Uncertainty quantification on pareto fronts and high-dimensional strategies in bayesian optimization, with applications in multi-objective automotive design

Mickaël Binois

NNT : 2015 EMSE 0805

THÈSE

présentée par

Mickaël BINOIS

pour obtenir le grade de
Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Mathématiques Appliquées

UNCERTAINTY QUANTIFICATION ON PARETO FRONTS AND HIGH-DIMENSIONAL STRATEGIES IN BAYESIAN OPTIMIZATION, WITH APPLICATIONS IN MULTI-OBJECTIVE AUTOMOTIVE DESIGN

QUANTIFICATION D'INCERTITUDE SUR FRONTS DE PARETO ET STRATÉGIES POUR L'OPTIMISATION BAYÉSIENNE EN GRANDE DIMENSION, AVEC APPLICATIONS EN CONCEPTION AUTOMOBILE

soutenue à Saint-Étienne, le 3 décembre 2015

Membres du jury

Président :	A. RUIZ-GAZEN	Professeur, Toulouse School of Economics
Rapporteurs :	N. LAWRENCE	Professor, University of Sheffield
	L. PRONZATO	Directeur de recherche, CNRS, UNICE
Examinateurs :	A. FORRESTER	Senior lecturer, University of Southampton
	R. LE RICHE	Directeur de recherche, CNRS, ENSM-SE
Directeurs de thèse :	D. GINSBOURGER	Senior researcher, Idiap & Dozent, UniBE
	O. ROUSTANT	Maître assistant, ENSM-SE
Co-encadrant :	F. MERCIER	Ingénieur de recherche, Renault
Invités:	T. ESPINASSE	Maître de conférences, Institut Camille Jordan
	Y. TOURBIER	Expert optimisation, Renault

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	A. Dolgui, Professeur
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	S. Dauzere Peres, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	CR	Génie industriel	CMP
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTAIA-GUSCHINSKAYA	Olga	CR	Génie industriel	FAYOL
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BERGER DOUCE	Sandrine	PR2	Sciences de gestion	FAYOL
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	MA(MDC)	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR1	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GAVET	Yann	MA(MDC)	Image Vision Signal	CIS
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean-Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MAURINE	Philippe	Ingénieur de recherche	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Frank	DR	Sciences et génie des matériaux	SMS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NEUBERT	Gilles	PR	Génie industriel	FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche		CMP
NORTIER	Patrice	PR1		SPIN
OWENS	Rosin	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROBISSON	Bruno	Ingénieur de recherche	Microélectronique	CMP
ROUSSY	Agnès	MA(MDC)	Génie industriel	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
ROUX	Christian	PR	Image Vision Signal	CIS
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FEULVARCH	Eric	MCF	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 29/06/2015

Remerciements

J'ai eu le plaisir d'assister aux cours d'Olivier Roustant et de David Ginsbourger lors de mon cursus aux Mines, et ils ne sont pas pour rien dans ma poursuite en thèse. Avec Frédéric Mercier, ils ont formé une équipe d'encadrement extraordinaire. Je ne saurais les remercier assez pour leur patience, leur disponibilité, leur enthousiasme, leurs conseils et leurs qualités tant scientifiques qu'humaines.

I would like to thank Neil Lawrence and Luc Pronzato for having accepted to review this manuscript and for their detailed reports and suggestions. I am also grateful to my jury members, Alexander Forrester, Rodolphe Le Riche, Anne Ruiz-Gazen, Thibault Espinasse and Yves Tourbier.

Lors de ces trois années, j'ai eu le plaisir de travailler dans trois environnements différents et d'assister à différents séminaires ou conférences où j'ai eu l'opportunité de rencontrer et de discuter avec de nombreuses personnes. Une mention spéciale à Didier Rullière pour la plongée dans le monde des copules et à Victor Picheny pour la motivation apportée sur le développement conjoint de *GPareto*. J'ai eu la chance de faire partie du consortium ReDICE qui a contribué à une très forte émulation. Dans ce sens, je remercie tout particulièrement Fabrice Gamboa, Luc Pronzato, Bertrand Iooss, Clément Chevalier, Yann Richet, Miguel Munoz Zuniga, Delphine Sinoquet, Jean Giorla, Céline Helbert, Jean-Marc Martinez, Jean Baccou, François Wahl, François Bachoc, Loïc le Gratiet, Clément Walter, Guillaume Damblin, et enfin Yves Deville pour ses précieux conseils en R. Le GdR Mascotnum a été une source de motivation supplémentaire et je salue ses membres avec qui j'ai pu échanger.

Je tiens maintenant à remercier toutes ces personnes à Saint-Etienne, Guyancourt et Berne qui ont rendu cette aventure possible et agréable, j'y repasserai toujours avec joie.

J'ai passé d'excellents moments à Saint-Étienne, à l'Institut Fayol. Je garderai notamment des souvenirs de ces pauses café rafraichissantes avec Éric (et ses énigmes), Olivier, Xavier, Mireille, Xavier, Frédéric, Nicolas, Rodolphe, Paolo, Jean-François, Olga et Christine (merci pour l'aide sur les questions logistiques et administratives). Bon courage aux doctorants qui ont bientôt fini, mon complice Espéran (expert en transport de canapé), Afafe (ting ting),

Hossein (pour une via ferrata mémorable), Hakim, Jean-Charles et bonne continuation à ceux qui ont terminé, Oussama, Akram (ting ting), Malik, José, Hassan, Jana, Momchil, John et Lounes. Un grand merci aux joggeurs qui m'ont permis de découvrir les environs parfois injustement sous-estimés (ou non) de la ville : Rodolphe, Nicolas, Gauthier, Didier, Nilou et Nicolas (pour cette Saintélyon 2015).

Mon passage au Technocentre a été non moins riche en émotions, pauses café et surprises. Merci à l'équipe optimisation du village gaulois : Pascal (pour tes blagues), Frédéric, Marc (pour ton enthousiasme), Marc, Christian, Daniel, Sylvain (merci pour le coup de main sur les cas tests), Gireg, Jean Luc, Paul, Pierre, Timothée, Cédric, Sarah, Félix, François, Thuy, Yves et à mon 'frère' de thèse Laurent. Merci aux squasheurs : Stéphane, Nicolas, Sylvain (pour l'organisation impeccable, les conseils et les rares 11-0), Jean-Gilles, Yannick et Éric.

Les passages en Suisse ont été trop courts, mais à chaque fois intenses. Heureusement que l'on a pu se croiser ailleurs aussi ! Merci donc à l'équipe de choc bernoise : Sébastien, David, Mint, Clément et Dario pour leur accueil toujours chaleureux.

Avant de remercier mes amis de plus longue date, je tiens à remercier de nouveau ceux qui m'ont hébergé un peu partout, et en particulier mes colocs parisiens David et Laurent. Et je remercie mes vrais colocs, Maud, Sarah, Pauline, Julien et Lorane qui ont illuminé cette dernière année. Parmi les anciens mineurs, merci à Simon (j'espère que l'on se recroisera aussi en conférence), Lauriane, Tarek, Maxime, Rémi et au 4. Enfin merci aux bretons, notamment Elodie et Thibaut qui se sont essayés à la relecture, Grégoire, David, Matthieu, Benjamin et Kévin.

Enfin, un immense merci à ma sœur Marilyne, à mes parents Maryse et Gilles ainsi qu'au reste de ma famille pour leur soutien indéfectible.

Contents

List of Figures

List of Tables

Part I

Introduction and context

Chapter 1

Introduction

In many engineering applications, physical experiments or computer codes may have prohibitive costs or evaluation times. Yet, they are now intensively used to design and optimize complex systems, such as vehicles in the automotive industry. In this context, the computational bottleneck is very often the evaluation time of these expensive functions. As a direct consequence, the evaluation budget dedicated to optimization is severely limited, rendering a manual trial and error process inadequate. In addition, no gradient nor information about properties of the considered function such as monotonicity are supposed available, i.e. we are in a *black-box* setting. Thus looking for the optimum by applying the steepest descent technique is inconvenient: the gradient needs to be approximated, which is quite costly and may only result in finding a local solution. Starting from many different points could be successful in finding the global solution, i.e. the best over the whole range of alternatives, but is even more costly.

A preferred option is to construct a surrogate model (also called metamodel) of the expensive function with as few evaluations as possible and use it to predict the outputs anywhere on the research domain. These techniques are very common in the computer experiments and machine learning literature, see e.g. [SWN03], [FLS05], [RW06], [Kle07], [FSK08], [HHLB11], [SLA12]. The principle is illustrated on Figure 1.1. Clearly, the initial surrogate model is only a raw approximation and is not suitable for optimizing directly on it. With Gaussian processes especially, the prediction of the outputs comes along with an estimation of the corresponding uncertainty. It enables the definition of statistical criteria providing a balance between exploration and exploitation of the research domain. The right figure shows the resulting surrogate model obtained after adding sequentially new observations with this point of view: it is much more precise in regions of interest, in particular with new observations near the three global optima.

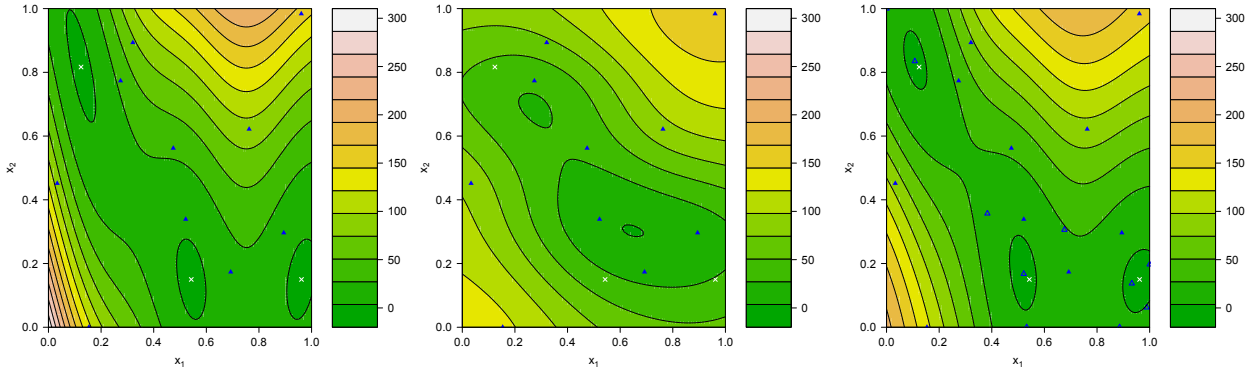


Figure 1.1 – Illustration of surrogate modeling. Left: output of an expensive function, known at 10 observations (blue filled triangles) with its global minima (white crosses). Center: approximation of the true function by a surrogate based on the initial observations. Right: approximation after adding sequentially ten new observations (empty blue triangles).

In its simplest form, an optimization problem is to find the minimum (say) of a function over a research space. Real life problems are in general more complex; for instance when constraints such as manufacturing restrictions are considered. Also, designing devices typically includes several possibly conflicting objectives. Specific techniques are used to handle these two cases, often by extending single objective unconstrained methods. We consider mostly two or three objectives at once, up to five in an application example. Taking more than three objectives into account ruins most of the attractiveness enabled by visualization of the compromise solutions and dedicated methods are in the "many-objective optimization" scope.

Another difficulty, that we discard here, is the case of noisy observations. When relevant, the possible extensions to cope with this situation will be mentioned. Let us remark that numerical simulation, if not submitted to observation noise as with real life experiments, may not be guarded from other sources of noise. Indeed, if running twice exactly the same calculation returns the same results in general, with codes based on Monte Carlo methods or finite elements, the output depends on the number of runs and the size or structure of meshes. Also, some physical phenomena such as crash-worthiness are intrinsically unstable, especially in high speed tests, where a piece can break differently depending on numerical noise (due e.g. to the number of cores used or the architecture of high performance computers).

Motivated in addition by test cases at Renault, the contributions of this thesis focus on both multi-objective optimization and high-dimensional research spaces. The corresponding results are built upon stochastic processes, Bayesian optimization, random closed sets and copulas. The structure of this document reflects these contributions with four parts:

- Part I introduces the general context and scope of the memoir, starting with this

Chapter 1. Chapter 2 briefly details surrogate-based optimization of an expensive black-box with Gaussian processes. It then focuses on Expected Improvement and its generalization to multi-objective optimization. In order to keep things readable in a reasonable number of pages, we have tried to let recalls at their strict minimum and sometimes voluntarily omit some topics when no contributions have been made. Also notions that are not needed throughout the whole thesis are detailed in due course.

- Part II contains two articles on uncertainty quantification on Pareto fronts from two different perspectives: using Gaussian processes with conditional simulations in Chapter 3 and with copulas in Chapter 4. While most methods focus on providing discrete approximations of the Pareto front, the goal is here to construct a continuous representations of this set of optimal solutions. This problem is cast either as considering a random closed set or as estimating extreme level lines of a multivariate distribution function.

- Part III summarizes contributions to overcome the challenge of optimizing in high dimensional research space with limited budgets. Backed up by empirical evidence, a hypothesis on a low number of unknown influential variable is made and the problem is tackled with random embeddings following the work of [WZH+13]. Chapter 5 starts with a description of the method, before proposing a covariance kernel that alleviates some of the previously associated shortcomings. Chapter 6 is concerned with selecting bounds for the low dimensional domain in the Random Embedding Bayesian Optimization (REMBO) method. In particular, some modifications of the random embedding are proposed, along with strategies for optimizing the infill criterion and an extension to multi-objective optimization. Combined together, the proposed modifications are shown to significantly improve the performances.

- Part IV tackles the implementation and applicative side of the two previous parts. In Chapter 7 is a description of the *GPareto* package that has been released on CRAN during this work. Chapter 8 is dedicated to an industrial test case in car crash-worthiness which has been used during this thesis to challenge the various contributions.

Three articles are integrated in Chapters 3, 4 and 5 respectively:

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

- M. Binois, D. Rullière, O. Roustant. On the estimation of Pareto fronts from the point of view of copula theory, *Information Sciences*, vol. 324, pp. 270-285 (2015).

- M. Binois, D. Ginsbourger, O. Roustant. A warped kernel improving robustness in Bayesian optimization via random embeddings, *Proceedings of the International Conference on Learning and Intelligent Optimization, LCNS*, vol. 8994, pp. 281 - 286 (2015).

The documentation of the *GPareto* package is available on CRAN:

- M. Binois, V. Picheny. GPareto: Gaussian Processes for Pareto Front Estimation and Optimization, R package version 1.0.1 (2015).

In the appendices are some additional contributions or promising ongoing works, including, but not restricted to, a fast approximation of the multipoint Expected Improvement, a Stepwise Uncertainty Reduction criterion and an interactive optimization procedure following Chapter 3, and complements on REMBO.

The computational experiments were performed on a PC with a quad core 2.80GHz processor and 32GB of RAM, a laptop with a dual core 2.9GHz processor and 16GB of RAM, or similar.

Chapter 2

Basics in Bayesian mono- and multi-objective optimization

Along this thesis, we consider an expensive-to-evaluate objective function $f : \mathrm{E} \subset \mathbb{R}^d \to S$ with $S = \mathbb{R}$ in the mono-objective case or $S = \mathbb{R}^m$ in the multi-objective case. The considered phenomena are complex, such that in general very little or nothing is known about their mathematical properties, that is why we treat them as black-boxes. We are interested in (say) minimizing f, possibly under constraints. E is the variable space, also called design, decision or parameter space and S is the objective space. We first describe this problem from the point of view of deterministic computer experiments and provide a brief background on Gaussian processes before detailing the mono-objective infill criteria built upon them, especially the Expected Improvement. Finally, multi-objective optimization concepts are exposed along with the corresponding extensions of the Expected Improvement.

2.1 Context and motivations

Expensive experiments or numerical simulations are commonly used in many fields, as for instance in the automotive and aeronautical industries or in nuclear safety. Indeed, with increasing computational power and precise physical modeling, it is often simpler, or just sometimes the only option, to perform numerical experiments instead of real ones. In concrete terms, crashing a prototype of a car costs from dozens to several hundreds of thousands of euros. As a consequence, performing it virtually seems much more affordable. It also offers insight on the results with step by step visualization of deformations and stresses, which are useful to correct or improve the structure. Note that for now, real experiments are still required for legal assessment of new vehicles.

Even if computer power has known a spectacular increase over the years, emphasis has

been put on precision such that the computational time remains long, several hours or days not being uncommon. In particular, reducing the size of elements in the finite element method allows predicting some physical phenomena that cannot be addressed at higher scale. The maximum number of simulations available is also limited by schedule constraints, especially in competitive sectors like the automotive one, where lead times of new vehicles are shrinking. On top of that, specifications on possibly antagonistic objectives such as pollutant emissions or safety are getting stricter and stricter, necessitating to work on broader perimeters, thus with more variables, to get sufficient improvement over the current designs.

To deal with this situation, approximate models have been used at least since [BW51] to select next evaluation points, with a variety of surrogate models ranging from polynomial regression to random forest, along with Support Vector Regression, neural network, wavelets or PolyMARS models. A review of their use can be found e.g. in [SPKA01], [WS07] or [FSK08]. We specifically focus on Gaussian process modeling, a probabilistic model that offers several key advantages over deterministic models such as the possibility of incorporating prior information from experts or a quantification of the modeling error (see e.g. [Gin09] for a discussion on this topic).

2.2 Gaussian Process Modeling (Kriging)

Due to a number of desirable features including their tractability and interpretability, Gaussian processes come into play in a variety of contexts, notably in spatial statistics, function approximation through splines and machine learning, see e.g. [Ste99], [Wah90] and [RW06]. Particularly, for the link between regularization or interpolation in Reproducing Kernel Hilbert Spaces and Gaussian Processes, we refer the reader to [Aro50], [KW70], [BTA04], [RW06]. For the sake of brevity, we only recall here the results we build upon in this thesis. More detailed introductions in a similar context may be found for instance in the recent thesis [Che13], [Bac13b] or [LG13].

2.2.1 Gaussian processes

A random process Y defined over a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and indexed by the parameter space E is said Gaussian if, and only if, for any finite set $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathrm{E}^n$, $n \in \mathbb{N}^*$, $Y(\mathbf{x}_{1:n}) = (Y(\mathbf{x}_1), \dots, Y(\mathbf{x}_n))$ is a multivariate Gaussian random variable. Gaussian processes are fully characterized (in distribution) by their mean $m(\cdot)$ and covariance (also called kernel) $k(\cdot, \cdot)$ functions:

$$m(\mathbf{x}) := \mathbb{E}\left(Y(\mathbf{x})\right), \mathbf{x} \in \mathrm{E} \tag{2.1}$$

$$k(\mathbf{x}, \mathbf{x}') := Cov\left(Y(\mathbf{x}), Y(\mathbf{x}')\right), (\mathbf{x}, \mathbf{x}') \in \mathrm{E}^2 \tag{2.2}$$

and as such define prior distributions over functions (see e.g. [CHS81], [RW06]).

In particular, $Y(\mathbf{x}_{1:n}) \sim \mathcal{N}(m(\mathbf{x}_{1:n}), k(\mathbf{x}_{1:n}, \mathbf{x}_{1:n}))$, where $k(\mathbf{x}_{1:n}, \mathbf{x}_{1:n})$ is the matrix of $k(\mathbf{x}_i, \mathbf{x}_j)$, $1 \leq i, j \leq n$, that we write in short $Y \sim \mathcal{GP}(m, k)$. The key advantage of GPs is that for any new point $\mathbf{x} \in \mathrm{E}$, the joint distribution of $Y(\mathbf{x}_{1:n})$, $Y(\mathbf{x})$ is simply given by:

$$\begin{pmatrix} Y(\mathbf{x}_{1:n}) \\ Y(\mathbf{x}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\mathbf{x}_{1:n}) \\ m(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}_{1:n}, \mathbf{x}_{1:n}) & k(\mathbf{x}_{1:n}, \mathbf{x}) \\ k(\mathbf{x}, \mathbf{x}_{1:n}) & k(\mathbf{x}, \mathbf{x}) \end{pmatrix}\right) \tag{2.3}$$

which provides tractable analytical expressions for marginal and conditional distributions. In particular, conditioning the GP on some observations results in another Gaussian process. Indeed, given the event $\mathcal{A}_n : \{Y(\mathbf{x}_{1:n}) = \mathbf{y}_{1:n}\}$, we have $\mathcal{L}(Y|\mathcal{A}_n) = \mathcal{GP}(m_n, k_n)$ with:

$$m_n(\mathbf{x}) = m(\mathbf{x}) + \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1}(\mathbf{y}_{1:n} - m(\mathbf{x}_{1:n})) \tag{2.4}$$

$$k_n(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}') \tag{2.5}$$

where $\mathbf{k}(\mathbf{x}) := (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^T$ and $\mathbf{K} := (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n}$ are the vector of covariances of $Y(\mathbf{x})$ with the $Y(\mathbf{x}_i)$'s and the covariance matrix of $Y(\mathbf{x}_{1:n})$, respectively.

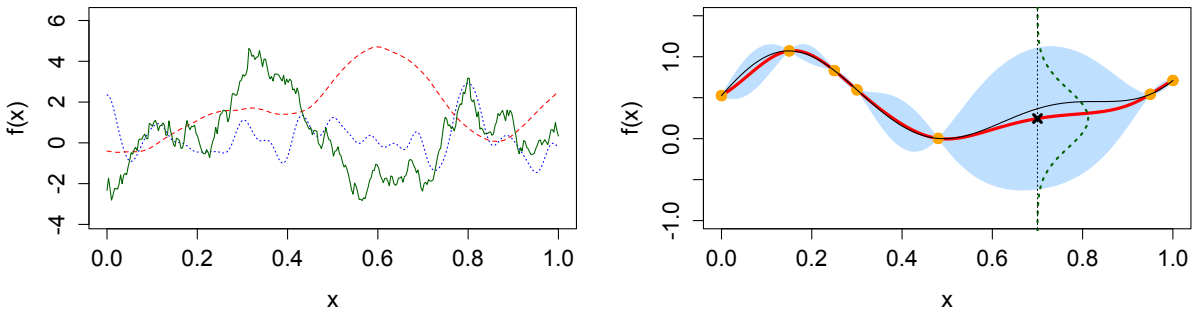
The choice of the covariance and mean function is of utmost importance as they dictates the properties of the corresponding GP, see e.g. [BTA04]. To be valid, a covariance function, $k : \mathrm{E} \times \mathrm{E} \to \mathbb{R}$ must be positive definite, i.e. if and only if for any $q \in \mathbb{N}, \mathbf{x}_1, \ldots, \mathbf{x}_q \in \mathrm{E}, \mathbf{a} \in \mathbb{R}^q$, $\sum_{i=1}^{q} \sum_{j=1}^{q} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$. Positive definiteness is a rather restrictive condition, but it is possible to build new kernels by combining several ones, e.g. by summing, multiplying, composition (warping) or convoluting, see e.g. [Abr97], [RW06]. A stationary kernel–when $k(\mathbf{x}, \mathbf{x}')$ is a function of $(\mathbf{x} - \mathbf{x}')$–can be written as the Fourier transform of a positive finite measure, see [Ste99], [RW06], providing a flexible tool to model most stationary kernels from a spectral density as instantiated in [WA13].

Sample paths (realizations) of the GP may be carried out at locations $\{\mathbf{e}_1, \ldots, \mathbf{e}_p\} \in \mathrm{E}^p$ from a variety of techniques. Perhaps the simplest approach is to cast the simulation of $Y(\mathbf{e}_1), \ldots, Y(\mathbf{e}_p)$ as a standard Gaussian vector simulation problem, thus relying on matrix decomposition approaches, such as the Cholesky decomposition. Unfortunately, when p in-

creases, the effort required to decompose the covariance matrix can become very cumbersome. Alternative methods to generate sample paths include e.g. turning bands, spectral methods, circulant embedding or Gibbs sampling [Jou74], [Cre93], [DR07].

Classical results about continuity and differentiability of sample paths of centered GPs with stationary or non-stationary kernels may be found in [CL67], [RW06], [Sch09] and references therein. Results in terms of invariance with respect to a linear operator are exposed in [GRD13], allowing to integrate structural priors such a zero mean property, harmonicity or symmetries. As a last examples of the efforts put in tuning and learning kernels, we refer to [Duv14] for a language to build kernels, to [GOR10] for applications in sensor networks by defining metrics over sets, to [CL12] for stationary GPs on hyperbolic spaces and Euclidean spheres, and to [Esp11] for GPs indexed by graphs.

To illustrate the above discussion, Figure 2.1a) presents several samples from GPs with various mean and covariance functions, highlighting the variety of possible prior encodings.



(a) (b)

Figure 2.1 – Left: three simulated sample paths of GPs with different mean and covariance functions, quadratic trend with Matérn 5/2 kernel (red dashed line), constant trend with exponential kernel (green solid line) and constant trend with a periodic Gaussian kernel (blue dotted line). Right: Gaussian process prediction (red line), with 95% prediction intervals (in blue) based on seven observations (orange dots), using a constant trend with a Matérn 5/2 kernel. The Gaussian distribution of the prediction at $x = 0.7$ is added in dashed, while the true underlying function is in black.

2.2.2 Predicting with Gaussian processes

The principle of Gaussian Process modeling, also known as Kriging, is to suppose that the considered objective function f is a sample path of a random field Y. In practice, taking a zero or fixed mean function is not the preferred solution as it is possible to incorporate some

basis functions. We suppose that the unknown mean function is of the form:

$$m(\mathbf{x}) = \sum_{i=1}^{l} \beta_i h_i(\mathbf{x}) = \mathbf{h}^T(\cdot)\boldsymbol{\beta} \tag{2.6}$$

where $\beta_1, \dots \beta_l \in \mathbb{R}$ are unknown coefficients and $h_1(\cdot), \dots, h_l(\cdot)$ are given basis functions. Supposing that the covariance kernel is known and putting an improper uniform prior on $\boldsymbol{\beta}$, the posterior distribution is $Y|\mathcal{A}_n \sim \mathcal{GP}(m_n(\cdot), k_n(\cdot,\cdot))$, with (see [OK78], [HS93]):

$$m_n(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T\hat{\boldsymbol{\beta}} + k_n(\mathbf{x})^T\mathbf{K}^{-1}\left(\mathbf{y}_{1:n} - \mathbb{H}_n\hat{\boldsymbol{\beta}}\right) \tag{2.7}$$

$$\begin{aligned} c_n(\mathbf{x},\mathbf{x}') = {} & k(\mathbf{x},\mathbf{x}') - k_n(\mathbf{x})^T\mathbf{K}^{-1}k_n(\mathbf{x}') \\ & + \left(\mathbf{h}(\mathbf{x})^T - k_n(\mathbf{x})^T\mathbf{K}^{-1}\mathbb{H}_n\right)\left(\mathbb{H}_n^T\mathbf{K}^{-1}\mathbb{H}_n\right)^{-1}\left(\mathbf{h}(\mathbf{x}')^T - k_n(\mathbf{x}')^T\mathbf{K}^{-1}\mathbb{H}_n\right)^T \end{aligned} \tag{2.8}$$

where $\mathbb{H}_n = \left(\mathbf{h}(\mathbf{x}_1)^T, \dots, \mathbf{h}(\mathbf{x}_n)^T\right)^T$ and $\hat{\boldsymbol{\beta}} = \left(\mathbb{H}_n^T\mathbf{K}_n^{-1}\mathbb{H}_n\right)^{-1}\mathbb{H}_n^T\mathbf{K}_n^{-1}\mathbf{y}_{1:n}$. Let us denote in addition $s_n^2(\mathbf{x}) = c_n(\mathbf{x},\mathbf{x})$ the prediction variance.

See e.g. [Cre93], [Mat69] for the equivalent derivation of the so called *Universal* Kriging formulas, corresponding to $Y(\cdot) = \mathbf{h}^T(\cdot)\boldsymbol{\beta} + Z(\cdot)$ with Z a zero mean GP. *Ordinary* and *Simple* Kriging are special cases, with one constant and no basis function respectively. There exist results and discussions about the choice of the basis function, see e.g. [JR89] or [MS05]. Other models for the mean may be found e.g. in [VWF05], [Meh15]. In the rest we will refer to Kriging or Gaussian process models interchangeably since they basically perform the same task in our context. A small example of Ordinary Kriging on the `fundet` function from the *KrigInv* R package [CPG14] is provided in Figure 2.1b, showing the ability of a GP to accurately learn from observations.

Also the hypothesis of a known covariance function is unrealistic in most configurations. In general, k is supposed to belong to parametric families of covariance functions such as the stationary *Gaussian*[1] and *Matérn* kernels, based on hypothesis about the smoothness of the underlying black-box function: infinitely differentiable in the first case, twice, once or only continuous in the second case with regularity parameter $\nu = 5/2,\ 3/2$ and $1/2$ respectively. Expressions of these kernels may be found e.g. in [RW06]. When the input dimension is greater than one, options presented e.g. in [Abr97], [RW06], include assuming isotropy– k as a function of $\|\mathbf{x} - \mathbf{x}'\|$–, considering that k is a function of $\sqrt{(\mathbf{x}-\mathbf{x}')^T\mathbf{R}(\mathbf{x}-\mathbf{x}')}$ with $\mathbf{R}$ a positive (often diagonal) semi-definite matrix (with coefficients as additional parame-

[1] a.k.a. squared exponential kernel, radial basis function kernel or exponentiated quadratic kernel [Duv14], [DL13]

ters to estimate) or a separable covariance model as in the *DiceKriging* package [RGD12]: $k = \prod_{i=1}^{d} k_i(x_i, x_i')$, with k_i one dimensional kernels. Other options like additive models will be detailed in Chapter 8 while kernels that account for ANOVA decomposition offer promising perspectives for modelling and learning sparsity in high-dimension [GBC+14].

Next the parameters are estimated e.g. from maximum likelihood or cross validation and the obtained covariance kernel is plugged in Equations 2.7 and 2.8. The main drawback of this approach is the underestimation of the posterior variance since the uncertainty on the covariance parameters is discarded. Note that, as presented e.g. in [RW06], it is also possible to use Expectation maximization, cross validation, variational estimation or a fully Bayesian approach, but then the predictive distribution has no more closed form expression, thus requiring the use of more computationally demanding techniques based e.g. on Markov Chain Monte-Carlo. In [Meh15], the variance of the predictor with plug-in is compared to variance estimated by conditional simulation and bootstrap. In the following, and unless stated otherwise, we will use maximum likelihood estimation within the plug-in approach. Nevertheless, the methods presented in this thesis would still apply if integrating the uncertainty on the covariance parameters in a *full Bayesian* setting, and presumably with a better performance.

2.3 Mono-objective infill criteria

Initially, as discussed e.g. in [Jon01], one strategy in surrogate-based methods consists in replacing the true function by a cheap-to-compute approximation like a polynomial one, to search the design space inexpensively, find its optimum and evaluate it. Then possibly perform the same task again. This generally performs poorly since the surrogate is usually quite coarse which few points. This puts too much emphasis on exploitation of the surrogate model without taking into account the uncertainty about its predictions. Moreover, optimization does not actually require a good model everywhere, as for instance in [GCLD09]. That is why more elaborated techniques towards optimization have emerged, such as the Efficient Global Optimization (EGO) algorithm [JSW98] which popularized the Expected Improvement [MTZ78], an infill criterion building on the error of the prediction offered by GP models. For more details about alternative methods for global optimization with surrogate models and a discussion about their respective flaws, the interested reader is referred e.g. to [Jon01] and [Vil08].

2.3.1 Bayesian optimization procedure

Bayesian optimization, initiated with works on the Expected Improvement [MTZ78], is built on two pillars: the first one is to consider the underlying black-box function as random and

to put a prior distribution that reflects beliefs about it. New observations then refine the posterior distribution. The second pillar is an acquisition function that selects new point locations using the posterior distribution to balance exploitation of promising areas and exploration of less known regions.

Algorithm 1 gives the general outline of Bayesian optimization algorithms. One such example is the now famous EGO algorithm [JSW98]. Its prior distribution is a Gaussian process with anisotropic generalized exponential covariance kernel, and a constant mean. The initial design of experiments is performed with Latin hypercubes, and then the model is validated before performing additional experiments with the Expected Improvement as acquisition function.

Algorithm 1 Sketch of a typical Bayesian optimization procedure

1: Perform an initial design of experiments
2: Train the Gaussian process model
3: Optimize the acquisition function
4: Evaluate f at the corresponding design
5: **if** Stopping criterion met **then**
6: Stop
7: **else**
8: Go to step 2
9: **end if**

The choice of an optimal initial design of experiments with GP modeling is a research domain in itself, thus in general we keep to the usual strategy of using a space filling design, including but not limited to maximin Latin Hypercube Samples, see e.g. [FK09], [DHF15] or [PM12] for a discussion. Some authors advocate skipping the construction of the design of experiments and directly start the sequential procedure, see e.g. [WZH+13]. Next we detail the crucial choice of the acquisition function.

2.3.2 Expected Improvement and other infill criteria

A key aspect for infill criteria is to balance between exploitation of promising areas and exploration of unknown ones. Particularly, optimizing directly on the prediction mean given by Kriging is known to be local, while focusing on the variance of the prediction only is too exploratory. These flaws are discussed e.g. in [Jon01], [GLRC10]. An alternative solution, initially proposed by [Kus64], is provided by the probability of improvement criterion: $PI(\mathbf{x}) = \mathbb{P}\left[Y(\mathbf{x}) \leq t_n | \mathcal{A}_n\right] = \Phi\left(\frac{t_n - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right)$ with $t_n = \min_{1 \leq i \leq n} f(\mathbf{x}_i)$. It has an analytical expression but is known to focus too much on exploitation since the magnitude of improvement is not taken into account, see e.g. [Jon01], [Liz08], [GLRC10]. A solution proposed in [Jon01] is

to use several thresholds instead of just t_n. To avoid this choice of thresholds and yet benefit from an analytical criterion, a simpler alternative is to use the *Expected Improvement* (EI) [MTZ78].

To better understand the roots of EI as well as the development of subsequent criteria, we take a small detour through decision theory, e.g. in the spirit of [MTZ78], [GLR10], [Che13]. Here we adopt the terminology of *optimality gap* (or *regret*) for the difference between the best response found so far and the true minimum: $t_n - f^*$, with $f^* = \min_{\mathbf{x}\in \mathrm{E}} f(\mathbf{x})$. When selecting $\mathbf{x}_{n+1}$, a natural aim is to minimize the future optimality gap $t_{n+1} - f^*$. Unfortunately, t_{n+1} is unknown beforehand and f^* is generally not known at all. But it can be shown that minimizing the expected regret is equivalent to maximizing $\mathbb{E}\left(\max(0, \left(\min_{1\leq i\leq n} Y(\mathbf{x}_i)\right) - Y(\mathbf{x}_{n+1})\right)$, which is known as the Expected Improvement criterion, see e.g. [JSW98] and references above. Indeed, in a minimization context, the improvement is defined as $I : \mathbb{R} \to \mathbb{R}^+$, $I(u; f_{min}) = \max(f_{min} - u, 0)$ where $f_{min} = t_n$ is the current minimum and u is typically the (unknown) response at $\mathbf{x}$. In general the notation f_{min} is dropped when there is no ambiguity. Integrating the improvement function I with respect to the distribution of $Y(\mathbf{x})|\mathcal{A}_n$ leads to a closed form when this posterior is Gaussian:

$$\mathbb{E}\left(I(Y(\mathbf{x}), f_{min})|\mathcal{A}_n\right) = \int_{-\infty}^{\infty} \frac{I(y, f_{min})}{s_n(\mathbf{x})}\phi\left(\frac{y - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) dy = \int_{-\infty}^{f_{min}} \frac{f_{min} - y}{s_n(\mathbf{x})}\phi\left(\frac{y - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) dy \tag{2.9}$$

$$= (f_{min} - m_n(\mathbf{x}))(\mathbf{x})\,\Phi\left(\frac{f_{min} - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) + s_n(\mathbf{x})\phi\left(\frac{f_{min} - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) \tag{2.10}$$

which additionally allows fast computation as well as gradient calculations, see e.g. [RGD12]. The Expected Improvement relies on both point-wise prediction mean and variance, resulting in a balance between exploitation of areas of low mean and exploration within areas of high variance, without any tuning parameters. A comparison between EI and PI on the Forrester function (see e.g. [FSK08]) is provided in Figure 2.2, where it can be seen that PI is indeed more local. To modify the exploration/exploitation trade-off, generalizations of EI have been proposed by taking exponents of I: $I^g(u, f_{min}) = max\left\{(f_{min} - u)^k, 0\right\}, k \in \mathbb{N}$ in [SWJ98], [PWBV08]. As for conditions for convergence, they are given in [VB10], [Bul11]. A similar infill criterion has been proposed in [OGR09], accounting for the expected loss instead of the improvement.

This point of view of considering only the next evaluation is known as a 1-step look-ahead (or myopic) strategy since it acts as if the next iteration were the last one. The Expected Im-

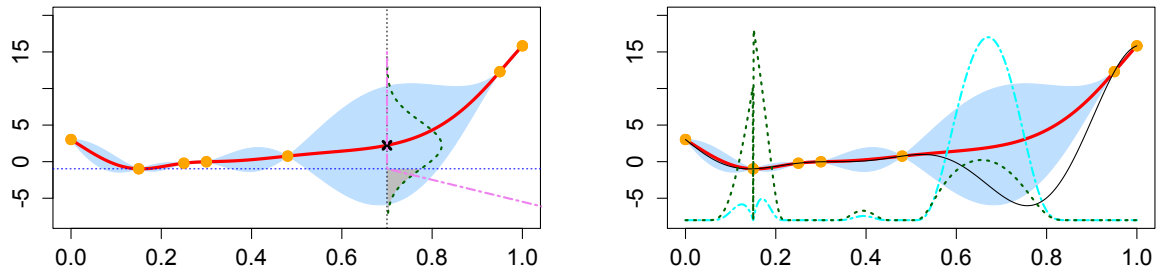


Figure 2.2 – Left: graphical interpretation of the Probability of Improvement and of the Expected Improvement. The probability of improvement at $x = 0.7$ is the integral of the Gaussian density (dashed green) below the threshold (dotted line), i.e. the gray part, while the Expected Improvement is the same density but integrated with respect to the improvement function (in dashed-dotted violet). Right: comparison of the values of the two criteria, dashed-dotted cyan for EI and green dashed for PI. Other elements are as in Figure 2.1.

provement is optimal in this case see e.g. [Che13]. A more efficient strategy would be to plan the next k points $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+k}$ in order to get the smallest regret after those k iteration. In particular, [GLR10] showed that sequential optimization of the Expected Improvement is sub-optimal. Optimal 2-step ahead strategies are tackled in one way or another in [Moc89], [OGR09], [GLR10] and [GAOST10]. As appealing as this may appear, their major drawback lies in computation: for a 3-step look-ahead strategy, the distribution of future location X_{n+3} depends on X_{n+2} which in turns depends on X_{n+1}. The optimal strategy involves random variables in a non-linear and highly intricated way, simulation is thus required for solving intermediate EI maximization problems. Hence the amount of sampling required is usually unaffordable and 1-step look-ahead strategies are used instead.

Variations on the Expected Improvement and recent development of EGO are numerous, and sometimes combined as in [HBdF11]. They include varying the metamodel as in [Meh15], fully Bayesian approaches [Osb10], [Ben13], or replacement of the observed minimum by an adaptive target [QPNV10], [CH14]. The Expected Improvement is concerned with minimizing the expected regret, but there also exists other criteria based for instance on Thomson sampling [CL11], [AG13], on mutual information [CPV14] or on entropy [VVW09], [HS12], [HLHG14] which illustrate the broader concept of uncertainty reduction. Given an uncertainty measure for a quantity of interest, e.g. the Shannon entropy of the position of the minimizer [VVW09] in the IAGO algorithm, the principle of Stepwise Uncertainty Reduction (SUR) strategies is to sequentially add new points which will reduce the most the expected uncertainty as e.g. in [BGL+12], [Che13], [Pic13]. These SUR strategies dedicated to optimization generally perform better than EI, see e.g. [VVW09] or [HS12], since the uncertainty

is a more global measure, but they tend to be much more difficult to compute or tune, which probably hinders their spreading among practitioners. Another popular acquisition function, which comes with theoretical guarantees, is the Gaussian Process Upper Confidence Bound (GP-UCB) criterion [SKSK10], [dFZS12] that writes $UCB_n(\mathbf{x}) = m_n(\mathbf{x}) + \beta_n^{1/2} s_n(\mathbf{x})$, with $\beta_n^{1/2}$ a tuning parameter, see e.g. [SKSK10] on how to determine it. Note that this latter adopts a *bandit* setting, i.e. considers the cumulative regret over iterations. It is mostly used in Machine Learning, motivated by applications. Here, in optimization, we do not care on the value of the regret over time as long as the last iteration is as close as possible to the optimum. Besides, in [Osb10], evaluation at designs for which the confidence of the model is high are not performed.

The last aspect we mention is about batches of new points instead of a single one: with the development of multi-core or grid architectures in modern computers, running several instances of the black-box code at once is more and more commonplace. Multipoint versions of the Expected Improvement have been proposed e.g. in [Sch97], [GLRC10], [CG13], [JLRGG12], [FC12]. Additional details are given in Appendix A, where we propose a very fast approximation of the exact formula. Parallel versions of GP-UCB may be found in [DKB12], [DKB14]. As for SUR criteria, they can usually be parallelized since it amounts to selecting batches of points inducing the highest decrease of uncertainty, the main challenge being to keep tractable criteria as in [CGB+14]. In Chapter 3, we only consider adding a single point at each iteration, but this could be extended to sequential batches directly. It is the same in Part III with contributions to adapt Bayesian optimization to high dimensional research spaces, where the Expected Improvement could be replaced by parallel criteria without additional changes.

2.3.3 Constrained infill criteria

A typical situation in optimization is to consider constraints. This is one side of the industrial test case of Chapter 8. They can correspond to given specifications or to incompatibility problems in the model used for computing some physical phenomenon. Denoting $\mathbf{g} : \mathrm{E} \to \mathbb{R}^r$, $\mathbf{g}(\mathbf{x}) = (g^1(\mathbf{x}), \ldots, g^r(\mathbf{x}))$ the vectorized output of constraints, the constrained optimization problem may be written: $\min_{\mathbf{x} \in \mathrm{E}} f(\mathbf{x})$ $s.t.$ $\mathbf{g}(\mathbf{x}) \leq \mathbf{T}$, where $\mathbf{T}$ is a vector of thresholds on the constraints. Equality constraints can always be divided in two inequality constraints.

Within EI-like criteria, taking into account expensive constraints has been done in a variety of ways, see e.g. [FK09], [Par12], [PKFH12], [GSA14] and references therein. One option is to use penalty methods with EGO, which consist in their simplest form to add a large constant to the objective function value whenever the constraints are violated. In this case, modeling

is usually quite difficult and it is best to fit a model per constraint, denoted G^i, $1 \leq i \leq r$. One such solution is to work with the probability of *feasibility* [SWJ98], i.e. that a new design is feasible, and expressed simply with GP models: $\mathbb{P}\left[G^i(\mathbf{x}) \leq T_i\right] = \Phi\left(\frac{T^i - m_n^i(\mathbf{x})}{s_n^i(\mathbf{x})}\right)$, with $m_n^i(\mathbf{x})$ and $s_n^i(\mathbf{x})$ the Kriging mean and standard deviation of G^i respectively. Coupled with the EI, one gets the Expected Feasible Improvement: $EFI(\mathbf{x}) = EI(\mathbf{x}) \prod_{i=1}^{r} \mathbb{P}\left[G^i(\mathbf{x}) \leq T_i\right]$, under the hypothesis that constraints and objective functions are independent. The main drawback is that EFI tends to add points quite far from the boundary of the admissible domain, see e.g. [Pic14] and references therein. If unfeasible points can still be evaluated, they may provide information as taken into account in [GGD+14]. Among other solutions are general formulation with subset estimation [FBV15], a hybrid method in [GGD+14] based on the augmented Lagrangian framework, or a dedicated SUR criterion aiming at reducing the expected volume of the admissible excursion set below the best known feasible point [Pic14] (requiring integration over the input domain of the updated probability of improvement times the probability of feasibility).

Even if using constraints may be fine, it also happens that they cannot be fulfilled simultaneously (empty feasible set) or that after some changes in the specifications, a new optimization process has to be started since there is not enough diversity in the obtained solutions. In this case, it is preferable to consider constraints as objectives instead, and apply multi-objective methods which possess several advantages. Last but not least, in some cases it is better to reformulate a mono-objective problem into a multi-objective one, a concept referred to as *multi-objectivization*, see e.g. [Deb08] and references therein. One example of application concerns the multipoint EI, where possible objectives for selection of batches are the prediction mean and variance, the value of EI, distance to neighbors or distance to known points [BWB+14].

2.4 Multi-objective optimization

Multi-Objective Optimization (MOO) appears in the frequent situation when one wants to optimize several objectives at the same time. Such problems have been studied in Economics, Game Theory and Engineering, resulting in a diversity of terms and concepts between those fields [MA04]. The reader interested by an historical review on the topic is referred to [Sta87]. The philosophy behind multi-objective optimization differs from the single objective one, which can already be observed with the more complex definition of a solution, based on the concept of trade-off. As we are working with generic black-box functions, methods working with linear or quadratic assumptions on the responses cannot be used. After a brief review of concepts, classical and evolutionary methods to solve these problems, we concentrate on

GP-based EI-like criteria developed in this field.

2.4.1 Preliminary concepts

Now f is an application of $\mathrm{E} \subset \mathbb{R}^d \to \mathrm{S} \subset \mathbb{R}^m$ and in this context, the focus is not only on the decision space E but also on the objective space $\mathrm{S} \subset \mathbb{R}^m$. A (naive) analogy with mono-objective optimization would lead to define a solution of a MO problem as a solution minimizing every objective at once. Unfortunately, there exists in general no such solution since objectives are very often conflicting (consider speed and cost in transportation as an example). This leads to consider compromises, and the standard definition (or concept) of solution is based on Pareto dominance:

Definition 2.4.1 (Pareto dominance)**.** *Given* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *two vectors of* E*:*

- $\boldsymbol{a} \preceq \boldsymbol{b}$ *(*$\boldsymbol{a}$ *weakly dominates* $\boldsymbol{b}$*) if and only if* $\forall i \in \{1, \dots, m\},\ f_i(\boldsymbol{a}) \leq f_i(\boldsymbol{b})$*;*
- $\boldsymbol{a} \prec \boldsymbol{b}$ *(*$\boldsymbol{a}$ *dominates* $\boldsymbol{b}$*) i.i.f.* $\boldsymbol{a} \preceq \boldsymbol{b}$ *and* $\exists i \in \{1, \dots, m\}$ *s.t.* $f_i(\boldsymbol{a}) < f_i(\boldsymbol{b})$*;*
- $\boldsymbol{a} \sim \boldsymbol{b}$ *(*$\boldsymbol{a}$ *is equivalent to* $\boldsymbol{b}$*) i.i.f.* $\boldsymbol{a} \not\prec \boldsymbol{b}$ *and* $\boldsymbol{b} \not\prec \boldsymbol{a}$*.*

The resulting notion of optimality is as follows:

Definition 2.4.2 (Pareto optimality)**.** *Given* $\boldsymbol{a} \in \mathrm{E}$*:*

- $\boldsymbol{a}$ *is Pareto optimal i.i.f.* $\nexists \boldsymbol{b} \in \mathrm{E}$ *s.t.* $\boldsymbol{a} \prec \boldsymbol{b}$*.*
  If $\boldsymbol{a}$ *is Pareto optimal,* $f(\boldsymbol{a})$ *is said Pareto efficient.*
- $\boldsymbol{a}$ *is said weakly Pareto optimal i.i.f.* $\nexists \boldsymbol{b} \in \mathrm{E}$ *s.t.* $\forall i \in \{1, \dots, m\},\ f_i(\boldsymbol{b}) < f_i(\boldsymbol{a})$*.*

These definitions along with relations derived from domination may be found e.g. in [Mie99], [CS03], [Ehr05]. Note that notations and terminology may differ between authors, as specified with further definitions in Chapter 4. All (possibly infinitely many) Pareto optimal solutions in E, also called non-dominated points, form the so-called Pareto set, and their image in the objective space is called Pareto front or Pareto frontier. It corresponds intuitively to the configuration where it is not possible to improve on one objective without deteriorating at least one other. When there is no ambiguity, we use both the terms optimal points and non-dominated points without further precision for points either in the input or output space. We also define two points of interest, the *ideal* and *nadir* vectors [Mie99], corresponding to the objective-wise minima and maxima respectively. They may be used to rescale the different objectives when the applied criterion is affected by their relative ranges. Figure 2.3 illustrates one example of Pareto set and Pareto front, which is not convex nor concave, and disconnected. As is sometimes added for visualization, the weakly dominated

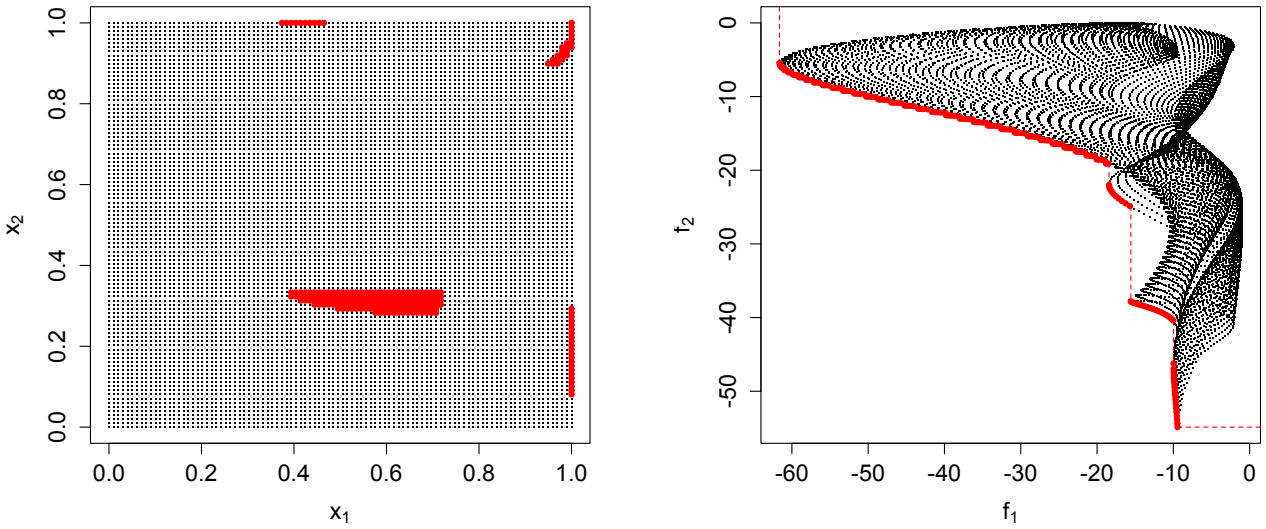


Figure 2.3 – Optimal points for the Poloni test problem [PGOP00] obtained on a 100×100 grid (black dots), with optimal points in red in the input space, i.e. the Pareto set (left) and in the objective space, i.e. the Pareto front (right). The red dotted line (right) along with the red points forms the weak Pareto front.

area by non-dominated objective points is marked with a step function. Note that if the Pareto front dimension is lower than the number of objectives minus one, for instance if the Pareto front of a bi-objective problem is a point, it is said to be *degenerate* [HHBW06].

The end product of an optimization study is commonly a unique solution, if possible on the unknown Pareto front. Depending on the knowledge about this Pareto front, one may directly orient the search toward a certain solution, or avoid intervening to get a greater range of alternatives. Briefly, this is the problem of selecting one solution out of the possibly infinite number of mathematically equivalent solutions, which is done based on preferences of the *decision maker* [Mie99], [MA04]. This problem is known in operation research literature under the terms Multiple-Criteria Decision Analysis and Multiple-Criteria Decision-Making. The stage at which this choice of a solution, a.k.a. articulation of preferences, is performed gives one possible classification of MOO methods:

- *a priori* methods: the relative importance of each objective is given before the optimization. A weighted aggregation of the objectives is then optimized to return a single solution of this scalarized problem.

- *a posteriori* methods: from a set of non-dominated solutions obtained without expressing preferences, decision makers select the one closest to their desiderata.

- *interactive* (or progressive) methods: optimization runs and preference articulation goes in turn to precise and adapt the needs and direct the search accordingly.

Due to the difficulty of anticipating the behavior in the objective space with *a priori* methods, it is more common in engineering to seek the whole set of possible solutions and then choose from it, i.e. using an *a posteriori* approach. We adopt this point of view in the following. For one proposed approach, we also describe how to use it interactively in Appendix B. Note that even with this setup, choices are sometimes left to the user to influence the search of new solutions, e.g. by choosing reference points.

2.4.2 Classical methods

Early techniques to tackle multi-objective problems convert them to mono-objective problems and apply the dedicated machinery to solve them (e.g. with Nelder-Mead, BFGS, CMA-ES, EGO, etc.). There are plenty of options to do so and we review here only the most common. Solving one such aggregated problem gives one solution on (or close to) the Pareto front. A Pareto front approximation is thereby obtained by varying the weights or search directions in the aggregation procedure.

The most intuitive solution to obtain a scalar function is with a weighted linear combination of the objectives, the problem is then to minimize $\sum_{i=1}^{m} w_i f_i(\mathbf{x})$ with $\sum_{i=1}^{m} w_i = 1$ and $\forall i \in \{1, \ldots, m\}\ w_i \geq 0$. It can be shown that the solution of the transformed problem is Pareto optimal [Mie99], under some regularity conditions. Nevertheless, there is a major drawback: only solutions on convex parts of the Pareto front may be found. This is not the case with the Tchebychev aggregation, where one would solve: $\min \left\{ \max_{i=1,\ldots,m} w_i |f_i(\mathbf{x}) - f_i^*| \right\}$ with weights as above and f_i^* the minimum of f_i. If the f_i^*'s, components of the ideal points are unknown, then the user must provide a point supposedly dominating it. In both cases, tuning the weights to obtain a specific solution on the Pareto front is in general arduous, as highlighted e.g. in [FSK08], [EU11].

Specifically designed methods for multi-objective optimization are *goal attainment* or *goal programming*, based on a target and a direction of search (see e.g. [Mie99], [MA04]). They consider minimizing the deviation from the target, resulting in a constrained mono-objective problem. Other constrained approaches include the ε-constraint method where the most important objective is optimized while putting constraints on values of other objectives. Lexicographic search is similar: objectives are optimized sequentially with constraints on previously optimized ones. More recent methods include for instance [EU11].

Equivalences between formulations depending on the distance or reference points used are discussed e.g. in [Mie99] and [TJR98]. The main advantages of these methods are the numerous optimality results associated, see e.g. [Mie99], [Ehr05] and the possibility of using

mono-objective optimizers with well-known properties.

Yet, there are several issues when using these methods. First the transformed problem may be quite difficult to solve, especially with non-linear aggregation. In addition, one need to incorporate *a priori* knowledge about the problem at hand, which is difficult with black-box functions. Even with precise preferences or target, the resulting solution might not be located where wanted on the Pareto front. This is especially true when running several instances with different weights, evenly spaced, which in general results in a poor coverage.

2.4.3 Multi-objective evolutionary algorithms

The field of Multi-Objective Evolutionary Algorithms (MOEA) has become quite popular from the 1990's on with many successes in dealing with complicated problems, see e.g. [CLVV07], [Deb08], [ZQL+11] or [Mon12] for a review. They often apply biomimetism and in particular take up the ideas of evolution and natural selection. They tend to be robust with respect to noise, discontinuities or non-differentiability, they are adaptable with the possibility to add constraints and extensions, and in addition it is possible to combine local and global search.

Some existing methods re-use the ideas of the classical methods described above with aggregation, or use several sub-populations accounting for the different objectives [CLVV07]. Perhaps the most popular method is the Nondominated Sorting Genetic Algorithm II (NSGA-II) [DPAM02], which often serves as a baseline for benchmarking new approaches. The main concepts applied are:

- *non-dominance sorting*: ranks of dominance are determined based on the values of the population in the objective space. Points of rank i are dominated only by points of rank $(i-1)$ or lower, and they dominate points of superior rank.
- *crowding distance*: a measure of the concentration of points in a given region of the objective space.
- *crowding comparison operator*: points are compared first on their rank, then on their crowding distance. They are assigned a fitness value accordingly.
- *archiving*: points with best fitness values are stored.
- *elitism*: only the best individuals are used to generate the next population.
- mechanisms to create a new population: *binary tournament, recombination*, *mutation operators*, *cross-over operators*.

Many alternatives exist to NSGA-II with variations on fitness attribution and generation of individuals, such as NPGA (Niched-Pareto Genetic Algorithm), MOGA (Multi-Objective Genetic Algorithm), or SPEA-II (Strength Pareto Evolutionary Algorithm). A more exhaustive description may be found e.g. in [CLVV07].

Before describing some more recent examples of MOEAs, let us precise what we mean when speaking of *quality* of approximation for a Pareto front. The result of an optimization is a set of non-dominated points, which the user wishes to be in some sense "close" to the true Pareto front. From this, as it involves comparing sets (on top of that, one of them being discrete while the other is generally not), judging the quality of the result or comparing it with the output of different optimizers is quite complex. Qualitatively, we think of an approximation as good when it offers a uniform covering of the true Pareto front, reflects the diversity of possible compromises and has as many points as possible (see e.g. [VK10] for details).

When the true Pareto front is unknown, a number of quality indicators may be used to compare approximations of Pareto fronts, a detailed survey can be found e.g. in [ZKT08]. Perhaps the most important one is the (unary) hypervolume indicator, which is the volume dominated by the approximation relatively to a reference point (the dominated area of the objective space being unbounded, defining volume would have no interest). For a pair of approximations, the (binary) hypervolume indicator is the volume dominated by the first one and not by the second. Besides this, a weighted function may be applied in the objective space to focus on specific parts. Another popular indicator is the additive epsilon indicator, corresponding to the smallest real number which must be added to the second approximation in order to be dominated[2] by the first. These two families of quality indicators are used to compare two sets of non-dominated points in Figure 2.4. Finally, they can also be applied to perform statistical tests when comparing stochastic algorithms. In this case the attainment function method can also be applied, consisting in computing the probability for a given point in the objective space to be dominated by a run of a method; these attainment functions are then confronted with dedicated statistical tests, see e.g. [ZTL+03], [ZKT08], [dFF10] and references therein.

Since features of interest for the Pareto front are evaluated with quality indicators, it makes sense to use them to guide the search. Examples are the S-metric selection - EMOA (SMS-EMOA) method [BNE07] with the hypervolume or the Indicator-Based Evolutionary Algorithm (IBEA) [ZK04] relying on the epsilon or hypervolume indicators. While being quite effective in practice, MOEAs also have a number of drawbacks, including the tuning of

[2]The set A is said dominated by the set B if for any point of A there is a point in B dominating it.

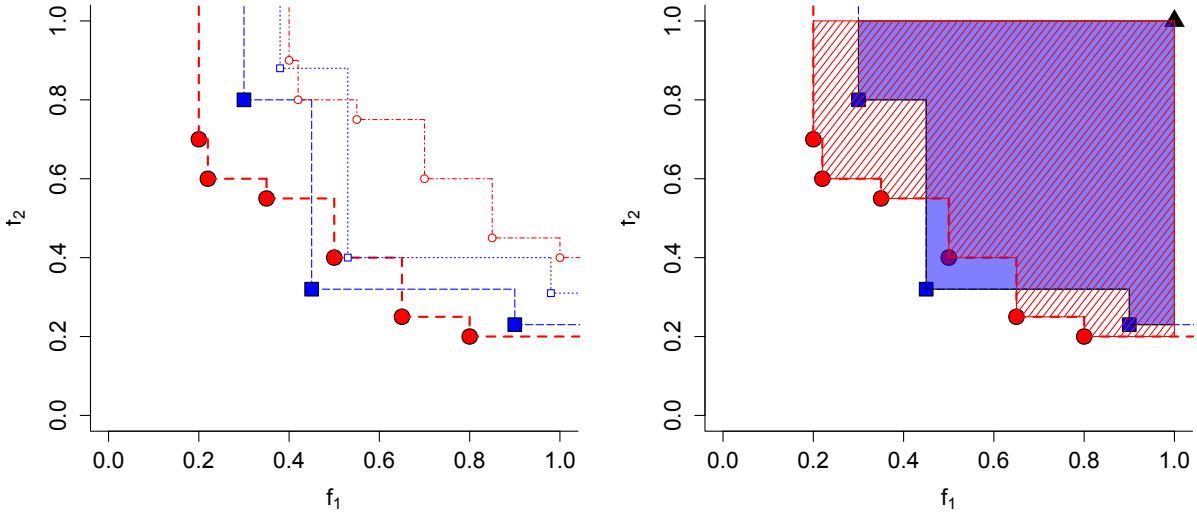


Figure 2.4 – Additive binary epsilon (left) and hypervolume (right) quality indicators for comparing two sets of non-dominated points, one with red points and the other with blue squares. The reference point for hypervolume computations is the black triangle.

parameters such as population size, number of iterations and probability of mutation. The number of runs is in addition frequently quite high, with thousands or much more of evaluations as budget. Finally, there is in general no known convergence properties, as opposed to some classical methods, explaining the growing development of hybrid combinations of global and local algorithms [Deb08].

2.4.4 Surrogate-based and Bayesian multi-objective optimization

When f is too expensive or time consuming, the previously described methods are in general not applicable directly. As with a single objective, the popular approach in this configuration is to rely on surrogate models. Many of them are used in practice: polynomials, splines, Support Vector Regression, Radial Basis Functions (RBF), random forests or GPs. They may be integrated in various strategies, see e.g. [WS07], [SQMC10], [THH+15] and references therein. Similarly, there are examples where the metamodel, after validation, simply replaces the true function in optimization, for instance with NSGA-II [VK10]. It is also possible to interleave updates of the model and optimization. Several models, possibly of different nature may also be used, see e.g. [Mon12]. Unfortunately, constructing a precise model everywhere is usually too expensive as soon as the input dimension increases, due to the curse of dimensionality, a cost which is this time multiplied by the number of objectives. Same causes leading to the same effects, a more efficient technique is, instead of taking the surrogate as a replacement of the true function, to take it as an aid to select the most

promising regions and sequentially add new points. This in fact leads to multi-objective extensions of the Expected Improvement with several objectives when working with GP models.

Before focusing more specifically on multi-objective variants of EI, let us remark that the idea of going back to a scalar problem to apply EGO has been exploited in several MOO methods. ParEGO [Kno06] applies EGO on a Tchebychev aggregation with randomly selected weights while MOEA/D-EGO [ZLTV10] builds several Tchebychev aggregations in parallel. Modeling a desirability function has also been proposed in [HK07]. However, problems related to aggregation are even more predominant. First, it causes a loss of information [HS08]. Then modeling one objective is already quite hard, hence modeling a non-linear combination of several is even trickier, due for instance to different scaling or characteristic length-scales. In addition, tuning the weights is again a severe difficulty, which results in more runs and this is typically what we want to avoid when dealing with limited budgets. We thus concentrate on truly multi-objective extensions of the EI, with sequential infill criteria that benefit from improvements directly expressed based on the current non-dominated points, as discussed e.g. in [Wag13]. Note that they may also be used as filters in evolutionary algorithms to select the most promising individuals, see e.g. [Emm05], [EGN06].

In Section 2.3.2, the Expected Improvement was emphasized for enabling a good balance between exploration and exploitation, from the magnitude of improvement regarding the best point so far. In a similar fashion, a multi-objective improvement function must be defined relatively to the non-dominated observations in lieu of the current minimum, with some desirable properties detailed e.g. in [WEDP10], [Sve11]. Hence a MO improvement logically becomes a function of $\mathbb{R}^m \to \mathbb{R}$ (dropping again the dependence to the observations). This leaves more room to tune the balance, with possibilities to put the focus either on a good coverage, on extremities, or on convergence toward the Pareto front. Multi-objective improvements are inspired by the mono-objective case and by metrics specific to MOO such as the hypervolume or epsilon indicators. We denote $\mathcal{P}_n$ and $\mathcal{R}_n$ the Pareto front approximation and the area dominated by the current n evaluations, respectively. Let the variable $\mathbf{u} \in \mathbb{R}^m$ denote where the improvement is calculated in the objective space, i.e. $\mathbf{u}$ accounts for the unknown objective function values at a new design $\mathbf{x} \in \mathrm{E}$, leading later on to multi-objective EI. Existing MO improvement functions of the literature include:

- *0-1 Improvement* [Sve11], [Par12]: $I_{\mathcal{PI}}(\mathbf{u}) = \mathbf{1}_{[\mathbf{u} \in \mathcal{R}_n]}$. It is a binary function, equal to 1 if the point is not dominated by $\mathcal{P}_n$, else 0. When integrated with the multivariate pdf of the posterior distribution, it gives the multi-objective probability of improvement;

- *Euclidean-based improvement* [Kea06], [Par12] also called *Keane's Distance Based Improvement* [Sve11]:

$I_{\mathcal{K}}(\mathbf{u}) = \begin{cases} \min\limits_{a_i \in \mathcal{P}_n} \|\mathbf{u} - a_i\|_2 & \text{if } \mathbf{u} \in \mathcal{R}_n \\ \text{else } 0 \end{cases}$.

If $\mathbf{u}$ is dominated by $\mathcal{P}_n$, the improvement is zero, else it is the minimal Euclidean distance between $\mathbf{u}$ and points in $\mathcal{P}_n$;

- *Hypervolume-based Improvement* [Sve11], [Par12]: denoting I_H the hypervolume indicator and R the reference point:
  $I_{\mathcal{H}}(\mathbf{u}) = \begin{cases} 0 \text{ if } \mathcal{P}_n \preceq \mathbf{u} \text{ or } R \preceq \mathbf{u} \\ I_H(\mathbf{u} \cup \mathcal{P}_n, R) - I_H(\mathcal{P}_n, R) \end{cases}$.
  Here the improvement is the contribution to the hypervolume of a new observation to the current Pareto front including the considered point. It is possible to apply a weighting function to promote certain areas, e.g. with the *Gaussian Weighted Hypervolume Improvement* $I_{\mathcal{WH}}$ [Sve11];

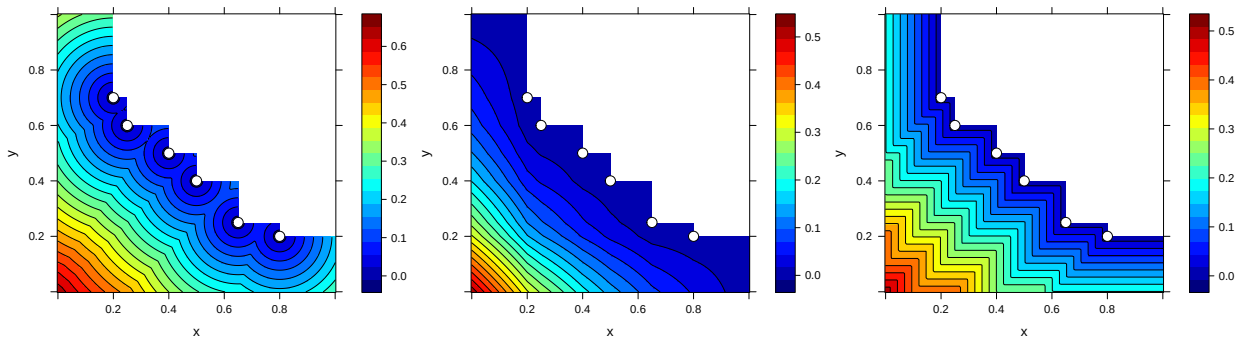
- *Pareto Improvement Function* [Bau09], [Sve11] : $I_{\mathcal{P}}(\mathbf{u}) = -\max\limits_{a^i \in \mathcal{P}_n} \min\limits_{j=1,\dots,m} (u_j - a_j^i)$.
  The Pareto Improvement (as well as the following Maximin Improvement) can be seen as a distance to the current Pareto front $\mathcal{P}_n$;

- *Maximin Improvement* [Sve11], [SS16]:
  $I_{\mathcal{M}}(\mathbf{u}) = -\max\limits_{a^i \in \mathcal{P}_n} \min\limits_{j=1,\dots,m} (u_j - a_j^i) \mathbf{1}_{\left[-\max\limits_{a^i \in \mathcal{P}_n} \min\limits_{j=1,\dots,m} (u_j - a_j^i) > 0\right]}$.
  This function is also the value of the *additive Binary-ϵ indicator* between the Pareto front obtained with $\mathcal{P}_n \cup \mathbf{u}$ and $\mathcal{P}_n$. It differs from the previous one with a zero improvement in the dominated area instead of negative values;

- *Completeness Indicator Improvement* [Sve11]: $I_{\mathcal{C}}(u) = I_{CP}(\mathcal{P}_n \cup u) - I_{CP}(\mathcal{P}^n)$ with I_{CP} the completeness indicator: $\mathbb{P}\left[\mathcal{P}_n \preceq f(\mathbf{U})\right]$ where $\mathbf{U}$ is a uniformly sampled point from the input space E.
  This indicator may be seen as the volume in the input space of points whose responses are dominated by $\mathcal{P}_n$. Further details on this indicator may be found e.g. in [ZKT08] or [Lot05].

The behavior of some of these improvement functions is illustrated in Figure 2.5. From the shape of their level lines, it can be observed that they provide different ways of balancing between adding points to extremities of the Pareto front, augmenting the current Pareto front or trying to dominate parts of it. Some undesirable properties can also be noticed, for instance with the Euclidean improvement, for which the dominance relationship is not respected. Indeed, a point dominated by another may still have a greater improvement. Other properties of interest for candidate improvement functions include absence of parameters, invariance with respect to rescaling of the objectives, and analytical expressions. Details, including summaries of the relative merits of these improvement functions (among others),

can be found e.g. in [Sve11], [WEDP10] and [Wag13].



(a) Euclidean improvement (b) Hypervolume improvement (c) Maximin improvement

Figure 2.5 – Contour plot of the values in the objective space of several improvement functions with respect to the observations (white points).

We can finally express the multi-objective expected improvement with arbitrary improvement function I^*. Note that, as stated e.g. in [WEDP10] and [Sve11], splitting the second term in Equation 2.9 yields $I(\bar{y}, f_{min})\Phi\left(\frac{f_{min}-m_n(\mathbf{x})}{s_n(\mathbf{x})}\right)$ with $\bar{y}$ the center of mass of the integral of the distribution of $Y(\mathbf{x})$ in the interval $]-\infty, f_{min}]$. These equivalent formulations when there is only one objective give rise to two different formulations when extended to multi-objective optimization, see e.g. [Sve11], [Wag13]:

- $QI(\mathbf{x}) = \mathbb{E}\left(I^*(\mathbf{Y}(\mathbf{x}))|\mathcal{A}_n\right)$, this is the natural extension of EI since this is an expectation;
- $QI(\mathbf{x}) = \mathbb{P}\left[\mathbf{Y}(\mathbf{x}) \in \mathcal{R}_n|\mathcal{A}_n\right] \times I^*(\bar{Y}(\mathbf{x}))$ with $\bar{Y}(\mathbf{x}) = \frac{\mathbb{E}\left(\mathbf{Y}(\mathbf{x})\mathbf{1}_{[\mathbf{Y}(\mathbf{x})\in\mathcal{R}_n]}|\mathcal{A}_n)\right)}{P(\mathbf{Y}(\mathbf{x})\in\mathcal{R}_n|\mathcal{A}_n)}$ the expectation of the distribution of $Y(\mathbf{x})$ conditioned to be under the Pareto front. This definition of QI is not an expectation, but it is used in practice since it avoids many computations of the improvement I^* when sampling is required and it allows computing some analytical expressions (see e.g. [Kea06]).

An extensive comparison in [Sve11] shows no clear superiority of one formulation over the other.

The multivariate cumulative distribution of the outputs plays a major role in these formulations. Notably, several options exist to introduce dependence between outputs within GP models, see e.g. the review of [ÁRL11]. The main difficulty is to obtain a properly defined cross-covariance structure between outputs while having an appropriate one for the inputs. In general, it is flexible only in one of these spaces and may require more hyperparameters. Due to these restrictions, integrating dependent models has not shown significant improvement

in practice, as stated in [Bau09], [Sve11], [KM14]. In addition, the derivation of analytical expressions depends on the structure of dependence used in the modeling: they have mostly been elicited when the multivariate pdf is the product of univariate ones (independent model), for two or three objectives (theoretically extensible but cumbersome for more objectives), see e.g. [Kea06], [SS16], [EDK11]. Recent faster algorithms to compute multivariate probability of improvement and expected hypervolume improvement may be found in [CDD13], [HDYE15]. Otherwise the computation is possible with Monte Carlo techniques from draws of the posterior distribution. In this case, the second interpretation of multi-objective EI is computationally more efficient when calculation of the improvement is costly, for instance with the hypervolume for many points and a lot of objectives. Apart from the probability of improvement which is not efficient, it was stated in [Sve11] that no approach nor improvement function presented a superior performance. Compared to simpler criteria such as using only the prediction mean or marginal EI over the different objectives, the Expected Hypervolume Improvement (EHI) has been shown to perform better in [SSJO12].

As in the mono-objective case, several alternatives to multi-objective EI can be found in the literature such as a variant of the EHI, i.e. SMS-EGO [PWBV08], [WEDP10], a SUR criterion for MOO proposed in [Pic13], an active learning algorithm with some convergence results [ZSKP13], which extends the GP-UCB method to classify solutions from a finite set of candidate as Pareto optimal, not Pareto optimal and undetermined ; another similar approach is proposed in [SHSMV14] that is only based on a Support Vector Machine classifier. In addition to the Expected Hypervolume and Maximim Improvement, we also implemented in *GPareto* [BP15] the SMS-EGO and SUR criterion, which we briefly detail.

The SMS-EGO infill criterion [PWBV08], [WEDP10] is basically the hypervolume added to the current Pareto front by the lower confidence bound of the prediction at $\mathbf{x}$, $\hat{\mathbf{y}} - \alpha\hat{\mathbf{s}}$ where $\hat{\mathbf{y}} = (m_n^{(1)}(\mathbf{x}), \ldots, m_n^{(m)}(\mathbf{x}))$, $\hat{\mathbf{s}} = (s_n^{(1)}(\mathbf{x}), \ldots, s_n^{(m)}(\mathbf{x}))$ and $\alpha(p) = -\Phi^{-1}(0.5\sqrt[m]{p})$, e.g. with $p = 0.5$. To account for possibly too optimistic lower confidence bounds and to favor good coverage, additive-epsilon dominance is considered, denoted $\preceq_\epsilon$, i.e. a vector $\mathbf{b}$ is said to be ε-dominated by $\mathbf{a}$ ($\mathbf{a} \preceq_\varepsilon \mathbf{b}$) if $a_i \leq b_i - \varepsilon$, $1 \leq i \leq m$. In case some solutions $\mathbf{y}^{(i)}$ in the current Pareto front are ϵ-dominating the lower prediction bound, the increment of hypervolume is replaced by the maximal penalty over these: $\max_{\mathbf{y}^{(i)} \text{ s.t.} \hat{\mathbf{y}} \preceq_\varepsilon \mathbf{y}^{(i)}} P(Y(\mathbf{x})) = -1 + \prod_{j=1}^{m} (1 + (m_n^{(j)} - y_j^{(i)}))$.

The SUR criterion of [Pic13] is in turn concerned with the probability of improvement. The uncertainty is the volume in the input space of the excursion set whose image in the objective space dominates the Pareto front. This is equal to the integral of the probability for a point $\mathbf{x}$ in the input space of not being dominated by any points in the current Pareto

optimal set $\mathbb{X}_n^*$, which is expressed by $ev_n(\mathbf{x}) = \int_{\mathrm{E}} \mathbb{P}\left[\mathbb{X}_n^* \npreceq \mathbf{x} | \mathcal{A}_n\right]$. The SUR criteria is then $J_n(\mathbf{x}) = \mathbb{E}\left(ev_{n+1}(\mathbf{x}) | \mathcal{A}_n\right)$. An analytical formula of the probability for a point of not being dominated at step $n+1$ is available but the integration over the input domain requires Monte-Carlo methods. Practical details about this are given in [Pic13] and references therein.

These sequential infill criteria are more or less similar to multi-objective EI and they share the common trait that they do not provide a continuous representation of the Pareto front but only consider improvement over the currently non-dominated points. This somehow missing point has been one of the lines of research followed within this PhD. The second limitation underlying in both mono and multi-objective optimization concerns the number of variables. When it increases, learning the surrogate model becomes impractical. It is also much longer: computations, for instance related to the inner optimization of the acquisition function, are also much more difficult. These restrictions are discussed in detail at the beginning of Chapter 5, before describing the Random EMbedding Bayesian Optimization paradigm, to which a few original contributions are presented. Some of the methods presented here have also been employed on a test case from Renault with 47 variables, as detailed in Chapter 8.

Part II

Uncertainty quantification on Pareto fronts

Chapter 3

Quantifying uncertainty on Pareto fronts with Gaussian processes

In Chapter 2, sequential infill criteria based on a Kriging surrogate have been presented. Here we consider GP metamodels not only for selecting new points, but as a tool for estimating the whole Pareto Front and quantifying how much uncertainty remains on it at any stage of Kriging-based multi-objective optimization algorithms. This chapter reproduces the article [BGR15a] that has been published in the European journal of Operational Research. Additional ongoing works realized after publication and complementing the article are briefly discussed at the very end of this chapter, while details are presented in Appendix B.

3.1 Introduction

The interest in Multi-Objective Optimization (MOO) has been growing over the last decades, resulting in the development of numerous dedicated methods, especially in evolutionary MOO [Deb08]. These methods are able to cope with challenging problems occurring when few information about the properties of the objective functions is available (black-box optimization). In some situations, as for example in car crash safety design [LLY+08], another difficulty comes from a limited budget of evaluations, because of expensive experiments or high fidelity simulations.

In this context, see e.g. [SQMC10] for a review, a common approach is to rely on a surrogate model or metamodel to alleviate the computational costs of the optimization process. In particular, Kriging metamodels have proven to be efficient because they not only give a response surface but also a quantification of prediction uncertainty. In mono-objective optimization, this property has been extensively used following the principles of the Efficient Global Optimization (EGO) algorithm [JSW98] where the Expected Improvement criterion

is used to balance between exploitation and exploration. Extensions to MOO have been developed, from scalarization approaches [Kno06, ZLTV10] to the use of multi-objective expected improvement criteria such as the Expected Hypervolume Improvement [EDK11].

While results about the optimality of solutions from aggregation approaches have been reported (see e.g. [Mie99]), things are more difficult to analyze for MOO and even worse in metamodel based MOO, where an additional source of uncertainty due to surrogate modeling must be taken into account. Besides, the study of the convergence in evolutionary MOO is an ongoing subject of research [WTM11].

Inspired by what has been proposed for Kriging-based excursion sets estimation in Chevalier et al. [Che13, CGBM13], we propose here to use notions from the theory of random sets [Mol05] for quantifying uncertainty on Pareto fronts, through conditional simulations. The latter are used to estimate the probability that any given point in the objective space is dominated, which is known in performance assessment of multi-objective optimizers as the attainment function [dFF10]. From this we obtain a metamodel-based estimation of the Pareto front using the Vorob'ev expectation [Mol05], with a value of the associated uncertainty: the Vorob'ev deviation. At each stage of the sequential optimization process, an insight is provided to the practitioner about convergence and possibilities of further improvements. Furthermore, with two or three objectives the proposed approach makes it possible to visualize the variability around the estimation of the Pareto front in the objective space.

The paper is organized as follows: Section 3.2 details notions in MOO and in Gaussian Process Regression upon which the proposed approach is based. In particular Section 3.2.4 is dedicated to conditional simulations. In Section 3.3, we propose an original definition of uncertainty using the Vorob'ev expectation and deviation. Finally, Section 3.4 is dedicated to applications of the proposed methodology to three different test cases, where the potential of the approach to quantify uncertainty and monitor convergence within a sequential MOO algorithm is illustrated.

3.2 Multi-objective optimization and Gaussian Process Regression

3.2.1 Notions in MOO

Multi-objective optimizers aim at optimizing (say minimizing) several objectives at once: $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ with $\mathbf{x} = (x_1, \ldots, x_d)^T$ a vector of decision variables in E (usually $\mathrm{E} \subset \mathbb{R}^d$)

and $f : \mathrm{E} \to \mathbb{R}^m$ the vector valued function whose coordinates are $f_1, \ldots, f_m$. As the objectives are usually in competition, the existence of an optimal solution minimizing all objectives simultaneously cannot generally be taken for granted. This leads to the definition of compromise solutions following the Pareto dominance: a vector is said to be *dominated* if there exists another vector which is not worse in any objective and better for at least one of them. If a vector is not dominated by any other vector, it is said to be optimal in the Pareto sense.

The set of optimal (or non-dominated) points in E is called *Pareto set* and the corresponding image by f, composed of non-dominated vectors, is called *Pareto front*. Multi-objective optimization algorithms aim at finding non-dominated objective vectors as close as possible to the true underlying Pareto front, creating a discrete approximation sometimes called a *Pareto front approximation* [ZKT08].

3.2.2 Kriging / Gaussian Process Regression

A common solution to perform optimization under a tight evaluation budget is to appeal to a mathematical surrogate of the objective function. Here we focus on a class of probabilistic metamodels relying on Gaussian random fields. Originating from geostatistics with a technique named *Kriging* [Mat63], predicting with such a metamodel is known in the machine learning community as *Gaussian Process Regression* (GPR) [RW06]. These Kriging/GPR metamodels have the property of interpolating the observations when noiseless data is considered (deterministic case). Furthermore, due to the probabilistic nature of these metamodels, they also provide a quantification of the prediction uncertainty.

Without loss of generality, here we do not assume a priori any stochastic dependency between the responses $f_1, \ldots, f_m$ and we treat them separately since the use of dependent models is significantly more cumbersome and has not been shown to perform better in state of the art studies [KM14, SS16]. Following the settings of Gaussian Process Regression [Ste99, SWMW89], each of the objective functions f_i is supposed to be a sample path of a random field Y_i:

$$Y_i(.) = \mathbf{g}_i^T(.)\boldsymbol{\beta}_i + Z_i(.) \qquad (\textit{Universal Kriging})$$

where $\mathbf{g}_i(.)^T$ is a vector of known basis functions, $\boldsymbol{\beta}_i$ a vector of unknown coefficient and $Z_i(.)$ is a zero mean Gaussian process (GP) with given covariance function, or kernel, k_i. With n evaluations at the same locations for the objectives $\{Y_i(\mathbf{x}_1) = y_{i,1}, \ldots, Y_i(\mathbf{x}_n) = y_{i,n}, 1 \leq i \leq m\}$ denoted $\mathcal{A}_n$, the predictor (or Kriging mean) and the prediction covariance (also referred to as Kriging covariance) of Universal Kriging (UK) are expressed as:

$$
\begin{aligned}
m_{i,n}(\mathbf{x}) &= \mathbf{g}_i(\mathbf{x})^T \hat{\boldsymbol{\beta}}_i + k_{i,n}(\mathbf{x})^T \mathbf{K}_{i,n}^{-1} \left(\mathbf{y}_{i,n} - \mathbb{G}_{i,n} \hat{\boldsymbol{\beta}}_i \right) \\
c_{i,n}(\mathbf{x}, \mathbf{x'}) &= k_i(\mathbf{x}, \mathbf{x'}) - k_{i,n}(\mathbf{x})^T \mathbf{K}_{i,n}^{-1} k_{i,n}(\mathbf{x'}) \\
&\quad + \left(\mathbf{g}_i(\mathbf{x})^T - k_{i,n}(\mathbf{x})^T \mathbf{K}_{i,n}^{-1} \mathbb{G}_{i,n} \right) \left(\mathbb{G}_{i,n}^T \mathbf{K}_{i,n}^{-1} \mathbb{G}_{i,n} \right)^{-1} \left(\mathbf{g}_i(\mathbf{x'})^T - k_{i,n}(\mathbf{x'})^T \mathbf{K}_{i,n}^{-1} \mathbb{G}_{i,n} \right)^T
\end{aligned}
$$

where $\mathbf{y}_{i,n} = (y_{i,1}, \ldots, y_{i,n})$, $\mathbf{K}_{i,n} = (k_i(\mathbf{x}_s, \mathbf{x}_t))_{1 \leq s,t \leq n}$, $k_{i,n}(\mathbf{x}) = (k_i(\mathbf{x}, \mathbf{x}_1), \ldots, k_i(\mathbf{x}, \mathbf{x}_n))^T$, $\mathbb{G}_{i,n} = \left(g_i(\mathbf{x}_1)^T, \ldots, g_i(\mathbf{x}_n)^T \right)^T$ and $\hat{\boldsymbol{\beta}}_i = \left(\mathbb{G}_{i,n}^T \mathbf{K}_{i,n}^{-1} \mathbb{G}_{i,n} \right)^{-1} \mathbb{G}_{i,n}^T \mathbf{K}_{i,n}^{-1} \mathbf{y}_{i,n}$.

Note that from a Bayesian point of view, assuming that the Y_i are Gaussian conditionally on $\boldsymbol{\beta}_i$ and putting an improper uniform prior on $\boldsymbol{\beta}_i$, it is known [HS93] that the Universal Kriging mean and covariance coincide with the conditional expectation and covariance of Y_i knowing $\mathcal{A}_n$, respectively: $m_{i,n}(\mathbf{x}) = \mathbb{E}(Y_i(\mathbf{x}) | \mathcal{A}_n)$ and $c_{i,n}(\mathbf{x}, \mathbf{x'}) = \mathrm{cov}\left(Y_i(\mathbf{x}), Y_i(\mathbf{x'}) | \mathcal{A}_n\right)$.

The covariance functions are chosen according to prior hypothesis about the unknown functions, such as regularity, sparsity, possible symmetries, etc. [GRD13]. While there exists a variety of admissible covariance functions, the most commonly used are the stationary "Gaussian" and "Matérn" kernels [Ste99]. Maximum likelihood estimation or cross validation techniques [Bac13a] are typically employed to estimate values for the kernel hyperparameters [RW06]. An example of a Kriging model with constant unknown trend and Matérn ($\nu = 5/2$) kernel is depicted in Figure 3.1a.

3.2.3 Multi-objective expected improvement

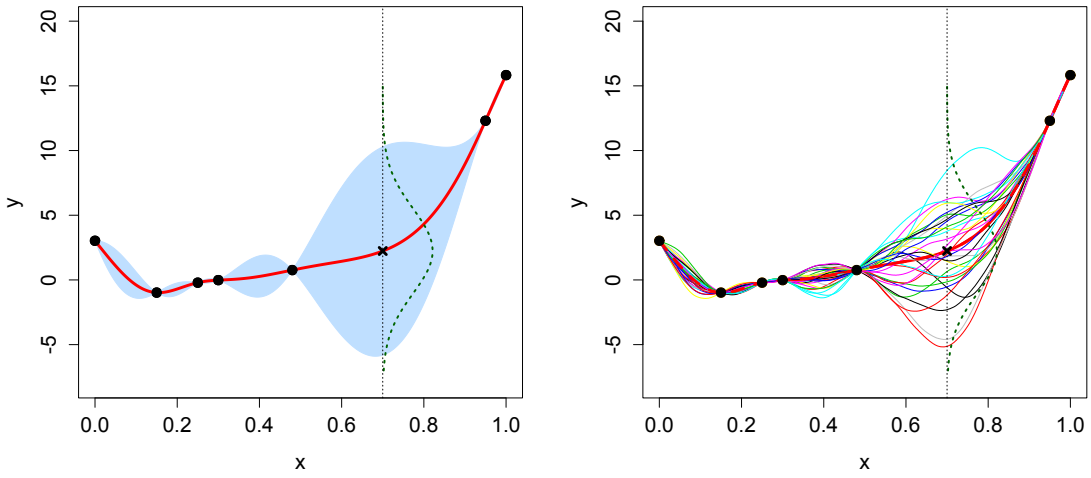
Sequential approaches in MOO aim at adding new observations with a balance between exploration and exploitation. Similar to [JSW98], several extensions of the EGO algorithm have been proposed for MOO. The main idea is to derive criteria in the vein of the *Expected Improvement* by defining a generalization of the notion of improvement for multiple objectives. Popular methods include scalarization approaches like ParEGO [Kno06] or MOEAD-EGO [ZLTV10] or truly multi-objective methods based on the definition of improvement functions over the current Pareto front $\mathcal{P}_n$ defined by the current observations. Considered improvement functions are respectively based on Euclidean distance [Kea06], Hypervolume [EGN06, WEDP10] or Maximin distance [Bau09, SS16] i.e. respectively the distance to the closest point of $\mathcal{P}_n$, the volume added over $\mathcal{P}_n$ and an axis-wise distance to $\mathcal{P}_n$.

In the applications of Section 3.4, we use the Expected Hypervolume Improvement to sequentially add points. This criterion has been successfully applied to problems with limited budget [EDK11], enjoys some beneficial properties [EDK11] and furthermore is related to the

concept of attainment function which is of particular importance in what follows.

3.2.4 Conditional simulations

From the Universal Kriging metamodels presented in Section 3.2.2, it is possible to generate samples interpolating the available evaluation results, called conditional simulations. They can be generated using a variety of methods, from matrix decomposition to spectral methods, as presented in [Jou74, Hos95, DR07, RGD12]. Examples of such conditional simulations are displayed in Figure 3.1b.



(a) (b)

Figure 3.1 – Left: example of Kriging model based on observations at $n = 7$ locations (black points), with Kriging mean (bold line) and Kriging pointwise 95% prediction intervals (shaded area). The Gaussian predictive distribution for $x = 0.7$ is represented by the vertical dashed line. Right: conditional simulations (colored thin lines) from the fitted UK metamodel.

They have been applied in mono-objective optimization in [VVW09] as a tool to estimate an information gain when no analytical formula is available, as opposed to the Expected Improvement. Until now, the computation of multi-objective Expected Improvement relies either on analytical formulas or on Monte Carlo estimation with draws of the posterior distribution at a single location $\mathbf{x}$. In contrast, conditional simulations consist in drawing posterior realizations of the unknown function at multiple points, say $\mathbf{E}_p : \{\mathbf{e}_1, \ldots, \mathbf{e}_p\} \subset \mathrm{E}$. Since exact[1] methods essentially depend on $p \times p$ covariance matrices, the number of simulation points is typically limited by storage and computational cost. Despite this limitation, conditional simulations (based on matrix decomposition) prove useful for Pareto front estimation, as presented in the next sections.

[1] with desired statistical properties, as opposed to approximate methods, see e.g. the discussion in [EL06].

3.3 Quantification of uncertainty

In this section, we assume that a Gaussian process model (see Section 3.2.2) has been estimated for each objective function from a set of n observations $\mathcal{A}_n$. These models allow us to generate conditional Pareto front realizations and further estimate the uncertainty on the Pareto front with concepts from random sets theory.

3.3.1 Conditional simulations for MOO: generation of conditional Pareto fronts and corresponding attained sets

Here we use conditional simulations to generate so-called *conditional Pareto fronts* (CPF). The first step is to simulate a finite number (say N) of vector-valued GP conditional simulations $\left\{\mathbf{Y}_1^{(1)}, \dots, \mathbf{Y}_m^{(1)}\right\}, \dots, \left\{\mathbf{Y}_1^{(N)}, \dots, \mathbf{Y}_m^{(N)}\right\}$ at some simulation points in the design space. Selecting the corresponding non-dominated simulation points and simulated responses then provides conditional Pareto sets and fronts as summarized in Algorithm 2 and illustrated in Figure 3.2.

Note that what we denote by CPF are actually approximations of conditional Pareto fronts, just like conditional simulations of Gaussian random fields are often approximated realizations relying on a finite number of points. Simulation points can be fixed for all the simulations or changed from one simulation to the other. Fixed simulation points accelerate the simulation generation but they may introduce a bias and lead to missing worthwhile areas. On the other hand, modifying the simulation points increases the computational burden but is more exploratory, which might be an asset in high dimensions. Besides, the procedure used to choose the location of simulation points may impact the results. Accordingly, two sampling strategies are investigated in Section 3.4.2.

Algorithm 2 Simulation of N conditional Pareto sets and fronts

for $i = 1, 2, \dots, N$ **do**
  Choose p simulation points $\mathbf{E}_p = \mathbf{e}_1, \dots, \mathbf{e}_p$ in $\mathbf{E}$ (fixed or different at each iteration).
  for $j = 1, 2, \dots, m$ **do**
    Generate a conditional simulation at $\mathbf{e}_1, \dots, \mathbf{e}_p$ for the j^{th} objective: $\mathbf{Y}_j^{(i)} = \left(Y_j^{(i)}(\mathbf{e}_1), \dots, Y_j^{(i)}(\mathbf{e}_p)\right)$.
  end for
  Determine the Pareto set and front of $\left\{\mathbf{Y}_1^{(i)}, \dots, \mathbf{Y}_m^{(i)}\right\}$.
end for

From now on we focus on the use of CPFs since the decision maker is mostly interested in visualizing results in the objective space. Each CPF is composed of non-dominated points

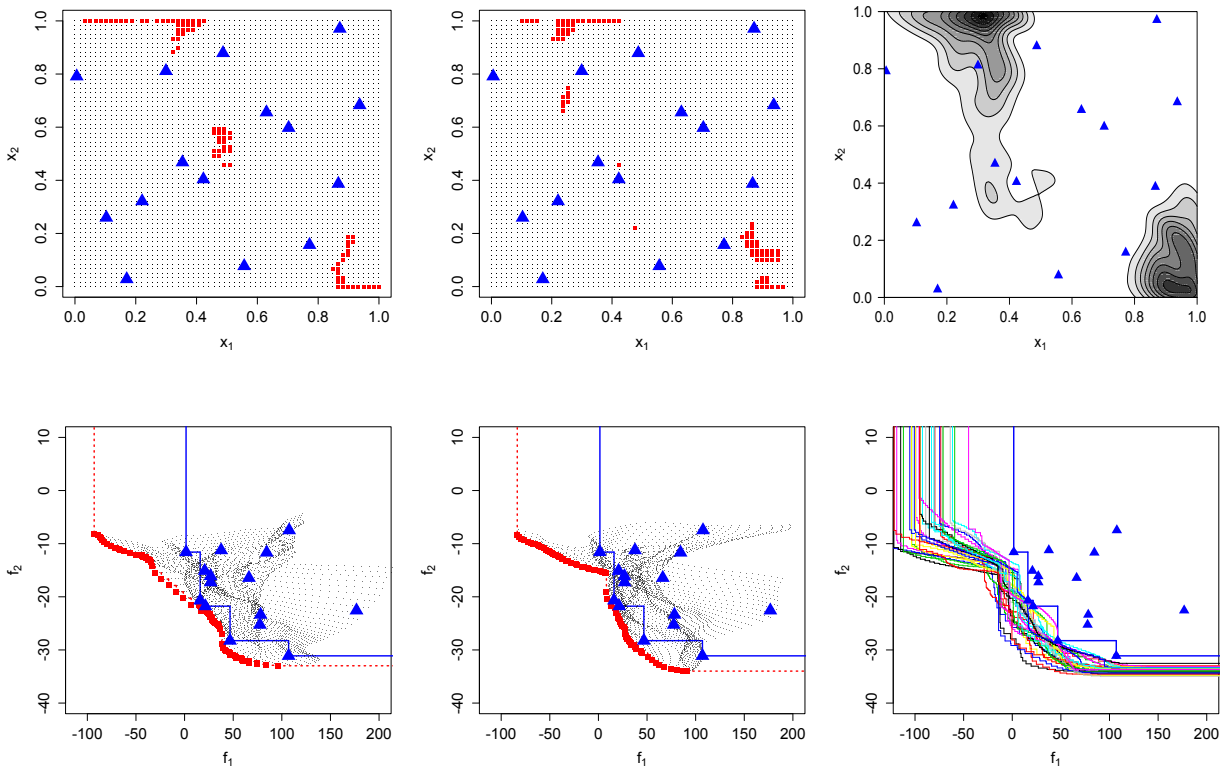


Figure 3.2 – Conditional Pareto sets and fronts corresponding to the GP models Y_1, Y_2, based on the observations $\mathcal{A}_n$ represented by blue triangles. Left and center: examples of two conditional simulations of Pareto sets (top) and fronts (bottom), where the simulations are performed on a regular 100×100 grid. The simulation points and simulated responses are plotted with dots. The corresponding non-dominated points are represented by red squares. Right: contour plot of the probability density of optimal points in the decision space estimated from 30 conditional simulations (top) and superposition of simulated conditional Pareto fronts (bottom).

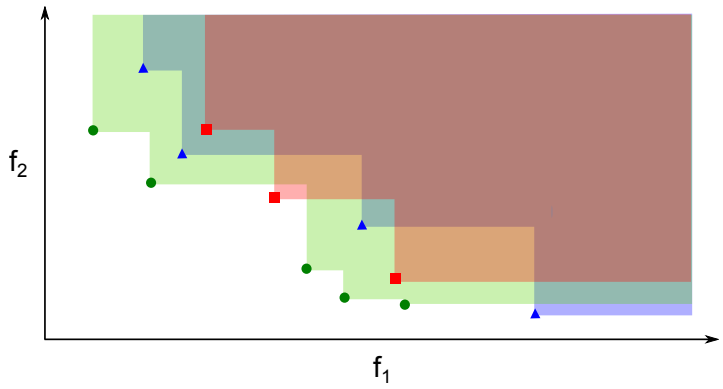


Figure 3.3 – Example of 3 realizations of RNP sets (points, triangles and squares) and the corresponding attained sets (shaded areas).

in the objective space. They have been considered to assess the performances of MO optimizers [FDFP05, ZKT08] under the term Random Non-dominated Point (RNP) sets: sets of random vectors in $\mathbb{R}^m$, non-dominated with respect to each other and with random finite cardinality (see e.g. [dFF10]). An alternative view is to consider the set of all objective vectors dominated by a realization of an RNP set, called an *attained set*. Realizations of RNP sets and the corresponding attained sets are presented in Figure 3.3.

In the mono-objective case, GP models provide analytical expressions of the expectation (Kriging mean) and uncertainty (variance of the pointwise prediction). It would be interesting to get their counterpart for the attained sets of simulated CPFs. Nevertheless, defining an expectation and/or an index of variability for sets is not straightforward and requires concepts from random sets theory [Mol05].

3.3.2 Basics from random sets theory: quantifying uncertainty with the Vorob'ev deviation

Before introducing related notions for CPFs, let us recall some general definitions. Set-valued random elements, in particular *random closed sets* [Mol05] received attention in the probability literature over the last decades. There exist several candidate notions to define the mean of a random closed set, see [Mol05] (Chapter 2). We choose a rather intuitive one, based on the notion of *coverage function*:

Definition 3.3.1 (Coverage function)**.** *Let $\mathcal{Y}$ be a random closed set on a topological space D (here $D \subset \mathbb{R}^m$ equipped with the topology induced by the Euclidean distance). The coverage function $p_{\mathcal{Y}}$ is defined by $p_{\mathcal{Y}} : x \in D \mapsto \mathbb{P}(x \in \mathcal{Y})$.*

This definition has been applied in the Kriging framework to estimate sets of critical input values [Che13, CGBM13]. It uses the Vorob'ev expectation, based on the upper level sets $\mathcal{Q}_\beta = \{z \in D, p_{\mathcal{Y}}(z) \geq \beta\}$, called β-quantiles.

Definition 3.3.2 (Vorob'ev expectation and deviation)**.** *Denoting by μ the Lebesgue measure on $\mathbb{R}^m$ and assuming that $\mathbb{E}(\mu(\mathcal{Y})) < +\infty$, the Vorob'ev expectation is the β^*-quantile $\mathcal{Q}_{\beta^*}$ such that $\mathbb{E}(\mu(\mathcal{Y})) = \mu(\mathcal{Q}_{\beta^*})$ if this equation has a solution, and otherwise it is defined from the condition $\mu(\mathcal{Q}_\beta) \leq \mathbb{E}(\mu(\mathcal{Y})) \leq \mu(\mathcal{Q}_{\beta^*})$, $\forall \beta > \beta^*$. The associated Vorob'ev deviation is the quantity $\mathbb{E}(\mu(\mathcal{Q}_{\beta^*} \Delta \mathcal{Y}))$, where Δ denotes the symmetric difference between sets, i.e. $\mathcal{Q}_{\beta^*} \Delta \mathcal{Y} = (\mathcal{Q}_{\beta^*} \cup \mathcal{Y}) \setminus (\mathcal{Q}_{\beta^*} \cap \mathcal{Y})$.*

The Vorob'ev expectation is a global minimizer of the deviation among all deterministic closed sets with volume equal to the average volume of $\mathcal{Y}$ (see [Mol05] for a proof): for any set M with $\mu(M) = \mathbb{E}\left(\mu\left(\mathcal{Y}\right)\right), \mathbb{E}\left(\mu\left(\mathcal{Q}_{\beta^*} \Delta \mathcal{Y}\right)\right) \leq \mathbb{E}\left(\mu\left(M \Delta \mathcal{Y}\right)\right)$.

3.3.3 Quantification of uncertainty on Pareto fronts using random set theory

In the MOO literature, the study of distribution location and spread of an attained set $\mathcal{X}$ rely on the attainment function $\alpha_{\mathcal{X}}$ [dFF10]: the probability for a given point in the objective space to be dominated by an RNP set, $\alpha_{\mathcal{X}} : x \in \mathbb{R}^m \mapsto \mathbb{P}(x \in \mathcal{Y})$.

Attained sets are also closed[2] and unbounded subsets in $\mathbb{R}^m$. Hence, the attained sets obtained with the simulated CPFs can be considered as realizations of a random closed set and are denoted by $\mathcal{Y}_i$, $(i = 1, \ldots, N)$. Looking again at Definition 3.3.1, one can see that the attainment function is in fact a coverage function. For proofs and discussions about the equivalence of the distribution of an RNP set and of the corresponding attained set, as well as for a definition of the attainment function in terms of coverage function, we refer to [dFF10]. This reference establishes the link between optimization results and random closed sets, in a case where the attainment function is computed from several runs of optimizers. Their comparison is then performed based on statistical hypothesis testing procedures.

In practice the attainment function is estimated by taking the proportion of RNP sets that dominates a given vector in the objective space:

Definition 3.3.3 (Empirical attainment function)**.** *Given a sample of attained sets* $\mathcal{Y}_1, \ldots, \mathcal{Y}_N$*, the empirical attainment function is defined as:* $\hat{\alpha}_N : \mathbb{R}^m \mapsto [0, 1]$*,* $\hat{\alpha}_N(z) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{z \in \mathcal{Y}_i\}}$ *where* $\mathbf{1}_{\{z \in \mathcal{Y}_i\}} = 1$ *if* $z \in \mathcal{Y}_i$*, 0 otherwise.*

An example of an empirical attainment function is presented in Figure 3.4, showing where in the objective space there is a high probability to improve on the current Pareto front.

Definition 3.3.2 requires that $\mathcal{Y}$ is bounded for its Vorob'ev expectation to exist. Hence it is necessary to define a reference point $\mathbf{R}$ to bound the integration domain. Since the Lebesgue measure of an attained set with respect to the reference point is the hypervolume indicator of the corresponding RNP set, denoted by $I_H(., \mathbf{R})$, the choice of $\mathbf{R}$ has a similar influence (see e.g. [ABBZ12]). Unless there is previous knowledge about the range of the objectives, we choose $\mathbf{R}$ as the maximum of each objective reached by the conditional simulations.

The determination of the Vorob'ev threshold β^* requires the volumes of β-quantiles:

$$\mu(\mathcal{Q}_\beta) = \int_{\Omega} \mathbf{1}_{\hat{\alpha}_N(z) \geq \beta} \mu(dz)$$

[2] As a finite union of closed sets (hyper quadrants).

They can be estimated by numerical quadrature, i.e. by computing the values of $\hat{\alpha}_N$ on a grid when there are few objectives or relying on Monte Carlo schemes. The other integrations are performed using hypervolume computation procedures and their complexity is related to the difficulty of computing the hypervolume in general. Hence measuring the uncertainty with the Vorob'ev deviation would be possible with any number of objectives but would require approximating integrals to keep the computations affordable. Note that since we take the objectives separately, simulating with more objectives simply requires computing the additional conditional simulations corresponding to those objectives.

The Pareto frontier of the Vorob'ev expectation provides us with an estimate of the Pareto front, as illustrated in Figure 3.4. The value of the Vorob'ev deviation gives an idea about the variability of the simulated CPF and can be monitored as observations are added, as will be shown in Section 3.4. The procedure to determine the value β^* corresponding to the Vorob'ev expectation (Vorob'ev threshold) as well as of the Vorob'ev deviation is described in Algorithm 3.

Algorithm 3 General procedure for estimating the Vorob'ev expectation and deviation

1: Generate N CPFs (see Algorithm 2).
2: **if R** is unknown **then** find the extremal values for the objectives over the RNP sets realizations:

$$\mathbf{R} = \left[\max_{i\in(1,\dots,N)} \mathbf{Y}_1^{(i)}, \dots, \max_{i\in(1,\dots,N)} \mathbf{Y}_m^{(i)} \right]$$

3: Define the integration domain: $\Omega = \left[\min_{i\in(1,\dots,N)} \mathbf{Y}_1^{(i)}, R_1 \right] \times \cdots \times \left[\min_{i\in(1,\dots,N)} \mathbf{Y}_m^{(i)}, R_m \right]$
4: Determine the average volume of the attained sets $\mathcal{Y}_i$:

$$\mathbb{E}(\mu(\mathcal{Y})) \approx \frac{1}{N}\sum_{i=1}^{N} \int_{\Omega} \mathbf{1}_{\{z\in\mathcal{Y}_i\}} \mu(\mathrm{d}z) \ = \frac{1}{N}\sum_{i=1}^{N} I_H(\mathcal{Y}_i, \mathbf{R})$$

5: Find the value of the Vorob'ev threshold β^* by dichotomy: set $a = 0, b = 1$:
while $b - a < \epsilon$ **do**
if $\mu\left(\mathcal{Q}_{\frac{a+b}{2}}\right) < \mathbb{E}(\mu(\mathcal{Y}))$ **then** $b = \frac{a+b}{2}$
else $a = \frac{a+b}{2}$
end if
end while, $\beta^* = \frac{a+b}{2}$
6: Estimate the Vorob'ev deviation:

$$\mathbb{E}(\mu(\mathcal{Q}_{\beta^*}\Delta\mathcal{Y})) \approx \frac{1}{N}\sum_{i=1}^{N} \int_{\Omega} \mathbf{1}_{\left(z\in\mathcal{Q}_{\beta^*}\Delta\mathcal{Y}_i\right)} \mu(\mathrm{d}z) = \frac{1}{N}\sum_{i=1}^{N} \left(2I_H(\mathcal{Q}_{\beta^*}\cup\mathcal{Y}_i, \mathbf{R}) - I_H(\mathcal{Q}_{\beta^*}, \mathbf{R}) - I_H(\mathcal{Y}_i, \mathbf{R})\right)$$

Remark 3.3.1. *The last equality in Algorithm 3 comes from the following:* $\int_\Omega \mathbf{1}_{z\in A\Delta B}\mu(dz) = I_{H2}(A,B,\boldsymbol{R}) + I_{H2}(B,A,\boldsymbol{R})$ *where* $I_{H2}(A,B,\boldsymbol{R})$ *is the binary hypervolume indicator, defined for instance in [ZTL+03]: the volume dominated by* A *and not by* B*, i.e.* $I_{H2}(A,B,\boldsymbol{R}) = I_H(A\cup B,\boldsymbol{R}) - I_H(B,\boldsymbol{R})$.

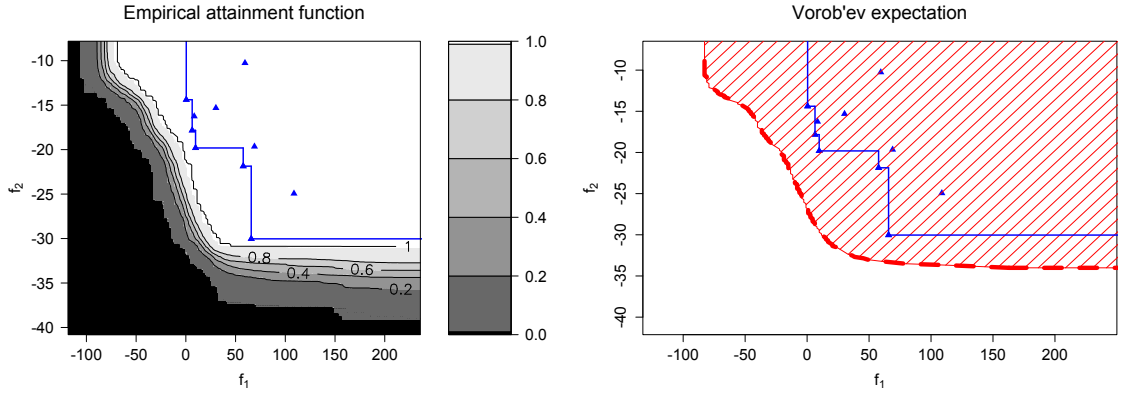


Figure 3.4 – Example of empirical attainment function (level sets, left) and corresponding Vorob'ev expectation (shaded area, right), based on 200 conditional simulations. The estimate of the underlying Pareto front (dashed line) is the Pareto frontier of the Vorob'ev expectation. The ten observations are marked by blue triangles.

Remark 3.3.2. *[dFF10] proposes the use of the Vorob'ev median* ($\mathcal{Q}_{0.5}$) *if no compact set is chosen for integration. While removing the problem of fixing the reference point, no equivalent of the Vorob'ev deviation seems available in this case.*

From a practical point of view, it is also useful for visualization purpose with few objectives to display the superposition of all the symmetric differences by defining an analogue of the attainment function:

Definition 3.3.4 (Symmetric-deviation function)**.** *The function* $\delta_{\mathcal{Y}} : z \in \mathbb{R}^m \mapsto P(z \in \mathcal{Q}_{\beta^*}\Delta\mathcal{Y})$ *is called the symmetric-deviation function of* $\mathcal{Y}$.

$\delta_{\mathcal{Y}}$ is the coverage function of $\mathcal{Q}_{\beta^*}\Delta\mathcal{Y}$. It is estimated with the empirical symmetric-deviation function:

$$\hat{\delta}_N(z) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{z\in\mathcal{Q}_{\beta^*}\Delta\mathcal{Y}_i\}}.$$

Figure 3.5 presents an example of a symmetric difference between two sets and an empirical symmetric-deviation function. This shows the variability around the estimated Pareto front: dark areas indicate regions where the estimation of the Pareto front is not known precisely.

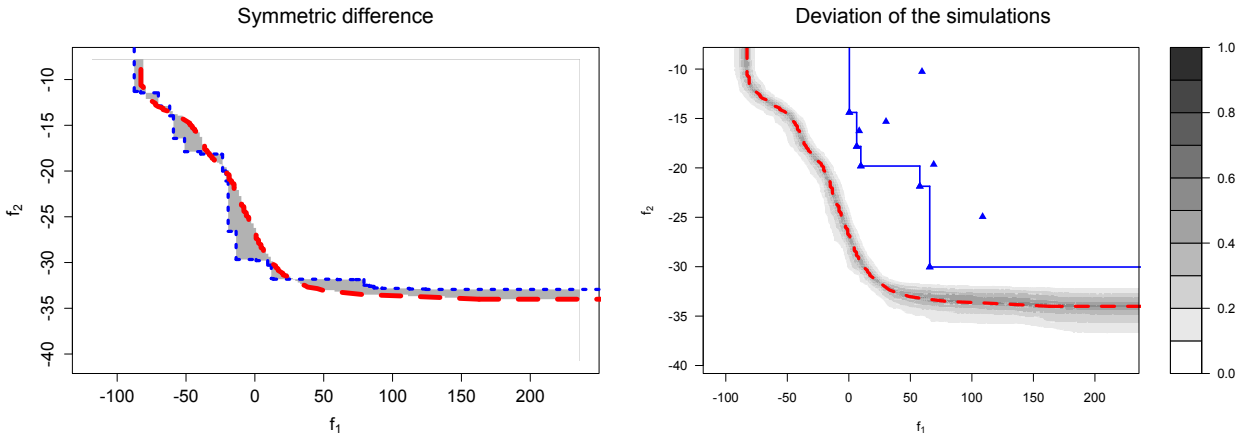


Figure 3.5 – Left: symmetric difference between the Vorob'ev expectation (the level line of the Vorob'ev threshold is represented by the red dashed line) and a simulated CPF's attained set (blue dotted line). Right: illustration of the deviation around the estimated Pareto front corresponding to Figure 3.4 with an example of empirical symmetric-deviation function (level plot).

3.4 Application

3.4.1 Two-dimensional bi-objective test problems

In this section, we illustrate the benefits of the proposed methodology for estimating Pareto fronts. We consider the following two variable, bi-objective optimization problems from the literature:

(P1) The problem presented in [Par12], which has a convex Pareto front:

$$f_1(\mathbf{x}) = \left(b_2 - \frac{5.1}{4\pi^2}b_1^2 + \frac{5}{\pi}b_1 - 6\right)^2 + 10\left[\left(1 - \frac{1}{8\pi}\right)\cos(b_1) + 1\right]$$

$$f_2(\mathbf{x}) = -\sqrt{(10.5 - b_1)(b_1 + 5.5)(b_2 + 0.5)} - \frac{1}{30}\left(b_2 - \frac{5.1}{4\pi^2}b_1^2 - 6\right)^2 - \frac{1}{3}\left[\left(1 - \frac{1}{8\pi}\right)\cos(b_1) + 1\right]$$

where $b_1 = 15x_1 - 5$, $b_2 = 15x_2$ and $x_1, x_2 \in [0, 1]$.

(P2) The ZDT3 problem [ZDT00] which has a disconnected Pareto front.

For each example, we start with a set of few observations that allow fitting initial Gaussian process models for the two objective functions. Then we add new points sequentially by maximizing the Expected Hypervolume Improvement, based on the formula detailed in [EDK11]. At each step, the Gaussian process models are updated and their hyperparameters re-estimated. These models are then used to simulate CPFs, from which we compute the

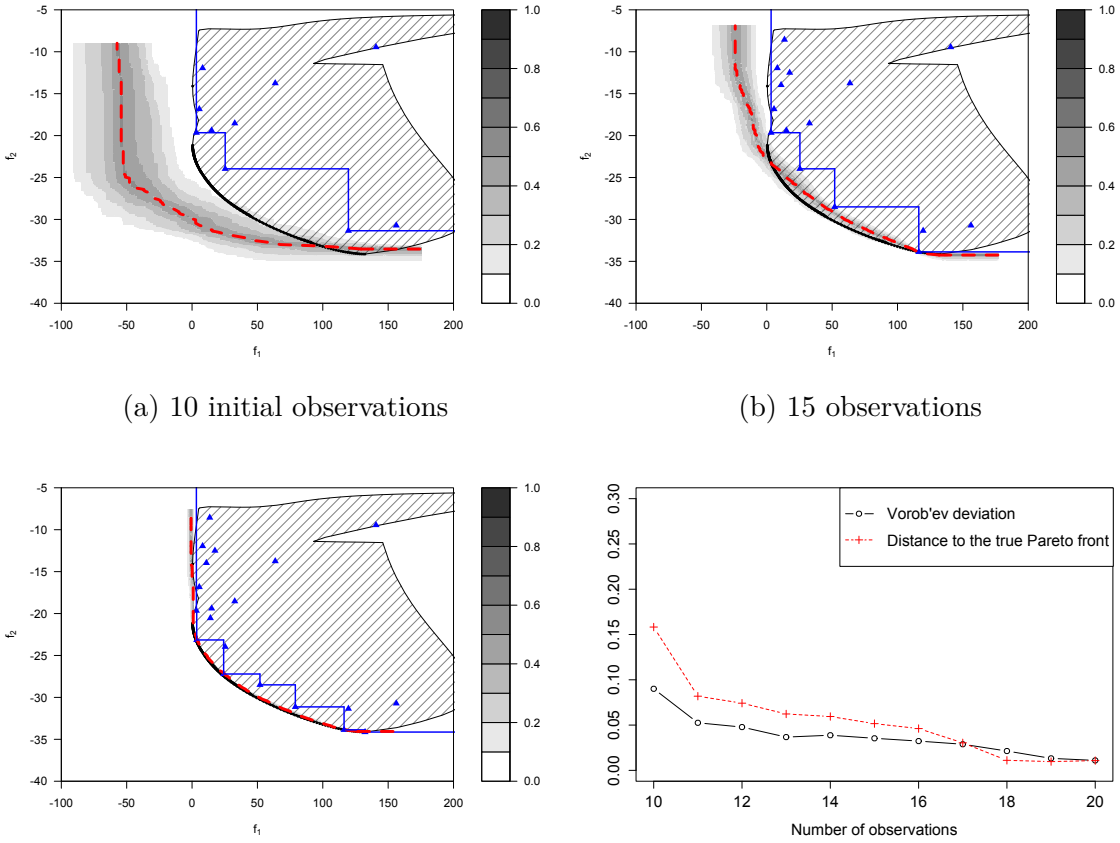
estimates of the Vorob'ev mean and the measures of uncertainty: Vorob'ev deviation and symmetric-deviation function. Since the integration domain varies as points are added, the values are displayed divided by the volume of this integration domain. The following test problems are fast to compute, so it is possible to compare the outcome of the proposed workflow to a reference Pareto front by using the volume of the symmetric difference.

The results are presented in Figure 3.6 and Figure 3.7, showing the evolution of the estimated Pareto fronts with the corresponding uncertainty around it. For the problem (P1) the sequence is detailed, demonstrating the strength of the proposed approach for giving insights about the uncertainty on the Pareto front. In particular, the uncertainty measures are helpful for choosing a minimal number of observations for approximating the Pareto front: while 10 initial observations may not be enough (Figure 3.6a) regarding the large symmetric-deviation, adding 10 more observations dramatically reduces the uncertainty (Figure 3.6c).

The conclusions are similar for problem (P2) Figure 3.7, where the Pareto front is disconnected, starting this time with 20 observations and sequentially adding ten more observations by Expected Hypervolume Improvement maximization. This example makes clear that the Vorob'ev expectation refers to the area dominated by the Pareto front. When the latter is disconnected, the dominated area's frontier is horizontal in the corresponding parts. The position of the cuts in the Pareto front depends on the model and simulations, resulting in a higher variation around cuts: a small change in the extent of a peak impacts the beginning of the next one (where the symmetric difference volume depends on the size of the disconnection).

One can note that the approximations obtained are dependent on the model accuracy. In Figure 3.6a the approximation is clearly too optimistic about the range of the first objective, due to an underestimation of the range parameters of the used Matérn covariance kernel ($\nu = 5/2$) combined with a misleading trend estimation in the corresponding surrogate. Similarly, for one objective, the expected value of the minimum would be misleading at the beginning.

Finally, we propose the use of the Vorob'ev deviation as a stopping criterion when the Pareto front location is known. An empirical rule could be a threshold on the Vorob'ev deviation (e.g. expected volume of the symmetric difference less than 1% of the integration volume) and detection of stagnation (e.g. under the threshold for several evaluations). On the examples Figures 3.6d and 3.7b, by considering the two last evaluations, the result would have been to stop for problem (P1) and continue for problem (P2). Note that this simple criterion would fail if the estimation of hyperparameters is misleading. A more robust version



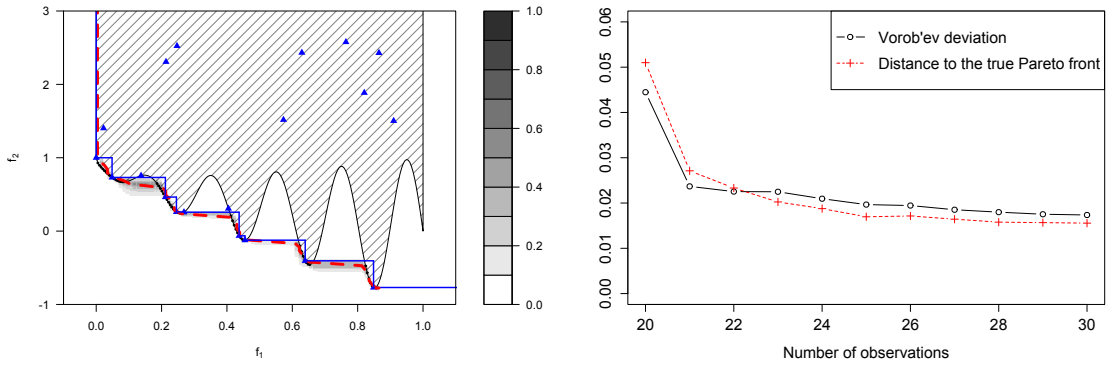
(a) 10 initial observations

(b) 15 observations

(c) 20 observations

(d)

Figure 3.6 – Evolution of the deviation with new observations added using Expected Hypervolume Improvement for Problem (P1). The shaded area represents the image of E by f with a thicker border for the Pareto front. Observations are marked with blue triangles and the blue solid line represents the current Pareto front. The dashed line is the estimated Pareto front, with the corresponding values of the symmetric-deviation in level plot. Bottom right: evolution of the Vorob'ev deviation scaled by the current integration volume (black solid line with circles) and evolution of the distance to the real Pareto front measured with the volume of the symmetric difference with the estimation (red dotted line with crosses).



(a) 30 observations (b)

Figure 3.7 – Evolution of the deviation with new observations added using Expected Hypervolume Improvement for Problem (P2). The figure description is the same as in Figure 3.6.

would be to integrate the uncertainty on the hyperparameters estimation in a full Bayesian framework [DR07]. More sophisticated values could also be derived inspired from [WTM11].

3.4.2 Additional experiments on conditional simulations

The aforementioned methodology depends on the number and location of simulation points used to obtain the CPFs from the Gaussian process models (cf. Algorithm 1 and first step of Algorithm 2). As a first study, we have compared two sampling strategies: i.i.d. uniform sampling and space-filling sampling relying on a Sobol sequence, again with 2 variables. The objective functions are taken as sample paths of centered Gaussian processes with Matérn covariance kernel ($\nu = 5/2$), with range parameters equal to $0.3/\sqrt{3}$ for f_1, and $0.5/\sqrt{3}$ for f_2.

We compute the approximation error over the set of non-dominated points obtained from the two sampling strategies. To compare the results with a reference set obtained with an NSGA-II [DPAM02] with archiving, three error indicators are used: hypervolume difference, epsilon and R2 quality indicators [ZTL+03]. The tests are repeated one hundred times. The results presented in Figure 3.8 show that space-filling sampling slightly outperforms uniform sampling to get an accurate estimation of the Pareto front. Additional tests performed showed a similar behavior with 3 variables but no difference for 10 variables, where the number of points considered was too small.

Here, only the impact of the error introduced by discretizing conditional simulations is studied. Given a few hundred to a few thousands of simulation points, it should remain negligible considering a relatively low number of variables, as is usually the case in appli-

cation examples up to six variables and six objectives [Sve11]. Alternatively, resorting to approximate spectral simulation methods such as the truncated Karhunen-Loève expansion could be considered.

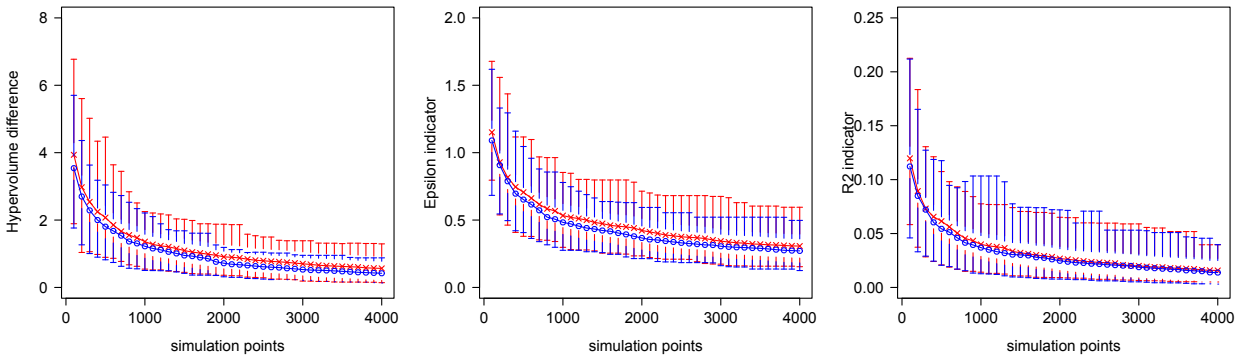


Figure 3.8 – Hypervolume difference, epsilon and R2 quality indicators between random reference Pareto fronts and approximations of them. A hundred conditional simulations for each of the two objectives considered are generated on a bi-dimensional grid with 225 points. Each of them is re-interpolated by Kriging and a reference Pareto front is obtained by applying an NSGA-II [DPAM02] with archiving. This reference set is used to compare the quality of approximations of the Pareto front obtained by uniform sampling (red crosses) or with a Sobol sequence (blue points) with respect to the number of simulation points. The error bars indicate the quantiles of level 5% and 95% for the hundred repetitions.

3.5 Conclusion and perspectives

We presented an original methodology to estimate and visualize the uncertainty of Pareto front approximations, based on Gaussian process conditional simulations. More precisely, the attainment function provides an estimation of the probability of dominating a given point in the objective domain. Then, a global uncertainty measure was defined relying on the theory of random sets through the concept of Vorob'ev deviation. It indicates the confidence of the model on the approximation of the attained set. The last tool is a visualization of the region of confidence for two or three objectives. Application on a higher number of objectives would be feasible, requiring the use of Monte Carlo methods for the computation of the various integrals. As illustrated on two bi-objective problems with convex or disconnected Pareto fronts, these measures can also be used as a basis to define stopping criteria in a sequential framework.

Further work is needed to analyze the different kinds of uncertainty and biases that may occur when applying the proposed methodology. In addition, techniques for simulating efficiently over more points and updating simulations with new observations [CEG14] should

be considered, as well as optimization of simulation points locations with re-interpolation or re-simulation [Oak99]. Another direction for future research includes the integration of the proposed uncertainty estimate in a Stepwise Uncertainty Reduction (SUR) strategy [CEG14] as an infill criterion.

Post-publication addendum

After the release of this article, further work has been performed following some of the given perspectives. The corresponding material is postponed in Appendix B. First, the SUR criterion based on the Vorob'ev deviation has been expressed and tested. It actually reduces the Vorob'ev deviation faster than EHI, but is more cumbersome to compute. In addition, even if the Pareto front estimation given by the Vorob'ev expectation is better, new observations may not be on it, which is not very appealing in practice. This statement motivated to consider adding new points based on the Vorob'ev expectation, i.e. the question of finding the best suited point in the input space to have a given location in the objective space. As a possible solution we propose to use the GP-LVM model [Law05] to build a model from objective to input space, based on the conditional Pareto sets that have not been used yet. Finally, to apply the quantification of uncertainty methodology to the case study in Chapter 8, we propose a procedure to select the simulation points from a multi-objective optimization conditional simulations.

Chapter 4

Quantifying uncertainty on Pareto fronts with copulas

In the previous chapter, the estimation of the location of the Pareto front is obtained from conditional simulations as a mean to get the probability of attainment, relying on metamodels. We propose here to study this problem from the point of view of multivariate analysis, introducing a probabilistic framework with the use of copulas. This approach enables the expression of level lines in the objective space, giving an estimation of the position of the Pareto front when the level tends to zero. In particular, when it is possible to use Archimedean copulas, analytical expressions for Pareto front estimators are available. This chapter has been published in the Information Sciences journal [BRR15]. Note that due to the hypothesis made in this chapter, i.e. that inputs are i.i.d. samples, the proposed approach is not suited for a direct use on expensive black-box simulators but is applicable on their surrogates. To emphasize this point, in addition to the content of the article, a comparison with the method developed in Chapter 3 is briefly discussed in the end and is the topic of Appendix C.

4.1 Introduction

Multi-objective optimization (MOO) received a lot of attention recently, including in particular developments on scalarization [GF15], hybrid approaches [GA10], evolutionary optimization (see e.g. [CZ14], [Deb08], [ZQL+11]) or surrogate-based optimization [VK10]. Since no solution usually minimizes every objective at once, the definition of a solution for a multi-objective optimization problem is generally defined as a compromise: a solution is said to be optimal in the Pareto sense if there exists no other solution which is better for every component. All the optimal points in the objective space form the Pareto front. As a result, optimizers provide a set of non-dominated points to approximate the Pareto front. Methods are then designed to seek some properties for these sets, such as uniformity and coverage.

Usually an optimization process starts with random sampling, either to generate an initial population or as a basis to construct a metamodel. The current Pareto front estimated from this first sample may be highly variable, especially when only a small number of function evaluations are available, corresponding to time-consuming functions. This is illustrated in Figure 4.1 for the bi-objective problem ZDT1 [ZDT00], with five 50-points initial samples. However, the stochastic nature of sampling provides a probabilistic framework that can be exploited to quantify this variability and to give a better initial localization of the Pareto front. More precisely, if $\mathbf{X} = (X_1, \ldots, X_d)$ is a d-dimensional random vector representing the inputs, and $f_1, \ldots, f_m$ the objective functions, then the Pareto front should be connected to the extreme level lines of the distribution of $\mathbf{Y} = (f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))$. To investigate such connection is the aim of the paper.

In the mono-objective situation, a similar probabilistic connection is studied by [QPNV10] to estimate the value of the extremum. Considering a small sample of n observations $(y_1, \ldots, y_n)$ of $\mathbf{Y}$, the minimum of $\mathbf{Y}$ is approximated using concepts from extreme order statistics. In multi-objective optimization, the connection seems to be new. Uncertainty quantification around the Pareto front has been recently considered by [BGR15a], using conditional simulations of Kriging metamodels and concepts from random sets theory. Whereas such approach is relevant in a sequential algorithm, it may be inappropriate in the initial stage that we consider here, due to a potentially large model error in metamodeling.

In this paper, we give a theoretical framework in which the Pareto front appears as a zero level line of the multivariate distribution $F_{\mathbf{Y}}$ of $\mathbf{Y}$. This problem is known in the probabilistic literature as support curve estimation (see e.g. [GGS12], [Hal82], [HNS97]). However, the existing methods rely on assumptions, such as domain of attraction or polynomial rate of decrease, which can hardly be checked in an optimization context. As an alternative, we propose to take advantage of copulas [Nel99] which are multivariate probability distributions with uniform marginals, allowing to consider separately the estimation of the marginals and the dependence structure. This allows estimating extreme level lines, without making specific assumptions about domain of attractions. Copulas have already been used in optimization, mainly in the variable space to estimate distribution in evolutionary algorithms, see e.g. [GPL$^+$13a], [GPL$^+$13b], [WZ10], while here we focus on the objective space. We propose a first estimation of the Pareto front relying on the empirical copula. Then, we consider the case where the copula belongs to the class of Archimedean copulas, parameterized by a function. This assumption can be checked visually or statistically with specific tests of the literature. If relevant, a better localization of the Pareto front is found. Furthermore, a parametric expression of the approximated Pareto front is available.

The paper is structured as follows. Section 4.2 proposes alternative definitions of the Pareto front from the point of view of the cumulative distribution function, presents some background about copulas and describes the estimation procedure in the Archimedean case. Section 4.3 discusses the applicability of the model and more specifically the consequences of the Archimedean copula model. Section 4.4 illustrates in several configurations the application of the proposed approach to Pareto front localization. Section 4.5 concludes and describes possibilities for further improvements.

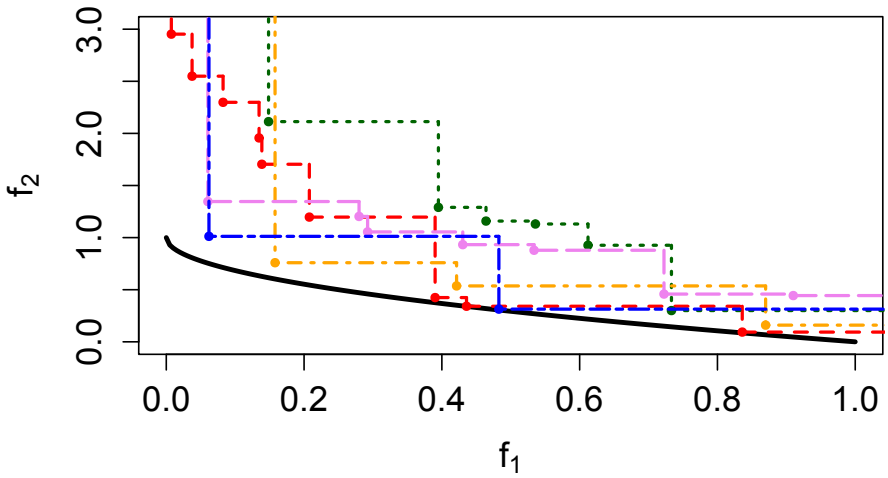


Figure 4.1 – Non-dominated points obtained with 5 different random samples (one color and type of line per sample) of 50 points for the bi-objective problem ZDT1. The true Pareto front is the black solid line.

4.2 Methodology

The present section describes the interest of using a probabilistic framework in multi-objective optimization by establishing the link between both domains. Based on the resulting theorem, the expression of level lines of the multivariate cumulative distribution functions $F_{\mathbf{Y}}$ using copulas is described as well as a procedure for their estimation. Empirical and parametric model are discussed, with emphasis on Archimedean models.

4.2.1 Link between Pareto front and level curves

For a variety of methods ranging from evolutionary optimization [Deb08] to surrogate-based methods [PWBV08], optimization starts with random sampling in the design space, with uniform sampling or with a random Latin Hypercube. In this case, it is possible to study the resulting observations in the objective space as a set of points. Specifically, assuming

that the outputs can be considered as independent and identically distributed (i.i.d.) random variables, they enter the scope of multivariate analysis.

Let us start with definitions of Pareto dominance and Pareto front, in a minimization context. For two points $\mathbf{y} = (y_1, \ldots, y_m)$ and $\mathbf{z} = (z_1, \ldots, z_m)$ of $\mathbb{R}^m$, $m \geq 2$, we first define the respective weak, strict and strong dominance operators $\preccurlyeq$, $\precnsim$ and $\prec$ as:

$$\begin{cases} \mathbf{z} \preccurlyeq \mathbf{y} & \Leftrightarrow & \forall i = 1, \ldots, m,\ z_i \leq y_i\,, \\ \mathbf{z} \precnsim \mathbf{y} & \Leftrightarrow & \forall i = 1, \ldots, m,\ z_i \leq y_i \ \text{ and } \ \exists i \in \{1, \ldots, k\}\,,\ z_i < y_i\,, \\ \mathbf{z} \prec \mathbf{y} & \Leftrightarrow & \forall i = 1, \ldots, m,\ z_i < y_i\,. \end{cases}$$

The expression *weak dominance* is used here as in [ZKT08], section 14.2, or [LZ11], *strict* dominance as in [DCD99], Definition 2.1, and *strong dominance* as in [Deb01], section 2.4.5. *Strict dominance* is usually referred simply as *dominance* or *Pareto dominance*. Notice that the terminology or symbols employed differ among authors.

Consider a subset $\mathbf{G}$ of $\mathbb{R}^m$. We define here the Pareto front $\mathcal{P}$ of the set $\mathbf{G}$ as the subset of $\mathbf{G}$ of all points that are weakly dominated only by themselves:

$$\mathbf{y} \in \mathcal{P} \ \Leftrightarrow \ \{\mathbf{z} \in \mathbf{G},\, \mathbf{z} \preccurlyeq \mathbf{y}\} = \{\mathbf{y}\} \tag{4.1}$$

This definition coincides with the more classical definition of Pareto front using strict dominance. The Pareto front is the set of Pareto optimal points, which are not strictly dominated:

$$\mathbf{y} \in \mathcal{P} \ \Leftrightarrow \ \forall \mathbf{z} \in \mathbf{G},\, \neg\,(\mathbf{z} \precnsim \mathbf{y})\ , \tag{4.2}$$

where $\neg$ is the logical not operator. The link with Equation (4.1) can be shown using the fact from Equation (4.2), if $\mathbf{y} \in \mathcal{P}$, $\{\mathbf{z} \in \mathbf{G}, \mathbf{z} \precnsim \mathbf{y}\} = \emptyset$ and using $\mathbf{z} \precnsim \mathbf{y} \ \Leftrightarrow \ (\mathbf{z} \preccurlyeq \mathbf{y}$ and $\mathbf{z} \neq \mathbf{y})$. This link has also been noticed, e.g. in [War83]. Definitions of weak Pareto front exist in the literature, using strong dominance, where $\mathbf{y} \in \mathcal{P}_{\text{weak}} \Leftrightarrow \forall z \in \mathbf{G}, z \not\prec y$, implying that $\mathcal{P} \subset \mathcal{P}_{\text{weak}}$.

Now assume that $\mathbf{G} = \mathbf{f}(\mathbf{E})$ is the image of a set $\mathbf{E} \subseteq \mathbb{R}^d$ by a vector-valued objective function $\mathbf{f} : \mathbf{E} \to \mathbb{R}^m$, with $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, $\mathbf{x} \in \mathbf{E}$. Then, the Pareto front of $\mathbf{f}$ is defined as the Pareto front of the image set $\mathbf{G}$. In this case we retrieve the usual interpretation that a solution in the objective space is Pareto-optimal if there exists no other solution which is better in every component: for $\mathbf{y} \in \mathbf{G}$, there exists no $\mathbf{z} \in \mathbf{G}$ such that $\mathbf{z} \preccurlyeq \mathbf{y}$ and $\mathbf{z} \neq \mathbf{y}$.

Assume that $\mathbf{X}$ is a random vector with values in $\mathbf{E}$ and let us denote $\mathbf{Y} = (Y_1, \ldots, Y_m)$

with $Y_i = f_i(\mathbf{X})$. Then if $\mathbf{Y}$ has an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbf{G}$, one easily gets $\mathbf{y} \in \mathcal{P} \Rightarrow \mathbb{P}\left[\mathbf{Y} \in \{\mathbf{z} \in \mathbf{G}, \mathbf{z} \preccurlyeq \mathbf{y}\}\right] = 0$. As a direct consequence, denoting $F_{\mathbf{Y}}(\mathbf{y}) = \mathbb{P}\left[\mathbf{Y} \preccurlyeq \mathbf{y}\right]$ the multivariate cumulative distribution function of $\mathbf{Y}$,

$$\mathbf{y} \in \mathcal{P} \quad \Rightarrow \quad F_{\mathbf{Y}}(\mathbf{y}) = 0\,. \tag{4.3}$$

The Pareto front thus belongs to the zero level set of $F_{\mathbf{Y}}$, $\{\mathbf{y} \in \mathbf{G}, F_{\mathbf{Y}}(\mathbf{y}) = 0\}$, which enlightens the connection between Pareto front and level sets of $F_{\mathbf{Y}}$.

Define the upper level set $L_\alpha^F = \{\mathbf{y} \in \mathbb{R}^m, F_{\mathbf{Y}}(\mathbf{y}) \geq \alpha\}$ with $\alpha \in (0,1)$, and the corresponding level line $\partial L_\alpha^F = \{\mathbf{y} \in \mathbb{R}^m, F_{\mathbf{Y}}(\mathbf{y}) = \alpha\}$. The following main result is that the upper level set L_α^F converges towards the area dominated by the Pareto front, when α tends to 0. This seems quite natural, as illustrated in Figure 4.2. However, the rigorous proof involves topological arguments that are different in the continuous and discrete cases. Some pitfalls are that for $\alpha > 0$, L_α^F are not necessarily included in $\mathcal{Y}$, the support of the probability distribution function of $\mathbf{Y}$, as illustrated in Figure 4.2, and that all points of $\mathcal{Y}$ are not necessarily dominated by the Pareto front, as when $\mathcal{Y} = \mathbb{R}^m$ for some unbounded objective functions[1].

Theorem 4.2.1. *Consider a random vector $\mathbf{Y}$ which admits a probability density function $f_{\mathbf{Y}}$ with respect to the Lebesgue measure on $\mathbb{R}^m$, and denote by $\mathcal{Y}$ its support (i.e. the essential support of the function $f_{\mathbf{Y}}$). Let $\mathcal{P}$ be the Pareto front of the set $\mathcal{Y}$. Define the respective weakly and strongly dominated sets:*

$$\mathcal{P}^{\succcurlyeq} = \bigcup_{\mathbf{y} \in \mathcal{P}} \{\mathbf{z} \in \mathbb{R}^m\,,\, \mathbf{y} \preccurlyeq \mathbf{z}\} \;\; \textit{and} \;\; \mathcal{P}^{\succ} = \bigcup_{\mathbf{y} \in \mathcal{P}} \{\mathbf{z} \in \mathbb{R}^m\,,\, \mathbf{y} \prec \mathbf{z}\}\,.$$

If all points of $\mathcal{Y}$ are dominated by the Pareto front, i.e. $\mathcal{Y} \subseteq \mathcal{P}^{\succcurlyeq}$, then the dominated area is obtained as the union of all upper level sets:

$$\mathcal{P}^{\succ} = \bigcup_{\alpha > 0} L_\alpha^F\,.$$

As a consequence we have $\lim_{\alpha \to 0} \mathbb{P}\left[\mathbf{Y} \in L_\alpha^F\right] = \mathbb{P}\left[\mathbf{Y} \in \mathcal{P}^{\succ}\right] = 1$.

Proof. We want to prove that the dominated area is equal to the area dominated by the set $L_0 = \bigcup_{\alpha > 0} L_\alpha^F$.

- $L_0 \subseteq \mathcal{P}^{\succ}$: It is sufficient to prove that if $\mathbf{y} \notin \mathcal{P}^{\succ}$, then $F_{\mathbf{Y}}(\mathbf{y}) = 0$.

[1] In particular, this is the case when objectives are GPs, see Appendix C.

– Assume first that $\mathbf{y} \notin \mathcal{P}^{\succcurlyeq}$, $F_{\mathbf{Y}}(\mathbf{y}) = \mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] = \mathbb{P}[\mathbf{Y} \in \{\mathbf{z} \in \mathcal{Y}, \mathbf{z} \preccurlyeq \mathbf{y}\}]$. One can show that if $\mathbf{y} \notin \mathcal{P}^{\succcurlyeq}$ and if $\mathbf{z} \preccurlyeq \mathbf{y}$ then $\mathbf{z} \notin \mathcal{P}^{\succcurlyeq}$, so that $\{\mathbf{z} \in \mathbb{R}^m, \mathbf{z} \preccurlyeq \mathbf{y}\} \cap \mathcal{P}^{\succcurlyeq} = \emptyset$. Finally $\{\mathbf{z} \in \mathbb{R}^m, \mathbf{z} \preccurlyeq \mathbf{y}\} \cap \mathcal{Y} = \emptyset$, since by assumption $\mathcal{Y} \subseteq \mathcal{P}^{\succcurlyeq}$, and $F_{\mathbf{Y}}(\mathbf{y}) = \mathbb{P}[\mathbf{Y} \in \emptyset] = 0$.

– Now assume that $\mathbf{y} \in \mathcal{P}^{\succcurlyeq} \backslash \mathcal{P}^{\succ}$. One can show that the Lebesgue measure $\mu(\mathcal{P}^{\succcurlyeq} \backslash \mathcal{P}^{\succ}) = 0$: Otherwise, there would exist a hypercube $\prod_{i=1}^{m}[a_i, b_i]$ included in $\mathcal{P}^{\succcurlyeq} \setminus \mathcal{P}^{\succ}$ such that for all $i = 1, \ldots, m$, $b_i > a_i$; This would be in contradiction with $\mathbf{b} = (b_1, \ldots, b_m) \in P^{\succ}$. Now, $F_{\mathbf{Y}}(\mathbf{y}) = \mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] = \mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y} \text{ and } \mathbf{Y} \in \mathcal{P}^{\succcurlyeq}]$ since by assumption $\mathcal{Y} \subseteq \mathcal{P}^{\succcurlyeq}$. This probability is equal to $\mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y} \text{ and } \mathbf{Y} \in \mathcal{P}^{\succcurlyeq} \setminus \mathcal{P}^{\succ}]$ because $\mathbf{y} \notin \mathcal{P}^{\succ}$ and $\mathbf{Y} \preccurlyeq \mathbf{y} \Rightarrow \mathbf{Y} \notin \mathcal{P}^{\succ}$. Thus this probability is 0 by absolute continuity of $\mathbf{Y}$ since $\mu(\mathcal{P}^{\succcurlyeq} \setminus \mathcal{P}^{\succ}) = 0$.

- $\mathcal{P}^{\succ} \subseteq L_0$: Recall that the complementary set $\mathcal{Y}^C$ of the support $\mathcal{Y}$ is defined as the union of all open sets Ω such that $f_{\mathbf{Y}}(.) = 0$ almost everywhere on Ω. Let $\mathbf{y} \in \mathcal{P}^{\succ}$, $\exists \mathbf{y}^* \in \mathcal{P}$ such that $\mathbf{y}^* \prec \mathbf{y}$. Denote $D_{\mathbf{y}} = \{\mathbf{z} \in \mathbb{R}^m, \mathbf{z} \preccurlyeq \mathbf{y}\}$. There exists an open set $B_{\mathbf{y}^*} \subseteq D_{\mathbf{y}}$ which contains $\mathbf{y}^*$. Now, we show that we cannot have $\mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] = 0$. Otherwise, by assumption of absolute continuity of $\mathbf{Y}$, this would imply that almost everywhere $f_{\mathbf{Y}}(.) = 0$ on $D_{\mathbf{y}}$. Then $B_{\mathbf{y}^*}$ would be an open set belonging to $\mathcal{Y}^C$. This would be in contradiction with $\mathcal{P} \subseteq \mathcal{Y}$, by definition of $\mathcal{P}$, which implies that the non-empty set $\mathcal{P} \cap B_{\mathbf{y}^*} \subseteq \mathcal{Y}$. Thus necessarily $\mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] > 0$, and there exists $\alpha > 0$ such that $\mathbf{y} \in L_\alpha^F$. Therefore $\mathcal{P}^{\succ} \subseteq L_0$.

Given that $\mathbb{P}[\mathbf{Y} \in \mathcal{P}^{\succcurlyeq}] = 1$ and $\mathbb{P}[\mathbf{Y} \in \mathcal{P}^{\succcurlyeq} \setminus \mathcal{P}^{\succ}] = 0$, the last part of the proposition is obtained by considering a decreasing sequence α_n and $L_{\alpha_n}^F$ and using Proposition 1.27 in [Bre92]. □

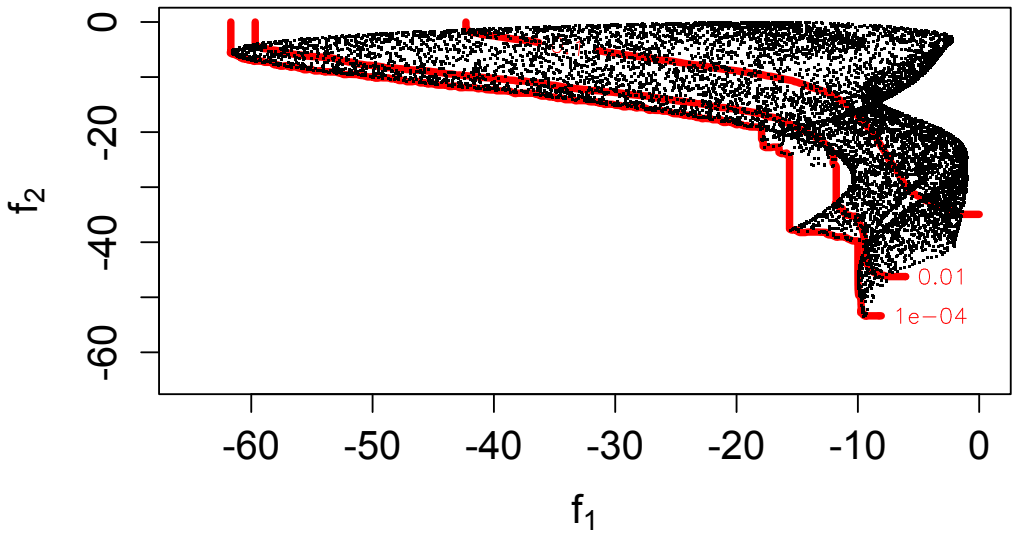


Figure 4.2 – Level lines ∂L_α^F with $\alpha = 0.0001, 0.01, 0.1$ of the empirical cumulative distribution function of $\mathbf{f}(\mathbf{X})$ obtained with sampled points (in black), showing the link between the level line of level α and the Pareto front $\mathcal{P}$ (apart from the vertical and horizontal components), as α tends to zero.

The case when $\mathbf{G}$ is discrete is also of practical interest and the corresponding result is detailed in Remark 4.2.1, slightly differing from Theorem 4.2.1.

Remark 4.2.1. *Let $\mathbf{Y}$ be a discrete random vector with support $\mathcal{Y}$. Let $\mathcal{P}$ be the Pareto front of the set $\mathcal{Y}$. Assume that all points of the support $\mathcal{Y}$ are dominated, i.e. $\mathcal{Y} \subseteq \mathcal{P}^{\succcurlyeq}$. Then*

$$\mathcal{P}^{\succcurlyeq} = \bigcup_{\alpha>0} L_\alpha^F \,.$$

Proof. Denote $L_0 = \bigcup_{\alpha>0} L_\alpha^F$. As $\mathbf{Y}$ is a discrete random vector, for any $\mathbf{y} \in \mathcal{Y}$, by definition of the support $\mathbb{P}[\mathbf{Y} = \mathbf{y}] > 0$. Now let us show that $\mathcal{P}^{\succcurlyeq} = L_0$.

- $\mathcal{P}^{\succcurlyeq} \subseteq L_0$: For any $\mathbf{z} \in \mathcal{P}^{\succcurlyeq}$, there exists $\mathbf{y} \in \mathcal{P}$ such that $\mathbf{y} \preccurlyeq \mathbf{z}$. Now for any $\mathbf{y} \in \mathcal{P}$, as $\mathcal{P} \subseteq \mathcal{Y}$ by definition of the Pareto front, then $\mathbb{P}[\mathbf{Y} = \mathbf{y}] > 0$. Then $\mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{z}] \geq \mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] > 0$ and $\mathcal{P}^{\succcurlyeq} \subseteq L_0$.
- $L_0 \subseteq \mathcal{P}^{\succcurlyeq}$: Let $\mathbf{y} \in L_\alpha^F$, $\alpha > 0$, then $\mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] \geq \alpha$. As $\mathbb{P}[\mathbf{Y} \preccurlyeq \mathbf{y}] = \sum_{\mathbf{y}_0 \in \mathcal{Y}, \mathbf{y}_0 \preccurlyeq \mathbf{y}} \mathbb{P}[\mathbf{Y} = \mathbf{y}_0] > 0$, there exists $\mathbf{y}_0 \in \mathcal{Y}$ such that $\mathbf{y}_0 \preccurlyeq \mathbf{y}$. Since by assumption $\mathcal{Y} \subseteq \mathcal{P}^{\succcurlyeq}$, $\mathbf{y}_0 \in \mathcal{P}^{\succcurlyeq}$ and since $\mathbf{y}_0 \preccurlyeq \mathbf{y}$, one gets $\mathbf{y} \in \mathcal{P}^{\succcurlyeq}$.

□

4.2.2 Expression of level curves using copulas

The m-dimensional distribution function $F_{\mathbf{Y}}$ contains all the information about the problem at hand, in particular about the Pareto front. The copula framework offers the possibility to study the dependence on the level lines separately from the univariate marginal distributions. Furthermore, under a particular Archimedean hypothesis detailed hereafter, the level lines have a parametric expression. To distinguish between the objective space and the copula space, we denote by $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{R}^m$ vectors in the objective space and by $\mathbf{u} = (u_1, \ldots, u_m) \in [0,1]^m$ vectors in the copula space.

Short summary on copulas

Consider some continuous random variables $Y_1, ..., Y_m$, and write $F_i(y_i) = \mathbb{P}[Y_i \leq y_i]$ the univariate cumulative distribution functions (cdf) of Y_i, $i = 1, \ldots, m$.

For independent random variables, the joint distribution of $(Y_1, \ldots, Y_m)$ is $F_{\mathbf{Y}}(y_1, \ldots, y_m) = \mathbb{P}[Y_1 \leq y_1, \ldots, Y_m \leq y_m] = \mathbb{P}[Y_1 \leq y_1] \cdot \ldots \cdot \mathbb{P}[Y_m \leq y_m]$, so that $F_{\mathbf{Y}}(y_1, \ldots, y_m) = C_\perp(F_1(y_1), .., F_m(y_m))$, where the product function $C_\perp(u_1, ..., u_m) = u_1 \ldots u_m$ is called the independence copula.
More generally, for possibly dependent random variables, Sklar's theorem [Skl59] states that

for any continuous multivariate distribution function $F_{\mathbf{Y}}$, there is a unique copula function C such that:

$$F_{\mathbf{Y}}(y_1, \dots, y_m) = C(F_1(y_1), \dots, F_m(y_m)) \,.$$

Copulas are essential tools for separating the univariate marginal distributions and the dependence structure of a random vector: first, a random vector has independent components if and only if $C = C_\perp$ (See [ELM03], Th. 2.5), making copulas more reliable than other dependence measures such as linear correlation coefficients. Moreover, strictly increasing transformations $g_1, \dots, g_m$ of the underlying random variables $Y_1, ..., Y_m$ do not change the copula of the joint random vector $(g_1(Y_1), \dots, g_m(Y_m))$ (See [ELM03], Th. 2.6). At last, by Sklar's theorem, a copula uniquely determines the joint distribution with given margins.

There naturally exist some constraints on copula functions. For continuous distributions, a function $C : [0,1]^m \to [0,1]$ is an m-dimensional copula if C is a joint cumulative distribution function of an m-dimensional random vector on the unit cube $[0,1]^m$ with uniform marginals, i.e. if there exist random variables $U_1, \dots, U_m$, uniformly distributed on $[0,1]$, such that

$$C(u_1, \dots, u_d) = \mathbb{P}\left[U_1 \leq u_1, \dots, U_m \leq u_m\right] \,.$$

Other classical properties like bounds on $C(u_1, \dots, u_d)$ are given in [Nel99].

Level curve expressions

Consider the Pareto front associated with the vector-valued objective function $\mathbf{f} : \mathbf{E} \to \mathbb{R}^m$, with $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, $\mathbf{x} \in \mathbf{E}$. We have seen in Theorem 4.2.1 that it was directly linked with level curves of the random vector $\mathbf{Y} = (Y_1, \dots, Y_m)$ where $Y_i = f_i(\mathbf{X})$, $i = 1, \dots, m$. We now use the copula framework to express these level curves.

From now on, we consider that the F_i's are continuous and invertible functions. Recall that the marginal distribution of Y_i is denoted F_i and that from Sklar's theorem [Skl59], there is a unique copula function C such that:

$$F_{\mathbf{Y}}(y_1, \dots, y_m) = C(F_1(y_1), \dots, F_m(y_m)) \,,$$

thus we can write for $\mathbf{u} \in [0,1]^m$:

$$C(u_1, \dots, u_m) = F_{\mathbf{Y}}(F_1^{-1}(u_1), \dots, F_m^{-1}(u_m)) \,.$$

Let $\alpha \in (0,1)$. The α-level lines of C, i.e. $\{\mathbf{u} \in [0,1]^m, C(u_1, \dots, u_m) = \alpha\}$ are denoted

∂L_α^C. They are connected to the level lines ∂L_α^F of $F_{\mathbf{Y}}$ by the following relationship:

$$\partial L_\alpha^F = \left\{ (y_1, \ldots, y_m) = (F_1^{-1}(u_1), \ldots, F_m^{-1}(u_m)) \in \mathbb{R}^m \, , \, \mathbf{u} \in \partial L_\alpha^C \right\}.$$

It follows that given a model of the copula and given the marginals, the levels lines of $F_{\mathbf{Y}}$ are obtained without additional effort. We describe next a specific model of copula which allows a parametric expression of those level lines. The methods to estimate both the copula model and the marginals will be detailed in Section 4.2.3.

Parametric form in the Archimedean case

A parameterization of the Pareto front has sometimes been proposed based on a metamodel of one output in function of the others [GVH+07] or using B-splines [BDD14]. It seems that in both cases the results do not necessarily follow the Pareto dominance, which might cause problems when dealing with Pareto fronts. Here we propose a method usable with any number of points, after sampling randomly in the design space, and respecting (weak) Pareto dominance for the proposed results.

Among other available parametric models of copulas (see e.g. [NQMRLÚF03]), a practical class of copula is the class of Archimedean copula, see e.g. [MN09]. The family of Archimedean copula is a flexible family that depends on a real function $\phi : \mathbb{R}^+ \to [0, 1]$, called the generator of the copula. An Archimedean copula is defined by

$$C_\phi(u_1, \ldots, u_m) = \phi\left(\phi^{-1}(u_1) + \ldots + \phi^{-1}(u_m)\right) \, ,$$

where the function ϕ^{-1} is the generalized inverse of the generator ϕ:

$$\phi^{-1}(t) = \inf\left\{ x \in \mathbb{R}^+, \phi(x) \leq t \right\} .$$

Note that depending on the author, ϕ and ϕ^{-1} are sometimes swapped. The generator ϕ is supposed to be continuous, m-monotone (see [MN09], which implies convexity), strictly decreasing on $[0, \phi^{-1}(0)]$ with $\phi(0) = 1$ and $\lim_{x \to +\infty} \phi(x) = 0$. If $\phi(x) > 0$ for all $x \in \mathbb{R}^+$, the generator and the corresponding Archimedean copula are said to be *strict*, otherwise they are called *non-strict*.

Also ϕ can be seen as a particular univariate survival function, so that in the following we will say that $\psi_0 = \phi^{-1}(0)$ is the *end-point* of the generator, with $\psi_0 < +\infty$ for non-strict generators, and $\psi_0 = +\infty$ for strict generators. Examples of generators of Archimedean copula models are given in Table 4.1. Clayton, Gumbel and Frank families are numbered No. 1, No.

4 and No. 5 respectively in [Nel99], along with more examples of strict and non-strict copulas.

Table 4.1 – Example of generators of classical Archimedean copulas from [KKPT14, Nel99], with Θ the definition domain of the parameter θ.

	$\phi(t)$	Θ	strict
Independent	$\exp(-t)$		yes
Clayton	$(1+\theta t)^{-1/\theta}$	$[-1,\infty)\backslash\{0\}$	$\theta > 0$
Gumbel	$\exp\left(-t^{1/\theta}\right)$	$[1,\infty)$	yes
Frank	$-\frac{1}{\theta}\log(1+\exp(-t)(\exp(-\theta)-1))$	$\mathbb{R}\backslash\{0\}$	yes
No. 2 in [Nel99]	$1-t^{1/\theta}$	$[1,\infty)$	no

The interest of representing C with an Archimedean copula (or a transformed copula [DBR13a]) is that we know how to express parametrically the level curves of such copulas, and consequently those of $F_{\mathbf{Y}}$.

Proposition 4.2.1 (Level curves for an Archimedean copula)**.** *Let $\mathcal{S}$ denotes the simplex $\mathcal{S} = \{\mathbf{s} \in [0,1]^m\,,\, s_1 + \ldots + s_m = 1\}$. If C_ϕ is an Archimedean copula with generator ϕ then for all $\alpha \in (0, \psi_0)$, we have*

$$\partial L_\alpha^{C_\phi} = \left\{\mathbf{u} \in [0,1]^m\,,\, u_i = \phi\left(s_i \phi^{-1}(\alpha)\right),\ 1 \le i \le m,\ \mathbf{s} \in \mathcal{S}\right\}, \tag{4.4}$$

and the level lines of $F_{\mathbf{Y}}$ are expressed as:

$$\partial L_\alpha^{F} = \left\{\mathbf{y} \in \mathbb{R}^m\,,\, y_i = F_i^{-1}(u_i)\,,\, u_i = \phi\left(s_i \phi^{-1}(\alpha)\right),\ 1 \le i \le m,\ \mathbf{s} \in \mathcal{S}\right\} \tag{4.5}$$

Proof. For an Archimedean copula with generator ϕ, the level curve of level $\alpha > 0$ is $\partial L_\alpha^{C_\phi} = \{\mathbf{u} \in [0,1]^m\,,\, C_\phi(u_1, \ldots, u_m) = \alpha\}$. Let $\mathbf{u} \in [0,1]^m$, $\mathbf{u} \in \partial L_\alpha^{C_\phi} \Leftrightarrow C_\phi(u_1, \ldots, u_m) = \alpha \Leftrightarrow \phi\left(\phi^{-1}(u_1) + \ldots + \phi^{-1}(u_m)\right) = \alpha$.
Suppose that in addition $\alpha \in (0, \psi_0)$, then $\mathbf{u} \in \partial L_\alpha^{C_\phi} \Leftrightarrow \frac{\phi^{-1}(u_1)+\ldots+\phi^{-1}(u_m)}{\phi^{-1}(\alpha)} = 1$.
By re-parameterizing with $s_i = \phi^{-1}(u_i)/\phi^{-1}(\alpha)$, $1 \le i \le m$ (equivalent to $u_i = \phi\left(s_i \phi^{-1}(\alpha)\right)$), we obtain that those s_i belongs to the simplex $\mathcal{S}$.
Hence $\partial L_\alpha^{C_\phi} = \{\mathbf{u} \in [0,1]^m\,,\, u_i = \phi\left(s_i \phi^{-1}(\alpha)\right), 1 \le i \le m, \mathbf{s} \in \mathcal{S}\}$. The expression of ∂L_α^F follows from the connection between ∂L_α^F and $\partial L_\alpha^{C_\phi}$. □

Other parameterizations of level curves of $F_{\mathbf{Y}}$ can be found in the literature (see e.g. [DBR13a], Proposition 2.4.).

A difference between strict and non-strict generators lies in the behavior of the level lines when α tends to 0:

Definition 4.2.1 (zero set, from [KKPT14, Nel99], extended to $m \geq 2$)**.** *The* zero set *of a copula C is the set*

$$S_0 = \{\mathbf{u} \in [0,1]^m \,,\, C(u_1, \ldots, u_m) = 0\} \,.$$

The Lebesgue measure on $\mathbb{R}^m$ of this zero set S_0 are denoted m_{S_0}.

As recalled in [KKPT14], based on [Nel99], the zero set is of Lebesgue measure zero if and only if the copula is strict. In the other case, for non-strict generators, the boundary of the zero set, $\{\mathbf{u} \in [0,1]^m \,,\, \phi^{-1}(u_1) + \ldots + \phi^{-1}(u_m) = \psi_0\}$ is called the *zero curve* of C_ϕ. For such a non-strict Archimedean copula and with $m = 2$, the zero curve can be expressed with

$$\partial L_0^{C_\phi} = \left\{(u_1, u_2) \in [0,1]^2 \,,\, u_2 = \phi(\phi^{-1}(0) - \phi^{-1}(u_1))\right\} \,.$$

This form can be extended to any dimension m by writing the m^{th} output as a function of the $m-1$ first ones. Still when $m = 2$, setting $\psi = \phi^{-1}$, the probability mass of the zero curve is equal to $-\frac{\psi(0)}{\psi'(0^+)}$, thus zero if the copula is strict or $\psi'(0^+) = -\infty$ (cf. Theorem 4.3.3. in [Nel99]).

Figure 4.3 illustrates the different cases described on the level lines of the copulas. With strict generators, the level lines converge towards the axis $[0,\infty) \times \{0\}$ and $\{0\} \times [0,\infty)$ as α tends to zero. This is not the case for non-strict generators, where zero sets have a strictly positive Lebesgue measure m_{S_0}, as visible on lower left corners of center and right panels of Figure 4.3.

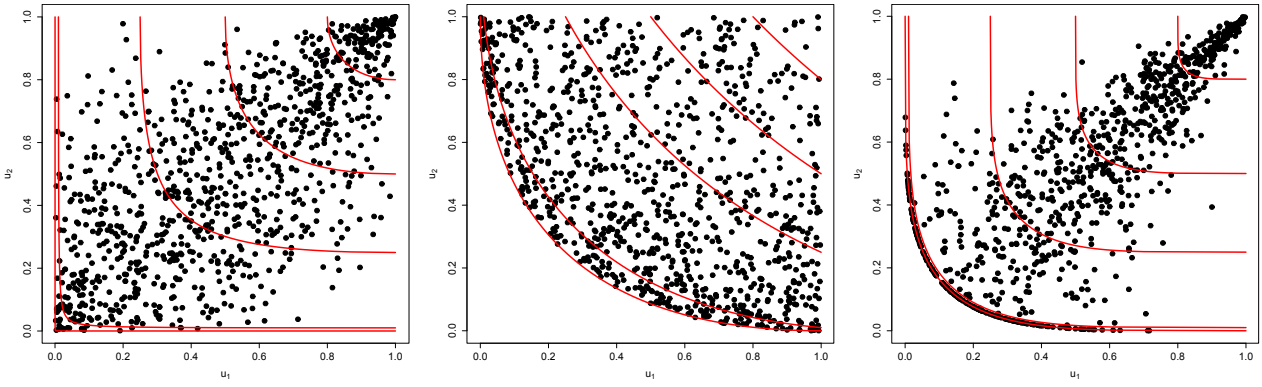


Figure 4.3 – Scatterplots of samples of a thousand points $\mathbf{U}^1, \ldots, \mathbf{U}^{1000}$ from Archimedean copulas with different generators and level lines with $\alpha = \{0, 0.01, 0.25, 0.5, 0.8\}$. Left: strict generator (Gumbel copula with $\theta = 2$). Center: non-strict generator (Clayton copula with $\theta = -0.8$). Right: non-strict generator with a probability mass on the zero curve (copula No.2 from [Nel99] with $\theta = 5$).

As a summary, the Archimedean family of copulas has the advantage to be very flexible

(it is indexed by a whole real function), to provide simple parametric expressions of the level curves, and to distinguish naturally degenerate or non degenerate Pareto front (via strict or non-strict generators). Note that other quantities related to the level curves, such as Kendall distributions, are derivable in the Archimedean case ([NS09], Section 2). In the next section, we explain how to get an approximation of the Pareto front $\mathcal{P}$ from this parametric expression of $\partial L_\alpha^{C_\phi}$. The relevance of this Archimedean model in practice is the subject of Section 4.3.

4.2.3 Estimation of the level lines

When working with black-box functions in order to find Pareto optimal solutions, the marginal distribution functions and copulas of the output $\mathbf{Y}$ must be estimated from the data. In the general case, only empirical estimation is possible while supposing that the copula is Archimedean gives parametric expressions for the level lines.

We aim here at proposing estimators of the level lines ∂L_α^F for small values of α. In particular, when α tends to 0, ∂L_α^F is directly related to the Pareto front $\mathcal{P}$ (see Theorem 4.2.1). As shown in Section 4.2.2, ∂L_α^F can be expressed as a function of ∂L_α^C and $F_1, \dots, F_m$. For $\alpha \in (0,1)$, the proposed plug-in estimators of the α-level lines are thus of the form

$$\widehat{\partial L}_\alpha^F = \left\{ (y_1, \dots, y_m) = (\hat{F}_1^{-1}(u_1), \dots, \hat{F}_m^{-1}(u_m)) \in \mathbb{R}^m, \mathbf{u} \in \partial L_\alpha^{\hat{C}} \right\}, \tag{4.6}$$

where $\hat{C}$ and $\hat{F}_1, \dots \hat{F}_m$ are respective estimators of C and $F_1, \dots F_m$, and where $\hat{F}_1^{-1}, \dots, \hat{F}_m^{-1}$ are generalized pseudo inverse of $\hat{F}_1, \dots \hat{F}_m$.

The proposed estimator of the Pareto front will be

$$\widehat{\mathcal{P}} = \widehat{\partial L}_{\alpha^*}^F$$

where $\alpha^* \in [0,1)$ is a small level value whose choice will be discussed hereafter. In the following, we first investigate the case where $\hat{C}$ is an empirical copula, and then the case where $\hat{C}$ is an Archimedean copula with generator ϕ.

Empirical copula

Several estimators of an empirical copula can be proposed, see e.g. [Deh79] and [OGV09]. Consider a set of n observations in $\mathbb{R}^m$: $\left\{ \mathbf{Y}^k = (Y_1^k, \dots, Y_m^k) \right\}_{k=1,\dots,n}$. Corresponding pseudo-observations are defined as $\left\{ \mathbf{U}^k = (U_1^k, \dots, U_m^k) \right\}_{k=1,\dots,n}$, with

$$U_i^k = \frac{1}{n+1} \sum_{j=1}^{n} \mathbb{1}_{\left\{ Y_i^j \leq Y_i^k \right\}}, \quad i \in \{1, \dots, m\}, \tag{4.7}$$

where $\mathbb{1}$ is the indicator function such that $\mathbb{1}_A = 1$ if the event A occurs and $\mathbb{1}_A = 0$ otherwise. The empirical copula can be estimated using the following formula:

$$\hat{C}_n(u_1, \ldots, u_m) = \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{\left\{U_1^k \leq u_1, \ldots, U_m^k \leq u_m\right\}}. \tag{4.8}$$

This is in fact the empirical distribution of the (normalized) ranks of the data. More details can be found in [Deh79] and [OGV09].

The empirical copula $\hat{C}_n$ being a step function, we mostly consider its level sets: $L_\alpha^{\hat{C}_n} = \left\{\mathbf{u} \in [0,1]^m, \hat{C}_n(u_1, \ldots, u_n) \geq \alpha\right\}$. In this case, different values of α may lead to the same level sets. An estimator of the level lines can be obtained by considering the frontiers of these upper level sets. This operation may be computationally costly, especially in large dimension. Furthermore, no simple analytical expression is available for these frontiers, justifying the use of parametric models when they are relevant.

Estimation in the Archimedean case

We consider here the case where the copula C is estimated by an Archimedean copula $\hat{C}_\phi$, having a generator ϕ. There exists a vast literature on the estimation of Archimedean copulas, see for example [GR93] or [KSS07]. In the case of parametric estimation, methods to fit an Archimedean copula C_ϕ rely for instance on Maximum Likelihood estimation or on dependence measures. A review of these methods and associated parameters estimators can be found e.g. in [KY10].

An important option is to consider a non-strict Archimedean copula, for which one has to estimate the *end-point* of the generator $\psi_0 = \phi^{-1}(0) = \inf\{x \in \mathbb{R}^+,\ \phi(x) = 0\}$. In parametric estimation, a recent method has been proposed in [KKPT14]. Among admissible parameters leading the zero curve to dominate all pseudo-observations, the choice is based on the functional form of the zero curve of the copula. The selected parameter is the one giving the closest zero curve to the pseudo-observations, under the assumption that the Lebesgue measure of the zero set is monotone with respect to the parameter. More formally, considering that the generator depends on ψ_0 and other parameters $\theta \in \Theta$, selected parameters are:

$$(\psi_0^*, \theta^*) = \underset{(\psi_0,\theta)\in\mathbb{R}^{*+}\times\Theta}{\operatorname{argmax}} m_{S_0}(\psi_0, \theta)\ s.t.\ \mathbf{U}^k \notin S_0, 1 \leq k \leq n\,, \tag{4.9}$$

where $m_{S_0}(\psi_0, \theta)$ represents the Lebesgue measure of the zero set S_0 of the copula (see Definition 4.2.1).

In the case of non-parametric estimation, among other different possible estimation pro-

cedures, one can cite [DKP08] or [GNZ11]. Under what is called *Frank's condition* (see [EGBHC13]), the Archimedean copula is uniquely determined by its diagonal section $\delta(u) = C(u, \ldots, u)$, $u \in [0, 1]$. For more details about the diagonal section of a copula, we refer to [Jaw09]. In this paper, for strict generators, we use a non-parametric estimator of the generator ϕ, based on an initial estimator of the diagonal section of the empirical copula, as detailed in Algorithm 2 in [DBR13b].

We summarize in Algorithm 4 a possible framework for estimating an Archimedean copula. One assumes that a catalog of methods is available along with the corresponding estimation procedures of the generators from the data. It may include parametric strict and non-strict estimators, and non-parametric strict estimators . In our applications examples, we used strict generators Clayton, Gumbel, Frank and non-strict generators of copulas No. 1 and No. 2 (see Table 4.1) with parametric estimation and in addition the non-parametric generator estimation from [DBR13b]. As discussed above, for each method an estimation procedure giving parameters from a data is available. The user has to select methods he wants to try. In order to select the best candidate model, it is possible to estimate a distance between the empirical copula $\hat{C}_n$ and a fitted copula C_ϕ, based on an integrated mean squared error (IMSE): $\int_{[0,1]^m} (\hat{C}_n(\mathbf{u}) - C_\phi(\mathbf{u}))^2 du_1 \ldots du_m$ or with a root mean squared error (RMSE): $\sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{C}_n(\mathbf{U}^i) - C_\phi(\mathbf{U}^i))^2}$. Notice that if the resulting distance is too high or if the Archimedean assumption seems irrelevant (see further Section 4.3), one may keep the empirical copula.

Algorithm 4 Estimation of the Archimedean copula

Input: Select candidates methods among a given catalog of generators (e.g. parametric strict and non strict, non-parametric).

1: Compute the pseudo-observations $\left\{\mathbf{U}^k\right\}_{k=1,\ldots,n}$ from the data, using Equation (4.7).
2: Compute the empirical copula $\hat{C}_n$ as in Equation (4.8).
3: **for** each selected candidate method **do**
4: Estimate parameters of the candidate method (see the corresponding literature).
5: Compute distance to empirical copula (e.g. RMSE).
6: **end for**

Output: Copula candidate $\hat{C}_\phi$ having smallest computed distance and corresponding distance.

Note that the storage of the copula is depending on the chosen method. A parametric copula can be characterized by a function for the generator and the value of its parameter whereas a non-parametric copula may be defined from a set of values of the generator together with an interpolation function.

Choice of the level α^*

Depending on the copula model: empirical copula or Archimedean with strict/non-strict generator, the behavior of L_α^C when α tends to zero differs. Consider an estimator $\hat{C}$ of the copula C. Notice that any admissible level α^* for the Pareto front estimator $\widehat{\mathcal{P}}$ should dominate all pseudo-observations (i.e. be such that $\forall k \in \{1,\dots,n\}, \mathbf{U}^k \in L_{\alpha^*}^{\hat{C}}$). Otherwise, pseudo-observations of the data would have a zero likelihood. Inspired by the method in [KKPT14], we also want to select the level α giving the closest zero curve to the pseudo-observations:

$$\alpha^* = \sup\left\{\alpha \in [0,1] : \forall k \in \{1,\dots,n\}, \mathbf{U}^k \in L_\alpha^{\hat{C}}\right\}.$$

It follows directly that:

Lemma 4.2.1 (conservative threshold α^*)**.**
Let us consider the threshold $\alpha^* = \sup\left\{\alpha \in [0,1] : \forall k \in \{1,\dots,n\}, \boldsymbol{U}^k \in L_\alpha^{\hat{C}}\right\}$, *then* $\alpha^* = \min_{k=1,\dots,n} \hat{C}(\boldsymbol{U}^k)$.

Proof. Let $\alpha_1 \leq \alpha_2$, for any $\mathbf{u} \in L_{\alpha_2}^{\hat{C}}$, $\mathbf{u}$ is also in $L_{\alpha_1}^{\hat{C}}$ by definition of the upper level sets. Hence by taking $\alpha = \min_{i=1\dots n} \hat{C}(\mathbf{U}^i)$, all $\mathbf{U}^k \in L_\alpha^{\hat{C}}$, $k \in \{1,\dots,n\}$. Furthermore, there exists $k^* \in \{1,\dots,n\}$ such that $\hat{C}(\mathbf{U}^{k^*}) = \alpha$, so that for any $\alpha' > \alpha$, $\mathbf{U}^{k^*} \notin L_{\alpha'}^{\hat{C}}$. □

We discuss here consequences of this choice of the level α^* on the estimated copulas considered in this paper:

- For empirical copulas, the conservative threshold is almost surely $\alpha^* = \frac{1}{n}$ since any inferior value results in a zero set included in the axis $[0,\infty) \times \{0\}$ and $\{0\} \times [0,\infty)$.
- For strict Archimedean copulas, this choice leads to $\alpha^* > 0$ as soon as pseudo-observations are all strictly positive. It thus avoids setting $\alpha^* = 0$ which would lead to a degenerate zero set included in the axis $[0,\infty) \times \{0\}$ and $\{0\} \times [0,\infty)$.
- For non-strict Archimedean copulas, the choice of ψ_0^* as in Equation (4.9) leads to $\alpha^* = 0$ by construction. It would be possible to set smaller values of ψ_0^* leading to admissible parameter $\alpha^* \geq 0$, but for the sake of simplicity, we have considered here only the case where ψ_0^* was given by Equation (4.9).

Estimation of the marginals

The univariate marginals and their inverses also need to be estimated. This can be performed with the empirical quantiles or any method using truncated or non-truncated kernel density estimation. In some experiments with scarce data, we use the method proposed in [QPNV10] to estimate the support of the cumulative distribution function and its inverse, based on a

catalog of beta distributions.

We summarize in Algorithm 5 a general methodology for estimating a marginal distribution: one assumes that a catalog of classical parametric and non-parametric estimators is available. The user has to select the estimators he wants to try, the algorithm selecting the best one using a chosen distance to the empirical distribution, e.g. Kolmogorov-Smirnov (K.-S.) distance.
Notice that if the resulting distance is too high, the user can try other members of the catalog or keep the empirical distribution function (thus losing the ability of smoothing and extrapolating).

Algorithm 5 Estimation of one marginal

Input: Select candidates estimators among a given catalog (including classical parametric or kernel-based estimators).
1: Compute empirical distribution function of selected marginal from data.
2: **for** each selected candidate **do**
3: Estimate parameters (e.g. by maximum likelihood estimation).
4: Compute distance to empirical marginal distribution (e.g. K.-S. distance).
5: **end for**
Output: Candidate distribution having smallest computed distance and corresponding distance.

Increasing the number of objectives usually implies to sample more points in the variable space to cover the objective space, providing more points to estimate each of the univariate marginals.

At last, the expression of the estimated level lines of the multivariate distribution also depends on the inverse functions of the marginal distributions, see Equation (4.6). Some parametric methods have been proposed in order to fit univariate distributions and to obtain straightforward simple expressions for their inverse functions, see e.g. [BR12].

General algorithm

We recapitulate the general procedure for estimating level lines of an Archimedean copula in Algorithm 6. If one would rather use the empirical copula, as discussed in the next Section, it is sufficient to compute the empirical copula and to estimate level lines from Equations (4.6) and (4.8).

Algorithm 6 Estimation of the level curves of $F_{\mathbf{Y}}$ and of the Pareto front with an Archimedean copula

Input: Set of levels $A = \{\alpha_1, \ldots, \alpha_n\}$.
1: Get estimation of the Archimedean copula, $\hat{C}_\phi$, from Algorithm 4.
2: Compute threshold α^* as prescribed in Section 4.2.3.
3: Compute levels lines of the copula, $\partial L_\alpha^{\hat{C}_\phi}$, $\alpha \in A \cup \alpha^*$, with Equation (4.4).
4: Get estimations of univariate marginal distributions by Algorithm 5.
5: Compute levels lines of the cdf of $F_{\mathbf{Y}}$, ∂L_α^F, $\alpha \in A \cup \alpha^*$, with Equation (4.5).
Output: Pareto front estimation $\widehat{\mathcal{P}} = \partial L_{\alpha^*}^F$ and levels lines ∂L_α^F, $\alpha \in A$.

4.3 Pertinence of the Archimedean model

The interest of such a model, if appropriate, lies in the fact that if the dependency is accurately modeled, every observation gives information about the whole Pareto front, providing a continuous and smooth estimation. The parametric expression for the level curve of $F_{\mathbf{Y}}$, written in Proposition 4.2.1, requires the assumption that the copula describing the dependency structure can be approximated by an Archimedean copula. This section provides a discussion of the associated restrictions in practice and about the choice of an Archimedean copula model from the alternatives presented in Section 4.2.3.

4.3.1 Properties of Archimedean copulas: convexity, symmetry and associativity

The Archimedean model is convenient and tractable even with many objectives, but it imposes symmetry and associativity. This corresponds, when $m = 2$, to $C(u_1, u_2) = C(u_2, u_1)$ and $C(C(u_1, u_2), u_3) = C(u_1, C(u_2, u_3))$ for any $(u_1, u_2, u_3) \in [0, 1]^3$. In addition the level lines of the copula are convex.

Proposition 4.3.1 (Convexity of $\partial L_\alpha^{C_\phi}$)**.** *The level curves of an Archimedean copula of dimension m are convex.*

Proof. This proposition is demonstrated in the case $m = 2$ in [Nel99]. In the case $m > 2$, the result is still valid.
Given $\mathbf{u} = (u_1, \ldots, u_m)$ and $\mathbf{v} = (v_1, \ldots, v_m)$ two points of $\partial L_\alpha^{C_\phi}$. Given $\lambda \in [0, 1]$, we denote $\mathbf{w} = \lambda\mathbf{u} + (1 - \lambda)\mathbf{v}$. In dimension m, the generator ϕ is a m-monotone function, implying in particular that ϕ^{-1} is a decreasing convex function. Hence for all $i \in \{1, \ldots, m\}$, $\phi^{-1}(w_i) = \phi^{-1}(\lambda u_i + (1 - \lambda)v_i) \leq \lambda\phi^{-1}(u_i) + (1 - \lambda)\phi^{-1}(v_i)$. Then
$\phi^{-1}(w_1) + \ldots + \phi^{-1}(w_m) \leq \lambda\left(\phi^{-1}(u_1) + \ldots + \phi^{-1}(u_m)\right) + (1 - \lambda)\left(\phi^{-1}(v_1) + \ldots + \phi^{-1}(v_m)\right)$.
Since $\mathbf{u}$ and $\mathbf{v}$ belongs to $\partial L_\alpha^{C_\phi}$,

$$\left(\phi^{-1}(u_1) + \ldots + \phi^{-1}(u_m)\right) = \left(\phi^{-1}(v_1) + \ldots + \phi^{-1}(v_m)\right) = \phi^{-1}(\alpha).$$

Then $\phi^{-1}(w_1) + \ldots + \phi^{-1}(w_m) \leq \phi^{-1}(\alpha)$, which is equivalent to $\mathbf{w} \in L_\alpha^{C_\phi}$. □

Note that having convex level lines does not imply that the level lines in the objective space will also be convex since it depends on the marginals. In the case when it is known that the Pareto front is convex, a sufficient condition to ensure the convexity of the Pareto front is to have concave marginals with an Archimedean copula.

Proposition 4.3.2 (Convexity of ∂L_α^F)**.** *If the marginals $F_1, \ldots, F_d$ are concave, then the level lines of ∂L_α^F are convex.*

Proof. Given $\mathbf{y} = (y_1, \ldots, y_m)$ and $\mathbf{z} = (z_1, \ldots, z_m)$ two points of ∂L_α^F. Given $\lambda \in [0, 1]$, we denote $\mathbf{w} = \lambda\mathbf{y} + (1 - \lambda)\mathbf{z}$. The F_i's are concave, hence for all $i \in (1, \ldots, m)$, $\lambda F_i(y_i) + (1 - \lambda)F_i(z_i) \leq F_i(w_i)$. Since the generator ϕ^{-1} is a decreasing convex function, $\lambda\phi^{-1}(F_i(y_i))+(1-\lambda)\phi^{-1}(F_i(z_i)) \geq \phi^{-1}(F_i(w_i))$. And thus by summation $\lambda(\phi^{-1}(F_1(y_1)) + \cdots + \phi^{-1}(F_m(y_m))) + (1 - \lambda)(\phi^{-1}(F_1(z_1)) + \cdots + \phi^{-1}(F_m(z_m))) \geq \phi^{-1}(F_1(w_1)) + \cdots + \phi^{-1}(F_m(w_m))$. Now, ϕ is a decreasing function:
$\phi\Big(\lambda(\phi^{-1}(F_1(y_1)) + \cdots + \phi^{-1}(F_m(y_m))) + (1 - \lambda)(\phi^{-1}(F_1(z_1)) + \cdots + \phi^{-1}(F_m(z_m)))\Big)$
$\leq \phi\left(\phi^{-1}(F_1(w_1)) + \cdots + \phi^{-1}(F_m(w_m))\right) = F_{\mathbf{Y}}(\mathbf{w})$.

Since $\mathbf{y}$ and $\mathbf{z}$ are in ∂L_α^F, $\phi^{-1}(F_1(y_1)) + \cdots + \phi^{-1}(F_m(y_m)) = \phi^{-1}(F_1(z_1)) + \cdots + \phi^{-1}(F_m(z_m)) = \phi^{-1}(\alpha)$. Then $F_{\mathbf{Y}}(\mathbf{w}) \geq \alpha$, which means that $\mathbf{w} \in \partial L_\alpha^F$. □

If the level curves must be concave, then the use of survival copulas (associated with $1 - F_{\mathbf{Y}}$) can be a solution.

It is important to mention that even if the hypothesis of Archimedeanity is restrictive, it can still cover a great variety of situations, as illustrated in Figure 4.4 with varying copulas and marginals. For the Frank copula, which is strict, the real Pareto front is reduced to (0,0). The assessment of this hypothesis is detailed in the next paragraph.

4.3.2 Archimedeanity tests—choosing between the different options

An immediate solution is to test whether the hypothesis of Archimedeanity holds or not. Recent works exist in the bivariate case, see e.g. [BDV12]. Otherwise a simple test is to compare visually the level curves of the empirical copula with those of the fitted Archimedean copula in the same spirit as the normal probability plot in dimension one.

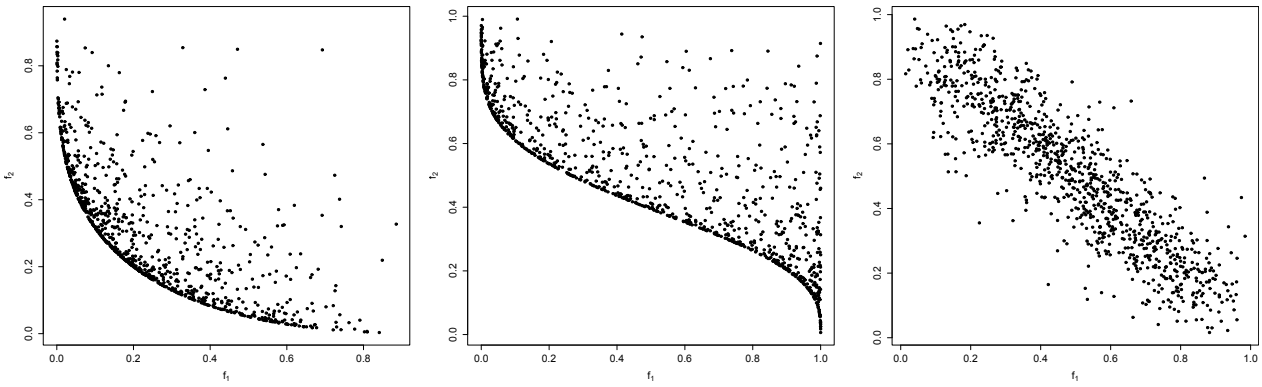


Figure 4.4 – Scatterplots (in the objective space) with a thousand of sample points $\mathbf{Y}^1, \ldots, \mathbf{Y}^{1000}$ generated from Archimedean copulas models and further applying inverse of beta distribution functions as univariate marginals. Left: Clayton copula, $\theta = -0.8$, $F_1 = Beta(1, 3)$ and $F_2 = Beta(1.5, 3.5)$. Center: Clayton copula, $\theta = -0.8$, $F_1 = Beta(0.5, 0.5)$ and $F_2 = Beta(2.5, 2.5)$. Right: Frank copula, $\theta = -12$, $F_1 = Beta(2, 2)$ and $F_2 = Beta(2, 2)$.

It remains to decide which Archimedean model is the best to estimate the Pareto front, by trying the different possibilities: parametric strict and non-strict models or non-parametric strict models. Non-strict parametric models seem best suited to estimate Pareto fronts due to the presence of the zero set but in certain circumstances non-parametric strict models perform better. For parametric families with analytical strict generator function, one can mention for instance the Clayton family ($\theta > 0$), Gumbel family or Frank families of Archimedean copulas. The parameters are evaluated using Maximum Likelihood.

Estimating a non-parametric generator from the data gives more flexibility when the Archimedean hypothesis is too strong, as illustrated in the applications. Even if it cannot capture the dissymmetry of the empirical copula, the fitted model is often more accurate with this non-parametric generator.

Non-strict Archimedean copulas play a particular role for modeling the Pareto front, due to their non degenerate zero-sets. A generator of such a copula can be linked to a non-observable univariate random variable (e.g. the radial part of the copula, see [MN09]). The maximum value of such random variable is directly related to the location of the Pareto front, and using *end-point* probabilistic literature would be an interesting perspective (see e.g. [GGS12], [HW99], [LPX11], [Loh84], and references therein).

4.4 Applications

To illustrate the benefits of the approach proposed in Algorithm 6 we take three classical bi-objective f_1, f_2 problems from the MOO literature: the ZDT1, ZDT6 [ZDT00] and Poloni

[PGOP00] test problems. They have respectively convex, concave and disconnected Pareto fronts. The variable dimension is two in all the examples, but it could be much higher since the estimation procedure only deals with the objective space. Note that with an increasing variable dimension, it becomes necessary to increase the sample size. We use R packages *copula* [HM11, HKMY14, KY10, Yan07] for estimating strict Archimedean copulas and *ks* [Duo14] for kernel density estimation.

4.4.1 Estimation of the Pareto front for the ZDT1 test problem

The first test problem, ZDT1, is a relatively simple benchmark problem:

ZDT1. Let $\mathbf{x} \in [0,1]^d$ and $g(\mathbf{x}) = 1 + \frac{9}{d-1}\sum_{i=2}^{d} x_i$. Consider:

$$f_1(\mathbf{x}) = x_1, \qquad f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}}\right).$$

Here we choose $d = 2$ and draw a sample of size $n = 100$, uniformly in $[0,1]^2$.

The first step is to estimate the marginals. As one can see from Figure 4.5, the parametric estimation based on beta distribution gives a good fit of the empirical inverse of the marginals while non-parametric estimation is clearly too optimistic[2] on the range of the ZDT1 test problem : $[0,1]$ for f_1 and $[0,10]$ for f_2. Then we select the model with the best fit for the copula, which is the non-parametric copula model in this case, based on Figure 4.6. Here several models would be acceptable, since all the other Archimedean models look close to the empirical copula, except the non-strict model No. 2. However, the RMSE error on the pseudo-observations is the lowest with the non-parametric generator.

Finally we obtain the estimation of the position of the Pareto front, cf. Figure 4.7. While being slightly too optimistic on the right side, it is more accurate than the Pareto front approximation from the non-dominated points of the observations. Also a comparison with what would have been obtained using only the empirical copula illustrates that the Archimedean hypothesis brings a smoother and better localization. It can be observed that due to errors on the estimation of the marginals, the Pareto front approximation using the empirical copula may be dominated by $\mathcal{P}_n$ on various parts.

[2] In the sense that it would give a better Pareto front than in reality, in terms of Pareto dominance.

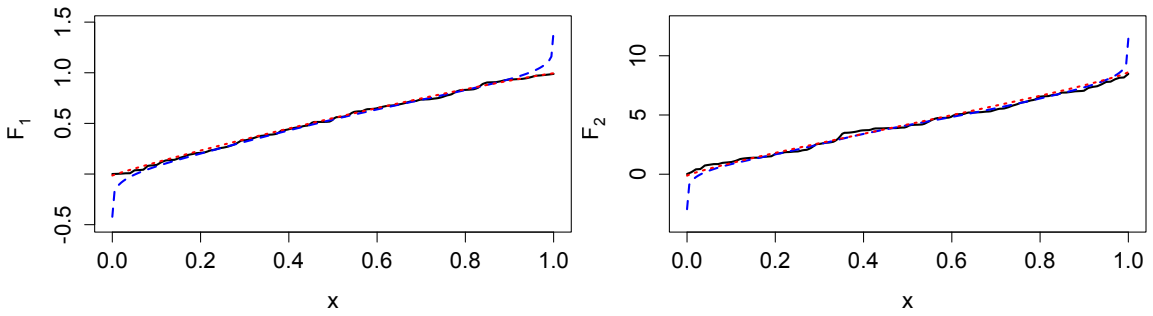


Figure 4.5 – ZDT1 test problem: comparison between three estimation methods of the marginals F_1 and F_2 – empirical (black solid line), kernel density (blue dashed line) and fit of a generalized beta distribution (red dotted line) – for the objectives f_1 (left) and f_2(right).

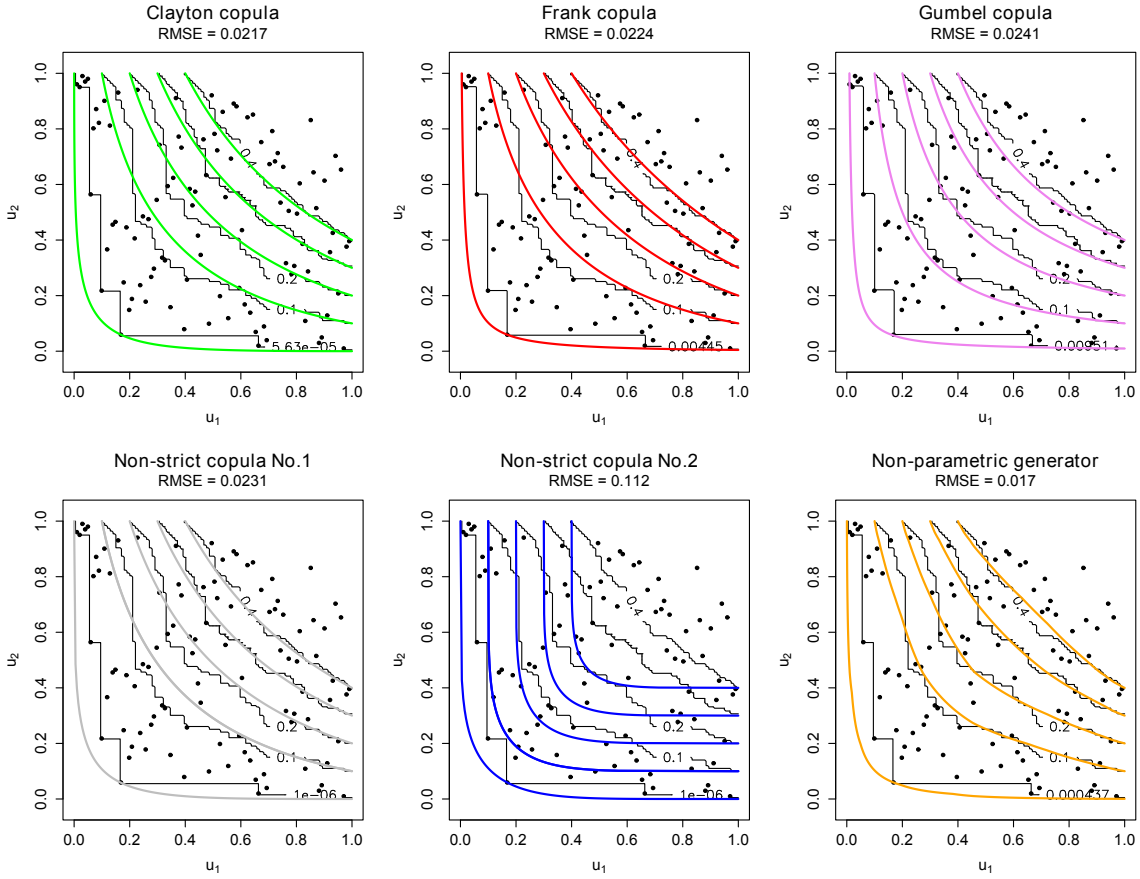


Figure 4.6 – Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data $\mathbf{U}^k$, $k = 1, \ldots, n$, from test problem ZDT1. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

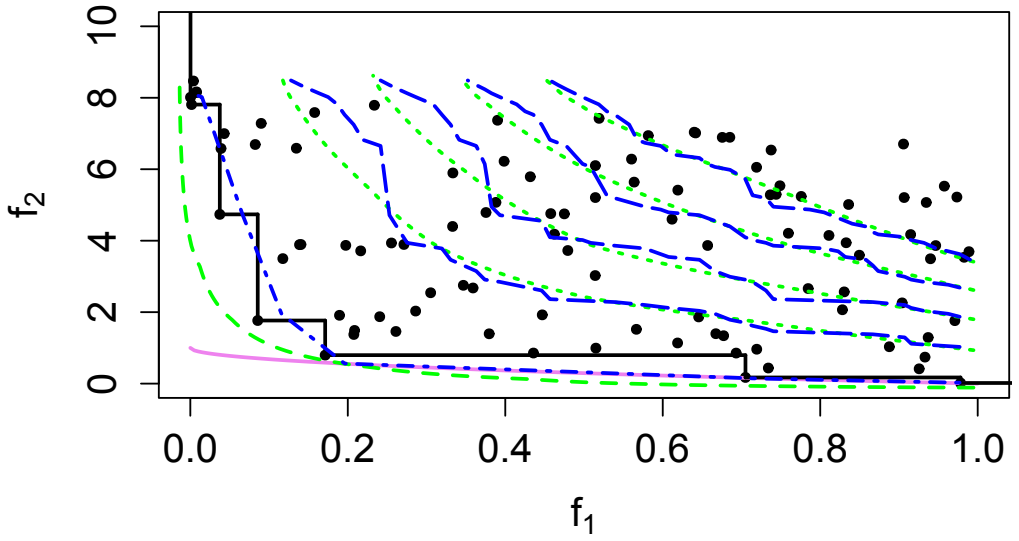


Figure 4.7 – Estimated level line $\partial L^F_{\alpha^*}$ with the best C_ϕ for the ZDT1 test problem (green dashed line), compared to the Pareto front approximation from the observations $\mathcal{P}_n$ (black line), the result with the empirical copula $\hat{C}_n$ (blue dashed-dotted line) and the true Pareto front $\mathcal{P}$ (violet solid line). Other level lines with levels 0.1, 0.2, 0.3 and 0.4 are also displayed with thinner lines.

4.4.2 Estimation of the Pareto front for the ZDT6 test problem

The second test problem has a concave Pareto front and is harder due to a very low density around the Pareto optimal area:

ZDT6. Let $\mathbf{x} \in [0,1]^d$ and $g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^{d} \frac{x_i}{i}\right)^{1/4}$. Consider:

$$f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1), \qquad f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})}\right)^2\right).$$

Again, we choose $d = 2$ and we draw a sample of size $n = 100$ uniformly in $[0,1]^2$, giving observations farther away from the true Pareto front.

This time kernel-based estimation gives the best fit of the marginal distributions, see Figure 4.8. The best copula model is given by the non-parametric copula model, see Figure 4.9. Here again only the non-strict model No.2 is clearly not relevant. For all the models the level lines with α^* closely approximate the corresponding level line of the empirical copula, indicating that the Archimedean hypothesis is acceptable. Note that the lowest RMSE error is also in this case obtained with the non-parametric generator, taking advantage of the higher flexibility offered by this option.

The estimation of the position of the Pareto front is presented in Figure 4.10, showing that

the model can extend the information of the extremal observations to improve estimation in the center of the attainable image space, where no observations are available. In this case, knowing the range of the objectives, for instance by minimizing each objective separately would help selecting the best estimation of marginals.

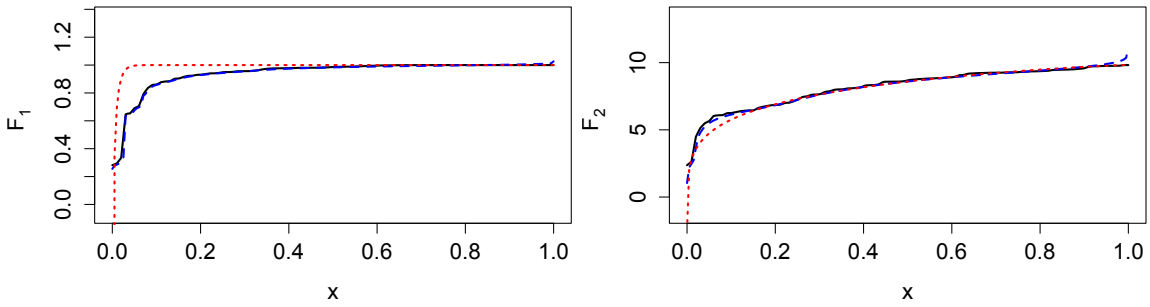


Figure 4.8 – ZDT6 test problem: comparison between three estimation methods of the marginals F_1 and F_2 – empirical (black solid line), kernel density (blue dashed line) and fit of a generalized beta distribution (red dotted line) – for the objectives f_1 (left) and f_2 (right).

4.4.3 Estimation of the Pareto front for the Poloni test problem

This last problem has a disconnected Pareto front with concave and convex parts. A mathematical description of the problem can be found in [PGOP00].

The estimation of marginals suggest the use of non-parametric estimation for f_1 and the estimation from the catalog of beta distribution for f_2, as visible in Figure 4.11. Concerning the copula model, it appears in Figure 4.12 that the level lines α^* from the Archimedean models do not approximate well the shape of the Pareto front. In particular, the lowest level of the empirical level lines is highly non-symmetric. Thus we discard the Archimedean assumption and we keep the empirical copula, getting the extrapolation from the marginals.

The estimation of the position of the Pareto front is shown in Figure 4.13, showing that the proposed approach is also suited when the Archimedean model hypothesis does not hold. Even if the approximation cannot be improved on the lowest part of the Pareto front due to the absence of observations in this area, it effectively gives a better estimation of the Pareto front in the other parts.

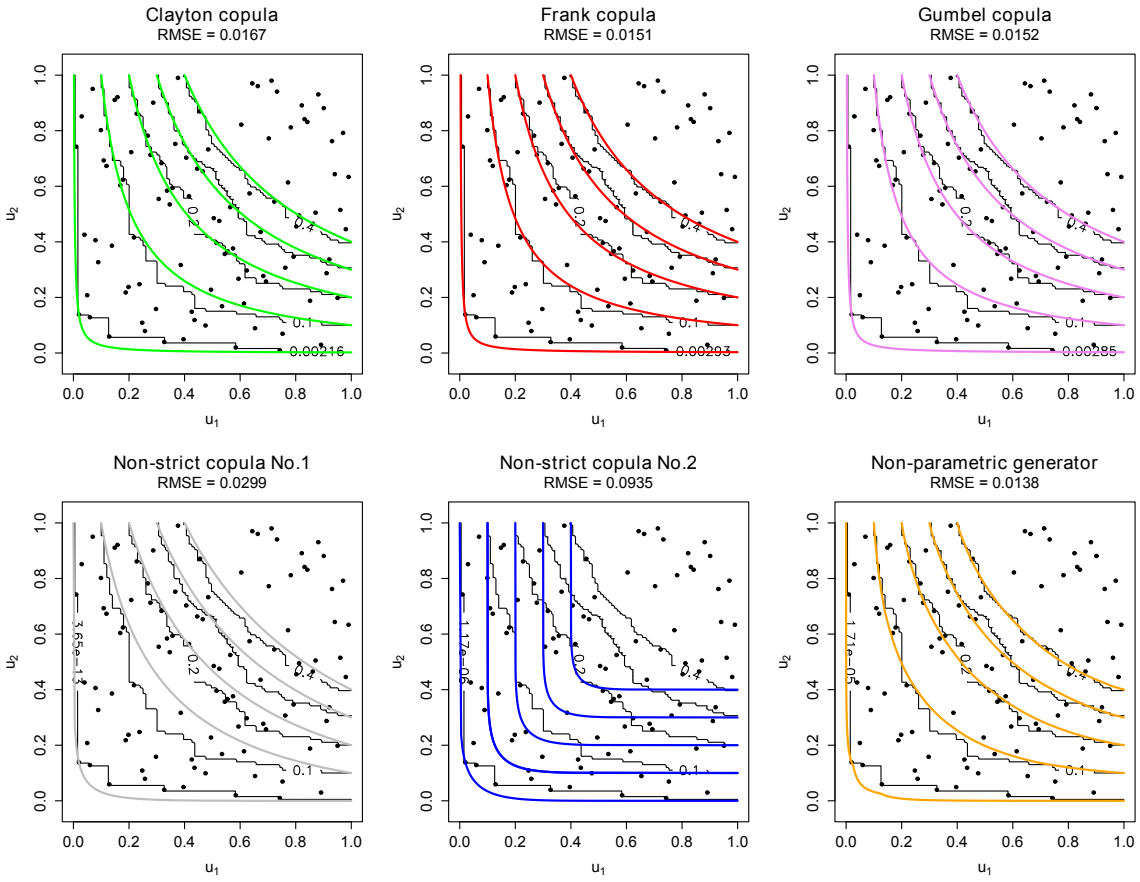


Figure 4.9 – Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data $\mathbf{U}^k$, $k = 1, \ldots, n$, from test problem ZDT6. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

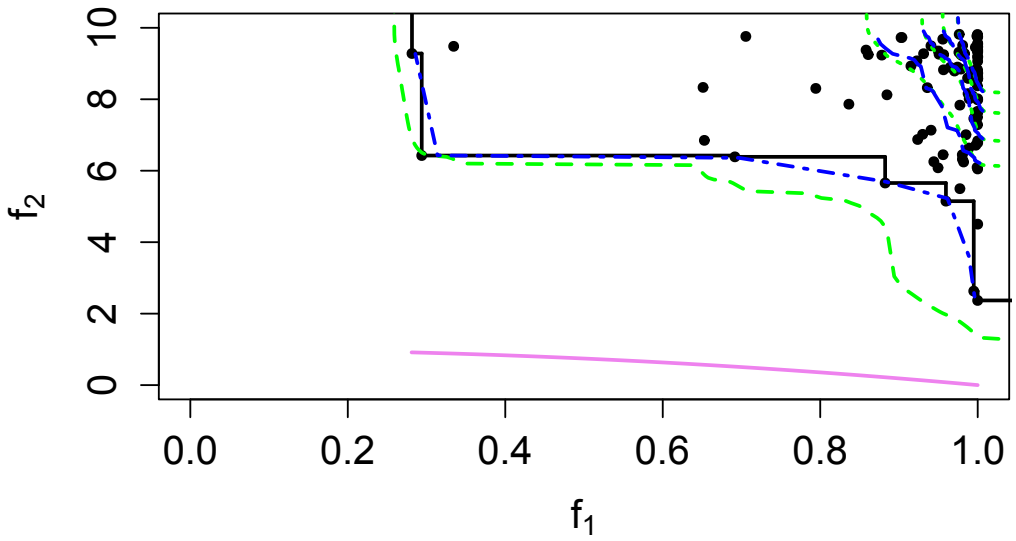


Figure 4.10 – Estimated level line $\partial L_{\alpha^*}^F$ with the best C_ϕ for the ZDT6 test problem (green dashed line), compared to the Pareto front approximation from the observations $\mathcal{P}_n$ (black line), the result with the empirical copula $\hat{C}_n$ (blue dashed-dotted line) and the true Pareto front $\mathcal{P}$ (violet solid line). Other level lines with levels 0.1, 0.2, 0.3 and 0.4 are also displayed with thinner lines.

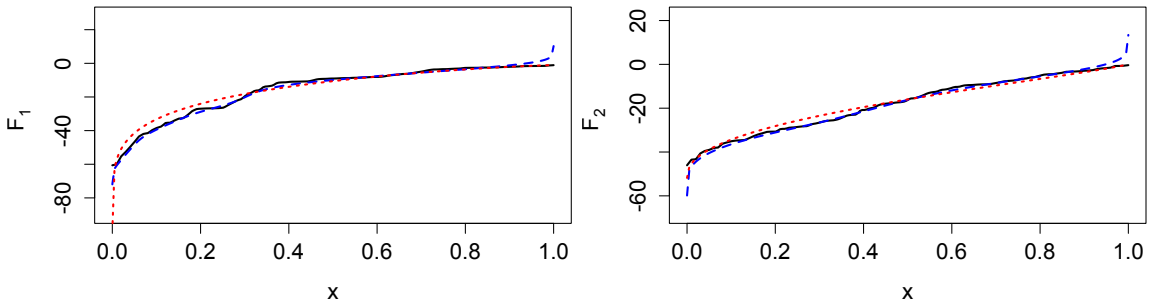


Figure 4.11 – Poloni test problem: comparison between three estimation methods of the marginals F_1 and F_2 – empirical (black solid line), kernel density (blue dashed line) and fit of a generalized beta distribution (red dotted line) – for the objectives f_1 (left) and f_2 (right).

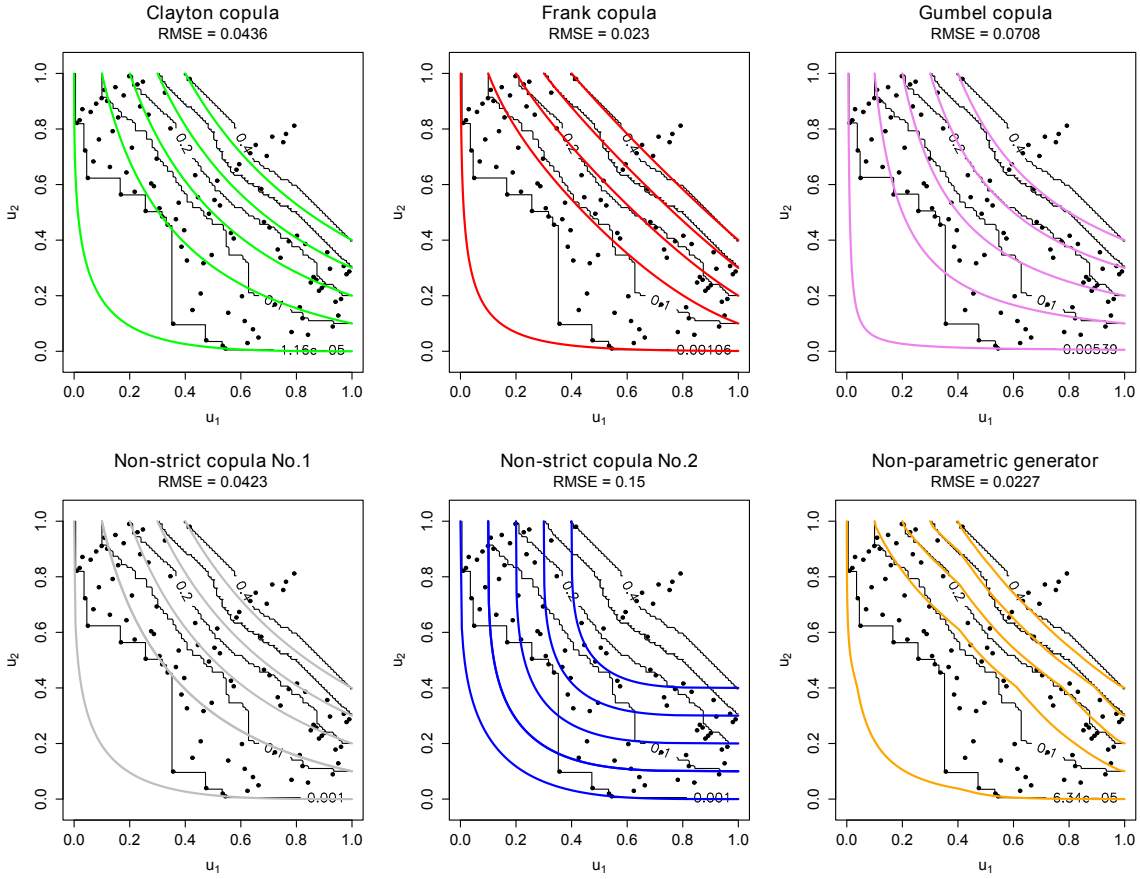


Figure 4.12 – Levels lines $\partial L_\alpha^{C_\phi}$ of the different fitted Archimedean models based on the pseudo-data $\mathbf{U}^k$, $k = 1, \ldots, n$, from test problem Poloni. The level lines correspond in each case to α^*, 0.1, 0.2, 0.3 and 0.4.

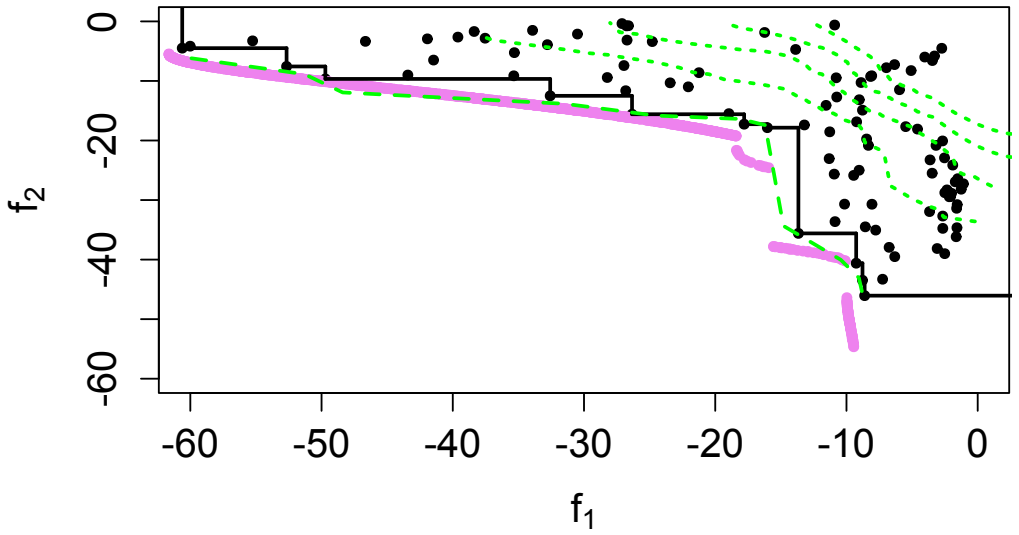


Figure 4.13 – Estimated level line $\partial L^F_{\alpha^*}$ for the Poloni test problem (green dashed line), compared to the Pareto front approximation from the observation $\mathcal{P}_n$ (black line) and the true Pareto front $\mathcal{P}$ (violet solid line). Other level lines with levels 0.1, 0.2, 0.3 and 0.4 are also displayed with thinner lines.

4.5 Conclusions and perspectives

In this paper, we addressed the problem of estimating the Pareto front in an initial phase of multi-objective problems when an i.i.d. sample is available.
At the theoretical level, we established a connection between Pareto fronts and upper level lines of the outputs sample. The approximation of these level lines can be done with very few natural assumptions by using the theory of copulas. An interesting particular case is for Archimedean copulas, for which analytical expressions are available. This assumption can be checked visually or statistically with specific tests of the literature.
The benefits of this methodology are illustrated on some common bi-objective problems from multi-objective optimization literature.

There are several perspectives of this research. Though the Archimedean assumption corresponds to a large range of copulas, it is sometimes inappropriate. As an intermediate solution to the general alternative proposed here – i.e. usage of empirical copula –, it may be interesting to consider nested Archimedean copulas see e.g. [HM11] and references therein, or other families of copulas. Further developments about non-strict generators have also been evocated in Section 4.3.
Secondly, the restriction to i.i.d. samples, while discarding most optimization procedures, still allows random search, which in some cases may perform relatively well – see e.g. [BB12] on hyperparameter optimization – and has known convergence properties [LZ11]. However, it might be possible to extend the approach of [Hüs10] to deal with non independent obser-

vations.

Finally, this methodology relies on the estimated distribution of the outputs. In the context of time-consuming objective functions, such estimation could be improved by using surrogate models.

Post-publication addendum

After the release of this article, further work, described in Appendix C, has been performed accounting for the expensive black-box functions case. In particular, a discussion of the relative merits of this approach with copulas and the one with GP conditional simulations is performed. Then, the application of the methodology to surrogate models instead of the real functions is discussed, which is much more computationally affordable that the one with conditional simulations.

Part III

Contributions to high-dimensional Bayesian optimization

Chapter 5

Bayesian optimization in high-dimension with random embeddings

The methods presented until now are generally applied on problems with relatively few variables, up to few dozens, see e.g. [JSW98], [VJFK11], [SFT+12], [PWG12], [SLA12], [Wag13], [CH14], [LZG14]. Extending the scope of Gaussian process-based optimization in terms of dimensionality of inputs is one of the main contemporary challenges faced by this class of methods, see e.g. [VSBT14]. We present results towards making these algorithms cope with high-dimensional search spaces, under the hypothesis that only a small number of variables are actually influential. Inspired by the Random Embedding Bayesian Optimization (REMBO) approach [WZH+13], we propose to integrate a warping of the high dimensional subspace within the covariance kernel. The proposed warping, that relies on elementary geometric considerations, allows mitigating the drawbacks of the high extrinsic dimensionality while preventing the algorithm to evaluate points giving redundant information. It also alleviates constraints on bound selection for the embedded domain, thus improving the robustness of the algorithm, as illustrated with a test case with 25 variables and intrinsic dimension 6.

Part of this chapter has been published recently in Lecture Notes in Computer Science [BGR15b]. An extended introduction to problems related to many variables as well as a more precise description of REMBO replace the original beginning of the article. In addition, some complementary experiments are provided in Appendix D as well as an alternative interpretation of the warping.

5.1 Challenge of high dimensionality and related works

The root of the difficulty with many variables is that the number of observations required to learn a function when using standard kernels, e.g. without sparsity assumption on f, increases exponentially with the dimension. This is not specific to Kriging and it is shared with other meta-modeling approaches [FSK08], [KCL11], if no structural assumption on the black-box is made. In a broader setting than kernel methods, this is referred to as the "curse of dimensionality", see e.g. [Don00], [HTFF05]. Here we describe briefly some of the techniques to handle expensive high-dimensional black-box functions, say with hundreds or thousands of variables. A detailed survey may be found e.g. in [SW10].

Selecting few variables based on their contribution to the variance is a rather natural idea to get back to a moderate search space. In [SK07], the authors select seven out of forty variables based on an initial metamodel taking into account all the variables, with variable screening and analysis of variance. The risk is to discard variables that are not influential in most of the design space but are critical for the optimum. Also, the phase dedicated to selecting variables may consume a part of the evaluation budget. Another common strategy in dimension reduction is to construct a mapping from the high-dimensional research space to a smaller one, see e.g. [VJFK11] or [LZG14] and references therein. Possible drawbacks are the computational effort required to build the mapping and that, since it depends on previous observations, it may not be suited to extrapolate at unobserved designs (a trait which is common with most methods).

Working on the modeling is another angle of attack. For instance, when using an anisotropic Gaussian kernel over the inputs, the length-scales hyperparameters may be used to perform an Automatic Relevance Determination since they reflect the influence of each input [Nea96], [RW06]. The rationale is then to remove the variables with highest estimated length-scales. A similar idea is exploited in [CCK12] to select only the relevant variables. Inspired by additive models in regression, additive GP models (in the sense of sum of univariate GP) have also been studied, see [DGR12], [Dur11], [Duv14] and references therein. They are very powerful if the problem is additive since they have a linear learning rate with respect to the problem dimensionality. Unfortunately, they may not be perfectly suited to optimization since the variance is possibly zero at unobserved locations. For more flexibility, FANOVA models [MRCK12], described further in Chapter 8, additionally allow to include some interactions between variables, and to perform optimization [IK14]. ANOVA kernels [GRS+14], [CLG15] are also quite promising for building sparse models.

Lastly, recent methods to cope with possibly very high dimensional spaces (up to dimen-

sion one billion) suppose that the black-box function is only varying along a low dimensional subspace, possibly not aligned with the canonical basis. One example is [DKC13], where a low rank matrix is learned to get the low dimensional subspace before optimization, within a cumulative regret bound settings, i.e. the sum of the differences between the observation values and the true minimum. Another method to recover the low dimensional linear structure is proposed in [GOH13] but not with an optimization purpose. Also, with few unknown active variables, [CM12] proposes to combine compressed sensing with linear bandits to optimize efficiently.

In most of the above references, a significant part of the budget is dedicated to uncover the structure of the black-box, which may impact optimization with very scarce budgets. The alternative proposed in [WZH+13] is to simply discard the problem of finding the low dimensional structure and optimize on a randomly selected subspace instead. We will focus especially on this case, which we found to be the most attractive in our settings.

5.2 Random EMbedding Bayesian Optimization

We present here the REMBO (Random EMbedding Bayesian Optimization) method described in [WZH+13] for mono-objective optimization and recall the main definitions and results. Denote $\mathcal{X} \subset \mathbb{R}^D$ the input domain and $f : \mathcal{X} \to \mathbb{R}$ the function to optimize. It is supposed that f is of *effective dimensionality* $d_e \ll D$, see Definition 5.2.1.

Definition 5.2.1 (Effective dimensionality [WZH+13])**.** *A function $f : \mathbb{R}^D \to \mathbb{R}$ is of effective dimensionality $d_e < D$ if there exists a linear subspace $\mathcal{T} \subset \mathbb{R}^{d_e}$ such that $\forall \boldsymbol{x}_\top \in \mathcal{T} \subset \mathbb{R}^D$ and $\boldsymbol{x}_\perp \in \mathcal{T}^\perp \subset \mathbb{R}^D$, $f(\boldsymbol{x}) = f(\boldsymbol{x}_\top + \boldsymbol{x}_\perp) = f(\boldsymbol{x}_\top)$ where $\mathcal{T}$ and $\mathcal{T}^\perp$ are the so-called* effective subspace *and* constant subspace*, i.e. the orthogonal complement of $\mathcal{T}$ in $\mathbb{R}^D$.*

In sensitivity analysis or in applications, it is commonly shown that in fact only a small number of variables are important, see e.g. [JSW98], [SK07], [IL15] or examples given in [WZH+13]. It coincides with the concept of effective dimensionality if the influential variables (or a linear combination of them) concentrate all the influence on the output. From this statement, the principle of REMBO is to map a smaller-dimensional domain $\mathcal{Y} \subset \mathbb{R}^d$ onto $\mathcal{X}$ using a random matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$ with independent $\mathcal{N}(0, 1)$ entries, where $d_e \leq d \ll D$. If $\mathcal{X}$ is $\mathbb{R}^D$, then an optimal solution can be found with probability 1 in $\mathbb{R}^d$, see Theorem 5.2.1.

Theorem 5.2.1. *(Theorem 2 in [WZH+13]). Suppose that $f : \mathbb{R}^D \to \mathbb{R}$ is of effective dimensionality d_e and that $\mathbf{A} \in \mathbb{R}^{D \times d}$ is sampled with independent $\mathcal{N}(0, 1)$ entries where $d \geq d_e$. Then with probability 1, $\forall \boldsymbol{x} \in \mathbb{R}^D$, $\exists \boldsymbol{y} \in \mathbb{R}^d$ such that $f(\boldsymbol{x}) = f(\mathbf{A}\boldsymbol{y})$.*

Sketch of proof. See [WZH+13] for a detailed proof. In particular, for all $\mathbf{x} \in \mathbb{R}^D$, decomposing between $\mathcal{T}$ and $\mathcal{T}^\perp$ gives $\mathbf{x} = \mathbf{x}_\top + \mathbf{x}_\perp$. Then it is possible to find $\mathbf{y} \in \mathbb{R}^d$ such that $\mathbf{A}\mathbf{y} = \mathbf{x}_\top + \mathbf{x}'$ with $\mathbf{x}' \in \mathcal{T}^\perp$ if $\mathbf{A}$ is of rank d_e, which occurs with probability 1. The result follows since $f(\mathbf{x}_\top + \mathbf{x}') = f(\mathbf{x}_\top) = f(\mathbf{x})$. □

In such case, one would simply optimize over $\mathbb{R}^d$ instead of $\mathbb{R}^D$. However, most optimization problems are defined on bounded domains, mostly with box constraints on the variables. Accordingly, in the rest it is supposed that $\mathcal{X} = [-1, 1]^D$, possibly obtained via rescaling. These box constraints are enforced with the convex projection onto $\mathcal{X}$: $\mathbb{R}^D \to \mathbb{R}^D$, $\mathbf{u} \to p_\mathcal{X}(\mathbf{u}) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{u}\|_2$, which can be computed simply in this case since it amounts to take $\max(-1, \min(1, \mathbf{A}_i\mathbf{y}))$, $1 \leq i \leq D$. As a result of the box constraints, a solution is to be found with probability 1 only if $\mathcal{T}$ is the span of d_e basis vectors. This is one part of the following Theorem 5.2.2, along with a result to define $\mathcal{Y}$.

Theorem 5.2.2. *(Theorem 3 in [WZH+13]). Suppose that $f : \mathcal{X} = [-1, 1]^D \subset \mathbb{R}^D \to \mathbb{R}$ is of effective dimensionality d_e, whose effective subspace $\mathcal{T}$ is the span of d_e basis vectors of $\mathbb{R}^D$ and where $\mathcal{X}$ is a centered box domain. Denote $\boldsymbol{x}^*_\top \in \mathcal{T} \cap \mathcal{X}$ a minimizer of f. If $\mathbf{A} \in \mathbb{R}^{D\times d}$ is sampled with independent $\mathcal{N}(0, 1)$ entries, $d \geq d_e$, then $\exists \boldsymbol{y}^* \in \mathbb{R}^d$ such that $f(\mathbf{A}\boldsymbol{y}^*) = f(\boldsymbol{x}^*_\top)$ with probability 1. In addition, $\|\boldsymbol{y}^*\|_d \leq \frac{\sqrt{d_e}}{\varepsilon}\|\boldsymbol{x}^*_\top\|_D$ with probability $1 - \varepsilon$.*

Sketch of proof. See [WZH+13] for a detailed proof. The existence of $\mathbf{x}^*_\top \in \mathcal{T} \cap \mathcal{X}$ is only granted if $\mathcal{T}$ is the span of d_e canonical basis vectors. Then as in Theorem 5.2.1, a solution can be found with probability 1 on $\mathbb{R}^d$. The rest follows from a result on the norm of the inverse of random matrices with i.i.d. $\mathcal{N}(0, 1)$ entries [SST06]. □

The concept is illustrated in Figure 5.1a with $d = 1$ and $D = 2$: the search of the optimum is restricted to the red line instead of the whole domain. A counterexample with a non-aligned influential subspace is provided in Figure 5.1b, where the optimum cannot be found on the drawn embedding. Note that the authors showed empirically, on a rotated Branin-Hoo function, that REMBO also works when the effective subspace is not the span of d_e variables. Arguably, this may be due to the presence of three global optima, which are not on the boundary of the domain.

It remains to find a solution in the low dimensional domain. To this end, Bayesian optimization is applied, e.g. with the Expected Improvement [JSW98], to optimize the low-dimensional function $g : \mathcal{Y} \to \mathbb{R}$, $g(\mathbf{y}) = f(\mathbf{u}(\mathbf{y}))$ with $\mathbf{u} : \mathcal{Y} \to \mathcal{X}$, $\mathbf{u}(\mathbf{y}) = p_\mathcal{X}(\mathbf{A}\mathbf{y})$. Two Gaussian kernels with length scales l were proposed to build the GP models: the first one is $k_\mathcal{Y}(\mathbf{y}, \mathbf{y}') = \exp\left(-\|\mathbf{y} - \mathbf{y}'\|_d^2/2l^2\right)$ and the second one, using the non-linear mapping (or *warping* [RW06]) $\mathbf{u}$, is $k_\mathcal{X}(\mathbf{y}, \mathbf{y}') = \exp(-\|\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{y}')\|_D^2/2l^2)$. The standard REMBO algorithm

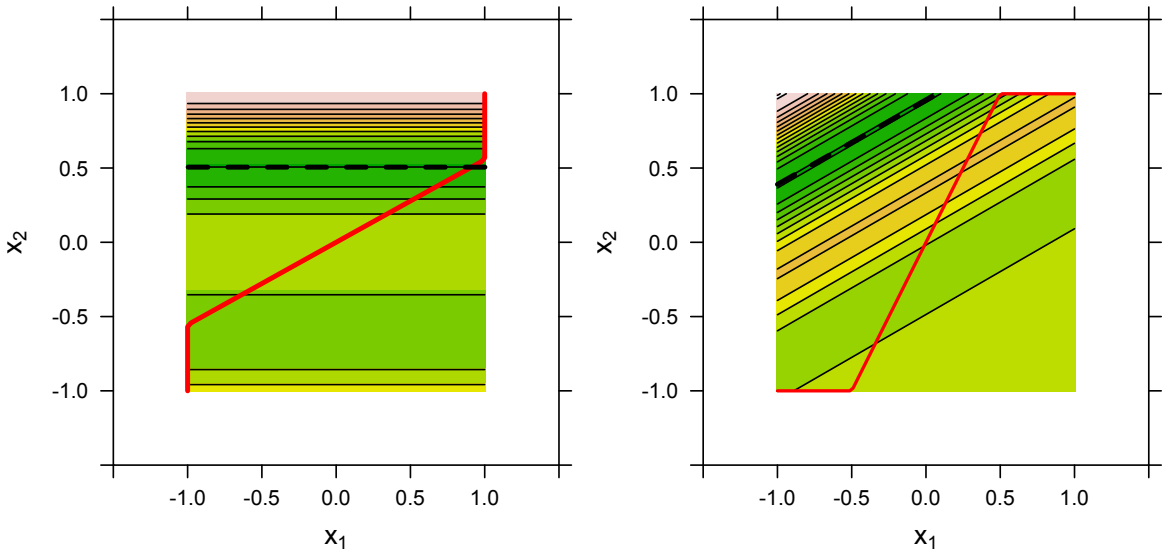


Figure 5.1 – Left: contour plot of a bi-variable function depending on x_2 only, the optimum value is reached on the black dotted line. Right: same but with a function varying along a rotated influential subspace.

is sketched in Algorithm 7. Selecting the small dimension d may follow previous knowledge on the problem at hand, with results from sensitivity analysis for instance or hypothesis on the behavior of the function as in Chapter 8. Otherwise, as preconized in [GOH13], d should be chosen based on the budget of evaluations at hand and computational constraints.

Algorithm 7 Summary of the standard REMBO procedure.

1: Select d empirically or based on previous knowledge.
2: Sample $\mathbf{A} \in \mathbb{R}^{D\times d}$ with independent standard Gaussian coefficients.
3: Set $g : \mathcal{Y} = [-\sqrt{d}, \sqrt{d}]^d \mapsto \mathbb{R}$, $g(\mathbf{y}) = f(\mathbf{Ay})$.
4: Build the surrogate model, either with kernel $k_{\mathcal{Y}}$ or $k_{\mathcal{X}}$.
5: **while** time/evaluation budget not exhausted
6: Optimize the acquisition function (Expected Improvement).
7: Evaluate the objective function at the corresponding design point.
8: Update the model with the new observation and, every few iterations, the hyperparameters.
9: **endwhile**

Other practical concerns arise from the selection of the low dimensional domain $\mathcal{Y}$, which is the focus of the following Chapter 6. In case $\mathcal{X} = [-1, 1]^D$, with probability $1 - \varepsilon$, $\|\mathbf{y}^*\|_d \leq d_e/\varepsilon$. But setting a low ε results in quite a large domain, with in addition problems of injectivity: distant points in $\mathcal{Y}$ may coincide in $\mathcal{X}$, especially far from the center, so that using $k_{\mathcal{Y}}$ leads to sample useless new points in $\mathcal{Y}$ corresponding to the same location in $\mathcal{X}$ after the convex projection. On the other hand, $k_{\mathcal{X}}$ suffers from the curse of dimensionality when $\mathcal{Y}$ is large enough so that most or all of the points of $\mathcal{X}$ belonging to the convex projection of the

subspace spanned by $\mathbf{A}$ onto $\mathcal{X}$ have at least one pre-image in $\mathcal{Y}$. Indeed, whereas embedded points $p_{\mathcal{X}}(\mathbf{Ay})$ lie in a d dimensional subspace when they are inside of $\mathcal{X}$, they belong to a D-dimensional domain when they are projected onto the faces and edges of $\mathcal{X}$. To alleviate these shortcomings, the authors of [WZH+13] suggest to set $\mathcal{Y} = [-\sqrt{d}, \sqrt{d}]^d$. In practice, they split the evaluation budget over several random embeddings or set $d > d_e$ to increase the probability for the optimum to actually be inside $\mathcal{Y}$, slowing down the convergence.

5.3 Proposed kernel and experimental results

Both $k_{\mathcal{Y}}$ and $k_{\mathcal{X}}$ suffering from limitations, it is desirable to have a kernel that retains as much as possible of the actual high dimensional distances between points while remaining of low dimension. This can be achieved by first orthogonally projecting $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ (the image of the low-dimensional space by embedding and convex projection) to the subspace spanned by $\mathbf{A}$: $\mathrm{Ran}(\mathbf{A})$, with $p_{\mathbf{A}} : \mathcal{X} \mapsto \mathbb{R}^D$, $p_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}$. Note that these back-projections can be outside of $\mathcal{X}$ for points on faces of $\mathcal{X}$. The calculation of the projection matrix is done only once, inverting a $d \times d$ matrix. This solves the problem of adding already evaluated points: their back-projections coincide. Nevertheless, distant points on the sides of $\mathcal{X}$ from the convex projection can be back-projected close to each other, which may cause troubles with the stationary kernels classically used.

The next step is to respect as much as possible distances on the border of $\mathcal{X}$, denoted $\partial\mathcal{X}$. Unfolding and parametrizing the manifold corresponding to the convex projection of the embedding of $\mathcal{Y}$ with $\mathbf{A}$ would be best but unfortunately it seems intractable with high D. Indeed, it amounts to finding each intersection of the d-dimensional subspace spanned by $\mathbf{A}$ with the faces of the D-hypercube, before describing the parts resulting from the convex projection. Alternatively, we propose to distort the back-projections which are outside of $\mathcal{X}$, corresponding to those convex-projected parts on the sides of $\partial\mathcal{X}$. In more details, from the back-projection of the initial mapping with $p_{\mathcal{X}}$, a pivot point is selected as the intersection between $\partial\mathcal{X}$ and the line $(O; p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{A}y))$. Then the back-projection is stretched out such that the distance between the pivot point and the initial convex projection are equal. It results in respecting the distance *on the embedding* between the center O and the initial convex projection. The resulting warping, denoted Ψ, is detailed in Algorithm 8 and illustrated in Figure 5.2. Based on this, any positive definite kernel k on $\mathcal{Y}$ can be used. For example, the resulting SE kernel is $k_\Psi(\mathbf{y}, \mathbf{y'}) = \exp\left(-\|\Psi(\mathbf{y}) - \Psi(\mathbf{y}')\|_D^2 / 2l_\Psi^2\right)$. Note that the function value corresponding to $\Psi(\mathbf{y})$ remains $g(\mathbf{y})$.

Like $k_{\mathcal{X}}$, k_Ψ is not hindered by the non-injectivity brought by the convex projection $p_{\mathcal{X}}$. Furthermore, it can explore sides of the hypercube without spending too much budget since

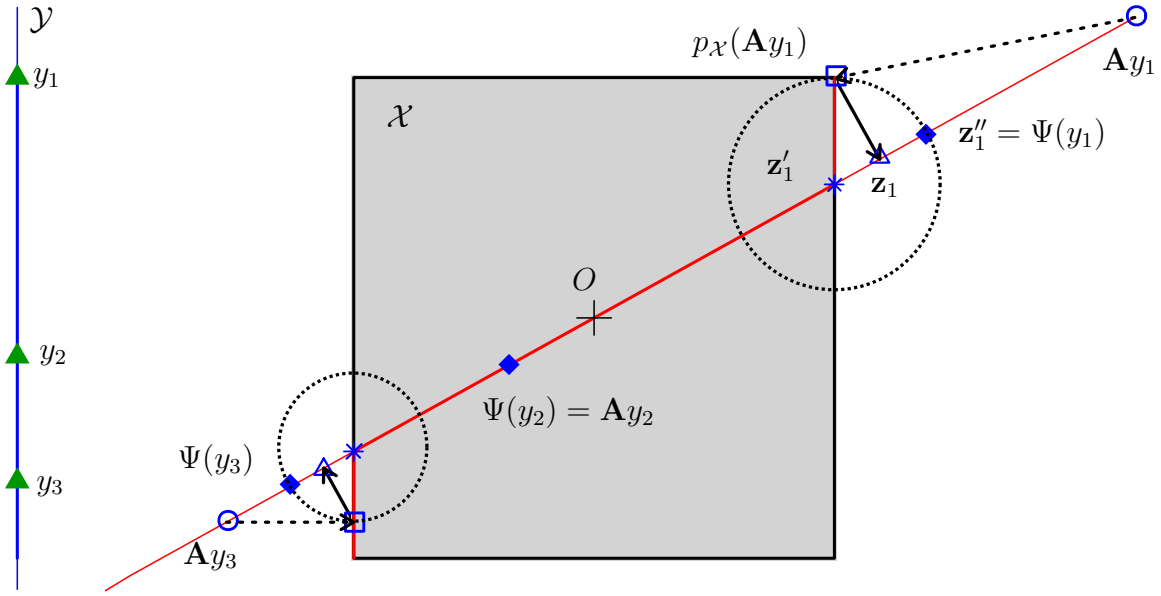


Figure 5.2 – Illustration of the new warping Ψ , $d = 1$ and $D = 2$, from triangles in $\mathcal{Y}$ to diamonds in $\mathcal{X}$, on three points y_1, y_2, y_3. As for REMBO, the points y_i are first mapped by $\mathbf{A}$ and convexly projected onto $\mathcal{X}$ (if out of $\mathcal{X}$). If the resulting image is strictly contained in $\mathcal{X}$ – as for y_2 – nothing else is done. Otherwise, the new warping is defined in two supplementary steps: back-projection onto $\mathrm{Ran}(\mathbf{A})$ (giving $\mathbf{z}_i$) and stretching out in the resulting line $[0, \mathbf{z}_i)$ (red solid line) by reporting the distance between the intersection of $[0, \mathbf{z}_i]$ on the frontier of $\mathcal{X}$, $\mathbf{z}_i'$, and the initial convex projection $p_{\mathcal{X}}(\mathbf{A}y_i)$. The points y_1 and y_3 correspond to cases where such projections are on a corner or a face of $\mathcal{X}$.

Algorithm 8 Calculation of Ψ.

Map $\mathbf{y} \in \mathcal{Y}$ to $\mathbf{Ay}$
If $\mathbf{Ay} \in \mathcal{X}$ **Then**
    Define $\Psi(\mathbf{y}) = \mathbf{Ay}$
Else
    Project onto $\mathcal{X}$ and back-project onto $\mathrm{Ran}(\mathbf{A})$: $\mathbf{z} = p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{Ay}))$
    Compute the intersection of $[O; \mathbf{z}]$ with $\partial\mathcal{X}$: $\mathbf{z}' = (\max_{i=1,\dots,D} |z_i|)^{-1}\mathbf{z}$
    Define $\Psi(\mathbf{y}) = \mathbf{z}' + \|p_{\mathcal{X}}(\mathbf{A}y) - \mathbf{z}'\|_D . \frac{\mathbf{z}'}{\|\mathbf{z}'\|_D}$
EndIf

belonging to Ran($\mathbf{A}$) (all distances between embedded points after warping are d-dimensional instead of D-dimensional, thus smaller, hence limiting the risk of over-exploring sides of $\mathcal{X}$). It is thus possible to extend the size of $\mathcal{Y}$ to avoid the risk of missing the optimum. For instance, one can check that $\mathcal{Y}$ is larger than $[-\gamma, \gamma]^d$ with γ such that $\gamma^{-1} = \min_{j \in [1,\dots,D]} \sum_{i=1}^{d} |A_{j,i}|$, with $A_{j,i}$ the components of $\mathbf{A}$, ensuring to span $[-1, 1]$ for each of the D variables.

We compare the performances of the usual REMBO method with $k_{\mathcal{Y}}$, $k_{\mathcal{X}}$ and the proposed k_{Ψ}, with a unique embedding. Tests are conducted with the *DiceKriging* and *DiceOptim* packages [RGD12]. We use the isotropic Matérn 5/2 kernel with hyperparameters estimated with Maximum Likelihood and we start optimization with space filling designs of size $10d$. Initial designs are modified such that no points are repeated in $\mathcal{X}$ for $k_{\mathcal{Y}}$ and $k_{\mathcal{X}}$. For k_{Ψ}, we apply Ψ to bigger initial designs, before selecting the right number of points by removing the points that are closest to each other. Experiments are repeated fifty times, taking the same random embeddings for all kernels. To allow a fair comparison, $\mathcal{Y}$ is set to $[-\sqrt{d}, \sqrt{d}]^d$ for all kernels and the computational efforts on the maximization of the Expected Improvement are the same.

Results in Figure 5.3 show that the proposed kernel k_{Ψ} outperforms both $k_{\mathcal{Y}}$ and $k_{\mathcal{X}}$ when $d = 6$. In particular, $k_{\mathcal{Y}}$ loses many evaluations on the sides of $\mathcal{Y}$ for already known points in $\mathcal{X}$ and $k_{\mathcal{X}}$ has a propensity to explore sides of $\mathcal{X}$, while k_{Ψ} avoids both pitfalls.

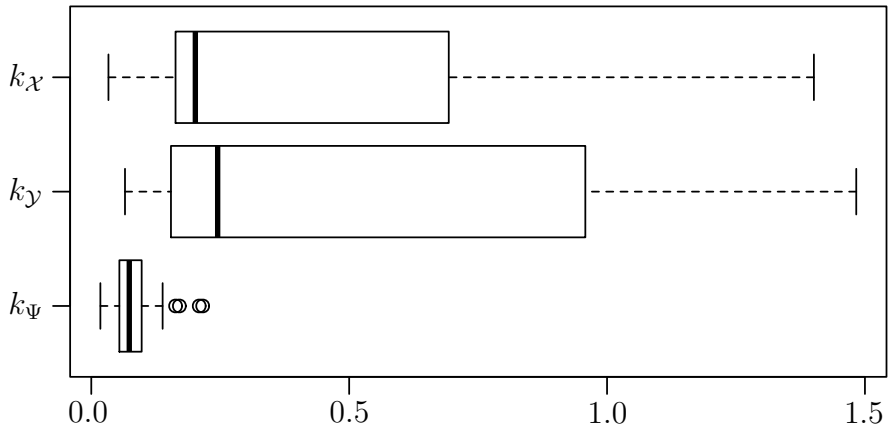


Figure 5.3 – Boxplot of the optimality gap (best value found minus actual minimum) for kernels $k_{\mathcal{X}}$, $k_{\mathcal{Y}}$ and k_{Ψ} on the Hartman6 test function (see e.g. [JSW98]) with 250 evaluations, $d = d_e = 6$, $D = 25$.

5.4 Conclusion and perspectives

The composition with a warping of the covariance kernel used with REMBO wipes out some of the previous shortcomings. It thus achieved the goal of improving the results with a single embedding, as was shown on the Hartman6 example. Studying the efficiency of splitting the evaluation budget between several random embeddings, compared to relying on a single one along with k_Ψ, would be the scope of future research. Of interest is also the study of the embedding itself, such as properties ensuring fast convergence in practice.

Chapter 6

Analysis of the REMBO method toward improved robustness

6.1 Motivations

Let us rewrite the problem tackled in the REMBO method [WZH+13]. We consider $f : \mathcal{X} = [-1, 1]^D \to \mathbb{R}$ and suppose that $f^* = \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ exists. Ultimately we are interested in finding $\mathbf{x}^* \in \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} f(\mathbf{x})$. To this end, the main hypothesis ($\mathcal{H}$) is that f only depends on a set of d variables, whose unknown indices are denoted $I = \{i_1, \ldots, i_d\} \subseteq \{1, \ldots, D\}$[1]. Denote $\mathbf{x}_I = (x_{i_1}, \ldots, x_{i_d})$ and $\mathbf{x}_{-I} = \mathbf{x}_{[1,\ldots,D]\setminus I}$; in fine, up to a permutation of indices[2], $\mathbf{x} = (\mathbf{x}_I, \mathbf{x}_{-I})$. Under ($\mathcal{H}$), there exists a function $h : \mathbb{R}^d \to \mathbb{R}$ such that $\forall \mathbf{x} \in \mathcal{X}, f(\mathbf{x}) = h(\mathbf{x}_I)$. Considering h makes it clear that working in a lower dimensional space is sufficient. One smart way of reducing dimensionality without knowing the d active variables in advance consists in appealing to a linear embedding, $\mathbf{A} : \mathbb{R}^d \to \mathbb{R}^D$, represented (in canonical coordinates) by a $D \times d$ matrix. Indeed, taking also into account box constraints with convex projection on $\mathcal{X}$, i.e. $p_{\mathcal{X}}$, the low dimensional function of interest is $g : \mathcal{Y} \subseteq \mathbb{R}^d \to \mathbb{R}$, $g(\mathbf{y}) = f(p_{\mathcal{X}}(\mathbf{A}\mathbf{y}))$. This leads to the problem considered in REMBO (Theorem 5.2.2 , i.e. Theorem 3 in [WZH+13]), ($\mathcal{R}$):

$$(\mathcal{R}) : \text{ Under } \mathcal{H}, \text{ find } \mathbf{y}^* \in \mathcal{Y} \subseteq \mathbb{R}^d \text{ such that } f(p_{\mathcal{X}}(\mathbf{A}\mathbf{y}^*)) = f^*.$$

Noting that $(p_{\mathcal{X}}(\mathbf{z}))_I = p_{\mathcal{X}_I}(\mathbf{z}_I)$ for all $\mathbf{z} \in \mathbb{R}^D$, this is equivalent to:

$$(\mathcal{R}') : \text{ Under } \mathcal{H}, \text{ find } \mathbf{y}^* \in \mathcal{Y} \subseteq \mathbb{R}^d \text{ such that } h(p_{\mathcal{X}_I}(\mathbf{A}_I\mathbf{y}^*)) = f^*$$

[1] This is stronger than the hypothesis of effective dimensionality, Definition 5.2.1 (Definition 1 in [WZH+13]) which is sufficient if $\mathcal{X} = \mathbb{R}^D$, see Theorem 5.2.1 (Theorem 2 in [WZH+13]).

[2] A permutation, omitted to clarify notations, is assumed such that coordinates are put in the right order when necessary.

where $\mathbf{A}_I$ is the submatrix with rows $i_1, \dots, i_d$ and $\mathcal{X}_I = [-1, 1]^d$.

In particular, if $\mathbf{A}$ is sampled with i.i.d. standard Gaussian components, a solution can be found with probability 1 when $\mathcal{Y} = \mathbb{R}^d$ [WZH+13]. However, for practical reasons, it is preferable to work with a *compact* $\mathcal{Y}$. Fortunately, it was shown in [WZH+13] that it is sufficient for $\mathcal{Y}$ to contain the ball $(\mathbf{0}, d/\varepsilon)$ for this probability to be greater than $1 - \varepsilon$. Heuristically, the authors recommend to limit the search domain by setting $\mathcal{Y} = [-\sqrt{d}, \sqrt{d}]^d$, which is included in the ball with $\varepsilon \leq 1$, arguing that optima are in general not on the boundary of $\mathcal{X}$. There are several arguments to do so. First, as discussed in [WZH+13] and Chapter 5, the kernels $k_{\mathcal{Y}}$ and $k_{\mathcal{X}}$ employed for Bayesian optimization have shortcomings that are more important on large domains, due to non-injectivity and high-dimensionality respectively. Also a larger domain renders optimization harder. As a workaround, strategies consisting respectively in splitting the budget between several instances with different random matrices or increasing the dimensionality of the low-dimensional subspace were proposed in [WZH+13].

In Chapter 5, we proposed a specific kernel for REMBO which does not suffer from the issues of the previous covariance kernels. It thus offers the possibility to define a larger low-dimensional domain $\mathcal{Y}$, which in turn increases the probability of having a solution in $\mathcal{Y}$ without splitting the budget, see Appendix D. Motivated by this, we now address the problem of finding a minimal compact set $\mathcal{Y}$ such that problem $(\mathcal{R})$ has a solution. This is achieved in Section 6.2 with the set $\mathcal{U}$, corresponding to points in $\mathcal{Y}$ whose image in $\mathcal{X}$ have at least d components in]-1,1[, and we discuss how to enclose it within a simple domain $\mathcal{Y}$ in practice.

This setting is in fact a worst case scenario and $\mathcal{Y}$ may become very large and unpractical. Therefore, in Section 6.3, we consider the problem of finding a matrix $\mathbf{A}$ such that $\mathcal{U}$ is, in a certain sense, as small as possible while still ensuring to find a solution to problem $(\mathcal{R})$. As a result, we propose some possible modifications of the matrix $\mathbf{A}$ along with the resulting variations for the REMBO algorithm. They are tested on the same 25-dimensional mono-objective test problem as in Chapter 5.

Motivated by the good performance in mono-objective optimization, we detail in Section 4 an extension of REMBO and its proposed variants to multi-objective optimization, which is later on also applied on an industrial test case in Chapter 8.

6.2 Sets of interest in the low dimensional space $\mathcal{Y}$

The embedding procedure is composed of a linear transformation $(\mathbf{A})$ and a convex projection $(p_{\mathcal{X}})$. As such, the low dimensional space $\mathbb{R}^d$ is divided between half-space intersections

corresponding to components $x_i = \pm 1$ in the high dimensional space $\mathbb{R}^D$. This structure is useful to extract sets of interest. We especially focus on a compact set, $\mathcal{U}$, which is sufficient to find a solution to $(\mathcal{R})$, instead of encompassing the entire d-dimensional space as input domain (i.e. $\mathcal{Y} = \mathbb{R}^d$). Nevertheless, this set is complex and difficult to exhibit and we discuss practical approaches to find simple supersets $\mathcal{Y}$ enclosing it, such as a d-ball or a d-square.

6.2.1 Preliminary definitions and properties

We begin with two elementary properties of the convex projection onto the hypercube $\mathcal{X} = [-1, 1]^D$:

Property 6.2.1 (Tensorization property)**.** $\forall \boldsymbol{x} \in \mathbb{R}^D$, $p_{\mathcal{X}} \begin{pmatrix} x_1 \\ \dots \\ x_D \end{pmatrix} = \begin{pmatrix} p_{[-1,1]}(x_1) \\ \dots \\ p_{[-1,1]}(x_D) \end{pmatrix}$.

Property 6.2.2 (Commutativity with some isometries)**.** *Let q be an isometry represented by a diagonal matrix with terms $\varepsilon_i = \pm 1$, $1 \leq i \leq D$. Then, for all $\boldsymbol{x} \in \mathbb{R}^D$, $p_{\mathcal{X}}(q(\boldsymbol{x})) = \begin{pmatrix} \varepsilon_1 p_{[-1,1]}(x_1) \\ \dots \\ \varepsilon_D p_{[-1,1]}(x_D) \end{pmatrix} = q(p_{\mathcal{X}}(\boldsymbol{x}))$.*

Now consider the low-dimensional space $\mathbb{R}^d$. Denote $H_{\mathbf{a},\delta}$ the hyperplane in $\mathbb{R}^d$ with normal vector $\mathbf{a} \in \mathbb{R}^d$ and offset $\delta \in \mathbb{R}$: $H_{\mathbf{a},\delta} = \left\{ \mathbf{y} \in \mathbb{R}^d \,,\, \langle \mathbf{a}, \mathbf{y} \rangle = \delta \right\}$. Our analysis in the low dimensional space begins by a general definition of what we call *strips*.

Definition 6.2.1. *The set of points between the parallel hyperplanes $H_{\boldsymbol{a},-\delta}$ and $H_{\boldsymbol{a},\delta}$, is called a* strip*, denoted $\mathcal{S}_{\boldsymbol{a},\delta}$: $\mathcal{S}_{\boldsymbol{a},\delta} = \left\{ \boldsymbol{y} \in \mathbb{R}^d \,,\, |\langle \boldsymbol{a}, \boldsymbol{y} \rangle| \leq |\delta| \right\}$.*

Let us now consider hyperplanes normal vectors given by the rows of a matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$ and with fixed $\delta = 1$. The D corresponding strips, simply denoted $\mathcal{S}_i$ for convenience, are given by:

$$\mathcal{S}_i = \left\{ \mathbf{y} \in \mathbb{R}^d, -1 \leq \mathbf{A}_i \mathbf{y} \leq 1 \right\}.$$

The intersection of all strips $\mathcal{S}_i$ is denoted $\mathcal{I}$. It corresponds to the pre-image of $\mathcal{X} \cap \mathrm{Ran}(\mathbf{A})$ by $\mathbf{A}$:

$$\mathcal{I} = \bigcap_{i=1}^{D} \mathcal{S}_i = \left\{ \mathbf{y} \in \mathbb{R}^d, \forall i = 1, \dots, D : -1 \leq \mathbf{A}_i \mathbf{y} \leq 1 \right\} = \left\{ \mathbf{y} \in \mathbb{R}^d \,,\, p_{\mathcal{X}}(\mathbf{A}\mathbf{y}) = \mathbf{A}\mathbf{y} \right\}.$$

Denoting $\mathcal{C}^d$ the d-cube $[-1, 1]^d$ and supposing temporarily that the I indices are known, there is a simple set $\mathcal{Y}$ such that problem $(\mathcal{R})$ has a solution, as shown now.

Lemma 6.2.1. *If $\mathbf{A}_I$ is invertible, then problem $(\mathcal{R})$ admits a solution in $\mathcal{Y} = \mathbf{A}_I^{-1}\mathcal{C}^d =: \mathfrak{P}_I$.*

Proof. Define $\mathbf{y}^* = \mathbf{A}_I^{-1}\mathbf{x}_I^*$. Then by Property 6.2.1:
$f(p_{\mathcal{X}}(\mathbf{A}\mathbf{y}^*)) = h((p_{\mathcal{X}}(\mathbf{A}\mathbf{y}^*))_I) = h(p_{\mathcal{X}_I}(\mathbf{A}_I\mathbf{y}^*)) = h(p_{\mathcal{X}_I}(\mathbf{x}_I^*)) = h(\mathbf{x}_I^*) = f^*$. □

Remark 6.2.1. *Notice that a milder sufficient solution is that $\boldsymbol{x}_I^* \in \mathrm{Ran}(\mathbf{A}_I)$, the subspace spanned by the columns of $\mathbf{A}$. However this of poor practical interest.*

The set of interest in Lemma 6.2.1, $\mathfrak{P}_I = \mathbf{A}_I^{-1}\mathcal{C}^d$, is a *parallelotope*, i.e. a linear transformation of a d-cube in a d-dimensional subspace, see e.g. [LSA+13]. In addition, it directly follows that:

$$\mathfrak{P}_I = \left\{\mathbf{y} \in \mathbb{R}^d, \forall i \in I : -1 \leq \mathbf{A}_i\mathbf{y} \leq 1\right\} = \bigcap_{i \in I} \mathcal{S}_i.$$

6.2.2 A minimal set for problem $(\mathcal{R})$

Since the d influential variables are unknown, Lemma 6.2.1 shows that a solution to $(\mathcal{R})$ exists if all submatrices extracted from rows of $\mathbf{A}$ are invertible (i.e. of rank d). In the following, we denote by $\mathcal{A}$ this class of matrices,

$$\mathcal{A} = \left\{\mathbf{A} \in \mathbb{R}^{D\times d} \text{ such that any } d \times d \text{ extracted submatrix is invertible}\right\}.$$

As the set I of active variables is unknown, a solution is to be found in one of the $\binom{D}{d}$ different parallelotopes $\mathfrak{P}_I$. We thus consider their union, which is referred to as $\mathcal{U}$:

$$\mathcal{U} = \bigcup_{I \subseteq \{1,\dots,D\}, |I|=d} \mathfrak{P}_I$$

where $|I|$ is the size of I. The sets $\mathcal{U}$, $\mathfrak{P}_I$ and $\mathcal{I}$ are illustrated with $d = 2$ in Figure 6.1. On the left figure, strips are marked by lines. In addition, the color of a point corresponds to D minus the dimension of the face it belongs to, i.e. the number of coordinates of its image by $p_{\mathcal{X}}(\mathbf{A}\cdot)$ that are equal to 1 in absolute value. For instance, edges of $\mathcal{X}$ (1-faces) have $D-1$ components fixed to ± 1. This is also D minus the number of strips they belong to.

From this definition, we directly have Proposition 6.2.1 as a direct consequence of Lemma 6.2.1.

Proposition 6.2.1. *If $\mathbf{A} \in \mathcal{A}$, then problem $(\mathcal{R})$ admits a solution in $\mathcal{Y} = \mathcal{U}$.*

In fact we show in the following Proposition that $\mathcal{U}$ is the smallest closed set such that the map $g_{|\mathcal{U}} : \mathcal{U} \to p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$, $\mathbf{y} \mapsto p_{\mathcal{X}}(\mathbf{A}\mathbf{y})$ is surjective.

Proposition 6.2.2. *If $\mathbf{A} \in \mathcal{A}$, then $\mathcal{U}$ is the smallest closed set $\mathcal{Y} \subseteq \mathbb{R}^d$ such that $p_{\mathcal{X}}(\mathbf{A}\mathcal{Y}) = p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$. Furthermore, $\mathcal{U}$ is a compact and star-shaped set with respect to every point in $\mathcal{I}$.*

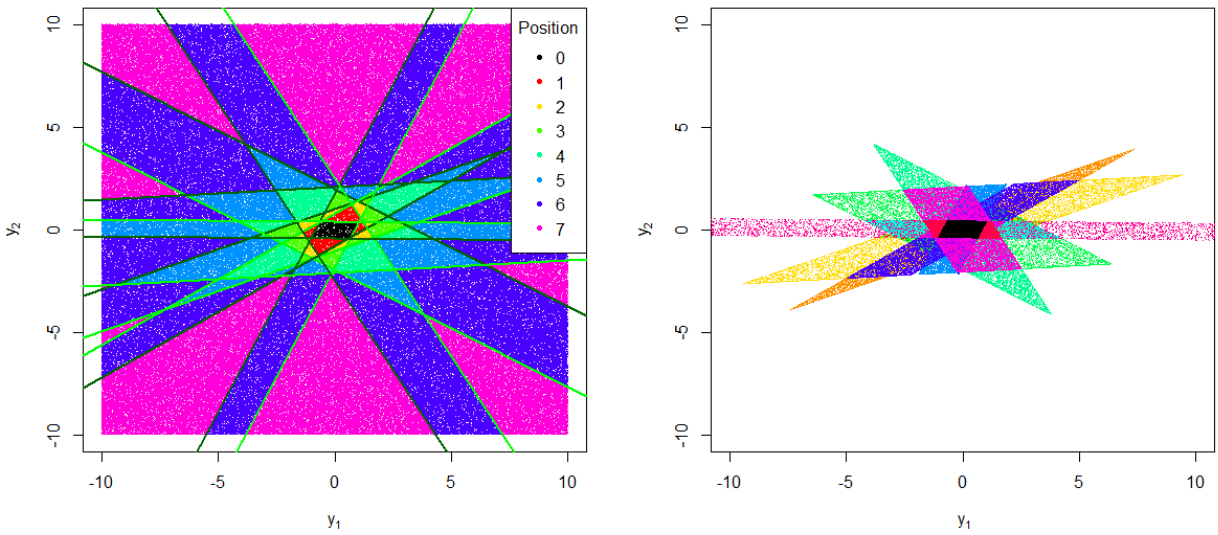


Figure 6.1 – Representation of the sets of interest introduced in Section 6.2 with $d = 2$, $D = 7$. Left: colors represents how many variables in $\mathcal{X}$ are not in $]-1, 1[$. The green lines are the parallel hyperplanes forming the strips $\mathcal{S}_i$. Right: each of the 21 parallelotopes (here parallelograms) $\mathfrak{P}_I$ is depicted with a different color and their union is the minimal set $\mathcal{U}$. The intersection of all parallelograms $\mathcal{I}$ is in black. For illustration purpose, $\mathcal{U}$ is truncated: the violet horizontal parallelogram actually spreads up to approximately $|y_1| = 35$.

Proof. First, note that $\mathcal{U}$ is a closed set as a finite union of closed sets. Then, let us show that $p_{\mathcal{X}}(\mathbf{A}\mathcal{U}) = p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$. Consider $\mathbf{x} \in p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$, hence $|x_i| \leq 1$ and $\exists \mathbf{y} \in \mathbb{R}^d$ s.t. $\mathbf{x} = p_{\mathcal{X}}(\mathbf{A}\mathbf{y})$. Denote $\mathbf{b} = \mathbf{A}\mathbf{y}$. We distinguish two cases:

1. More than d components of $\mathbf{b}$ are in $[-1, 1]$. Then there exists a set I of size d such that $\mathbf{y} \in \bigcap_{i \in I} \mathcal{S}_i = \mathfrak{P}_I \subseteq \mathcal{U}$, implying that $\mathbf{x} \in p_{\mathcal{X}}(\mathbf{A}\mathcal{U})$.

2. $0 \leq k < d$ components of $\mathbf{b}$ are in $[-1, 1]$. It is enough to consider that $\mathbf{b} \in [0, +\infty)^D$. Indeed, for any $\mathbf{x} \in p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$, any $\mathbf{A} \in \mathcal{A}$, let $\boldsymbol{\varepsilon}$ be the isometry given by the diagonal $D \times D$ matrix $\boldsymbol{\varepsilon}$ with elements ± 1 such that $\boldsymbol{\varepsilon}\mathbf{x} \in [0, +\infty)^D$. It follows that $\boldsymbol{\varepsilon}\mathbf{b}$ is in $[0, +\infty)^D$ too. Denote $\mathbf{x}' = \boldsymbol{\varepsilon}\mathbf{x}$, $\mathbf{b}' = \boldsymbol{\varepsilon}\mathbf{b}$ and $\mathbf{A}' = \boldsymbol{\varepsilon}\mathbf{A}$. Thus if $\exists \mathbf{u} \in \mathcal{U}$ such that $\mathbf{x}' = p_{\mathcal{X}}(\mathbf{b}') = p_{\mathcal{X}}(\mathbf{A}'\mathbf{u})$, by property 6.2.2: $\boldsymbol{\varepsilon}\mathbf{x} = \boldsymbol{\varepsilon} p_{\mathcal{X}}(\mathbf{A}\mathbf{u})$ leading to $\mathbf{x} = p_{\mathcal{X}}(\mathbf{b}) = p_{\mathcal{X}}(\mathbf{A}\mathbf{u})$. From now on, we therefore assume that $b_i \geq 0$, $1 \leq i \leq D$. Furthermore, we can assume that $0 \leq b_1 \leq \cdots \leq b_D$, from a permutation of indices. Hence $b_i > 1$ if $i > k$ and $\mathbf{x} = (x_1 = b_1, \ldots, x_k = b_k, 1, \ldots, 1)^T$.

   Let $\mathbf{y}' \in \mathbb{R}^d$ be the solution of $\mathbf{A}_{1,\ldots,d}\mathbf{y}' = (b_1, \ldots b_k, 1, \ldots, 1)^T$ (vector of size d). Such a solution exists since $\mathbf{A}_{1,\ldots,d}$ is invertible by hypothesis. Then define $\mathbf{b}' = \mathbf{A}\mathbf{y}'$, $\mathbf{b}' = (b_1, \ldots, b_k, 1, \ldots, 1, b'_{d+1}, \ldots, b'_D)^T$. We have $\mathbf{b}' \in \mathrm{Ran}(\mathbf{A})$ and $\mathbf{y}' \in \mathfrak{P}_{1,\ldots,d} \subseteq \mathcal{U}$.

- If $\min_{i\in\{d+1,\dots,D\}}(b'_i) \geq 1$, then $p_{\mathcal{X}}(\mathbf{b}') = p_{\mathcal{X}}(\mathbf{b}) = \mathbf{x}$, and thus $\mathbf{x} = p_{\mathcal{X}}(\mathbf{A}\mathbf{y}') \in p_{\mathcal{X}}(\mathbf{A}\mathcal{U})$.
- Else, the set $L = \{i \in \mathbb{N} : d+1 \leq i \leq D\,,\, b'_i < 1\}$ is not empty. Consider $\mathbf{c} = \lambda\mathbf{b}' + (1-\lambda)\mathbf{b}$, $\lambda \in]0,1[$. By linearity, since both $\mathbf{b}$ and $\mathbf{b}'$ belong to $\mathrm{Ran}(\mathbf{A})$, $\mathbf{c} \in \mathrm{Ran}(\mathbf{A})$.
  - For $1 \leq i \leq k$, $c_i = x_i$.
  - For $k+1 \leq i \leq d$, $c_i = \lambda + (1-\lambda)b_i \geq 1$ since $b_i > 1$.
  - For $i \in \{d+1,\dots,D\} \setminus L$, $b'_i \geq 1$ and $b_i > 1$ hence $c_i \geq 1$.
  - We now focus on the remaining components in L. For all $i \in L$, we solve $c_i = 1$, i.e. $\lambda b'_i + (1-\lambda)b_i = \lambda(b'_i - b_i) + b_i = 1$. The solution is $\lambda_i = \frac{b_i - 1}{b_i - b'_i}$, with $b_i - b'_i > 0$ since $b'_i < 1$. Also $b_i - 1 > 0$ and $b_i - 1 < b_i - b'_i$ such that we have $\lambda_i \in]0,1[$. Denote $\lambda^* = \min_{i\in L}\lambda_i$ and the corresponding index i^*. By construction, $c_{i^*} = 1$ and $\forall i \in L$, $c_i = \lambda^*(b'_i - b_i) + b_i \geq \lambda_i(b'_i - b_i) + b_i = 1$ since $\lambda_i \geq \lambda^*$ and $b'_i - b_i < 0$.

  To summarize, we can construct $\mathbf{c}^*$ with λ^* that has $k+1$ components in $[-1,1]$ (the first k and the i^{*th} ones), the others are greater or equal than 1. Moreover, $\mathbf{c}^* \in \mathrm{Ran}(\mathbf{A})$ and fulfills $p_{\mathcal{X}}(\mathbf{c}^*) = p_{\mathcal{X}}(\mathbf{b}) = \mathbf{x}$ by Property 6.2.1. If $k+1 = d$, this corresponds to case 1 above, otherwise, it is possible to reiterate by taking $\mathbf{b} = \mathbf{c}$. Hence we have a pre-image of $\mathbf{x}$ by g in $\mathcal{U}$.

Thus the surjection property is shown. There remains to show that $\mathcal{U}$ is the smallest closed set achieving this, along with additional topological properties.

Let us show that any closed set $\mathcal{Y} \in \mathbb{R}^d$ such that $p_{\mathcal{X}}(\mathbf{A}\mathcal{Y}) = p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ contains $\mathcal{U}$. To this end, we consider $\mathcal{U}^* = \bigcup_{I\subseteq\{1,\dots,D\},|I|=d} \mathring{\mathfrak{P}}_I$ with $\mathring{\mathfrak{P}}_I = \left\{\mathbf{y} \in \mathbb{R}^d\,,\, \forall i \in I, -1 < \mathbf{A}_i\mathbf{y} < 1\right\}$, the interior of the parallelotopes. We have $g_{|\mathring{U}}$ bijective. Indeed, all $\mathbf{x} \in p_{\mathcal{X}}(\mathbf{A}\mathcal{U}^*)$ have (at least) d components whose absolute value is strictly lower that 1. Without loss of generality, we suppose that they are the d first ones, $I = \{1,\dots,d\}$. Then there exists a *unique* $\mathbf{y} \in \mathbb{R}^d$ s.t. $\mathbf{x} = p_{\mathcal{X}}(\mathbf{A}\mathbf{y})$ because $\mathbf{x}_I = (\mathbf{A}\mathbf{y})_I = \mathbf{A}_I\mathbf{y}$ has a unique solution with $\mathbf{A}_I$ is invertible. Since $\mathcal{Y}$ is in surjection with $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ for $g_{|\mathcal{Y}}$ and $g_{|\mathcal{U}^*}$ is bijective, $\mathcal{U}^* \subseteq \mathcal{Y}$. Additionally, $\mathcal{Y}$ is a closed set so it must contain the closure $\mathcal{U}$ of $\mathcal{U}^*$.

Finally let us prove the topological properties of $\mathcal{U}$. Recall that parallelotopes $\mathfrak{P}_I$ ($I \subseteq \{1,\dots,D\}$) are compact convex sets as linear transformations of d-cubes. Thus $\mathcal{I} = \bigcap_{I\subseteq\{1,\dots,D\},|I|=d} \mathfrak{P}_I$ is a compact convex set as the intersection of compact convex sets, which is non empty ($O \in \mathcal{I}$). It follows that $\mathcal{U} = \bigcup_{I\subseteq\{1,\dots,D\},|I|=d} \mathfrak{P}_I$ is compact and connected

as a finite union of compact connected sets with a non-empty intersection, i.e. $\mathcal{I}$. Additionally $\mathcal{U}$ is star-shaped with respect to any point in $\mathcal{I}$ (since $\mathcal{I}$ belongs to all parallelotopes in $\mathcal{U}$). □

To sum up, we have three different sets of interest: the (unknown) parallelotope corresponding to the influential variables $\mathfrak{P}_I$, the intersection of all of them $\mathcal{I}$, and their union $\mathcal{U}$.

For solving problem $(\mathcal{R})$, setting $\mathcal{Y} = \mathcal{U}$ seems appealing. Unfortunately this is impractical for combinatorial reasons: as a union of $\binom{d}{D}$ parallelotopes, $\mathcal{U}$ is not easy to directly work with. Consequently, we aim at finding some tractable $\mathcal{Y}$ of small volume while satisfying $\mathcal{U} \subseteq \mathcal{Y}$. A first natural idea is to enclose $\mathcal{U}$ in a ball. For that we need the diameter of $\mathcal{U}$, which is what we pursue next.

6.2.3 Estimation of the diameter of the minimal set $\mathcal{U}$

Suppose that we want to define $\mathcal{Y}$ enclosing $\mathcal{U}$ in order to find a solution to problem $(\mathcal{R})$. We then need to find the diameter of $\mathcal{U}$, i.e. $\delta(\mathcal{U}) = \max_{\mathbf{y}_1,\mathbf{y}_2 \in \mathcal{U}} \|\mathbf{y}_1 - \mathbf{y}_2\|$, or equivalently since $\mathcal{U}$ is centrally symmetric, its radius $r^* = \delta(\mathcal{U})/2 = \max_{\mathbf{y} \in \mathcal{U}} \|\mathbf{y}\|$. Simple forms for $\mathcal{Y}$ would then either be a ball of radius r^* or a square of side $2r^*$. In both cases, r^* is needed. However, unless the dimensions d and D are small, solving $\max_{\mathbf{y} \in \mathcal{U}} \|\mathbf{y}\|$ appears to be out of reach.

Nevertheless, due to the structure of the problem, it can be decomposed over all parallelotopes. In particular, the maximum of the norm over a parallelotope is reached on its vertices since the norm is a convex function and parallelotopes are convex sets, see e.g. Chapter 32 in [Roc70]. The problem is then: $\max_{\mathbf{y} \in \mathcal{U}} \|\mathbf{y}\| = \max_{I \subseteq \{1,\dots,D\}, |I|=d} \left(\max_{\mathbf{v} \in \mathcal{V}^d} \left(\|\mathbf{A}_I^{-1} \mathbf{v}\| \right) \right)$ where $\mathcal{V}^d$ is the set of vertices of $\mathcal{C}^d$.

Maximizing the norm over only one parallelotope is relatively easy with moderate d since it amounts to inverting a $d \times d$ matrix $\mathbf{A}_I$ and considering the 2^d vertices[3] where the optimum is possibly reached. But there are quickly too many possible parallelotopes, i.e. $\binom{D}{d}$, to consider for getting r^* in general.

One option is to obtain an estimate by stochastic simulation, as proposed in Algorithm 9. The principle is to sample points on an initially large d-sphere of radius ρ and determine whether or not some are in a parallelotope. If not, the size of the sphere is reduced. Images of points in parallelotopes have at least d coordinates lower than 1 in absolute value in the high-dimensional domain. We concentrate on the d smallest ones, whose indices are denoted

[3] In fact half of them since $\|\mathbf{v}\| = \|-\mathbf{v}\|$ with $\mathbf{v} \in \mathcal{V}^d$.

by I. It remains to find the vertex of the parallelotope $\mathfrak{P}_I$ with largest norm, i.e. most distant to the center. Notice that Theorem 5.2.2 (Theorem 3 in [WZH+13]) provides a way to initialize ρ: taking $\varepsilon = 10^{-6}$ gives a probability of less than one in a million of not being too large.

Algorithm 9 Pseudo-code for the estimation of the radius of $\mathcal{U}$

Input: $k > 1$, matrix $\mathbf{A}$, number of sampled points N, $\mathcal{Q} = \emptyset$, initial radius ρ

while $\mathcal{Q} = \emptyset$ **do**
  Sample N points on $\mathcal{B}(\mathbf{O}_d, \rho)$: $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(N)}$.
  Compute $\mathbf{x}^{(i)} = p_{\mathcal{X}}(\mathbf{A}\mathbf{y}^{(i)})$, $1 \leq i \leq N$.
  Define $\mathcal{Q} = \left\{\mathbf{x}^{(j)}, 1 \leq j \leq N\,,\, \exists I = \{i_1, \dots, i_d\} \subseteq \{1, \dots, D\}\,, |x_{i_1}^{(j)}| \leq 1, \dots, |x_{i_d}^{(j)}| \leq 1\right\}$.
  $\hat{r} \leftarrow \rho$; $\rho \leftarrow \rho/k$.
end while
for $j = 1, \dots, |\mathcal{Q}|$ **do**
  Define $I = (i_1, \dots, i_d)$, the indices of the d smallest coordinates of $\mathbf{x}^{(j)} \in \mathcal{Q}$ (in absolute value).
  Compute $\hat{r} = \max(\hat{r}, \max_{\mathbf{s} \in \mathcal{V}^d} \|\mathbf{A}_I^{-1}\mathbf{s}\|)$.
end for

Output: $\hat{r}$

Note that the estimation of r^* is biased since $\hat{r} \leq r^*$ for any sample but it is useful to give first initial values for r^* in the remainder of this chapter. An alternative for enclosing a parallelotope is to compute the smallest box that contains it, as given e.g. in [LSA+13]: $[-C_1, C_1] \times \cdots \times [-C_d, C_d]$ with $C_i = \sum_{j=1}^{d} |\mathbf{B}_{i,j}|$ with $\mathbf{B} = \mathbf{A}_I^{-1}$.

6.3 Practical considerations depending on A and proposed modifications

Having $\mathcal{U} \subseteq \mathcal{Y}$ ensures to find a solution of the optimization problem $(\mathcal{R})$. We now focus on the matrix $\mathbf{A}$ with the aim of making $\mathcal{U}$ as small as possible. We start by defining properly this goal, for which analytical solutions exist with $d = 1, 2$. Then we propose an extension to $d \geq 3$ that we validate with simulation results. It allows us to propose a modified REMBO algorithm with improved robustness properties in the case of a single random embedding, along with some guidelines.

6.3.1 Objective of modifying A

Considering simply r^* (see Section 6.2.3) to define $\mathcal{Y}$ is insufficient: multiplying $\mathbf{A}$ by a positive constant k does not change $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ but changes r^* since $r^*(k\mathbf{A}) = r^*(\mathbf{A})/k$ (through

$\mathbf{A}_I^{-1}$ with parallelotopes). Similarly for strips, the half length l_i of the i^{th} strip of $k\mathbf{A}$ is l_i/k: l_i is half the distance between two parallel hyperplanes, i.e. $l_i = 1/\|\mathbf{A}_i\|_d$.

We thus focus on the quantity r^*/l^* where l^* is half the length of the smallest strip, which is not affected by a rescaling of $\mathbf{A}$. This is problem $(\mathcal{D})$:

$$(\mathcal{D}) : \min_{\mathbf{A}\in\mathcal{A}} \eta(\mathbf{A}) = \frac{r^*(\mathbf{A})}{l^*(\mathbf{A})}.$$

There are several justifications to this choice of l^* in problem $(\mathcal{D})$. First, it is simple to compute for any matrix $\mathbf{A}$, contrarily to other possibilities such as for instance the diameter of $\mathcal{I}$. In addition, $\mathcal{I}$ is closely related to the warping $\Psi : \mathbb{R}^d \to \mathbb{R}^D$ of Chapter 5. Recall that Ψ is constructed in two steps for a point in $\mathcal{Y}$: first by applying the orthogonal projection onto $\mathrm{Ran}(\mathbf{A})$ to $p_{\mathcal{X}}(\mathbf{Ay})$ to get $\mathbf{z} = p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{Ay}))$ and then, if $\mathbf{z} \notin \mathcal{X}$, by distorting it to take into account the distance between the *pivot point*: $\mathbf{z}' = (\max_{i=1,\dots,D} |z_i|)^{-1}\mathbf{z}$ and $p_{\mathcal{X}}(\mathbf{Ay})$. It follows from their construction that pivot points correspond to the exterior of $\mathbf{A}\mathcal{I}$, i.e. of $\mathcal{X} \cap \mathrm{Ran}(\mathbf{A})$, and that $\Psi(\mathcal{I}) = \mathbf{A}\mathcal{I}$. The smallest strip is thus directly connected to the pivot points of the warping.

The image of $\mathbb{R}^d$ by Ψ belongs to $\mathrm{Ran}(\mathbf{A})$, a d-dimensional subspace. To get the coordinates on $\mathrm{Ran}(\mathbf{A})$, we use $\mathbf{A}^\dagger$, the Moore-Penrose pseudo-inverse of $\mathbf{A}$ [Bar90], which is a $d \times D$ matrix such that for all $\mathbf{x} = \mathbf{Ay} \in \mathrm{Ran}(\mathbf{A})$, $\mathbf{A}^\dagger\mathbf{x} = \mathbf{y}$. In our case where $\mathbf{A}$ has full column rank, we have $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$. Furthermore the orthogonal projection onto $\mathrm{Ran}(\mathbf{A})$ is $p_{\mathbf{A}} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$: we thus have $\mathbf{A}^\dagger\mathbf{A} = Id$ while $p_{\mathbf{A}} = \mathbf{A}\mathbf{A}^\dagger\mathbf{x}$. We refer to $\mathbf{A}^\dagger\Psi(\mathbb{R}^d) = \left\{\mathbf{A}^\dagger\Psi(\mathbf{y}) \text{ with } \mathbf{y} \in \mathbb{R}^d\right\}$ as the warped space. The counterpart of r^*/l^* after warping is r^*_Ψ/l^*, with $r^*_\Psi = \max_{\mathbf{y}\in\mathcal{U}} \|\mathbf{A}^\dagger\Psi(\mathbf{y})\|$. This ratio controls the relative importance of points in $p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d))$ over those of $p_{\mathbf{A}}(\mathcal{X} \cap \mathrm{Ran}(\mathbf{A}))$. While distances between points belonging to $\mathcal{X} \cap \mathrm{Ran}(\mathbf{A})$ are d-dimensional, those between pivot points and vertices of $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ may be D-dimensional. As a consequence, most of the volume of the warped space may correspond to the exterior of $\mathcal{X}$.

Furthermore, instead of looking for a solution to problem $(\mathcal{D})$ in the entire set $\mathcal{A}$, according to Lemma 6.3.1 it is possible to restrict the search to row normalized matrices.

Lemma 6.3.1. *Let $\mathbf{A} \in \mathcal{A}$ be optimal in the sense of problem $(\mathcal{D})$. Then $\mathbf{A}'$ defined by $\mathbf{A}'_i = \mathbf{A}_i/\|\mathbf{A}_i\|, 1 \le i \le D$, is also in $\mathcal{A}$ and optimal in the same sense.*

Proof. Let $\mathbf{A} \in \mathcal{A}$ optimal in the sense of problem $(\mathcal{D})$. Reducing the length of a strip decreases the diameter of the parallelotopes that it forms. So by making all strip lengths equal to the minimal strip length, r^* decreases or remains unchanged, while l^* is not affected.

Since strip lengths are equal to $2/\|\mathbf{A}_i\|$, it amounts to have rows of equal norms for $\mathbf{A}$. Furthermore, recall that $\eta(\mathbf{A})$ is invariant when replacing $\mathbf{A}$ by $k(\mathbf{A})$, $k \neq 0$. Thus $\eta(\mathbf{A})$ is invariant by normalization of the rows of $\mathbf{A}$. Finally, $\mathbf{A}'$ is also optimal and because the rank of a matrix is not altered by multiplying its rows by a scalar, $\mathbf{A}' \in \mathcal{A}$. □

Importantly, note that even if a bias is introduced by restricting the set of possible matrices when modifying $\mathbf{A}$ to get $\mathbf{A}'$, this has no consequence on the existence of an optimal solution to problem $(\mathcal{R})$ since $\mathbf{A}'$ is still in $\mathcal{A}$.

6.3.2 Solutions for problem $(\mathcal{D})$ with $d = 1, 2$

When $d = 1$, we show in Proposition 6.3.1 that the optimal solution of $(\mathcal{D})$ is when $\mathrm{Ran}(\mathbf{A})$ corresponds to a diagonal of $\mathcal{X}$. This is stronger than Lemma 6.3.1 since here an optimal matrix has *necessarily* the same row lengths. An example is provided in Figure 6.2.

Proposition 6.3.1. *Solutions to problem $(\mathcal{D})$ with $d = 1$ are the matrices $\mathbf{A} = (\pm a, \ldots, \pm a)^T \in \mathbb{R}^{D\times 1}$ with $a \in \mathbb{R}^*$, i.e. such that $\mathbf{A}$ corresponds to a diagonal of $\mathcal{X}$. The optimal value is $r^*/l^* = 1$.*

Proof. When $d = 1$, $A_i \neq 0$ and $\mathcal{S}_i = [-1/A_i, 1/A_i]$, $1 \leq i \leq D$, thus $\mathcal{I} = [-l^*, l^*]$ and $\mathcal{U} = [-r^*, r^*]$ with $l^* = \min_{1\leq i\leq D}(1/|A_i|)$ and $r^* = \max_{1\leq i\leq D}(1/|A_i|)$. Therefore the minimum of r^*/l^* is reached when $r^* = l^*$, forcing the $|A_i|$ to be equal to a same constant a. □

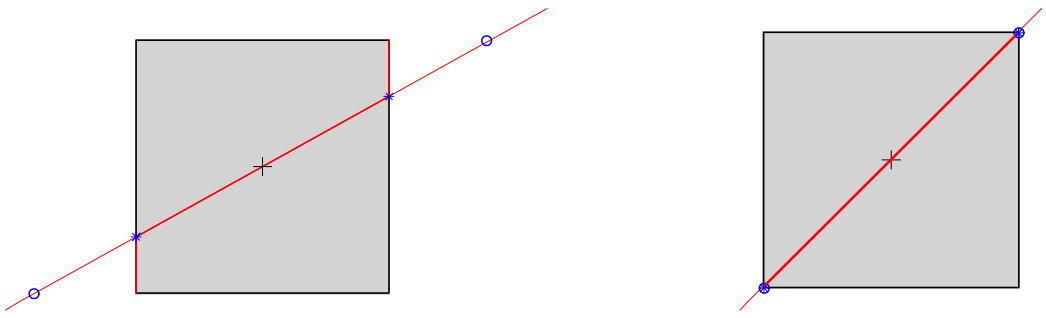


Figure 6.2 – Examples in $\mathcal{X}$ of optimal (right) and non-optimal (left) embeddings in the sense of problem $(\mathcal{D})$ with $d = 1$ and $D = 2$. The asterisks and circles denote the images by $\mathbf{A}$ in $\mathcal{X}$ of centered segments in $\mathcal{Y}$ of lengths l^* and r^*, respectively.

As a direct by-product, with this choice of $\mathbf{A}$, $\Psi = Id$. Then there is no need for the convex projection, and r^*_Ψ/l^* is also equal to 1 and optimal. With $d = 1$, optimal $\mathbf{A}$ have rows of equal length. In the $d = 2$ case, a sufficient condition for optimality is obtained if in addition the strips are evenly spread as detailed now.

Proposition 6.3.2. *Solutions to problem $(\mathcal{D})$ with $d = 2$ are matrices $\mathbf{A}$ with rows of equal norms and corresponding to regularly spaced points on a circle. In addition, let (i, j) be a*

couple of indices such that the angle between $\mathbf{A}_i$ *and* $\mathbf{A}_j$ *is equal to* π/D*. The criterion value at optimum is* r^*/l^* *with* $r^* = \max\left(\left\|\mathbf{A}_{i,j}^{-1}\begin{pmatrix}1\\1\end{pmatrix}\right\|_d, \left\|\mathbf{A}_{i,j}^{-1}\begin{pmatrix}1\\-1\end{pmatrix}\right\|_d\right)$ *and* $l^* = l_1 = \cdots = l_D = 1/\|\mathbf{A}_1\|$.

Sketch of proof. For illustration the reader is referred to Figure 6.3. Parallelotopes are here parallelograms. Let $\mathbf{A} \in \mathcal{A}$ optimal in the sense of problem $(\mathcal{D})$, with rows of equal norms as assumed from Lemma 6.3.1. The sum of the D smallest angles between any two strips is π if these angles are taken positive. The diameter of $\mathcal{U}$ is equal to the largest diameter of the parallelograms constituting it. Since all strips have equal length, the difference of diameters between parallelograms only depends on angles between strips, and the diameter of a parallelogram is a decreasing function of this angle. Hence r^* is minimal when the smallest angles between any two strips is equal to π/D, i.e. when all angles are equal since their sum is π. Otherwise at least one angle is strictly lower that π/D, which contradicts $\mathbf{A}$ optimal. Furthermore, notice that here the condition of rows of equal norms is also necessary: otherwise reducing the strip length will reduce the diameter of the parallelograms.

For the second part, from the hypothesis on $\mathbf{A}_i$ and $\mathbf{A}_j$, r^* is equal to the diameter of $\mathfrak{P}_I$ with $I = \{i, j\}$. Then the diameter of this parallelogram is equal to the length of its largest diagonal, hence $r^* = \max\left(\left\|\mathbf{A}_I^{-1}\begin{pmatrix}1\\1\end{pmatrix}\right\|_d, \left\|\mathbf{A}_I^{-1}\begin{pmatrix}1\\-1\end{pmatrix}\right\|_d\right)$, see Section 6.2.3. □

Let us remark that the naive generalization of the $d = 1$ case results in a matrix $\mathbf{A} \notin \mathcal{A}$. If $\mathbf{A}$ has to columns $(\pm a, \ldots, \pm a)^T$ and $(\pm b, \ldots, \pm b)^T$, $a, b > 0$, then rows are either $\pm(a, b)$ or $\pm(-a, b)$. Hence if $D > 2$, some rows are equal up to a sign difference, implying the existence of singular extracted $d \times d$ matrices.

An illustrative example is given, see Figure 6.3. Interestingly, for an optimal embedding, there is no violet area enclosed in the black circle, where each violet area maps to one vertex of $\mathcal{X}$. Thus having more than one point in a violet area has no interest since the value of the function is already known. As discussed in Chapter 5, it may happen when using $k_{\mathcal{Y}}$ but not with $k_{\mathcal{X}}$ and k_Ψ. In addition, representations in the warped space are given, where it can be observed that the ratio of radius of the black circle over the yellow one, i.e. r_Ψ^*/l^*, is again in favor of the optimal embedding.

In the $d = 2$ case, optimal embeddings are very regular and symmetrical. Consequently a number of quantities such as r_Ψ^* are derivable from planar geometry considerations. A summary is given in Appendix D. Specifically, optimal matrices for problem $(\mathcal{D})$ have their two columns orthogonal. With optimal matrices, the randomness initially on both strips

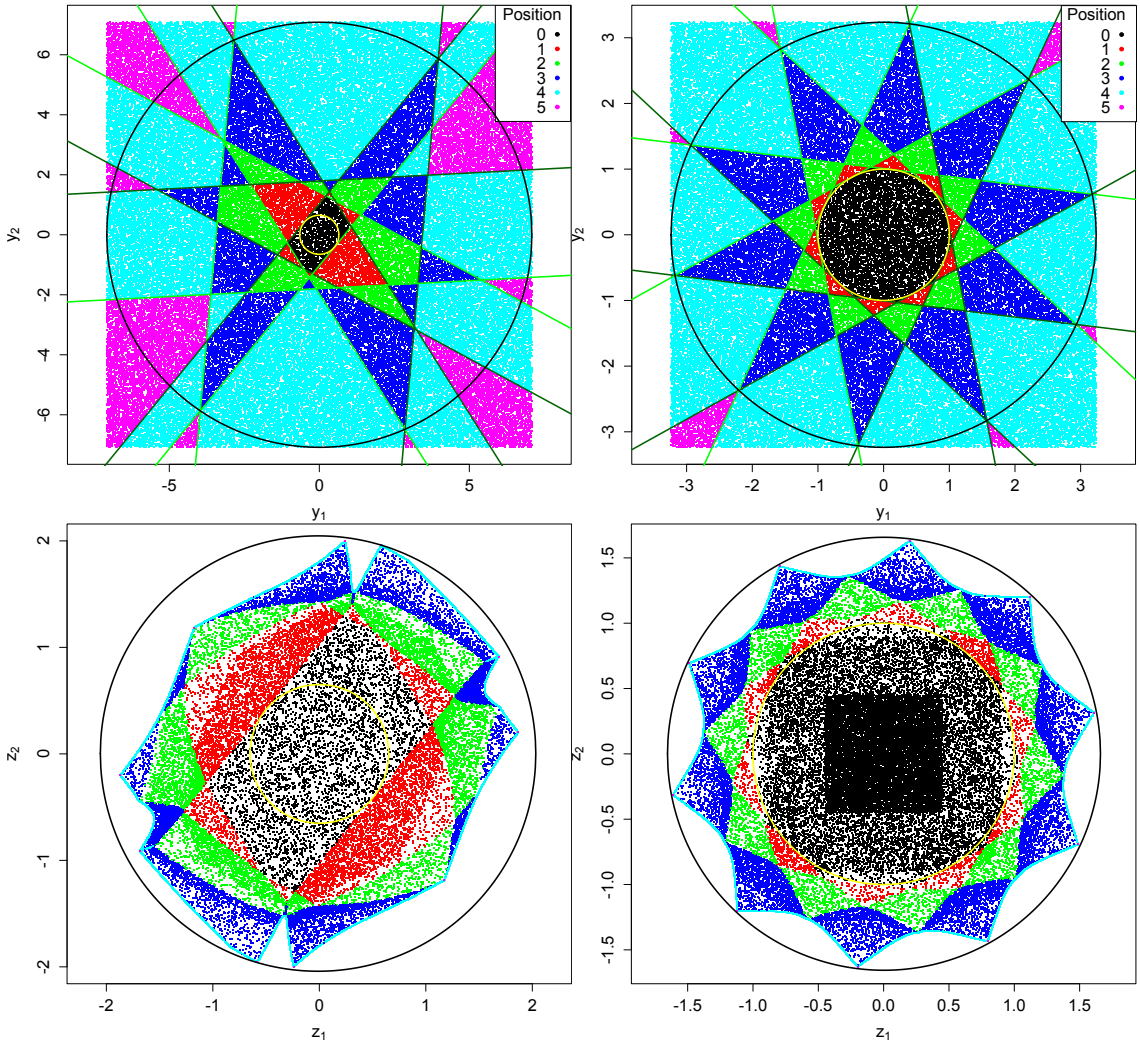


Figure 6.3 – Example of optimal (right) and non-optimal (left) embeddings in the sense of problem $(\mathcal{D})$ with $d = 2$, $D = 5$ in the low dimensional space $\mathcal{Y}$ (top) and in the warped space $\mathbf{A}^\dagger\Psi(\mathcal{Y})$ (bottom). Colors represent the number of coordinates equal to ± 1 in $\mathcal{X}$. The yellow and black circles are of radius l^* and r^* (top) or r^*_Ψ (bottom) respectively.

orientations and lengths is reduced to randomness on the indexing of stripes for variables.

6.3.3 General case

From the $d = 2$ case, we infer that optimal solutions to problem $(\mathcal{D})$ for arbitrary d would also correspond to evenly spaced points on the d-sphere. Except that formally defining "regularly" with higher d leads to several definitions, whose optimal solutions for an arbitrary D are generally unknown anyway. This is linked to the existence of regular convex polygons in arbitrary dimension with any number of vertices. The reader interested in this problem of distributing points on a hypersphere is referred to [SK97], [LDS01] and references therein.

As a preliminary and heuristic study on this topic, we propose two modifications for a matrix sampled with independent standard Gaussian entries $\mathbf{A}$:

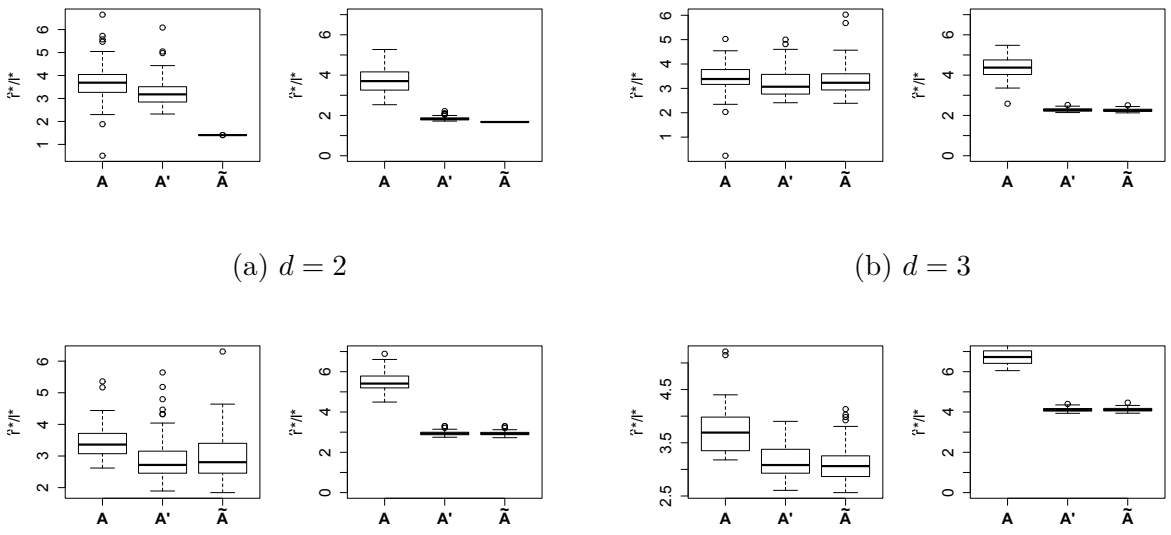
1. Normalizing the rows of $\mathbf{A}$, since this can only improve the ratio r^*/l^* (see Lemma 6.3.1). Rows of $\mathbf{A}$ are then points on the unit d-sphere. The obtained matrix is denoted $\mathbf{A}'$. This acts on l^* and all strips then have equal lengths.

2. In a second time, spreading of points on the unit d-sphere $\mathbf{S}$ by maximizing the minimal distance between any two points: $\max_{\mathbf{x}_1,\dots\mathbf{x}_D\in\mathbf{S}} \min_{1\leq j<k\leq D} \|\mathbf{x}_j - \mathbf{x}_k\|_d$. In [SK97], this is called maximizing a potential. This is computationally demanding: a $D \times D$ matrix of distances needs to be computed for the potential. Here we propose to optimize locally this function starting from $\mathbf{A}'$ to obtain $\widetilde{\mathbf{A}}$. If D is too high, random search is still possible by sampling several matrices, normalizing their rows and estimating r^*/l^* before selecting the best one.

Note that the rank property required for $\mathbf{A}$ in Proposition 6.2.2 holds with probability 1 for matrices with standard i.i.d. Gaussian entries, [WZH+13] (proof of Theorem 2).

We compare our proposed modifications for $d = 2, 3, 5$ and 10, with $D = 20d$. In addition to r^*/l^* we also estimate its equivalent in the warped space: r^*_Ψ/l^*. The corresponding results are given in Figure 6.4. As $\mathcal{U}$ is only estimated, it may not be found precisely every time, especially when increasing d. Nevertheless, the results clearly show the relevance of the proposed modifications for the r^*/l^* ratio as well as for the r^*_Ψ/l^* one. Normalizing over the rows ($\mathbf{A} \to \mathbf{A}'$) offers a great improvement compared to the one brought by the second step ($\mathbf{A}' \to \widetilde{\mathbf{A}}$).

By analyzing the results of the maximization of the potential to obtain $\widetilde{\mathbf{A}}$ from $\mathbf{A}'$, in several cases the result obtained is close to a matrix with orthogonal rows. This could be investigated further in future works. Next we consider the benefits for Bayesian optimization.



(a) $d = 2$ (b) $d = 3$

(c) $d = 5$ (d) $d = 10$

Figure 6.4 – Simulation results comparing a matrix $\mathbf{A} \in \mathbb{R}^{20d \times d}$ with independent standard Gaussian entries and its modifications $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$ for two criteria (left, right) over 100 repetitions (50 in dimension 10). The left criterion is the ratio r^*/l^*, in $\log_{10}$ scale, the right one is r^*_Ψ/l^*.

6.3.4 Impact on the REMBO algorithm and experiments

Based on all previous comments and results, we now discuss variations of the REMBO method regarding both the choice of the random embedding matrix and the selection of the low dimensional domain $\mathcal{Y}$. For the choice of the random embedding, three alternatives are available: $\mathbf{A}$ as in [WZH+13] with Gaussian i.i.d. entries, $\mathbf{A}'$ with normalized rows or $\widetilde{\mathbf{A}}$. When required, r^* is estimated with Algorithm 9.

We also propose two new options for selecting the bounds of $\mathcal{Y}$, with dedicated strategies for the optimization of the acquisition function, here the Expected Improvement:

1. $\mathcal{Y} = [-\sqrt{d}, \sqrt{d}]^d$, as in the standard REMBO method, with a constrained optimization of the acquisition function;

2. $\mathcal{Y} = [-r^*, r^*]^d$, i.e. such that $\mathcal{U} \subseteq \mathcal{Y}$, also with a constrained optimization of the acquisition function;

3. $\mathcal{Y} = [-r^*, r^*]^d$, combining a non-constrained optimization initialized in $[-l^*, l^*]^d$ with a constrained optimization in $\mathcal{Y}$.

This third strategy takes into account that while a solution to problem $(\mathcal{R})$ is to be found on $\mathcal{U}$ with probability 1, this domain is very large while $\mathfrak{P}_I$ may only represent a small frac-

tion of it. Hence it may not be worth spending too much effort on it. On the other hand, in the small domain $\mathcal{I}$, all variables – including the influential ones – are varying, making it a good start for EI optimization.

Note that as also proposed in Chapter 5 for box constraints on $\mathcal{Y}$ in l_1-norm, it is good to ensure that $\mathcal{Y}$ at least contains the ball of radius $\max_{1\leq i\leq D} l_i$ in order that each components x_i spans over $[-1, 1]$.

To compare $\mathbf{A}$, $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$ with the different strategies described above, the Hartman6 mono-objective example is considered, as in Chapter 5. In all tests, only d randomly selected variables out of the D possible ones have an influence on the function. We use the covariance kernels $k_{\mathcal{Y}}$ and k_{Ψ} in these tests. To construct designs of experiments, a simple way is to use space-filling designs in $\mathcal{Y}$. If some points have the same image by $p_{\mathcal{X}}(\mathbf{A}\cdot)$, redundant points are replaced until the required number of points is obtained. Another option when working with k_{Ψ} is to sample more points than needed in $\mathcal{Y}$ and to remove the surplus sequentially based on their distances in $\Psi(\mathbb{R}^d)$. These tests are conducted in R, with packages *DiceKriging* and *DiceOptim* for the GP modeling and Expected Improvement function. *rgenoud* and *pso* are used for optimizing the acquisition function, in addition to random search. The budget for this inner optimization is the same for all instances, as are the realizations of random matrices (before modification if performed).

The results are given in Figure 6.5. First, k_{Ψ} performs much better than $k_{\mathcal{Y}}$ (except for strategy 2 where both show a similar behavior). Normalizing the rows of $\mathbf{A}$ decreases significantly the optimality gap, while $\widetilde{\mathbf{A}}$ has only a marginal positive effect. With $k_{\mathcal{Y}}$, taking a larger domain with strategies 2 and 3 deteriorates the performance, since this kernel only consider distances in $\mathcal{Y}$. Results of $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$ with classical bounds (strategy 1) or unconstrained optimization starting from $\mathcal{I}$ (with $[-l^*, l^*]^d$) coupled with optimization over $\mathcal{U}$ (strategy 3) are very similar and they outperform those with $\mathbf{A}$, both in average and for the third quartile.

Unsurprisingly, taking $\mathcal{U}$ for bounds without adapting the optimization of the acquisition function (strategy 2) gives very poor results. This under-performance may be explained by considering the deformation of the function h with g, i.e. $g(\mathbf{y}) = h(p_{\mathcal{X}_I}(\mathbf{A}_I\mathbf{y}))$:

(a) over $\mathfrak{P}_I$, $g(\mathbf{y}) = h(\mathbf{A}_I\mathbf{y})$;

(b) over $\mathbb{R}^d \setminus \left(\bigcup_{i\in I} \mathcal{S}_i\right)$, g is piece-wise constant (all influential variables are equal to ± 1);

(c) over $\left(\bigcup_{i\in I} \mathcal{S}_i\right) \setminus \mathfrak{P}_I$, the deformation of g compared to h depends on the number of

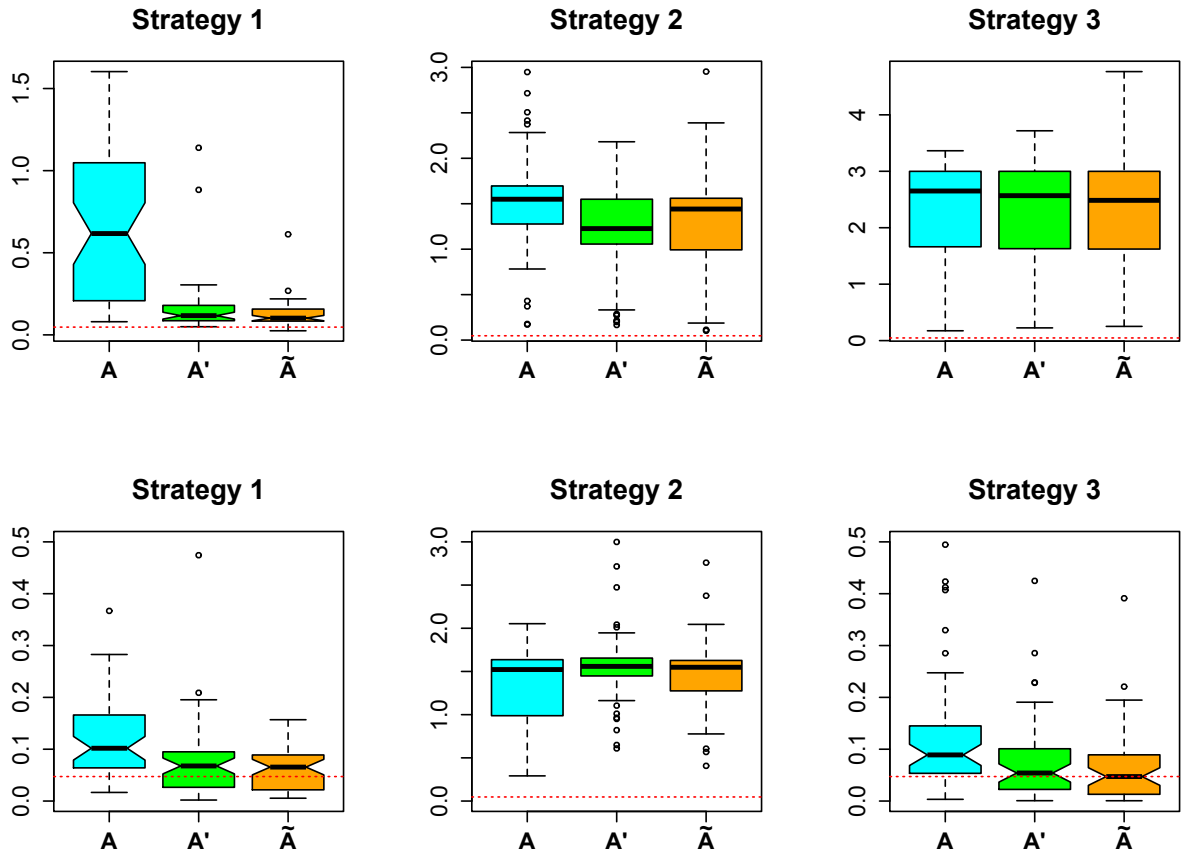


Figure 6.5 – Optimality gap for the Hartman6 function with $D = 25$ over 50 runs with a budget of 250 evaluations, with $k_{\mathcal{Y}}$ (top) and k_{Ψ} (bottom). For each strategy, matrices $\mathbf{A}$, $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$ are tested. The red dashed line indicates the best median performance, i.e. for strategy 3 with $\widetilde{\mathbf{A}}$.

influential variables that are not fixed to ± 1.

Hence the function g has three different behaviors over $\mathbb{R}^d$, which might be difficult to learn for a surrogate model. With relatively small bounds, as in strategy 1, case (a) and (b) may be predominant while for strategy 2 and 3, case (b) is expected to be predominant and case (c) is more probable. In strategy 3, the optimization budget of the acquisition function is split to focus on $\mathcal{I}$, i.e. on case (a). An illustration of these deformations and behaviors with $d = 2$, $D = 3$ for the Branin-Hoo test function (see e.g. [Gin09] for its formulation), is given in Figure 6.6.

Taking into account these instationnarities with the GP modeling would probably be beneficial. Otherwise, modeling difficulties are less important with small bounds. Yet, a case when restricting to the center is more dangerous occurs in the multi-objective setting that we consider in the following.

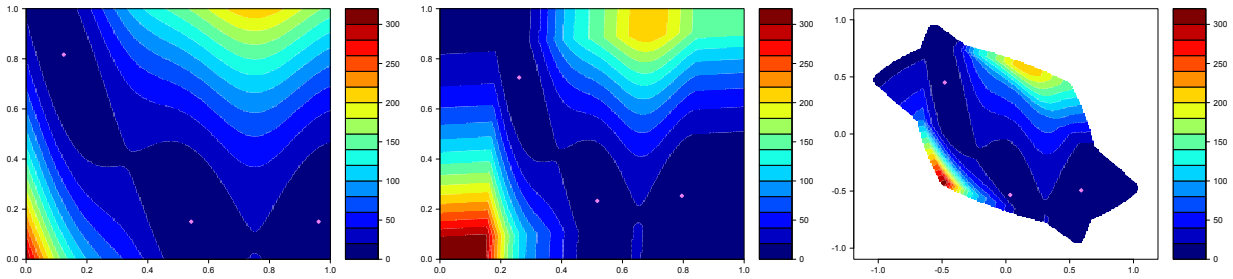


Figure 6.6 – Branin-Hoo function ($d = 2$) with 1 added non-influential dimension ($D = 3$): original (left), in the low-dimensional space $\mathcal{Y}$ with g (center) and in the warped space (right). Global minima are marked by points.

6.4 Multi-objective REMBO optimization

The REMBO method has been originally proposed for mono-objective optimization. The good performance shown in high dimension advocates questioning its transposition to the multi-objective case. However, since the set of optimal solution is on the Pareto set, failing to enclose parts of it in the research domain may cause more troubles than in mono-objective optimization. As such, it offers a challenging case study to the present work.

6.4.1 Theoretical extension

We consider that the parameter space is shared between objectives. Then the difference when dealing with several expensive objectives in the Bayesian optimization framework concerns

the acquisition function, which depends on several models, see Chapter 2. We show that this is also the case with REMBO and modify Theorem 5.2.2 to take into account several objectives in Lemma 6.4.1.

Lemma 6.4.1. *Let $f = (f_1, \dots, f_m)$ be a vector-valued function whose coordinate functions $f_i : \mathbb{R}^D \mapsto \mathbb{R}, (1 \leq i \leq m)$ are with effective dimensionalities d_i and effective subspace τ_i. Denote $\mathbf{A} \in \mathbb{R}^{D \times d}$ a random matrix with standard Gaussian independent entries and $d \geq \text{Rank} \bigoplus_{i=1}^{m} \tau_i$. Then, with probability 1, for any $\boldsymbol{x} \in \mathbb{R}^d$, there exists $\boldsymbol{y} \in \mathbb{R}^d$ such that $f(\boldsymbol{x}) = f(\mathbf{A}\boldsymbol{y})$.*

Proof. This proof is slightly modified from [WZH+13]. By definition of effective dimensionality, for each f_i there exists an effective subspace $\tau_i \subseteq \mathbb{R}^D$ such that $\text{Rank}(\tau_i) = d_i$. Denote $\mathcal{T}$ the span of all the effective subspaces, it has rank $d = \text{Rank}(\bigoplus_{i=1}^{m} \tau_i) \leq \sum_{i=1}^{m} d_i$. In addition, any $\mathbf{x} \in \mathbb{R}^D$ decomposes as $\mathbf{x} = \mathbf{x}_\top + \mathbf{x}_\perp$, $\mathbf{x}_\top \in \mathcal{T}$, $\mathbf{x}_\perp \in \mathcal{T}^\perp$. Denoting $\mathbf{x}_{\tau_i}$ and $\mathbf{x}_{\tau_i,\perp}$ the decomposition of $\mathbf{x}_\top$ between τ_i and $\mathcal{T} \cap \tau_i^\perp$, this gives for each objective $\mathbf{x} = \mathbf{x}_{\tau_i} + \mathbf{x}_{\tau_i,\perp} + \mathbf{x}_\perp$, leading to $f_i(\mathbf{x}) = f(\mathbf{x}_\top) = f_i(\mathbf{x}_{\tau_i})$. The results follows since it is shown in [WZH+13] that with probability 1, $\forall \mathbf{x} \in \mathbb{R}^D$, $\mathbf{x} = \mathbf{x}_\top + \mathbf{x}_\perp$, $\exists \mathbf{y} \in \mathbb{R}^d$ such that $\mathbf{A}\mathbf{y} = \mathbf{x}_\top + \mathbf{x}'$, with $\mathbf{x}' \in \mathcal{T}^\perp$. □

For box constraints on $\mathcal{X}$, Theorem 5.2.2 (Theorem 3 in [WZH+13]), which holds if restricting influential variables to basis variables, i.e. hypothesis $(\mathcal{H})$, is still valid with several objectives as shown in Lemma 6.4.2.

Lemma 6.4.2. *Let $f : \mathbb{R}^D \to \mathbb{R}^m$ be such that each objective f_i is of effective dimensionality d_i. Suppose that all their respective influential subspaces τ_i are the span of d_i basis vectors. Denote $\mathcal{T}$ the span of all τ_i, of rank d and $\mathbf{A}$ a $D \times d$ random matrix with independent standard Gaussian entries. Then, for any Pareto optimal solution $\boldsymbol{x}^*$ of f, its projection $\boldsymbol{x}^*_\top$ onto $\mathcal{T}$ is such that $\boldsymbol{x}^*_\top \in \mathcal{T} \cap \mathcal{X}$. Also, $\exists \boldsymbol{y}^* \in \mathbb{R}^d$ such that $f(\mathbf{A}\boldsymbol{y}^*) = f(\boldsymbol{x}^*_\top)$ with probability 1. In addition, $\|\boldsymbol{y}^*\|_2 \leq \frac{\sqrt{d}}{\varepsilon} \|\boldsymbol{x}^*_\top\|_2$ with probability $1 - \varepsilon$.*

Proof. It is sufficient to show that under the hypothesis on the effective subspaces, any Pareto optimal solution $\mathbf{x}^*$ may be written as $\mathbf{x}^*_\top + \mathbf{x}^*_\perp$ where $\mathbf{x}^*_\top \in \mathcal{T} \cap \mathcal{X}$. Denote $\mathbf{e}_1, \dots, \mathbf{e}_D$ the basis vectors of $\mathcal{X}$ and I the d indices of those spanning $\mathcal{T}$. Then, $\forall \mathbf{x} \in \mathcal{X}$, $\mathbf{x} = \sum_{i=1}^{D} x_i \mathbf{e}_i = \sum_{i \in I} x_i \mathbf{e}_i + \sum_{i \notin I} x_i \mathbf{e}_i = \mathbf{x}_\top + \mathbf{x}_\perp$. Since for all i, $|x_i| \leq 1$, $\mathbf{x}_\top$ belongs to $\mathcal{X}$. As this is true for any $\mathbf{x}$, it is true for any Pareto optimal solution $\mathbf{x}^*$. Then the existence of $\mathbf{y}^*$ with probability 1 follows from Lemma 6.4.1. The proof of the last result on $\|\mathbf{y}^*\|_2$ does not differ from the proof in [WZH+13]. □

Hence we have shown that the REMBO method is applicable to multi-objective optimization under the same hypotheses than for mono-objective optimization. Now consider

the constrained problem $\min_{\mathbf{x}\in\mathbb{E}} f_l(\mathbf{x})$, $1 \leq l \leq m$, such that $f_j(\mathbf{x}) \leq \varepsilon_j$, $1 \leq j \leq m$, $j \neq l$. This formulation of a multi-objective problem as a constrained problem is called the ε-constraint method. Since any solution of this constrained problem, if it exists, is Pareto optimal (see e.g. Theorem 3.2.2, Part. II, in [Mie99]), REMBO is also applicable to constrained optimization. We use it for instance on the industrial test case of Chapter 8.

Modifications to Algorithm 7 consist in building and updating several models instead of one (steps 4 and 8). Then in step 6, the Expected Improvement is replaced by a dedicated multi-objective [EDK11], [Pic13], [SS16] or constrained expected improvement [SWJ98], [Pic14], which are also functions of $\mathbb{R}^D \to \mathbb{R}$. Note that except in the favorable case when all influential subspaces are shared, the low dimension d is increasing linearly with more constraints and objectives.

6.4.2 Multi-objective tests

We now perform tests on modifications of the matrix $\mathbf{A}$ and optimization strategies on bi-objective problems from the literature: (P1) [Par12] as in Chapter 3, Deb3, Fonseca2 [CLVV07] and ZDT3 [ZDT00]. The results of a comparison between the standard setting (strategy 1 with $\mathbf{A}$ and $k_{\mathcal{Y}}$) and k_Ψ with $\widetilde{\mathbf{A}}$ for strategy 3 are presented in Figure 6.7 (lower is better). Except for test problem Deb3, the results are better using the proposed modifications. Note that test functions Fonseca2 and Deb3 are very difficult to model with standard GPs due to their landscapes composed of large plateaus and peaks.

Extended results corresponding to the tests of Section 6.3.4 in a multi-objective setup are given in Figures 6.8 and 6.9 for problem (P1) and ZDT3 ($d = 4$) respectively. Based on the previous experiments, only k_Ψ is tested extensively, while results for $k_{\mathcal{Y}}$ with standard bounds $[-\sqrt{d}, \sqrt{d}]$ are given as a baseline for comparison.

Results for problem (P1) show that for strategies 2 and 3, the modifications of the matrix $\mathbf{A}$ are beneficial to increase the robustness, as can be seen on box sizes (inter-quartile interval). For strategy 1 the third quartile is slightly higher after modification of $\mathbf{A}$ but the median is much lower. Notice that this time strategy 2 is competitive with the other strategies and bring the most concentrated results when coupled with $\widetilde{\mathbf{A}}$. The best median are obtained with strategy 1 with $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$, but with higher third quartile than other strategies.

For problem ZDT3, results are more mitigated. Modifications to the matrix $\mathbf{A}$ only bring an enhancement with strategy 3, with best third quartile. The reasons of this counter-performance may be various. First the Pareto set is located on the boundary on the domain,

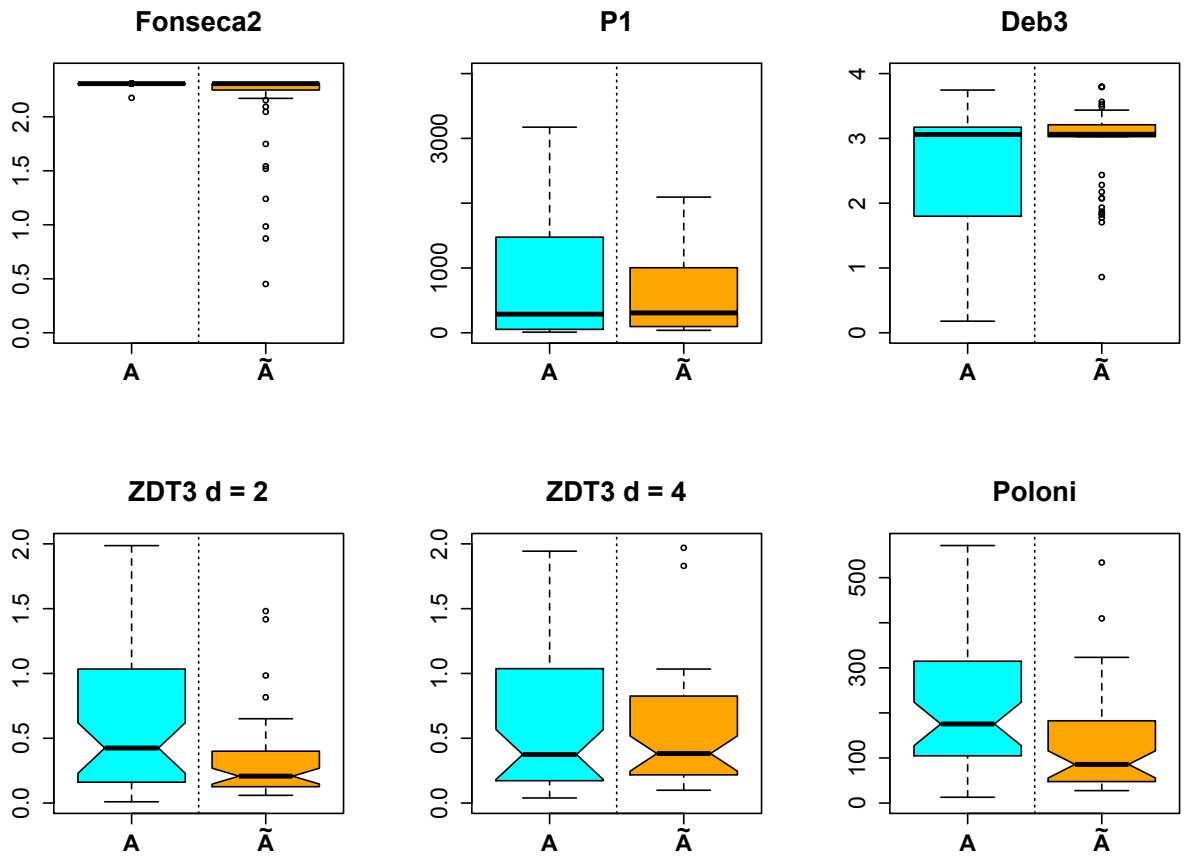


Figure 6.7 – Hypervolume difference to a reference Pareto front for bi-objective test problems Fonseca2 ($d = 6$), P1 ($d = 2$), Deb3 ($d = 2$), ZDT3 ($d = 2$, $d = 4$) and Poloni ($d = 2$), $D = 50$ and for 50 runs with a budget of 100 evaluations using $k_{\mathcal{Y}}$ with strategy 1 (left) or k_{Ψ} with strategy 3 (right).

hence a too small domain may not contain it. This is what may happen with strategy 1 with $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$: strip lengths with $\mathbf{A}$ follows an inverse χ^2 law, of mean $(1/(d-2))$ for $d > 2$ while $\mathbf{A}'$ and $\widetilde{\mathbf{A}}$ have strips of length 2. Hence classical bounds are relatively larger with $\mathbf{A}$. The second difficulty is that while the first objective of this test problem is very simple: $f_1(\mathbf{x}) = x_1$, it becomes more difficult to model in the REMBO setting since $\mathbf{y} \mapsto p_{\mathcal{X}}(\mathbf{A}_1\mathbf{y})$ is piecewise linear[4].

6.5 Conclusion and perspectives

From our analysis of the low dimensional space, we have found the smallest compact connected set $\mathcal{U}$ that contains a solution if influential variables are canonical variables. Then we have considered the problem of reducing the volume of $\mathcal{U}$ relatively to the volume of $\mathcal{I}$, the intersection of $\mathrm{Ran}(\mathbf{A}) \cap \mathcal{X}$, by modifying the matrix $\mathbf{A}$. Optimal solutions have been described for $d = 1$ and $d = 2$ while options have been proposed for the general case. We additionally proposed two new strategies to optimize the acquisition function, with a low dimensional domain $\mathcal{Y}$ such that $\mathcal{U} \subseteq \mathcal{Y}$. We subsequently tested the proposed modifications on test functions, mono and multi-objective ones, after extending REMBO to this latter case. These first results indicates that selecting the bounds adaptively with $\mathcal{U}$ along with a modified matrix for the random embedding has in general a positive impact on the performance of REMBO, especially with respect to high quantiles.

There are several attractive perspectives from remaining open questions and difficulties. The tests conducted have highlighted the importance of appropriately selecting the low-dimensional domain $\mathcal{Y}$. Considering other bounds, based for instance on the average diameter of parallelotopes, may be of interest. The recent work on Bayesian optimization with automatic resizing of bounds of [SBCdF15] could also help to this task. Regarding the selection of the matrix $\mathbf{A}$, further work is needed on optimal matrices for problem $(\mathcal{D})$ such as considering orthogonality of columns or better procedures to obtain regularly spaced points on the d-sphere. The estimation of the diameter of $\mathcal{U}$ could also be improved, potentially considering the orthogonal projection of $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ onto $\mathrm{Ran}\,\mathbf{A}$, which is detailed in Appendix D. While under hypothesis $(\mathcal{H})$, the proposed modifications of $\mathbf{A}$ still ensure that a solution is to be found in $\mathcal{U}$, more analysis is needed outside this scope.

Moreover, hypothesis $(\mathcal{H})$ is not exploited in GP modeling. Then recovering the parallelotope corresponding to influential variables in the low-dimensional space to define the

[4]This drawback can be mitigated. First, in very high dimension, optimizing even a very simple function is hard. Second, for one objective, even a poor modeling of slopes breaks should catch the general trend. This may be more troublesome when several objectives are considered simultaneously.

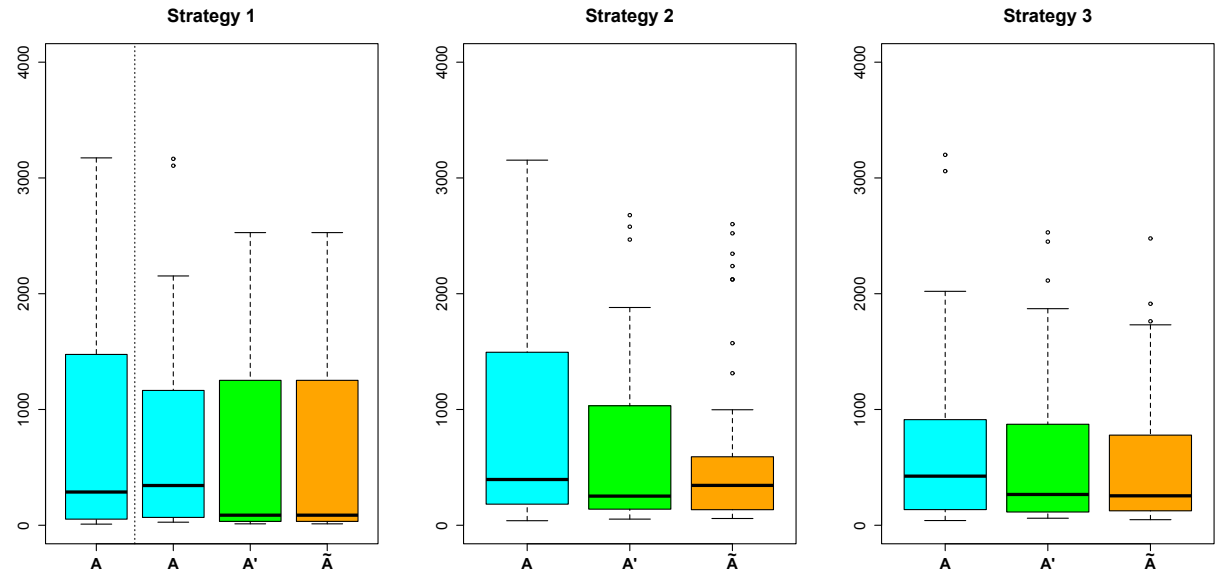


Figure 6.8 – Hypervolume difference to a reference Pareto front for the bi-objective test problem (P1), $d = 2$, $D = 50$ for 50 runs with a budget of 100 evaluations. The reference with standard REMBO using $k_{\mathcal{Y}}$ is the leftmost boxplot, separated from all other tests using k_{Ψ} by a dashed line.

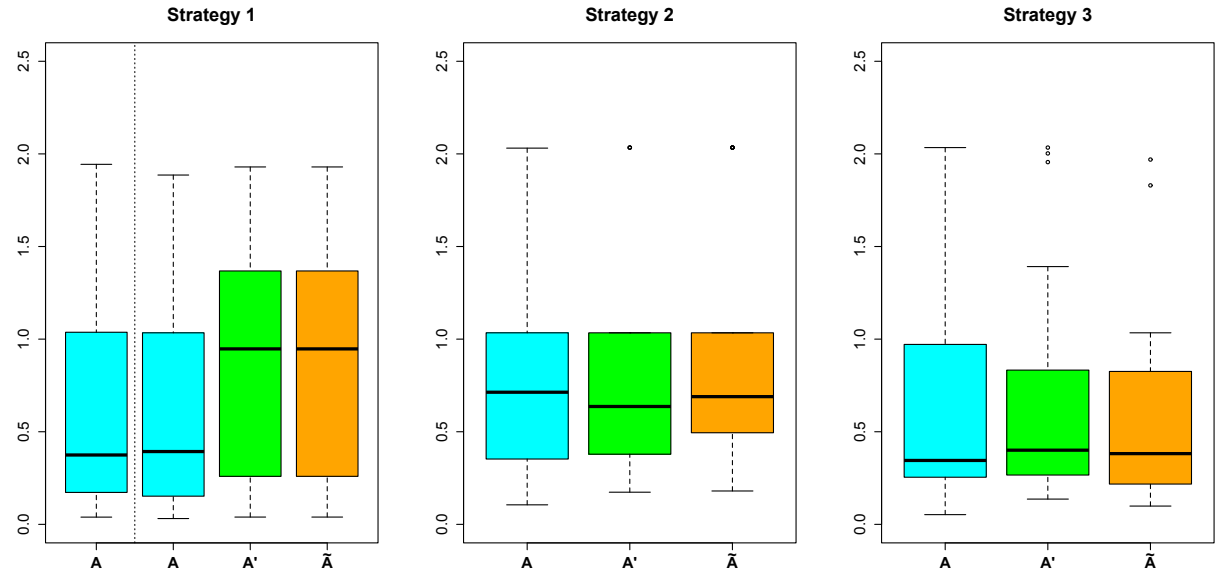


Figure 6.9 – Hypervolume difference to a reference Pareto front for the bi-objective test problem ZDT3, $d = 4$, $D = 50$ and for 50 runs with a budget of 100 evaluations. The reference with standard REMBO using $k_{\mathcal{Y}}$ is the leftmost boxplot, separated from all other tests using k_{Ψ} by a dashed line.

search domain would alleviate troubles related to instationnarity, with a knowledge on the influence of each variable as a by-product. Alternatively, the work in [MGB+15] which develops instationnarity over a cliff given by a direction is expected to adapt well in the strips configuration. Finally, combining appropriately active learning of the linear embedding as in [GOH13] and REMBO with optimization on a low dimensional subspace is a promising direction for future research.

Part IV

Contribution in implementation and test case

Chapter 7

Contributions in software for multi-objective optimization

The aim of this chapter is to give a brief overview of the *GPareto* package's features [BP15] that has been released as a contribution of this PhD work. It provides in R [R C15] multi-objective optimization algorithms for expensive black-box functions and the quantification of uncertainty method described in Chapter 3. The copula approach of Chapter 4 is also written in R, but is not integrated in the package.

7.1 Presentation of *GPareto*

As of September 2015 and to the best of the author's knowledge, multi-objective objective optimization is scarcely represented in R. There are a few packages on this subject: *nsga2R*, *emoa*, *mopsocd* and *mco*, which provides tools and algorithms such as NSGA-II [DPAM02] or hypervolume computations. As for methods available for expensive black-box functions optimization, the package *SPOT* [BBZ12] seems to be the only alternative to *GPareto*.

Building upon the *DiceKriging* [RGD12] package that offers Kriging model training for computer experiments, several associated packages deal with various related problems:

- *DiceOptim*: mono-objective optimization with criteria such as Expected Improvement, multipoint Expected Improvement or criteria suited for noisy function evaluations;
- *DiceDesign*: construction of initial designs of experiments (especially space-filling);
- *KrigInv*: algorithms for inversion problems such as contour line and excursion set estimation;
- *DiceEval*, *DiceView*: evaluation and visualization of metamodels;

- *fanovaGraph*: Kriging models from FANOVA graphs;
- *MuFiCokriging*: multi-fidelity cokriging models;
- *kergp*: alternative to *DiceKriging* for building GP models with tunable and user-defined covariance kernels.

In this suite of packages, the recently released *GPareto* offers the possibility to tackle multi-objective optimization problems. Its functions names and descriptions are given in Table 7.1.

Table 7.1 – Overview of the GPareto functions

Name	Category	Description
`checkPredict`	Test	Prevention of numerical instability for a new observation
`CPF`	Entry point	Conditional Pareto Front simulations
`crit_EHI`	Criterion	Expected Hypervolume Improvement with m objectives
`crit_EMI`	Criterion	Expected Maximin Improvement with m objectives
`crit_optimizer`	Entry point	Maximization of multi-objective Expected Improvement criteria
`crit_SMS`	Criterion	Analytical expression of the SMS-EGO criterion
`crit_SUR`	Criterion	Analytical expression of the SUR criterion for 2 or 3 objectives
`fastfun`	Class	Fastfun function
`GParetoptim`	Entry point	Sequential MO EI maximization and model re-estimation, with a number of iterations fixed in advance by the user
`integration_design...`	Initialization	Function to build integration points (for the SUR criterion)
`plotParetoEmp`	Graphical	Pareto Front visualization
`plotParetoGrid`	Graphical	Visualization of Pareto front and set
`plotSymDevFun`	Graphical	Display of the Symmetric Deviation Function
`plotSymDifRNP`	Graphical	Display of the symmetric difference of RNP sets
test functions		ZDT1-3, ZDT5-6, DTLZ1-3, DTLZ7, P1, P2, MOP2, MOP3

The background on GP modeling and multi-objective optimization is presented in Chapter 2. The two complementary research lines of the package, quantification of uncertainty and multi-objective optimization are presented with an emphasis on computational challenges and implementation choices. The possible integration in this framework of the work on high-dimensional optimization discussed in Chapters 5 and 6 concludes this chapter.

7.2 Multi-objective optimization using *GPareto*

Concepts in multi-objective optimization and corresponding EGO-like or SUR methods are presented in Chapter 2. The implemented criteria are briefly described from a computational point of view. Then criteria-optimization and sequential algorithms are presented before detailing some more advanced options.

7.2.1 Available infill criteria

Four criteria are available in *GPareto* 1.0.1:

- `crit_SMS` for the SMS-EGO criterion [PWBV08], [WEDP10] (based on the Matlab source code of the authors);
- `crit_EHI` for the Expected Hypervolume Improvement criterion [EDK11] (based on the Matlab source code of the authors in the bi-objective case);
- `crit_EMI` for the Expected Maximin Improvement criterion [SS16], [Sve11];
- `crit_SUR` for the Expected Excursion Volume Reduction criterion [Pic13].

The `crit_SMS` criterion has an analytical expression for any number of objectives while the one for `crit_EHI` is only for the bi-objective case. There is a semi-analytical[1] formula for `crit_EMI` but it has not been implemented in *GPareto* yet. Note that the formula for `crit_EHI` is coded in *Rcpp* [EF11], [Edd13], which offers considerable speed-up over an R implementation. In *SPOT*, the Expected Hypervolume Improvement also relies on the formulas of [EDK11] for the bi-objective case, coded in R. For more objectives, a variation of the SMS-EGO infill criterion as well as other lower confidence bounds criteria are available in *SPOT*. Hence the multi-objective SUR approach as well as Expected Maximin or Hypervolume Improvement criteria for any number of objectives are new in R.

For `crit_EMI` and `crit_EHI` with $m > 2$, computations rely on sample average approximation (SAA) [Sha03] as proposed in [Sve11]. The principle is to take samples from the posterior distribution of $\mathbf{Y}(\mathbf{x})$, i.e. $\mathbf{Y}(\mathbf{x})^{(1)}, \ldots, \mathbf{Y}(\mathbf{x})^{(p)}$, and take the average of the improvement function over these samples: $\mathbb{E}\left(I(\mathbf{Y}(\mathbf{x}))|\mathcal{A}_n\right) \approx \frac{1}{p}\sum_{j=1}^{p} I(\mathbf{Y}^{(j)}(\mathbf{x}))$. The larger the number of samples p, the better the approximation. In addition, when samples of the posterior distribution $\mathbf{Y}(\mathbf{x})$ are generated from Cholesky decomposition, i.e. $\mathbf{Y}(\mathbf{x})^{(j)} = \mathbf{m}_n(\mathbf{x}) + \mathbf{C}(\mathbf{x})\boldsymbol{\xi}_j$, with a fixed sample $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p \sim \mathcal{N}(\mathbf{0}_m, I_m)$, $\mathbf{m}_n(\mathbf{x}) = (m_n^{(1)}(\mathbf{x}), \ldots, m_n^{(m)}(\mathbf{x}))^T$ and $\mathbf{C}(\mathbf{x})$ the Cholesky decomposition of the covariance matrix of $Y_1(\mathbf{x}), \ldots, Y_m(\mathbf{x})$, then the SAA estimation is a deterministic function.

`crit_SUR` requires Monte-Carlo integration of the probability for a point of not being dominated at the next step. Similarly to the *KrigInv* package [CPG14], possible alternatives to select integration points are with uniformly distributed random points in E or points from a low discrepancy or quasi Monte Carlo sequence such as a Sobol sequence. No importance sampling scheme is implemented yet, see e.g. [Pic13], [CPG14], but users have the option to

[1] Numerical quadrature is needed for some 1-dmensional integrals, see [Sve11].

provide their own integration points and weights. For now `crit_SUR` is available for two or three objectives.

In terms of complexity, both `crit_EHI` with $m > 2$ and `crit_SMS` use hypervolume computations provided in the *emoa* package (much more frequently for the first one, which is thus slower). Those have an exponential complexity in the number of objectives and also depend on the number of points in the Pareto front. For `crit_EMI` the complexity mainly depends on the number of sample points for the SAA approximation and linearly in the number of objectives, it is more affordable than `crit_EHI` for more than two objectives. For `crit_SUR`, the complexity is essentially related to the integration over the input domain which can become cumbersome with many variables.

Importantly, except for `crit_SUR`, these criteria depend on the relative scaling of the objectives, i.e. multiplying one objective by a constant modifies the results. Scaling may be performed by the user, e.g. from the maximum and minimum values observed for each objective as in [Par12] or [Sve11]. In addition, `crit_EHI` and `crit_SMS` need a reference point for bounding hypervolume computations. If no reference point is given by the user, we set it to $\max_{\mathbf{y}_j \in \mathcal{P}_n} (y_1^{(i)}, \ldots, y_n^{(i)}) + 1$, $1 \leq i \leq m$, as done in [PWBV08] and references therein. The scaling and additional parameters are some of the drawbacks of multi-objective infill criteria, as discussed in [WEDP10] and [Sve11].

A brief comparison of the different criteria is given in Table 7.2.

Table 7.2 – Summary of the characteristics of infill criteria available in *GPareto*. The computational costs are given for the bi-objective example of Figure 7.2. Note that the cost of `crit_EHI` is low in this case but increase exponentially with the output dimension. `SURcontrol` is a list of parameters depending on the integration strategy chosen.

Name	Indicator	Analytical	m	Cost	Parameters
`crit_EHI`	Hypervolume	Yes ($m = 2$)	Any	+ (to +++)	`refPoint`, `nb.samp` ($m > 2$)
`crit_EMI`	Additive-ϵ	No	Any	++	`nb.samp`
`crit_SMS`	Hypervolume	Yes	Any	+	`refPoint`
`crit_SUR`		No	$m \leq 3$	+++	`SURcontrol`

To illustrate the functionalities of *GPareto* discussed here and highlight differences between criteria, we take the (P1) test problem from [Par12], described in Chapter 3. Its Pareto set and Pareto front are displayed in Figure 7.1. The four criteria are represented in Figure 7.2. Their maxima appear to be located quite close to Pareto optimal solutions, with emphasis on different locations. Notice also the proximity of the results between `crit_EHI` and `crit_SMS` (negative values for penalized areas are set to 0 in order to keep similar color-scales).

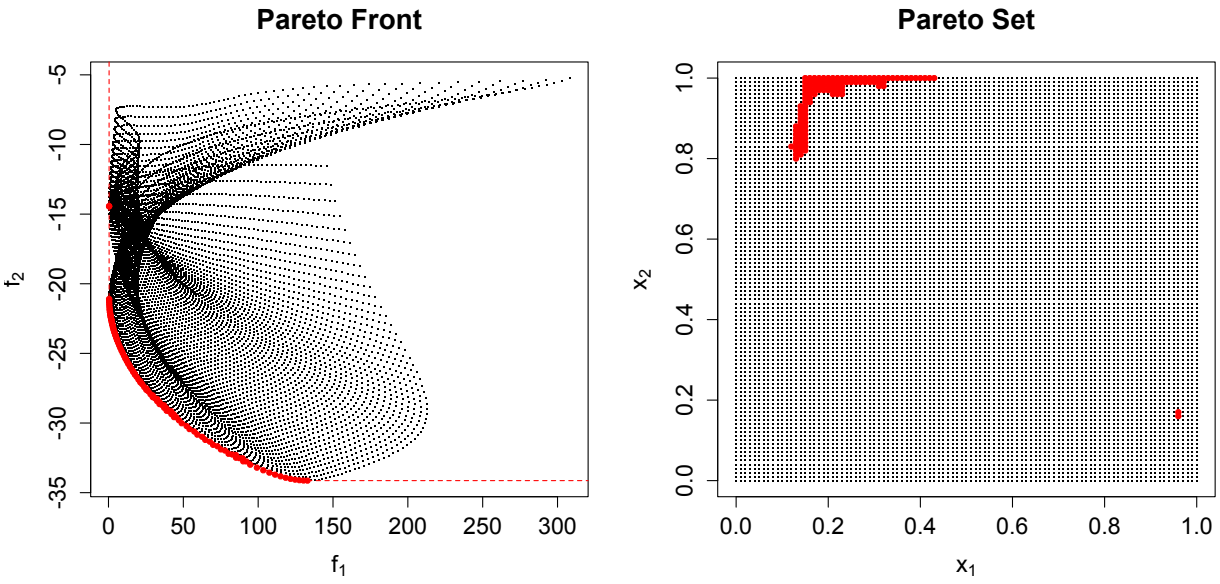


Figure 7.1 – Optimal points for the (P1) test problem, with optimal points in red in the input space, i.e. the Pareto set (left) and in the objective space, i.e. the Pareto front (right).

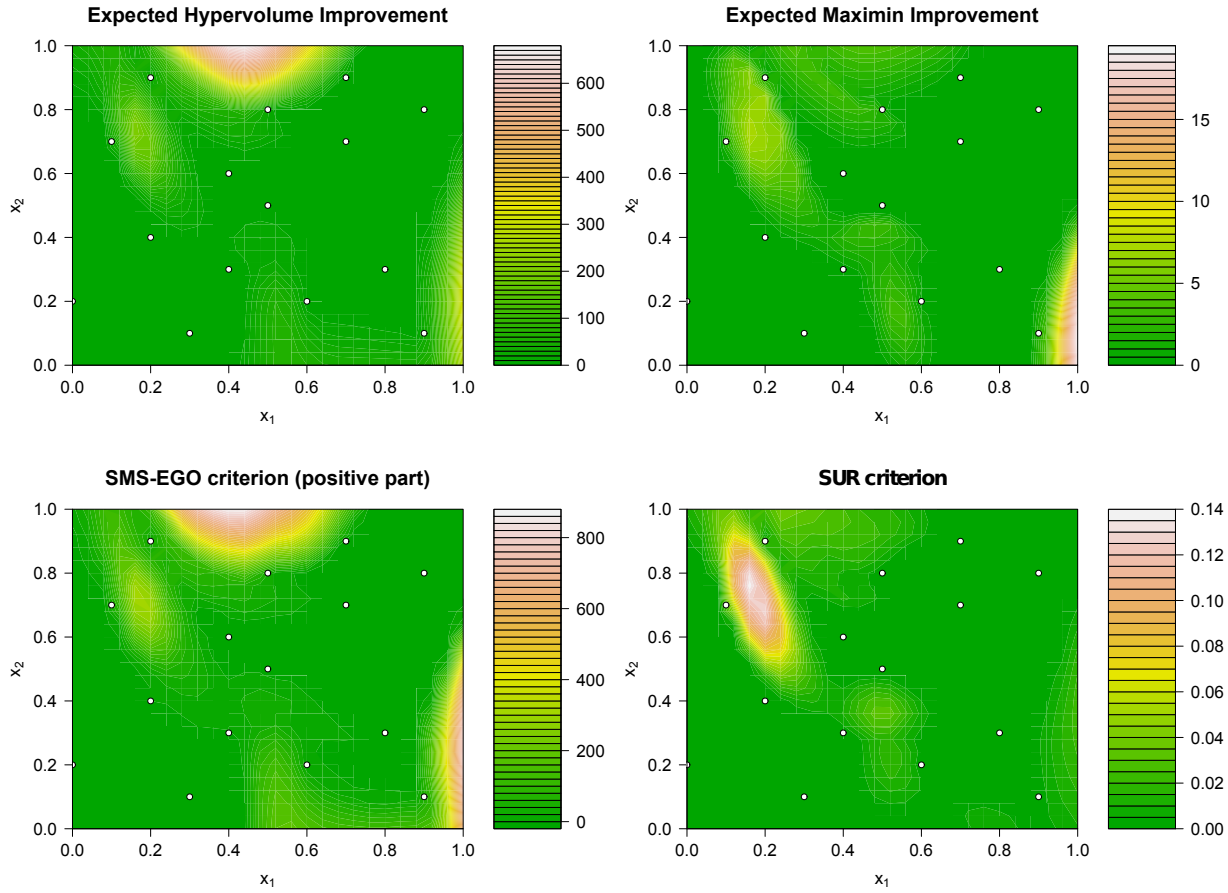


Figure 7.2 – Values in colorscale of the four infill criteria available in GPareto on a 26×26 grid, with 15 observations of the (P1) function.

Other test functions are provided in *GPareto*, such as problems in the MOP [VVL99], ZDT [ZDT00] and DTLZ [DTLZ05] test suites. As a perspective, it would be possible to develop multipoint versions of the criteria, e.g. as in [Sve11] and [Par12]. Also faster algorithms for the EHI and hypervolume are now available, see e.g. [CDD13], [HDYE15] and could be integrated.

7.2.2 Optimization of the criteria

The same framework is applied to the four possible criteria with two interfaces: one-shot optimization with `crit_optimizer` or sequential optimizations with `GParetoptim`. Optimizing the criteria, a.k.a. acquisition functions, is quite complicated due to their multi-modality. Besides, in general, no derivative expressions are available and there are large plateaus (see for instance the landscapes of Figure 7.2). On top of that, the attraction basin of the global optimum of the infill criterion may have a very small volume in E. Nonetheless, acquisition functions are typically much cheaper to evaluate than the objective functions and intensive optimization can be carried out.

Three solutions to perform this inner optimization are provided in *GPareto*:

1. the user can provide a set of candidate points with `optimcontrol` in `crit_optimizer` and `GParetoptim`;
2. the default optimization routine is `genoud` [MS11], a genetic algorithm;
3. the `psoptim` optimization method [Ben12], a particle swarm algorithm is also provided;

and the corresponding tuning parameters may be passed to `optimcontrol`. For now, passing any optimizer as an argument is not possible. Results after several iterations of criteria optimization and model update on the same example as previously are presented in Figure 7.3. All criteria perform remarkably well, with points added on or very close to the Pareto front. A perspective to improve the user-friendliness of the package would be to provide an entry point taking only the search domain and the objective functions.

7.2.3 Advanced options

Motivated by applications such as the one presented in Chapter 8, where the mass objective function is computable at a negligible cost compare to other objectives, *GPareto* offers an option for MOO in case of co-existing cheap- and expensive-to-evaluate objectives. To ensure compatibility with the infill criteria, fast objectives are wrapped in the `fastfun` class which mimics the behavior of methods such as `predict` or `update`. Then predicting the value at a new point amounts to evaluating the fast function, which returns the corresponding value

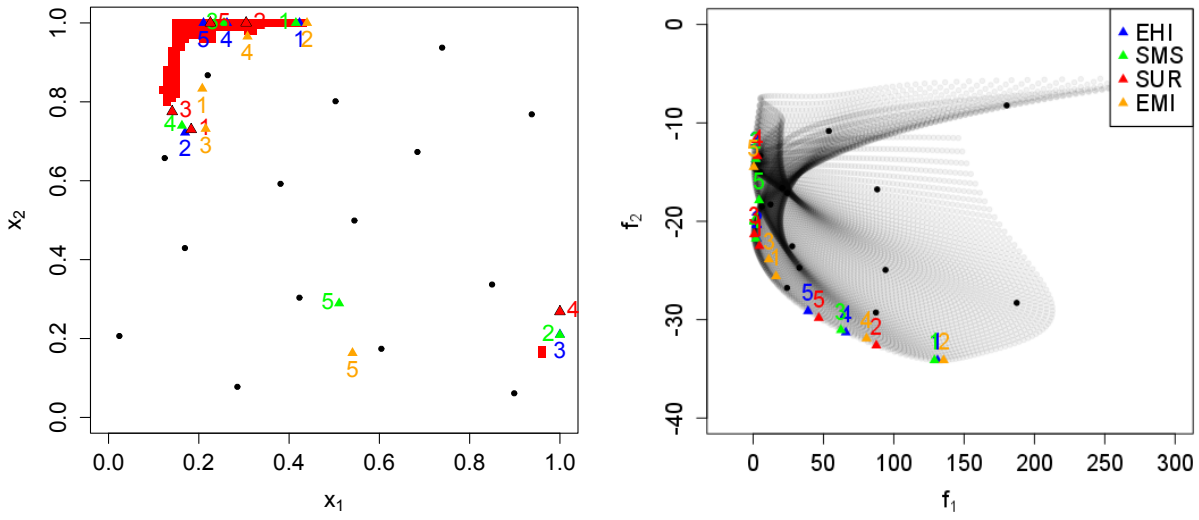


Figure 7.3 – Results of five iterations with `GParetoptim` with the four possible infill criteria in the input (left) and output spaces (right). The true Pareto set (left) is represented in red.

with a zero prediction variance, exactly like what happens for already evaluated points. Figure 7.4 illustrates on a small example that using fast objectives, if possible, improves the results since there is no longer a prediction error on them.

Another computational challenge with Kriging, discussed, e.g. in [RGD12], is the numerical non invertibility of covariance matrices. It usually happens whenever design points are too close. This is especially troublesome in optimization since when converging, points are likely to be added close to each other. In *GPareto*, preventing this problem is achieved with the `checkPredict` function. Before evaluating the selected criterion, `checkPredict` tests whether the new point $\mathbf{x}$ is too close to existing ones, with a tunable threshold that can be passed as argument. Three options are available to define when designs are considered as "too close":

- minimal Euclidean distance in the input space: $\min_{1\leq i\leq n} d(\mathbf{x}, \mathbf{x}_i)$;
- ratio of the predictive variance $s_n(\mathbf{x})^2$ over the variance parameter for stationary kernels;
- minimal "canonical distance" associated with k_n: $\min_{1\leq i\leq n} \sqrt{k_n(\mathbf{x},\mathbf{x}) - 2k_n(\mathbf{x},\mathbf{x}_i) + k_n(\mathbf{x}_i,\mathbf{x}_i)}$.

The first two options are also used in *KrigInv* and *DiceOptim* respectively. The first one is the less computationally demanding but also the less robust.

Moreover, to improve stability of the update of already existing models with new observations, first an update with re-estimation of the hyperparameters is performed. Then, if it

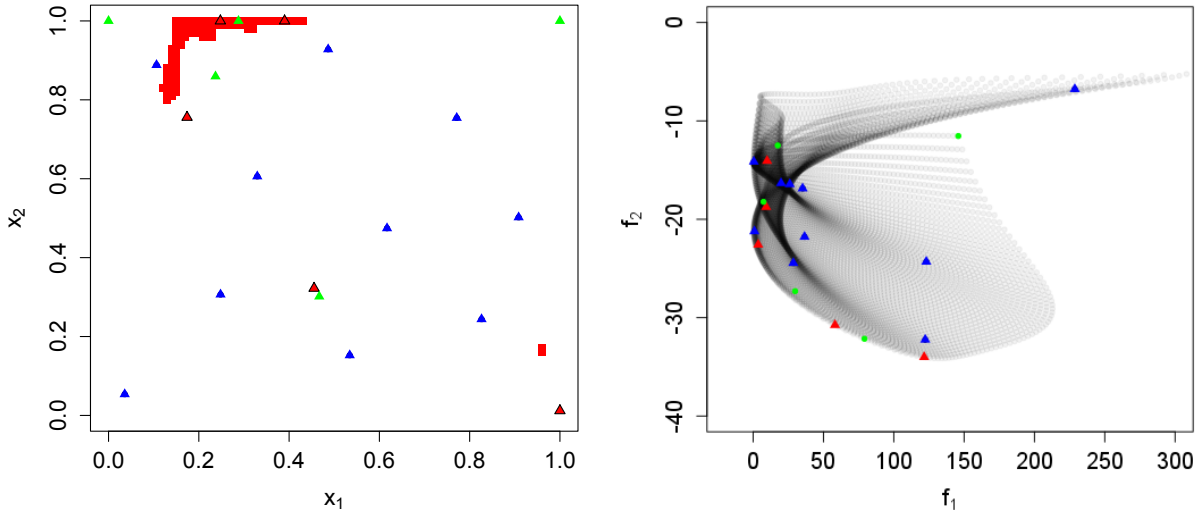


Figure 7.4 – Results of five iterations with `GParetoptim` with Expected Hypervolume Improvement, in the input (left) and output spaces (right), considering the second objective as expensive (green triangles) or cheap (red triangles). Initial observations are depicted as blue triangles and the true Pareto set (left) is plotted in red.

has failed a new update is tested with the old hyperparameters. If this is still insufficient to train the model with all observations, the user may try to remove some points or apply the *jitter* technique consisting in adding a small constant to the diagonal of the covariance matrix to improve its condition number, see e.g. [RGD12]. Replacing two close observations by one observation and its estimated derivative as proposed in [Osb10] is another appealing solution.

7.3 Uncertainty quantification using *GPareto*

This is the computational twin of Chapter 3 in the bi-objective case. A possible solution to compute conditional simulations is to rely on the *DiceKriging* package but other packages or methods could be used.

The entry function is the creator of the `CPF` class, which deals with computing the attainment function, Vorob'ev threshold β^*, Vorob'ev Expectation (VE) and Vorob'ev deviation (VD), from a grid discretization. It takes as arguments:

- `fun1sims`, `fun2sims` the sets of conditional simulations for both objectives;
- `response` the known objective values;

and, optionally:

- `paretoFront` the current Pareto front;
- `f1lim`, `f2lim` evenly spaced sets of points that define the grid;
- `refPoint` reference point for hypervolume computations;
- `n.grid` the length of `f1lim` and `f2lim`;
- `compute.VorobExp` and `compute.VorobExp` booleans to compute or not VE and VD.

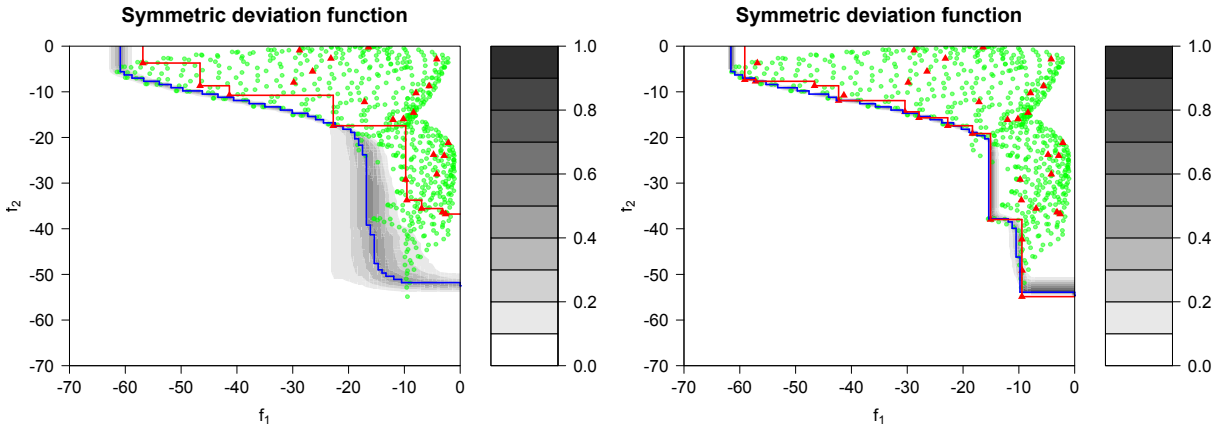
If not provided, `f1lim`, `f2lim` are taken from objective-wise minima and maxima of the conditional simulations. When `refPoint` is provided, it defines the maxima of `f1lim` and `f2lim`.

The empirical attainment function is calculated on the grid given by `f1lim` and `f2lim` from the RNP sets computed using the `nondominated_points` function of *emoa*. Taking advantage of the regularity of the grid to compute volumes, the Vorob'ev threshold is computed quickly by dichotomy. Then the Vorob'ev deviation is a sum of hypervolume indicator values, using the `dominated_hypervolume` function of *emoa*. The method `plot` applied to `CPF` objects display the attainment function in grey-scale, and possibly the VE.

In addition, for visualization of the remaining uncertainty, the `plotSymDefFun` function computes the empirical symmetric deviation function. Again, a grid is used to estimate the probability for a cell to belong to the symmetric difference between the random attained set and the Vorob'ev expectation. As an illustration of the interest of using the quantification of uncertainty, we present the results of the empirical symmetric deviation function before and after optimization on the Poloni test problem [PGOP00] in Figure 7.5 (results for the (P1) problem are in Figure 3.6). At an initial stage, here with 15 observations, the Vorob'ev expectation and symmetric deviation function highlight portion of the objective space where improvements are possible, here especially at the bottom of the initial Pareto front. At the end, the Vorob'ev expectation provides a rather accurate approximation of the Pareto front, with a lower uncertainty.

The `CPF` class is used in Appendix B to test the SUR criterion associated this time with the Vorob'ev deviation. However the computational overhead associated is still too important as is. Passing some routines of `CPF` in *Rcpp* could alleviate this limitation as well as fast methods to generate conditional simulations. Future research may also include applying the ideas of Appendix C.

At last, all the methods described so far are limited in terms of input dimension. Extending them relying on random embeddings is one possible solution that does not require re-implementing everything from scratch.



(a) $VD = 181$ (b) $VD = 89$

Figure 7.5 – Symmetric deviation function in grey-scale with 25-points initial design of experiments and after ten iteration of `GParetoptim` with the SUR criterion (red triangles). The Vorob'ev expectation is in blue and green points are the image of a 100×100 grid in the objective space by function `P2` (Poloni test problem [PGOP00]).

7.4 Perspectives toward higher dimensions

The REMBO method along with the modifications proposed in Chapters 5 and 6 have been implemented in R. The corresponding codes are not yet released and we discuss here a possible integration within the DICE and ReDICE packages.

On the first hand, training a GP model is already performed with *DiceKriging* and now also with *kergp*. On the other hand, possible acquisition functions for mono- or multi-objective optimization are implemented within *DiceOptim* and *GPareto*. As for constrained infill criteria, they have been implementing by modifying *DiceOptim*. The missing components for a REMBO method are then mostly mapping functions and connection methods.

The skeleton for a REMBO method using R packages is given as Algorithm 10 with notations from Chapters 5 and 6. Methods to select the low dimensional domain as well as for selecting a matrix $\mathbf{A}$ could also be included.

Algorithm 10 A possible framework for REMBO in R

Input: $m \geq 1$ objective functions of $\mathcal{X} \subset \mathbb{R}^D \to \mathbb{R}$, a budget of evaluations n_m, n_b the number of evaluations for the design of experiments, a low-dimensional domain $\mathcal{Y} \subset \mathbb{R}^d$, a matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$ and a mapping Ξ (e.g. Ψ, u, Id)

1: Construct an initial design of experiments in $\mathcal{Y}$: $\mathbf{y}_1, \ldots, \mathbf{y}_{n_b}$, e.g. using *lhs* or *DiceDesign*.
2: Evaluate $g = f \circ u$ on the design of experiments.
3: **while** $n \leq n_m$ **do**
4: Train the Kriging models with inputs $\Xi(\mathbf{y}_1), \ldots, \Xi(\mathbf{y}_n)$ and outputs $g(\mathbf{y}_1), \ldots, g(\mathbf{y}_n)$, e.g. using `km`.
5: Optimize the acquisition function evaluated on $\Xi(\mathbf{y})$, e.g. to get $\mathbf{y}_{n+1} \in \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\, \texttt{EI}(\Xi(\mathbf{y}))$, e.g. with acquisition functions from *DiceOptim* or *GPareto*.
6: Evaluate $g(\mathbf{y}_{n+1})$.
7: **end while**

Chapter 8

Industrial test case

Complementing and/or substituting real life experiments by high-fidelity simulations has been a significant trend in the industry during the last decades. Some phenomena are now very precisely described while for others, such as for composite materials, accuracy is still perfectible. As would also happen with physical testing, the preparation of the *black-box* is quite complex. The standard procedure is to first create and parametrize a numerical representation of the considered device using Computer Assisted Design (CAD). The definition and selection of parameters at this stage is of uttermost importance since it could result in instability later if some configurations end up to be unfeasible. Then the behavior of the device is simulated using a dedicated solver, here Pam-Crash, an explicit finite element program for fast dynamic impacts. Finally, a post-processing step is needed to extract the value(s) of interest.

In the case of crash-worthiness, with its inherent bifurcation behavior, see e.g. [Ros12], results may change even with exactly the same design due to numerical noise, e.g. from different meshing in the finite elements resolution. In physical experiments, slightly different environmental conditions would lead to variability as well. Other difficulties not considered here are the influence of some uncontrolled variations of the thickness due to the fabrication process, possibly having a substantial impact on the robustness of the solution found.

Throughout these three years, we have worked on a test case to confront with a real application. The corresponding model was created around the year 2007 for the design of a car now on the market. It is composed of 47 continuous parameters; four crash scenarii define four objectives. The main goal being to reduce the mass of the device, these four objectives are sometimes considered as constraints. In the remainder of this chapter, we take several viewpoints reflecting these goals. After presenting the application, we discuss the results of a sensitivity analysis and propose customized surrogate modeling. Then we

present the outcomes obtained with the *GPareto* package. They ultimately provide a baseline for a comparison when using the REMBO algorithm.

8.1 Presentation of the rear shock absorber

The rear shock absorber, illustrated in Figure 8.1, is a plastic device located directly behind the rear bumper. Its aim is to prevent intrusion in case of a shock and in particular the deformation of the underlying metallic structure in low speed impact scenarii ECE42. In the considered set-ups, the impact occurs in the middle or on the side of the vehicle, on either an empty or a loaded car, see Figure 8.2. The impactor speed, 4 or 2.5km/h, is sufficiently slow to discard dispersion of the results and to consider them as deterministic. The absorber is composed of 47 parameters: thicknesses of 39 *stiffening ribs* plus 8 structuring parts. The main objective is to reduce as much as possible the mass of the device while ensuring that some thresholds are not exceeded in terms of intrusion. These values are extracted with a post-processing of the crash simulation, by considering the displacement of a specific node. Two points of view were adopted: either taking five objectives or one objective with four constraints into account. The first approach is potentially more interesting if the specifications are not fixed or may vary while the second one is focused on a specific portion of the Pareto front.

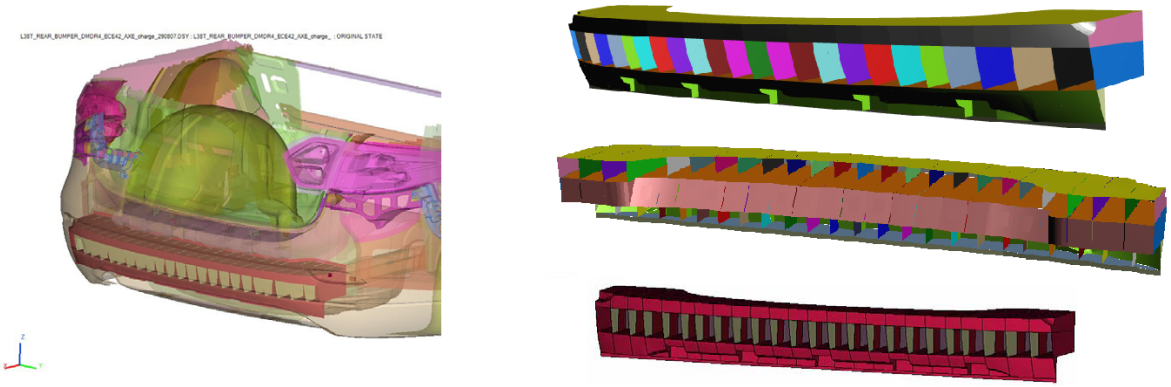


Figure 8.1 – Presentation of the rear shock absorber. Left: global vue on the rear of the vehicle, right: different views of the considered device.

This application has been an opportunity to test several approaches and confront with real industrial problems. Indeed, it has 47 parameters, which is close to the limit of what one can treat with standard GP models. In 2007, an optimization study has been conducted using the Alternova software, relying on expert models. Here this latter consists of two PolyMARS and one regression models. Note that, back then, the strategy employed was first to reduce the perimeter to 35 variables and work only on the axial impacts before integrating

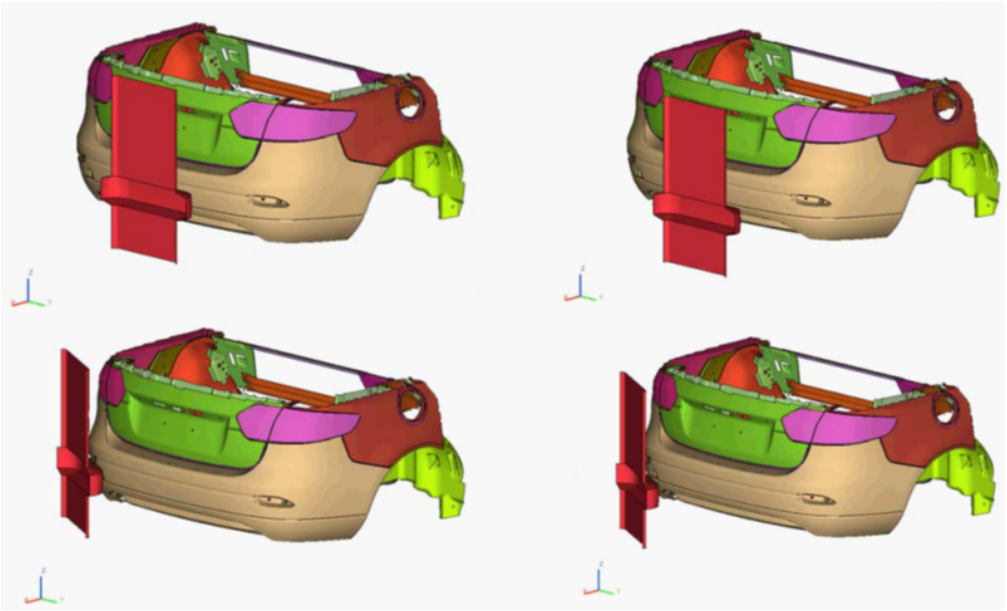


Figure 8.2 – Illustration of the four impact scenarii: axial (top) or lateral (bottom) impact on a loaded (left) or an empty (right) vehicle.

all variables and scenarii. This should be kept in mind when comparing the results and number of evaluations. This test case has also been experimented on, in [Vil08], but only with a subset of the objectives.

8.2 Creation of customized kernels and sensitivity analysis

During the various stages of the optimization study, the data base of observations has been filled with a variety of points coming from different types of experimental designs (a resolution III classical design and several maximin LHS designs), various optimization strategies (only some objectives considered for instance) and algorithms (EHI, NSGA-II, REMBO). They have been used to perform some sanity checks on quality of candidate metamodels, as advocated e.g. in [JSW98].

8.2.1 Initial sensitivity analysis

Once a surrogate model such as a GP model is fitted, global sensitivity analysis can be performed directly on it. To do so, a popular tool is the Functional ANOVA (FANOVA) decomposition, see e.g. [ES81], [Sob01], [MRCK12], [FRM13], which writes:

$$f(\mathbf{X}) = \mu_0 + \sum_{i=1}^{d} \mu_i(X_i) + \sum_{i<j} \mu_{ij}(X_i, X_j) + \cdots + \mu_{1,\ldots,d}(X_1, \ldots, X_d) \tag{8.1}$$

with centered and orthogonal terms, i.e. $\mathbb{E}(\mu_J(X_j)) = 0$ and $\forall J' \neq J, \mathbb{E}(\mu_J(X_J)\mu_{J'}(X_{J'})) = 0$, using standard index set notations. In particular, the μ_i are the main effects and the μ_{ij} are twofold interactions.

From this decomposition, FANOVA graphs, available in the R package *fanovaGraph* [FMRJ14], are a tool to visualize with a graph the main effects and the additive structure, represented by cliques. In a graph, a clique is a subset of vertices such that they are all adjacent, i.e. connected by an edge. More precisely, main effects are represented by the thickness of the sphere corresponding to a variable while segments joining variables gives the importance of the interaction. An edge between two variables is removed if all possible Sobol indices in which they intervene are null. As all effects are estimated, their values are possibly small but not zero. Hence the raw graph contains all interactions and selecting a threshold on the value of the total interaction indices determines the separation of variables into cliques as well as their number. The graphs without thresholding corresponding to the four impact configurations are depicted in Figure 8.3. Even if it is highly dependent on the (possibly terrible) quality of the fitted model, it clearly shows that only a handful of variables really are influential in the different cases, with very low interactions. This statement justifies the application of REMBO in the following.

Let us remark that it is not a real surprise that only few parameters have an important impact on the response. Particularly, due to the geometrical configuration of the problem, lateral ribs are not expected to have a great influence in axial impact and vice versa. Also note that this can be analyzed directly on the outputs of the numerical simulations. For instance, taking the results for the minimum, maximum and reference design configurations[1] and considering stresses or strains on the different parts of the absorber, it can be seen that they are concentrated where the impact occurs. For instance, see Figure 8.4 for an illustration. In Table 8.1 is a summary of strains in the different configurations, making it clear that large deformations only occur for a few components. This is useful in particular to adapt the covariance kernel in the GP modeling.

8.2.2 Customization of kernels

Due to the dimensionality of the problem, GP modeling is expected to be difficult. In this case, several adaptations of the structure of the covariance kernel are possible. We thus compare some options, relying on the analysis above.

We consider here the Matérn 5/2 kernel, which is recommended in [Ste99] over the Gaus-

[1] with all thickness to minimum, maximum or as in the configuration before optimization.

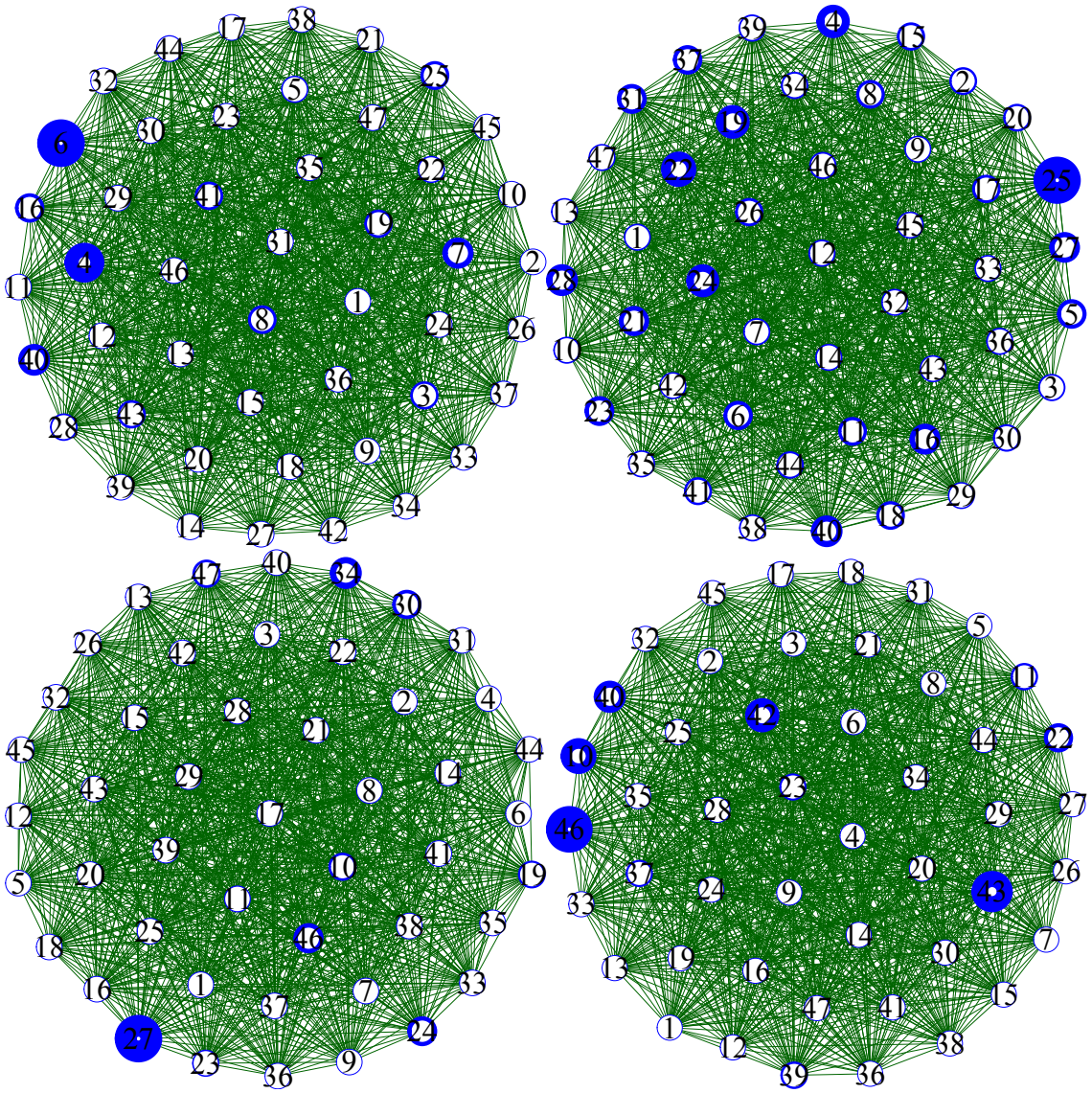


Figure 8.3 – FANOVA graphs corresponding to the four impact scenarii: axial (top) and lateral (bottom), loaded (left) or empty (right). Each circle stands for a variable's main effect while segments are for interaction indices, whose values may be deduced from thicknesses. They are computed based on isotropic Kriging models built from 380 observations for axial impact, 347 in lateral loaded impact and 312 in lateral empty impact. Interactions being of the same order of magnitude, they are very low due to their high number. Also circles may not be seen if they are too thin compared to the few bigger ones.

Table 8.1 – Maximal deformation ($\Delta L/L$) observed on the elements (from the mesh) of the different parts corresponding to variables in the Axial (A.), Lateral (L.), Charged (C.) and Empty (E.) cases. The last column is the percentage of mass of the given component for the reference device; if more than 5% it is marked in bold.

x_i	Ref				Min				Max				Mass
	A. C.	A. E.	L. C.	L. E	A. C.	A. E.	L. C.	L. E	A. C.	A. E.	L. C.	L. E	%
1							0.1						1.29
2		0.1				0.1	0.1			0.5			**5.62**
3		0.3	0.1	0.1	0.1	0.1	0.1			0.5	0.1	0.1	**9.94**
4	0.5	0.3	0.3	0.3	0.1		0.5		0.3	0.1	0.1	0.3	**21.4**
5	0.5	0.3			0.1	0.1			0.5	0.1			**8.11**
6	0.5	0.1	0.5		0.1	0.1	0.3	0.1	0.3	0.1	0.1		**21.1**
7	0.1		0.3		0.1				0.1		0.3		4.37
8	0.3		0.1		0.1				0.1		0.1		**16.8**
9			0.1	0.3		0.1	0.5	0.1			0.1	0.3	0.17
10			0.3	0.3			0.1				0.3	0.5	0.37
11			0.3					0.1			0.3		0.21
12													0.07
13	0.3	0.1			0.1	0.1			0.3	0.1			0.28
14													0.10
15					0.1								0.16
16	0.1	0.1			0.1	0.1			0.3	0.1			0.55
17					0.1								0.20
18					0.1								0.18
19	0.1								0.1				0.55
20		0.3			0.1					0.3			0.21
21		0.1			0.1								0.16
22	0.1								0.3				0.55
23		0.3			0.3					0.3			0.24
24		0.1			0.1					0.1			0.16
25	0.3								0.3				0.50
26		0.1			0.1					0.1			0.21
27		0.1			0.1								0.23
28	0.3				0.1				0.3				0.45
29		0.1			0.1								0.21
30					0.3								0.23
31	0.1	0.1	0.1	0.1	0.1	0.1			0.1	0.1	0.1	0.1	0.46
32					0.1								0.21
33					0.1								0.17
34					0.1		0.1						0.46
35													0.25
36					0.1								0.11
37					0.1			0.1					0.51
38													0.23
39				0.1								0.1	0.16
40	0.1			0.3					0.1			0.3	0.44
41	0.1	0.1						0.1	0.1	0.1			0.31
42				0.3			0.3	0.1				0.3	0.32
43			0.1	0.5								0.5	0.48
44			0.1					0.1	0.1				0.38
45			0.1	0.3		0.1		0.1				0.1	0.21
46			0.3	0.5			0.1	0.1			0.3	0.5	0.42
47			0.5					0.1			0.5		0.28

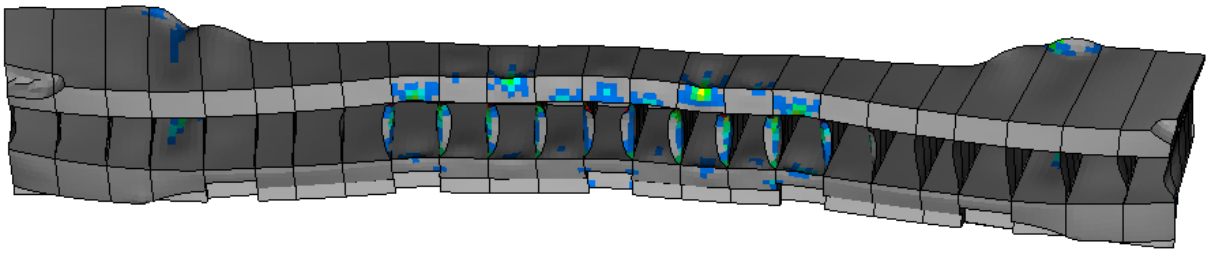


Figure 8.4 – Visualization of the output of the numerical simulator in axial loaded impact for the reference configuration, important deformations range from blue to yellow.

sian kernel. The one dimensional Matérn 5/2 kernel with hyperparameter lengthscale θ, denoting $h = |x - x'|$, is expressed: $k_M(h;\theta) = (1 + \sqrt{5}h/\theta + 5/3(h/\theta)^2)\exp(-\sqrt{5}h/\theta)$. Then extensions for more dimensions are with tensor product, tensor sum or by supposing isotropy (i.e. a one dimensional kernel depending on $\|\mathbf{x} - \mathbf{x}'\|$). In particular, the number of hyperparameters of a tensor product of Matérn 5/2 kernel raises linearly with the number of inputs: $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}, \sigma) = \sigma \prod_{i=1}^{d} k_M(x_i, x_i'; \theta_i)$. Their estimation is troublesome and learning becomes very slow, not only from an informational point of view because of the curse of dimensionality but also computationally, from the cubic cost in the number of inputs (when computing the inverse or Cholesky decomposition of the covariance matrix). Alternatively, with $2d$ hyperparameters, additive models have a linear learning rate in the input dimension but the additivity hypothesis is much more limiting, see e.g. [DGR12]. The corresponding covariance kernel is written $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{i=1}^{d} \alpha_i k_M(x_i, x_i', \theta_i)$. In between are kernels mixing tensor product and sum, relying for instance on the FANOVA decomposition of Equation 8.1. The corresponding covariance model, following [MRCK12], is written $k(\mathbf{x}, \mathbf{x}') = \sum_{l=1}^{L} k_{C_l}(\mathbf{x}, \mathbf{x}')$ with L the number of different cliques and the k_{C_l} the covariance kernel between variables in the clique C_l, $1 \leq l \leq L$.

From these, we tested a variety of parametrizations alternatives offered in *DiceOptim*, *kergp* and *fanovaGraph* to account for the structure of the problem and report here the hypothesis made for the axial loaded configuration:

1. additive kernels with different clique configurations:

   - M3/M3bis: based on geometrical considerations, a clique structure is constructed for the structuring parts (variables 4, 5, 6, 7 and 8) and for the ribs at the center, directly impacted (variables 13, 16, 19, 22, 25, 28, 31, 40 and 41). Each clique is supposed anisotropic for M3 (here with a tensor sum) and isotropic for M3bis.
   - M4/M4bis: again from geometrical considerations, the structuring parts form one

clique, the top ribs one, the middle ribs another and the last for bottom ribs. Each clique is supposed anisotropic for M4 and isotropic for M4bis.

- M13: a fully additive model for reference.

2. tensor product kernels with some shared hyper-parameters θ_i between variables, either automatically, from geometrical or deformation considerations:

   - M5: the geometrical considerations are similar to the ones of M4, this time in a tensor product situation. Length-scales parameters are fitted to each of the structuring parts and there is one for each layer of ribs (top, middle and bottom).
   - M6: the geometrical considerations reflects those of M3 with a tensor product: length-scales are estimated for the structuring parts and variables (13, 16, 19, 22, 25, 28, 31, 40 and 41), all other share the same hyperparameter.
   - M7: from the strains observed on the reference, min and max designs, the 15 variables with most deformation have their own length-scale hyperparameter, while the rest share the same one.
   - M8: from the observed deformations, the structuring parts, axial ribs and ribs with important deformations are regrouped together with a shared hyperparameter, a fourth group is created with the remaining variables.
   - M16/M17/M18/M19: the 4, 6, 8 and 10 variables with biggest main effect from FANOVA graphs have their own lengthscales while the remaining variables are grouped.

3. fully isotropic or anisotropic kernels:

   - Mfull: anisotropic (separable tensor product) Matérn 5/2 kernel with length-scale parameters for each variable
   - Miso: isotropic Matérn 5/2 kernel

Note that models with tensor product covariances have been built with a linear trend, which gives better results that either a constant trend or a trend with linear plus second order terms without interaction. To compare the different models, built with 140 observations, we used either leave one out, or external validation with 40 test samples selected as suggested in [IBFM10] with selection of the designs farthest away from the design samples from Sobol and Halton sequences. We also constructed some regression models as well as polyMARS models to compare. The corresponding results are provided in Tables 8.2. A somewhat disappointing result is that first order linear regression performs better in external validation than most models, probably due to over-fitting with Kriging. Another trend that appears is that models with less hyperparameters perform in general better. Models M16 and M17

have the best overall performance.

Based on Leave One Out or external validation results, the selection of the best model would be different. A preliminary explanation is that design points comes from different stages, i.e design of experiments or optimization runs and they appear to be all located on the boundary of the design space, i.e. with at least one design variable equals to its minimal or maximal possible value. As for the validation test set, from its generation procedure, it is composed of points not on the surface. Due to the high dimensionality of the problem, they are closer to the center of the domain: the average distance to the center after rescaling the inputs to [0,1] is of approximately 3.28 for points in the DOE and of approximately 1.52 for validation points.

Table 8.2 – Leave One Out and external validation results for the different tested models. "+" stands for additive models, "shared θs" for models where some variables are grouped with common length scale parameters. "Geometry", "deformations" and "FANOVA" correspond to several ways of grouping variables, according to their localization on the considered device, to observed deformation on simulated outputs, or to a FANOVA analysis, respectively. "Nb. Hyp." stands for the number of hyperparameters.

	+	Shared θs	From geometry	From deformations	From FANOVA	Nb. Hyp.	LOO error	Validation error	Overall rank
Mfull (cst.)						48	2.61	1.26	14
Mfull (quad.)						48	3.14	6.40	18
Miso (cst.)						2	2.66	1.28	17
Miso (quad.)						2	3.28	6.80	20
Mfull						48	2.27	1.24	11
Miso						2	2.12	1.06	4
M3	X		X			83	1.49	1.97	15
M3bis	X	X	X			19	1.32	1.97	13
M4	X		X			51	1.86	1.28	16
M4bis	X	X	X			8	1.62	**1.03**	3
M5		X	X			12	1.80	1.22	9
M6		X	X			19	1.36	1.26	8
M7		X		X		17	**1.09**	1.46	5
M8		X		X		5	1.65	1.21	6
M13	X					95	8.36	13.7	22
M16		X			X	6	1.24	1.06	2
M17		X			X	8	1.22	1.13	**1**
M18		X			X	10	1.13	1.34	7
M19		X			X	12	1.22	1.46	10
Reg1							2.65	1.21	12
Reg2							4.27	9.58	21
PolyMARS							3.91	2.5	19

In a second stage we also considered the modeling process of the device in the CAD model: indeed, ribs need to have a triangular shape due to molding constraints. They are then modeled with rectangular blocks (PIDs) of decreasing width to get the triangular form.

When a rectangle is too small, i.e. under a certain threshold, it must be removed, which may cause jumps in the objective value. A visual representation is provided in Figure 8.5. It suggests using the volume of the ribs instead of their base length as parameter for modeling. However, this has almost no or a negative effect on the accuracy of the model, hence we have not pushed this idea further.

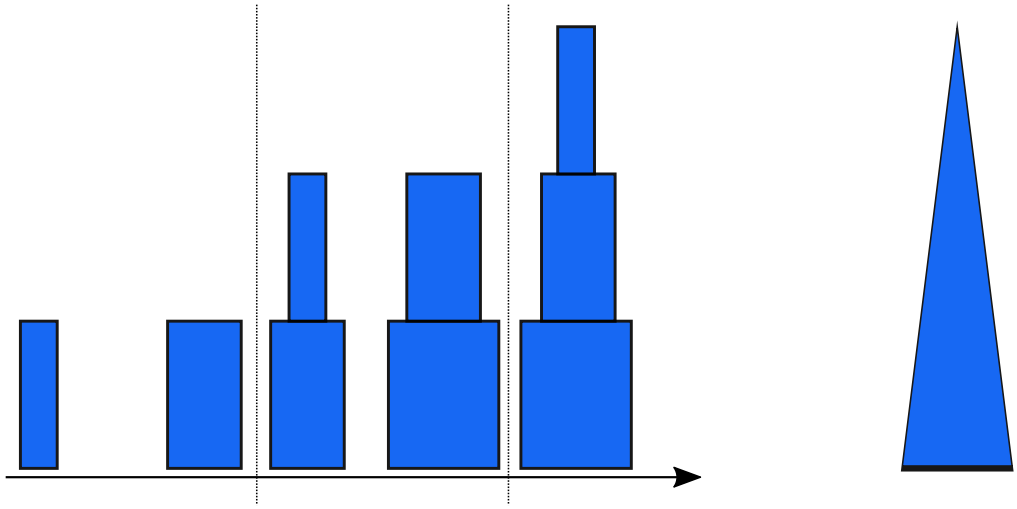


Figure 8.5 – Left: modeled stiffening rib in the CAD software, piloted by the base length. Right: real shape of a rib.

Contrarily to what could be expected, standard GP modeling performs honorably well. Consequently, the tests on optimization have been performed with them.

8.3 Tests using GPareto

8.3.1 Multi-objective and optimization tests

We have here five objectives where one is fairly easy to compute: the mass. It has motivated the *fastfun* option available in *GPareto*, described in Chapter 5, to account for objectives that can be computed quickly.

MOO has been performed with the SMS-EGO criterion, as it is much faster to compute than EHI when the number of objectives increases. Results on the five objectives are difficult to visualize and also to analyze since many points are equivalent in this case. Anyway they are presented in Figure 8.6 along with results of a constrained optimization. A comparison with the 2007 study results shows that they are clearly outperformed by the obtained solutions with *GPareto* in terms of coverage of the Pareto front. Indeed, the results obtained in 2007 are concentrated on a small portion of the Pareto front and they are dominated by those

obtained with *GPareto*: 7 out of the 19 non-dominated points of the old Pareto front are dominated by the new candidate solutions (while previous solutions dominate none of the 34 points of the new Pareto front). The small number of non-dominated solutions is explained partly by the fact that not all impact configurations have been calculated for each design point. Note that insertion in the lateral loaded scenario is almost always under the threshold and this objective could be removed in a real industrial study. This was the strategy adopted in 2007, first to concentrate on axial configurations before also taking into account the lateral ones; this may mitigate partly the lower performance.

Since specifications have been fixed during the development of the car, it is possible to test whether a better solution can be found in a constrained setup. Constrained optimization using the Expected Feasibility Improvement described in Chapter 2 has been used, taking the same initial design of experiments as for the multi-objective study. The new best solution offers an additional mass reduction of the absorber of more than 5% compared to the previous best one, as presented in Table 8.3. The number of evaluations, identical for multi-objective and constrained optimization, is also reduced compared to the budget used in 2007 for axial scenarii, see Table 8.4. Note that as expected, constrained optimization results are much more concentrated on a specific part of the Pareto front.

8.3.2 Uncertainty quantification

The approach proposed in Chapter 3 to quantify the uncertainty on Pareto fronts has been experimented on this test case. The main difficulty is the number of parameters, which is quite high when it comes to performing conditional simulations. This is why we applied the procedure detailed in Algorithm 13, Appendix B, based on optimization of the simulation points. The obtained result is presented in Figure 8.7 for 2 objectives: the mass and one intrusion. Due to the low number of observation points (200) relatively to the dimension, the remaining uncertainty and position of the Vorob'ev deviation are optimistic since they predict the location of the Pareto front well behind the non-dominated points. More runs of optimization are needed to assess if this optimism is realistic, in the sense that solutions on the Vorob'ev expectation are reachable.

8.4 Tests under the REMBO paradigm

It has been observed at the beginning of this chapter that the REMBO method should work pretty well in this case. But if this is true for the four crash objectives, it must be mitigated for the mass: indeed, from Table 8.1, all variables have an influence on the mass, even if relatively small. It resulted in the rather conservative choice of $d = 18$ as dimension

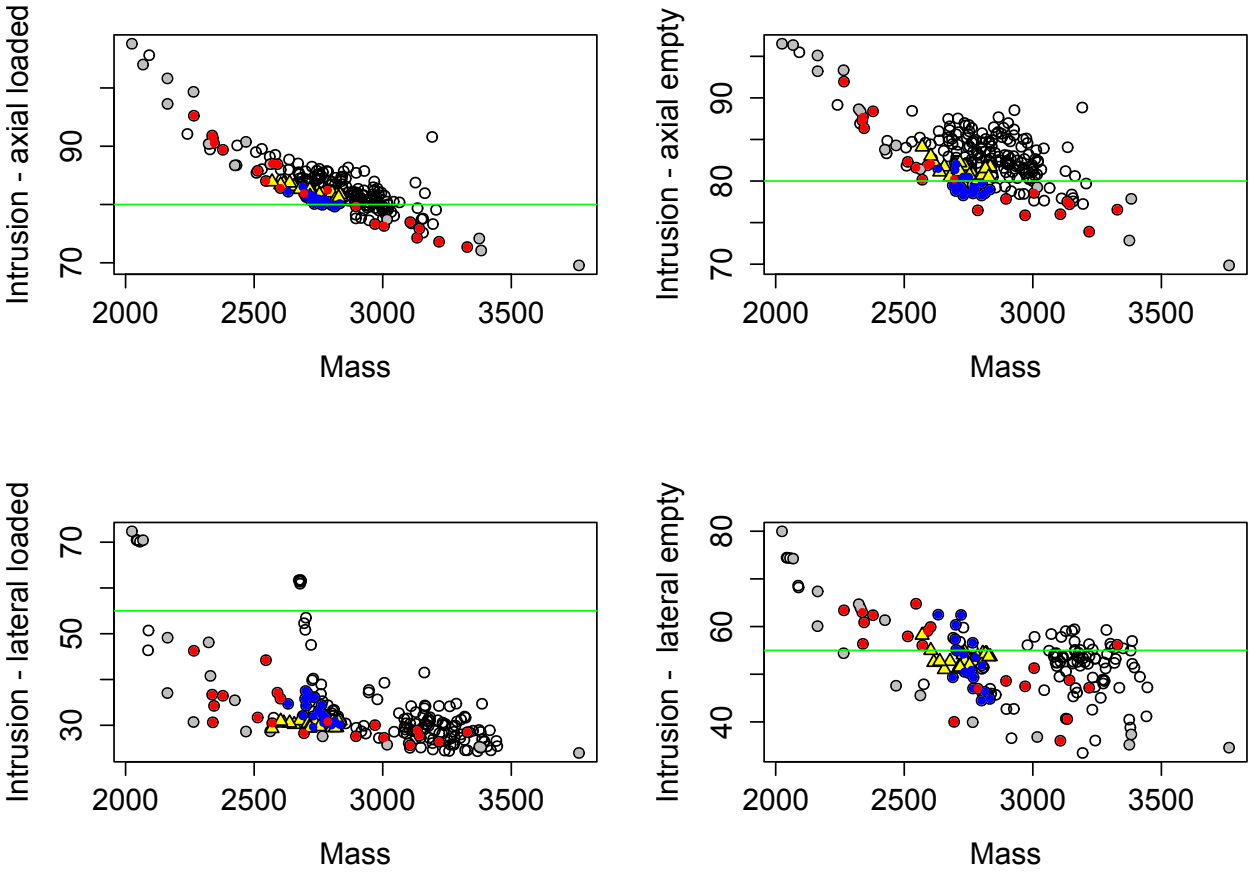


Figure 8.6 – Optimization result for multi-objective (red) and constrained optimization (blue), with (shared) design of experiments represented with black circles, filled in gray if all objectives are evaluated. Coordinates in the five dimensional space can be deducted from the mass value. Triangles represents the responses obtained from a previous study in 2007. The green lines represent the maximum values of the intrusion as defined in the specifications.

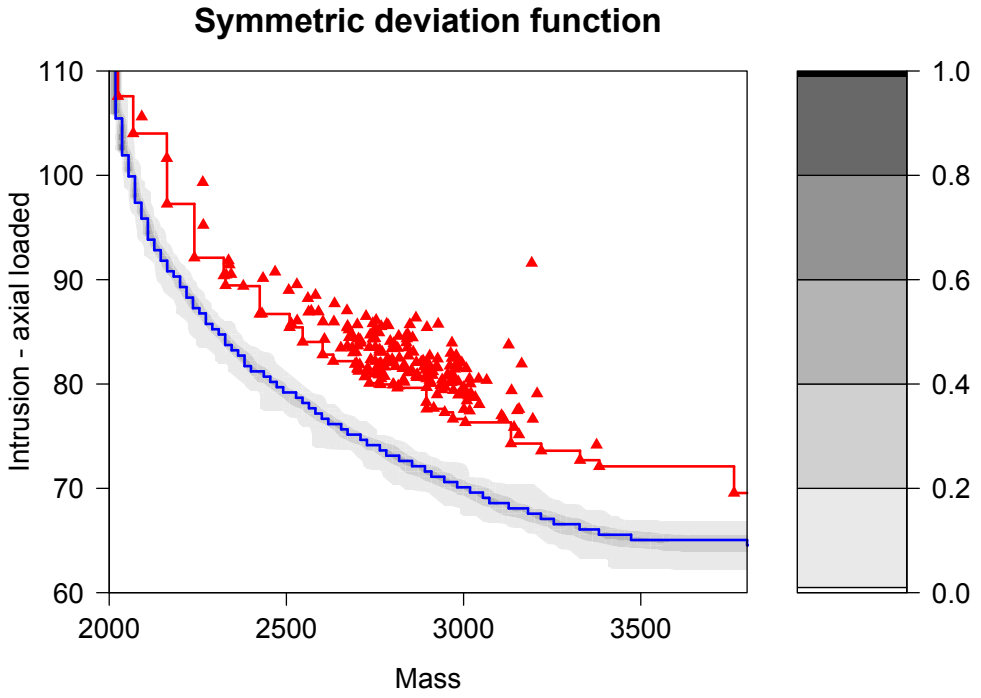


Figure 8.7 – Uncertainty quantification on the axial loaded impact configuration, with 47 parameters from 50 simulations at 8000 points, with optimization of the locations. The Vorob'ev expectation is in blue and the observations are denoted with red triangles.

for $\mathcal{Y}$, let say low dimensionality of 3 for each lateral impact, 4 for each axial one, and 4 for the mass (some can be shared). It was also motivated by a drastic reduction of the evaluation budget, i.e. by taking 120 + 20 as evaluation budget instead of 286 (maximum over the different objectives), which is between five and ten times the input dimension as initial number of evaluations to build the design of experiments, as also considered in [Sve11].

Similar experiments as in the previous section have been conducted, this time within the REMBO procedure. The multi-objective results are presented in Figure 8.8, and a summary of the constrained ones in Table 8.3 with corresponding number of evaluations in Table 8.4. None of the 19 points of the old Pareto front are dominated by the ones of REMBO, while 7 out of the 97 non-dominated points with REMBO are dominated (32 out of the 97 are dominated by the Pareto front with *GPareto* without REMBO). The best explanation is that for the mass, all the variables have an influence, even if it is one percent. The constrained REMBO optimization provides a bit more than 1% of gain over the previous best solution but with a halved budget of evaluations (considering the axial configurations). Note that the best feasible solution found in 2007 in far from the constraint boundary while better solutions are possible if relaxing the thresholds. Also, the constrained criterion used with *GPareto* and *GPareto*+REMBO is known for not often adding points close to the boundary

either, as mentioned e.g. in [Pic14], thus better results could possibly be obtained with more efficient ones.

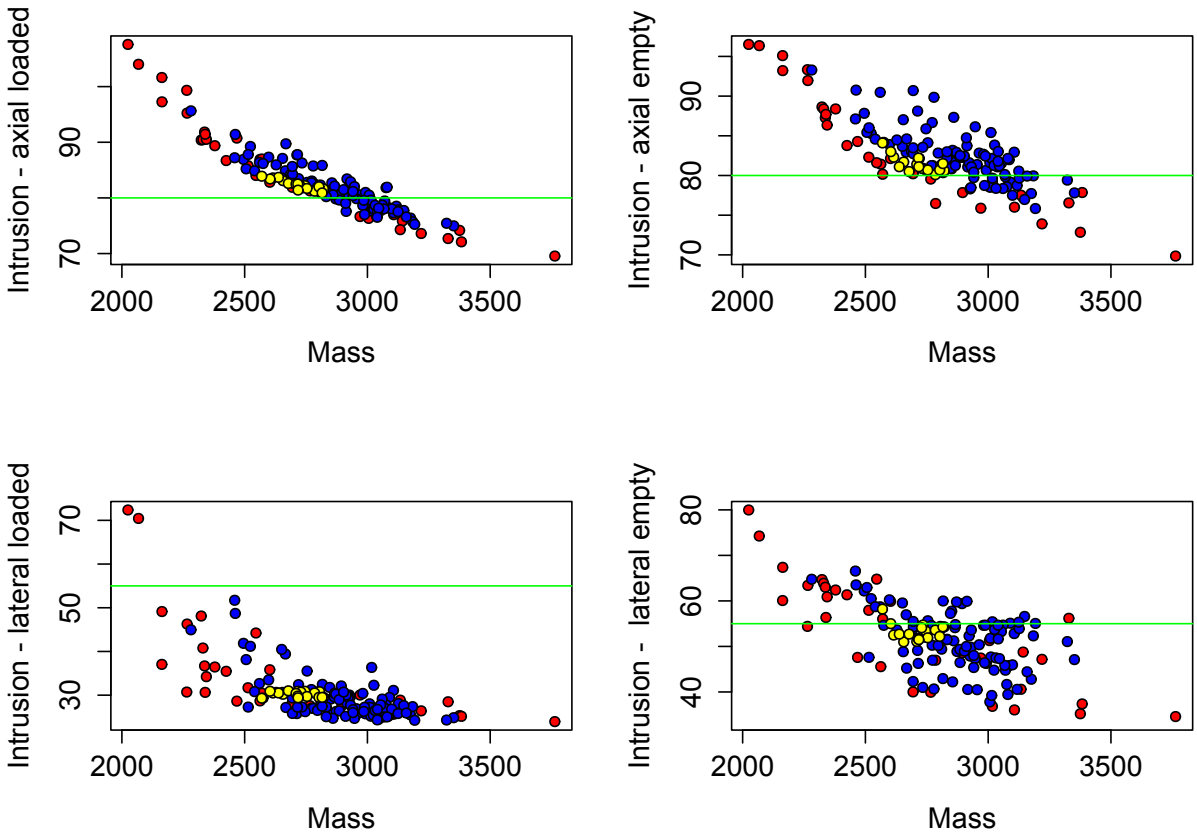


Figure 8.8 – Pareto optimal points obtained with GPareto combined with REMBO (blue) compared to those obtained with GPareto only (red). Pareto optimal points obtained in 2007 are also represented in yellow. Coordinates in the five dimensional space can deducted from the mass value. The green lines marks the specifications used in constrained optimization.

The question of the choice of the low dimension need further investigation, even if selecting it from maximal budget considerations seems suited, the complex question of determining what is lost if taking a lower one is still of interest. Also, from the specific configuration of the problem, with the mass monotonically increasing with the thicknesses, it raises the question of adapting the selection of the random matrix $\mathbf{A}$ to this case. Indeed, the random embedding should intuitively cover in priority configurations with low mass.

8.5 Concluding remarks on the test case

This test case in car crash-worthiness is, by its number of input variables, quite challenging. Notwithstanding this difficulty, the performed sensitivity analysis is in line with the state-

Table 8.3 – Comparison of number of results of constrained optimization, abbreviations are Axial (A.), Lateral (L.), Charged (C.), Empty (E.), constrained (cstr.).

	Mass (g)	A. C.	A. E.	L. C.	L. E.
Specifications		$\leq$ 80mm	$\leq$ 80mm	$\leq$ 55mm	$\leq$ 55mm
Reference	3375	74.1	72.9	25	35
2007 study (1)	2820 (-16.4%)	80.2	78.8	30.3	54.3
2007 study (2)	2730 (-19.1%)	81.0	80.9	28	51.4
2007 study (3)	2847 (-15.6%)	80.1	79.3	29.0	52.4
2007 study (4)	2942 (-12.8%)	78.4	77.6	26	33.4
Cstr. GP (1)	**2762 (-18.2%)**	80.0	78.7	34.4	50.9
Cstr. GP (2)	2734 (-19.0%)	80.1	78.7	36.2	50.5
Cstr. GP (3)	2729 (-19.1%)	81.1	78.3	29.6	54.2
Cstr. REMBO (1)	2749 (-18.6%)	80.8	81.0	27.2	54.2
Cstr. REMBO (2)	2904 (-14,0%)	79.0	80.0	27.1	53.3
Cstr. REMBO (3)	2885 (-14,5%)	80.2	80.1	27.4	55.0

Table 8.4 – Comparison of the numbers of evaluations

Number of experiments	Axial loaded	Axial empty	Lateral loaded	Lateral empty
2007 study	286	286	62	92
Initial design of experiments	180	180	147	112
Constrained GP-optimization	210	210	177	142
Constrained REMBO	140	140	140	140

ment that, in most applications, the number of important variables is actually low, see e.g. [WZH+13], [IL15]. A customization work on covariance kernels demonstrates the interest of adapting k to the task at hand, with promising perspectives offered by other modeling options such as in [Duv14] or [GRS+14]. Methods such as in [MIDV08] with a progressive estimation of the hyperparameters and regression functions may also be of interest. From an optimization point of view, the results obtained with GP-based methods clearly outperform those of a previous study, which was based on optimizing regression and PolyMARS models relying on the NSGA-II algorithm [DPAM02]. The REMBO [WZH+13] algorithm also showed its adequacy for reducing the number of runs in high dimensional settings, while providing competitive results. Future work options include the use of customized kernel within optimization as well as developing REMBO further.

Conclusion and future works

Chapter 9

Conclusion and future works

In this thesis, we mainly study the problem of multi-objective optimization of expensive black-box functions. These problems arise daily in an industrial context and, as an example, we detail experimentation on a test case in car crash-worthiness.

Existing works on Kriging-based multi-objective optimization methods already provide efficient solutions to obtain Pareto optimal solutions. Yet, the approximation of the Pareto front obtained is only discrete. Two main contributions of this thesis aim at providing a continuous picture of the Pareto front. One proposed solution in Chapter 3, adapted from the work of [Che13], is to rely on Gaussian process conditional simulations along with concepts from random closed sets theory to capture the variability of the Pareto front given by the surrogate models. Resorting to conditional simulations to compute the attainment function is a hindrance in computational terms, especially if considering the associated stepwise uncertainty reduction criterion. In a second contribution, this stage is replaced by an estimation of the multivariate cumulative distribution of the surrogate models using copulas, see Chapter 4. Both approaches provide practitioners with a Pareto front approximation they can rely on when deciding to stop, intensify or guide the optimization process.

A possibly high number of variables has been identified as one of the main challenges for surrogate based methods in general. Whether in terms of learning rate or practical (often computational) tractability, a change of viewpoint may be needed to take these methods further in terms of dimensionality. A potential break-through has been performed with the REMBO algorithm [WZH+13], using random embeddings of a low dimensional subspace into the initial high dimensional one. We contributed to the study of this method and proposed original extensions addressing several key issues. First, in Chapter 5, with a covariance kernel that avoid pitfalls associated with the originally proposed ones. Then, in Chapter 6, we worked on the difficult problem of selecting a proper low dimensional domain in order to

avoid missing an optimum and at the same time not impeding the optimization process by considering a too large domain. We showed a clear improvement of the performance that, in particular, enables to apply it with multi-objective optimization. It paves the way to more developments, possibly in sensitivity analysis or in hybridization with active learning of linear embeddings (see e.g. [GOH13], [DKC13]).

To propagate and spread the use of these methods in engineering, the *GPareto* package has been released to complement *DiceKriging* and related packages by multi-objective optimization. Chapter 7 consists in a tutorial of *GPareto*. In addition, to ensure ourselves of the applicability of the contributions in an industrial context, the methods have been tested on a Renault case study in Chapter 8, outperforming older results, both in terms of fitness of solutions and in number of evaluations required to reach them.

Moreover, some attempts to tune kernels have been performed and they provide one of the most important directions for future research. Indeed, our contributions bring some pieces to the puzzle of general Bayesian optimization and we believe in the potential offered by combinations with some of the recent developments, in particular related to surrogate modeling or acquisition functions. As an example, a promising perspective for REMBO is to enhance the surrogate model by accounting for instationnarities that are present, due to both the embedding process and the black-box function optimized. Several recent works, e.g. [SSZA14], [AWdF14], [MC15], propose various possibilities to learn these instationnarities, giving more flexibility and showing improved performance. Extensions of Gaussian processes are also a current trend, with options such as Deep GPs [DL13] or Student-t processes [SWG14] that could replace the more standard GP models we have used.

Remaining obstacles are possibly a lack of user-friendly solutions for flexible modeling accounting for the complexity in test cases. One example occurs with variables of mixed nature, e.g. continuous or discrete [ZQZ11], as well as with nested parameters, e.g. with several alternative components in a device that all have their own parameters. On the other hand, opportunities are offered with integration of physical behavior with latent force models [TL10], by exploiting multi-fidelity models or gradient observations that are now given by adjoint solvers or from approximations, see e.g. [Fro14], [GJGM15]. Finally, ongoing works on other test cases in car crash-worthiness raise the question of taking into account noisy simulations with both multi-objective infill criteria like e.g. in [KWE+15] and REMBO.

Appendices

Appendix A

A very fast approximation of the multipoint Expected Improvement

First parallel versions of the Expected Improvement were either analytical for batches of two points or based on heuristics, such as Kriging believer (KB) and Constant liar (CL) [Gin09]. The exact formula for the multipoint expected improvement at q points $\mathbf{x}_1, \ldots, \mathbf{x}_q$, $qEI(\mathbf{x}_1, \ldots, \mathbf{x}_q) = \mathbb{E}\left(\left(\max_{i\in\{1,\ldots,q\}} Y_i - T\right)_+\right)$ was defined in [Sch97] but derived later in [CG13], while its gradient was expressed recently in [MCG15]. They are implemented in the R package *DiceOptim* [RGD12].

The exact formula allows optimizing in one round, but the computation is quickly hindered by calls to Φ_{q-1}, the cdf of the centered multivariate Gaussian distribution, which becomes computationally demanding as q increase. Here we propose an approximation of the exact qEI which is linear in q, with simple expressions, thus allowing intensive optimization of the qEI, or large values for q.

The qEI may be rewritten $\mathbb{E}\left(\max(Y_1 - T, \ldots, Y_q - T, 0)\right)$, i.e. the first moment of the maximum of correlated Gaussian variables. A fast approximation of this first moment was proposed by [Cla61] and as been used e.g. in [DVD13]. In particular, for three Gaussian random variables ξ_1, ξ_2 and ξ_3, to approximate $\mathbb{E}\left(\max(\xi_1, \xi_2, \xi_3)\right)$, the principle is simply to write $\max(\xi_1, \xi_2, \xi_3)$ as $\max(\xi_1, \max(\xi_2, \xi_3))$ and suppose that $\max(\xi_2, \xi_3)$ is normally distributed. This is of course not the case but the error made is very small.

In more details, denote $\mu_i = \mathbb{E}\left(\xi_i\right)$, $\sigma_i = Var\left(\xi_i\right)$, $\rho_{i,j} = Cor\left(\xi_i, \xi_j\right)$, $1 \leq i, j \leq 3$ and additionally $a^2 = \sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho_{1,2}$, $\alpha = (\mu_1 - \mu_2)/a$. If ξ_1 and ξ_2 do not fulfill

$\sigma_1 - \sigma_2 = \rho_{1,2} - 1 = 0$, the following results are given in [Cla61]:

$$\mathbb{E}\left(\max(\xi_1, \xi_2)\right) = \mu_1\Phi(\alpha) + \mu_2\Phi(-\alpha) + a\phi(\alpha) \tag{A.1}$$

$$\mathbb{E}\left(\max(\xi_1, \xi_2)^2\right) = (\mu_1^2 + \sigma_1^2)\Phi(\alpha) + (\mu_2^2 + \sigma_2^2)\Phi(-\alpha) + (\mu_1 + \mu_2)a\phi(\alpha) \tag{A.2}$$

$$Cor\left(\xi_3, \max(\xi_1, \xi_2)\right) = \frac{\sigma_1\rho_{1,3}\Phi(\alpha) + \sigma_2\rho_{2,3}\Phi(-\alpha)}{\sqrt{\mathbb{E}\left(\max(\xi_1, \xi_2)^2\right) - \mathbb{E}\left(\max(\xi_1, \xi_2)\right)}} \tag{A.3}$$

Then, it is possible to approximate $\mathbb{E}\left(\max(\xi_1, \xi_2, \xi_3)\right)$ from equation A.1, by applying equations A.1 and A.2 to $\max(\xi_2, \xi_3)$ and equation A.3 to get $Cor\left(\xi_1, \max(\xi_2, \xi_3)\right)$. With four or more normal variables, the idea is simply to successively apply the approximations, $\mathbb{E}\left(\max(\xi_1, \xi_2, \xi_3, \xi_4)\right) = \mathbb{E}\left(\max(\xi_1, \max(\xi_2, \max(\xi_3, \xi_4)))\right)$ and so on. To get an approximated qEI, it remains to treat 0 as a normal variable of mean 0 and standard deviation 0 (or a small constant if necessary for computational reasons), uncorrelated with all other variables.

The worst case for the approximation, as discussed in the corresponding paper, is when ξ_i and ξ_j have very close first and second moment. It could motivate to switch the order of variables $\mathbf{x}_i$, to avoid this situation. Anyway, intuitively the values of the qEI will increase when the $\mathbf{x}_i$, $1 \leq i \leq q$ are spread. Also this criterion is more intended to provide a simple mean to filter or massively optimize the qEI, before using the exact expressions in a second stage. In Figure A.1 we compare the values given by the exact qEI and several approximations included our proposition. First the difference using the *fast* option in *DiceOptim*[1] cannot be seen on the graphs. The Monte-Carlo estimation is also quite reliable while the proposed fast estimation gives a crude estimate of the qEI, but at a much lower computational cost: see Table A.1. Finally remark that the mean of the qEI is higher when q is greater, as can be expected.

Table A.1 – Average computation times (milliseconds) of the qEI for different q and methods over a hundred repetitions. With $q > 20$, the function `qEI` of the R package *DiceOptim* (v.1.5) is no longer available. It also has an option `fast` to save some time on computations.

q	3	5	10	20	30	50
`qEI`	2.58	7.66	21.3	163	-	-
`qEI` (option `fast = FALSE`)	3.98	11.6	63.1	851	-	-
`qEI` (MC)	24.8	27.0	32.7	51.4	82.8	137
fast qEI	1.09	1.13	1.28	1.49	1.62	2.21

[1]This option relies on a reformulation of the multipoint EI as a sum of derivatives estimated using finite differences, resulting in a reduced numerical complexity, see [Mar14].

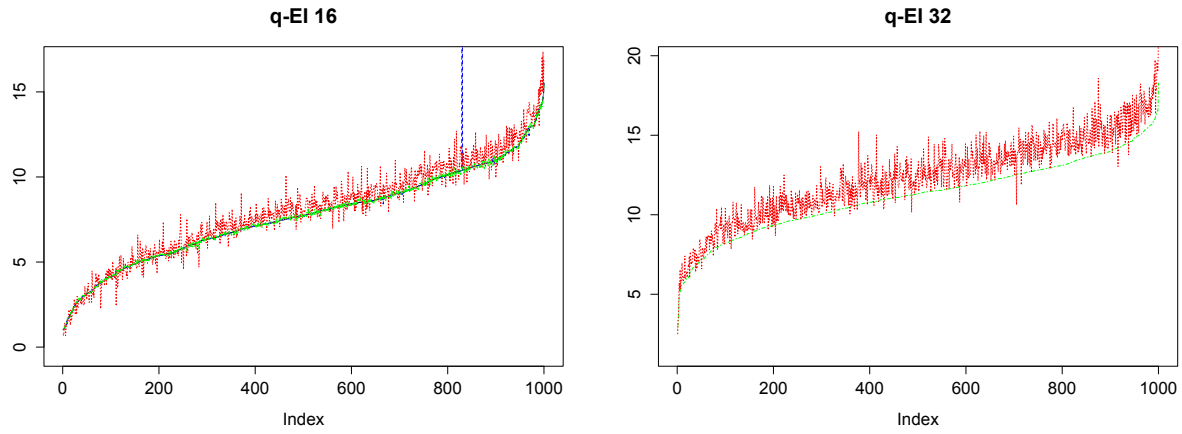


Figure A.1 – Comparison of the values of the qEI with four different algorithms with $q = 16$ and $q = 32$ over a thousand batches of size q, given by a random Latin hypercube (to avoid the situation of points to close to each other), and sorted in ascending order. The exact qEI from *DiceOptim* is in solid black, the exact qEI with option `fast = TRUE` in dashed blue, the Monte Carlo estimation (with a sample of size 10000) in dotted green and the fast approximation in dashed-dotted red.

Appendix B

Ongoing work on Chapter 3

B.1 Associated optimization criterion

The quantification of uncertainty on Pareto fronts was initially intended to provide a possible stopping criterion of the optimization process as well as a practical tool for visualization, while optimization was conducted e.g. with Expected Hypervolume Improvement. Nevertheless, the Vorob'ev deviation may also be used as an infill criterion since it quantifies the uncertainty on the Pareto front $\mathcal{P}$. The criterion of [Pic13], detailed in Chapter 2 aims at reducing the volume of the excursion set below the Pareto front, that is the volume in the input space of the probability of being non-dominated. The main difference is that in our case the uncertainty is measured directly in the objective space instead of the design space.

The SUR paradigm, introduced in Chapter 2, involves adding a new observation where the expectation of the future associated uncertainty is minimal. Since the Vorob'ev deviation (VD) measures the uncertainty on the Pareto front location, the corresponding criterion to minimize is:

$$J(\mathbf{x}_{n+1}) = \mathbb{E}\left(\mu(\mathcal{Q}_{\beta^*_{n+1}}\Delta\mathcal{Y}_{n+1})|\mathcal{A}_n\right), \tag{B.1}$$

where $\mathbf{x}_{n+1} \in \mathrm{E}$ is a new candidate location. With the next observation, both β^*-quantile $\mathcal{Q}_{\beta^*}$ and attained set $\mathcal{Y}$ will be modified and thus need to be recomputed since it seems unreachable to obtain an analytical expression.

Monte Carlo estimation with conditional simulations of the corresponding Vorob'ev deviation requires conditional simulations for each draw of $Y(\mathbf{x}_{n+1})$. In comparison, Monte Carlo estimation of multi-objective EI criteria without analytical expression needs uniquely draws of $Y(\mathbf{x}_{n+1})$ and the estimation of the corresponding improvement. Furthermore, due to the Pareto dominance relationship for several objectives, the computational savings and analytical formulas available for excursion sets [Che13] does not seem to transpose in this

case. Recently in [CEG14], a method to update conditional simulations has been proposed that enables computational savings. Indeed, the authors show that a set of conditional simulation can be updated to integrate a new observation at a reduced computational cost if this response has been previously simulated. This is the main ingredient in Algorithm 11 to estimate J.

Algorithm 11 Estimation of $J(\mathbf{x}_{n+1})$

Input: p conditional simulations knowing $\mathcal{A}_n$

1: **for** $i \in (1, \ldots, q)$ **do**
2: Sample $\mathbf{z} \sim \mathcal{N}(m_n(\mathbf{x}_{n+1}), s_n(\mathbf{x}_{n+1})^2)$.
3: $\mathbf{Y} \leftarrow$ Update of the p conditional simulations with $\mathbf{z}$.
4: $VE \leftarrow$ Estimation of Vorob'ev expectation from the RNP sets extracted from $\mathbf{Y}$.
5: $VD^{(i)} \leftarrow$ Estimation of the Vorob'ev deviation of $\mathbf{Y}$.
6: **end for**
7: $J(\mathbf{x}_{n+1}) \approx \frac{1}{q} \sum_{i=1}^{q} VD^{(i)}$.

Although easily parallelizable, this is still quite an expensive criterion, especially when the numbers of simulations, simulation points and samples of the response at $\mathbf{x}_{n+1}$ increase. It may only be used for now on very expensive problems which takes days to compute. Still, a comparison with the EHI and random sampling is provided on the same problem as in Chapter 3. Again a ten points maximin LHS is used, and ten points added sequentially. The choice of a new point is done on a discrete 51×51 grid, corresponding also to conditional simulation locations. The values of the Vorob'ev deviation at each iteration are reported in Figure B.1. Despite the fact that parameter re-estimation might cause unfavorable effects to it, the SUR criterion (Equation B.1) is the most efficient to reduce VD while the EHI criterion performs also well. The final symmetric deviations are displayed in Figure B.2 to provide intuition on the different behaviors: the EHI criterion tends to add points on or very close to the Pareto front, while the SUR criterion is more exploratory. As is, knowing precisely the expected location of the Pareto front but with known observations only quite far from it may not be very appealing to a practitioner, who would prefer having points on it. Since EHI is faster and gives more usable results, for the VD-based SUR criterion to become more competitive, one needs to be able to find a solution in the design space corresponding to a given point on the Vorob'ev Expectation, which is discussed in the following.

B.2 Toward interactive optimization

The Pareto front estimation, along with the corresponding uncertainty, provides practitioners with useful information on where possible improvements may be found. Unfortunately, even if it would be tempting to select new points directly from the estimation, there is no straight-

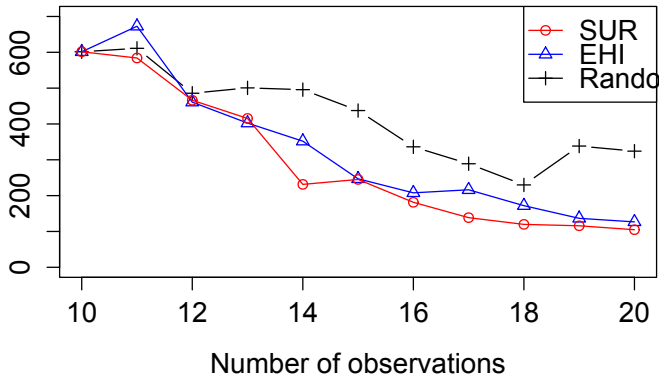
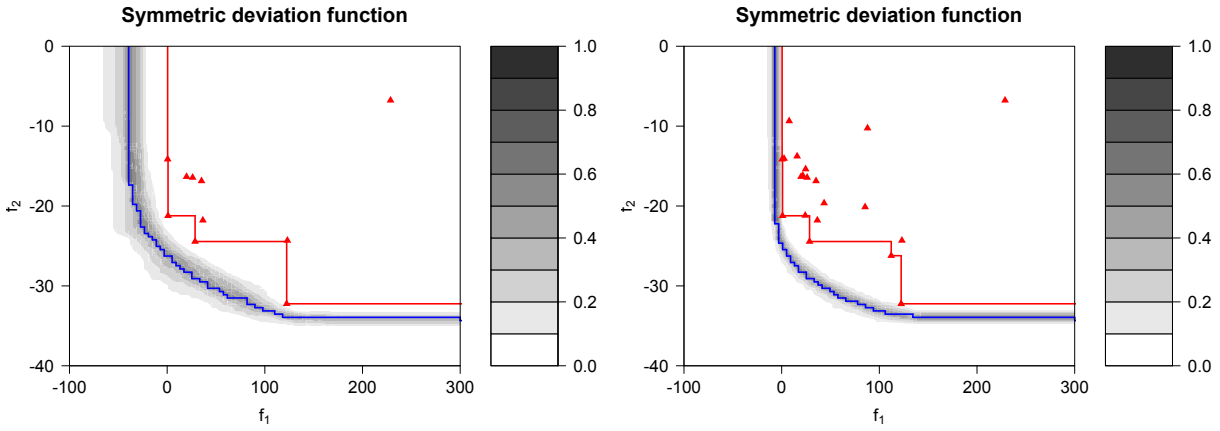
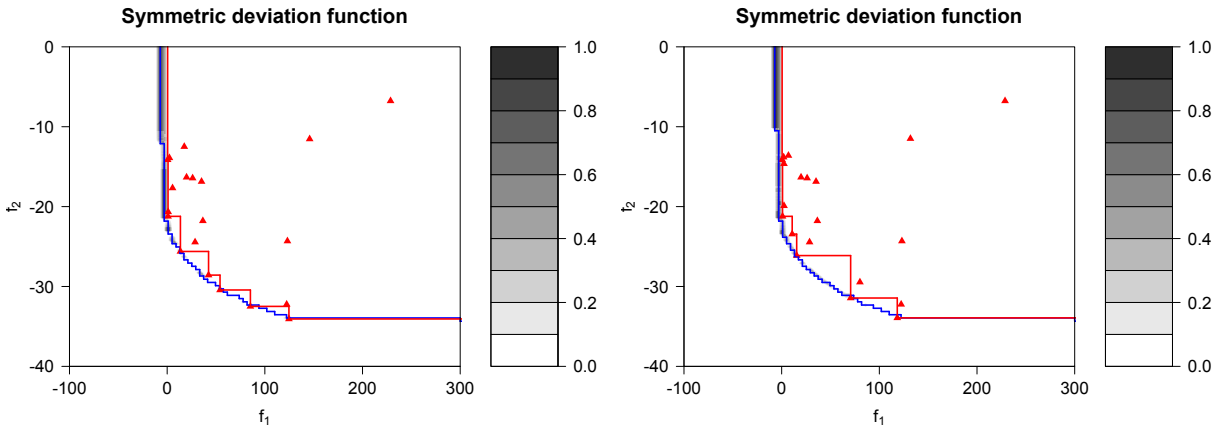


Figure B.1 – Monitoring of the symmetric difference with respect to the true Pareto front while adding ten points sequentially with either random sampling, EHI or the SUR criterion.



(a) Initial state $n = 10$

(b) Random sampling



(c) EHI criterion

(d) SUR criterion

Figure B.2 – Symmetric deviation function before and after optimization (i.e. $n = 20$), for the different sampling strategies.

forward way from the objective space to come back to the variable space. An algorithm to perform this reverse mapping has been proposed by [GF14], where the authors take a convex set in the objective space from the non-dominated points returned by an optimizer, and then they use an RBF neural network for modeling the mapping. Focusing on the relationships between design and objective space is also conceptualized by *innovization* [DS06], which attempt to decipher design rules from the optimal solutions found.

When using GPs as surrogates, a simple solution to find a pre-image to $\mathbf{v} \in \mathbb{R}^m$, here supposedly on the Vorob'ev expectation, is to maximize the probability of dominating $\mathbf{v}$, which is the multi-objective probability of improvement with respect to $\mathbf{v}$. Another option is to maximize the probability of being in a hypercube of side $\delta > 0$ centered on $\mathbf{v}$. There are several drawbacks and limitations with these approaches:

- the multi-objective probability of improvement has been shown to perform rather poorly as an infill criterion, see e.g. [Par12], and selecting δ is by no means straightforward;
- with many variables, this task may be quite difficult;
- last but not least, it does not use at all the properties of the Pareto set.

Indeed, when focusing on the Pareto set, many authors (e.g. [JS03], [LZ09], [ZZJ08], [Lov12] or [BDD14]) have used that under some "mild" regularity conditions, derived from the Karush-Kuhn-Tucker optimality conditions, e.g. in [Hil01], the Pareto set is a piecewise connected manifold of dimension $m - 1$.

So we propose here to take advantage of the special structure of the Pareto set and benefit from the Conditional Pareto Sets (CPS) on which we did not put a focus until now. Empirically, when adding new observations by sequential optimization, the CPF and estimated Pareto front are expected to get closer to the true one and we can also expect a similar behavior for the Conditional Pareto Sets with respect the true Pareto set. The problem is then to be able to find a piecewise connected set from a number of points surrounding it. One such method is the Gaussian Process Latent Variable Model (GP-LVM) [Law05], which performs non-linear principal components analysis. GP-LVM basically learns a GP mapping from a latent space to the data space, here E, where positions of points in the latent space corresponding to those in the data space are determined by likelihood maximization. It allows representing the data set, here composed of the CPS, in the latent space and, more importantly, to map from this latent space to the original data space, with the uncertainty associated to this prediction, given by the GP. We thus are able to go back and forth from design to objective spaces with the GP models of f and the GP-LVM model. To take into

account the Gaussian distribution of the prediction of the inputs given by the GP-LVM mapping from latent to variable space, we use the Monte-Carlo approximation detailed in [Gir04]. That is, suppose that $\mathbf{x}$ is corrupted with some Gaussian noise $\boldsymbol{\varepsilon} = \mathcal{N}(0, \Sigma_{\mathbf{x}})$, i.e. $\mathbf{x} = \mathbf{u} + \boldsymbol{\varepsilon}$ such that $\mathbf{x} \sim \mathcal{N}(\mathbf{u}, \Sigma_{\mathbf{x}})$. Then the mean prediction or standard deviation at $\mathbf{x}$ is the average of the mean predictions (standard deviation respectively) of the GP on samples of $\mathcal{N}(\mathbf{u}, \Sigma_{\mathbf{x}})$.

As a proof of concept, we propose Algorithm 12 and an illustrative example. To keep things displayable, we only present here results on a bi-objective problem with three variables. This problem, the F3 function in [LZ09], has been designed to have a more complicated Pareto set than usual toy functions. The design of experiments is composed of 50 points from a maximin LHS design and 70 points were added with Expected Hypervolume Improvement. We use the *vargplvm* R package, available on the repository of the authors of the method, to perform the analysis. This variant of GP-LVM applies variational Bayesian integration of the latent variables instead of maximum a posteriori [TL10]. Again for a visualization motive, the latent dimension is set to two instead of one ($=m-1$). The obtained results are quite promising: first the Pareto set is relatively well approximated, and the CPF points (green) have the desired behavior around the true Pareto set. Then, the latent space seems to have found the spiral structure of the Pareto set, as can also be noticed from the mapping back to the input space (blue points). Finally, the proposed method to find designs in E which are predicted to be on the Vorob'ev expectation is quite conclusive: the black points are very close to the Pareto front approximation and with low variance. Note that compared to directly searching in E a solution with maximal probability of mapping to a hypercube centered on a target, the search is conducted in the latent space which is of smaller dimension.

Algorithm 12 Interactive optimization combining GP and GP-LVM models

Input: GP models $(Y_1, \dots Y_m)$ of the objective functions with n observations and conditional simulations

1: Compute Conditional Pareto Fronts and Conditional Pareto Sets of the simulations.
2: Compute Vorob'ev expectation VE from the CPFs.
3: Fit GP-LVM model from all CPS points to obtain $\Xi : \mathbb{R}^{m-1} \to$ E, the mapping from latent to data space.
4: Select a target point $\mathbf{v}$ on the VE.
5: Find in the latent space a potential candidate $\mathbf{u}^*$ mapping to $\mathbf{v}$, e.g. by maximizing the probability that $(Y_1(\Xi(\mathbf{u})), \dots, Y_m(\Xi(\mathbf{u})))$ belongs to the hypercube of center $\mathbf{v}$ and radius $\boldsymbol{\delta}$, with $\boldsymbol{\delta} \in \mathbb{R}^m$.

Output: $\Xi(\mathbf{u}^*)$ and optionally, $(Y_1(\Xi(\mathbf{u}^*), \dots, Y_m(\Xi(\mathbf{u}^*))$

There are several possible extensions to this idea. In GP-LVM [Law05], the main focus is on keeping points distant in the data space distant in the latent space; the modified version

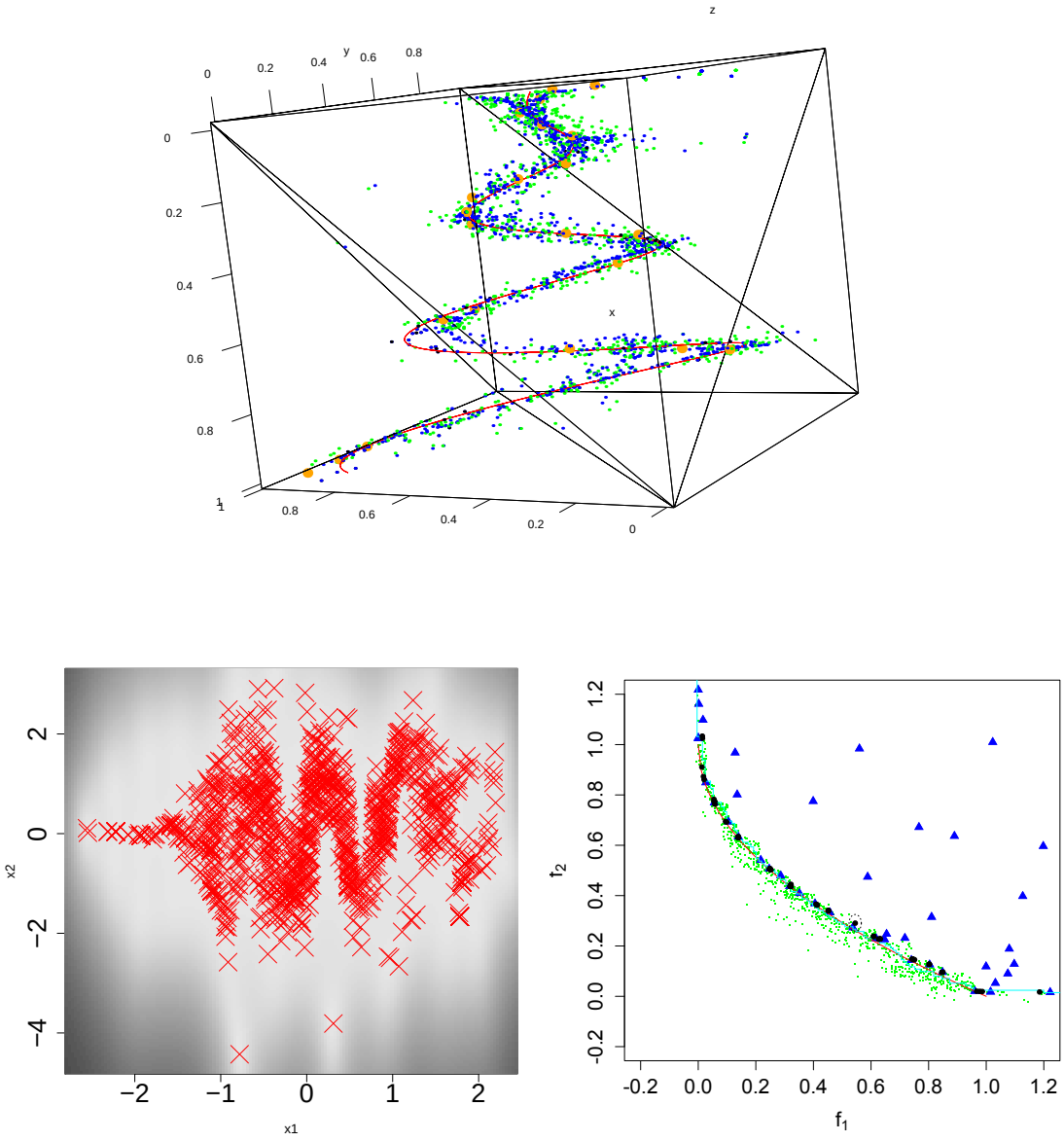


Figure B.3 – Top: parameter space, with non-dominated points of the observations (big orange points), non-dominated points obtained from 50 simulations at 2000 uniformly selected locations (green points), image of the points in the latent space corresponding to the green points (blue points) and true Pareto set (red line). Bottom left: latent space with CPS (red crosses), the uncertainty is represented in gray scale (white is better). Bottom right: objective space with observations in blue, the Vorob'ev expectation (cyan step line), non-dominated points of the CPF (green small points), image of points found in the latent space with predicted image on the Vorob'ev expectation (black point with uncertainty in dotted ellipses) and real Pareto front (red line).

of [LQC06] introduces back constraints (through a constrained likelihood) to also preserve some local distances. Then points close in the data space should somehow be close in the latent space. In addition, the latent space could be used to sample new points, to perform the optimization of criteria too expensive if applied in the original design space or to select next simulation points.

B.3 Sequential generation of conditional simulation points

Until now, we have had conditional simulations performed on a grid in numerous situations. This becomes quickly intractable when the number of variables increases and alternative approaches are needed. Given a simulation on p points of a Gaussian process conditioned on the n observations of the expensive function, it is possible to augment this simulation at q new designs, this time conditioned on both n real observations and p simulated ones. In this sense, we propose to optimize simulated realizations, where evaluating a conditional simulation at $\mathbf{x}$ amounts to performing a simulation conditioned on the n observations of f and on all previously simulated observations. This way, re-evaluating a point still give the same result[1]. In case of a population based optimization method, the procedure is briefly summarized in Algorithm 13. An example is provided in Figure B.4, where the simulation points are the points visited by an NSGA-II algorithm [DPAM02] optimizing the corresponding simulation. As a result, areas of interest are more explored than regions of the design space mapping to dominated points, with a higher density of points close to Pareto optimal regions.

Algorithm 13 Selection of simulation points with a population based optimization

Input: Initial population for the algorithm, with corresponding simulated values
1: **while** Number of simulation points < budget **do**
2: Get new population from the evolution mechanism of the optimization method
3: Simulate the corresponding value(s) conditionally on the previous ones
4: **end while**
Output: All populations as simulation points and corresponding simulated values

Depending on computational constraints, one can optimize a single conditional simulation and keep the same simulation points for all other simulations, do it for bundles of several simulations or treat each of them separately. Note that a faster alternative to Algorithm 13 is to optimize directly on the Kriging mean, with the risk of not exploring areas of high variance.

We propose to compare the Vorob'ev deviation obtained by selecting the simulation points by optimization, taking them from a Sobol sequence or from a uniform random sampling. The setup is rather similar to the one in Chapter 3, considering centered GPs with Matérn

[1] In [Oak99], the kriging prediction based on simulation points replaces the full conditional simulation.

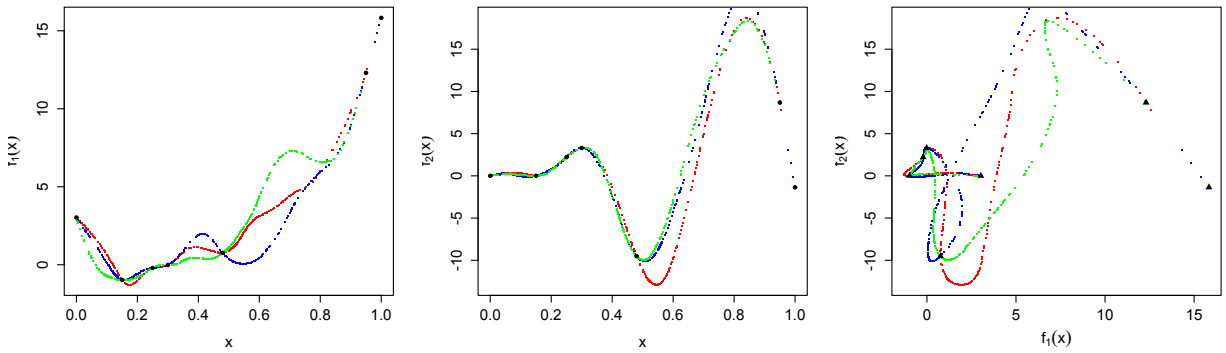
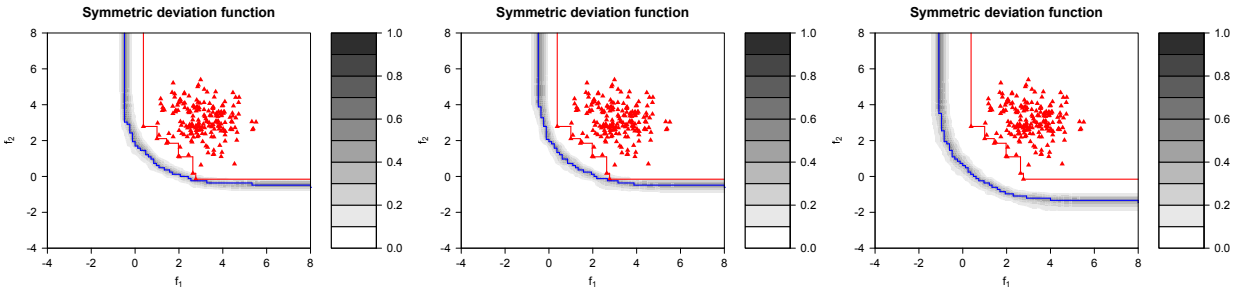


Figure B.4 – Example of the optimization of the location of simulation points with two objectives functions of one variable. Left and center: three conditional realizations with optimized locations (green, blue, red). Known observations points are marked with bigger black points. Right: corresponding image in the objective space.



(a) Uniform sampling, $VD = 3.05$

(b) Sobol sequence, $VD = 3.45$

(c) With optimization, $VD = 4.31$

Figure B.5 – Symmetric deviation function obtained with three different strategies for conditional simulation generation.

kernel for f_1 and f_2 but this time with an input dimension of twenty and 200 observations. The three strategies are compared considering a fixed number of simulation points, here 6000. The approach giving the results with highest Vorob'ev deviation is with optimization of the simulation, see the results Figure B.5. The corresponding Vorob'ev expectation widely dominates the two others, as was expected since the Pareto set is a (possibly disconnected) curve of dimension 1 hidden in this twenty dimensional space and the probability to be relatively close to it is very low. Also the Sobol sequence and the uniform sampling give similar results. In Chapter 8, the optimization of conditional simulation points is also applied to a problem in dimension 47. Finally, another perspective is to select simulation points maximizing the Vorob'ev deviation (the aim remaining to reduce it the most), e.g. as in [ABCG15].

Appendix C

Follow-up on Chapter 4

C.1 Comparison with the Gaussian processes method

We summarize here the main differences and similarities between approaches of Chapter 3 with Gaussian processes and Chapter 4 with copulas:

- both focus on the objective space, but while there is a strong link to the input space with GP models, it is much more tenuous with copulas (only by sampling);
- the dependence structure between objectives can be restricted to be Archimedean with the copula framework, while integrating one with GP models is also possible but at the cost of losing flexibility in the design space [ÁRL11] and is generally not done in practice;
- the GP approach can be applied sequentially, while for now the copula modeling works only for i.i.d. observations;
- the copula approach is much faster than the one using GPs, in particular, the computational complexity is not affected by an increase of the variable dimension;
- prior information on f may be integrated via the kernel function for GPs. With copulas it amounts to selecting the proper univariate marginals or dependence model, which are seldom known.

Next we describe the inner motivation which initially started the work on copula, i.e. to find an alternative to the use of conditional simulations.

C.2 Combination with the GP approach

While giving promising results, the quantification of uncertainty based on the Vorob'ev deviation described in Chapter 3 is computationally expensive when using conditional sim-

ulations, and consequently the associated SUR criterion, see Appendix B, is very expensive. Conditional simulations are used mostly to estimate the attainment function. Indeed, going back to the article of Chevalier *et al.*, [CGBM13], once the coverage (or attainment) function $p_{\mathcal{Y}}$ is estimated, the computation of the Vorob'ev expectation and deviation does not require conditional simulations anymore: $\mu(Q_{\beta^*}) = \int_{\Omega} p_{\mathcal{Y}}(u)\mu(du)$ and $\mathbb{E}\left(\mu(Q_{\beta^*}\Delta\mathcal{Y})\right) = \int_{Q_{\beta^*}} (1 - p_{\mathcal{Y}}(u))\mu(du) + \int_{Q^c_{\beta^*}} p_{\mathcal{Y}}(u)\mu(du)$.

Recall that in Chapter 3, Conditional Pareto Fronts $\mathcal{Y}_i$, $1 \leq i \leq N$ obtained from conditional simulations of the GP models $Y_1, \ldots, Y_m$ of $f_1, \ldots, f_m$ are realizations of a random closed set $\mathcal{Y}$. The coverage function of a random closed set, see Definition 3.3.1, is $\alpha : \mathbf{u} \in S, u \to \mathbb{P}\left[\mathbf{u} \in \mathcal{Y}\right] = \mathbb{P}\left[\exists \mathbf{x} \in \mathrm{E}\,,\, Y_1(\mathbf{x}) \leq u_1 \cap \cdots \cap Y_m(\mathbf{x}) \leq u_m\right]$. No closed form are available for these probabilities (in fact not even with one objective, see e.g [AW09]).

In Chapter 3, the coverage function is estimated using the empirical attainment function, see Definition 3.3.3, i.e. $\hat{\alpha}_N(\mathbf{u}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{\mathbf{u} \in \mathcal{Y}_i\}}$. Algorithm 2 provides $\mathcal{Y}_1, \ldots, \mathcal{Y}_N$, at p simulations points. Here, these p simulation points are a sequence of p i.i.d. random vectors $\mathbf{X}_1, \ldots, \mathbf{X}_p \sim \mathbf{X}$, with $\mathbf{X}$ a random vector with support E and of absolutely continuous distribution with respect to the Lebesgue measure. Consequently, $\hat{\alpha}_N(\mathbf{u})$ is an estimator of $\alpha_p(\mathbf{u}) = \mathbb{P}\left[\bigcup_{i=1}^{p} (Y_1(\mathbf{X}_i) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}_i) \leq u_m)\right]$ and $\alpha(\mathbf{u}) = \lim_{p \to +\infty} \alpha_p(\mathbf{u})$.

Let us now detail the link between the empirical attainment and the multivariate cumulative distribution function that has been the focus of Chapter 4. From the Bonferroni formulas, we get:

$$\mathbb{P}\left[\bigcup_{i=1}^{p} (Y_1(\mathbf{X}_i) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}_i) \leq u_m)\right] \leq \sum_{i=1}^{p} \mathbb{P}\left[Y_1(\mathbf{X}_i) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}_i) \leq u_m)\right].$$

Since $\mathbf{X}_1, \ldots, \mathbf{X}_p$ are identically distributed:

$$\sum_{i=1}^{p} \mathbb{P}\left[Y_1(\mathbf{X}_i) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}_i) \leq u_m)\right] = p\ \mathbb{P}\left[Y_1(\mathbf{X}) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}) \leq u_m\right].$$

We thus have an upper bound on α_p: $\alpha_p(\mathbf{u}) \leq p\ \mathbb{P}\left[Y_1(\mathbf{X}) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}) \leq u_m\right]$, where one would recognize the joint multivariate cumulative distribution function of $(Y_1(\mathbf{X}), \ldots, Y_m(\mathbf{X}))$, enabling the possibility to apply the proposed copula method this random vector. For estimation, sampling from the posterior distributions of the GPs at $\mathbf{X} \in \mathrm{E}$ is far less computationally expensive than generating conditional simulations. The general approach is described in Algorithm 14.

Algorithm 14 Estimation of an upper bound on the attainment function with Archimedean copulas

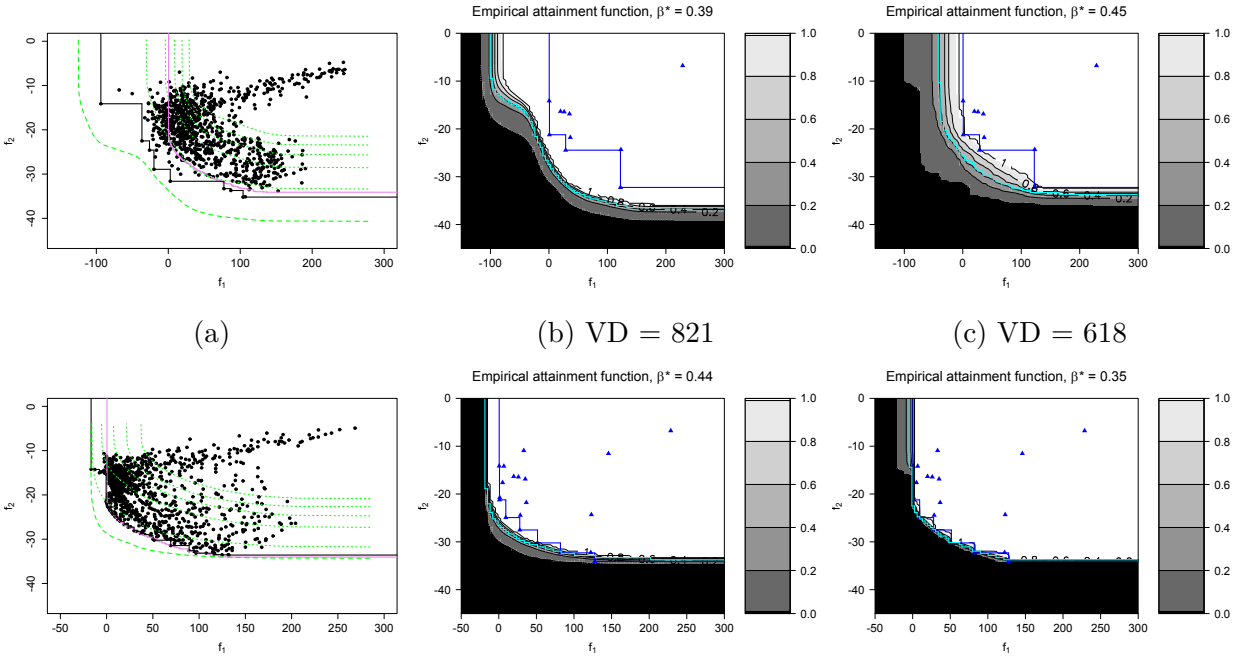
Input: GP models $Y_1, \ldots, Y_m$ corresponding to $f_1, \ldots, f_m$.
1: Sample p i.i.d. points in E (e.g. uniformly): $\mathbf{X}_1, \ldots, \mathbf{X}_p$.
2: Sample the corresponding values in S, i.e. when objectives are independently modeled: $(\mathcal{N}(\hat{y}_1(\mathbf{X}_i), \hat{s}_1(\mathbf{X}_i)), \ldots, \mathcal{N}(\hat{y}_m(\mathbf{X}_i), \hat{s}_m(\mathbf{X}_i)))_{1 \leq i \leq p}$.
3: Estimate the level lines or values of the cdf with Algorithm 6.

The GP posterior has already been used in MOO, for instance to obtain a dense approximation of the Pareto front [CPD14]. Note that using the empirical estimator of the multivariate cdf at $\mathbf{u}$, i.e. $\frac{1}{p} \sum \mathbf{1}_{Y_1(\mathbf{X}_i) \leq u_1 \cap \cdots \cap Y_m(\mathbf{X}_i) \leq u_m}$, to compute the upper bound on the attainment with samples from the posterior only results in a field of 0 and 1, i.e. if there is no or at least one point dominating the current location in the sample respectively. Indeed, the possible discrete values for this empirical estimation goes from 0 to 1 with step of $1/p$, hence the upper bound goes from 0 to p by steps of 1, thresholded to 1 since it is a probability[1]. Then the Vorob'ev expectation is undefined, all $\mathcal{Q}_\beta$-quantiles with $\beta > 0$ are identical. A similar problem may occur when using the empirical copula, since it is again a step function, but not with Archimedean copulas.

In the end, the upper bound on the attainment may be rather crude, but it is far cheaper to compute, especially when the input dimension increases. The results obtained on a synthetic 2-dimensional function, problem (P1) (see Chapter 3) are presented in Figure C.1. They correspond to one hundred conditional simulations on a thousand locations uniformly sampled in E or from a thousand samples of the posterior distribution at these uniformly sampled points in E, respectively.

Perspectives include applying the proposed estimation approach to the SUR criterion based on the Vorob'ev deviation. The attainment function might be replaced by its upper bound in Algorithm 11, with posterior distributions derived from the Kriging update formulas, see e.g. [CEG14] and references therein. An open issue remaining is about possible approximations or expressions of the marginal of outputs from a GP, instead of using more standard marginal estimators as proposed in Chapter 4. Studies of adequate models of copulas for the dependency between GPs may also be beneficial. The approach developed in [TWG07], applicable on a grid, might be employed for comparison.

[1] Taking $p - 1$ instead of p results in three levels, 0, $p/(p+1)$ and 1, which is not much more interesting.



(a) (b) VD = 821 (c) VD = 618

(d) (e) VD = 370 (f) VD = 120

Figure C.1 – Top: results after the design of experiments, $n = 10$. Bottom: results after ten rounds of optimization with EHI, $n = 20$. Left: samples from the posterior distribution of (Y_1, Y_2) (points) with the true Pareto front of f in violet and level lines of level $(\alpha^*, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5)$ in green dashed lines. Center and right: filled level lines of the upper bound on the attainment function α_p and of the empirical attainment function $\hat{\alpha}_N$ respectively, in gray-scale. The Vorob'ev expectation is in cyan and observations are denoted by blue triangles.

Appendix D

Complements on REMBO

Some additional results following Chapter 5 are presented here concerning the size of the low-dimensional search domain $\mathcal{Y}$ as well as the splitting strategy proposed in [WZH+13], to avoid the risk of missing the optimum. Then additional interpretation for the warping procedure are given before detailing analytical results on optimal matrices for the problem $(\mathcal{D})$ addressed in Chapter 6. Some results concern ongoing works and several proofs are reduced to sketchs. Notations are as in Chapters 5 and 6.

D.1 Additional experiments following Chapter 5

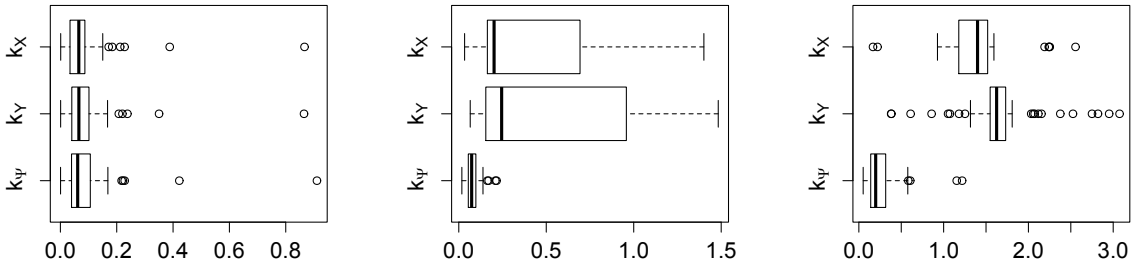
D.1.1 Influence of the bounds and of the high dimensionality

In Chapter 5, the proposed covariance kernel k_Ψ is shown to increase the robustness of REMBO. We illustrate in Figure D.1 that k_Ψ, as opposed to both $k_\mathcal{X}$ and $k_\mathcal{Y}$, allows to take larger bounds without hindering the performance. It also appears that if $\mathcal{Y}$ is sufficiently small such that $\mathbf{A}\mathcal{Y} \subset \mathcal{X}$, then all three considered kernels give similar results. It has motivated the work on bounds in Chapter 6.

One may also wonder if the performance shown is independent of D as in [WZH+13]. We thus replicated the experiments of Chapter 5 with $D = 1000$. The results in Figure D.2a) show no difference with those where $D = 25$.

D.1.2 Comparison to splitting

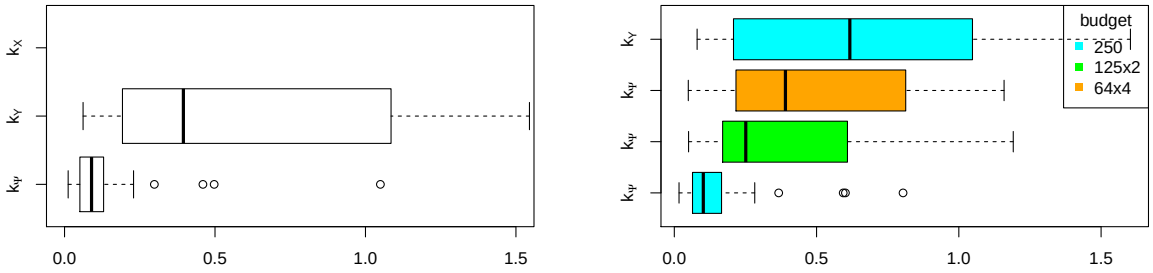
In [WZH+13], splitting the evaluation budget between several different random embeddings improves the performance of the method, on the Branin-Hoo test function with total budget of 500 evaluations. It also decreases the performance if there are not enough evaluations left for each sub-optimization.



(a) $\mathcal{Y} = 0.2 \times [-\sqrt{6}, \sqrt{6}]^6$ (b) $\mathcal{Y} = [-\sqrt{6}, \sqrt{6}]^6$ (c) $\mathcal{Y} = 5 \times [-\sqrt{6}, \sqrt{6}]^6$

Figure D.1 – Optimality gap for the Hartman6 function with $D = 25$ over 50 instances with a budget of 250 evaluations for various bounds.

For the example considered in Chapter 5, i.e. the Hartman function in dimension 6 with 250 evaluations as total budget, we also test the splitting strategy, using k_Ψ. In such case, keeping all the budget for one single embedding is much better, see Figure D.2b.



(a) (b)

Figure D.2 – Optimality gap for the Hartman6 test function over 50 runs. Left: $D = 1000$ with a budget of 250 evaluations for k_Ψ and $k_\mathcal{Y}$. Right: $D = 25$, the budget of evaluations is split according to the legend.

D.2 Insight on the warping Ψ

We give here further details on the warped kernel k_Ψ, see Chapter 5. It has two additional components over the high dimensional kernel $k_\mathcal{X}$ proposed in [WZH+13], namely orthogonal projection onto Ran($\mathbf{A}$) and a distortion.

The set obtained by orthogonal projection of $p_{\mathcal{X}}(\mathcal{X})$ onto $\mathrm{Ran}(\mathbf{A})$ is known to be a zonotope, a special class of convex symmetric polytopes, see Definition D.2.1 and e.g. in [McM71], [Zie95], [LSA+13].

Definition D.2.1 (Zonotope as Hypercube affine projection, adapted from [LSA+13])**.** *A D-zonotope in $\mathbb{R}^d$ is the translation by the center $\boldsymbol{p} \in \mathbb{R}^d$ of the image of the $[-1,1]^D$ hypercube $\mathcal{C}^D$ under a linear mapping. Given a matrix $\mathbf{H} \in \mathbb{R}^{d\times D}$ representing the linear mapping, the zonotope $\mathcal{Z}$ is defined by $\mathcal{Z} = \boldsymbol{p} + \mathbf{H}\mathcal{C}^D$.*

In fact, we show in Proposition D.2.1 that the orthogonal projection of the set $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ onto $\mathrm{Ran}(\mathbf{A})$ is equal to the orthogonal projection of the set $\mathcal{X}$ onto $\mathrm{Ran}(\mathbf{A})$.

Proposition D.2.1. *$p_{\mathbf{A}}(\mathcal{X})$ and $p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d))$ are equal zonotopes.*

Sketch of proof. First, $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d) \subset \mathcal{X}$, hence $p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)) \subseteq p_{\mathbf{A}}(\mathcal{X})$.

It follows from Definition D.2.1 that $p_A(\mathcal{X})$ is a zonotope of center O, obtained from the orthogonal projection of the D-hypercube $\mathcal{X}$. As such, $p_{\mathbf{A}}(\mathcal{X})$ is a convex polytope, which can be described directly from its vertices.

Let $\mathbf{x} \in \mathbb{R}^D$ be a vertex of $p_{\mathbf{A}}(\mathcal{X})$.
If $\mathbf{x} \in \mathcal{X}$, then $p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{x})) = p_{\mathbf{A}}(\mathbf{x}) = \mathbf{x}$, i.e. $\mathbf{x}$ has a pre-image in $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$.
Else, if $\mathbf{x} \notin \mathcal{X}$, consider the vertex $\mathbf{v}$ of $\mathcal{X}$ such that $p_{\mathbf{A}}(\mathbf{v}) = \mathbf{x}$. Suppose that $\mathbf{v} \notin p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$. Let us remark that if $\mathbf{v}$ is a vertex of $\mathcal{X}$ such that $\mathbf{v} \notin p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$, then $\mathrm{Ran}(\mathbf{A}) \cap \{h \in H_v\} = \emptyset$, where H_v is the open hyper-octant (with strict inequalities) that contains $\mathbf{v}$. Indeed, if $\mathbf{x} \in \mathrm{Ran}(\mathbf{A}) \cap \{h \in H_v\}$, $\exists k \in \mathbb{R}^*$ such that $p_{\mathcal{X}}(k\mathbf{x}) = \mathbf{v}$, which contradicts $\mathbf{v} \notin p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$. Denote by $\mathbf{u}$ the intersection of the line $(O\mathbf{x})$ with $\mathcal{X}$, since $\mathbf{x} \notin H_v$, $\mathbf{u} \notin H_v$ either, hence $\widehat{\mathbf{xuv}} > \pi/2$. Then $\|\mathbf{u} - \mathbf{v}\| \leq \|\mathbf{x} - \mathbf{v}\|$, which contradicts $\mathbf{x} = p_{\mathbf{A}}(\mathbf{v})$. Hence $\mathbf{v} \in p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ and $p_{\mathbf{A}}(\mathcal{X}) \subseteq p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d))$.

□

As a zonotope, several properties of $p_{\mathbf{A}}(p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d))$ are available, such as its number of vertices and facets, see e.g. [TOG04]. In addition, as stated in [Zie95], "the faces of Z can be uniquely associated with the faces of the cube it is projected from", here the faces of $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$. Exploiting the rich literature on zonotopes is thus one possible line for future research.

Finally, Lemma D.2.1 explains why the distortion part of the warping Ψ is necessary: vertices of $p_{\mathbf{A}}(\mathcal{X})$ farther away from O are actually the closest to $\mathrm{Ran}(\mathbf{A})$, which do not represent well distances on convex projected parts. This is corrected by the distortion: distances on $\mathrm{Ran}(\mathbf{A})$ with respect to the origin are then equal to distances between pivot point and

vertices (thus bigger than the norm of the orthogonal projection).

Lemma D.2.1. *Vertices of $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ farthest away from* $\mathrm{Ran}(\mathbf{A})$ *correspond to vertices of $p_{\mathbf{A}}(\mathcal{X})$ closest to O.*

Proof. Consider a vertex $\mathbf{x}$ of $\mathcal{X}$ and $\mathbf{x}_\top$ its orthogonal projection on $\mathrm{Ran}(\mathbf{A})$. From Pythagoras, $\|\mathbf{x}\|^2 = \|\mathbf{x}_\top\|^2 + \|\mathbf{x} - \mathbf{x}_\top\|^2$; since the norm of a vertex is fixed, it follows that $\|\mathbf{x} - \mathbf{x}_\top\|$ decreases as $\|\mathbf{x}_\top\|$ increases. □

D.3 Special case with $d = 2$ for A optimal in the sense of problem $(\mathcal{D})$

In Chapter 6, optimal solutions with $d = 2$ in the sense of problem $(\mathcal{D})$ turned out to be matrices with lines corresponding to vertices of a regular polygon. We detail briefly some additional properties that can be deduced from the symmetry of these polygons. Without loss of generality, we suppose that rows of $\mathbf{A}$ are of norm 1. We then describe further optimal matrices with $d = 2$ in Lemma D.3.1.

Lemma D.3.1. *Optimal matrices in the sense of problem $(\mathcal{D})$ with $d = 2$ have, up to rotation and multiplication by a positive scalar, rows of the form: $\mathbf{A}_i = (\rho\cos(\theta), \rho\sin(\theta))$ with $\rho = \pm 1$ and $\theta \in \{0 \leq k \leq D-1, k\pi/D\}$.*

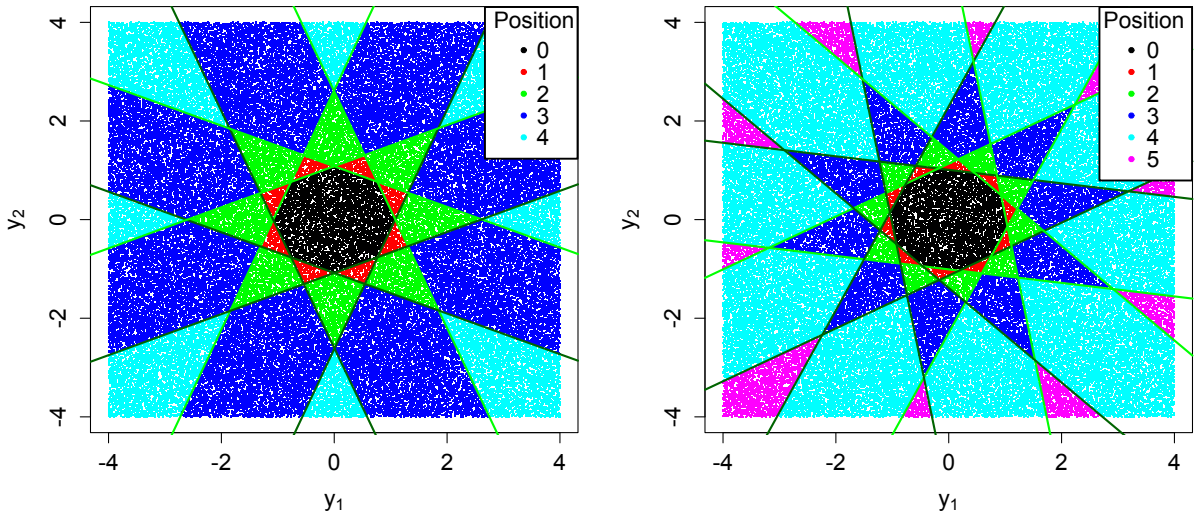
Sketch of proof. These are (half) the vertices of the regular polygon with $2D$ vertices. Applying a rotation of the low dimensional domain or a rescaling by a scalar does not affect $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$. □

We thus deduce that rows of an optimal $\mathbf{A}$ in the sense of problem $(\mathcal{D})$ are orthogonal:

$$\mathbf{A}_1 \cdot \mathbf{A}_2 = \sum_{k=0}^{D-1} \rho^2 \sin(k\pi/D)\cos(k\pi/D) = \sum_{k=0}^{D-1} \sin(2k\pi/D)/2 = 0.$$

Concerning pivot points, they are on $\mathcal{I}$, which is also in this case a regular polygon, see Figure D.3 (black polygon). It appears after warping that extreme points (i.e. points most distant to the center) in $\Psi(\mathbb{R}^d)$ are also extreme points of $\mathcal{U}$ (since columns are orthogonal and of same norm). Pivot points corresponding to vertices of parallelograms with biggest diameter, i.e. extreme points of $\mathcal{U}$, are either vertices of $\mathcal{I}$ if D is even or at the middle of the edge joining two adjacent vertices of $\mathcal{I}$ if D is odd. With rows of norm 1, $\mathcal{I}$ has apothem 1 (radius of the inscribed circle) and radius $1/\cos(\pi/(2D))$ (radius of the circle enclosing $\mathcal{I}$). We thus can express the coordinates of these pivot points in $\mathcal{X}$ (up to a rotation, for symmetry reasons):

- D even: with $\mathbf{A}$ as expressed above in Lemma D.3.1, $\mathbf{y} = \left(1, \sin\left(\frac{\pi}{2D}\right) / \cos\left(\frac{\pi}{2D}\right)\right)$ is a pivot point for an extreme vertex. Then $\mathbf{Ay}$ has coordinates of the form $\pm\cos(\frac{\pi}{2D} - \frac{k\pi}{D})\cos(\frac{\pi}{2D})$, $0 \leq k \leq D$;
- D odd: $\mathbf{y} = (0, 1)$ is a pivot point for an extreme vertex, then $\mathbf{Ay}$ has coordinates of the form $\pm\cos(\frac{k\pi}{D})$, $0 \leq k \leq D$.



(a) $D = 4$ (b) $D = 5$

Figure D.3 – Example with $d = 2$ for $D = 4$ and $D = 5$, in $\mathcal{Y}$. Colors correspond to orders of the face on the D-cube of the image by $p_{\mathcal{X}}(\mathbf{A}\cdot)$, i.e. the number of variables equal to 1 in absolute value; in particular black points are in $\mathcal{I}$.

For extreme vertices, the norm of pivot points as well as distance between pivot points and vertices have been computed with the online symbolic computational software WolframAlpha [Wol], using that coordinates of the pivot point and the corresponding vertex are of same sign. A summary of the tentative results is given is Table D.1. It follows from these expressions that for vertices of $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ and their pivot points:

$$\lim_{D\to\infty} \frac{\|\text{pivot point}\| + \|\text{vertex} - \text{pivot}\|}{\|\text{pivot point}\|} = \lim_{D\to\infty} r_{\Psi}^*/l^* = 1 + \sqrt{3 - 8/\pi}.$$

In contrast, for a random $\mathbf{A}$, this ratio is probably unbounded.

Table D.1 – Quantities of interest when rows of $\mathbf{A}$ are vertices of a convex regular polygon in $\mathbb{R}^2$. The distinction between D even or odd is simply that the position of pivot points for vertices of $p_{\mathcal{X}}(\mathbf{A}\mathbb{R}^d)$ is either on a vertex of $\mathcal{I}$ or at the center of the edge between two adjacent vertices.

	D even	D odd
coordinates of pivot points, $0 \leq k \leq D-1$	$\cos(\frac{\pi}{2D} - \frac{k\pi}{D})\cos(\frac{\pi}{2D})$	$\cos(\frac{k\pi}{D})$
$\|$vertex point $\|$	$\sqrt{D}$	$\sqrt{D}$
$\|$ pivot point $\|$	$\sqrt{D/2} \times \frac{1}{\cos(\pi/(2D))}$	$\sqrt{D/2}$
$\|$vertex - pivot $\|$	$\sqrt{D\left(1 + \frac{1}{2\cos(\pi/2D)^2}\right) - \frac{8}{\sin(\pi/D)}}$	$\sqrt{\frac{3D}{2} - \frac{2}{\sin(\pi/2D)}}$

Appendix E

Résumés des chapitres en français

E.1 Introduction

Dans le domaine de la conception en ingénierie, les essais physiques et les codes de calcul numériques peuvent avoir des coûts ou des temps d'évaluation prohibitifs. De nos jours, ils restent utilisés intensivement pour concevoir et optimiser des systèmes complexes tels que des automobiles. Une conséquence directe est un budget d'évaluation dédié à l'optimisation extrêmement limité, ce qui rend une procédure par tâtonnement inadaptée. En sus, le gradient et les propriétés mathématiques de la fonction considérée sont très souvent indisponibles, ce qui correspond à un cadre *boîte noire*. Par conséquent, chercher l'optimum en appliquant des méthodes de descente de gradient n'est pas réalisable : le gradient doit être approché, ce qui est très coûteux, et au risque de n'identifier que des solutions locales. Utiliser différents points de départ peut permettre de trouver l'optimum global, c'est-à-dire le meilleur possible, mais nécessite encore plus d'évaluations.

Une solution préférable est d'établir un modèle de substitution, ou métamodèle (modèle de modèle), de la fonction coûteuse avec aussi peu d'observations que possible, pour ensuite pouvoir prédire la réponse en tout point du domaine. Ces techniques sont très communes en *computer experiments* et en *machine learning*, c.f. e.g. [SWN03], [FLS05], [RW06], [Kle07], [FSK08], [HHLB11], [SLA12]. Le principe est illustré en Figure 1.1. Clairement, le modèle de substitution initial n'est qu'une approximation grossière et n'est pas adapté pour optimiser directement. Néanmoins, et en particulier avec les processus gaussiens, la prédiction des réponses est donnée avec une estimation de l'incertitude associée. Cela permet de définir des critères statistiques qui fournissent un équilibre entre exploration et exploitation du domaine de recherche. La figure de droite montre le métamodèle obtenu après ajout séquentiel de nouvelles observations, en prenant ce point de vue : le modèle est plus précis dans les régions d'intérêt, en particulier avec de nouvelles observations proches des trois optima globaux.

Dans sa forme la plus simple, un problème d'optimisation consiste à trouver le minimum (ou le maximum) d'une fonction pour un périmètre donné. Les problèmes réels sont toutefois plus complexes, comme par exemple lorsque des contraintes de fabrication sont prises en compte. De plus, la conception est typiquement sujette à différents objectifs, possiblement antagonistes. Des techniques spécifiques existent pour traiter ces deux cas, souvent en étendant les méthodes mono-objectives non contraintes. Nous considérons ici en général deux ou trois objectifs simultanément, jusqu'à cinq lors d'une application. Avec plus de trois objectifs, il n'est alors plus possible de visualiser la surface des compromis optimaux et des méthodes dédiées se retrouvent dans la littérature sur l'optimisation *many-objective*.

Une autre difficulté, écartée ici, est le cas d'observations bruitées. Lorsque c'est pertinent, les extensions possibles pour traiter cette situation seront mentionnées. Remarquons que la simulation numérique, si elle n'est pas soumise au bruit d'observation comme les expériences réelles, n'est pas exempte d'autres sources de bruit. En effet, si répéter exactement le même calcul renvoie le même résultat en général, avec des codes Monte Carlo ou par éléments finis, la sortie dépend du nombre d'itérations et de la taille du maillage. Par ailleurs, certains phénomènes physiques tels que le crash sont intrinsèquement instables, encore plus à grande vitesse, où une pièce peut se rompre différemment à cause d'un bruit numérique (par exemple à cause du nombre de cœurs utilisés ou de l'architecture en calcul haute performance).

Motivées également par des cas tests Renault, les contributions de cette thèse traitent à la fois d'optimisation multiobjectif et d'espace de recherche en grande dimension. Les résultats correspondants sont basés sur les processus aléatoires, l'optimisation bayésienne, les ensembles aléatoires et les copules. La structure du document reflète ces contributions avec quatre parties :

- La Partie I introduit le contexte général et le périmètre de ce manuscrit, en commençant par ce Chapitre 1. Le Chapitre 2 détaille brièvement l'optimisation avec métamodèle de fonctions boîtes noires coûteuses par processus gaussiens. Il se concentre ensuite sur l'amélioration espérée (*Expected Improvement*) et ses généralisations multiobjectifs. Dans un souci de concision et de clarté, les rappels sont limités au strict minimum et certains sujets ont parfois été omis lorsqu'aucune contribution n'a été apportée. Les notions qui ne sont pas nécessaires pour l'ensemble de la thèse sont détaillées dans les parties correspondantes.

- La Partie II contient deux articles en quantification d'incertitude sur les fronts de Pareto, à partir de deux points de vue : en utilisant des processus gaussiens et des simulations conditionnelles dans le Chapitre 3 et avec des copules dans le Chapitre

4. Alors que la plupart des méthodes fournissent une approximation discrète du front de Pareto, le but est ici de construire une représentation continue de l'ensemble des solutions optimales. Ce problème est traité soit en considérant un ensemble aléatoire fermé, soit en estimant les lignes de niveau extrêmes d'une fonction de répartition multivariée.

- La Partie III reprend les contributions proposées pour s'affranchir du challenge posé par des espaces de recherche en grande dimension avec des budgets limités. En s'appuyant sur un constat empirique, l'hypothèse d'un faible nombre de variables influentes (et inconnues) est faite et le problème est traité à partir de plongements aléatoires d'après les travaux de [WZH+13]. Le Chapitre 5 débute par une description de la méthode, avant de proposer un noyau de covariance qui s'affranchit de certains problèmes présents initialement. Le Chapitre 6 traite de la sélection des bornes du domaine de petite dimension dans la méthode REMBO (Random Embedding Bayesian Optimization). En particulier, des modifications du plongement aléatoire sont proposées, aux côtés de stratégies pour optimiser le critère d'ajout et une extension au cas multiobjectif. Combinées, ces modifications apportent un gain significatif aux performances.

- La Partie IV traite du côté implémentation logicielle et applicatif des deux parties précédentes. Dans le Chapitre 7 se trouve une description du package *GPareto* qui a été déposé sur le CRAN. Le Chapitre 8 est lui dédié à un cas test industriel en crash, qui a été utilisé durant cette thèse pour éprouver les différentes contributions.

Trois articles sont intégrés en Chapitres 3, 4 et 5 respectivement :

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

- M. Binois, D. Rullière, O. Roustant. On the estimation of Pareto fronts from the point of view of copula theory, *Information Sciences*, vol. 324, pp. 270-285 (2015).

- M. Binois, D. Ginsbourger, O. Roustant. A warped kernel improving robustness in Bayesian optimization via random embeddings, *Proceedings of the International Conference on Learning and Intelligent Optimization, LCNS*, vol. 8994, pp. 281-286 (2015).

La documentation du package *GPareto* est, quant à elle, disponible sur le CRAN :

- M. Binois, V. Picheny. GPareto : Gaussian Processes for Pareto Front Estimation and Optimization, R package version 1.0.1 (2015).

En annexes sont décrites des contributions additionnelles ou des travaux en cours prometteurs, incluant notamment une approximation grossière mais très rapide de l'Expected Improvement multi-point, un critère *Stepwise Uncertainty Reduction* et une procédure interactive d'optimisation complétant le Chapitre 3, ou encore de compléments sur REMBO.

Les tests numériques ont été réalisés sur un PC avec un quadri-cœur cadencé à 2.80GHz et 32Go de RAM, un portable avec un bi-cœur cadencé à 2.9GHz et 16Go de RAM, ou équivalents.

E.2 II - État de l'art en optimisation bayésienne

Tout au long de cette thèse, nous considérons une fonction coûteuse à évaluer, $f : \mathrm{E} \subset \mathbb{R}^d \to \mathrm{S}$ avec $\mathrm{S} = \mathbb{R}$ dans le cas d'un unique objectif et $\mathrm{S} = \mathbb{R}^d$ dans le cas multiobjectif. Les phénomènes considérés étant complexes, en général très peu voire rien n'est connu sur leurs propriétés mathématiques, et on les traite donc comme des boîtes noires. Nous nous intéressons à la minimization de f, possiblement sous contraintes. E est l'espace des variables (ou encore espace de décision, des paramètres), et S l'espace des objectifs. Nous décrivons tout d'abord ce problème du point de vue des codes de calcul déterministes, puis, brièvement, les processus gaussiens, avant de détailler les critères mono-objectifs qu'ils permettent de construire, en particulier l'*Expected Improvement*. Enfin, les concepts d'optimisation multiobjectif sont exposés, dont les généralisations correspondantes de l'*Expected Improvement*.

E.3 III - Quantification d'incertitude sur fronts de Pareto par processus gaussiens

Les algorithmes d'optimisation multiobjectif ont pour but de trouver les solutions Pareto optimales. Les retrouver à partir d'un nombre limité d'observations est un problème difficile. Une approche communément utilisée dans le cas de fonctions coûteuses à évaluer est de faire appel à un métamodèle. Le Krigeage a démontré son efficacité dans de nombreux cas comme base pour l'optimisation multiobjectif séquentielle, notamment par des critères d'ajout qui proposent un équilibre entre exploitation et exploration tels que l'*Expected Hypervolume Improvement*. Ici nous considérons des métamodèles de Krigeage non seulement pour sélectionner de nouveaux points, mais aussi comme outil pour estimer le front de Pareto dans son ensemble et quantifier l'incertitude restante à n'importe quelle étape du processus d'optimisation. Notre approche repose sur l'interprétation en tant que processus gaussiens du Krigeage, par l'utilisation de simulations conditionnelles. A partir de concepts tirés de la théorie des ensembles aléatoires fermés, nous proposons d'adapter l'espérance ainsi que la

déviation de Vorob'ev pour capturer la variabilité de l'ensemble des points non-dominés. Des expériences numériques illustrent l'intérêt de la procédure proposée, et il est montré sur des exemples comment les simulations conditionnelles de processus gaussiens et l'estimation de la déviation de Vorob'ev peuvent être utilisées pour suivre la capacité des algorithmes MO basés sur le Krigeage à apprendre précisément le front de Pareto.

E.4 IV - Quantification d'incertitude sur fronts de Pareto à partir de copules

Il est courant en optimisation de débuter par un tirage aléatoire dans l'espace des variables pour initialiser une population ou créer un métamodèle. En particulier, dans le cas multiobjectif, cela conduit à un ensemble de points non-dominés qui ne renseignent que peu sur le vrai front de Pareto. Nous proposons d'étudier ce problème du point de vue de l'analyse multivariée, en introduisant un cadre probabiliste et en particulier en utilisant le formalisme des copules. Ainsi, des expressions pour les lignes de niveau sont accessibles dans l'espace des objectifs et permettent par conséquent d'obtenir une estimation de la position du front de Pareto, lorsque le niveau tend vers zéro. Des expressions analytiques explicites sont disponibles quand des copules archimédiennes sont utilisées. La procédure d'estimation correspondante est détaillée puis appliquée sur plusieurs exemples.

E.5 V - Optimisation bayésienne en grande dimension par plongements aléatoires

Ce chapitre traite de l'impact d'un grand nombre de variables sur l'optimisation à partir de processus gaussiens. En particulier, la méthode REMBO est décrite avant de détailler une modification qui intègre un warping de l'espace de grande dimension dans le noyau de covariance. Le warping proposé, qui s'appuie sur des considérations géométriques simples, permet d'atténuer les inconvénients liés à la grande dimension tout en évitant à l'algorithme d'évaluer des points qui fournissent des informations redondantes. Cela permet également de relâcher les contraintes sur la sélection des bornes du domaine de petite dimension, et donc d'améliorer la robustesse de la méthode, tel qu'illustré sur un exemple en dimension 25 et de dimensionnalité intrinsèque 6.

E.6 VI - Analyse de la méthode REMBO pour une robustesse améliorée

La méthode REMBO apporte une solution possible pour optimiser en grande dimension, sous l'hypothèse qu'un faible nombre de variables soient effectivement influentes. Néanmoins, certains éléments pratiques restent problématiques, tels que la sélection des bornes du domaine de petite dimension. En particulier, lorsque l'on étend le paradigme à l'optimisation contrainte ou multiobjectif, supposer que l'optimum sera proche au centre du domaine ne tient pas. Il peut alors être nécessaire d'explorer également à la périphérie, c'est-à-dire au niveau des faces du domaine de grande dimension, sans pour autant s'y focaliser. En effet, de par la projection convexe, le domaine potentiellement explorable pour ajouter un nouveau point peut être très large et sans précaution particulière la convergence sera freinée. Pour contrer cet effet, nous proposons une étude des propriétés, notamment géométriques, du plongement aléatoire. Nous discutons ensuite d'options pour sélectionner les bornes du domaine de petite dimension ainsi que de modifications de la matrice contrôlant le plongement. Finalement, nous détaillons l'extension à l'optimisation multiobjectif et présentons les résultats sur plusieurs exemples.

E.7 VII - Contributions logicielles à l'optimisation multiobjectif

Ce chapitre est un tutorial du package R *GPareto*, qui a été livré sur le CRAN comme contribution de ces travaux de thèse. Il permet de résoudre des problèmes multiobjectifs et de quantification d'incertitude, telle que décrite dans le Chapitre 3. Après une brève description des packages liés en R, la structure du package est explicitée. Ensuite l'implémentation des différents critères d'optimisation disponibles (présentés en Chapitre 2) est détaillée, ainsi que certaines fonctionnalités. Des exemples illustratifs sont également fournis, avant d'étendre sur la possibilité d'implémentation de la méthode REMBO comme surcouche pour *GPareto*.

E.8 VIII - Cas test industriel

Ce chapitre détaille les expérimentations effectuées sur un cas test industriel en crash. Il s'agit d'optimiser l'absorbeur de choc situé derrière le bouclier arrière en considérant cinq objectifs : la masse ainsi que quatre enfoncements pour des scenarii de crash donnés. Ce problème peut être traité de manière multiobjectif ou avec des contraintes puisque le but final, une fois des seuils fixés sur les enfoncements, est de diminuer la masse du dispositif. Ce dernier comprend 47 paramètres, ce qui en fait un cas difficile pour l'optimisation bayésienne. Il a par conséquent

servi à éprouver les méthodes de quantification d'incertitude ou d'optimisation décrites dans les chapitres précédents, et de tester le package *GPareto*. Comparés à une ancienne étude utilisant NSGA-II sur modèles d'experts (régression + polyMARS), les résultats obtenus sont meilleurs en matière de performance et en nombre d'appels aux codes de calcul.

E.9 Conclusion

Dans cette thèse, nous étudions principalement le problème de l'optimisation multiobjectif de fonctions boîtes noires coûteuses. Ces problèmes sont très courants dans des contextes industriels et, comme exemple, nous détaillons des expérimentations en crash de voitures.

Les travaux existants en optimisation multiobjectif par modèle de Krigeage proposent déjà des solutions efficaces pour obtenir des solutions Pareto optimales. Cependant, les approximations du front de Pareto obtenues sont seulement discrètes. Deux principales contributions de cette thèse ont pour but de donner une représentation continue du front de Pareto. La solution proposée dans le Chapitre 3, adaptée du travail de [Che13], est de faire appel à des simulations conditionnelles de processus gaussiens en conjonction avec des concepts tirés de la théorie des ensembles aléatoires fermés pour capturer la variabilité autour du front de Pareto donné par les modèles de substitution. Recourir aux simulations conditionnelles pour calculer la fonction d'*attainment* est une gageure au niveau du temps de calcul, et plus encore en considérant le critère d'optimisation associé. Dans une deuxième contribution, cette étape est remplacée par une estimation de la fonction de répartition multivariée des modèles de substitution, en utilisant le formalisme des copules, voir Chapitre 4. Les deux approches apportent aux praticiens une approximation du front de Pareto sur laquelle ils peuvent s'appuyer pour décider de stopper, d'intensifier ou d'orienter le processus d'optimisation.

Un grand nombre de variables est identifié comme l'un des principaux challenges pour l'optimisation par modèle de substitution. En effet, il impacte la vitesse d'apprentissage et l'utilisation pratique ; ainsi un changement de point vue est sans doute requis pour amener ces méthodes plus loin sur le plan de la dimensionnalité. Une avancée potentielle a été effectuée avec l'algorithme REMBO [WZH+13], qui utilise des plongements aléatoires d'un espace de petite dimension vers l'espace initial de grande dimension. Nous avons contribué à l'analyse de cette méthode et avons proposé des extensions inédites sur certains points bloquants. Premièrement, dans le Chapitre 5, avec une fonction de covariance qui évite les écueils associés aux noyaux proposés initialement pour REMBO. Ensuite, au Chapitre 6, nous avons travaillé sur la question de la sélection d'un domaine de petite dimension adéquat afin d'éviter de manquer l'optimum, tout en essayant de ne pas trop impacter le processus d'optimisation avec un domaine de recherche trop étendu. Nous montrons une nette amélioration

des performances, qui, notamment, favorise l'application au cas multiobjectif. Cela ouvre le chemin vers de nouveaux développements, possiblement en analyse de sensibilité ou en hybridant avec des méthodes d'apprentissage actif de sous-espace linéaires (cf. [GOH13], [DKC13]).

Pour propager et étendre l'utilisation de ce type de méthodes en ingénierie, le package R *GPareto* a été déposé sur le CRAN pour compléter *DiceKriging*, et les packages liés, par des méthodes d'optimisation multiobjectif. Le Chapitre 7 fournit un tutorial pour GPareto. De plus, pour s'assurer de l'applicabilité des contributions dans un contexte industriel, les méthodes ont été testées sur un cas d'étude Renault, présenté en Chapitre 8, avec des performances supérieures à des résultats précédents, à la fois au niveau de la qualité des solutions et en nombre d'évaluations requis pour les obtenir.

En supplément, des essais d'adaptation de noyaux ont été réalisés, et ils forment l'une des plus importantes directions pour des recherches ultérieures. En effet, nos contributions apportent des éléments à l'optimisation bayésienne en général et nous croyons au potentiel offert par des combinaisons avec certains développements récents, en particulier concernant les modèles de substitution ou les fonctions d'acquisition. Par exemple, une perspective attrayante pour REMBO est d'améliorer le modèle de substitution en tenant compte des instationnarités qui sont présentes, tant par le procédé de plongement qu'au niveau de la fonction boîte noire sous-jacente elle-même. Quelques travaux récents, e.g. [SSZA14], [AWdF14], [MC15], proposent différentes options pour les apprendre, donnant plus de flexibilité et montrant des performances intéressantes. Des extensions des processus gaussiens sont également une tendance actuelle, avec des alternatives telles que les *Deep GPs* [DL13] ou les processus de Student-t [SWG14], qui pourraient remplacer les processus plus standards utilisés ici.

Les obstacles restants sont un possible manque de solutions simples à prendre en main tout en tenant compte de la complexité des cas tests. Un exemple apparaît avec les variables mixtes, continues et discrètes [ZQZ11], ou encore avec des paramètres emboîtés : avec des alternatives de pièces dans un dispositif, ayant toutes leurs propres paramètres. D'un autre côté, des opportunités apparaissent avec l'intégration d'un comportement physique avec les modèles de *latent force* [TL10], par l'exploitation de modèle avec différents niveaux de fidélité, des observations du gradient qui peuvent maintenant être obtenues avec les solveurs adjoints ou par des approximations, voir [Fro14], [GJGM15]. Enfin, les travaux en cours sur d'autres cas tests en crash soulèvent la question de la prise en compte d'observations bruitées dans les critères d'ajout multiobjectifs, comme dans [KWE+15], et dans REMBO.

Appendix F

Main notations

Vectors are denoted with bold lowercase letters, such as $\mathbf{x}$, scalars with lowercase letters (x), while uppercase bold letters, i.e. $\mathbf{A}$, denote matrices. Below are summarized the main notations used throughout this manuscript. In an attempt to respect common notations from communities on the various topics treated, some conflicts remain between chapters and additional notations are detailed upon appearance.

Symbol	Description
	Mathematical notations
$\mathbb{N}^*$	$\mathbb{N} \setminus \{0\}$
$\mathbb{R}^*$	$\mathbb{R} \setminus \{0\}$
Ran	range/image of an application
μ	Lebesgue measure
ϕ	probability density function of the standard Gaussian law $\mathcal{N}(0,1)$
Φ	cumulative distribution function of the standard Gaussian law $\mathcal{N}(0,1)$
$\|\cdot\|$	Euclidean norm
	defined variable names (general)
f	objective function, when $m > 1$, $f = f_1, \ldots, f_m$
k	covariance kernel of a GP
m	number of objectives
m_n	Kriging mean
n	number of evaluations
s_n	Kriging standard deviation
$\mathbf{x}^*$	global minimizer of f

E	parameter space
I	improvement function
S	objective space
Y	Gaussian process model of f
$Y_1, \ldots, Y_m$	Gaussian process models of $f_1, \ldots, f_m$
$\mathcal{A}_n$	$\{f(\mathbf{x}_1) = y_1, \ldots, f(\mathbf{x}_n) = y_n\}$

defined variable names in Part II

$p_{\mathcal{Y}}$	coverage function
C	copula function
$F_{\mathbf{Y}}$	multivariate cumulative distribution function of $\mathbf{Y}$
I_H	hypervolume indicator
L_α^F	upper level set of level α of $F_{\mathbf{Y}}$
∂L_α^F	α-level line of $F_{\mathbf{Y}}$
$\mathcal{P}$	true Pareto front
$\mathcal{P}_n$	Pareto front of the n observations
$\mathcal{Q}_\beta$	β-quantile
$\mathcal{X}$	random non-dominated point set
$\mathcal{Y}$	random attained set
$\alpha_{\mathcal{X}}$	attainment function
$\hat{\alpha}_N$	empirical attainment function
ϕ	generator of an Archimedean copula
Δ	symmetric difference between sets

defined variable names in Part III

d	dimension of the low-dimensional domain
g	function defined over $\mathcal{Y}$, $g(\mathbf{y}) = f(p_{\mathcal{X}}(\mathbf{A}\mathbf{y}))$
$p_{\mathcal{X}}$	convex projection onto $\mathcal{X}$
$p_{\mathbf{A}}$	orthogonal projection onto $\mathrm{Ran}(\mathbf{A})$
$\mathbf{A}$	$D \times d$ matrix
D	dimension of the variable domain
I	set of variable indices, subset of $\{1, \ldots, D\}$
$\mathcal{A}$	set of $D \times d$ matrices for which all $d \times d$ submatrices are invertible
$\mathcal{I}$	intersection of all parallelotopes given by the matrix $\mathbf{A}$
$\mathcal{S}_i$	strip defined by the i^{th} row of $\mathbf{A}$
$\mathcal{U}$	union of all parallelotopes given by the matrix $\mathbf{A}$
$\mathcal{X}$	$[-1, 1]^D$

$\mathcal{Y}$	subset of $\mathbb{R}^d$
$\mathfrak{P}_I$	parallelotope given by strips of indices in I
$\|\cdot\|_d$	Euclidean norm in $\mathbb{R}^d$
$\|\cdot\|_D$	Euclidean norm in $\mathbb{R}^D$
Ψ	warping
$\mathcal{T}$	effective subspace

Abbreviations

cdf	cumulative distribution function
ANOVA	ANalysis Of VAriance
CPF	Conditional Pareto Front
DOE	Design Of Experiments
EHI	Expected Hypervolume Improvement
EI	Expected Improvement
FANOVA	Functional ANalysis Of VAriance
GP	Gaussian Process
GP-LVM	Gaussian Process Latent Variable Model
LHS	Latin Hypercube Sampling
MO	Multi-Objective
MOO	Multi-Objective Optimization
PI	Probability of Improvement
REMBO	Random EMbedding Bayesian Optimization
RNP set	Random Non-dominated Point set
SUR	Stepwise Uncertainty Reduction
VD	Vorob'ev Deviation
VE	Vorob'ev Expectation

Bibliography

[ABBZ12] Auger A., Bader J., Brockhoff D., and Zitzler E. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75–103, 2012.

[ABCG15] Azzimonti D., Bect J., Chevalier C., and Ginsbourger D. Quantifying uncertainties on excursion sets under a Gaussian random field prior. *arXiv preprint arXiv:1501.03659*, 2015.

[Abr97] Abrahamsen P. *A review of Gaussian random fields and correlation functions*. Norsk Regnesentral/Norwegian Computing Center, 1997.

[AG13] Agrawal S. and Goyal N. Thompson sampling for contextual bandits with linear payoffs. *JMLR W&CP*, 28(3):127–135, 2013.

[ÁRL11] Álvarez M. A., Rosasco L., and Lawrence N. D. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2011.

[Aro50] Aronszajn N. Theory of reproducing kernels. *Transactions of the American mathematical society*, pages 337–404, 1950.

[AW09] Azaïs J.-M. and Wschebor M. *Level sets and extrema of random processes and fields*. Wiley, 2009.

[AWdF14] Assael J.-A. M., Wang Z., and de Freitas N. Heteroscedastic treed Bayesian optimisation. *arXiv preprint arXiv:1410.7172*, 2014.

[Bac13a] Bachoc F. Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69, 2013.

[Bac13b] Bachoc F. *Estimation paramétrique de la fonction de covariance dans le modèle de Krigeage par processus Gaussiens. Application à la quantification des incertitudes en simulation numérique*. PhD thesis, Université Paris-Diderot-Paris VII, 2013.

[Bar90] Barnett S. Matrices. Methods and applications. *Oxford Applied Mathematics and Computing Science Series, Oxford: Clarendon Press, 1990*, 1, 1990.

[Bau09] Bautista D. C. *A sequential design for approximating the Pareto front using the expected Pareto improvement function*. PhD thesis, The Ohio State University, 2009.

[BB12] Bergstra J. and Bengio Y. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305, 2012.

[BBZ12] Bartz-Beielstein T. and Zaefferer M. A gentle introduction to sequential parameter optimization. Technical Report 2, Bibliothek der Fachhochschule Koeln, 2012.

[BDD14] Bhardwaj P., Dasgupta B., and Deb K. Modelling the Pareto-optimal set using B-spline basis functions for continuous multi-objective optimization problems. *Engineering Optimization*, 46(7):912–938, 2014.

[BDV12] Bücher A., Dette H., and Volgushev S. A test for Archimedeanity in bivariate copula models. *Journal of Multivariate Analysis*, 110:121–132, 2012.

[Ben12] Bendtsen C. *pso: Particle Swarm Optimization*, 2012. R package version 1.0.3.

[Ben13] Benassi R. *New Bayesian optimization algorithm using a sequential Monte-Carlo approach*. Theses, Supélec, June 2013.

[BGL+12] Bect J., Ginsbourger D., Li L., Picheny V., and Vazquez E. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

[BGR15a] Binois M., Ginsbourger D., and Roustant O. Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations. *European Journal of Operational Research*, 243(2):386 – 394, 2015.

[BGR15b] Binois M., Ginsbourger D., and Roustant O. A warped kernel improving robustness in Bayesian optimization via random embeddings. In *Learning and Intelligent Optimization*, pages 281–286. Springer, 2015.

[BNE07] Beume N., Naujoks B., and Emmerich M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653 – 1669, 2007.

[BP15] Binois M. and Picheny V. *GPareto: Gaussian Processes for Pareto Front Estimation and Optimization*, 2015. R package version 1.0.1.

[BR12] Bienvenüe A. and Rullière D. Iterative adjustment of survival functions by composed probability distortions. *The Geneva Risk and Insurance Review*, 37(2):156–179, 2012.

[Bre92] Breiman L. *Probability*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[BRR15] Binois M., Rullière D., and Roustant O. On the estimation of Pareto fronts from the point of view of copula theory. *Information Sciences*, 324:270 – 285, 2015.

[BTA04] Berlinet A. and Thomas-Agnan C. *Reproducing kernel Hilbert spaces in probability and statistics*, volume 3. Springer, 2004.

[Bul11] Bull A. D. Convergence rates of efficient global optimization algorithms. *The Journal of Machine Learning Research*, 12:2879–2904, 2011.

[BW51] Box G. E. and Wilson K. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45, 1951.

[BWB+14] Bischl B., Wessing S., Bauer N., Friedrichs K., and Weihs C. MOI-MBO: multiobjective infill for parallel model-based optimization. In *Learning and Intelligent Optimization*, pages 173–186. Springer, 2014.

[CCK12] Chen B., Castro R., and Krause A. Joint optimization and variable selection of high-dimensional Gaussian processes. In *Proc. International Conference on Machine Learning (ICML)*, 2012.

[CDD13] Couckuyt I., Deschrijver D., and Dhaene T. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, pages 1–20, 2013.

[CEG14] Chevalier C., Emery X., and Ginsbourger D. Fast update of conditional simulation ensembles. *Mathematical Geosciences*, pages 1–19, 2014.

[CG13] Chevalier C. and Ginsbourger D. Fast computation of the multi-points expected improvement with applications in batch selection. In *Learning and Intelligent Optimization*, pages 59–69. Springer, 2013.

[CGB+14] Chevalier C., Ginsbourger D., Bect J., Vazquez E., Picheny V., and Richet Y. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4), 2014.

[CGBM13] Chevalier C., Ginsbourger D., Bect J., and Molchanov I. Estimating and quantifying uncertainties on level sets using the Vorob'ev expectation and deviation with Gaussian process models. In Ucinski D., Atkinson A. C., and Patan M., editors, *mODa 10 - Advances in Model-Oriented Design and Analysis*, Contributions to Statistics, pages 35–43. Springer International Publishing, 2013.

[CH14] Chaudhuri A. and Haftka R. T. Efficient global optimization with adaptive target setting. *AIAA Journal*, 52(7):1573–1578, 2014.

[Che13] Chevalier C. *Fast uncertainty reduction strategies relying on Gaussian process models.* PhD thesis, University of Bern, 2013.

[CHS81] Cambanis S., Huang S., and Simons G. On the theory of elliptically contoured distributions. *Journal of Multivariate Analysis*, 11(3):368 – 385, 1981.

[CL67] Cramer H. and Leadbetter M. R. *Stationary and related stochastic processes sample function properties and their applications.* Wiley, New York, 1967.

[CL11] Chapelle O. and Li L. An empirical evaluation of Thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[CL12] Cohen S. and Lifshits M. Stationary Gaussian random fields on hyperbolic spaces and on Euclidean spheres. *ESAIM: Probability and Statistics*, 16:165–221, 2012.

[Cla61] Clark C. E. The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162, 1961.

[CLG15] Chastaing G. and Le Gratiet L. ANOVA decomposition of conditional Gaussian processes for sensitivity analysis with dependent inputs. *Journal of Statistical Computation and Simulation*, 85(11):2164–2186, 2015.

[CLVV07] Coello C. C., Lamont G. B., and Van Veldhuizen D. A. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[CM12] Carpentier A. and Munos R. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *International conference on Artificial Intelligence and Statistics*, 2012.

[CPD14] Calandra R., Peters J., and Deisenroth M. Pareto front modeling for sensitivity analysis in multi-objective Bayesian optimization. In *NIPS Workshop on Bayesian Optimization 2014*, 2014.

[CPG14] Chevalier C., Picheny V., and Ginsbourger D. KrigInv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational Statistics & Data Analysis*, 71:1021–1034, 2014.

[CPV14] Contal E., Perchet V., and Vayatis N. Gaussian process optimization with mutual information. In *Proceedings of The 31st International Conference on Machine Learning*, pages 253–261, 2014.

[Cre93] Cressie N. *Statistics for Spatial Data*. Wiley-Interscience New York, 1993.

[CS03] Collette Y. and Siarry P. *Multiobjective Optimization: Principles and Case Studies*. Springer Science & Business Media, 2003.

[CZ14] Chen Y. and Zou X. Runtime analysis of a multi-objective evolutionary algorithm for obtaining finite approximations of Pareto fronts. *Information Sciences*, 262:62–77, 2014.

[DBR13a] Di Bernardino E. and Rullière D. Distortions of multivariate distribution functions and associated level curves: Applications in multivariate risk theory. *Insurance: Mathematics and Economics*, 53(1):190 – 205, 2013.

[DBR13b] Di Bernardino E. and Rullière D. On certain transformations of Archimedean copulas : Application to the non-parametric estimation of their generators. *Dependence Modeling*, 1:1–36, 2013.

[DCD99] Dasgupta P., Chakrabarti P., and DeSarkar S. *Multiobjective heuristic search: An introduction to intelligent search methods for multicriteria optimization*. Springer Science & Business Media, 1999.

[Deb01] Deb K. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[Deb08] Deb K. Introduction to evolutionary multiobjective optimization. In Branke J., Deb K., Miettinen K., and Słowiński R., editors, *Multiobjective Optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 59–96. Springer Berlin Heidelberg, 2008.

[Deh79] Deheuvels P. La fonction de dépendance empirique et ses propriétés. *Acad. Roy. Belg. Bull. Cl. Sci.*, 65(5):274–292, 1979.

[dFF10] da Fonseca V. G. and Fonseca C. M. The attainment-function approach to stochastic multiobjective optimizer assessment and comparison. In *Experimental methods for the analysis of optimization algorithms*, pages 103–130. Springer, 2010.

[dFZS12] de Freitas N., Zoghi M., and Smola A. J. Exponential regret bounds for Gaussian process bandits with deterministic observations. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1743–1750, 2012.

[DGR12] Durrande N., Ginsbourger D., and Roustant O. Additive kernels for Gaussian process modeling. *Annales de la Facultée de Sciences de Toulouse*, 21(3):481–499, 2012.

[DHF15] Dupuy D., Helbert C., and Franco J. DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38, 2015.

[DKB12] Desautels T., Krause A., and Burdick J. Parallelizing exploration-exploitation tradeoffs with Gaussian process bandit optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2012.

[DKB14] Desautels T., Krause A., and Burdick J. W. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *The Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

[DKC13] Djolonga J., Krause A., and Cevher V. High-dimensional Gaussian process bandits. In *Advances in Neural Information Processing Systems*, pages 1025–1033, 2013.

[DKP08] Dimitrova D., Kaishev V., and Penev S. GeD spline estimation of multivariate Archimedean copulas. *Computational Statistics & Data Analysis*, 52(7):3570–3582, 2008.

[DL13] Damianou A. C. and Lawrence N. D. Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 207–215, 2013.

[Don00] Donoho D. L. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000.

[DPAM02] Deb K., Pratap A., Agarwal S., and Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[DR07] Diggle P. and Ribeiro P. J. *Model-based geostatistics*. Springer, 2007.

[DS06] Deb K. and Srinivasan A. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1629–1636. ACM, 2006.

[DTLZ05] Deb K., Thiele L., Laumanns M., and Zitzler E. Scalable test problems for evolutionary multiobjective optimization. In Abraham A., Jain L., and Goldberg R., editors, *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pages 105–145. Springer London, 2005.

[Duo14] Duong T. *ks: Kernel smoothing*, 2014. R package version 1.9.2.

[Dur11] Durrande N. *Étude de classes de noyaux adaptées à la simplification et à l'interprétation des modèles d'approximation. Une approche fonctionnelle et probabiliste.* PhD thesis, Saint-Etienne, EMSE, 2011.

[Duv14] Duvenaud D. K. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

[DVD13] Da Veiga S. and Delbos F. Robust optimization for expensive simulators with surrogate models: application to well placement for oil recovery. In *11th International Conference on Structural Safety & Reliability*, 2013.

[Edd13] Eddelbuettel D. *Seamless R and C++ Integration with Rcpp*. Springer, New York, 2013. ISBN 978-1-4614-6867-7.

[EDK11] Emmerich M. T., Deutz A. H., and Klinkenberg J. W. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE, 2011.

[EF11] Eddelbuettel D. and François R. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

[EGBHC13] Erdely A., González-Barrios J., and Hernández-Cedillo M. Frank's condition for multivariate Archimedean copulas. *Fuzzy Sets and Systems*, In press(Available online), 2013.

[EGN06] Emmerich M., Giannakoglou K., and Naujoks B. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *Evolutionary Computation, IEEE Transactions on*, 10(4):421–439, 2006.

[Ehr05] Ehrgott M. *Multicriteria optimization*. Springer, 2005.

[EL06] Emery X. and Lantuéjoul C. Tbsim: A computer program for conditional simulation of three-dimensional gaussian random fields via the turning bands method. *Computers & Geosciences*, 32(10):1615–1628, 2006.

[ELM03] Embrechts P., Lindskog F., and McNeil A. Modelling dependence with copulas and applications to risk management. *Handbook of heavy tailed distributions in finance*, 8(1):329–384, 2003.

[Emm05] Emmerich M. T. M. *Single- and Multi-objective Evolutionary Design Optimization*. PhD thesis, TU Dortmund, 2005.

[ES81] Efron B. and Stein C. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.

[Esp11] Espinasse T. *Champs et processus gaussiens indexés par des graphes, estimation et prédiction*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2011.

[EU11] Erfani T. and Utyuzhnikov S. Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization. *Engineering Optimization*, 43(5):467–484, 2011.

[FBV15] Feliot P., Bect J., and Vazquez E. A Bayesian approach to constrained multi-objective optimization. In *Learning and Intelligent Optimization: 9th International Conference, LION 9, Lille, France, January 12-15, 2015. Revised Selected Papers*, volume 8994, page 256. Springer, 2015.

[FC12] Frazier P. I. and Clark S. C. Parallel global optimization using an improved multi-points expected improvement criterion. In *INFORMS Optimization Society Conference, Miami FL*, volume 26, 2012.

[FDFP05] Fonseca C. M., Da Fonseca V. G., and Paquete L. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In *Evolutionary Multi-Criterion Optimization*, pages 250–264. Springer, 2005.

[FK09] Forrester A. and Keane A. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

[FLS05] Fang K.-T., Li R., and Sudjianto A. *Design and modeling for computer experiments*. CRC Press, 2005.

[FMRJ14] Fruth J., Muehlenstaedt T., Roustant O., and Jastrow M. *fanovaGraph: Building Kriging models from FANOVA graphs*, 2014. R package version 1.4.7.

[FRM13] Fruth J., Roustant O., and Muehlenstaedt T. The fanovaGraph package: Visualization of interaction structures and construction of block-additive kriging models. *http://hal.archives-ouvertes.fr/hal-00795229*, 2013.

[Fro14] Froment P. *Optimisation de formes paramétriques en grande dimension*. PhD thesis, Ecole centrale de Lyon, 2014.

[FSK08] Forrester A., Sobester A., and Keane A. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[GA10] Grosan C. and Abraham A. Approximating Pareto frontier using a hybrid line search approach. *Information Sciences*, 180(14):2674 – 2695, 2010. Including Special Section on Hybrid Intelligent Algorithms and Applications.

[GAOST10] Grünewälder S., Audibert J.-Y., Opper M., and Shawe-Taylor J. Regret bounds for Gaussian process bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 273–280, 2010.

[GBC+14] Ginsbourger D., Baccou J., Chevalier C., Perales F., Garland N., and Monerie Y. Bayesian adaptive reconstruction of profile optima and optimizers. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):490–510, 2014.

[GCLD09] Gorissen D., Couckuyt I., Laermans E., and Dhaene T. Pareto-based multi-output metamodeling with active learning. In *EANN 2009*, volume 43 of *CCIS*, page 389–400. Springer, 2009.

[GF14] Giagkiozis I. and Fleming P. J. Pareto front estimation for decision making. *Evolutionary computation*, 22(4):651–678, 2014.

[GF15] Giagkiozis I. and Fleming P. Methods for multi-objective optimization: An analysis. *Information Sciences*, 293(0):338 – 350, 2015.

[GGD+14] Gramacy R. B., Gray G. A., Digabel S. L., Lee H. K., Ranjan P., Wells G., and Wild S. M. Modeling an augmented Lagrangian for improved blackbox constrained optimization. *arXiv preprint arXiv:1403.4890*, 2014.

[GGS12] Girard S., Guillou A., and Stupfler G. Estimating an endpoint with high-order moments. *test*, 21(4):697–729, 2012.

[Gin09] Ginsbourger D. *Multiples metamodeles pour l'approximation et l'optimisation de fonctions numeriques multivariables*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2009.

[Gir04] Girard A. *Approximate methods for propagation of uncertainty with Gaussian process models*. PhD thesis, Citeseer, 2004.

[GJGM15] Genest L., Jézéquel L., Gillot F., and Mercier F. Shape optimization method for crashworthiness design based on equivalent static loads concept. In *11th World Congress on Structural and Multidisciplinary Optimization*, 2015.

[GLR10] Ginsbourger D. and Le Riche R. Towards Gaussian process-based optimization with finite time horizon. In *mODa 9–Advances in Model-Oriented Design and Analysis*, pages 89–96. Springer, 2010.

[GLRC10] Ginsbourger D., Le Riche R., and Carraro L. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer, 2010.

[GNZ11] Genest C., Nešlehová J., and Ziegel J. Inference in multivariate Archimedean copula models. *TEST*, 20(2):223–256, 2011.

[GOH13] Garnett R., Osborne M. A., and Hennig P. Active learning of linear embeddings for Gaussian processes. *arXiv preprint arXiv:1310.6740*, 2013.

[GOR10] Garnett R., Osborne M. A., and Roberts S. J. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 209–219. ACM, 2010.

[GPL+13a] Gao Y., Peng L., Li F., Liu M., and Hu X. Estimation of distribution algorithm with multivariate t-copulas for multi-objective optimization. *Intelligent Control and Automation*, 4:63, 2013.

[GPL+13b] Gao Y., Peng L., Li F., Liu M., and Liu W. Archimedean copula–based estimation of distribution algorithm for multi–objective optimisation. *International Journal of Trust Management in Computing and Communications*, 1(3):200–211, 2013.

[GR93] Genest C. and Rivest L.-P. Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American statistical Association*, 88(423):1034–1043, 1993.

[GRD13] Ginsbourger D., Roustant O., and Durrande N. Invariances of random fields paths, with applications in Gaussian process regression. *arXiv preprint arXiv:1308.1359*, 2013.

[GRS+14] Ginsbourger D., Roustant O., Schuhmacher D., Durrande N., and Lenz N. On ANOVA decompositions of kernels and Gaussian random field paths. *arXiv preprint arXiv:1409.6008*, 2014.

[GSA14] Gelbart M. A., Snoek J., and Adams R. P. Bayesian optimization with unknown constraints. In *UAI*, 2014.

[GVH+07] Goel T., Vaidyanathan R., Haftka R., Shyy W., Queipo N., and Tucker K. Response surface approximation of Pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, 196(4):879–893, 2007.

[Hal82] Hall P. On estimating the endpoint of a distribution. *The Annals of Statistics*, pages 556–568, 1982.

[HBdF11] Hoffman M. D., Brochu E., and de Freitas N. Portfolio allocation for Bayesian optimization. In *UAI*, pages 327–336. Citeseer, 2011.

[HDYE15] Hupkens I., Deutz A., Yang K., and Emmerich M. Faster exact algorithms for computing expected hypervolume improvement. In Gaspar-Cunha A., Henggeler Antunes C., and Coello C. C., editors, *Evolutionary Multi-Criterion Optimization*, volume 9019 of *Lecture Notes in Computer Science*, pages 65–79. Springer International Publishing, 2015.

[HHBW06] Huband S., Hingston P., Barone L., and While L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, Oct 2006.

[HHLB11] Hutter F., Hoos H. H., and Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

[Hil01] Hillermeier C. *Nonlinear multiobjective optimization: a generalized homotopy approach*, volume 135. Springer Science & Business Media, 2001.

[HK07] Henkenjohann N. and Kunert J. An efficient sequential optimization approach based on the multivariate expected improvement criterion. *Quality Engineering*, 2007.

[HKMY14] Hofert M., Kojadinovic I., Maechler M., and Yan J. *copula: Multivariate Dependence with Copulas*, 2014. R package version 0.999-10.

[HLHG14] Hernández-Lobato J. M., Hoffman M. W., and Ghahramani Z. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pages 918–926, 2014.

[HM11] Hofert M. and Mächler M. Nested Archimedean copulas meet R: The nacopula package. *Journal of Statistical Software*, 39(9):1–20, 2011.

[HNS97] Hall P., Nussbaum M., and Stern S. On the estimation of a support curve of indeterminate sharpness. *Journal of Multivariate Analysis*, 62(2):204–232, 1997.

[Hos95] Hoshiya M. Kriging and conditional simulation of Gaussian field. *Journal of engineering mechanics*, 121(2):181–186, 1995.

[HS93] Handcock M. S. and Stein M. L. A Bayesian analysis of kriging. *Technometrics*, 35(4):403–410, 1993.

[HS08] Hawe G. and Sykulski J. Probability of improvement methods for constrained multi-objective optimization. In *Computation in Electromagnetics, 2008. CEM 2008. 2008 IET 7th International Conference on*, pages 50–51. IET, 2008.

[HS12] Hennig P. and Schuler C. J. Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 13(1):1809–1837, 2012.

[HTFF05] Hastie T., Tibshirani R., Friedman J., and Franklin J. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[Hüs10] Hüsler J. On applications of extreme value theory in optimization. In *Experimental Methods for the Analysis of Optimization Algorithms*, pages 185–207. Springer, 2010.

[HW99] Hall P. and Wang J. Estimating the end-point of a probability distribution using minimum-distance methods. *Bernoulli*, pages 177–189, 1999.

[IBFM10] Iooss B., Boussouf L., Feuillard V., and Marrel A. Numerical studies of the metamodel fitting and validation processes. *International Journal On Advances in Systems and Measurements*, 3(1 and 2):11–21, 2010.

[IK14] Ivanov M. and Kuhnt S. A parallel optimization algorithm based on FANOVA decomposition. *Quality and Reliability Engineering International*, 30(7):961–974, 2014.

[IL15] Iooss B. and Lemaître P. A review on global sensitivity analysis methods. In Meloni C. and Dellino G., editors, *Uncertainty management in Simulation-Optimization of Complex Systems: Algorithms and Applications*. Springer, 2015.

[Jaw09] Jaworski P. On copulas and their diagonals. *Information Sciences*, 179(17):2863–2871, 2009.

[JLRGG12] Janusevskis J., Le Riche R., Ginsbourger D., and Girdziusas R. Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In *Learning and Intelligent Optimization*, pages 413–418. Springer, 2012.

[Jon01] Jones D. R. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[Jou74] Journel A. G. Geostatistics for conditional simulation of ore bodies. *Economic Geology*, 69(5):673–687, 1974.

[JR89] Journel A. G. and Rossi M. When do we need a trend model in kriging? *Mathematical Geology*, 21(7):715–739, 1989.

[JS03] Jin Y. and Sendhoff B. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 1910–1917. IEEE, 2003.

[JSW98] Jones D., Schonlau M., and Welch W. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[KCL11] Koziel S., Ciaurri D. E., and Leifsson L. Surrogate-based methods. In *Computational Optimization, Methods and Algorithms*, pages 33–59. Springer, 2011.

[Kea06] Keane A. J. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.

[KKPT14] König S., Kazianka H., Pilz J., and Temme J. Estimation of nonstrict Archimedean copulas and its application to quantum networks. *Applied Stochastic Models in Business and Industry*, 2014.

[Kle07] Kleijnen J. P. *Design and analysis of simulation experiments*, volume 111. Springer Science & Business Media, 2007.

[KM14] Kleijnen J. P. and Mehdad E. Multivariate versus univariate kriging metamodels for multi-response simulation models. *European Journal of Operational Research*, 236(2):573–582, 2014.

[Kno06] Knowles J. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, February 2006.

[KSS07] Kim G., Silvapulle M. J., and Silvapulle P. Comparison of semiparametric and parametric methods for estimating copulas. *Computational Statistics & Data Analysis*, 51(6):2836–2850, 2007.

[Kus64] Kushner H. J. 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.

[KW70] Kimeldorf G. S. and Wahba G. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, pages 495–502, 1970.

[KWE+15] Koch P., Wagner T., Emmerich M. T., Bäck T., and Konen W. Efficient multi-criteria optimization on noisy machine learning problems. *Applied Soft Computing*, 29:357–370, 2015.

[KY10] Kojadinovic I. and Yan J. Comparison of three semiparametric methods for estimating dependence parameters in copula models. *Insurance: Mathematics and Economics*, 47(1):52–63, 2010.

[Law05] Lawrence N. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

[LDS01] Lovisolo L. and Da Silva E. Uniform distribution of points on a hypersphere with applications to vector bit-plane encoding. *IEE Proceedings-Vision, Image and Signal Processing*, 148(3):187–193, 2001.

[LG13] Le Gratiet L. *Multi-fidelity Gaussian process regression for computer experiments*. PhD thesis, Université Paris-Diderot-Paris VII, 2013.

[Liz08] Lizotte D. J. *Practical Bayesian optimization*. University of Alberta, 2008.

[LLY+08] Liao X., Li Q., Yang X., Zhang W., and Li W. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35(6):561–569, June 2008.

[Loh84] Loh W.-Y. Estimating an endpoint of a distribution with resampling methods. *The Annals of Statistics*, pages 1543–1550, 1984.

[Lot05] Lotov A. V. Approximation and visualization of Pareto frontier in the framework of classical approach to multi-objective optimization. In *Practical Approaches to Multi-Objective Optimization*, 2005.

[Lov12] Lovison A. Global search perspectives for multiobjective optimization. *Journal of Global Optimization*, June 2012.

[LPX11] Li D., Peng L., and Xu X. Bias reduction for endpoint estimation. *Extremes*, 14(4):393–412, 2011.

[LQC06] Lawrence N. D. and Quinonero-Candela J. Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the 23rd international conference on Machine learning*, pages 513–520. ACM, 2006.

[LSA+13] Le V. T. H., Stoica C., Alamo T., Camacho E. F., and Dumur D. Uncertainty representation based on set theory. *Zonotopes*, pages 1–26, 2013.

[LZ09] Li H. and Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 13(2):284–302, 2009.

[LZ11] Laumanns M. and Zenklusen R. Stochastic convergence of random search methods to fixed size Pareto front approximations. *European Journal of Operational Research*, 213(2):414 – 421, 2011.

[LZG14] Liu B., Zhang Q., and Gielen G. G. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *Evolutionary Computation, IEEE Transactions on*, 18(2):180–192, 2014.

[MA04] Marler R. and Arora J. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, April 2004.

[Mar14] Marmin S. Developpements pour l'evaluation et la maximisation du critere d'amelioration esperee multipoint en optimisation globale. Master's thesis, Ecole nationale supérieure des Mines de Saint-Etienne, 2014.

[Mat63] Matheron G. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[Mat69] Matheron G. Le krigeage universel. *Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau*, 1, 1969.

[MC15] Martinez-Cantin R. Local nonstationarity for efficient Bayesian optimization. *arXiv preprint arXiv:1506.02080*, 2015.

[MCG15] Marmin S., Chevalier C., and Ginsbourger D. Differentiating the multipoint expected improvement for optimal batch design. *arXiv preprint arXiv:1503.05509*, 2015.

[McM71] McMullen P. On zonotopes. *Transactions of the American Mathematical Society*, 159:91–109, 1971.

[Meh15] Mehdad E. *Kriging metamodels and global opimization in simulation*. PhD thesis, Tilburg University, School of Economics and Management, 2015.

[MGB+15] Marmin S., Ginsbourger D., Baccou J., Perales F., and Liandrat J. Processus gaussiens déformés pour l'apprentissage de zones instationnaires. In *47th annual meeting of the French Statistical Society*, 2015.

[MIDV08] Marrel A., Iooss B., Dorpe F. V., and Volkova E. An efficient methodology for modeling complex computer codes with Gaussian processes. *Computational Statistics & Data Analysis*, 52(10):4731 – 4744, 2008.

[Mie99] Miettinen K. *Nonlinear multiobjective optimization*, volume 12. Springer, 1999.

[MN09] McNeil A. and Nešlehová J. Multivariate Archimedean copulas, d-monotone functions and l_1-norm symmetric distributions. *The Annals of Statistics*, 37(5B):3059–3097, 2009.

[Moc89] Mockus J. *Bayesian approach to global optimization*. Springer, 1989.

[Mol05] Molchanov I. S. *Theory of random sets*. Springer, 2005.

[Mon12] Montaño A. A. *Design of Multi-Objective Evolutionary Algorithms for Aeronautical Problems*. PhD thesis, Instituto Politécnico nacional, 2012.

[MRCK12] Muehlenstaedt T., Roustant O., Carraro L., and Kuhnt S. Data-driven kriging models based on FANOVA-decomposition. *Statistics and Computing*, 22(3):723–738, 2012.

[MS05] Martin J. D. and Simpson T. W. Use of kriging models to approximate deterministic computer models. *AIAA Journal*, 43(4):853–863, 2005.

[MS11] Mebane W. R. J. and Sekhon J. S. Genetic optimization using derivatives: the rgenoud package for R. *Journal of Statistical Software*, 42(11):1–26, 2011.

[MTZ78] Mockus J., Tiesis V., and Zilinskas A. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[Nea96] Neal R. M. *Bayesian learning for neural networks*, volume 118 of *Lecture Notes in Statistics*. Springer, 1996.

[Nel99] Nelsen R. *An introduction to copulas*, volume 139 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1999.

[NQMRLÚF03] Nelsen R., Quesada-Molinab J., Rodriguez-Lallenac J., and Úbeda-Floresc M. Kendall distribution functions. *Statistics and Probability Letters*, 65:263–268, 2003.

[NS09] Nappo G. and Spizzichino F. Kendall distributions and level sets in bivariate exchangeable survival models. *Information Sciences*, 179(17):2878–2890, 2009.

[Oak99] Oakley J. *Bayesian uncertainty analysis for complex computer codes*. PhD thesis, University of Sheffield, 1999.

[OGR09] Osborne M. A., Garnett R., and Roberts S. J. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15, 2009.

[OGV09] Omelka M., Gijbels I., and Veraverbeke N. Improved kernel estimation of copulas: weak convergence and goodness-of-fit testing. *The Annals of Statistics*, 37(5B):3023–3058, 2009.

[OK78] O'Hagan A. and Kingman J. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

[Osb10] Osborne M. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University New College, 2010.

[Par12] Parr J. M. *Improvement Criteria for Constraint Handling and Multiobjective Optimization*. PhD thesis, University of Southampton, 2012.

[PGOP00] Poloni C., Giurgevich A., Onesti L., and Pediroda V. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 186(2):403–420, 2000.

[Pic13] Picheny V. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, pages 1–16, 2013.

[Pic14] Picheny V. A stepwise uncertainty reduction approach to constrained global optimization. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 2014.

[PKFH12] Parr J., Keane A., Forrester A. I., and Holden C. Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization*, 44(10):1147–1166, 2012.

[PM12] Pronzato L. and Müller W. G. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[PWBV08] Ponweiser W., Wagner T., Biermann D., and Vincze M. Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection. In *Parallel Problem Solving from Nature–PPSN X*, pages 784–794. Springer, 2008.

[PWG12] Preuss M., Wagner T., and Ginsbourger D. High-dimensional model-based optimization based on noisy evaluations of computer games. In *LION*, pages 145–159. Springer, 2012.

[QPNV10] Queipo N., Pintos S., Nava E., and Verde A. Setting targets for surrogate-based optimization. *Journal of Global Optimization*, pages 1–19, 2010.

[R C15] R Core Team . *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[RGD12] Roustant O., Ginsbourger D., and Deville Y. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

[Roc70] Rockafellar R. T. *Convex analysis*. Princeton mathematical series 28. Princeton University Press, 1970.

[Ros12] Rosenblatt N. *Contribution à la conception robuste de véhicules en choc frontal : détection de défaillances en crash*. PhD thesis, Ecole centrale de Lyon, 2012.

[RW06] Rasmussen C. E. and Williams C. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[SBCdF15] Shahriari B., Bouchard-Côté A., and de Freitas N. Unbounded Bayesian optimization via regularization. *arXiv preprint arXiv:1508.03666*, 2015.

[Sch97] Schonlau M. *Computer experiments and global optimization*. PhD thesis, University of Waterloo, 1997.

[Sch09] Scheuerer M. *A comparison of models and methods for spatial interpolation in statistics and numerical analysis*. PhD thesis, Ph. D. thesis, Univ. Göttingen, 2009.

[SFT+12] Sóbester A., Forrester A. I., Toal D. J., Tresidder E., and Tucker S. Engineering design applications of surrogate-assisted optimization techniques. *Optimization and Engineering*, pages 1–23, 2012.

[Sha03] Shapiro A. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[SHSMV14] Steponavičė I., Hyndman R. J., Smith-Miles K., and Villanova L. Efficient identification of the Pareto optimal set. In *Learning and Intelligent Optimization*, pages 341–352. Springer, 2014.

[SK97] Saff E. B. and Kuijlaars A. B. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1):5–11, 1997.

[SK07] Song W. and Keane A. J. Surrogate-based aerodynamic shape optimization of a civil aircraft engine nacelle. *AIAA journal*, 45(10):2565–2574, 2007.

[Skl59] Sklar M. *Fonctions de répartition à n dimensions et leurs marges*. Université Paris 8, 1959.

[SKSK10] Srinivas N., Krause A., Seeger M., and Kakade S. M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1015–1022, 2010.

[SLA12] Snoek J., Larochelle H., and Adams R. P. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[Sob01] Sobol I. M. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation*, 55(1):271–280, 2001.

[SPKA01] Simpson T. W., Poplinski J., Koch P. N., and Allen J. K. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with computers*, 17(2):129–150, 2001.

[SQMC10] Santana-Quintero L., Montano A., and Coello C. A review of techniques for handling expensive functions in evolutionary multi-objective optimization. *Computational Intelligence in Expensive Optimization Problems*, pages 29–59, 2010.

[SS16] Svenson J. and Santner T. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, 94:250 – 264, 2016.

[SSJO12] Shimoyama K., Sato K., Jeong S., and Obayashi S. Comparison of the criteria for updating kriging response surface models in multi-objective optimization. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

[SST06] Sankar A., Spielman D. A., and Teng S.-H. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM Journal on Matrix Analysis and Applications*, 28(2):446–476, 2006.

[SSZA14] Snoek J., Swersky K., Zemel R. S., and Adams R. P. Input warping for Bayesian optimization of non-stationary functions. *In ICML*, 2014.

[Sta87] Stadler W. Initiators of multicriteria optimization. In *Recent Advances and Historical Development of Vector Optimization*, pages 3–47. Springer, 1987.

[Ste99] Stein M. L. *Interpolation of spatial data: some theory for kriging*. Springer, 1999.

[Sve11] Svenson J. D. *Computer Experiments: Multiobjective Optimization and Sensitivity Analysis*. PhD thesis, The Ohio State University, 2011.

[SW10] Shan S. and Wang G. G. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010.

[SWG14] Shah A., Wilson A. G., and Ghahramani Z. Student-t processes as alternatives to Gaussian processes. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 877–885, 2014.

[SWJ98] Schonlau M., Welch W. J., and Jones D. R. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25, 1998.

[SWMW89] Sacks J., Welch W. J., Mitchell T. J., and Wynn H. P. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

[SWN03] Santner T. J., Williams B. J., and Notz W. I. *The design and analysis of computer experiments*. Springer Science & Business Media, 2003.

[THH+15] Tabatabaei M., Hakanen J., Hartikainen M., Miettinen K., and Sindhya K. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52(1):1–25, 2015.

[TJR98] Tamiz M., Jones D., and Romero C. Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of Operational Research*, 111(3):569 – 581, 1998.

[TL10] Titsias M. K. and Lawrence N. D. Bayesian Gaussian process latent variable model. In *International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

[TOG04] Toth C. D., O'Rourke J., and Goodman J. E. *Handbook of discrete and computational geometry*. CRC press, 2004.

[TWG07] Taylor J. E., Worsley K. J., and Gosselin F. Maxima of discretely sampled random fields, with an application to 'bubbles'. *Biometrika*, 94(1):1–18, 2007.

[VB10] Vazquez E. and Bect J. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095, 2010.

[Vil08] Villemonteix J. *Optimisation de fonctions coûteuses Modèles gaussiens pour une utilisation efficace du budget d'évaluations : théorie et pratique industrielle*. These, Université Paris Sud - Paris XI, December 2008.

[VJFK11] Viswanath A., J. Forrester A., and Keane A. Dimension reduction for aerodynamic design optimization. *AIAA journal*, 49(6):1256–1266, 2011.

[VK10] Voutchkov I. and Keane A. Multi-objective optimization using surrogates. *Computational Intelligence in Optimization*, pages 155–175, 2010.

[VSBT14] Viana F. A., Simpson T. W., Balabanov V., and Toropov V. Metamodeling in multidisciplinary design optimization: How far have we really come? *AIAA Journal*, 52(4):670–690, 2014.

[VVL99] Van Veldhuizen D. A. and Lamont G. B. Multiobjective evolutionary algorithm test suites. In *Proceedings of the 1999 ACM symposium on Applied computing*, pages 351–357. ACM, 1999.

[VVW09] Villemonteix J., Vazquez E., and Walter E. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.

[VWF05] Vazquez E., Walter E., and Fleury G. Intrinsic kriging and prior information. *Applied Stochastic Models in Business and Industry*, 21(2):215–226, 2005.

[WA13] Wilson A. and Adams R. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1067–1075, 2013.

[Wag13] Wagner T. *Planning and Multi-Objective Optimization of Manufacturing Processes by Means of Empirical Surrogate Models*. PhD thesis, Technische Universität Dortmund, 2013.

[Wah90] Wahba G. *Spline models for observational data*, volume 59. Siam, 1990.

[War83] Warburton A. Quasiconcave vector maximization: connectedness of the sets of Pareto-optimal and weak Pareto-optimal alternatives. *Journal of optimization theory and applications*, 40(4):537–557, 1983.

[WEDP10] Wagner T., Emmerich M., Deutz A., and Ponweiser W. On expected-improvement criteria for model-based multi-objective optimization. *Parallel Problem Solving from Nature–PPSN XI*, pages 718–727, 2010.

[Wol] WolframAlpha . Wolfram research, accessed August 2014 at http://www.wolframalpha.com/.

[WS07] Wang G. and Shan S. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370, 2007.

[WTM11] Wagner T., Trautmann H., and Martí L. A taxonomy of online stopping criteria for multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 16–30. Springer, 2011.

[WZ10] Wang L.-F. and Zeng J.-C. Estimation of distribution algorithm based on copula theory. In *Exploitation of linkage learning in evolutionary algorithms*, pages 139–162. Springer, 2010.

[WZH+13] Wang Z., Zoghi M., Hutter F., Matheson D., and de Freitas N. Bayesian optimization in high dimensions via random embeddings. *In IJCAI*, 2013.

[Yan07] Yan J. Enjoy the joy of copulas: with a package copula. *Journal of Statistical Software*, 21(4):1–21, 2007.

[ZDT00] Zitzler E., Deb K., and Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[Zie95] Ziegler G. M. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 1995.

[ZK04] Zitzler E. and Künzli S. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.

[ZKT08] Zitzler E., Knowles J., and Thiele L. Quality assessment of Pareto set approximations. In *Multiobjective Optimization*, pages 373–404. Springer, 2008.

[ZLTV10] Zhang Q., Liu W., Tsang E., and Virginas B. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *Evolutionary Computation, IEEE Transactions on*, 14(3):456–474, 2010.

[ZQL+11] Zhou A., Qu B.-Y., Li H., Zhao S.-Z., Suganthan P. N., and Zhang Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[ZQZ11] Zhou Q., Qian P. Z., and Zhou S. A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics*, 53(3), 2011.

[ZSKP13] Zuluaga M., Sergent G., Krause A., and Püschel M. Active learning for multi-objective optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 462–470, 2013.

[ZTL+03] Zitzler E., Thiele L., Laumanns M., Fonseca C. M., and da Fonseca V. G. Performance assessment of multiobjective optimizers: An analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.

[ZZJ08] Zhang Q., Zhou A., and Jin Y. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *Evolutionary Computation, IEEE Transactions on*, 12(1):41–63, 2008.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT: 2015 EMSE 0805

Mickaël BINOIS

Uncertainty quantification on Pareto fronts and high-dimensional strategies in Bayesian optimization, with applications in multi-objective automotive design

Speciality: Applied Mathematics

Keywords: Multi-objective optimization, Gaussian processes, Expected Improvement, SUR, Vorob'ev deviation, REMBO, copulas

Abstract:

This dissertation deals with optimizing expensive or time-consuming black-box functions to obtain the set of all optimal compromise solutions, i.e. the Pareto front. In automotive design, the evaluation budget is severely limited by numerical simulation times of the considered physical phenomena. In this context, it is common to resort to "metamodels" (models of models) of the numerical simulators, especially using Gaussian processes. They enable adding sequentially new observations while balancing local search and exploration. Complementing existing multi-objective Expected Improvement criteria, we propose to estimate the position of the whole Pareto front along with a quantification of the associated uncertainty, from conditional simulations of Gaussian processes. A second contribution addresses this problem from a different angle, using copulas to model the multi-variate cumulative distribution function. To cope with a possibly high number of variables, we adopt the REMBO algorithm. From a randomly selected direction, defined by a matrix, it allows a fast optimization when only a few number of variables are actually influential, but unknown. Several improvements are proposed, such as a dedicated covariance kernel, a selection procedure for the low dimensional domain and of the random directions, as well as an extension to the multi-objective setup. Finally, an industrial application in car crash-worthiness demonstrates significant benefits in terms of performance and number of simulations required. It has also been used to test the R package *GPareto* developed during this thesis.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2015 EMSE 0805

Mickaël BINOIS

Quantification d'incertitude sur fronts de Pareto et stratégies pour l'optimisation bayésienne en grande dimension, avec applications en conception automobile

Spécialité : Mathématiques Appliquées

Mots clefs : Optimisation multiobjectif, Processus Gaussiens, Expected Improvement, SUR, Vorob'ev deviation, REMBO, copules

Résumé:

Cette thèse traite de l'optimisation multiobjectif de fonctions coûteuses, aboutissant à la construction d'un front de Pareto représentant l'ensemble des compromis optimaux. En conception automobile, le budget d'évaluations est fortement limité par les temps de simulation numérique des phénomènes physiques considérés. Dans ce contexte, il est courant d'avoir recours à des « métamodèles » (ou modèles de modèles) des simulateurs numériques, en se basant notamment sur des processus gaussiens. Ils permettent d'ajouter séquentiellement des observations en conciliant recherche locale et exploration. En complément des critères d'optimisation existants tels que des versions multiobjectifs du critère d'amélioration espérée, nous proposons d'estimer la position de l'ensemble du front de Pareto avec une quantification de l'incertitude associée, à partir de simulations conditionnelles de processus gaussiens. Une deuxième contribution reprend ce problème à partir de copules. Pour pouvoir traiter le cas d'un grand nombre de variables d'entrées, nous nous basons sur l'algorithme *REMBO*. Par un tirage aléatoire directionnel, défini par une matrice, il permet de trouver un optimum rapidement lorsque seules quelques variables sont réellement influentes (mais inconnues). Plusieurs améliorations sont proposées, elles comprennent un noyau de covariance dédié, une sélection du domaine de petite dimension et des directions aléatoires mais aussi l'extension au cas multiobjectif. Enfin, un cas d'application industriel en crash a permis d'obtenir des gains significatifs en performance et en nombre de calculs requis, ainsi que de tester le package R *GPareto* développé dans le cadre de cette thèse.