inverse Brayton thermodynamic cycle. Hot and pressurized air is taken from the engines at the compressor stage through the bleed and ram air from the outside of the aircraft is used as coolant. For safety reasons, the hot air from the engines passes through a first heat exchanger where it is cooled down below the critical fuel ignition temperature. Then it is pressurized through a compressor and cooled again using a second heat exchanger. It then passes through a turbine where work is extracted to propel the compressor. The cooled and expanded air exiting the turbine is eventually mixed with hot air from the first heat exchanger outflow to reach the desired temperature and pressure before injection into the cabin.

The design of an optimal ECS is a complex problem in practice. It has been addressed in previous studies under different optimality conditions and modelling assumptions (see, e.g., Vargas and Bejan (2001); Bejan and Siems (2001); Pérez-Grande and Leo (2002)). In their article, Pérez-Grande and Leo (2002) study an aircraft-on-cruse scenario and propose a one dimensional model of the two heat exchangers. The system is designed in order to achieve minimal mass and entropy generation, two objectives that are shown to be antagonistic and which both affect the overall performance of the aircraft.

In this section, we extend their work by considering also the sizing of the rotating machines and by considering an aircraft-on-ground scenario, which corresponds to the most critical situation for the ECS in terms of cold production, and is therefore dimensioning. The design optimization of the system is performed using the BMOO algorithm (Feliot et al., 2017), which implements a Bayesian approach to the multi-objective optimization problem in the presence of non-linear constraints. The problem consists in finding an approximation of the set

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

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)).

The section is organized as follows. First we detail in Section 4.2.2 the model that is used to estimate the performances and main characteristics of the system. A one-dimensional analysis is

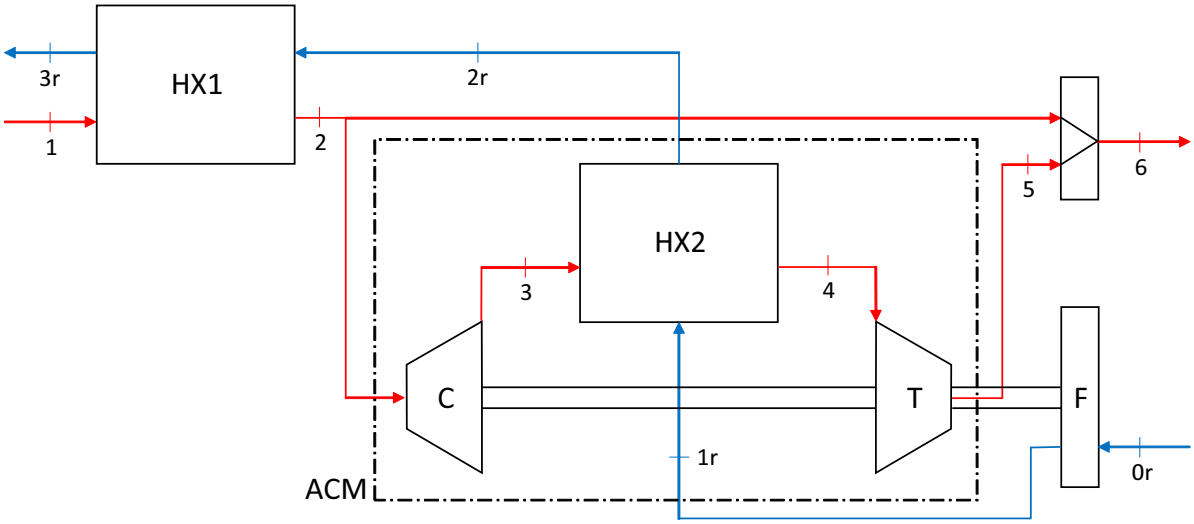


Figure 4.1: Architecture of the environment control system of a commercial aircraft

performed to establish the state equations of the system and link the physical values of interest to the geometrical parameters of the system components. Then we introduce in Section 4.2.3 the optimization algorithm that is used to conduct the system optimization. The results of the optimization are analyzed and possible directions for future work are discussed. Finally, conclusions are drawn in Section 4.2.4.

4.2.2 Thermodynamic analysis of the ECS

Sizing scenario

The architecture of the ECS is represented on Figure 4.1. Bleed air from the engines arrives into the system at location 1. Ram air from the outside of the aircraft is levied at location 0r and is used as coolant. The hot air enters a first heat exchanger where it is cooled down below the fuel ignition temperature. A by-pass at location 2 then permits to regulate the system by controlling the air flowrate entering the air cycle machine (ACM). Cooled and expanded air exits the ACM at location 5 and is mixed with warm air from the by-pass to reach the desired pressure and temperature before injection into the cabin at location 6.

In practice, the system must be able to satisfy strict specifications under different environmental conditions and operating situations. In this work, we consider a scenario where the aircraft is on ground, full of passengers, equipments running, and with an outside temperature of 50°C. In that situation, the ECS must be able to maintain the cabin temperature at $T_c = 24$°C. This scenario corresponds to the most demanding specification in terms of cold production, and is therefore dimensioning for the system. Formally, this means that the ECS must be able to dissipate enough heat to compensate for the thermal power $\mathcal{P}_{HT}$ produced by the passengers, the crew, the equipments and the environment:

$$\mathcal{P}_{HT} = \mathcal{P}_{out} + \mathcal{P}_{eq} + N_{pax}\mathcal{P}_{pax} + N_{crew}\mathcal{P}_{crew},$$

where $\mathcal{P}_{out}$ is the outside flow dissipation, $\mathcal{P}_{eq}$ is the thermal power produced by the equipments, $\mathcal{P}_{pax}$ and $\mathcal{P}_{crew}$ are the thermal powers produced by a passenger and by a crewmember and N_{pax} and N_{crew} are the number of passengers and the number of crewmembers in the aircraft.

In this scenario, the by-pass is wide open so that all the air from location 2 goes to the ACM. Also, there is no relative velocity between the aircraft and the ambient air when it is grounded and therefore there is no natural coolant flowrate. In this work, we consider a system where the ram flowrate is created by an auxiliary fan placed at the ram air entrance and powered by the turbine of the ACM. The sizing of this auxiliary fan is not taken into account in this study but we will assume that the ram flowrate can be controlled.

Heat exchangers

We now detail the model that is used to emulate the system. For the heat exchangers HX1 and HX2, we use a model from Pérez-Grande and Leo (2002). The two heat exchangers are compact cross-flow heat exchangers with unmixed fluids. For this kind of heat exchangers, the energy exchanged per unit time between the ram and bleed air can be formulated as:

$$\dot{m}c_p(T_{t1} - T_{t2}) = \dot{m}_r c_p(T_{t3r} - T_{t2r}), \tag{4.1}$$
$$\dot{m}c_p(T_{t3} - T_{t4}) = \dot{m}_r c_p(T_{t2r} - T_{t1r}), \tag{4.2}$$

where $\dot{m}$ and $\dot{m}_r$ denote respectively the bleed and ram air flowrates, c_p is the thermal capacity of the air and is assumed constant, and T_{ti} represents the stagnation temperature at location $i \in \{1, 2, 3, 4, 5, 1r, 2r, 3r\}$. Besides, we can define the efficiencies ϵ_1 and ϵ_2 of the two heat exchangers HX1 and HX2 as the ratio between the energy effectively exchanged and the total energy exchangeable:

$$c_p(T_{t1} - T_{t2}) = \epsilon_1 c_p(T_{t1} - T_{t2r}), \tag{4.3}$$
$$c_p(T_{t3} - T_{t4}) = \epsilon_2 c_p(T_{t3} - T_{t1r}). \tag{4.4}$$

Note that Eq.(3-4) only hold when $\dot{m} \leq \dot{m}_r$. Otherwise, $T_{t1} - T_{t2r}$ should be replace by $T_{t3r} - T_{t2r}$ in Eq.(3) and $T_{t3} - T_{t1r}$ should be replaced by $T_{t2r} - T_{t1r}$ in Eq.(4). The efficiencies ϵ_1 and ϵ_2 of the two heat exchangers depend on their geometry and we use the ϵ-Ntu model detailed in Pérez-Grande and Leo (2002) to estimate them.

In this study, the pressure drops as the air passes through the heat exchangers are considered constant:

$$P_{t2} - P_{t1} = \Delta P_{HX}, \tag{4.5}$$
$$P_{t4} - P_{t3} = \Delta P_{HX}, \tag{4.6}$$

where P_{ti} represents the stagnation pressure at location i and ΔP_{HX} is a constant pressure loss. While this permits to simplify the model, it is inaccurate because the pressure losses do also depend on the geometries of the heat exchangers and are responsible for a non-negligible

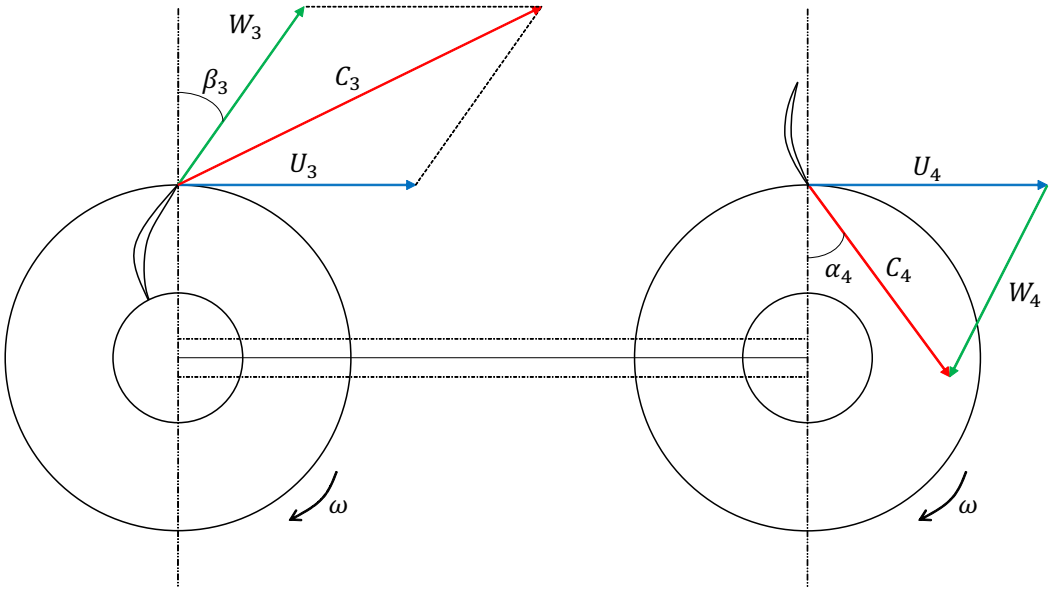


Figure 4.2: Compressor (left) and turbine (right) velocity triangles.

proportion of the entropy generated by the system. In particular, the frictions are expected to rise as the volume of the heat exchangers decreases, which further increases the necessary balance between entropy generation and mass. Again, the reader is referred to Pérez-Grande and Leo (2002) for a discussion on a possible model of the pressure drops.

Compressor and turbine

We consider a centrifugal compressor and an axial flow turbine. It is assumed that the air enters the compressor axially and exits parallel to the blades (the slip factor is neglected), with an angle β_3 as illustrated on Figure 4.2. Similarly, it is assumed that the air enters the turbine with an angle α_4 corresponding to the stator blades angle and goes out axially. The rotational speed of the shaft linking the compressor with the turbine is denoted ω. The letters C, U and W on Figure 4.2 denote respectively the air absolute velocity vector, the blade tip speed vector and the relative velocity vector, such that $C = U + W$. In the following, the subscripts u, m, and x will stand respectively for the tangential, meridional and axial components of the velocity vectors.

The power exchanged per unit time between the machines and the fluid and the change of momentum of the fluid are related by Euler's theorem as:

$$\begin{aligned} \dot{W}_C &= \dot{m}(U_3 C_{3u} - U_2 C_{2u}), \\ \dot{W}_T &= \dot{m}(U_5 C_{5u} - U_4 C_{4u}), \end{aligned}$$

where $\dot{W}_C$ and $\dot{W}_T$ denote respectively the power received by the fluid from the compressor and from the turbine. Note that with this convention $\dot{W}_T < 0$ and $\dot{W}_C > 0$. The turbine is converting part of the fluid energy into rotation speed, and the compressor is augmenting the energy of the fluid through its mechanical work, which increases the temperature of the fluid.

Under the assumption that the flow enters the compressor and exits the turbine axially, the

tangential components of the air velocity vectors are neglected and $C_{2u} = C_{5u} = 0$. The Euler theorem simplifies to:

$$\begin{aligned} \dot{W}_C &= \dot{m} U_3 C_{3u}, \\ \dot{W}_T &= -\dot{m} U_4 C_{4u}, \end{aligned}$$

The conservation of the mass at locations 3 and 4 gives the following additional equations, relating the air flowrate to its velocity and a control surface.

$$\begin{aligned} \dot{m} &= 2\pi\rho r_3 b_3 W_{3m}, \\ \dot{m} &= 2\pi\rho r_4 b_4 C_{4m}, \end{aligned}$$

where ρ is the air density (which is assumed constant), r_3 and r_4 are respectively the compressor outlet blade radius and turbine inlet blade radius, and b_3 and b_4 are respectively the compressor and turbine tip blade heights (see Figure 4.3). Using the velocity triangles of Figure 4.2, the Euler theorem can then be rewritten using the rotating machines geometries and rotational speed:

$$\dot{W}_C = \dot{m}\left(r_3^2\omega^2 - \frac{\dot{m}\tan(\beta_3)}{2\pi\rho b_3}\omega\right), \tag{4.7}$$

$$\dot{W}_T = -\frac{\dot{m}^2\tan(\alpha_4)}{2\pi\rho b_4}\omega. \tag{4.8}$$

Besides, the work extracted from the turbine is used to propel the compressor and the auxiliary fan. Writing down the conservation of energy per unit time we get:

$$\dot{W}_C + \dot{W}_T + \frac{1}{\eta_F}\frac{\dot{m}_r^3}{2\rho^2 A_r^2} = 0, \tag{4.9}$$

where A_r is a control surface at the ram air entrance and η_F is the ratio between the kinetic energy per unit time produced by the auxiliary fan and the power furnished by the turbine to the fan.

The powers $\dot{W}_C$ and $\dot{W}_T$ can also be expressed as functions of the stagnation temperatures by considering the change in total enthalpy of the fluid passing through the rotating machines (the other contributions are neglected):

$$\dot{W}_C = \eta_C \dot{m} c_p (T_{t3} - T_{t2}), \tag{4.10}$$

$$\dot{W}_T = \frac{1}{\eta_T}\dot{m} c_p (T_{t5} - T_{t4}), \tag{4.11}$$

where η_C and η_T are respectively the compressor and turbine isentropic efficiencies. Finally, the

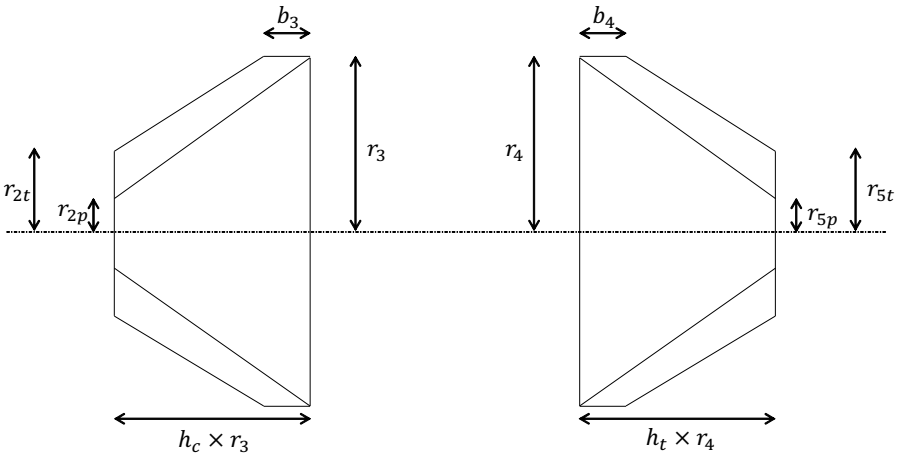


Figure 4.3: Geometrical parametrization of the compressor (left) and of the turbine (right).

stagnation pressure ratios are given by the isentropic relations:

$$\frac{P_{t3}}{P_{t2}} = \left(1 + \eta_C \frac{T_{t3} - T_{t2}}{T_{t2}}\right)^{\frac{\gamma}{\gamma-1}}, \tag{4.12}$$

$$\frac{P_{t5}}{P_{t4}} = \left(1 + \frac{1}{\eta_T} \frac{T_{t5} - T_{t4}}{T_{t4}}\right)^{\frac{\gamma}{\gamma-1}}, \tag{4.13}$$

where γ is the air isentropic coefficient.

Mass and entropy generation rate of the system

The mass of the system can be approximated by considering estimates of the volumes of its components and representative densities. For the two heat exchangers, we consider rectangular volumes and a representative density ρ_{HX} (see Table 4.3).

$$\begin{aligned} \mathcal{M}_{HX1} &= \rho_{HX} \cdot L_{x1} L_{y1} L_{z1}, \\ \mathcal{M}_{HX2} &= \rho_{HX} \cdot L_{x2} L_{y2} L_{z2}, \end{aligned}$$

where L_{x1}, L_{y1}, L_{z1}, L_{x2}, L_{y2}, L_{z2} are the heat exchangers dimensions (see Table 4.1), and $\mathcal{M}_{HX1}$ and $\mathcal{M}_{HX2}$ are the mass of the heat exchangers. For the compressor and turbine of the ACM, we consider separately the volumes of the blades and the volume of the machine body (see Figure 4.3):

$$\begin{aligned} V_{C,blade} &= e_c \left(\frac{h_c r_3 (r_3 - r_{2p})}{2} - \frac{(r_3 - r_{2t})(h_c r_3 - b_3)}{2} \right), \\ V_{C,body} &= \frac{\pi r_3^2 h_c (r_3 + r_{2p})}{3} - \frac{\pi h_c r_{2p}^3}{3}, \end{aligned}$$

for the compressor, and

$$\begin{aligned} V_{T,blade} &= e_t \left(\frac{h_t r_4 (r_4 - r_{5p})}{2} - \frac{(r_4 - r_{5t})(h_t r_4 - b_4)}{2} \right), \\ V_{T,body} &= \frac{\pi r_4^2 h_t (r_4 + r_{5p})}{3} - \frac{\pi h_t r_{5p}^3}{3}, \end{aligned}$$

for the turbine, where e_c and e_t are the compressor and turbine blades thickness, and h_c and h_t are aspect ratios. The mass of the system is then given by the following.

$$\begin{aligned} \mathcal{M} = \mathcal{M}_{HX1} + \mathcal{M}_{HX2} + \rho_{steel}(Z_C V_{C,blade} + V_{C,body}) \\ + \rho_{steel}(Z_T V_{T,blade} + V_{T,body}), \end{aligned} \quad (4.14)$$

where Z_C and Z_T are respectively the number of blades of the compressor and of the turbine.

The entropy generation rate of the ECS is the sum of the contributions along the bleed stream and along the ram stream, from entrance to exit:

$$\dot{\mathcal{S}} = \dot{m} \left(c_p \log \frac{T_5}{T_a} - R \log \frac{P_5}{P_a} \right) + \dot{m}_r \left(c_p \log \frac{T_{3r}}{T_a} - R \log \frac{P_{3r}}{P_a} \right), \quad (4.15)$$

where R is the perfect gas constant, T_a and P_a are the ambient temperature and pressure, and T_i and P_i are respectively the static temperature and the static pressure at location $i \in \{5, 3r\}$. The equations giving the static properties for the bleed stream are gathered in Table 4.3 in the additional material. For the ram stream, it is assumed that the Mach number remains low through the heat exchangers. Thus, $T_{3r} = T_{t3r}$, $T_{2r} = T_{t2r}$, $T_{1r} = T_{t1r}$. For the static pressures P_{2r} and P_{3r}, the law of perfect gas is used.

4.2.3 Optimization of the system

Formulation of the optimization problem

We consider an optimization problem using the 18 design variables given in Table 4.1. All other design parameters and physical properties are fixed (see Tables 4.4, 4.5 and 4.6 in Section 4.2.5). Under the model developed in Section 4.2.2, the ECS is thus ruled by a system of 13 equations (Eq. (4.1)–(4.13)) with 13 unknowns, which are the stagnation temperatures and pressures, the powers exchanged between the fluid and the compressor and turbine, and the rotational speed of the rotating ensemble: T_{t2}, T_{t3}, T_{t4}, T_{t5}, T_{t2r}, T_{t3r}, P_{t2}, P_{t3}, P_{t4}, P_{t5}, $\dot{W}_C$, $\dot{W}_T$ and ω. Eq.(4.14) and Eq.(4.15) give respectively the mass and the entropy generation rate, which are the objectives of the optimization. Additionally, we formulate the following 15 inequality

constraints (see Table 4.3):

$$\begin{cases} c_{1-2} & : & T_{min} & \leq & T_5 & \leq & T_{max}, \\ c_{3-4} & : & P_{min} & \leq & P_5 & \leq & P_{max}, \\ c_{5-6} & : & 0.5 & \leq & \epsilon_1 & \leq & 0.9, \\ c_{7-8} & : & 0.5 & \leq & \epsilon_2 & \leq & 0.9, \\ c_9 & : & \|C_2\| & \leq & 0.95\sqrt{\gamma R T_2}, & & \\ c_{10} & : & \|C_3\| & \leq & 0.95\sqrt{\gamma R T_3}, & & \\ c_{11} & : & \|C_4\| & \leq & 0.95\sqrt{\gamma R T_4}, & & \\ c_{12} & : & \|C_5\| & \leq & 0.95\sqrt{\gamma R T_5}, & & \\ c_{13} & : & r_3\omega & \leq & \sqrt{\gamma R T_3}, & & \\ c_{14} & : & r_4\omega & \leq & \sqrt{\gamma R T_4}, & & \\ c_{15} & : & \mathcal{P}_{\mathcal{HT}} & \leq & \dot{m} c_p \left(T_c - T_5\right). & & \end{cases}$$

The constraints c_1 to c_4 are standard specifications. The air injected into the cabin must lie between $T_{min} = 15°$ C and $T_{max} = 25°$ C and at a pressure close to the atmospheric pressure. Thus we take $P_{min} = 101.3$ kPa and $P_{max} = 1.05 P_{min}$. The constraints c_5 to c_8 are on the heat exchangers efficiencies. The design should be efficient enough but not too expensive to manufacture. The constraints c_9 to c_{12} are on the air velocity. In the model, we have assumed that the air density remains constant throughout the bleed stream, which is inaccurate if the flow becomes supersonic. We take a 5% margin to account for the possible variations of uncertain parameters and avoid numerical instabilities. Similarly, it is required via constraints c_{13} and c_{14} that the compressor and turbine blade tip speeds be subsonic. Constraint c_{15} stems from the sizing scenario considered in this study: The dissipated power must be greater than the power produced by the passengers, the crew, the equipments and the environment (see Section 4.2.2). Note that an equality constraint is not necessary because the constraint is expected to be active at the optima.

To ensure the feasibility of the system and avoid numerical issues, we enforce the following restrictions on the design variables (see Figure 4.3):

$$\begin{cases} d_1 & : & \dot{m} & \leq & \dot{m}_r, & & \\ d_2 & : & b_3 & \leq & h_c r_3, & & \\ d_3 & : & b_4 & \leq & h_t r_4, & & \\ d_{4-5} & : & r_{2p} + 0.02 & \leq & r_{2t} & \leq & r_3, \\ d_{6-7} & : & r_{5p} + 0.02 & \leq & r_{5t} & \leq & r_4, \\ d_8 & : & \Delta & \geq & 0, & & \\ d_9 & : & \frac{\tan(\beta_3)}{b_3} & \geq & -\frac{\tan(\alpha_4)}{b_4}. & & \end{cases}$$

where Δ in d_8 is the discriminant of Eq (4.7)–(4.9), seen as a second order polynomial equation in ω. The conditions d_8 and d_9 are necessary to ensure that there exists a real solution $\omega > 0$ to Eq (4.9). When two such solutions are possible, we take the largest one. Note that this parametrization implies that the optimization needs to be performed on a non-hypercubic design

Description	Not.	Domain
Bleed flowrate (kg.s^{-1})	$\dot{m}$	[2, 8]
Ram flowrate (kg.s^{-1})	$\dot{m}_r$	[2, 8]
Compressor outlet radius (m)	r_3	[0.1, 0.3]
Turbine inlet radius (m)	r_4	[0.1, 0.3]
Compressor inlet foot radius (m)	r_{2p}	[0.03, 0.1]
Turbine outlet foot radius (m)	r_{5p}	[0.03, 0.1]
Compressor inlet tip radius (m)	r_{2t}	[0.04, 0.2]
Turbine outlet tip radius (m)	r_{5t}	[0.04, 0.2]
Compressor outlet blade height (m)	b_3	[0.01, 0.1]
Turbine inlet blade height (m)	b_4	[0.01, 0.1]
Compressor outlet angle (rad)	β_3	$[-\frac{\pi}{3}, \frac{\pi}{3}]$
Turbine inlet angle (rad)	α_4	$[0, \frac{\pi}{3}]$
Heat exchanger 1: x length (m)	L_{x1}	[0.025, 0.7]
Heat exchanger 2: x length (m)	L_{x2}	[0.025, 0.7]
Heat exchanger 1: y length (m)	L_{y1}	[0.025, 0.7]
Heat exchanger 2: y length (m)	L_{y2}	[0.025, 0.7]
Heat exchanger 1: z length (m)	L_{z1}	[0.025, 0.7]
Heat exchanger 2: z length (m)	L_{z2}	[0.025, 0.7]

Table 4.1: Design variables description

domain.

Optimization algorithm

The optimization is performed using the BMOO algorithm (Feliot et al., 2017). This algorithm implements a Bayesian approach to the multi-objective optimization problem in the presence of non-linear constraints. The objectives and constraints of the problem are modeled using Gaussian process emulators (see, e.g., Williams and Rasmussen (2006)) and the algorithm performs a sequential optimization procedure where the next sample is chosen as the maximizer of an extended version of the expected improvement sampling criterion (see, e.g., Jones et al. (1998)). In practice, this requires to solve an auxiliary optimization problem at each iteration. The BMOO algorithm uses sequential Monte Carlo techniques to conduct this auxiliary optimization (see, e.g., Del Moral et al. (2006)). A population of candidate designs distributed according to a density of interest in the design space is sampled at each iteration and the maximizer of the extended expected improvement is chosen out of this population.

We take advantage of this to handle non-hypercubic design domains by truncating the density

of interest so as to propose only candidates that lie in the desired region. This is straightforward because sequential Monte Carlo methods do not require that the normalizing constant of the target density be known. For initialization, a pseudo maximin design of experiments on the non-hypercubic design domain (see e.g. Auffray et al. (2012)) can be achieved using rejection sampling. A large population of particles is sampled uniformly on the containing hypercube defined using the values of Table 4.1. The particles which do not respect the constraints d_1 to d_9 are then discarded and the population of surviving particles is pruned until the desired population size is reached. During the pruning step, particles that are too close to other particles are discarded, thus raising the maximin distance. Note that in practice, this requires that the volume of the design domain be not too small compared with the volume of the containing hypercube (the ratio of volumes was estimated close to 6% for this particular application).

Because the computation of the objectives and constraints values for a given design requires to solve the non-linear system formed by Eq.(4.1) to Eq.(4.13), it may happen that no solution can be found, in which case it is not possible to provide values of the constraints and objectives for the design under study. Also, some designs can lead to supersonic solutions for which the values of temperatures and pressures predicted by the model can be inaccurate. When this happens, we prefer to consider such designs as simulation failures and not use the values returned by the model. In the optimization procedure, this is taken into account in order to prevent the optimizer to explore regions where simulation failures are likely, by multiplying both the sampling criterion of BMOO and the density in the sequential Monte Carlo procedure by a probability of observability. This technique has been proposed by Lee and co-authors Lee et al. (2011). A statistical model is learned on the observed/non-observed data and provides a probability of satisfying the hidden constraints leading to simulation failures. In this work, a nearest-neighbours classifier using 5 neighbours and the L_2 distance is used to that purpose.

Optimization results

The algorithm is run with a limiting budget of $N_{max} = 500$ calls to the simulation model, and an initial design of $N_{init} = 90$ samples. The set of optimal trade-off solutions found by the algorithm is shown on Figure 4.4.

Among the initial design of experiments, 44 experiments led to simulation failures and 92 additional failures occurred during the optimization process. Further investigation revealed that most of the simulation failures occurred because the flow was supersonic in the compressor, which happens with high probability when the bleed flowrate is high and the compressor radii are low. Regarding the constraints satisfaction, no feasible observations were made in the initial sample and the algorithm found one after 25 iterations.

The design parameters associated to 7 trade-off solutions chosen along the Pareto front are given in Table 4.2. Several observations can be made on these results. First we note that the bleed flowrate remains constant along the front. This is because c_{15} is active (see Section 4.2.3) and $T_5 = T_{min}$ for optimal designs, which forces the value of the bleed flowrate. The ram flowrate is less constrained and varies along the front. We note its strong influence on the entropy generation rate (see Eq.(15)). The variation of the mass on the other hand mostly comes from the variations

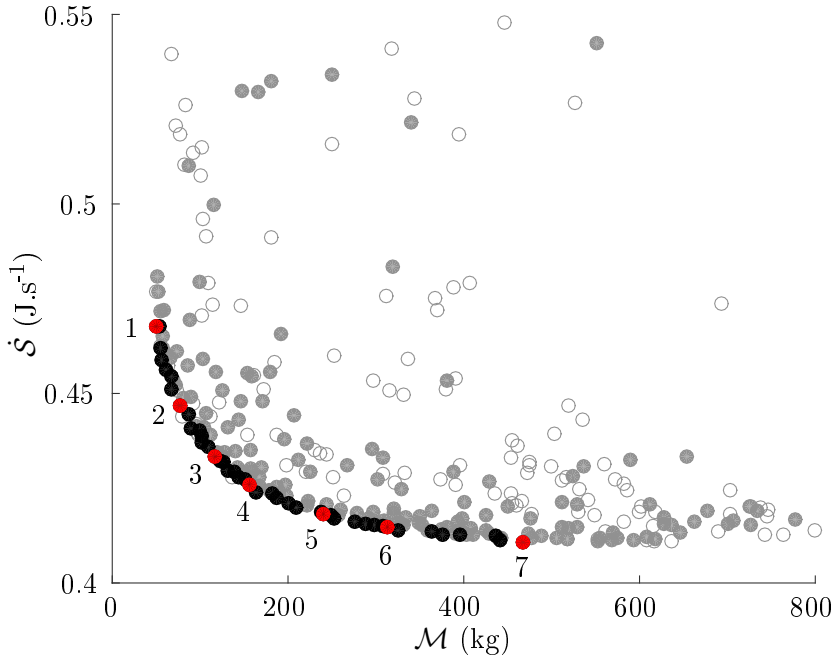


Figure 4.4: Pareto front obtained with the BMOO optimizer using 500 samples. Empty circles are non-feasible solutions. Grey disks are feasible but dominated solutions. Black and red disks are feasible and non-dominated solutions.

of L_{z1} and L_{z2}. As the heat exchangers height is raised, the entropy generation rate is lowered but the mass augments. The values of L_{x1}, L_{y1}, L_{x2}, and L_{y2} are set close to their maximal values, which permits to achieve efficiencies between 0.7 and 0.8. Note that the pressure losses are assumed constant in this study. Further work is required to better understand their impact on the entropy generation rate when the heat exchangers dimensions become small. Regarding the turbine and compressor dimensions, they are set as small as possible, which keeps the mass low and augments the fluid velocity, thus achieving good performances.

4.2.4 Conclusions

In this section, a one dimensional model of the environment control system of a commercial aircraft is proposed. The model permits to emulate the behaviour of the system when the geometries of its components vary, for a scenario where the aircraft is on ground, full of passengers, equipments running, and with an outside temperature of 50°C. The system is optimized using the BMOO algorithm, which implements a Bayesian approach to the multi-objective optimization problem in the presence of non-linear constraints, and trade-off design solutions in terms of mass and entropy generation rate of the system are identified.

As a particularity, the optimization is performed on a non-hypercubic design domain and involves hidden constraints. This is a situation that is often encountered in engineering design optimization. The BMOO algorithm is successfully adapted to this new setup, which makes it possible to conduct a multi-objective optimization using a reasonable number of calls to the numerical simulation model.

	1	2	3	4	5	6	7
$\dot{m}$	2.95	2.92	2.94	2.94	2.94	2.95	2.94
$\dot{m}_r$	**7.74**	**6.86**	**5.63**	**5.06**	**4.64**	**4.40**	**4.27**
r_{2p}	0.07	0.05	0.05	0.03	0.03	0.07	0.04
r_{2t}	0.10	0.08	0.08	0.08	0.06	0.09	0.10
r_3	0.10	0.11	0.10	0.10	0.12	0.12	0.13
b_3	0.01	0.01	0.05	0.05	0.04	0.02	0.03
β_3	0.36	0.74	0.97	-0.16	0.61	0.94	0.48
r_{5p}	0.03	0.03	0.03	0.03	0.03	0.03	0.03
r_{5t}	0.05	0.05	0.05	0.05	0.05	0.05	0.05
r_4	0.10	0.10	0.11	0.12	0.11	0.10	0.11
b_4	0.02	0.02	0.04	0.02	0.04	0.03	0.03
α_4	1.04	0.50	0.89	1.01	0.44	0.79	0.30
L_{x1}	0.67	0.65	0.68	0.68	0.63	0.69	0.70
L_{y1}	0.65	0.68	0.61	0.67	0.67	0.66	0.65
L_{z1}	**0.03**	**0.04**	**0.07**	**0.12**	**0.17**	**0.20**	**0.32**
L_{x2}	0.66	0.69	0.66	0.66	0.70	0.68	0.69
L_{y2}	0.69	0.53	0.68	0.65	0.65	0.68	0.65
L_{z2}	**0.03**	**0.06**	**0.09**	**0.10**	**0.17**	**0.25**	**0.36**
$\mathcal{M}$	49.78	77.13	117.00	156.57	240.03	312.40	466.69
$\dot{\mathcal{S}}$	0.47	0.45	0.43	0.43	0.42	0.41	0.41

Table 4.2: Optimal design variables values found by the optimization algorithm for the points 1 to 7 (in red) of Figure 4.4. The values of the most influential variables are in bold.

The BMOO algorithm is primarily designed to address problems where the computational time associated to the model evaluation is high, which is not the case here. In this study, most of the computational time required to conduct the optimization was taken by the optimizer and more work is needed to make the implementation of the algorithm more efficient. Nevertheless, the algorithm achieves very satisfactory results and is a competitive algorithm to address multi-objective optimization problems with several constraints.

4.2.5 Additional material

Static temperatures	Static pressures	Fluid velocities
$T_5 = T_{t5} - \frac{C_{5x}^2}{2}$	$\frac{P_{t5}}{P_5} = \left(1 + \frac{\gamma-1}{2}\left(\frac{C_{5x}}{\gamma R T_5}\right)^2\right)^{\frac{\gamma}{\gamma-1}}$	$C_{5x} = \frac{\dot{m}}{\pi\left(r_{5t}^2 - r_{5p}^2\right)\rho}$
$T_4 = T_{t4} - \frac{C_{4m}^2}{2}$	$\frac{P_{t4}}{P_4} = \left(1 + \frac{\gamma-1}{2}\left(\frac{C_{4m}}{\gamma R T_4}\right)^2\right)^{\frac{\gamma}{\gamma-1}}$	$C_{4m} = \frac{\dot{m}}{2\pi r_4 b_4 \rho \cos\alpha_4}$
$T_3 = T_{t3} - \frac{C_{3m}^2}{2}$	$\frac{P_{t3}}{P_3} = \left(1 + \frac{\gamma-1}{2}\left(\frac{C_{3m}}{\gamma R T_3}\right)^2\right)^{\frac{\gamma}{\gamma-1}}$	$C_{3m} = \sqrt{\left(\frac{r_3\omega - \dot{m}\tan\beta_3}{2\pi r_3 b_3 \rho}\right)^2 + \left(\frac{\dot{m}}{2\pi r_3 b_3 \rho}\right)^2}$
$T_2 = T_{t2} - \frac{C_{2x}^2}{2}$	$\frac{P_{t2}}{P_2} = \left(1 + \frac{\gamma-1}{2}\left(\frac{C_{2x}}{\gamma R T_2}\right)^2\right)^{\frac{\gamma}{\gamma-1}}$	$C_{2x} = \frac{\dot{m}}{\pi\left(r_{2t}^2 - r_{2p}^2\right)\rho}$

Table 4.3: Static properties equations

Description	Not.	Value
Ambient temperature (K)	T_a	323
Ambient pressure (Pa)	P_a	101.3e3
Number of passengers	N_{pax}	120
Number of crewmembers	N_{crew}	5
Thermal power passengers (W)	$\mathcal{P}_{pax}$	70
Thermal power crew (W)	$\mathcal{P}_{crew}$	100
Thermal power equipments (W)	$\mathcal{P}_{eq}$	4800
Outside flow dissipation (W)	$\mathcal{P}_{out}$	3000
Bleed temperature (K)	T_1	473
Bleed pressure (Pa)	P_1	260e3
Pressure losses (Pa)	ΔP_{HX}	40e3
Valve opening	θ	0
Ram stream cross surface (m^2)	A_r	0.20
Fan efficiency	η_F	0.95
Air specific heat ($J.kg^{-1}.K^{-1}$)	c_p	1004
Air isentropic coefficient	γ	1.4
Perfect gaz constant ($J.kg^{-1}.K^{-1}$)	R	287

Table 4.4: Simulation parameters values used in the experiments of Section 4.2.3

Description	Not.	Value
Viscosity bleed ($kg.m^{-1}.s^{-1}$)	μ	2.28e-5
Viscosity ram ($kg.m^{-1}.s^{-1}$)	μ_r	2.28e-5
H.T. ratio bleed stream (m^{-1})	β	2231
H.T. ratio ram stream (m^{-1})	β_r	1115
Plate spacing bleed stream (m)	b	5.21e-3
Plate spacing ram stream (m)	b_r	12.3e-3
Prandtl number bleed stream	Pr	0.7
Prandtl number ram stream	Pr_r	0.7
Hydraulic diameter bleed (m)	Dh	1.54e-3
Hydraulic diameter ram (m)	Dh_r	3.41e-3
Convection length bleed (m)	λ	0.035
Convection length ram (m)	λ_r	0.035
Representative density ($kg.m^{-3}$)	ρ_{HX}	1415
Fin thickness (m)	δ	0.102e-3
Wall thickness (m)	t_w	6e-4
Thermal conductivity ($W.m^{-1}.K^{-1}$)	k_w	237

Table 4.5: Heat exchangers parameters values used in the experiments of Section 4.2.3

Description	Not.	Value
Compressor adiabatic efficiency	η_c	0.8
Turbine adiabatic efficiency	η_t	0.92
Compressor aspect ratio	h_c	0.7
Turbine aspect ratio	h_t	0.5
Compressor blades thickness (m)	e_c	0.01
Turbine blades thickness (m)	e_t	0.01
Compressor number of blades	Z_c	21
Turbine number of blades	Z_t	21

Table 4.6: Compressor and turbine parameters values used in the experiments of Section 4.2.3

4.3 Design of an electric vehicle powertrain

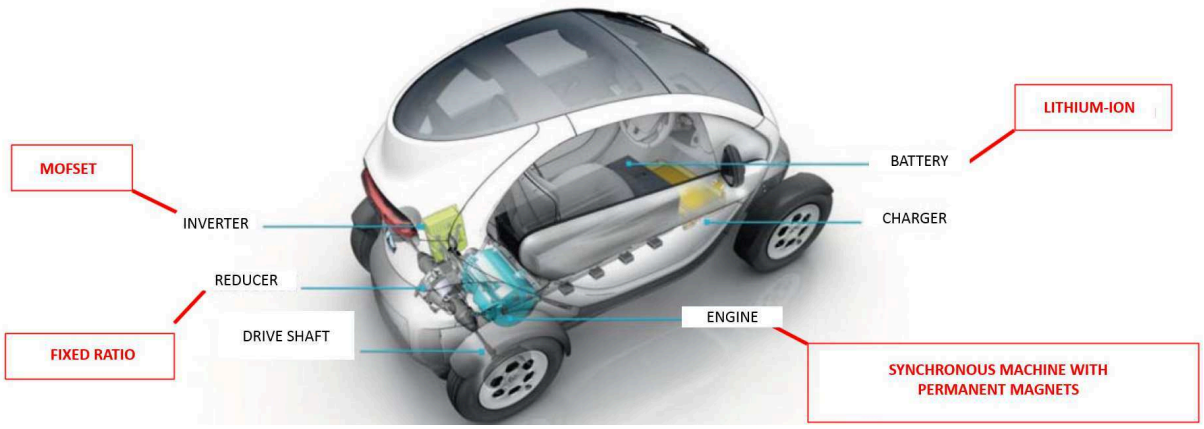


Figure 4.5: Architecture of the Twizzy powertrain.

4.3.1 Introduction

This study deals with the design of the powertrain of an electric vehicle. It was led within the framework of the Technological Research Institute SystemX in collaboration with Renault.

The vehicle that is considered is a Twizzy, and we place ourselves in the situation of Renault in the early stages of the development of the Twizzy 2. The architecture and the technological choices are imposed—they are the same as for the first version of the vehicle (see Figure 4.5)—and the objective is to assess the possibility of improving certain characteristics of the original vehicle, such as its autonomy for example, early in the design process, i.e. using simple numerical models.

The structure of the section is as follows. First, we present the specifications of the study in Section 4.3.2. Then, in Section 4.3.3, we detail the numerical model that was developed to meet the requirements of the study. The formulation of an optimization problem and the optimization results are presented in Section 4.3.4. Finally, conclusions are drawn in Section 4.3.5.

4.3.2 Specifications

Customer use cycles

In this study, we consider data that is representative of a customer's typical use of the vehicle in urban and in jammed traffic situations. The data takes the form of recorded speed over time and is shown in Figure 4.6. For the comfort of the driver, it is required that the vehicle be able to mimic these driving cycles.

Let $t = (t_1, \dots, t_n)$ and $v = (v_1, ..., v_n)$ denote respectively the measurements times and measured speeds for one of the two driving cycles (urban or traffic jam). The acceleration of the vehicle over time can be computed using a first order finite differences approximation:

$$a_i = \frac{v_{i+1} - v_i}{t_{i+1} - t_i}, \quad i \in [\![1, n-1]\!]. \tag{4.16}$$

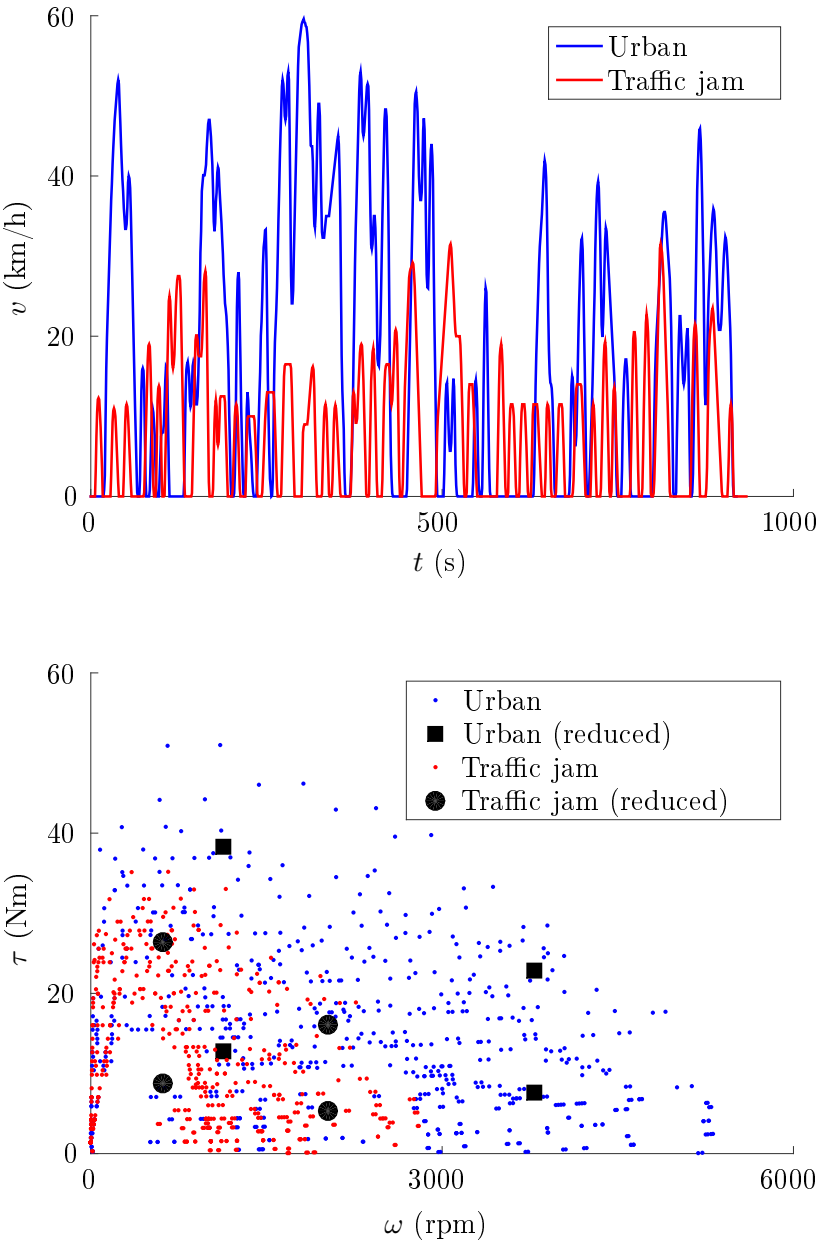


Figure 4.6: Customer use data in urban situation and in jammed traffic situation.

Using Newton's second law, the force F_i that is necessary to produce the acceleration a_i, $i \in [\![1, n-1]\!]$, is then

$$F_i = \mathcal{M}_{\text{veh}} \left(K \cos(\text{asin}(\alpha)) + g\alpha\right) + \mathcal{M}_{\text{veh}} a_i + \frac{1}{2}\rho C_x v_i^2, \tag{4.17}$$

where $\mathcal{M}_{\text{veh}}$ is the mass of the vehicle, K is a coefficient of friction with the ground, α is the slope of the road, g is the gravity acceleration, ρ is the air density and C_x is the drag coefficient of the vehicle (see Table 4.11).

Denote $I = \{i \in [\![1, n-1]\!] \,;\, a_i \geq 0\}$ the set of the indices that correspond to a positive acceleration. For every $i \in I$, the force F_i and the vehicle speed v_i can be linked respectively to the torque τ_i produced by the electric machine and to its rotation speed ω_i:

$$\left\{ \begin{array}{rcl} \tau_i & = & \dfrac{R_w F_i}{r \eta_{\text{red}}}, \\ \omega_i & = & \dfrac{30}{\pi} \dfrac{r v_i}{R_w}, \end{array} \right. \tag{4.18}$$

where R_w is the vehicle's wheels radius, η_{red} is the efficiency of the reducer and r is its reduction ratio. Using this transformation, the raw data (t, v) can be represented in a diagram (ω, τ), as illustrated in Figure 4.6.

The specifications can thus be formulated as $|I|$ equality constraints of the form

$$\Gamma(\omega_i, \xi_i) = \tau_i \,, \quad i \in I \,, \tag{4.19}$$

where ξ_i, $i \in I$, denotes the set of parameters other than the rotation speed that influence the torque Γ produced by the electric machine. These will be detailed in Section 4.3.3.

For every $i \in I$, denote $\mathcal{P}_i$ the power associated to the data point (ω_i, τ_i):

$$\mathcal{P}_i = \frac{2\pi}{60} \omega_i \tau_i \,, \quad i \in I \,. \tag{4.20}$$

Further denote

$$\left\{ \begin{array}{rcl} \omega_{\max} & = & \max_{i \in I} \omega_i \,, \\ \tau_{\max} & = & \max_{i \in I} \tau_i \,, \\ \mathcal{P}_{\max} & = & \max_{i \in I} \mathcal{P}_i \,, \end{array} \right. \tag{4.21}$$

and introduce

$$\omega_{\text{base}} = \frac{30}{\pi} \frac{\mathcal{P}_{\max}}{\tau_{\max}}. \tag{4.22}$$

To reduce the number of constraints (4.19), in this work, we consider, for each cycle, only the four representative points defined in Table 4.7. The eight points thus defined are represented in Figure 4.6. In the following, we shall denote respectively Λ_{urban} and Λ_{jam} the sets made of these points for both driving cycles.

ω	τ
$\frac{\omega_{\text{base}}}{2}$	$\frac{\tau_{\max}}{4}$
$\frac{\omega_{\text{base}}+\omega_{\max}}{2}$	$\frac{15\mathcal{P}_{\max}}{\pi(\omega_{\text{base}}+\omega_{\max})}$
$\frac{\omega_{\text{base}}}{2}$	$\frac{3\tau_{\max}}{4}$
$\frac{\omega_{\text{base}}+\omega_{\max}}{2}$	$\frac{45\mathcal{P}_{\max}}{\pi(\omega_{\text{base}}+\omega_{\max})}$

Table 4.7: Representative points in the diagram (ω, τ).

Additional requirements

In addition to the driving cycles, the specifications impose the following requirements:

- The maximum speed on flat ground must be greater than 85 km/h.
- The maximum speed on a 5% slope must be greater than 65 km/h.
- The hill start acceleration for a 20% slope must be greater than 1.5 m/s^2.
- The vehicle must be able to go from 0 to 45 km/h in less than 3 s.
- The autonomy of the vehicle should be greater than 100 km.

The first four requirements can be formulated as inequality constraints on the torque of the electric machine using (4.17) and (4.18). As regards the constraint on the autonomy, it can be written as:

$$3.6\frac{\kappa}{F_{\max}} \geq 100\,, \tag{4.23}$$

where $F_{\max} = \max_{i\in I} F_i$ and κ is the capacity of the battery (see Section 4.3.3).

To summarize, the specifications for this study take the form of eight equality constraints of the form (4.19) with $(\omega, \tau) \in \Lambda_{\text{urban}}$ and $(\omega, \tau) \in \Lambda_{\text{jam}}$ plus four inequality constraints on the torque of the electric machine and one inequality constraint on the autonomy of the vehicle given by (4.23).

4.3.3 Numerical model

Battery

The battery is a lithium-ion battery. It is characterised by its number n_s of cells in series and its number n_p of cells in parallel. From these, we compute the following quantities:

Number of cells	:	$n_c = n_s n_p$,
Vacuum voltage (V)	:	$E = E_0 n_s$,
Capacity (Wh)	:	$\kappa = n_c \frac{\kappa_0 E_0}{1000}$,
Resistance (Ω)	:	$R = R_0 \frac{n_s}{n_p}$,

where R_0 is the internal resistance of a cell, E_0 is the vacuum voltage of a cell and κ_0 is the capacity of a cell (see Table 4.9). For this study, we take $n_s = 22$ and $n_p = 2$. Note however that these could be optimized as well.

Inverter and reducer

In our model, the inverter is characterised by its maximum effective current $I_{\max}^{\text{eff}}$, its voltage factor per phase k and its efficiency η_{inv}. The reducer is characterised by its reduction ratio r and its efficiency η_{red}. Both r and $I_{\max}^{\text{eff}}$ are considered as design variables. The efficiencies of the inverter and of the reducer; and the voltage factor per phase of the inverter, are fixed parameters (see Tables 4.10 and 4.12).

Under this model, the maximum effective voltage $U_{\max}^{\text{eff}}$ that can be applied to the machine and the maximum available power $\mathcal{P}_{\max}^{\text{eff}}$ are given by:

$$\left\{ \begin{array}{rcl} U_{\max}^{\text{eff}} & = & kE\,, \\ \mathcal{P}_{\max}^{\text{eff}} & = & \frac{3 I_{\max}^{\text{eff}} U_{\max}^{\text{eff}}}{1000}\,. \end{array} \right. \tag{4.24}$$

Electric machine

The magnetic torque produced by the electric machine is a function of its geometry, its rotation speed, and of the intensities of the current passing through the coils. To parametrize its geometry, we use the following seven design variables:

L	:	Length of the machine.
r_{rot}	:	Rotor inner radius.
ϵ_{rot}	:	Rotor thickness.
ϵ_{gap}	:	Air gap.
ϵ_{mag}	:	Magnet thickness.
α_{mag}	:	Magnet angular spread.
α_{notch}	:	Opening ratio between notch and dent gap.

The geometry of the machine thus parametrized is illustrated in Figure 4.7. To help the understanding of the parametrization, the following radii are shown in the figure:

$$\left\{ \begin{array}{rcl} R_1 & = & r_{\text{rot}}\,, \\ R_2 & = & R_1 + \epsilon_{\text{rot}}\,, \\ R_3 & = & R_2 + \epsilon_{\text{mag}}\,, \\ R_4 & = & R_3 + \epsilon_{\text{gap}}\,, \\ R_5 & = & R_4 + \epsilon_{\text{mouth}}\,, \\ R_6 & = & \sqrt{\frac{S_{\text{notch}} + R_5^2 \sin\left(\frac{\beta_{\text{notch}}}{2}\right)}{\left(1 + \frac{\pi}{2}\sin\left(\frac{\beta_{\text{notch}}}{2}\right)\right)\sin\left(\frac{\beta_{\text{notch}}}{2}\right)}}\,, \\ R_7 & = & R_6\left(1 + \sin\left(\frac{\beta_{\text{notch}}}{2}\right)\right) + \epsilon_{\text{rot}}\,, \end{array} \right. \tag{4.25}$$

where ϵ_{mouth} is the mouth depth and β_{notch} and S_{notch} are respectively the angular spread and the total section of a notch.

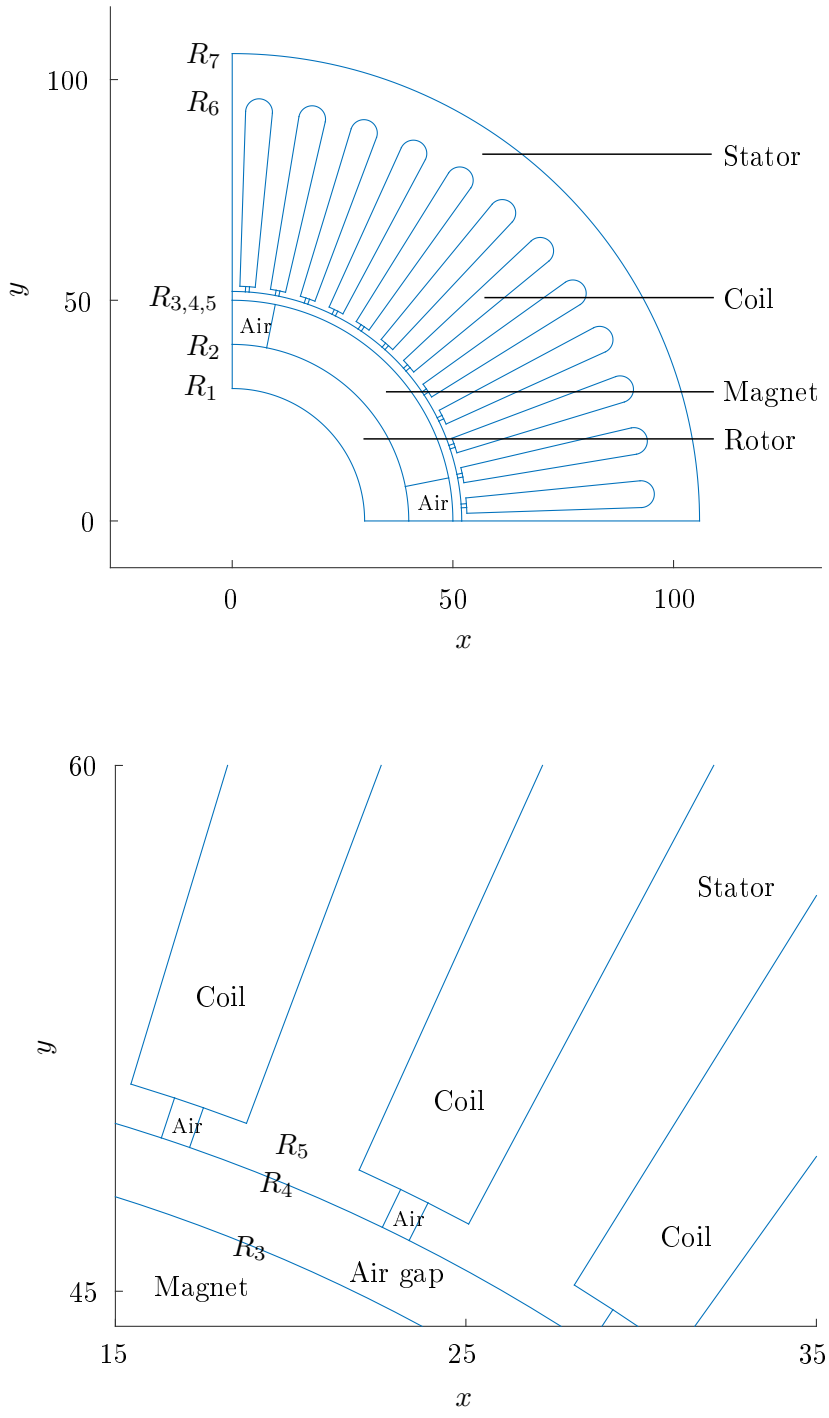


Figure 4.7: Geometry of the electric machine.

These are computed using

$$\begin{cases} \beta_{\text{notch}} &= \alpha_{\text{notch}} \frac{2\pi}{2N_p N_n}\,, \\ S_{\text{notch}} &= \frac{1}{\tau_{\text{copper}}} \left(\frac{I_{\max}^{\text{eff}}}{J_{\max}}\right) N_w\,, \end{cases} \tag{4.26}$$

where N_p is the number of pairs of poles, N_n is the number of notches per pole, τ_{copper} is the notch copper ratio, $J_{\max}$ is the maximum current density and N_w is the number of electric wires per notch (see Table 4.13).

To compute the magnetic torque Γ produced by the electric machine and the iron losses L_ϕ, we use a generalized Bertotti model (Bertotti, 1985):

$$\begin{cases} \Gamma(\omega, i_d, i_q) = N_p \left[\phi_d(i_d, i_q) i_q - \phi_q(i_d, i_q) i_d\right] - L_\phi(\omega, i_d, i_q)\,, \\ L_\phi(\omega, i_d, i_q) = \frac{2\pi\omega}{60} \left[\alpha_{\text{hys}}(i_d, i_q) + \alpha_{\text{exc}}(i_d, i_q)\sqrt{\frac{2\pi\omega}{60}} + \alpha_{\text{fou}}(i_d, i_q)\frac{2\pi\omega}{60}\right]\,, \end{cases} \tag{4.27}$$

where (d, q) is a referential attached to the rotor (Park transform; see, e.g., Cardoso et al. (1999)), ϕ_d and ϕ_q are the components of the magnetic flux in the referential (d, q), i_d and i_q are the components of the current intensity in the referential (d, q), and α_{hys}, α_{exc} and α_{fou} are loss coefficients associated respectively to the loss by hysteresis, excess and Foucault currents.

Note the dependence of Γ and L_ϕ to the current intensities i_d and i_q and to the rotation speed ω of the machine. For better performances, the currents must be chosen as a function of ω (recall that the specifications impose constraints on the torque for twelve different values of ω) to minimize the losses and maximize the torque, which is an optimal control problem. For this study, the choice was made to consider this auxiliary optimization problem as part of the main optimization problem. In other words, the currents i_d and i_q for the twelve values of ω that are considered (see Section 4.3.2) will be design variables, which augments the dimension of the problem of 24. This is done to lower the computational cost of the simulation and to avoid having to use an auxiliary optimization procedure, which may result in simulation failures in cases where it does not converge.

To compute the magnetic flux ϕ_d and ϕ_q and the loss coefficients α_{hys}, α_{exc} and α_{fou}, it is necessary to solve Maxwell's equations (see Figure 4.8). To this end, we use a finite element method implemented in the FEMM open source solver (Baltzis, 2008). This step is responsible for most of the computing time of the simulation chain. Besides, for each tested geometry, it has to be done twelve times for the different values of ω, i_d and i_q. For better performances, these computations are performed in parallel in our implementation.

In addition to the constraints placed on the torque for the differents values of ω, i_d and i_q, constraints on the voltages and intensities are required to ensure the feasibility of the system:

$$\begin{cases} \sqrt{\frac{u_d^2+u_q^2}{3}} + R_{\text{ext}}\sqrt{\frac{u_d^2+u_q^2}{3}} &\leq U_{\max}^{\text{eff}}\,, \\ \sqrt{\frac{i_d^2+i_q^2}{3}} &\leq I_{\max}^{\text{eff}}\,, \end{cases} \tag{4.28}$$

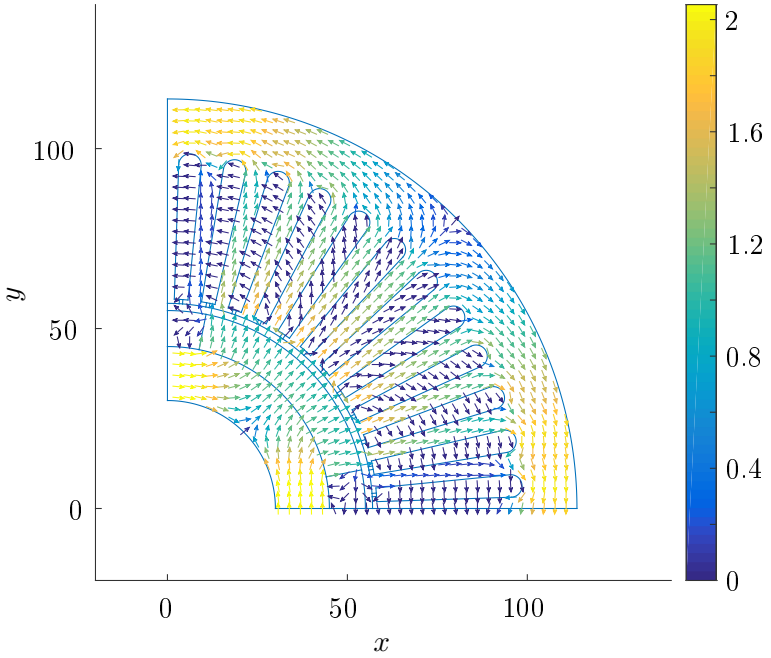


Figure 4.8: Illustration of the magnetic flux passing through the machine in the absence of currents. These are obtained using a finite element method to solve Maxwell's equations.

where R_{ext} is the external resistance of the machine:

$$R_{\text{ext}} = 3k^2 \frac{R}{\eta_{\text{inv}}}, \tag{4.29}$$

and u_d and u_q are voltages associated to the intensities i_d and i_q:

$$\begin{cases} u_d &= Ri_d - N_p \omega \phi_q(i_d, i_q), \\ u_q &= Ri_q + N_p \omega \phi_d(i_d, i_q). \end{cases}$$

Masses and costs

To compute the mass $\mathcal{M}_{\text{veh}}$ of the vehicle and the cost $\mathcal{C}_{\text{pt}}$ of the powertrain, it is necessary to compute the masses and costs of their components:

$$\begin{cases} \mathcal{M}_{\text{veh}} &= \mathcal{M}_{\text{wpt}} + \mathcal{M}_{\text{bat}} + \mathcal{M}_{\text{inv}} + \mathcal{M}_{\text{red}} + \mathcal{M}_{\text{em}}, \\ \mathcal{C}_{\text{pt}} &= \mathcal{C}_{\text{bat}} + \mathcal{C}_{\text{inv}} + \mathcal{C}_{\text{red}} + \mathcal{C}_{\text{em}}, \end{cases} \tag{4.30}$$

where $\mathcal{M}_{\text{wpt}}$, $\mathcal{M}_{\text{bat}}$, $\mathcal{M}_{\text{inv}}$, $\mathcal{M}_{\text{red}}$ and $\mathcal{M}_{\text{em}}$ are respectively the mass of the vehicle without the powertrain, the mass of the battery, the mass of the inverter, the mass of the reducer and the mass of the electric machine, and $\mathcal{C}_{\text{bat}}$, $\mathcal{C}_{\text{inv}}$, $\mathcal{C}_{\text{red}}$ and $\mathcal{C}_{\text{em}}$ are respectively the cost of the battery, the cost of the inverter, the cost of the reducer and the cost of the electric machine.

For the battery, the inverter and the reducer, we use simple linear models:

$$\left\{\begin{array}{rcl}\mathcal{M}_{\text{bat}} &=& \lambda_{\text{bat,mass}} M_0 n_c\,,\\ \mathcal{C}_{\text{bat}} &=& \lambda_{\text{bat,cost,1}}\kappa + \lambda_{\text{bat,cost,2}} n_c\,,\end{array}\right. \tag{4.31}$$

for the battery,

$$\left\{\begin{array}{rcl}\mathcal{M}_{\text{inv}} &=& \lambda_{\text{inv,mass,1}}\mathcal{P}^{\text{eff}}_{\text{max}} + \lambda_{\text{inv,mass,2}}\,,\\ \mathcal{C}_{\text{inv}} &=& \lambda_{\text{inv,cost,1}}\mathcal{P}^{\text{eff}}_{\text{max}} + \lambda_{\text{inv,cost,2}}\,,\end{array}\right. \tag{4.32}$$

for the inverter and

$$\left\{\begin{array}{rcl}\mathcal{M}_{\text{red}} &=& \lambda_{\text{red,mass,1}}\mathcal{P}^{\text{eff}}_{\text{max}} + \lambda_{\text{red,mass,2}}\,,\\ \mathcal{C}_{\text{red}} &=& \lambda_{\text{red,cost,1}}\mathcal{P}^{\text{eff}}_{\text{max}} + \lambda_{\text{red,cost,2}}\,,\end{array}\right. \tag{4.33}$$

for the reducer, where $\lambda_{\text{bat,mass}}$, $\lambda_{\text{inv,mass,1}}$, $\lambda_{\text{inv,mass,2}}$, $\lambda_{\text{red,mass,1}}$ and $\lambda_{\text{red,mass,2}}$ are mass coefficients, $\lambda_{\text{bat,cost,1}}$, $\lambda_{\text{bat,cost,2}}$, $\lambda_{\text{inv,cost,1}}$, $\lambda_{\text{inv,cost,2}}$, $\lambda_{\text{red,cost,1}}$ and $\lambda_{\text{red,cost,2}}$ are cost coefficients and M_0 is the mass of a cell (see Tables 4.9, 4.10 and 4.12).

For the electric machine, it is necessary to compute the mass $\mathcal{M}_{\text{copper}}$ of copper, the mass $\mathcal{M}_{\text{steel}}$ of steel and the mass $\mathcal{M}_{\text{magnet}}$ of magnet the machine is composed of:

$$\left\{\begin{array}{rcl}M_{\text{steel}} &=& \left(\pi\left(R_2^2 - R_1^2\right) + \pi\left(R_7^2 - R_4^2\right) - 2S_{\text{notch}}N_pN_n\right) L\rho_{\text{steel}}\cdot 10^{-9}\,,\\ M_{\text{copper}} &=& 2\tau_{\text{copper}}S_{\text{notch}}N_pN_n\left(L + \frac{\pi}{2N_p}(R_5+R_6)\right)\rho_{\text{copper}}\cdot 10^{-9}\,,\\ M_{\text{magnet}} &=& \pi\left(R_3^2 - R_2^2\right)\alpha_{\text{mag}}L\rho_{\text{magnet}}\cdot 10^{-9}\,,\end{array}\right. \tag{4.34}$$

where ρ_{magnet}, ρ_{copper} and ρ_{steel} are respectively the magnet, copper and steel densities (see Table 4.14). The mass and cost of the electric machine are then given by

$$\left\{\begin{array}{rcl}\mathcal{M}_{\text{em}} &=& \mathcal{M}_{\text{copper}} + \mathcal{M}_{\text{steel}} + \mathcal{M}_{\text{magnet}}\,,\\ \mathcal{C}_{\text{em}} &=& \mathcal{M}_{\text{copper}}\mathcal{C}_{\text{copper}} + \mathcal{M}_{\text{steel}}\mathcal{C}_{\text{steel}} + \mathcal{M}_{\text{magnet}}\mathcal{C}_{\text{magnet}}\,,\end{array}\right. \tag{4.35}$$

where $\mathcal{C}_{\text{copper}}$, $\mathcal{C}_{\text{steel}}$ and $\mathcal{C}_{\text{magnet}}$ are respectively the magnet, copper and steel costs per kilogram (see Table 4.14).

4.3.4 Optimization of the system

Formulation of an optimization problem

The design variables are the seven electric machine parameters, the inverter maximum effective current $I^{\text{eff}}_{\text{max}}$, the reducer reduction ratio r and the twelve pairs of intensities $(i_{d,(\omega,\tau)}, i_{q,(\omega,\tau)})$ associated to the different pairs (ω,τ) in Λ_{urban} and Λ_{jam}. See Table 4.8 for a description of the design variables ranges of variation.

The problem features eight equality constraints coming from the customer use cycles (four constraints by cycle, see Section 4.3.2), twenty four inequality constraints on the voltages and intensities given by (4.28), and five inequality constraints coming from the additional requirements (four on the torque plus one on the autonomy of the vehicle given by (4.23), see Section 4.3.2).

To handle the equality constraints, we formulate them as eight pairs of inequality constraints

Description	Nb	Not.	Range
Length of the machine (mm)	$\times 1$	L	$[50, 250]$
Rotor inner radius (mm)	$\times 1$	r_{rot}	$[30, 125]$
Rotor thickness (mm)	$\times 1$	ϵ_{rot}	$[10, 30]$
Air gap (mm)	$\times 1$	ϵ_{gap}	$[0.3, 2]$
Magnet thickness (mm)	$\times 1$	ϵ_{mag}	$[2, 10]$
Magnet angular spread	$\times 1$	α_{mag}	$[0.75, 0.9]$
Opening ratio between notch and dent gap	$\times 1$	α_{notch}	$[0.3, 0.9]$
Maximum effective current (A)	$\times 1$	$I_{\text{max}}^{\text{eff}}$	$[40, 160]$
Reduction ratio	$\times 1$	r	$[5, 20]$
Current intensity on axis d (A)	$\times 12$	i_d	$[-160\sqrt{3}, 0]$
Current intensity on axis q (A)	$\times 12$	i_q	$[0, 160\sqrt{3}]$

Table 4.8: Design variables description.

of the form:

$$\begin{cases} \Gamma\left(\omega, i_{d,(\omega,\tau)}, i_{q,(\omega,\tau)}\right) & \leq \quad \tau + \varepsilon\,, \\ \Gamma\left(\omega, i_{d,(\omega,\tau)}, i_{q,(\omega,\tau)}\right) & \geq \quad \tau - \varepsilon\,, \end{cases} \tag{4.36}$$

where Γ is the torque as given by (4.27), ε is a relaxation parameter and $(\omega, \tau) \in \Lambda_{\text{urban}} \cup \Lambda_{\text{jam}}$. In our experiments, we take $\varepsilon = 1$ Nm.

In the settings that we consider, the pairs $(i_{d,(\omega,\tau)}, i_{q,(\omega,\tau)})$, $(\omega, \tau) \in \Lambda_{\text{urban}} \cup \Lambda_{\text{jam}}$ and the inverter maximum effective current $I_{\text{max}}^{\text{eff}}$ are design variables. Therefore, the constraints (4.28) on the intensities can be considered as cheap to evaluate. The problem can thus be formulated using $q = 33$ expensive-to-evaluate constraints and twelve cheap-to-evaluate constraints.

We consider an optimization problem where the objectives are to minimize the cost $\mathcal{C}_{\text{pt}}$ of the powertrain, which is given by (4.30), and the average electromagnetic losses L_ϕ^{avg}. To get a meaningful expression of the average electromagnetic losses L_ϕ^{avg}, the losses associated to the two driving cycles are agregated as follows:

$$L_\phi^{\text{avg}} = 0.6\, L_{\phi,\text{urban}}^{\text{avg}} + 0.4\, L_{\phi,\text{jam}}^{\text{avg}}\,, \tag{4.37}$$

where $L_{\phi,\text{urban}}^{\text{avg}}$ and $L_{\phi,\text{jam}}^{\text{avg}}$ are the average losses associated respectively to the urban cycle and to the traffic jam cycle. To compute these, we attribute to each pair $(\omega, \tau) \in \Lambda_{\text{urban}} \cup \Lambda_{\text{jam}}$ the

following coefficients:

$$\begin{cases} E_{\left(\frac{\omega_{\text{base}}}{2},\frac{\tau_{\max}}{4}\right)} &= \displaystyle\sum_{i\in I}\frac{\mathcal{P}_i}{t_{i+1}-t_i}\mathbb{1}_{\omega_i\leq\omega_{\text{base}},\tau_i\leq 0.5\tau_{\max}}\,, \\ E_{\left(\frac{\omega_{\text{base}}+\omega_{\max}}{2},\frac{15\mathcal{P}_{\max}}{\pi(\omega_{\text{base}}+\omega_{\max})}\right)} &= \displaystyle\sum_{i\in I}\frac{\mathcal{P}_i}{t_{i+1}-t_i}\mathbb{1}_{\omega_i>\omega_{\text{base}},\mathcal{P}_i\leq 0.5\mathcal{P}_{\max}}\,, \\ E_{\left(\frac{\omega_{\text{base}}}{2},\frac{3\tau_{\max}}{4}\right)} &= \displaystyle\sum_{i\in I}\frac{\mathcal{P}_i}{t_{i+1}-t_i}\mathbb{1}_{\omega_i\leq\omega_{\text{base}},\tau_i>0.5\tau_{\max}}\,, \\ E_{\left(\frac{\omega_{\text{base}}+\omega_{\max}}{2},\frac{45\mathcal{P}_{\max}}{\pi(\omega_{\text{base}}+\omega_{\max})}\right)} &= \displaystyle\sum_{i\in I}\frac{\mathcal{P}_i}{t_{i+1}-t_i}\mathbb{1}_{\omega_i>\omega_{\text{base}},\mathcal{P}_i>0.5\mathcal{P}_{\max}}\,. \end{cases} \tag{4.38}$$

The average electromagnetic losses associated to both driving cycles are then computed as

$$\begin{cases} L_{\phi,\text{urban}}^{\text{avg}} &= \displaystyle\sum_{(\omega,\tau)\in\Lambda_{\text{urban}}}\overline{E}_{(\omega,\tau)}\left[R\left(i_{d,(\omega,\tau)}^2+i_{q,(\omega,\tau)}^2\right)+L_\phi(\omega,i_{d,(\omega,\tau)},i_{q,(\omega,\tau)})\right]\,, \\ L_{\phi,\text{jam}}^{\text{avg}} &= \displaystyle\sum_{(\omega,\tau)\in\Lambda_{\text{jam}}}\overline{E}_{(\omega,\tau)}\left[R\left(i_{d,(\omega,\tau)}^2+i_{q,(\omega,\tau)}^2\right)+L_\phi(\omega,i_{d,(\omega,\tau)},i_{q,(\omega,\tau)})\right]\,, \end{cases} \tag{4.39}$$

where the overline symbol indicates that the coefficients $E_{(\omega,\tau)}$, $(\omega,\tau)\in\Lambda_{\text{urban}}$ or Λ_{jam}, are normalized to sum to one.

To summarize, the optimization problem that is considered in this study has $d=33$ design variables, $p=2$ objectives and $q=33$ constraints. The design space is defined using bound constraints on the design variables (see Table 4.8) and twelve constraints on the intensities of the current passing through the coils. It does not feature hidden constraints.

Optimization results

In Figures 4.9 and 4.10 we show the results obtained by BMOO on this problem. The algorithm is initialized using $N_{init}=100$ experiments and run with a limiting number of functions evaluations $N_{max}=500$. See Section 4.1 for more details about the settings of the algorithm.

In the figures, we show the progress of the algorithm every fifty iterations. The two phases of the optimization process clearly appear in the figures. First, the algorithm makes progress on the resolution of the constraints. This phase corresponds to subfigures (a) to (d) in Figure 4.9. It can be seen that the algorithm progressively focuses the search in the regions of low constraint violation until finding a feasible solutions, which happens at $N=202$. Among the 298 subsequent iterations, only 42 resulted in non-feasible solutions, which indicates that the algorithm correctly identified the feasible region. Then, the algorithm makes progress on the minimization of the objectives. It can be seen in the figures that it explores simultaneously the different regions of the Pareto front.

Observe in particular the difference of scale between subfigures (a) of Figure 4.9 and (c) of Figure 4.10. This is typically a situation where the maximum of the objectives over the entire $\mathbb{X}$ domain are large compared to their maximum value on the Pareto front. In this study, we use the adaptive procedure described in Section 3.6.2 to keep y^{upp} at a "reasonable" distance of the set of non-dominated observations during the optimization process, as recommended in

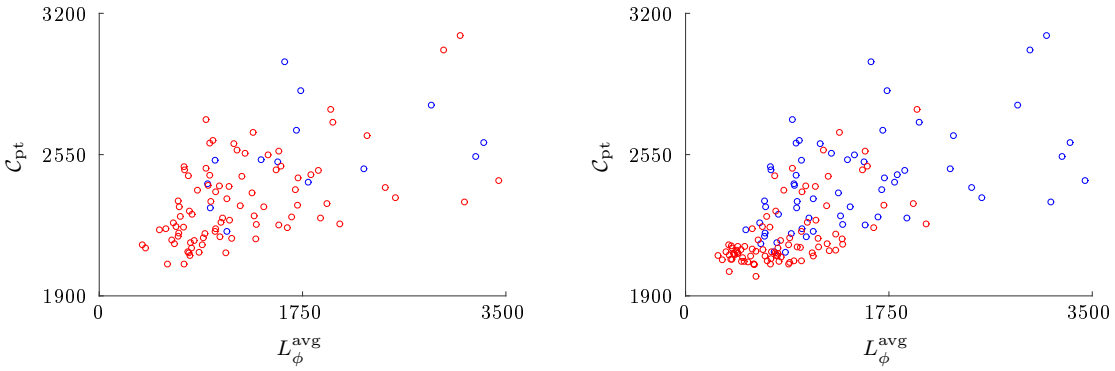
Section 3.4.

4.3.5 Conclusions

In this section, BMOO is applied to the design of an electric vehicle powertrain. The problem takes the form of a bi-objective optimization problem defined on a non-hypercubic design space and subject to both equality and inequality constraints. To handle equality constraints, a relaxation method is used. This makes it possible to transform equality constraints as pairs of inequality constraints.

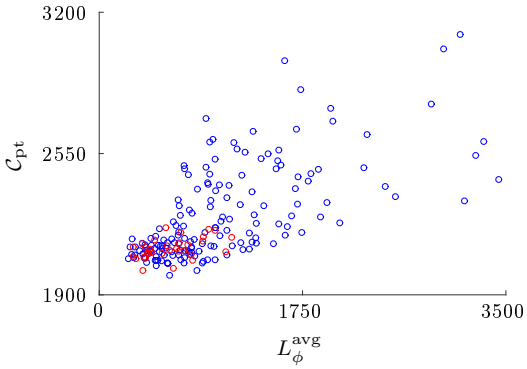
For better performances of the system, it would be interesting to consider also the sizing of the battery. However, it is defined using discrete variables which BMOO cannot handle in its present form. Extending BMOO to this new setup could be an interesting direction for future research work.

Acknowledgements: The authors would like to thank Amin El-Bakkali (Renault), who developped the simulation model and formulated the specifications, Karim Cammoun (IRT SystemX), who made it available to the institute, and Yves Tourbier (Renault).

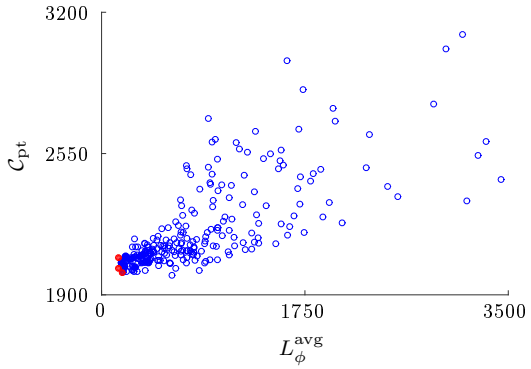


(a) $N = 100$

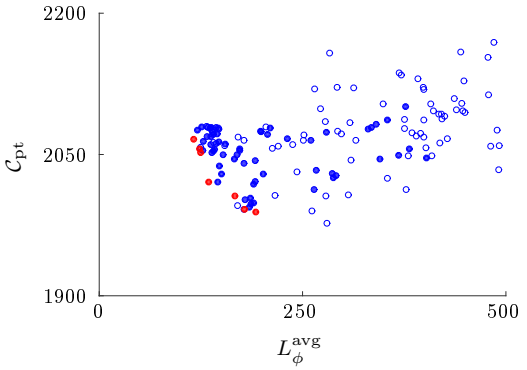
(b) $N = 150$



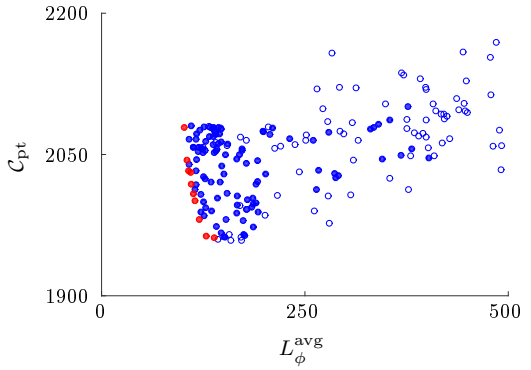
(c) $N = 200$



(d) $N = 250$

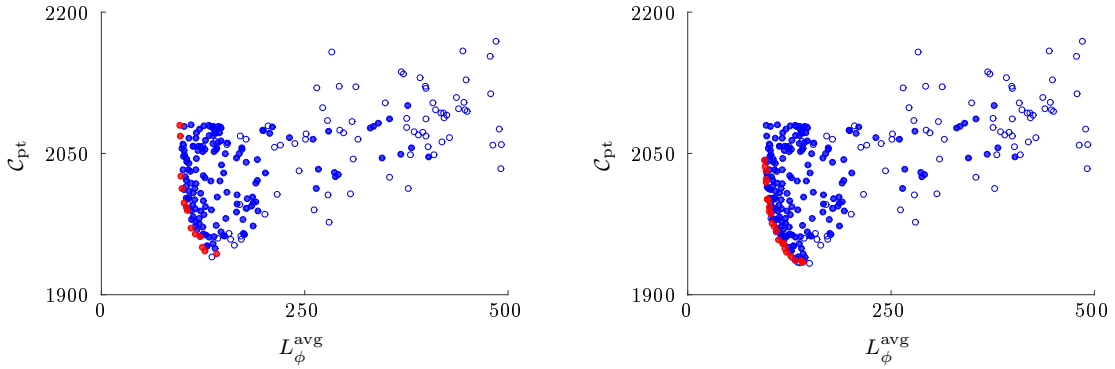


(e) $N = 300$

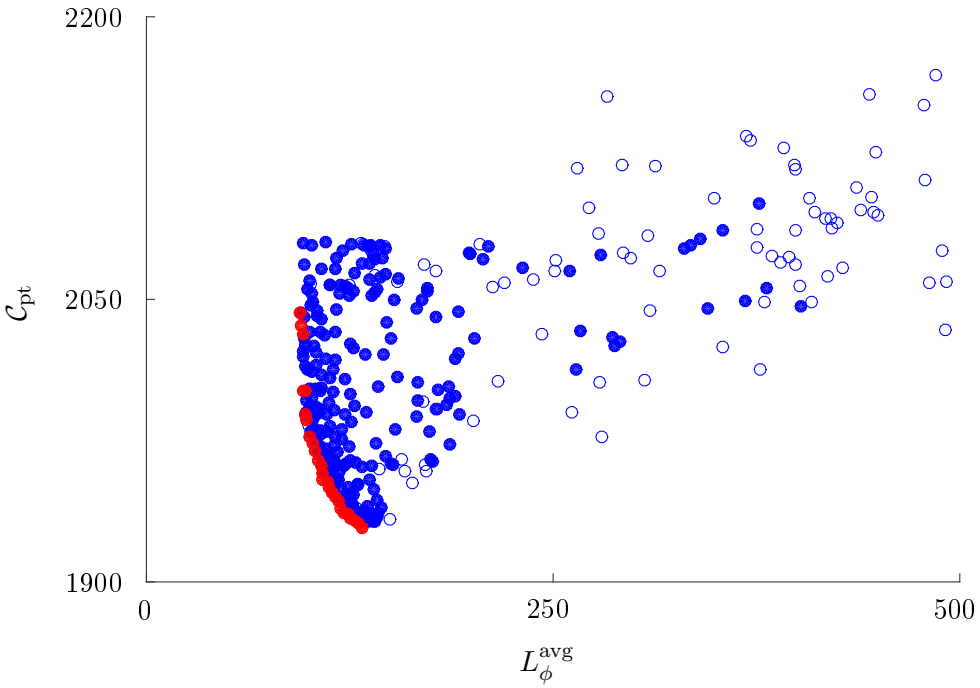


(f) $N = 350$

Figure 4.9: Objectives values of the observations made by BMOO at different stages of the optimization process. Circles indicate non-feasible observations and disks indicate feasible ones. Dominated observations are shown in blue and non-dominated ones in red.



(a) $N = 400$ (b) $N = 450$



(c) $N = 500$

Figure 4.10: Objectives values of the observations made by BMOO at different stages of the optimization process. Circles indicate non-feasible observations and disks indicate feasible ones. Dominated observations are shown in blue and non-dominated ones in red.

4.3.6 Additional material

Description	Not.	Value
Vacuum voltage of a cell (V)	E_0	3.75
Capacity of a cell (Ah)	κ_0	43
Mass of a cell (kg)	M_0	1.05
Internal resistance of a cell (Ω)	R_0	0.001
Mass coefficient	$\lambda_{\text{bat,mass}}$	1.65
Cost coefficient 1	$\lambda_{\text{bat,cost,1}}$	160
Cost coefficient 2	$\lambda_{\text{bat,cost,2}}$	6.95

Table 4.9: Battery parameters.

Description	Not.	Value
Voltage factor per phase	k	1.3505
Efficiency	η_{inv}	0.9
Mass coefficient 1	$\lambda_{\text{inv,mass,1}}$	0.06
Mass coefficient 2	$\lambda_{\text{inv,mass,2}}$	1.1
Cost coefficient 1	$\lambda_{\text{inv,cost,1}}$	1.4
Cost coefficient 2	$\lambda_{\text{inv,cost,2}}$	150

Table 4.10: Inverter parameters.

Description	Not.	Value
Drag coefficient (m^2)	C_x	0.64
Friction coefficient ($N.kg^{-1}$)	K	0.07
Wheel radius (m)	R_w	0.2811

Table 4.11: Vehicle parameters.

Description	Not.	Value
Efficiency	η_{red}	0.95
Mass coefficient 1	$\lambda_{\text{red,mass,1}}$	0.15
Mass coefficient 2	$\lambda_{\text{red,mass,2}}$	10
Cost coefficient 1	$\lambda_{\text{red,cost,1}}$	1.75
Cost coefficient 2	$\lambda_{\text{red,cost,2}}$	130

Table 4.12: Reducer parameters.

Description	Not.	Value
Number of pole pairs	N_p	2
Number of notches per pole	N_n	4
Number of wires per notch	N_w	12
Maximum current density ($A.mm^{-2}$)	J_{max}	10
Notch copper ratio	r_{copper}	0.5
Mouth depth (mm)	ϵ_{mouth}	1.2
Opening ratio between mouth and notch	r_{mouth}	0.25
Maximum rotation speed (rpm)	Ω_{max}	7000
Steel plate thickness (mm)	e_{plate}	0.35
Thickness ratio between plate and insulator	r_{plate}	0.97
Section ratio between copper and notch	τ_{copper}	0.5
Loss coefficient	α_{hys}	2
Loss coefficient	k_{hys}	0.003
Loss coefficient	k_{exc}	0.004

Table 4.13: Machine parameters.

Description	Not.	Value
Air density ($kg.m^{-3}$)	ρ	1.2
Gravity acceleration ($m.s^{-2}$)	g	9.81
Electrical vacuum permeability	μ_0	$1.257\,10^{-6}$
Copper cost ($euros.kg^{-1}$)	C_{copper}	6.7
Steel electrical conductivity ($MS.m^{-1}$)	σ_{steel}	0.5
Steel density ($kg.m^{-3}$)	ρ_{steel}	0.5
Steel control parameter	J_{steel}	2
Steel relative permeability	$\mu_{\text{r,steel}}$	7000
Steel H definition	$H_{\text{max,steel}}$	8000
Steel cost ($euros.kg^{-1}$)	C_{steel}	1.2
Magnet electrical conductivity ($MS.m^{-1}$)	σ_{magnet}	0.667
Magnet density ($kg.m^{-3}$)	ρ_{magnet}	3500
Magnet relative permeability	$\mu_{\text{r,magnet}}$	1.05
Magnet value of B for H=0	$B_{\text{r,magnet}}$	1.27
Magnet cost ($euros.kg^{-1}$)	C_{magnet}	125
Coil electrical conductivity ($MS.m^{-1}$)	σ_{coil}	58
Coil density ($kg.m^{-3}$)	ρ_{coil}	8920

Table 4.14: Miscellaneous parameters.

4.4 Tuning of a Line of Sight controller

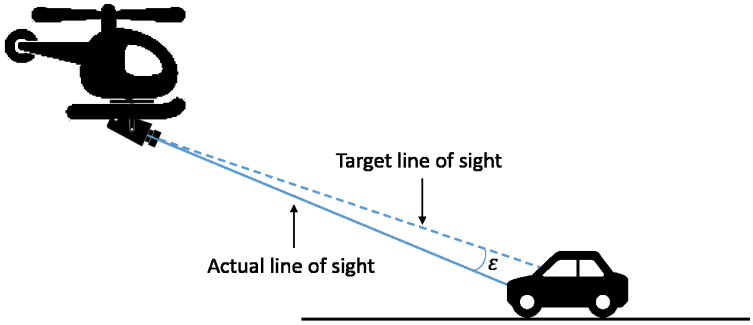


Figure 4.11: Example of an application where an inertially stabilized platform is required. The optronic device is a tracking camera mounted on an helicopter. The line of sight of the camera must be isolated from the perturbations induced by the movements and vibrations of the helicopter and aligned with the target line of sight. On the illustration, ε is the angular error between the target line of sight and the actual line of sight. The objective is to control ε to obtain a good image quality. Illustration reproduced from Frasnedo (2016).

4.4.1 Introduction

This study is part of a joint work between CentraleSupélec and Safran Electronics & Defense on the development and tuning of the controller of an inertially stabilized platform (ISP). It is the fruit of a collaboration with Sophie Frasnedo and is based on her thesis work (Frasnedo et al., 2015b,a; Frasnedo, 2016).

The purpose of an ISP is to isolate an optronic device from the movements of its support, in order to obtain a good image acquisition quality. An example of a situation where such a system is required is illustrated in Figure 4.11. In this section, we limit ourselves to presenting results obtained using the BMOO algorithm to optimize the parameters of the controller of the system. For more details on the system itself, the interested reader is referred to the the PhD thesis of Frasnedo (2016).

The section is organized as follows. First we present in Section 4.4.2 the architecture of a typical ISP and introduce its main components. The model that is used to emulate the system is also briefly discussed. Then, we discuss in Section 4.4.3 the choice of image quality criteria to be used as objectives of the optimization problem. The full formulation of the optimization problem and the optimization results are presented in Section 4.4.4 and conclusions are drawn in Section 4.4.5.

4.4.2 Stabilization architecture model

The schematic architecture of an inertially stabilized platform is represented in Figure 4.12. Two pivot connections in pitch and yaw are used to isolate the platform from the movement of the support. The optronic device and a gyrometer are fixed on the yaw gimbal. The gyrometer

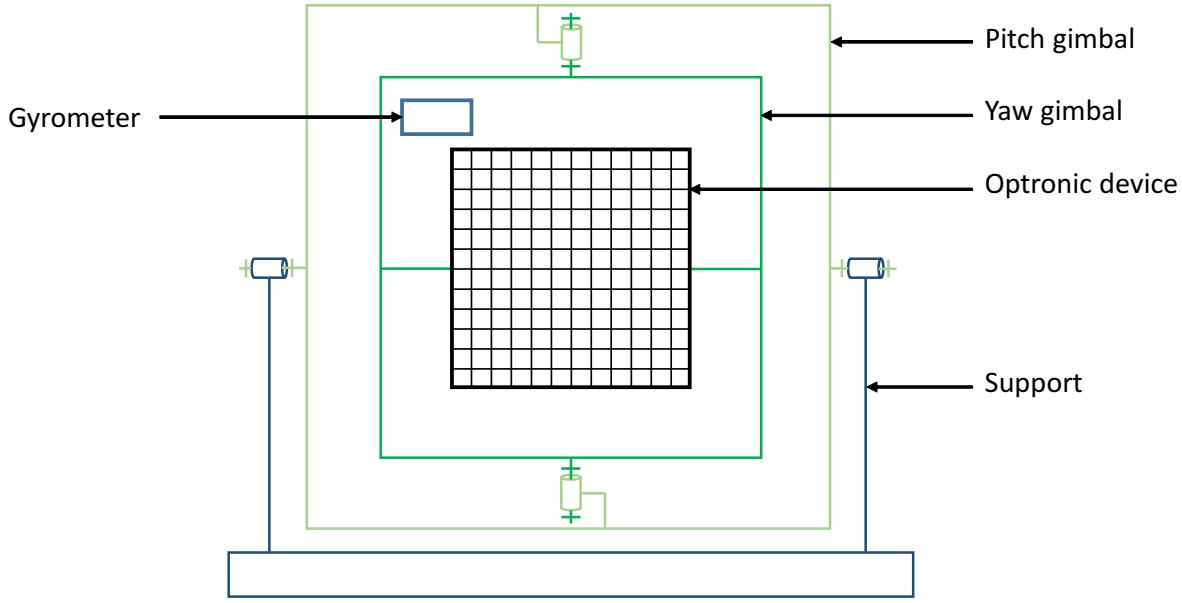


Figure 4.12: Architecture of the inertially stabilized platform. Illustration reproduced from Frasnedo (2016).

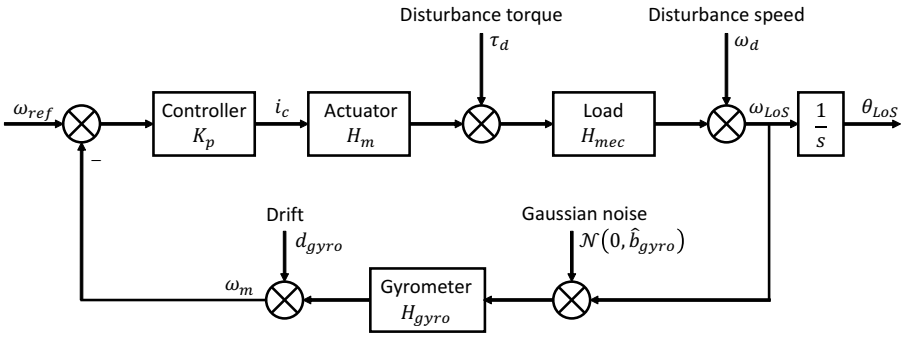


Figure 4.13: Block diagram of the inertially stabilized platform. Illustration reproduced from Frasnedo (2016).

measures the rotational speed of the gimbal and provides feedback to a controller which is not represented in the illustration. Actuators placed at the pivot connections and monitored by the controller are then used to compensate the movements of the support and stabilize the platform. For the sake of simplicity, in the following, the two rotations are assumed independent and only the pitch is considered (the yaw gimbal is fixed).

The system is emulated using a Simulink model based on the block diagram of Figure 4.13. The actuator is a DC motor with a torque constant $K_\tau = 0.9N.m/A$ and a transfer function

$$H_m(s) = K_\tau \cdot \frac{A(s)}{A(s) + B(s)}, \tag{4.40}$$

where s is the Laplace variable and

$$\begin{cases} A(s) &= 0.012 \cdot (1 + 0.088s) \cdot (1 + 0.001s) \cdot (1 + 0.126s), \\ B(s) &= (1 + 0.095s) \cdot (1 + 0.031s) \cdot 0.088s \cdot 0.001s. \end{cases} \tag{4.41}$$

The load that is put in motion by the actuator is the pitch gimbal, which supports the yaw gimbal, the optronic device and the gyrometer. If the mechanical resonant modes are neglected, it can be modelled as a pure inertia.

$$H_{mec}(s) = \frac{1}{0.4s}. \tag{4.42}$$

The movement of the support is represented as a torque disturbance τ_d (low frequency perturbation) and its vibrations are represented as a rotation speed disturbance ω_d (high frequency perturbation).

A gyrometer placed in the same referential as the optronic device is used to evaluate its rotation speed with respect to an inertial referential and provide feedback to the controller. The transfer function H_{gyro} of the gyrometer is taken as the product of two first order filters of respective cut-off frequencies f_{gyro} and f_{filter}.

$$H_{gyro}(s) = \frac{1}{1 + \frac{s}{2\pi f_{gyro}}} \cdot \frac{1}{1 + \frac{s}{2\pi f_{filter}}} \tag{4.43}$$

It is assumed that the measures of the gyrometer are subject to a white noise which is Gaussian with a variance $\widehat{b}_{gyro}$ which has been estimated empirically beforehand (see Section 2.4 of Frasnedo (2016)). The drift d_{gyro} of the gyrometer is given by the constructor and is taken into account in this study.

Finally, the controller controls the torque delivered by the actuator. It is composed of four terms. First, a double integrator is used to obtain a zero steady state position error for a torque disturbance step (final value theorem). To compensate the negative effect of the double integrator on the system stability, a second order numerator parametrized by ω_i and ξ_i is used. A phase advance function parametrized by a and T is also introduced to further reinforce the stability margins of the system. Finally, a gain K is used to adjust the bandwidth and a second order denominator parametrized by ω_{ro} and ξ_{ro} permits to attenuate high frequencies. This last term is introduced to lower the effect of the noise and account for potential dynamics otherwise neglected.

$$K_p(s) = K \cdot \frac{1 + aTs}{1 + Ts} \cdot \frac{1 + \frac{2\xi_i s}{\omega_i} + \frac{s^2}{\omega_i^2}}{s^2} \cdot \frac{1}{1 + \frac{2\xi_{ro} s}{\omega_{ro}} + \frac{s^2}{\omega_{ro}^2}} \tag{4.44}$$

In the following, we denote $\eta = (K, \omega_i, \xi_i, a, T, \omega_{ro}, \xi_{ro})$ the vector of the parameters of the controller and the goal of the study is to tune η.

4.4.3 Image quality criteria

The influence of an error in position of the line of sight over time on the image quality is twofold. First, the optronic device has a certain acquisition time and if the line of sight moves during this time, it introduces a blur in the image because the pixels receive overlapping information. To quantify this effect, in this work, the motion transfer function (MTF) is used. This function provides a measure of the degradation of the contrast in the image as a function of the spatial

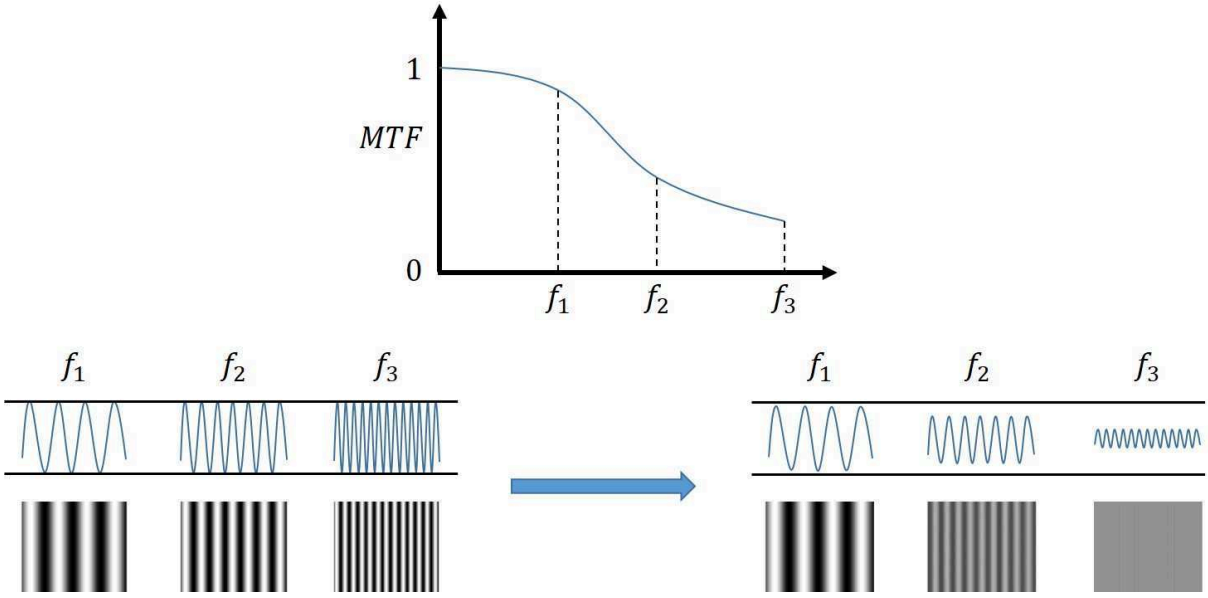


Figure 4.14: Attenuation of constrast caused by the movement of the line of sight during its acquisition time. The motion transfer function measures the attenuation in intensity of the contrast as a function of the spatial frequency. In general, high frequencies are more attenuated than low frequencies, i.e. small details of the image are lost in the blur of the image. Illustration reproduced from Frasnedo (2016).

frequency of the image details and of the movements of the line of sight. A detailed description of the origin of this function and of its computation is out of the scope of this report and the reader is referred to Section 3.3 of Frasnedo (2016) for more details on the MTF. Here, we limit ourselves to illustrating how this function can be used for evaluating the attenuation of the contrast in Figure 4.14.

The second criterion that we consider is related to the integration time τ_{eye} of the human eye. When the magnitude of the movement of the line of sight is too important during this time, this introduces a frame-to-frame shift that is perceptible to the human eye and constitutes a nuisance. To measure this effect, we consider the amplitude d_i of the movement of the line of sight over intervals of time $[t_i - \tau, t_i]$, with $\tau < \tau_{eye}$, $i \in [\![1, n]\!]$ and $t_i \in [\tau, T]$. Then, to summarize the information measured at all t_i and to be conservative, we consider the maximum movement magnitude $d_{max} = \max_{1 \leq i \leq n} d_i$ as unique criterion (see Figure 4.15).

These two criteria may seem redundant since they are both related to the movement of the line of sight during a time interval. However, in general, the eye integration time is higher than the optronic device acquisition time ($0.1s \leq \tau_{eye} \leq 0.2s$ for most people whereas the acquisition time of the optronic device is of the order of the millisecond) and the two criteria are defined over different different time scales. Besides, the MTF depends on the characteristics of the optronic device and on the spatial frequency whereas the d_{max} criterion is a general image quality metric that is solely related to the operator.

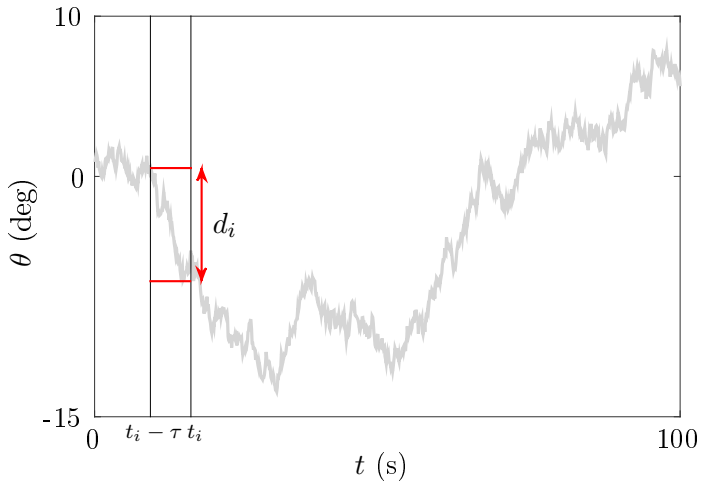


Figure 4.15: Angular movement of the line of sight over time. The magnitude of the movement in the time interval $[t_i - \tau, t_i]$ is measured by d_i. When d_i is large, it causes a frame-to-frame shift during the eye integration time. Illustration reproduced from Frasnedo (2016).

4.4.4 Tuning of the controller

Besides maximizing the motion transfer function and minimizing the d_{max} criterion to obtain a good image quality, the tuning of the controller should also take into account the power consumption of the system. Indeed, the ISP is an embedded system powered by a battery and it has a limited autonomy. Therefore, in this work, the minimization of the mean power consumption P_{moy} of the system is also considered in the tuning of the controller.

Furthermore, the system should respect specifications. The current intensity I applied to the motor coils should not exceed a threshold value to avoid damages, the eigenvalues of the closed loop system must be lower than a fixed threshold, and the damping and the modulus margin of the system must be greater than fixed thresholds. Additionally, constraints are placed on the objectives d_{max}, P_{moy} and on the MTF to restrict the range of the Pareto front to acceptable designs (see Remark 33). To constrain the MTF, we consider its values at eight equally spaced frequencies between 0 and 1 and impose that they be above the values of a template function at the same frequencies (see Figure 4.17).

For this application, most of the computing time comes from the evaluation of the MTF and of d_{max}, which requires simulating the whole trajectory of the line of sight (this takes a few minutes). In comparison, the constraints placed on the eigenvalues of the closed loop system, on the damping, and on the modulus margin of the system are cheap to evaluate (in the order of the millisecond). Therefore, in this work, they are considered as restrictions of the design space (see Section 3.5.2). As an additional benefit, this prevents the code to be subject to simulation failures, which may happen otherwise when the system is unstable.

We consider two formulations of the optimization problem. First, we consider a bi-objective formulation where only d_{max} and P_{moy} are minimized. The MTF is constrained to be above the template function of Figure 4.17 but we do not try to maximize its values. In the second formulation, we consider the simultaneous optimization of d_{max}, P_{moy} and of the values of the

MTF at three equally spaced frequencies (see Figure 4.16). The problem is thus a five-objectives optimization problem. For both formulations, the optimization problem has $d = 7$ design variables (the parameters of the controller) and $q = 13$ expensive to evaluate constraints (d_{max}, P_{moy}, I and ten constraints on the MTF). Additional results obtained with the batch version of the BMOO algorithm presented in Section 3.5.4 for a formulation with $p = 7$ objectives are presented in Section 4.4.6.

The results obtained with BMOO for the two formulations are reported in Figures 4.16 and 4.17. For both formulations, the algorithm is initialized with a pseudo-maximin Latin-hypercube design of $N_{init} = 30$ experiments and run with a limiting number of functions evaluations $N_{max} = 500$. See Section 4.1 for more details about the settings of the algorithm.

In Figure 4.16, we show the projection of the non-dominated solutions found by BMOO in the plan of d_{max} and P_{moy}. The value of the MTF at the Nyquist frequency (corresponding to $f_N = 0.5$ in Figure 4.17, see Section 3.4 of Frasnedo (2016)) is shown in color. The solutions with the lowest d_{max} value and the solution with the highest MTF value at the Nyquist frequency are numbered from 1 to 3 in the figure and their MTF are represented in Figure 4.17. The MTF values of the other non-dominated solutions found by BMOO are shown as a shaded area in Figure 4.17.

In the light of these experiments, it appears that the objectives are not strongly antagonists. The results obtained for the first formulation with $p = 2$ show that it is possible to minimize d_{max} without impacting too much on the mean power consumption (the specification only imposes $P_{moy} \leq 0$). The results obtained for the second formulation with $p = 5$ show that the maximization of the MTF is not strongly concurrent with the minimization of d_{max} either. Indeed, the loss in terms of P_{moy} and d_{max} is negligible when the solution with the highest MTF value at the Nyquist frequency is chosen (point 3 in Figure 4.16). Then, looking at the MTF curves represented in Figure 4.17, it can be seen that the maximization of the MTF values at different frequencies are not concurrent objectives either. Indeed, the solution that maximizes the MTF value at the Nyquist frequency (curve 3 in red in Figure 4.17) also maximizes the MTF at the other frequencies (it is above the shaded region).

For this problem, a three-objectives formulation would thus have been sufficient to highlight the trade-off between d_{max}, P_{moy} and the MTF. However, this information was not known beforehand; it is a result of the many-objective formulation.

Remark 32 *Numerical values such as the constraints thresholds and the variables range of variation are either not given or normalized when they are. This is done to preserve the confidentiality of the results.*

Remark 33 *Note that for this application, constraints are placed on the objectives and the two are not independent. Besides, the eight constraints placed on the MTF are not independent either. In practice though we use the algorithm as if it was the case because taking this information into account would require a significant work. A rigorous treatment of such situations is left for future work.*

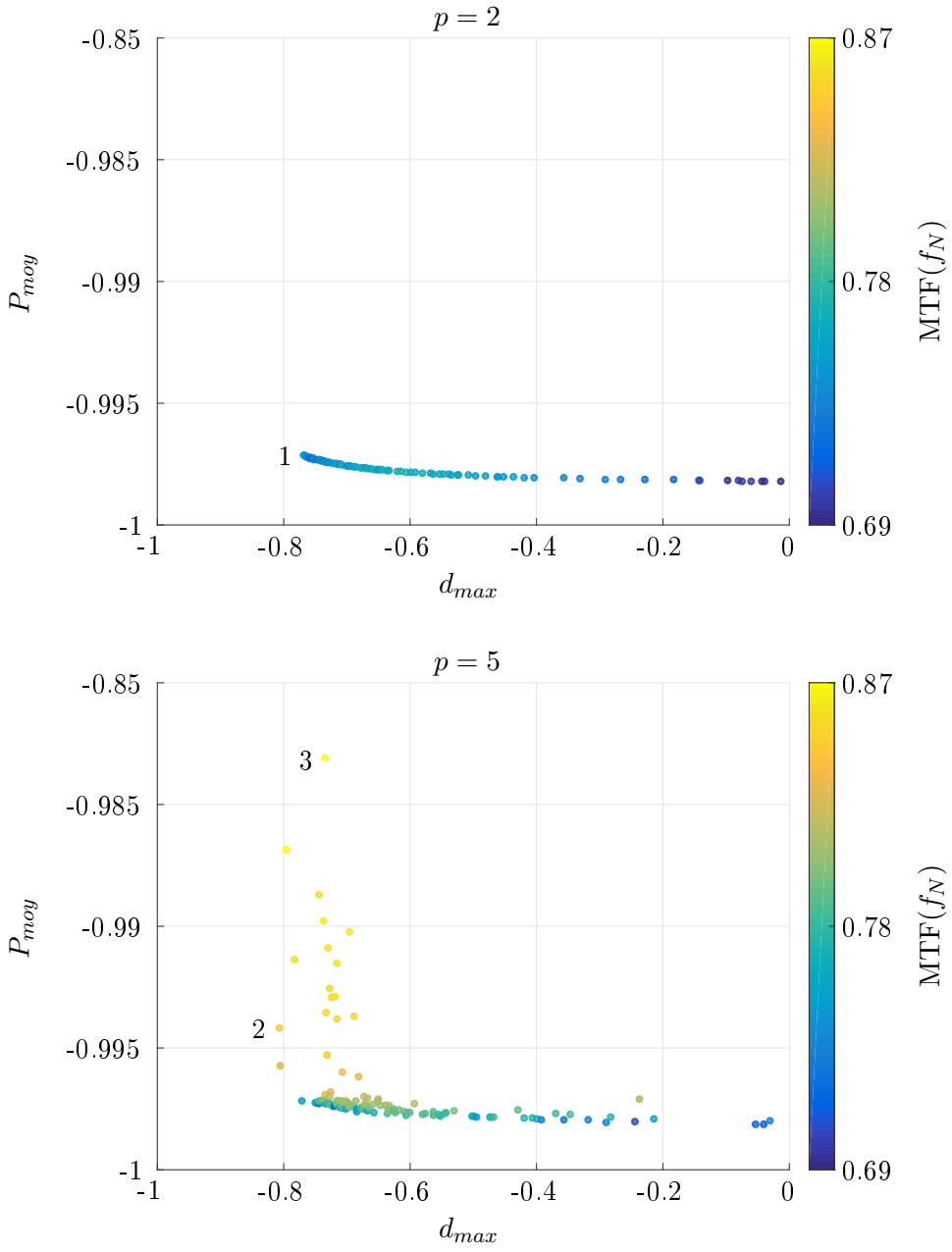


Figure 4.16: Fronts of non-dominated solutions found by the BMOO algorithm when the number of objectives is $p = 2$ (top) and $p = 5$ (bottom). The color bars on the right side of the figures indicate the value of the MTF at the Nyquist frequency (the higher the better). On each figure, the numbered dots correspond to the solutions with the lowest d_{max} values (1 & 2) and with the highest MTF value at the Nyquist frequency (3). Their MTF are shown in Figure 4.17.

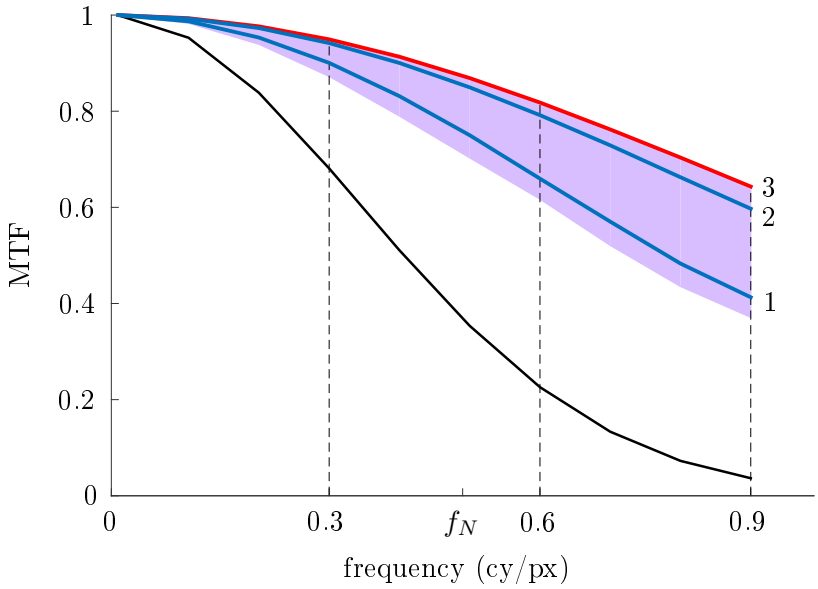


Figure 4.17: MTF associated with the three numbered solutions in Figure 4.16. The shaded region corresponds to the MTF values of the other non-dominated solutions found by BMOO on the problem with $p = 5$, i.e. the solutions represented in Figure 4.16. The dashed vertical lines correspond to the frequencies at which the MTF is optimized. The black curve is the template function and the MTF is constrained to remain over it.

4.4.5 Conclusions

In this section, the BMOO algorithm is applied to the tuning of a line of sight controller based on image quality criteria. The problem takes the form of a constrained many-objective optimization problem defined on a non-hypercubic design space. Two formulations of the optimization problem are considered: a bi-ojective formulation and a five-objective formulation. The results obtained for the two formulations reveal that a good image quality can be achieved with a small cost in terms of mean power consumption. In particular, the use of a many-objective formulation makes it possible to determine that the objectives are not strongly antagonists; information which was not known beforehand.

Also, additional experiments with the batch version of the algorithm presented in Section 3.5.4 are made on a seven-objective formulation of the problem. The results are consistent with those found sequentialy, which validates the proposed approach for selecting batches of experiments.

In the considered optimization problem formulations, constraints are placed on the objectives to limit the range of the Pareto front. This defeats the assumption made in this work that the functions of the problem are independent. It is our belief that such a situation often occurs in real-life applications and this shall motivate future work on the algorithm to address this particular issue more rigorously.

4.4.6 Additional material

In Figures 4.18 and 4.19 we present additional results obtained with BMOO in batch mode (see Section 3.5.4). This time, we consider an optimization problem with $p = 7$ objectives: minimizing

d_{max} and P_{moy}, and maximizing the values of the MTF at five equally spaced frequencies. For this experiment, BMOO is initialized with an initial design of $N_{init} = 50$ experiments and run with a limiting number of functions evaluations $N_{max} = 500$. At each iterations of the algorithm, batches of $N = 50$ points are chosen using the procedure of Section 3.5.4. Note that in this setting, the algorithm makes only nine iterations.

The results obtained in batch mode are close to those obtained sequentially, which validates the batch selection procedure of Section 3.5.4. A small loss can be observed on the MTF but the overall shape of the Pareto front is well captured. For this application, we do not actually simulate the fifty new samples in parallel. However, if it was the case, significant savings in terms of computing time could be achieved.

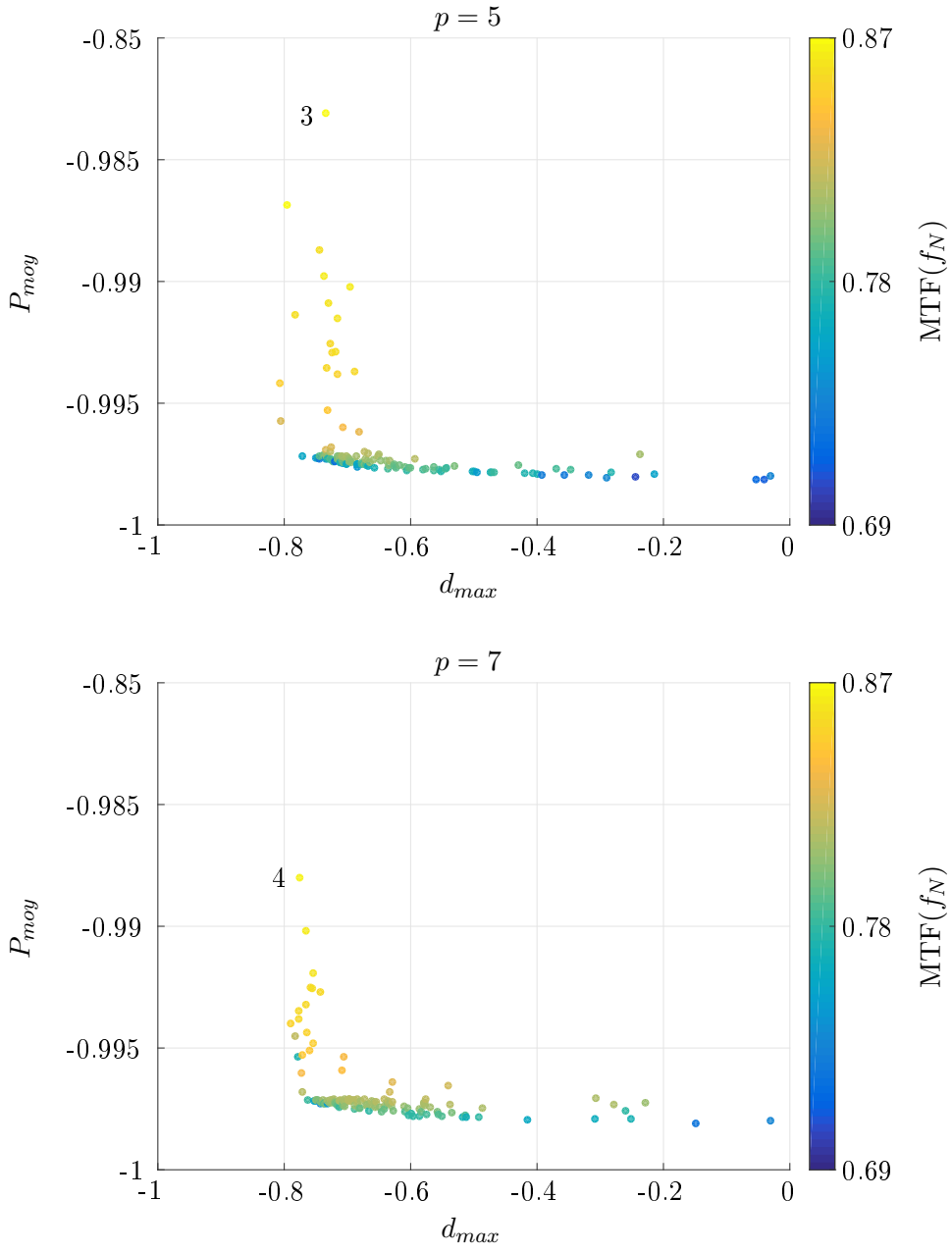


Figure 4.18: Fronts of non-dominated solutions found by the BMOO algorithm when the number of objectives is $p = 5$ (top) and $p = 7$ (bottom). The color bars on the right side of the figures indicate the value of the MTF at the Nyquist frequency (the higher the better). On each figure, the numbered dots correspond to the solutions with the highest MTF value at the Nyquist frequency. Their MTF are shown in Figure 4.19.

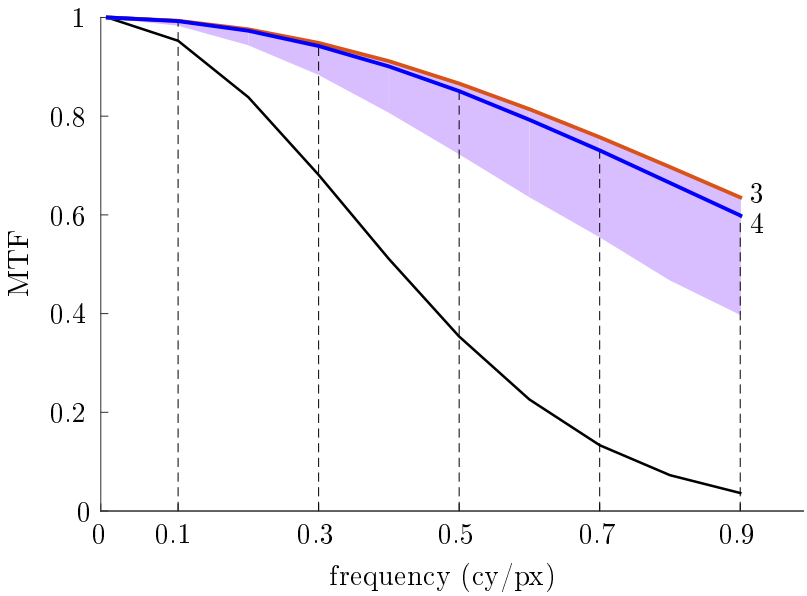


Figure 4.19: MTF associated with the solutions numbered 3 and 4 in Figure 4.18. The shaded region corresponds to the MTF values of the other non-dominated solutions found by BMOO on the problem with $p = 5$, i.e. the solutions represented in Figure 4.18. The dashed vertical lines correspond to the frequencies at which the MTF is optimized in the formulatin with $p = 7$. The black curve is the template function and the MTF is constrained to remain over it.

4.5 Design of a turbomachine fan blade

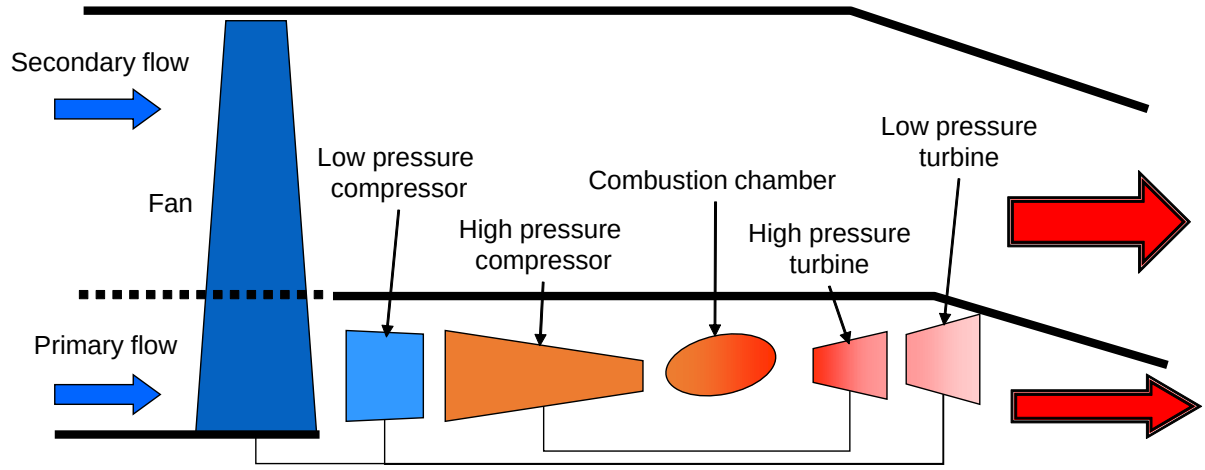


Figure 4.20: Global architecture of a turbomachine.

4.5.1 Introduction

This study deals with the design of a commercial aircraft turbomachine fan blade. It is the fruit of a collaboration between the Technological Research Institute SystemX, CentraleSupélec, Cénaéro and Safran Aircraft Engines.

A typical aircraft turbomachine is made of several components (see Figure 4.20): a fan, a low pressure compressor, an high pressure compressor, a combustion chamber, an high pressure turbine, a low pressure turbine, and several equipments and integration systems not represented in the figure. The fan is one of the most dimensioning components of a turbomachine. It is composed of several blades of high dimension (the diameter of the fan inner inlet can reach up to 1.80 m on recent engines), the profile of which varies from foot to head (see Figure 4.21).

The design of a turbomachine fan blade is made difficult by the fact that it requires an optimization of its aerodynamic performances under multiple mechanical and acoustic constraints, such as impact of foreign bodies, frequency margins, fatigue resistance and noise at take-off and landing. Its optimization is multi-disciplinary and possibly multi-scale, depending on the materials characteristics (composite material for example) and on the regions under study.

4.5.2 Simulation chain

In this study, we consider different regimens of the turbomachine, which correspond to different phases of the aircraft flight:

- landing,
- cruise,
- take-off,
- and redline (maximum operating regime).

Figure 4.21: Typical shape of a turbomachine fan blade.

The thrust is subject to constraints at take-off, the efficiency must be maximal on cruise and the noise must be bellow an acceptance threshold at landing. For these four regimens, we consider different operating conditions, that correspond to different values of pressure and mass flows (see Figure 4.22). Two operating points are considered at landing, four on cruise, two at take-off and only static and dynamic mechanical computations are made at the redline.

A schematic view of the architecture of the simulation chain used in this study is proposed in Figure 4.23. First, a cold blade geometry is built based on the chosen geometrical parameters (design variables and fixed parameters). This geometry corresponds to the geometry of the blade in the absence of rotation. After this first step, nonlinear static and dynamic computations are made at all considered regimens to determine the mechanical constraints, the deformations, and to infer the blade natural frequencies. Then, aerodynamic computations based on the deformed geometries are made for the 8 operating points considered in the study. These computations deliver the mass flow, efficiencies, noise, etc, at the different regimens. This step is distributed on 192 processors. Even then, it is responsible for most of the computing time of the simulation chain (approximately 4h30 in total for one simulation). The values of the objectives and constraints are obtained from these computations after a post-processing step.

4.5.3 Blade optimization

We consider an optimization problem with $p = 3$ objectives and $q = 9$ constraints. The objectives are the maximization of the efficiencies on cruise and at take-off and the maximization of the natural frequency of the blade at the redline. The constraints are five mechanical constraints, two aerodynamic constraints and two geometrical constraints. For the parametrization of the blade, $d = 26$ variables subject to bound constraints are used:

- incidence and trailing angles at five levels along the blade height (10 variables in total);
- skeleton parameters at four levels along the blade height (16 variables in total).

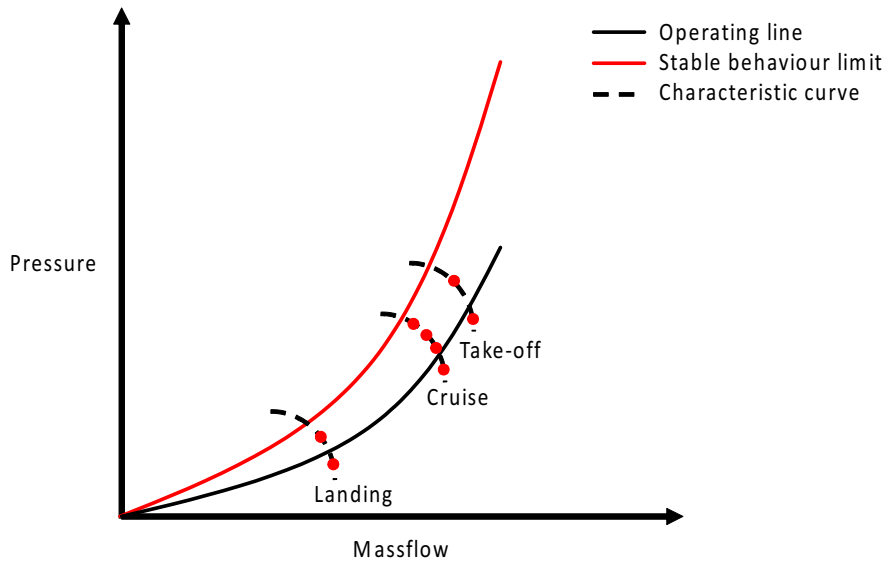


Figure 4.22: Operating conditions considered in this study.

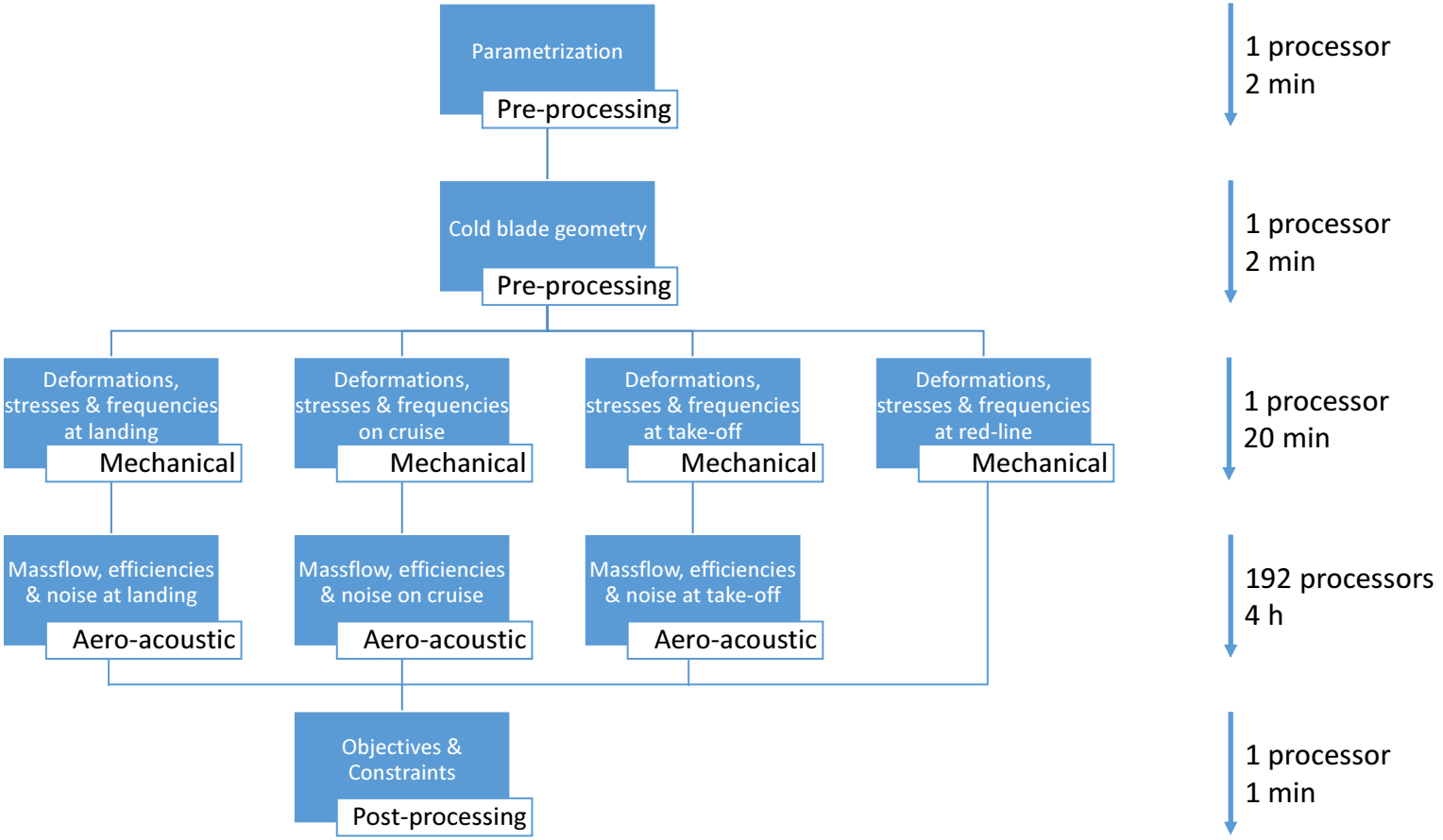


Figure 4.23: Schematic architecture of the simulation chain and typical runtime of the successive computations.

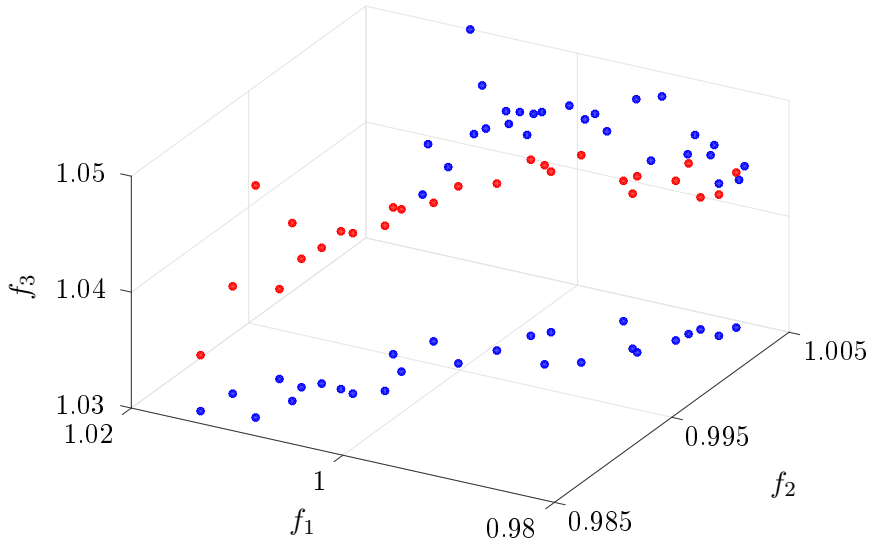


Figure 4.24: Non-dominated solutions obtained with the BMOO algorithm (red disks). To help the visualization, projections in the coordinates planes (f_1, f_2) and (f_1, f_3) are shown as blue disks. The values are scaled around one to preserve the confidentiality of the results.

The optimization is made using the BMOO algorithm. The algorithm is initialized using an initial design of 120 experiments that were computed beforehand. Among these, 25 resulted in simulation failures. These can be due to a combination of the design parameters resulting in a non-feasible blade geometry, to an insufficient meshing quality or to a failure of the aerodynamic computations to converge.

Due to complications, the optimization was run over 77 iterations only. The non-dominated solutions found by the algorithm are shown in Figure 4.24. The results reveal that the maximization of the efficiencies at take-off f_1 and on cruise f_2 are antagonist objectives, which was not obvious beforehand. Also, we note a drop of the values of the natural frequency of the blade at the redline f_3 when the values of the efficiency at take-off become high. The presence of an outlier with an high value of this frequency and an high value of efficiency seems to contradict this observation though, which may indicate that the algorithm was stopped too prematurely. Nevertheless, the shape of the front seems established even though further refining would certainly be possible.

Among the 77 additional evaluations, 28 resulted in simulation failures, which seems to indicate that the region of optimal solutions is close to the non-observable region. This could also be due to a failure of the classification model[1]. However, such a behaviour was not observed in the other applications presented in this chapter. An expert analysis of the causes of failure of the simulation would be required to better understand this particular point but it could not be made during this thesis.

[1] In this work, we use a nearest neighboor classifier with $k = 5$ neighboors (see Section 3.5.3).

4.5.4 Conclusions

In this section, BMOO has been applied to the design of a turbomachine fan blade. This component of the turbomachine is one of the most dimensioning and its optimization has to take into account mechanical, aerodynamic and acoustic aspects for different regimens of the turbomachine. Besides, the optimization is made difficult by the heavy computational cost associated with the numerical simulation of the system.

The considered optimization problem takes the form of a constrained multi-objective problem with three objectives and twenty six design variables. BMOO obtains a satisfying approximation of the Pareto front using only 77 iterations, hence achieving a significant reduction of the computing time required for solving this kind of problem. In particular, the analysis of the results provides a better understanding of the interactions between the objectives.

Acknowledgements: The authors would like to thank Emmanuel Cheriere (Cénaéro), Caroline Sainvitu (Cénaéro), Vincent Baudoui (Cénaéro) and Abdelkader Otsmane (Safran Aircraft Engines) for their contributions to this work.

4.6 Conclusions

In this chapter, we have presented applications of the BMOO algorithm to design problems that are representative of the kind of optimization problems encountered in the industry. The applications feature different physics: thermodynamics, mechanics, aerodynamics, acoustics, electromagnetism and automatic control and it is shown that BMOO is a transverse algorithm that can be used to solve this kind of problems using a reasonable number of functions evaluations.

For all four considered applications, the use of a multi-objective formulation enables the user to choose design solutions with a posteriori knowledge about the possible trade-off between the objectives. In particular, the analysis of the variation of the design variables along the front of non-dominated solutions makes it possible to better understand the influence of design choices on the performances of the system.

Furthermore, these applications were the opportunity to emphasize the effectiveness of the algorithm's extensions presented in Section 3.5. First, viewing the cheap-to-evaluate constraint functions as restrictions of the design space makes it possible to decrease the size of the optimization problem and to avoid regions where the physical model's assumptions are not met. This results in fewer simulation failures and a better control over the simulator. Second, to address simulation failures, in this work, a simple classification model is used (see Section 3.5.3). This makes it possible to perform the optimization despite the presence of non-observable regions. However, the results obtained on the turbomachine fan blade application seem to indicate that a more elaborate approach might be required on some problems. This shall motivate future work on the handling of the hidden constraints. Finally, as regards the use of batches of experiment, it is shown on the line of sight control problem that significant savings can be achieved in terms of computing time without impacting too much on the quality of the optimization. We believe that the use of batch approaches within multi-objective formulations such as the one we propose should receive more attention in the near future because they are very effective.

Chapter 5

Conclusions

5.1 Summary of contributions

In this thesis, we address the problem of the derivative-free multi-objective optimization of real-valued functions subject to multiple inequality constraints. More specifically, we consider a setting where the objectives and constraints of the problem are evaluated simultaneously using a potentially time-consuming computer program. In this setting, it is highly desirable to solve the optimization problem using as few functions evaluations as possible. Morevover, we focus on problems that are heavily constrained, in the sense that finding feasible solutions; i.e. solutions that respect all the constraints of the problem, is difficult. This is a setting that one often encounters when dealing with the design optimization of complex systems and which poses difficulties to many optimization algorithms.

To solve this problem, we propose a Bayesian optimization algorithm called BMOO[1]. This algorithm implements a new expected improvement sampling criterion crafted to apply to potentially heavily constrained problems and to many-objective problems, i.e. problems with several objective functions. This criterion stems from the use of the hypervolume of the dominated region as a loss function, where the dominated hypervolume is defined using an extended domination rule that applies jointly on the objectives and constraints. Several criteria from the Bayesian optimization literature are recovered as special cases.

The criterion takes the form of an integral over the space of objectives and constraints for which no closed form expression exists in the general case. Besides, it has to be optimized at every iteration of the algorithm to select the next evaluation point, which is known to be difficult because expected improvement criteria tend to be sharp multi-modal functions. To solve these difficulties, original sequential Monte-Carlo algorithms in line with previous work carried out by Benassi (2013) in the unconstrained global optimization setting are developped.

Moreover, four extensions of the algorithm are proposed, which constitute each a contribution of independent interest. First, BMOO is extended to handle problems defined on non-hypercubic design spaces. These can be defined by cheap-to-evaluate constraints and/or membership functions. Second, it is extended to handle problems with hidden constraints. These arise, for example, when the computer program used to evaluate the functions of the problem systematically fails to return a result for some combinations of the design variables. Third, to take advantage of parallel computation facilities when available, a multi-point version of the algorithm is proposed. Last, we propose an alternative sampling criterion called the expected weighted hypervolume improvement criterion. This criterion makes it possible to orient the search for optimal solutions according to user preferences, thus allowing the expert to step in the optimization loop.

5.2 Main achievements and limitations

The proposed algorithm achieves good results on classical single- and multi-objective test problems from the literature when compared with state-of-the-art algorithms. It is shown to apply to a large class of problems with a good repetability. In particular, it is capable to solve heavily

[1] BMOO stands for Bayesian Many-Objective Optimizaton.

constrained problems using few functions evaluations, which was one of the objectives of this work; and to address many-objective problems, which was another one.

BMOO is also successfully applied to four industrial-like design optimization problems from different fields. It is applied to the design of a commercial aircraft environment control system (collaboration with Airbus), to the design of an electric vehicle powertrain (collaboratin with Renault), to the tuning of a line of sight controller (collaboration with Safran Electronics & Defense) and to the design of a turbomachine fan blade (collaboration with Cénaéro and Safran Aircraft Engines). It is shown that the proposed extensions are valuable tools that can be used to help solve complex optimization problems.

The algorithm is subject to limitations though, which are highlighted throughout the manuscript. First, it is not suitable for problems having non-stationary objectives and/or constraints, which is typical of Bayesian optimization algorithms relying on stationary Gaussian processes if nothing is done on the modelling aspect to address this limitation. Secondly, the algorithm uses the dominated hypervolume as a loss function. As such, it can achieve mitigated results on multi-objective problems with concave Pareto fronts. Again, this is an expected behaviour.

5.3 Perspectives for future work

Several research tracks could be explored in the futur. In this work, we have been focusing on EI-based approaches. However, other approaches in the Bayesian optimization literature, such as *Stepwise Uncertainty reduction* approaches (see e.g Villemonteix et al. (2009); Vazquez and Bect (2014); Chevalier et al. (2014a); Picheny (2014a,b); Hernández-Lobato et al. (2015, 2016b); Garrido-Merchán and Hernández-Lobato (2016)) and the *Augmented Lagrangian* approach of Gramacy et al. (2016), and, more generally, in the model-based optimization literature (see e.g. Regis (2016) and references therein), provide interesting alternatives that call for a comparison of optimization performances.

As dicussed above, the proposed algorithm is subject to limitations. First, stationary Gaussian process models are not suitable for problems with non-stationary functions. In Section 2.5.3 of this manuscript it is shown that this issue can sometimes be solved using simple transformations of the non-stationary functions. However, finding such a transformation is not straightforward and it remains up to the practitionner. Several types of models proposed in the literature— warped Gaussian processes (Snelson et al., 2004), non-stationary Gaussian processes (Toal and Keane, 2012), deep Gaussian processes (Damianou and Lawrence, 2013), etc.— provide interesting directions regarding this issue that could be the object of future work.

Secondly, as mentionned in Section 4.4, the proposed algorithm is not suitable as is for problems having constraints on the objectives, situation that often occurs in practice when one is interested in discovering the optimal trade-off between different objectives but does not want to deteriorate one in favour of an other too much. Recent work from Yang et al. (2016a,b) provide interesting directions for future work on this issue.

Computational aspects could also be discussed. While the overall computing time of the algorithm remains reasonable on small to medium size problems such as the ones adressed in

this manuscript, it can become prohibitive on large problems (this is to be balanced with the computing time of the functions of the problem). In this work, we do not challenge our choice of an anisotropic Gaussian random-walk Metropolis-Hastings algorithm to move the particles within the sequential Monte-Carlo algorithms used for the computation and optimization of the expected improvement criterion. However, much improvement could probably be achieved by using more suitable algorithms. In particular, the choice of the number of particles to be used in the algorithms, though briefly discussed in Section 3.3 for the criterion computation problem, remains empirical and improvements could probably be achieved on this aspect.

The extensions proposed in Section 3.5 deserve more attention. Empirical evidence of their effectiveness is given in Chapter 4 but they could be improved. First, the choice was made to consider applications where the objectives and constraints are obtained simultaneously from one call to some possibly computationaly intensive computer program. This assumption was relaxed to allow for the introduction of cheap-to-evaluate constraint functions that can be evaluated separately, these being considered as restrictions of the search space, or expensive-to-evaluate. In some applications though, it may happen that the computational cost associated with the problem's functions vary from one function to the other and that these functions can be evaluated independently. In such a setting, significant time savings could probably be achieved by using a more adapted strategy.

Secondly, the choice of a nearest-neighboor classifier to handle hidden constraints is questioned in Section 4.5. More sophisticated approaches such as random forest classification, support vector machines or artificial neural network classification might be required on some problems and could trigger further improvement of the proposed algorithm.

Third, much work remains to make the EWHI criterion proposed in Section 3.5 usable in practice. In particular, the choice *a priori* of the weight function, i.e. before the position of the Pareto front is known, is an open research question.

Other extensions could be developped to make BMOO applicable to a broader range of problems, such as problems featuring equality constraints, discrete variables (see Section 4.3) or noisy functions. The capability to use gradient information when available, to deal with multi-fidelity models, to stop the algorithm when knowledge of the Pareto front is sufficient and to adress high dimensional problems would also be useful extensions. Recent works from Binois et al. (2015b,a); Wang et al. (2016b); Azzimonti et al. (2016); Roy et al. (2017); Garrido-Merchán and Hernández-Lobato (2017); Wu et al. (2017); Qian and Yu (2017) on some of these aspects could be a start point for further investigations.

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

Bibliography

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

C. Andrieu and J. Thoms. A tutorial on adaptive mcmc. *Statistics and Computing*, 18(4): 343–373, 2008.

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.

C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006.

Y. Auffray, P. Barbillon, and J. M. Marin. Maximin design on non hypercube domains and kernel interpolation. *Statistics and Computing*, 22(3):703–712, 2012.

A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 555–562. ACM, 2009a.

A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 563–570. ACM, 2009b.

A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102. ACM, 2009c.

A. Auger, J. Bader, and D. Brockhoff. Theoretically investigating optimal μ-distributions for the hypervolume indicator: first results for three objectives. In *International Conference on Parallel Problem Solving from Nature*, pages 586–596. Springer, 2010.

J. Azimi, A. Fern, and X. Z. Fern. Batch bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems*, pages 109–117, 2010.

D. Azzimonti, J. Bect, C. Chevalier, and D. Ginsbourger. Quantifying uncertainties on excursion sets under a gaussian random field prior. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):850–874, 2016.

T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996. ISBN 0-19-509971-0.

T. Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

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

KB Baltzis. The femm package: A simple, fast, and accurate open source electromagnetic tool in science and engineering. *Journal of Engineering Science & Technology Review*, 1(1), 2008.

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

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, L. Li, and E. Vazquez. Bayesian subset simulation. *arXiv preprint arXiv:1601.02557*, 2016a.

J. Bect, E. Vazquez, et al. STK: a Small (Matlab/Octave) Toolbox for Kriging. Release 2.4 (to appear), 2016b. URL http://kriging.sourceforge.net.

A. Bejan and D. L. Siems. The need for exergy analysis and thermodynamic optimization in aircraft development. *Exergy, An International Journal*, 1(1):14–24, 2001.

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.

G. Bertotti. Physical interpretation of eddy current losses in ferromagnetic materials. i. theoretical considerations. *Journal of applied Physics*, 57(6):2110–2117, 1985.

N. Beume. S-metric calculation by considering dominated hypervolume as klee's measure problem. *Evolutionary Computation*, 17(4):477–492, 2009.

N. Beume, B. Naujoks, and M. Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

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.

M. Binois, D. Ginsbourger, and O. Roustant. Quantifying uncertainty on pareto fronts with gaussian process conditional simulations. *European Journal of Operational Research*, 243(2): 386–394, 2015a.

M. Binois, D. Ginsbourger, and O. Roustant. A warped kernel improving robustness in bayesian optimization via random embeddings. In *International Conference on Learning and Intelligent Optimization*, pages 281–286. Springer, 2015b.

J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

A. J. Booker, J. E. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13, 1999.

P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE transactions on evolutionary computation*, 7(2):174–188, 2003.

G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

K. Bringmann and T. Friedrich. The maximum hypervolume set yields near-optimal approximation. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 511–518. ACM, 2010.

D. Brockhoff, T. Friedrich, and F. Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature–PPSN X*, pages 651–660. Springer, 2008.

R. H. Byrd, M. E Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

A. J. M. Cardoso, S. M. A. Cruz, and D. S. B. Fonseca. Inter-turn stator winding fault diagnosis in three-phase induction motors, by park's vector approach. *IEEE Transactions on Energy Conversion*, 14(3):595–598, 1999.

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.

R. B. Chen, Y. W. Hsu, Y. Hung, and W. Wang. Discrete particle swarm optimization for constructing uniform design on irregular regions. *Computational Statistics & Data Analysis*, 72:282–297, 2014.

C. Chevalier and D. Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. In *International Conference on Learning and Intelligent Optimization*, pages 59–69. Springer, 2013.

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):455–465, 2014a.

C. Chevalier, V. Picheny, and D. Ginsbourger. Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational Statistics & Data Analysis*, 71:1021–1034, 2014b.

C. A. Coello. An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)*, 32(2):109–143, 2000.

C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary algorithms for solving multi-objective problems*, volume 242. Springer, 2002.

Y. Collette and P. Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2013.

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.

I. Couckuyt, D. Deschrijver, and T. Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. *Journal of Global Optimization*, 60(3):575–594, 2014.

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

E. A. Daxberger and B. K. H. Low. Distributed batch gaussian process optimization. In *International Conference on Machine Learning*, pages 951–960, 2017.

K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

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.

K. Deb, M. Mohan, and S. Mishra. Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. *Evolutionary computation*, 13(4):501–525, 2005.

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.

T. Desautels, A. Krause, and J. W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

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.

V. Dubourg, B. Sudret, and J. M. Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673–690, 2011.

M. Emerich, K. Yang, A. Deutz, H. Wang, and M. Fonseca. Multicriteria generalization of bayesian global optimization. *Advances in Stochastic and Global Optimization. Springer*, pages 223–236, 2016.

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. *Technical report, Leiden University*, 2008.

M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer, 2005.

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.

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

M. Emmerich, A. H. Deutz, and I. Yevseyeva. On reference point free weighted hypervolume indicators based on desirability functions and their probabilistic interpretation. *Procedia Technology*, 16:532–541, 2014.

P. Feliot, Y. Le Guennec, J. Bect, and E. Vazquez. Design of a commercial aircraft environment control system using bayesian optimization techniques. *arXiv preprint arXiv:1610.02271*, 2016.

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

M. Fleischer. The measure of pareto optima. applications to multi-objective metaheuristics. In *Evolutionary multi-criterion optimization*, pages 519–533. Springer, 2003.

P. J. Fleming, R. C. Purshouse, and R. J. Lygoe. Many-objective optimization: An engineering design perspective. In *International conference on evolutionary multi-criterion optimization*, pages 14–32. Springer, 2005.

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.

A. I. J. Forrester, A. Sobester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

S. Frasnedo. *Optimisation des lois de commande d'un imageur sur critère optronique. Application à un imageur à deux étages de stabilisation.* PhD thesis, Centrales-Supélec, 2016.

S. Frasnedo, J. Bect, C. Chapuis, G. Duc, P. Feyel, and G. Sandou. Line of sight controller tuning using bayesian optimization of a high-level optronic criterion. *IFAC-PapersOnLine*, 48 (25):56–61, 2015a.

S. Frasnedo, C. Chapuis, G. Duc, P. Feyel, and G. Sandou. Line of sight stabilization controllers tuning from high-level modulation transfer function specifications. *IFAC-PapersOnLine*, 48 (9):13–18, 2015b.

P. I. Frazier and S. C. Clark. Parallel global optimization using an improved multi-points expected improvement criterion. In *INFORMS Optimization Society Conference, Miami FL*, volume 26, 2012.

T. Friedrich, C. Horoba, and F. Neumann. Multiplicative approximations and the hypervolume indicator. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 571–578. ACM, 2009.

M. C. Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.

E. C. Garrido-Merchán and D. Hernández-Lobato. Predictive entropy search for multi-objective bayesian optimization with constraints. *arXiv preprint arXiv:1609.01051*, 2016.

E. C. Garrido-Merchán and D. Hernández-Lobato. Dealing with integer-valued variables in bayesian optimization with gaussian processes. *arXiv preprint arXiv:1706.03673*, 2017.

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

M. A. Gelbart, J. Snoek, and R. P. Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, turbulence and combustion*, 65(3-4):393–415, 2000.

D. Ginsbourger and R. Le Riche. Towards Gaussian process-based optimization with finite time horizon. In *mODa 9 – Advances in Model-Oriented Design and Analysis*, pages 89–96. Springer, 2010.

D. Ginsbourger, J. Janusevskis, and R. Le Riche. Dealing with asynchronicity in parallel gaussian process based global optimization. *hal preprint hal-00507632*, 2010a.

D. Ginsbourger, R. Le Riche, and L. Carraro. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162, 2010b.

R. Girdziusas, R. Le Riche, F. Viale, and D. Ginsbourger. Parallel budgeted optimization applied to the design of an air duct. *hal preprint hal-00723427*, 2012.

J. González, Z. Dai, P. Hennig, and N. D. Lawrence. Batch bayesian optimization via local penalization. *arXiv preprint arXiv:1505.08052*, 2015.

R. B. 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*, 58 (1):1–11, 2016.

J. Guerra. *Optimisation multi-objectif sous incertitudes de phénomènes de thermique transitoire.* PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), 2016.

A. Habib, H. K. Singh, and T. Ray. A multi-objective batch infill strategy for efficient global optimization. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 4336–4343. IEEE, 2016.

S. P. Han. A globally convergent method for nonlinear programming. *Journal of optimization theory and applications*, 22(3):297–309, 1977.

N. Hansen and S. Kern. Evaluating the cma evolution strategy on multimodal test functions. In *International Conference on Parallel Problem Solving from Nature*, pages 282–291. Springer, 2004.

E. C. Harrington. The desirability function. *Industrial quality control*, 21(10):494–498, 1965.

M. Hauschild and M. Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, 2011.

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

J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani. A general framework for constrained bayesian optimization using information-based search. *Journal of Machine Learning Research*, 2016a.

J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani. A general framework for constrained bayesian optimization using information-based search. *Journal of Machine Learning Research*, 2016b.

José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International Conference on Machine Learning*, pages 1699–1707, 2015.

R. Hooke and T. A. Jeeves. "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl. Model-based multi-objective optimization: Taxonomy, multi-point proposal, toolbox and benchmark. In *Evolutionary Multi-Criterion Optimization*, pages 64–78. Springer, 2015.

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

I. Hupkens, A. Deutz, K. Yang, and M. Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 65–79. Springer, 2015.

W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, 1999.

C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28, 2007.

H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *IEEE congress on evolutionary computation*, pages 2419–2426, 2008.

J. Janusevskis, R. Le Riche, D. Ginsbourger, and R. Girdziusas. Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In *Learning and Intelligent Optimization*, pages 413–418. Springer, 2012.

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

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.

S. Jiang, Y. S. Ong, J. Zhang, and L. Feng. Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE transactions on cybernetics*, 44(12):2391–2404, 2014.

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, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

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.

T. Kathuria, A. Deshpande, and P. Kohli. Batched gaussian process bandit optimization via determinantal point processes. *arXiv preprint arXiv:1611.04088*, 2016.

A. J. Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.

S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

J. P. C. Kleijnen. *Design and analysis of simulation experiments*, volume 20. Springer, 2008.

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 D. Corne. On metrics for comparing nondominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02.*, volume 1, pages 711–716. IEEE, 2002.

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

J. D. Knowles, D. W. Corne, and M. Fleischer. Bounded archiving using the lebesgue measure. In *Proceedings of the 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 4, pages 2490–2497. IEEE, 2003.

S. Koziel, D. E. Ciaurri, and L. Leifsson. Surrogate-based methods. In *Computational Optimization, Methods and Algorithms*, pages 33–59. Springer, 2011.

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.

R. Lacour, K. Klamroth, and C. M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79:347–360, 2017.

M. Laumanns, G. Rudolph, and H. P. Schwefel. *Approximating the pareto set: Concepts, diversity issues, and performance assessment.* Secretary of the SFB 531, 1999.

R. Le Riche, R. Girdziušas, and J. Janusevskis. A study of asynchronous budgeted optimization. In *NIPS workshop on Bayesian Optimization and Decision Making*, 2012.

H. K. H. Lee, R. B. Gramacy, C. Linkletter, and G. A. Gray. Optimization subject to hidden constraints via statistical emulation. *Pacific Journal of Optimization*, 7(3):467–478, 2011.

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.

L. Li, I. Yevseyeva, V. Basto-Fernandes, H. Trautmann, N. Jing, and M. Emmerich. An ontology of preference-based multiobjective evolutionary algorithms. *arXiv preprint arXiv:1609.08082*, 2016a.

M. Li. *Evolutionary many-objective optimisation: pushing the boundaries.* PhD thesis, Brunel University London, 2015.

Z. Li, S. Ruan, J. Gu, X. Wang, and C. Shen. Investigation on parallel algorithms in efficient global optimization based on multiple points infill criterion and domain decomposition. *Structural and Multidisciplinary Optimization*, pages 1–27, 2016b.

J. S. Liu. *Monte Carlo strategies in scientific computing.* Springer, 2001.

M. Locatelli. Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, 10(1):57–76, 1997.

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

C. Luo, K. Shimoyama, and S. Obayashi. A study on many-objective optimization using the kriging-surrogate-based evolutionary algorithm maximizing expected hypervolume improvement. *Mathematical Problems in Engineering*, 2015, 2015.

R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

S. Marmin, C. Chevalier, and D. Ginsbourger. Differentiating the multipoint expected improvement for optimal batch design. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 37–48. Springer, 2015.

S. Marmin, C. Chevalier, and D. Ginsbourger. Efficient batch-sequential bayesian optimization with moments of truncated gaussian vectors. *arXiv preprint arXiv:1609.02700*, 2016.

E. Mezura-Montes and C. A. C. Coello. Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm and Evolutionary Computation*, 1(4):173–194, 2011.

K. Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

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

J. Mockus. *Bayesian approach to global optimization: theory and applications*, volume 37. Springer, 2012.

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

J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

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.

I. Pérez-Grande and T. J. Leo. Optimization of a commercial aircraft environmental control system. *Applied thermal engineering*, 22(17):1885–1904, 2002.

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, 2014a.

V. Picheny. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, DOI:10.1007/s11222-014-9477-x:1–16, 2014b.

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. Springer, 2008.

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162, 1964.

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.

H. Qian and Y. Yu. Solving high-dimensional multi-objective optimization problems with low effective dimensions. In *AAAI*, pages 875–881, 2017.

N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005.

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.

R. G. Regis. Multi-objective constrained black-box optimization using radial basis function surrogates. *Journal of Computational Science*, 16:140–155, 2016.

C. Robert and G. Casella. *Monte Carlo statistical methods. Second Edition.* Springer, 2004.

G. O. Roberts and J. S. Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

S. Roy, K. Moore, J. T. Hwang, J. S. Gray, W. A. Crossley, and J. Martins. A mixed integer efficient global optimization algorithm for the simultaneous aircraft allocation-mission-design problem. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1305, 2017.

Optimization with missing data, volume 462, 2006. The Royal Society.

S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River, 2003.

T. J. Santner, B. J. Williams, and W. Notz. *The design and analysis of computer experiments.* Springer, 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.

A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, pages 3330–3338, 2015.

K. Shimoyama, S. Jeong, and S. Obayashi. Kriging-surrogate-based optimization considering expected hypervolume improvement in non-constrained many-objective test problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 658–665. IEEE, 2013a.

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, 2013b.

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.

E. Stinstra, D. den Hertog, P. Stehouwer, and A. Vestjens. Constrained maximin designs for computer experiments. *Technometrics*, 45(4):340–346, 2003.

R. Suich and G. C. Derringer. Is the regression equation adequate?: One criterion. *Technometrics*, pages 213–216, 1977.

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

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.

E. Tekin and I. Sabuncuoglu. Simulation optimization: A comprehensive review on theory and applications. *IIE transactions*, 36(11):1067–1081, 2004.

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

V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7 (1):1–25, 1997.

A. Torn and A. Zilinskas. *Global optimization*. Springer-Verlag New York, Inc., 1989.

J. V. C. Vargas and A. Bejan. Thermodynamic optimization of finned crossflow heat exchangers for aircraft environmental control systems. *International Journal of Heat and Fluid Flow*, 22 (6):657–665, 2001.

E. Vazquez and J. Bect. A new integral loss function for Bayesian optimization. *arXiv preprint arXiv:1408.4622*, 2014.

F. A. C. Viana and R. T. Haftka. Surrogate-based optimization with parallel simulations using the probability of improvement. In *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 13–15. Fort Worth USA, 2010.

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 and H. Trautmann. Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation*, 14(5):688–701, 2010.

T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *International conference on evolutionary multi-criterion optimization*, pages 742–756. Springer, 2007.

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.

G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, 2007.

J. Wang, S. C. Clark, E. Liu, and P. I. Frazier. Parallel bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*, 2016a.

Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016b.

T. Weise. *Global optimization algorithms-theory and application*. Citeseer, 2009.

L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, 2012.

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. Springer, 2010.

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

J. Wu and P. I. Frazier. The parallel knowledge gradient method for batch bayesian optimization. *arXiv preprint arXiv:1606.04414*, 2016.

J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier. Bayesian optimization with gradients. *arXiv preprint arXiv:1703.04389*, 2017.

K. Yang, A. Deutz, Z. Yang, T. Bäck, and M. Emmerich. Truncated expected hypervolume improvement: Exact computation and application. In *Evolutionary Computation (CEC), 2016 IEEE Congress on, Submitted*, 2016a.

K. Yang, L. Li, A. Deutz, T. Back, and M. Emmerich. Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on*, pages 276–281. IEEE, 2016b.

S. Yang, M. Li, X. Liu, and J. Zheng. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013.

Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by MOEA/D with gaussian process model. *Evolutionary Computation, IEEE Transactions on*, 14(3):456–474, 2010.

A. A. Zhigljavsky. *Theory of global random search*, volume 65. Springer Science & Business Media, 2012.

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

E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms–a comparative case study. In *Parallel problem solving from nature–PPSN V*, pages 292–301. Springer, 1998.

E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving Strength Pareto Evolutionary Algorithm. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.

E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

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

université PARIS-SACLAY

CentraleSupélec

Titre: Une approche Bayésienne pour l'optimisation multi-objectif sous contraintes

Mots clés: Optimisation Bayésienne, Multi-objectif, Monte Carlo séquentiel

Résumé: Ces travaux de thèse portent sur l'optimisation multi-objectif de fonctions à valeurs réelles sous contraintes d'inégalités. En particulier, nous nous intéressons à des problèmes pour lesquels les fonctions objectifs et contraintes sont évaluées au moyen d'un programme informatique nécessitant potentiellement plusieurs heures de calcul pour retourner un résultat. Dans ce cadre, il est souhaitable de résoudre le problème d'optimisation en utilisant le moins possible d'appels au code de calcul. Afin de résoudre ce problème, nous proposons dans cette thèse un algorithme d'optimisation Bayésienne baptisé BMOO. Cet algorithme est basé sur un nouveau critère d'amélioration espérée spécifiquement développé afin d'être applicable à des problèmes fortement contraints et/ou avec de nombreux objectifs. Le critère proposé généralise plusieurs critères classiques d'amélioration espérée issus de la littérature. Il prend la forme d'une intégrale définie sur l'espace des objectifs et des contraintes pour laquelle aucune forme fermée n'est connue dans le cas général. De plus, il doit être optimisé à chaque itération de l'algorithme. Afin de résoudre ces difficultés, des algorithmes de Monte-Carlo séquentiel sont également proposés. L'efficacité de BMOO est démontrée à la fois sur des cas tests académiques et sur quatre problème d'optimisation représentant de réels problémes de conception.

Title: A Bayesian approach to constrained multi-objective optimization

Keywords: Bayesian optimization, multi-objective, sequential Monte Carlo

Summary: In this thesis, we address the problem of the derivative-free multi-objective optimization of real-valued functions subject to multiple inequality constraints. In particular, we consider a setting where the objectives and constraints of the problem are evaluated simultaneously using a potentially time-consuming computer program. To solve this problem, we propose a Bayesian optimization algorithm called BMOO. This algorithm implements a new expected improvement sampling criterion crafted to apply to potentially heavily constrained problems and to many-objective problems. This criterion stems from the use of the hypervolume of the dominated region as a loss function, where the dominated region is defined using an extended domination rule that applies jointly on the objectives and constraints. Several criteria from the Bayesian optimization literature are recovered as special cases. The criterion takes the form of an integral over the space of objectives and constraints for which no closed form expression exists in the general case. Besides, it has to be optimized at every iteration of the algorithm. To solve these difficulties, specific sequential Monte-Carlo algorithms are also proposed. The effectiveness of BMOO is shown on academic test problems and on four real-life design optimization problems.