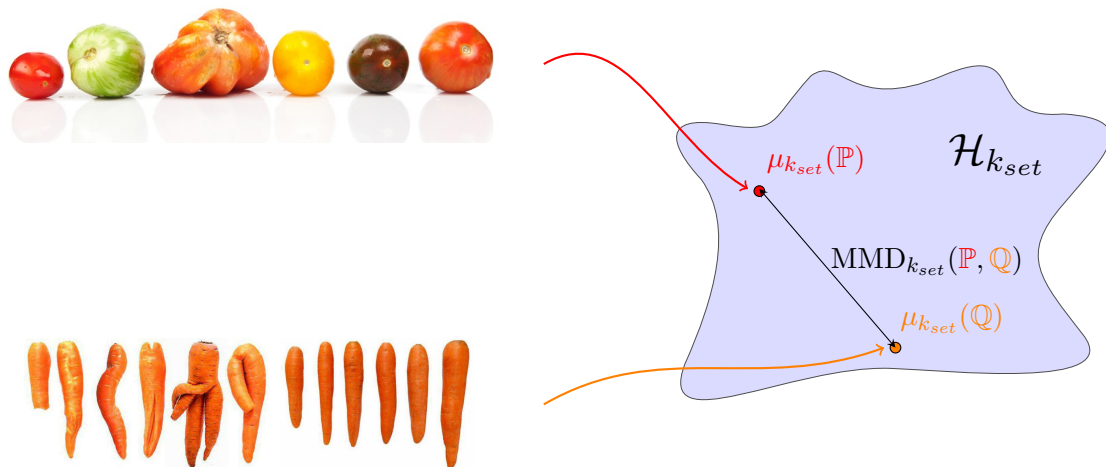


# Sensitivity analysis for Bayesian optimization with uncertainties

Sensitivity analysis of random sets



Noé Fellmann

Thèse de doctorat



**THÈSE DE DOCTORAT DE L'ÉCOLE CENTRALE DE LYON**  
**membre de l'Université de Lyon**

**École Doctorale N° 512**  
**InfoMaths - Informatique et Mathématiques de Lyon**

**Spécialité de doctorat : Mathématiques**

Soutenance publique le 12 décembre 2024, par :

**Noé FELLMANN**

---

**Analyse de sensibilité pour l'optimisation bayésienne de  
boîtes noires coûteuses sous contraintes et sous  
incertitudes.**  
**Analyse de sensibilité d'ensembles aléatoires.**

---

Devant le jury composé de :

|                        |                                                   |                        |
|------------------------|---------------------------------------------------|------------------------|
| Sébastien Da Veiga     | Associate Professor, CREST-ENSAI                  | Rapporteur             |
| Luc Pronzato           | Directeur de recherche, Université Côte d'Azur    | Rapporteur             |
| Agnès Lagnoux          | Maîtresse de conférences, Université Toulouse     | Examinatrice           |
| Sébastien Le Digabel   | Professeur, Polytechnique Montréal                | Examineur              |
| Guillaume Perrin       | Directeur de recherche, Université Gustave Eiffel | Examineur              |
| Clémentine Prieur      | Professeure, Université Grenoble-Alpes            | Examinatrice           |
| Céline Helbert         | Maîtresse de conférences, École Centrale de Lyon  | Directrice de thèse    |
| Christophette Blanchet | Maîtresse de conférences, École Centrale de Lyon  | Co-directrice de thèse |
| Delphine Sinoquet      | Ingénieure docteure, IFP Énergies Nouvelles Rueil | Encadrante de thèse    |
| Adrien Spagnol         | Ingénieur docteur, IFP Énergies Nouvelles Solaize | Encadrant de thèse     |

Unité de recherche : ICJ - Institut Camille Jordan



## Remerciements

M'y voilà enfin, les remerciements. Ils sont pour vous le début de ce manuscrit et peut-être la seule partie que vous lirez en détail, mais ils sont pour moi les derniers mots que j'y écris. Je vais faire de mon mieux pour oublier le moins de personnes possible. Si vous ne vous y trouvez pas, ne vous vexez pas, ça ne veut pas (forcément) dire que je vous déteste.

Merci tout d'abord aux membres de mon jury. Merci à Sébastien Da Veiga et Luc Pronzato d'avoir relu mon manuscrit, et merci pour vos commentaires pertinents. Merci à Agnès Lagnoux, Clémentine Prieur, Guillaume Perrin et Sébastien Le Digabel d'avoir accepté d'être les examinateurs de ma défense de thèse.

Merci à mon équipe encadrante qui m'a accompagné pendant ces trois années. Merci Céline pour ton encadrement de qualité, ta bonne humeur permanente, tes idées, ta vision toujours positive, ton intérêt non seulement pour le bon déroulement de la thèse, mais aussi pour le bien-être de ton/tes doctorant-e-s. Merci Christophette pour ta présence, tes idées, ta rigueur, merci d'avoir toujours été disponible dès que j'avais des questions. Merci Delphine pour ta bonne humeur, ta motivation, ton optimisme, tes remarques pertinentes, et ton suivi malgré le distanciel (et merci pour les bières en conf). Merci Adrien pour ton encadrement, et aussi pour les discussions sur autre chose que la thèse (qui font souvent du bien !). Encore merci à tous les quatre.

Merci aux collègues de travail, que ce soit côté Centrale ou IFPEN, et aux nombreuses rencontres en conférence ou autres. Merci Gabriel pour les nombreuses remarques sur mon premier article. Merci Mathis et Adan pour les échanges et pour vos cas tests faciles d'utilisation. Merci aussi à la team des doctorant-e-s de l'IFPEN que j'ai perdue de vue mais qui m'ont accompagné surtout pendant le début de la thèse.

Merci aux passagers et passagères du bureau 6234 (et pas 394) dans lequel s'est déroulée la grande majorité de ma thèse. Merci Angèle et Thierry pour votre accueil à mon arrivée et pour les nombreuses parties de Hanabi (entre autres). Merci encore Angèle, pour tes nombreux conseils et pour (entre autres) le template du manuscrit que vous êtes en train de lire ! Merci Nicolas pour les séances d'escalade et pour ma première manif. Merci aux stagiaires qui ont parsemé ma période de thèse. Merci Antoine, d'abord co-bureau puis colocataire, pour les  $n \uparrow \uparrow n$  parties de jeu de société (et pour les trois où je t'ai battu et encore je suis généreux). Merci Guerlain, alias *GuéGué, meilleur partenaire-de-coinche-qui-n'ait-jamais-existé-et-qui-n'existera-(presque-sûrement)-jamais, Michelin-fan, One-Piece-kai-rusher, new-VP-VP, Frostpunk-oublier-de-règle*, d'être toujours à l'écoute surtout quand il faut redire ce qu'il y a après di, re ce qu'il y a après di. Merci Victor, le plus fort. Merci Elliott la racailliotte. Merci Romane, la vegane (ouais je te définis par ça dsl mais ça rimait). Merci Ève pour le rêve.

Merci à tous mes ami-e-s ( $\notin$  Bureau 6234) qui m'ont accompagné pendant cette thèse. D'abord merci à mes premiers colocs, Antoine, Sarah et Corentin. Vous avez pu juger de mon investissement et de mon sérieux. Merci à ceux qui ont suivi, Bastien, Bloohm (ou Bluhm?), Astrrr et la Caps bien sûr. Merci Astrée pour l'énergie et la joie que tu partages, pour les puzzles, et bien sûr

pour 100% physique. Merci la Caps, tu m'as apporté (quasi) que de la joie par ta choupinesse. Merci Mathieu, merci pour tes discussions toujours intéressantes et pertinentes, merci à toi d'être quelqu'un d'ouvert, gentil, à l'écoute, inspirant. Merci Thomas pour les tours de stade, merci d'être là pour moi, merci d'écouter ma vie et merci de me raconter la tienne, merci pour tes convictions et ce que tu en fais. Merci Olivia, pour ta gentillesse et ton affection. Je sens que tu fais attention à moi et ça me fait vraiment plaisir. Merci encore à tous les trois, vous comptez vraiment pour moi. Merci à toi, Valentin, et merci à toi, Apolline, d'être des personnes géniales et d'accompagner les deux précédemment cités. Merci aux autres amis centraliens principalement du V3, best étage. Merci Ozy, Tythos, vous êtes les bests 2As. Merci Stagou, Methrib, Schödou et Koudh, vous formez une bien belle brochette. Merci Athénais, "ma pote de conf" comme je t'appelle souvent auprès des autres. Je t'ai rencontré à l'Appart puis on ne s'est plus quitté. Tu as rendu mes conférences d'autant plus amusantes, en les parsemant de discussions toujours sérieuses et professionnelles. Merci pour ça. Merci Julien, the bro de l'ifp. Merci de ne jamais rager sur lol, et merci pour les ragots G\*\*\*\*\*iens. Merci à toi, Nathan, partenaire de la génération des Uns, toujours présent de loin, pour ton humour, ta spontanéité et pour être toi. Merci aussi à celles et ceux que je vois moins mais qui comptent pour moi. Merci Kenza, Clémentine, Simon, I (still) ♥ Germain. Merci Louis, Pierre et Adrien, on a quand même passé "quelques" années ensemble.

Et un dernier merci à ma famille. Merci à tous-tes mes cousins, cousines, oncles et tantes. Merci aux participant·e·s du Jandée Globe, vous êtes une bien belle famille. Merci Mamama (c'est le grand frisé qui parle). Merci Clara, d'avoir été là pendant ma thèse, merci pour les séances d'escalade et pour les discussions autour de bière(s). Merci Bulle d'être une personne formidable. Merci Rémi, merci Papa, merci Maman. Merci d'être de belles personnes. Merci d'avoir toujours pris soin de moi. Continuez à vivre votre vie comme vous le faites, vous me faites rêver. Je vous aime.

# Introduction (en français)

## Contexte de la thèse

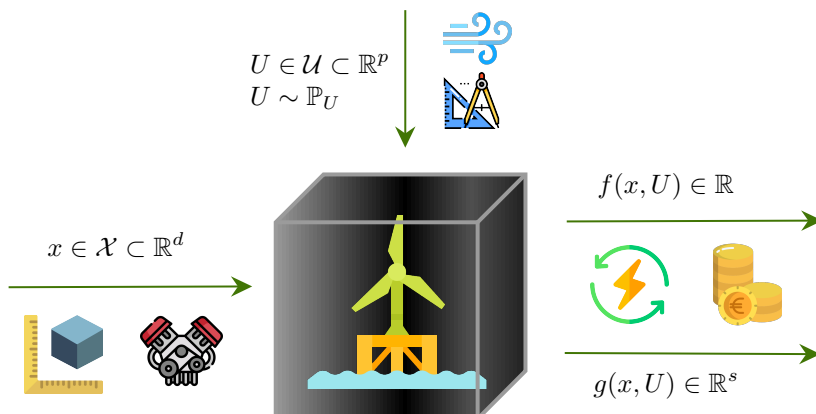
En une phrase, le sujet de cette thèse est de développer des méthodes d'analyse de sensibilité pour faciliter l'optimisation bayésienne de boîtes noires coûteuses sous contraintes et en présence d'entrées incertaines. Ce travail est motivé par deux applications industrielles principales : l'optimisation d'une éolienne flottante et la conception de moteurs électriques. Dans la suite, nous nous appuierons sur le cas de l'éolienne flottante pour illustrer et expliquer chacun des termes du sujet.

Bien que le terme *boîte noire* fasse penser aux enregistreurs de données de vol dans les avions, il désigne ici un modèle numérique décrivant n'importe quel système composé d'entrées et de sorties. Il est représenté par une boîte aux "parois noires" (voir Figure 1) pour signifier que sa particularité est d'avoir un fonctionnement interne inconnu et inaccessible. L'utilisateur peut uniquement fournir des entrées au modèle et récupérer des sorties. Ces modèles se sont démocratisés notamment par l'essor des méthodes d'apprentissage et de l'intelligence artificielle dont les modèles sont souvent vus comme des boîtes noires parce que leur complexité les rend difficilement interprétables. Dans certains cas, les modèles sont aussi volontairement considérés comme des boîtes noires lorsque leur fonctionnement est privé ou doit être protégé. Par exemple, la conception d'une éolienne flottante peut être modélisée par un simulateur numérique, vu comme une boîte noire. Elle prend en entrées des paramètres de conception et des paramètres environnementaux et produit des sorties d'intérêt telles que la production d'énergie, le coût, l'impact environnemental, après un temps de calcul potentiellement très long.

L'utilisation d'une boîte noire modélisant un système ouvre la porte à la recherche de la configuration optimale de paramètres d'entrée, c'est-à-dire celle induisant les meilleures sorties. Cette *optimisation* se traduit par la maximisation ou minimisation de quantités d'intérêt, appelées fonctions objectives. Pour l'éolienne flottante, on peut par exemple chercher à concevoir une structure qui maximise la production d'énergie tout en minimisant le coût et l'impact environnemental. Dans cette thèse, nous ne considérerons qu'une seule fonction objective scalaire à minimiser, notée  $f$ .

En plus de maximiser ou de minimiser des quantités, des sorties doivent également vérifier des conditions, modélisées par des seuils à atteindre ou à ne pas dépasser sur des fonctions de *contraintes*. Par exemple, la vitesse de rotation du rotor de l'éolienne doit être limitée. Dans le cas de plusieurs sorties à optimiser simultanément, il est aussi courant et plus commode d'en désigner certaines comme contraintes. Par exemple, on peut choisir de construire une éolienne avec un impact environnemental limité, plutôt que de chercher à le minimiser. Dans notre cas, nous considérerons  $s$  contraintes scalaires définies par  $g_i \leq 0$  pour  $1 \leq i \leq s$  qu'on notera  $g \leq 0$  lorsque toutes les contraintes sont respectées.

Cette optimisation sous contraintes peut être réalisée avec des méthodes classiques qui nécessitent souvent de nombreuses évaluations de  $f$  et  $g$ . Dans le cas de boîtes noires *coûteuses*, c'est-à-dire



**Figure 1:** Schéma d'une boîte noire où  $x$  et  $U$  sont les entrées de conception et les entrées incertaines et  $f$  et  $g$  sont la fonction objective et les fonctions de contrainte

où une seule évaluation de  $f$  et  $g$  nécessite un temps conséquent, ces approches ne sont pas envisageables. Pour limiter grandement le nombre d'évaluations de  $f$  et  $g$ , des méthodes d'optimisation ont été adaptées donnant notamment naissance à l'*optimisation bayésienne* sur laquelle nous allons nous concentrer dans cette thèse. Elle s'appuie sur un modèle peu coûteux appelé méta-modèle, construit à partir d'un faible nombre d'évaluations de  $f$  et  $g$  en des points déterminés séquentiellement via des critères d'acquisition bien choisis.

Bien que déjà complexe, l'optimisation peut le devenir encore plus en présence d'*incertitudes*. Elles peuvent porter sur les entrées de conception, qui sont notées  $x = (x_1, \dots, x_d)$ , comme l'incertitude de fabrication des dimensions d'un composant. Mais une grande partie de l'aléa provient d'entrées environnementales qui sont intrinsèquement aléatoires souvent liés à des phénomènes météorologiques, tels que la vitesse du vent ou la hauteur de houle dans le cas de l'éolienne flottante. Ces entrées incertaines sont modélisées par un vecteur aléatoire  $U = (U_1, \dots, U_p)$  de loi connue  $\mathbb{P}_U$ . Les sorties héritent alors de l'aléa des  $U$  et il devient nécessaire de définir comment minimiser une fonction objective tout en respectant des contraintes lorsque celles-ci sont aléatoires. Dans cette thèse, on cherchera à minimiser la moyenne de la fonction objective tout en ayant une probabilité de satisfaire les contraintes au dessus d'un seuil  $\alpha$ . Ce problème d'optimisation robuste est défini par la recherche de la configuration d'entrées de conception optimale  $x^*$  définie par

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E}[f(x, U)] \text{ où } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P}[g(x, U) \leq 0] \geq \alpha\}, \quad (1)$$

où  $\mathcal{X} \subset \mathbb{R}^d$  est l'espace des entrées de conception.

Pour gérer ces incertitudes qui complexifient grandement le problème, des méthodes d'optimisation bayésienne ont été adaptées. Elle sont cependant grandement complexifiées par une augmentation de la dimension d'entrée c'est-à-dire du nombre d'entrées. Heureusement, un grand nombre d'entrées implique souvent que certaines ne sont pas essentielles pour résoudre le problème d'optimisation. Par exemple, la couleur de l'éolienne n'est probablement pas le facteur le plus significatif sur la production d'énergie, au contraire de la vitesse du vent ou de la longueur des pâles. Il est alors crucial de comprendre comment chaque entrée affecte les sorties afin de faciliter le processus d'optimisation en se concentrant sur les entrées importantes. Dans un contexte de boîtes noires, cette connaissance des entrées importantes peut être obtenue en mesurant la sensibilité des sorties aux entrées, domaine de l'*analyse de sensibilité*.

Cette thèse vise alors à étudier comment l'analyse de sensibilité peut être utilisée pour faciliter

l'optimisation bayésienne sous contraintes et en présence d'entrées incertaines de boîtes noires coûteuses.

## Travaux existants et objectifs de la thèse

Dans [17] et [157], des méthodes d'analyse de sensibilité permettent de réduire la dimension avant l'optimisation (bayésienne ou non). Dans le cas de l'optimisation bayésienne, les auteurs de [156] proposent une réduction de dimension par analyse de sensibilité à chaque itération de l'optimisation. La grande différence entre ces approches et le sujet de cette thèse est qu'elles ne considèrent pas la présence d'entrées incertaines et ne mesurent que l'influence des entrées de conception sur une optimisation déterministe. L'objectif principal de la thèse est alors d'adapter les approches de [157] et [156] à la présence d'entrées incertaines.

Cette adaptation soulève une question majeure de la thèse qui est celle du rôle des entrées incertaines dans l'optimisation robuste. Quelle influence des entrées incertaines sur le problème d'optimisation robuste une analyse de sensibilité permettrait de mesurer ? Dans cette thèse, nous proposons de mesurer l'effet des  $U$  sur des ensembles d'excursion  $\Gamma$ , qui sont des ensembles d'entrées de conception  $x$  produisant des sorties pertinentes pour l'optimisation, au sens où  $f$  est inférieure à un seuil  $q$  et  $g \leq 0$ . Ces ensembles sont définis par:

$$\Gamma = \{x \in \mathcal{X}, f(x, U) \leq q, g(x, U) \leq 0\},$$

et sont des ensembles aléatoires par le biais des  $U$ . Le second objectif de la thèse est alors de proposer comment réaliser une analyse de sensibilité de ces ensembles d'excursion et plus généralement de n'importe quel modèle dont les sorties sont des ensembles, sans nécessairement faire de lien avec l'optimisation. Malgré sa conception initiale pour des modèles scalaires, l'analyse de sensibilité est une approche qui peut s'adapter à différents types de sorties, comme des vecteurs [87] ou des fonctions [43]. Cependant, à notre connaissance, son utilisation pour des ensembles en sortie n'a pas encore été étudiée.

## Contributions et structure du manuscrit

Pour atteindre ces deux objectifs, le manuscrit commence par un chapitre introduisant les notions utilisées par la suite. Il présente d'abord le domaine de l'analyse de sensibilité puis celui de l'optimisation bayésienne avec et sans entrées incertaines. Les deux domaines sont introduits séparément puis des travaux qui les mixent sont présentés et servent alors de base pour combiner optimisation bayésienne en présence d'incertitudes et analyse de sensibilité. Le manuscrit est ensuite séparé en deux parties, répondant chacun à un des deux objectifs de la thèse.

La première partie vise à répondre à l'objectif de la thèse qui est de quantifier, par une analyse de sensibilité, l'impact des entrées d'un modèle dont les sorties sont des ensembles. Intitulée *Analyse de sensibilité d'ensembles aléatoires*, cette partie est divisée en deux chapitres, chacun étant un article de recherche rédigé au cours de la thèse. Les articles ont été inclus sans modification, ce qui peut entraîner quelques répétitions et des différences de notation. Le premier article, publié dans la revue *Technometrics*, propose une approche basée sur des méthodes à noyaux pour effectuer une analyse de sensibilité des ensembles d'excursion. Le deuxième article, publié dans la revue *Quality and Reliability Engineering*, est motivé par l'étude d'un modèle dont les sorties sont des cartes de concentration de polluants. L'objectif est de proposer des outils pour mesurer l'impact des paramètres d'entrée sur les cartes de sortie qui sont vues comme des ensembles. L'approche présentée dans le premier article, accompagnée de deux autres approches d'analyse de sensibilité d'ensemble aléatoires, sont testées et comparées sur ce modèle.

Dans la deuxième partie de la thèse, l'optimisation bayésienne robuste est l'objectif principal. Cette partie n'a pas encore été soumise à publication, mais a pour vocation de constituer un futur article. Dans cette section, nous nous concentrons sur un algorithme existant d'optimisation bayésienne robuste nommé EFISUR introduit dans [33] qui permet de résoudre efficacement des problèmes d'optimisation robuste sous contraintes. Cependant, EFISUR est sujet à des limitations significatives à mesure que la dimension d'entrée augmente. C'est pourquoi, nous proposons de l'adapter en incorporant des étapes d'analyse de sensibilité pour atténuer les effets de l'augmentation de la dimension. Un premier chapitre est consacré aux développements méthodologiques et présente des algorithmes décrivant différentes façons d'incorporer l'analyse de sensibilité dans EFISUR. Le deuxième chapitre est une étude numérique des algorithmes proposés, testés et analysés sur des fonctions jouets puis appliqués à un cas test réel.

Pour finir, une conclusion synthétise les principales contributions de la thèse et présente des perspectives de recherche.

# Introduction (in English)

## Thesis context

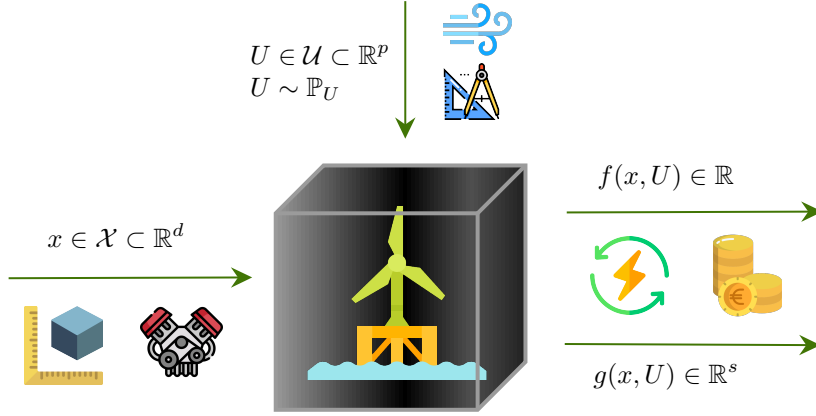
In one sentence, the subject of this thesis is to develop sensitivity analysis methods to facilitate Bayesian optimization of costly black-boxes under constraints and with uncertain inputs. This work is motivated by two main industrial applications: the optimization of a floating wind turbine and the design of electric motors. In the following, we will focus on the floating wind turbine case to illustrate and explain each term of the subject.

While the term *black box* commonly refers to the flight data recorders in airplanes, here it refers to a numerical model that represents any physical system that can be viewed through inputs and outputs. It is represented as a box with "black walls" (see Figure 2) to emphasize its property of having an unknown and inaccessible inner working. The user can then only provide inputs to the model and retrieve outputs. These models have been democratized by the rise of machine learning and artificial intelligence, which have produced numerous models that are considered black boxes because their complexity makes them difficult to interpret. In some cases, models are simply considered black boxes when their inner working needs to be protected. For example, the design of a floating wind turbine can be modeled by a numerical simulator, which is considered a black box. It then takes design and environmental parameters as inputs and, after a potentially very long computation time, produces outputs of interest such as energy production, cost, and environmental impact.

Using a black box to model a system opens the door to searching for the optimal configuration of input parameters, i.e., the one that induces the best outputs. This *optimization* corresponds to the maximization or minimization of quantities of interest, called objective functions. To return to the offshore wind turbine, it is a real industrial challenge to design an optimal structure so that it maximizes the energy production while minimizing the cost and the environmental impact. In this thesis, we will consider only a scalar function to be minimized, denoted  $f$ , called the objective function.

In addition to maximizing or minimizing quantities, some outputs must also satisfy threshold conditions. These *constraints* allow the optimal design to verify essential requirements. For example, the speed of the wind turbine rotor must be limited. In the case of multiple outputs to be optimized simultaneously, it is also common to designate some of them as constraints. For example, one might choose to build a wind turbine with a limited environmental impact, rather than aiming to minimize it. In our case, we will consider  $s$  scalar constraints defined by  $g_i \leq 0$  for  $1 \leq i \leq s$  and denoted by  $g \leq 0$  when all constraints are satisfied.

This constrained optimization can be done by classical methods, but often requires numerous evaluations of  $f$  and  $g$ . However, these are limited in the case of *costly* black boxes, i.e. where a single evaluation of  $f$  and  $g$  takes a considerable amount of time. This costly optimization context



**Figure 2:** Black-box scheme with  $x$  the design inputs,  $U$  the uncertain inputs,  $f$  the objective function and  $g$  the constraint functions

has given rise to optimization methods that attempt to drastically limit the number of evaluations. One such method is *Bayesian optimization*, which we will focus on in this thesis. It uses a surrogate model built from sequentially selected evaluations of  $f$  and  $g$ .

Optimizing a costly black-box model under constraints can become even more complex when potential *uncertainties* arise. For example, design inputs, denoted  $x = (x_1, \dots, x_d)$ , often have an associated uncertainty, such as the manufacturing uncertainty of a component's dimensions. But most uncertainties come from environmental inputs that are inherently random. These can be meteorological parameters, such as wind speed in the case of the wind turbine. Uncertain inputs are modeled by random variables, denoted  $U = (U_1, \dots, U_p)$  of known probability distribution  $\mathbb{P}_U$ , and the outputs inherit of their randomness. It is then necessary to define how to minimize the objective function while respecting the constraints when they both come with uncertainty. In this study, we will aim to minimize the mean of the objective function while ensuring that the probability of satisfying the constraint is greater than a given threshold  $\alpha$ . This robust optimization problem can then be defined as

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E}[f(x, U)] \text{ where } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P}[g(x, U) \leq 0] \geq \alpha\}, \quad (2)$$

where  $\mathcal{X} \subset \mathbb{R}^d$  is the design input space.

Bayesian optimization methods have been adapted to deal with these uncertainties. However, they are very sensitive to increasing the number of inputs which makes optimization more complex, if not impossible in a reasonable time. Fortunately, a large number of inputs often means that some of them are not essential for solving the optimization problem. For example, the color of the wind turbine may not have a significant effect on energy production, unlike wind speed and blade length. It is then critical to understand how each input affects the outputs, as this knowledge can facilitate the optimization process by focusing on the most important inputs. Such knowledge can be obtained by measuring the sensitivity of the outputs to the inputs, a field known as *sensitivity analysis*.

This thesis then aims to study how sensitivity analysis can be used to facilitate expensive black-box Bayesian optimization under constraints and in the presence of uncertain inputs.

## Existing work and thesis objectives

In [17] and [157], sensitivity analysis is used for dimension reduction prior to optimization (Bayesian or not). In the case of Bayesian optimization, the authors of [156] propose dimension reduction by sensitivity analysis at each optimization iteration. The main difference between their approaches and the subject of this thesis is that they do not consider the presence of uncertain inputs and only measure the influence of design inputs on deterministic optimization. The main objective of the thesis is to adapt [157] and [156] approaches in the presence of uncertain inputs.

This adaptation raises a major question of the thesis, namely the role of uncertain inputs in robust optimization. What influence of uncertain inputs on the robust optimization problem would a sensitivity analysis measure? In this thesis, we propose to measure the impact of  $U$  on excursion sets  $\Gamma$ , which are sets of design inputs  $x$  that produce optimization-relevant outputs in the sense that  $f$  is below a threshold  $q$  and  $g \leq 0$ . These sets are defined as

$$\Gamma = \{x \in \mathcal{X}, f(x, U) \leq q, g(x, U) \leq 0\},$$

and are random sets. The second objective of this thesis is to propose how to carry out a sensitivity analysis of these excursion sets, and more generally of any model whose outputs are sets, without necessarily making a link with optimization. Despite its initial conception for scalar models, sensitivity analysis is an approach that can be adapted to different types of outputs, such as vectors [87] or functions [43]. However, to our knowledge, its use for output sets has not yet been studied.

## Contributions and structure of the manuscript

To achieve these two goals, the manuscript begins with a chapter that introduces the concepts used in the rest of the manuscript. It first introduces the fields of sensitivity analysis and of Bayesian optimization. The two fields are introduced separately, and then work is presented that mixes them without uncertainties. It then serves as the basis for combining sensitivity analysis and Bayesian optimization in the presence of uncertainties. The manuscript is then divided into two parts, each corresponding to one of the two objectives of the thesis.

The first part addresses the thesis objective of quantifying, through sensitivity analysis, the impact of inputs to a model whose outputs are sets. Entitled *Sensitivity analysis of set-valued models*, it is divided into two chapters, each corresponding to a research article written during the course of the thesis. The articles have been included without modification, which may result in some repetitions and notation discrepancies, despite our efforts to maintain consistency between the two articles. The first article, published in the journal *Technometrics*, proposes a kernel-based approach to perform a sensitivity analysis of excursion sets. The second paper, published in the journal *Quality and Reliability Engineering*, is driven by a model that produces pollutant concentration maps. The aim is to propose methods to measure the impact of input parameters on output maps, seen as sets. The approach presented in the first article accompanied by two other sensitivity analysis approaches are compared on a toy function and on the model of pollutant concentration maps.

In the second part of the thesis, we return to robust Bayesian optimization. This section has not yet been submitted for publication, but will be part of a future article. In this section, we propose to focus on an existing Bayesian optimization algorithm, called EFISUR, which allows to efficiently solve robust optimization problems under constraints. However, EFISUR is subject to significant limitations as the input dimension increases. Therefore, we propose to adapt it by incorporating sensitivity analysis steps to mitigate the effects of the curse of dimensionality. A first chapter is devoted to methodological developments based on different ways of incorporating

sensitivity analysis into EFISUR. The second chapter is a numerical study of the proposed methods, which are tested and analyzed on toy functions. The algorithms are then tested on a real-life test case.

A final conclusion provides a synthesis of the main contributions of the thesis and presents multiple perspectives on the research topic.

# Contents

|                                                                              |               |
|------------------------------------------------------------------------------|---------------|
| Remerciements                                                                | iii           |
| Introduction (en français)                                                   | v             |
| Contexte de la thèse                                                         | v             |
| Travaux existants et objectifs de la thèse                                   | vii           |
| Contributions et structure du manuscrit                                      | vii           |
| Introduction (in English)                                                    | ix            |
| Thesis context                                                               | ix            |
| Existing work and thesis objectives                                          | xi            |
| Contributions and structure of the manuscript                                | xi            |
| Notations                                                                    | xvii          |
| <b>1 Introduction to Sensitivity Analysis and Bayesian Optimization</b>      | <b>3</b>      |
| <b>1.1 Global Sensitivity Analysis : Sensitivity Indices</b>                 | <b>4</b>      |
| 1.1.1 Sobol' indices                                                         | 5             |
| 1.1.2 Universal indices                                                      | 8             |
| 1.1.3 Hilbert-Schmidt Independence Criterion                                 | 9             |
| <b>1.2 Robust Bayesian Optimization</b>                                      | <b>13</b>     |
| 1.2.1 Gaussian Process Regression                                            | 14            |
| 1.2.1.a Noisy GP                                                             | 15            |
| 1.2.2 Bayesian Optimization                                                  | 16            |
| 1.2.2.a Without constraints                                                  | 16            |
| 1.2.2.b With constraints                                                     | 17            |
| 1.2.3 Robust Bayesian Optimization                                           | 19            |
| 1.2.3.a Without constraints                                                  | 19            |
| 1.2.3.b With constraints: EFISUR                                             | 20            |
| 1.2.4 Bayesian optimization in high dimension                                | 21            |
| <b>1.3 Goal-oriented Sensitivity Analysis for (Bayesian) Optimization</b>    | <b>22</b>     |
| 1.3.1 Goal-Oriented Sensitivity Analysis                                     | 22            |
| 1.3.1.a Target Sensitivity Analysis                                          | 23            |
| 1.3.1.b Conditional Sensitivity Analysis                                     | 23            |
| 1.3.2 Offline and Online Goal-oriented Sensitivity Analysis for Optimization | 24            |
| <br><b>Part I Sensitivity Analysis of set-valued models</b>                  | <br><b>28</b> |
| <b>Introduction</b>                                                          | <b>29</b>     |

|                |                                                                                               |           |
|----------------|-----------------------------------------------------------------------------------------------|-----------|
| <b>2</b>       | <b>Article 1 : Kernel-based Sensitivity Analysis for (Excursion) Sets</b>                     | <b>31</b> |
| 2.1            | Introduction                                                                                  | 33        |
| 2.2            | Kernel-based SA in a generic space $\mathcal{Z}$                                              | 34        |
| 2.2.1          | Kernel Embedding, MMD and HSIC                                                                | 34        |
| 2.2.2          | HSIC-ANOVA indices                                                                            | 36        |
| 2.3            | Kernel-based Sensitivity Analysis for sets                                                    | 38        |
| 2.3.1          | Probabilistic framework of random sets                                                        | 38        |
| 2.3.2          | A kernel between sets                                                                         | 38        |
| 2.3.3          | Estimation of HSIC-ANOVA indices for sets                                                     | 39        |
| 2.4            | Numerical Experiments                                                                         | 41        |
| 2.4.1          | Excursion sets of a toy function                                                              | 41        |
| 2.4.2          | Excursion sets of the optimization of an oscillator                                           | 43        |
| 2.4.3          | Sensitivity analysis for robust conception of an electrical machine                           | 45        |
| 2.5            | Conclusion                                                                                    | 46        |
| <b>3</b>       | <b>Article 2 : Sensitivity Analysis for sets: application to Pollutant concentration maps</b> | <b>51</b> |
| 3.1            | Introduction                                                                                  | 53        |
| 3.2            | Motivating test case                                                                          | 54        |
| 3.3            | New sensitivity analysis indices for sets-valued outputs                                      | 56        |
| 3.3.1          | Sensitivity analysis based on random set theory                                               | 57        |
| 3.3.2          | Universal sensitivity indices                                                                 | 59        |
| 3.3.3          | HSIC-based sensitivity analysis                                                               | 60        |
| 3.4            | Results                                                                                       | 62        |
| 3.4.1          | Vorob'ev based indices                                                                        | 62        |
| 3.4.2          | Universal indices                                                                             | 62        |
| 3.4.3          | HSIC                                                                                          | 65        |
| 3.4.4          | Comparison                                                                                    | 68        |
| 3.5            | Discussion                                                                                    | 69        |
| <b>Part II</b> | <b>Sensitivity Analysis for Robust Bayesian Optimization</b>                                  | <b>74</b> |
|                | <b>Introduction</b>                                                                           | <b>75</b> |
| <b>4</b>       | <b>Sensitivity Analysis and EFISUR : Methodology</b>                                          | <b>77</b> |
| 4.1            | Dimension reduction by Offline Sensitivity Analysis                                           | 78        |
| 4.1.1          | Off-SA : One SA step before EFISUR                                                            | 78        |
| 4.1.1.a        | Method and algorithm                                                                          | 78        |
| 4.1.1.b        | Discussion and perspectives                                                                   | 80        |
| 4.1.2          | Off-seq-SA: Multiple SA steps outside EFISUR iterations                                       | 82        |
| 4.2            | Online Sensitivity Analysis in EFISUR algorithm                                               | 84        |
| 4.2.1          | On-SA: Dimension reduction by SA on the acquisition criteria optimization                     | 84        |
| 4.2.1.a        | Method and algorithm                                                                          | 85        |
| 4.2.1.b        | Discussion and perspectives                                                                   | 85        |
| 4.2.2          | On-2steps-SA: Two-steps optimization of the acquisition functions guided by SA                | 86        |
| <b>5</b>       | <b>Sensitivity Analysis and EFISUR : Numerical study</b>                                      | <b>91</b> |
| 5.1            | Construction of a toy function for robust optimization and EFISUR results                     | 92        |
| 5.1.1          | Toy functions $f_{Glob}^d$ and $g_{Glob}^d$ with global impact far from the optimum           | 93        |
| 5.1.1.a        | Definition of the test case                                                                   | 93        |

|         |                                                                                           |            |
|---------|-------------------------------------------------------------------------------------------|------------|
| 5.1.1.b | Sensitivity Analysis of $f_{Glob}^{10}$ and $g_{Glob}^{10}$                               | 94         |
| 5.1.1.c | Optimization results with EFISUR on the toy functions $f_{Glob}^{10}$ and $g_{Glob}^{10}$ | 96         |
| 5.1.2   | Toy functions $f_{Loc}^d$ and $g_{Loc}^d$ with local influence near the optimum           | 100        |
| 5.1.2.a | Definition of the test case                                                               | 100        |
| 5.1.2.b | Sensitivity Analysis of $f_{Loc}^{10}$ and $g_{Loc}^{10}$                                 | 100        |
| 5.1.2.c | EFISUR on the toy function                                                                | 102        |
| 5.2     | <b>Offline Sensitivity Analysis in EFISUR: numerical results on the toy functions</b>     | <b>102</b> |
| 5.2.1   | Off-SA: One SA step before EFISUR                                                         | 102        |
| 5.2.2   | Off-seq-SA: Multiple SA steps outside EFISUR iterations                                   | 107        |
| 5.2.2.a | Off-seq-SA with two SA steps                                                              | 107        |
| 5.2.2.b | Off-seq-SA with multiple SA steps                                                         | 109        |
| 5.3     | <b>Online Sensitivity Analysis in EFISUR: numerical results on the toy functions</b>      | <b>111</b> |
| 5.3.1   | On-SA: Dimension reduction by SA on the acquisition functions                             | 111        |
| 5.3.2   | On-2steps-SA: Two-steps dimension reduction by SA on the acquisition functions            | 115        |
| 5.3.2.a | On-2steps-SA                                                                              | 115        |
| 5.3.2.b | On-2steps-SA-free                                                                         | 115        |
| 5.4     | <b>Numerical comparison of the algorithms on the toy functions</b>                        | <b>118</b> |
| 5.5     | <b>Application: Optimization of an electric machine</b>                                   | <b>119</b> |
| 5.5.1   | Validation                                                                                | 121        |
| 6       | <b>Conclusion and perspectives</b>                                                        | <b>127</b> |
| 6.1     | <b>Summary of the main contributions</b>                                                  | <b>127</b> |
| 6.2     | <b>Summary of the perspectives</b>                                                        | <b>128</b> |
| 6.2.1   | Part 1: SA of set-valueds models                                                          | 128        |
| 6.2.2   | Part 2: SA and EFISUR                                                                     | 129        |
|         | <b>Appendices</b>                                                                         | <b>131</b> |
| A       | <b>Appendix of chapter 1: Some elements of random set theory</b>                          | <b>131</b> |
| A.1     | <b>A bit of random set theory</b>                                                         | <b>131</b> |
| A.1.1   | Random set definition                                                                     | 131        |
| A.1.2   | Vorob'ev expectation and deviation                                                        | 132        |
| A.2     | <b>A bit more of random set theory</b>                                                    | <b>135</b> |
| A.2.1   | Alternative expectations of random closed sets                                            | 135        |
| A.2.1.a | Debreu Expectation                                                                        | 135        |
| A.2.1.b | Selection expectation                                                                     | 136        |
| B       | <b>Appendix of chapter 2: Proofs and additional plots</b>                                 | <b>139</b> |
| B.1     | <b>Proofs</b>                                                                             | <b>139</b> |
| B.1.1   | Proof of Lemma 2.1                                                                        | 139        |
| B.1.2   | Proof of Proposition 2.1                                                                  | 140        |
| B.1.3   | Proof of Proposition 2.2                                                                  | 140        |
| B.1.4   | Proof of Proposition 2.3                                                                  | 141        |
| B.1.5   | Proof of Proposition 2.4                                                                  | 142        |
| B.1.6   | Convergence rate in the case of Independent $m$ sample                                    | 143        |
| B.2     | <b>Additional plots</b>                                                                   | <b>144</b> |
| C       | <b>Appendix of chapter 3: Alternative Vorob'ev-based indices</b>                          | <b>149</b> |

|     |                                                               |     |
|-----|---------------------------------------------------------------|-----|
| D   | Appendix of chapter 4: On-seq-SA algorithm                    | 151 |
| E   | Appendix of chapter 5: Additional numerical results           | 155 |
| E.1 | Additional Sensitivity Analysis plots                         | 155 |
| E.2 | Complete screening tables of Off-seq-SA                       | 158 |
| E.3 | Comparison of the algorithms for different budget allocations | 160 |

## Essential notations

### Mathematical notations

|                                     |                                                                    |
|-------------------------------------|--------------------------------------------------------------------|
| $f$                                 | Objective function                                                 |
| $g = (g_i)_{i=1}^s$                 | Constraints functions                                              |
| $x$                                 | Design input vector                                                |
| $x_i$                               | $i$ -th component of $x$                                           |
| $x^k$                               | Vector corresponding to $k$ -th value of $x$                       |
| $\mathcal{X}$                       | Design input space                                                 |
| $d$                                 | Design input space dimension                                       |
| $(\Omega, \mathcal{F}, \mathbb{P})$ | Generic probability space                                          |
| $\omega$                            | Element of $\Omega$                                                |
| $U$                                 | Uncertain input random vector                                      |
| $\mathcal{U}$                       | Uncertain input space                                              |
| $p$                                 | Uncertain input space dimension                                    |
| $\mathbb{P}_U$                      | Probability distribution of a random vector $U$                    |
| $U_i \perp U_j$                     | $U_i$ and $U_j$ are independent                                    |
| $\Gamma$                            | Random excursion set                                               |
| $\mathbb{1}_E$                      | Indicator function of a set $E$                                    |
| $F, G$                              | Gaussian Processes modeling $f$ and $g$                            |
| $\mu_F, \mu_G$                      | Predicted mean of $F$ and $G$                                      |
| $X$                                 | Random design input                                                |
| $A, B$                              | Subsets of $\{1, \dots, d\}$ (or of $\{1, \dots, p\}$ )            |
| $\mathcal{M}$                       | Generic model                                                      |
| $\mathcal{Z}$                       | Generic measurable space                                           |
| $\mathcal{M}_+^1(\mathcal{Z})$      | Space of Borel Probability measures of $\mathcal{Z}$               |
| $k, k_{\mathcal{Z}}$                | Generic kernel, kernel defined on $\mathcal{Z} \times \mathcal{Z}$ |
| $\mathcal{H}_k$                     | RKHS of reproducing kernel $k$                                     |
| $\mu_k$                             | Kernel mean embedding of kernel $k$                                |
| $\lambda$                           | Lebesgue measure                                                   |
| $\mathcal{U}(\mathcal{X})$          | Uniform law on $\mathcal{X}$                                       |
| $\#A$ or $ A $                      | Cardinality of $A$                                                 |

**Abbreviations**

|       |                                        |
|-------|----------------------------------------|
| SA    | Sensitivity Analysis                   |
| GSA   | Global Sensitivity Analysis            |
| GOSA  | Goal-oriented Sensitivity Analysis     |
| TSA   | Target Sensitivity Analysis            |
| CSA   | Conditional Sensitivity Analysis       |
| GP    | Gaussian Process                       |
| BO    | Bayesian Optimization                  |
| EI    | Expected Improvement                   |
| EFI   | Expected Feasible Improvement          |
| EGO   | Efficient Global Optimization          |
| DoE   | Design of Experiment                   |
| ANOVA | ANalysis Of VAriance                   |
| HSIC  | Hilbert-Schmidt Independence Criterion |
| MMD   | Maximum-Mean Discrepancy               |
| RKHS  | Reproducing Kernel Hilbert Space       |
| VMD   | Vorob'ev Median Deviation              |
| Off   | Offline                                |
| On    | Online                                 |





# 1

## Introduction to Sensitivity Analysis and Bayesian Optimization

---

|            |                                                                        |           |
|------------|------------------------------------------------------------------------|-----------|
| <b>1.1</b> | <b>Global Sensitivity Analysis : Sensitivity Indices</b>               | <b>4</b>  |
| 1.1.1      | Sobol' indices                                                         | 5         |
| 1.1.2      | Universal indices                                                      | 8         |
| 1.1.3      | Hilbert-Schmidt Independence Criterion                                 | 9         |
| <b>1.2</b> | <b>Robust Bayesian Optimization</b>                                    | <b>13</b> |
| 1.2.1      | Gaussian Process Regression                                            | 14        |
| 1.2.1.a    | Noisy GP                                                               | 15        |
| 1.2.2      | Bayesian Optimization                                                  | 16        |
| 1.2.2.a    | Without constraints                                                    | 16        |
| 1.2.2.b    | With constraints                                                       | 17        |
| 1.2.3      | Robust Bayesian Optimization                                           | 19        |
| 1.2.3.a    | Without constraints                                                    | 19        |
| 1.2.3.b    | With constraints: EFISUR                                               | 20        |
| 1.2.4      | Bayesian optimization in high dimension                                | 21        |
| <b>1.3</b> | <b>Goal-oriented Sensitivity Analysis for (Bayesian) Optimization</b>  | <b>22</b> |
| 1.3.1      | Goal-Oriented Sensitivity Analysis                                     | 22        |
| 1.3.1.a    | Target Sensitivity Analysis                                            | 23        |
| 1.3.1.b    | Conditional Sensitivity Analysis                                       | 23        |
| 1.3.2      | Offline and Online Goal-oriented Sensitivity Analysis for Optimization | 24        |

---

This first chapter is an introduction presenting the fields and the essential notions necessary for the rest of the thesis. The main goal of the thesis is to study how Sensitivity Analysis (SA) can serve robust Bayesian Optimization (BO). Therefore, in a first part, the field of sensitivity analysis is introduced. Then, in a second part, Bayesian optimization is presented with and without uncertain inputs. Finally, a third part presents studies that mix the two.

For the first part of the thesis, which develops methods to perform SA on sets, the reader can only read Section 1.1 for details on the SA tools used. Additional notions on random sets are presented in the appendix and may also help to understand the first part, but are not strictly necessary, especially since they are introduced in the corresponding chapters.

Then, the other two sections (1.2 and 1.3), together with the tools of Section 1.1, are necessary for the second part of the thesis, which mixes robust Bayesian Optimization and Sensitivity Analysis.

## 1.1 Global Sensitivity Analysis : Sensitivity Indices

In [134], sensitivity analysis (SA) is defined as the study to determine how possible changes or errors in parameter values affect the output of the model. Given an input-output model  $\mathcal{M}$  defined by :

$$\mathcal{M} : \begin{cases} \mathcal{X} & \longrightarrow \mathbb{R} \\ x = (x_1, \dots, x_d) & \longmapsto \mathcal{M}(x_1, \dots, x_d) \end{cases} ,$$

SA aims to analyze the sensitivity of output to input variations. In more recent work [144, 26], the terms *changes* and *errors* are replaced by *uncertainty* to define SA as the study of how the uncertainty in a model's output can be apportioned to its inputs. This emphasizes the specificity of SA for modeling input variation through random variables:

$$X = (X_1, \dots, X_d) \sim \mathbb{P}_X,$$

which uncertainty is propagated through  $\mathcal{M}$  to the random output  $Y$ :

$$Y = \mathcal{M}(X).$$

As described in [16], a distinction must be made between *local* and *global* SA. *Local* SA measures the impact of input variations around a specific value of the inputs (see [14] for a recent review that includes local SA). On the other hand, *Global Sensitivity Analysis* (GSA) measures how the output is affected when the inputs vary across their entire domain (see [143] for an exhaustive presentation of GSA methods, [70] for a review and [26] for an overview of recent GSA methods along with practical use in R).

In [26], the authors classify GSA methods into three approaches:

- *Screening Methods* explore the output behavior on the entire input space through its coarse discretization. Such methods require a reasonable amount of evaluation of the model, but give only qualitative information about the influence of the inputs. They can be used for *screening*, i.e., to select important inputs and remove negligible ones. The best known are the Morris methods [109], which consist in discretizing the input space for each variable and then performing a given number of One At a Time designs, i.e., modifying one input while the others are fixed.
- *Importance Measures* or *Sensitivity Indices* are scalars estimated on a sample of inputs and outputs and allow to measure quantitatively the influence of the inputs on the output. They can be used in both *screening* and *ranking*, i.e., to rank the inputs according to their influence.
- *Deep exploration of sensitivities* goes beyond quantitative value and aims to understand the inputs influence through plots or models. For example, Cobweb (or parallel) plots [83] allow to visualize inputs influence and interactions.

In this thesis we focus on sensitivity indices. In the next parts we describe three of them in detail. We start with the most well-known and used sensitivity indices: Sobol' indices.

### 1.1.1 Sobol' indices

Sobol' indices (originally introduced under the name of *sensitivities* in [152]) are variance-based sensitivity indices that can be defined in two different ways. For the sake of understanding, we explain both.

**SOBOL' INDICES AS PART OF OUTPUT VARIANCE** The first one directly defines Sobol' indices as the part of the output variance for which an input (or group of inputs) is responsible.

**Definition 1.1.** For  $i \in \{1, \dots, d\}$ , first-order Sobol' indices are defined as

$$S_i = \frac{\text{Var}[\mathbb{E}[Y|X_i]]}{\text{Var}[Y]}. \quad (1.1)$$

Since  $\mathbb{E}[Y|X_i]$  is by definition the random variable, function only of  $X_i$ , that best describes  $Y$  (with  $L_2$  distance),  $S_i$  actually measures the portion of the output variance explained by  $X_i$  alone.

To define higher order indices, but also for the rest of the manuscript, we denote by  $A$  (or  $B$ ) subsets of  $\{1, \dots, d\}$  and associated vectors of components in  $A$  by  $X_A = (X_i)_{i \in A}$ , where for only two components  $X_{i,j} = X_{\{i,j\}}$ . Used later, we also denote  $-A = \{i \in \{1, \dots, d\}, i \notin A\}$  such that  $X_{-A}$  is the vector of all the component not in  $A$ .

Second order indices  $S_{i,j}$ , which measure the effect of interactions between two inputs  $X_i$  and  $X_j$ , are then defined as

$$S_{i,j} = \frac{\text{Var}[\mathbb{E}[Y|X_{i,j}]] - \text{Var}[\mathbb{E}[Y|X_i]] - \text{Var}[\mathbb{E}[Y|X_j]]}{\text{Var}[Y]},$$

where the effect of each input alone, contained in the first-order indices, are removed from  $\text{Var}[\mathbb{E}[Y|X_{i,j}]]$ . We can derive the indices of any order  $|A|$ , where  $|A|$  denotes the cardinality of  $A$ . A distinction must be made with Sobol' indices of a group of inputs, called closed Sobol' indices.

**Definition 1.2.** Let  $A \subset \{1, \dots, d\}$ ,

- Sobol' indices  $S_A$  of order  $|A|$  are defined as

$$S_A = \frac{\sum_{B \subset A} (-1)^{|A|-|B|} \text{Var}(\mathbb{E}[Y | X_B])}{\text{Var}(Y)}.$$

- Closed Sobol' indices  $S_A^{clos}$  of a group of inputs  $A$  are defined as

$$S_A^{clos} = \sum_{B \subset A} S_B = \frac{\text{Var}[\mathbb{E}[Y|X_A]]}{\text{Var}[Y]}.$$

$S_A$  measures the effect of the interaction of order  $|A|$  of the inputs  $X_A$ , while  $S_A^{clos}$  measures the effect of a group of inputs together, counting both individual contributions and any order interactions between members of this group.

Finally, to measure the effect of an input, alone and through its interactions, *total-order* indices [67] are defined by summing each index where the input appears:

**Definition 1.3.** For  $i \in \{1, \dots, d\}$ , total-Sobol' indices are defined by

$$S_i^T = \sum_{\substack{A \subset \{1, \dots, d\} \\ i \in A}} S_A.$$

To speak of "part of the output variance", all order indices must sum to one. Luckily this is the case under *independence* of the inputs.

**Proposition 1.1.** *If  $X_1, \dots, X_d$  are mutually independent,*

$$\sum_{A \subset \{1, \dots, d\}} S_A = 1.$$

This result is based on variance decomposition, also called ANOVA decomposition, or Hoeffding variance decomposition, or Hoeffding-ANOVA decomposition. It is shown relying on the Hoeffding decomposition [66].

**SOBOL' INDICES FROM A FUNCTIONAL DECOMPOSITION** Alternatively, Sobol' indices can be obtained from a functional decomposition, as it was first introduced by Sobol' in [152] (and from which they get their name). He starts with a functional decomposition of  $\mathcal{M}$  as the sum of the contributions of each of the inputs:

$$\begin{aligned} \mathcal{M}(x) &= \sum_{A \subset \{1, \dots, d\}} \mathcal{M}_A(x_A) \\ &= \mathcal{M}_\emptyset + \sum_{i=1}^d \mathcal{M}_i(x_i) + \sum_{i < j}^d \mathcal{M}_{i,j}(x_i, x_j) + \dots + \mathcal{M}_{1, \dots, d}(x_1, \dots, x_d), \end{aligned} \quad (1.2)$$

shown to be unique under the orthogonality condition that,

$$\forall A \subset \{1, \dots, d\}, \quad \forall i \in A, \quad \int_0^1 \mathcal{M}_A(x_A) dx_i = 0. \quad (1.3)$$

The indices (written here for one input only) can then be derived:

$$S_i = \frac{\int_0^1 \mathcal{M}_i^2(x_i) dx_i}{\int \mathcal{M}^2(x) dx - \mathcal{M}_\emptyset^2}.$$

We retrieve the Sobol' indices by replacing  $x$  in (1.2) with the corresponding random vector that is marginally uniformly distributed in  $[0, 1]$  to obtain a decomposition of  $Y$ :

$$Y = \mathcal{M}(X) = \mathcal{M}_\emptyset + \sum_{i=1}^d \mathcal{M}_i(X_i) + \dots + \mathcal{M}_{1, \dots, d}(X_1, \dots, X_d), \quad (1.4)$$

where (1.3) implies that:

$$\begin{aligned} - \mathcal{M}_\emptyset &= \mathbb{E}[Y], \\ - \mathcal{M}_i(X_i) &= \mathbb{E}[Y | X_i] - \mathbb{E}[Y], \\ - \mathcal{M}_{i,j}(X_i, X_j) &= \mathbb{E}[Y | X_i, X_j] - \mathbb{E}[Y | X_i] - \mathbb{E}[Y | X_j] + \mathbb{E}[Y], \\ &\dots, \\ - \mathcal{M}_A(X_A) &= \mathbb{E}[Y | X_A] - \sum_{B \subset A} \mathcal{M}_B(X_B) = \mathbb{E}[Y | X_A] - \sum_{B \subset A} (-1)^{|A|-|B|} \mathbb{E}[Y | X_B]. \end{aligned}$$

Adding the independence assumption and taking the variance of (1.4), we obtain the variance decomposition leading to Sobol' indices. If the  $X_i$  are not uniformly distributed in  $[0, 1]$ , simply

applying the same steps to the uniformly distributed inputs  $U_i = F_{X_i}(X_i)$ , where  $F_{X_i}$  is the cumulative distribution function of  $X_i$ , leads to the variance decomposition for any independent inputs.

Sobol' indices are clearly adapted to ranking by sorting the inputs according to their first or total index. Screening can also be done, for example, by setting a threshold above which the inputs are considered influential. However, except in very rare cases of analytically known Sobol' indices, they have to be estimated with input-output samples of the model.

**PICK-FREEZE ESTIMATION** As written in Equation 1.1, estimating Sobol' indices requires estimating the variance of a conditional expectation, which can be done with a double-loop estimation using the classical Monte Carlo method, but which implies a important number of model evaluations. Already in [152], the double loop is bypassed by *freezing* inputs. This *pick-freeze* approach relies on the result of [153]:

**Lemma 1.1.** *Given  $Y = f(X)$ , with  $X \sim \mathbb{P}_X$ , and  $Y_A = f(X_A, X'_{-A})$  with  $A \subset \{1, \dots, d\}$ ,  $X' \sim \mathbb{P}_X$  and  $X' \perp X$ , we have:*

$$\text{Var}(\mathbb{E}(Y \mid X_A)) = \text{Cov}(Y, Y_A).$$

This allows the variance of the conditional expectation estimate to be replaced by an estimate of a covariance in which the inputs of interest are frozen to compute  $Y_A$ . Then the first-order index  $S_i$  is estimated by:

$$S_i^{N,PF} = \frac{\frac{1}{N} \sum_k Y^k Y_i^k - \left(\frac{1}{N} \sum_k Y^k\right) \left(\frac{1}{N} \sum_k Y_i^k\right)}{\frac{1}{N} \sum_k (Y^k)^2 - \left(\frac{1}{N} \sum_k Y^k\right)^2},$$

where  $Y^k = f(X^k)$  and  $Y_i^k = f(X_i^k, X'^k_{-i})$ , with  $(X^1, \dots, X^N)$  and  $(X'^1, \dots, X'^N)$  independent i.i.d.  $N$ -samples of  $X$  [67]. This estimator converges with rate  $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$  as shown in [73] along with good statistical properties of the estimator. Alternative expressions are proposed in [119, 73, 144]. A weakness of the pick-freeze approach is that it requires  $N$  evaluations of  $f$  to compute  $Y^k = f(X^k)$ , plus  $N$  additional evaluations for each input  $X_i$  to compute  $Y_i^k = f(X_i^k, X'^k_{-i})$ . This leads to a total of  $N \times (d+1)$  evaluations, which grows (linearly) with the dimension. In addition, to construct such a design of experiments, it is necessary to be able to choose the specific points of the simulation, which is not always possible. For these reasons, much work has been done to propose better estimates of Sobol' indices. We focus on a recent rank-based method proposed in [42], which allows to estimate first-order Sobol' indices with a single  $N$ -sample of  $(X, Y)$ .

**RANK-BASED ESTIMATION** Using a correlation coefficient proposed in [18], the authors of [42] define rank-based estimation of sensitivity indices. The general idea is to sort the sample of pairs  $((X^1, Y^1), \dots, (X^N, Y^N))$  according to the order obtained by ranking the  $i$ -th component  $X_i$  and derive an estimate of  $S_i$ . In practice, the rank of each input is defined by a permutation  $\sigma_i$  such that

$$X_i^{\sigma_i^{-1}(1)} \leq \dots \leq X_i^{\sigma_i^{-1}(N)},$$

where  $\sigma_i(k)$  is the rank of  $X_i^k$ . Then, the covariance term is estimated using only the pair of neighbor samples, resulting in rank-based first-order Sobol' indices:

$$S_i^{N,rank} = \frac{\frac{1}{n} \sum_{j=1}^{n-1} Y^{\sigma_i^{-1}(j)} Y^{\sigma_i^{-1}(j+1)} + Y^{\sigma_i^{-1}(N)} Y^{\sigma_i^{-1}(1)} - \left(\frac{1}{n} \sum_{j=1}^n Y^j\right)^2}{\frac{1}{n} \sum_{j=1}^n (Y^j)^2 - \left(\frac{1}{n} \sum_{j=1}^n Y^j\right)^2}.$$

With this approach, a single  $N$ -sample  $((X^1, Y^1), \dots, (X^N, Y^N))$  allows to estimate all first-order indices (but only first-order) with the simple effort of sorting the output according to the induce rank of each input. Asymptotic properties are studied in [42], notably providing almost certain convergence and central limit theorem.

**SOBOL' INDICES WITH COMPLEX OUTPUTS** So far, Sobol' indices are defined only for scalar output, i.e., where  $f(x) \in \mathbb{R}$ . However, it has been adapted for other types of output. For multivariate outputs, i.e., when  $f(x) \in \mathbb{R}^s$ , [87] introduces Sobol' indices for **vectorial outputs** in an empirical way and [43] properly defines them and studies properties of their estimators. In the end, *aggregated* Sobol' indices correspond to a weighted sum of the Sobol' indices of each output component, where the weight corresponds to the proportion of the output variance of that component. Given  $Y = (Y_1, \dots, Y_s) = f(X)$  and corresponding Sobol' indices  $(S_{i1}, \dots, S_{is})$ , aggregated Sobol' indices are defined as

$$GS_i = \frac{\sum_{j=1}^s \text{Var}(Y_j) S_{ij}}{\sum_{j=1}^s \text{Var}(Y_j)}.$$

In [43], they are extended to **functional outputs**, where their estimation is similar to the case of vectorial outputs when truncating an expansion of the output norm. In [59, 71, 98], the case of stochastic computer code outputs is studied. For set-valued models, adaptations of Sobol' indices have not been studied and would require equivalent for expectation and variance but of random sets.

### 1.1.2 Universal indices

To deal with any type of output  $Y \in \mathcal{Y}$  of probability distribution  $\mathbb{P}_Y$ , where  $\mathcal{Y}$  is a metric space, and to generalize existing sensitivity analysis, the authors of [37] introduces *universal* indices, generalizing the *general metric space* sensitivity indices of [45].

**Definition 1.4.** Let  $\mathcal{A}$  be a measurable space endowed with the probability measure  $\mathbb{Q}$ , and let the test functions  $T$  be defined as

$$\begin{aligned} \mathcal{A} \times \mathcal{Y} &\rightarrow \mathbb{R} \\ T : a, y &\mapsto T_a(y). \end{aligned}$$

Assuming that  $T$  is  $L^2$  with respect to  $\mathbb{Q} \otimes \mathbb{P}_Y$ , the universal indices  $S_i^{\text{Univ}}(T_a, \mathbb{Q})$  are defined for  $i \in \{1, \dots, d\}$  as

$$S_i^{\text{Univ}} = \frac{\int_{\mathcal{A}} \text{Var}(\mathbb{E}[T_a(Y) | X_i]) d\mathbb{Q}(a)}{\int_{\mathcal{A}} \text{Var}(T_a(Y)) d\mathbb{Q}(a)}.$$

These indices consist in measuring the influence of  $X_i$  on an output transformations  $T_a(Y)$  through  $\text{Var}(\mathbb{E}[T_a(Y) | X_i])$ . Then these effects are averaged over all possible transformations (taking into account their "occurrence" through  $\mathbb{Q}$ ). Like Sobol' indices, *universal* indices sum to 1 and are estimated using similar methods such as pick-freeze or rank-based methods as given in [37].

The use of test functions allows to consider any kind of output. In [37], they are used to handle computer codes valued in general Wasserstein spaces. Their use for set-valued models has not yet been explored, but would be possible if test functions are defined on a space of sets.

They are called universal in the sense that they generalize many indices starting from Sobol' indices with  $T_a(x) = x$  and any  $\mathbb{Q}$  as the sum is one. The universality of these indices comes from the liberty in the choices of  $(\mathcal{A}, \mathbb{Q}, T)$ . However, some choices may lead to meaningless indices that don't measure any interpretable quantity. In [37], it is explained that these choices should be made so that  $(T_a(Y))_{a \in \mathcal{A}}$  characterize enough the distribution of  $Y$ . For example, for Sobol' indices,

only the effect of the inputs on the second-order moment is measured, which can sometimes be a poor representation of the output variability. Other types of indices, called moment-independent sensitivity indices, have been especially developed to measure input impacts through the entire probability distributions (both of the inputs and outputs). In the next part, we introduce one of them: the Hilbert-Schmidt Independence Criterion (HSIC).

### 1.1.3 Hilbert-Schmidt Independence Criterion

The search for moment-independent measures may have started with distribution-based indices of [122], where an index based on Kullback-Leiber divergence is proposed. Then [21] proposes an index based on cumulative distribution functions (cdf), to which the popular Borgonovo indices (based on probability distribution functions) are compared in [13]. Also based on cdf, [44] proposes Cramer von Mises indices and [128] proposes indices based on Kolmogorov-Smirnov statistics.

The previous indices can often be written as measures of dissimilarity between  $Y$  and  $Y|X_i$  to define an index  $S_i^d$ :

$$S_i^d = \mathbb{E}(d(\mathbb{P}_Y, \mathbb{P}_{Y|X_i})),$$

where  $d$  is the dissimilarity measure, i.e., a function that quantifies how different two mathematical objects are. Often  $d$  varies from 0 for similar objects to  $+\infty$  for very different objects. If  $\mathbb{P}_Y$  and  $\mathbb{P}_{Y|X_i}$  are close, it means that  $X_i$  has a small effect on the output variability. There are many types of dissimilarity measures. Among them, the F-divergence dissimilarity [23] generalizes many of the previously mentioned indices as explained in chapter 6 of [26].

An (often cheaper) alternative to F-divergence is the use of kernel-based dissimilarity, which is a special case of integral probability metric dissimilarity. This has led to the Maximum Mean Discrepancy (MMD)-based indices proposed in [25]. Also based on MMD and kernels, [24] propose to use the Hilbert-Schmidt Independence Criterion (HSIC), defined in [51], as a sensitivity index. Instead of measuring a dissimilarity between  $Y$  and  $Y|X_i$ , they rely directly on measuring the dependence between  $Y$  and  $X_i$ . If  $X_i$  and  $Y$  are strongly dependent, it is interpreted that  $X_i$  strongly affects  $Y$ .

To properly introduce HSIC, some details about kernels are needed. We also give them in Section 2.2 of Chapter 2, corresponding to [35].

A BIT OF RKHS THEORY First, let us define a kernel.

**Definition 1.5.** Let  $\mathcal{Z}$  be any measurable space. A function  $k_{\mathcal{Z}}$  defined as

$$k_{\mathcal{Z}} : \begin{cases} \mathcal{Z} \times \mathcal{Z} & \longrightarrow \mathbb{R} \\ z, z' & \longmapsto k_{\mathcal{Z}}(z, z') \end{cases} ,$$

is a kernel if and only if it is symmetric and positive definite.

Then, based on Aronszajn's theorem [6], Theorem 4.21 of [161] states the existence of a unique Reproducing Kernel Hilbert Space (RKHS)  $\mathcal{H}_{k_{\mathcal{Z}}}$  of reproducing kernel  $k_{\mathcal{Z}}$ , i.e., an Hilbert space such that

- $\forall z \in \mathcal{Z}, k_{\mathcal{Z}}(\cdot, z) \in \mathcal{H}_{k_{\mathcal{Z}}}$ ,
- $\forall z \in \mathcal{Z}, \forall f \in \mathcal{H}_{k_{\mathcal{Z}}}, \langle f, k_{\mathcal{Z}}(\cdot, z) \rangle_{\mathcal{H}_{k_{\mathcal{Z}}}} = f(z)$  (reproducing property).

$\mathcal{H}_{k_{\mathcal{Z}}}$  is then a space of functions from  $\mathcal{Z}$  to  $\mathbb{R}$  in which the probability distribution on  $\mathcal{Z}$  are embedded by the kernel mean embedding defined (assuming a sufficient condition that the kernel is bounded and measurable [111]):

**Definition 1.6.** The kernel mean embedding of  $\mathcal{M}_+^1(\mathcal{Z})$  in  $\mathcal{H}_{k_{\mathcal{Z}}}$  is defined as

$$\begin{aligned} \mathcal{M}_+^1(\mathcal{Z}) &\rightarrow \mathcal{H}_{k_{\mathcal{Z}}} \\ \mu_{k_{\mathcal{Z}}} : \mathbb{P} &\mapsto \mu_{k_{\mathcal{Z}}}(\mathbb{P}) = \int_{\mathcal{Z}} k_{\mathcal{Z}}(z, \cdot) d\mathbb{P}(z), \end{aligned} \quad (1.5)$$

where  $\mathcal{M}_+^1(\mathcal{Z})$  is the space of probability measures on  $\mathcal{Z}$ .

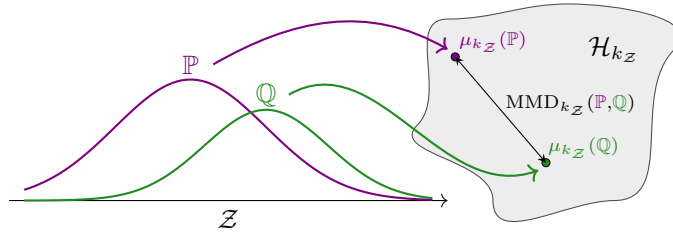
Then, instead of comparing probability distributions in  $\mathcal{M}_+^1(\mathcal{Z})$ , the distance between their respective mean embedding is measured in  $\mathcal{H}_{k_{\mathcal{Z}}}$  defining the MMD (see Figure 1.1). Using the reproducing property, an expression of the MMD using the kernels is given by:

$$\begin{aligned} \text{MMD}_{k_{\mathcal{Z}}}^2(\mathbb{P}, \mathbb{Q}) &= \|\mu_{k_{\mathcal{Z}}}(\mathbb{P}) - \mu_{k_{\mathcal{Z}}}(\mathbb{Q})\|_{\mathcal{H}_{k_{\mathcal{Z}}}}^2 \\ &= \mathbb{E}_{Z_1, Z_1' \sim \mathbb{P}} [k_{\mathcal{Z}}(Z_1, Z_1')] + \mathbb{E}_{Z_2, Z_2' \sim \mathbb{Q}} [k_{\mathcal{Z}}(Z_2, Z_2')] \\ &\quad - \mathbb{E}_{Z_1 \sim \mathbb{P}, Z_2 \sim \mathbb{Q}} [k_{\mathcal{Z}}(Z_1, Z_2)], \end{aligned} \quad (1.6)$$

where  $\|\cdot\|_{\mathcal{H}_{k_{\mathcal{Z}}}}$  is the Hilbertian norm. When  $k_{\mathcal{Z}}$  is *characteristic*, defined by the injectivity of (1.5), the MMD becomes a distance, i.e.,

$$\text{MMD}_{k_{\mathcal{Z}}}(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q},$$

and allows to efficiently compare probability distributions.



**Figure 1.1:** Kernel mean embedding

**HILBERT SCHMIDT INDEPENDENCE CRITERION** Relying on the MMD, the Hilbert-Schmidt Independence Criterion (HSIC) allows to measure dependence between two random variables.

**Definition 1.7.** For  $i \in \{1, \dots, d\}$ ,  $k_{\mathcal{X}_i}$  and  $k_{\mathcal{Y}}$  kernels on  $\mathcal{X}_i$  and  $\mathcal{Y}$ , and  $k = k_{\mathcal{X}_i} \otimes k_{\mathcal{Y}}$  the product kernel inducing a RKHS  $\mathcal{H}_k$ , the Hilbert-Schmidt Independence Criterion  $\text{HSIC}_k(X_i, Y)$  between two random variables  $X_i$  and  $Y$  of joint probability distribution  $\mathbb{P}_{X_i, Y}$  and marginals  $\mathbb{P}_{X_i}$  and  $\mathbb{P}_Y$  is defined as

$$\begin{aligned} \text{HSIC}_k(X_i, Y) &= \text{MMD}_k^2(\mathbb{P}_{X_i, Y}, \mathbb{P}_{X_i} \otimes \mathbb{P}_Y) \\ &= \|\mu_k(\mathbb{P}_{X_i, Y}) - \mu_k(\mathbb{P}_{X_i} \otimes \mathbb{P}_Y)\|_{\mathcal{H}_k}^2. \end{aligned}$$

It can be written using (1.6) as a sum of kernel expectations:

$$\begin{aligned} \text{HSIC}_k(X_i, Y) &= \mathbb{E}[k_{\mathcal{X}_i}(X_i, X_i') k_{\mathcal{Y}}(Y, Y')] \\ &\quad + \mathbb{E}[k_{\mathcal{X}_i}(X_i, X_i')] \times \mathbb{E}[k_{\mathcal{Y}}(Y, Y')] \\ &\quad - 2\mathbb{E}[\mathbb{E}[k_{\mathcal{X}_i}(X_i, X_i') | X_i] \mathbb{E}[k_{\mathcal{Y}}(Y, Y') | Y]]. \end{aligned} \quad (1.7)$$

where  $Y' = f(X')$  with  $X'$  an independent copy of  $X$ .

Using both characteristic kernels for  $k_{\mathcal{X}_i}$  and  $k_Y$  is sufficient to have:

$$\text{HSIC}_k(X_i, Y) = 0 \Leftrightarrow X_i \perp Y.$$

From this, a statistical test of independence is derived [52],

$$(\mathcal{H}_0^i) : \text{HSIC}_k(X_i, Y) = 0 \quad \text{vs} \quad (\mathcal{H}_1^i) : \text{HSIC}(X_i, Y) > 0,$$

making HSIC ideal for screening.

Given a single  $N$ -sample  $((X^1, Y^1), \dots, (X^N, Y^N))$ , the estimation of HSIC is derived from (1.7) in terms of V-statistic [51]:

$$\begin{aligned} \widehat{\text{HSIC}}_v(X_i, Y) &= \frac{1}{N^2} \sum_{s,j} K_{sj} L_{sj} + \frac{1}{N^4} \sum_{s,j,q,r} K_{sj} L_{q,r} - \frac{2}{N^3} \sum_{s,j,q} K_{sj} L_{j,q} \\ &= \frac{1}{N^2} \text{tr}(KHLH), \end{aligned} \quad (1.8)$$

where  $K$  and  $L$  are the Gram matrices associated to the kernels  $k_{\mathcal{X}_i}$  and  $k_Y$  defined for all  $s, j \in \{1, \dots, N\}$  by  $K_{sj} = k_{\mathcal{X}_i}(X_i^s, X_i^j)$  and  $L_{sj} = k_Y(Y^s, Y^j)$  and  $H$  is defined by  $H_{sj} = \delta_{sj} - \frac{1}{N}$ . An unbiased estimator by U-statistics is also proposed in [154]. The asymptotic distribution of  $\widehat{\text{HSIC}}_v(X_i, Y)$  under  $(\mathcal{H}_0^i)$  can be used to estimate p-values and perform the independence test for each input [52]. Alternative p-value estimators have been developed in [27]. A major strength of HSIC is that its estimation requires only a single  $N$ -sample (regardless of the number of inputs), and converges fast enough (see [51] for a convergence rate) that small  $N$  (around a hundred, even less as in [25]) can be used. Moreover, the  $N$ -sample doesn't require any particular DoE. It allows the use of a chosen design that reduces the estimator variance such as Latin Hypercube Sampling (LHS) or variants as in [85] where quantization-based LHS are tested to reduce the HSIC variance.

Instead of computing p-values, HSIC itself can be considered a sensitivity index. However, for comparisons between inputs, such as comparing the influence of  $X_1$  and  $X_2$ , having  $\text{HSIC}(X_1, Y) < \text{HSIC}(X_2, Y)$  isn't enough to conclude that  $X_2$  is more influential than  $X_1$ . Normalized versions of HSIC [24] allow this comparison, but are not as satisfactory as the ANOVA decomposition of the Sobol' indices. Fortunately, recent work [25] shows an ANOVA-like decomposition of HSIC.

**HSIC-ANOVA [25]** The first step to obtain an ANOVA-like decomposition of HSIC is to restrict the input kernels  $k_{\mathcal{X}}$  to an ANOVA class of kernels [30], i.e., to have the form

$$k_{\mathcal{X}}(x, x') = \prod_{i=1}^d (1 + k_{\mathcal{X}_i}(x_i, x'_i)),$$

with  $1 \notin \mathcal{H}_{k_{\mathcal{X}_i}}$  (i.e., the RKHS induced by  $k_{\mathcal{X}_i}$  does not contain constant functions) and

$$\forall i \in \{1, \dots, d\}, \forall x_i \in \mathcal{X}_i, \int_{\mathcal{X}_i} k_{\mathcal{X}_i}(x_i, x'_i) d\mathbb{P}_{X_i}(x'_i) = 0.$$

Under the previous assumptions and the independence between the inputs, [26] show that :

$$\text{HSIC}_{k_{\mathcal{X}} \times k_Y}(X, Y) = \sum_{A \subseteq \{1, \dots, d\}} \sum_{B \subseteq A} (-1)^{|A|-|B|} \text{HSIC}_{k_{\mathcal{X}_B} \times k_Y}(X_B, Y), \quad (1.9)$$

where  $k_{\mathcal{X}_B} = \prod_{i \in B} (1 + k_{\mathcal{X}_i}(x_i, x'_i))$ . The requirement of using an ANOVA input kernel is not too restrictive, since classical kernels can be modified to become ANOVA [30]. Sobolev kernels defined as

$$k(x, x') = \frac{B_{2r}(|x - x'|)}{(-1)^{r+1}(2r)!} + \sum_{i=1}^r \frac{B_i(x)B_i(x')}{(i!)^2},$$

where  $B_i$  denotes the Bernoulli polynomial of degree  $i$  and  $r \geq 0$ , are also ANOVA kernels by construction [82] and are studied in [146] for their use in SA.

From (1.9), [26] defines HSIC-ANOVA, in the same way as Sobol's indices, normalizing by  $\text{HSIC}_k(X, Y)$ .

**Definition 1.8.** For  $i \in \{1, \dots, d\}$ ,

- *First-order HSIC-ANOVA are defined as*

$$S_i^{\text{HSIC}} = \frac{\text{HSIC}_{k_{\mathcal{X}_i} \times k_{\mathcal{Y}}}(X_i, Y)}{\text{HSIC}_{k_{\mathcal{X}} \times k_{\mathcal{Y}}}(X, Y)},$$

- *Total-order HSIC-ANOVA are defined as*

$$S_i^{T, \text{HSIC}} = 1 - \frac{\text{HSIC}_{k_{\mathcal{X}_{-i}} \times k_{\mathcal{Y}}}(X_{-i}, Y)}{\text{HSIC}_{k_{\mathcal{X}} \times k_{\mathcal{Y}}}(X, Y)}.$$

Higher-order and closed HSIC-ANOVA are defined in the same way as Sobol' indices and add up to 1 using (1.9). With these indices, the inputs  $X_1$  to  $X_d$  are ranked according to their percentage of the explained  $\text{HSIC}_k(X, Y)$ . However, such percentages are difficult to interpret, which is a major drawback of HSIC-ANOVA compared to Sobol's indices, which measure a portion of the output variance.

For complex outputs, HSIC (with or without normalization) and HSIC-ANOVA rely only on the presence of a (computable) characteristic kernel in the output space. Given any output space where a characteristic kernel is defined (and computable), HSIC and HSIC-ANOVA can be used. For example, a  $s$ -dimensional output kernel can be used if  $f(x) \in \mathbb{R}^s$ . Functional output has also been treated with respect to independence testing in [178], and for SA in [32]. However, the case of set-valued models has not yet been explored.

### Take-home message on Sensitivity Analysis

#### What is sensitivity analysis?

- Sensitivity analysis (SA) is the study of how the uncertainty in the output can be partitioned and allocated to the inputs.
- SA can be used to separate inputs into those that are significant and those that are not (screening), or to rank inputs according to their impact (ranking).
- A popular SA approach is to estimate sensitivity indices that quantify the precise impact of each input. For example
  - Sobol' indices measure the proportion of output variance explained by an input.
  - Universal indices measure a portion of output uncertainty through test functions.
  - HSIC measures input-output dependence using kernels.

#### Our contribution to sensitivity analysis :

In the following first part of the thesis, we propose three sensitivity indices in the context of set-valued outputs. They are inspired by the three indices presented above:

- HSIC for sets (detailed in Chapter 2) are based on an introduced kernel of sets  $k_{set}$ .
- Vorob'ev-based indices are an adaptation of Sobol' indices based on random set theory (briefly presented in the appendix).
- Universal indices are used with test functions that compare sets.

The proposed indices are tested on excursion sets defined as

$$\Gamma = \{x \in \mathcal{X}, f(x, U) \leq q, g(x, U) \leq 0\},$$

which appear in the context of robust optimization.

## 1.2 Robust Bayesian Optimization

Optimization, whether under uncertainty or in deterministic settings, is a well-established field of study. It has been extensively explored through well-known and widely taught optimization algorithms. Examples include gradient descent, Newton's method, Lagrange multipliers, dynamic programming, genetic algorithms, particle swarm optimization, and their various variants. Some of these algorithms deal with uncertainty efficiently, while others require adaptations. With expensive models to optimize, many existing algorithms cannot be used because they require too many model evaluations, especially since many of them rely on derivative values, the estimation of which greatly increases the cost.

To limit the number of simulations, efficient algorithms have emerged that rely on surrogate modeling (or metamodeling), i.e., low-cost models of the costly model itself. A first approach to using surrogates for optimization is to simply approach the costly model with a metamodel before using classical optimization methods on it. However, fitting the surrogate with a small amount of data can lead to large approximation errors that induce errors in the found optimum. An alternative is then to choose wisely where to spend the simulation budget, i.e., to select points of interest where the costly simulator is evaluated. This idea has led to a large number of algorithms based on iteratively constructed surrogates. As explained in [169], such algorithms have a common structure based on three steps:

- Step 1: Construct an initial design of experiments (DoE) on which to evaluate the model.
- While there is still budget:
  - Step 2: Fit the surrogate to the DoE.
  - Step 3: Use an acquisition criterion to find a new point where it is relevant to enrich the surrogate. Simulate this point and add it to the DoE.

Therefore, the number of algorithms is determined by the various combinations of the three steps: constructing the DoE, selecting the appropriate surrogate model, and choosing the acquisition criterion for adding new simulations points. For example, [136] starts with symmetric Latin hypercubes, the surrogates are based on Radial Basis Functions, and the new point is selected by maximin approach.

Among these methods, Bayesian Optimization (BO), introduced in [84, 182, 104], and popularized in [75], is characterized by the use of Bayesian statistics in surrogate modeling to quantify its uncertainty. To use Bayesian statistics, the metamodeling step is done with Gaussian process regression (GPR), and different acquisition criteria adapted to Gaussian processes (GP) have been introduced.

In the following, we first recall GPR. We then describe BO in detail, with notably a focus on robust BO, i.e., in presence of uncertainties. We refer to [173] for recent advances in BO.

### 1.2.1 Gaussian Process Regression

In this section, we assume that the reader is familiar with stochastic processes, especially Gaussian processes. We only recall the prerequisites for understanding GPR. For details, the main reference is [176]. In the following, we also use the term *kriging* to qualify GPR, even though kriging (introduced in [102]) is a specific case of it.

Let  $f$  be an unknown and expensive to evaluate function defined by

$$f : \begin{cases} \mathcal{X} & \longrightarrow \mathbb{R} \\ x = (x_1, \dots, x_d) & \longmapsto f(x) \end{cases} ,$$

where  $\mathcal{X} \subset \mathbb{R}^d$ .  $f$  is only evaluated at  $t \in \mathbb{N}^*$  points that define a design  $\mathcal{X}^{(t)} = \{x^1, \dots, x^t\}$  to get  $f(\mathcal{X}^{(t)}) = (f(x^1), \dots, f(x^t))$ . To approximate  $f$  on the whole domain, GPR relies on the assumption that  $f$  is a sample path of a Gaussian process, i.e., that there exists a Gaussian Process  $F$  on  $\mathcal{X}$  and  $\omega \in \Omega$  such that  $f(x) = F_x(\omega)$  for all  $x \in \mathcal{X}$  where  $(\Omega, \mathcal{F}, \mathbb{P})$  is a probability space (used for the rest of the manuscript). The mean and covariance of  $F$  are denoted by  $\mu_F$  and  $k_F$ , respectively, such that for all  $x, x' \in \mathcal{X}$ ,

$$\begin{aligned} \mu_F(x) &= \mathbb{E}[F_x], \\ k_F(x, x') &= \text{Cov}[F_x, F_{x'}]. \end{aligned}$$

Making assumptions about the previous mean (often called *trend*) leads to different types of GPR. For simplicity, we focus here on a constant and known trend,  $\mu_F(x) = \mu_F$  for all  $x \in \mathcal{X}$ , called *Simple Kriging*. When the trend is constant and unknown, we speak of *Ordinary Kriging*, and finally of *Universal Kriging* when a polynomial trend is assumed.

Then, the evaluations of  $f$  are used to determine a sample path of  $F$  that best fits  $f$ . To do this,  $F$  is simply conditioned on the evaluations, i.e., such that

$$F_{x^1} = f(x^1), \dots, F_{x^t} = f(x^t).$$

This leads to a conditioned process  $F^{(t)}$  which remains Gaussian with mean and covariance of known closed form called *kriging equations* (here given for simple kriging):

$$\begin{aligned}\mu_{F^{(t)}}(x) &= \mu_F + k_F(x, \mathcal{X}^{(t)})K_F^{-1} \left( f(\mathcal{X}^{(t)}) - \mu_F \right), \\ k_{F^{(t)}}(x, x') &= k_F(x, x') - k_F(x, \mathcal{X}^{(t)})K_F^{-1}k_F(x', \mathcal{X}^{(t)})^T,\end{aligned}\tag{1.10}$$

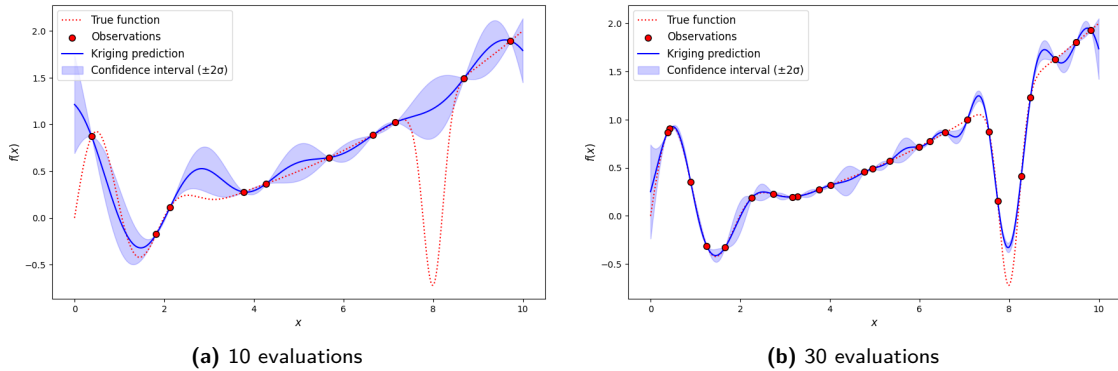
where  $k_F(x, \mathcal{X}^{(t)}) = (k_F(x, x^1), \dots, k_F(x, x^t))$  and  $K_F = (k_F(x^i, x^j))_{1 \leq i, j \leq t}$ .  $\mu_{F^{(t)}}(x)$  is then an interpolating surrogate of  $f$ , with easily computable predictions and a control of the uncertainty obtained directly as

$$F_x^{(t)} \sim \mathcal{N}(\mu_{F^{(t)}}(x), k_{F^{(t)}}(x, x)).$$

Apart from the choice of trend, the resulting surrogate strongly depends on the prior covariance function  $k_F$ . As long as  $k_F$  is positive semidefinite, any  $k_F$  can be used. In practice, however, products of one-dimensional stationary kernels are often used, such as Gaussian, exponential, Matérn, Laplace kernels, with a parameter called the length scale for each dimension. Using the model evaluations, these length scales (with the trend if necessary) are estimated according to different criteria such as Maximum Likelihood Estimation (MLE). Examples of GPR with 10 and 30 simulations are given in Figure 1.2 to fit a function  $f_{test}$  defined as

$$f_{test}(x) = e^{-\frac{x^2}{3}} \sin(3x) + \frac{x^2}{50} - 2e^{-10|x-8|^2}.$$

We observe that with a classic LHS the function is well fitted with 30 observations except near the



**Figure 1.2:** Gaussian Process regression of  $f_{test}$  with different number of points of a lhs design, using a Gaussian kernel

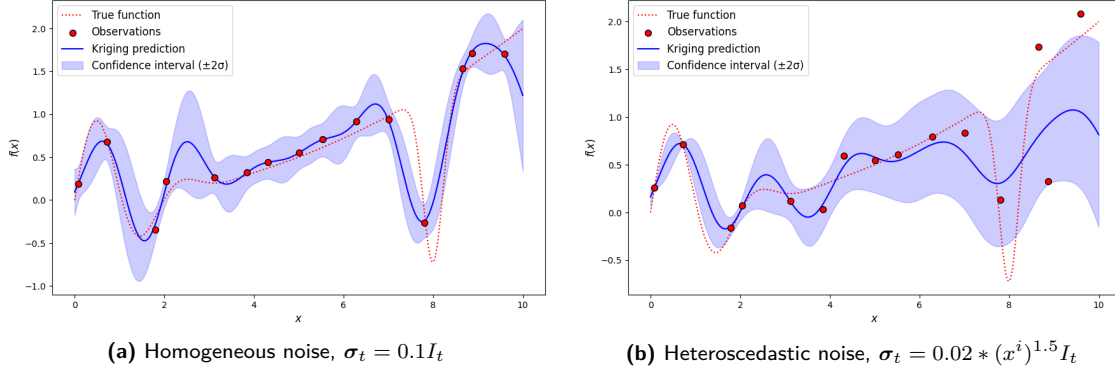
sharp minimum at  $x = 8$ . If  $f$  is to be minimized, this is very problematic, and it shows that the points should be selected according to the goal of minimizing  $f_{test}$ . This is the concept of Bayesian optimization, which is presented after a short digression on noisy GPs.

### 1.2.1.a Noisy GP

Sometimes the evaluations of the simulator  $f$  are noisy. This means that we don't have access to the evaluations  $(f(x^1), \dots, f(x^t))$  but only to noisy observations  $\tilde{f}^1, \dots, \tilde{f}^t$  such that for all  $i \in \{1, \dots, t\}$ ,

$$\tilde{f}^i = f(x^i) + e^i,$$

where the noise is supposed Gaussian, i.e.,  $e^i$  are realizations of  $\epsilon^i$  where  $\epsilon = (\epsilon^1, \dots, \epsilon^t) \sim \mathcal{N}_t(0, \sigma_t^2)$ . If the variance noise is constant for all evaluations, i.e.,  $\sigma_t^2 = \sigma^2 I_t$ , we speak of *homogeneous* noise. Otherwise it is called *heterogeneous* or *heteroscedastic* noise. Examples of GPR with homogeneous and heterogeneous noise added to  $f_{test}$  are given in Figure 1.3.



**Figure 1.3:** Noisy Gaussian Process regression of  $f_{test}$  with a lhs of 15 points and a Gaussian kernel

Adding noise doesn't add much complexity to the kriging equations (1.10). Using a fixed small *homogeneous* noise variance, called the *nugget* effect, is even a very common practice as it prevents numerical instabilities in the inversion of the covariance matrix. Homogeneous noise is also easy to estimate because it adds only one unknown parameter. On the other hand, heteroscedastic noise is much harder to deal with, as its estimation would imply one more parameter to be estimated for each model evaluation. In [48, 88, 78], alternatives have been studied where the heteroscedastic noise is assumed to be a smooth function of the inputs and is modeled by a second GP. [68] proposes to weight the initial data with the estimated noise. However, all of these methods require estimation of the noise, which often involves repeated sampling at the same locations, which is not always affordable or even possible. In [92], the work of [68] is extended to deal with noise that can only be estimated for some points and not for others.

With or without noise, GPR allows any black-box function to be efficiently modeled with a small amount of data, and to have a freely tractable uncertainty, making it suitable for Bayesian optimization.

## 1.2.2 Bayesian Optimization

In Bayesian Optimization (BO), the algorithm iteratively selects new points to be added to the training set of kriging surrogate models. This selection is based on Bayes' rule: given the observed data and the prior, Bayes' rule allows us to update our belief about the objective function. A new point is then selected according to some acquisition functions based on this updated posterior.

### 1.2.2.a Without constraints

BO was popularized by the work of [75], who introduced Efficient Global Optimization (EGO). It aims at solving the optimization problem:

$$x^* = \arg \min_{x \in \mathcal{X}} f(x). \quad (1.11)$$

This is done by fitting a kriging model  $F^{(t)}$  of  $f$  to an initial DoE. Then, the kriging model is used to quantify how likely a new point will be an *improvement*. This improvement is defined as

$$I_{F^{(t)}}(x) = \max(f_{min}^{(t)} - F_x^{(t)}, 0),$$

where  $f_{min}^{(t)} = \min(f(x^1), \dots, f(x^t))$  is the current found minimum. At each  $x$ , if  $F_x^{(t)}$  is lower than  $f_{min}^{(t)}$ , the improvement measures of how much  $F_x^{(t)}$  upgrades the minimum. Otherwise, it is 0. Then, as the improvement depends on which GP path is selected, the *Expected Improvement* (EI), introduced in [104], defined as

$$EI_{F^{(t)}}(x) = \mathbb{E}[I_{F^{(t)}}(x)] = \mathbb{E}[\max(f_{min}^{(t)} - F_x^{(t)}, 0)], \quad (1.12)$$

is used to measure the average improvement at each  $x$ . The new point  $x^{(t+1)}$  is then selected by maximizing the EI, i.e., by solving

$$x^{(t+1)} = \arg \max_{x \in \mathcal{X}} EI_{F^{(t)}}(x).$$

Then,  $f$  is evaluated at the next point  $x^{(t+1)}$  and the GP is updated. Until a stopping criterion, or while there is budget, the steps are repeated. For example, in [75], the EGO stops if a maximal EI is small than 1% of the best current function value. An example of BO of  $f_{test}$  with the EFI is given in Figure 1.4 where the conditioned GP is plotted along with the associate EI used to select the next point. We observe that the maximization of the EI allows to evaluate points near the optimum that we missed in Figure 1.2 even with more points.

Apart from EI and its variants (see [180] for a review and [126, 3] for recent work), many acquisition functions have been used and have led to many BO algorithms. Among the most well-known and used acquisition functions are the Probability of Improvement [84], the Knowledge Gradient [39], the Upper Confidence Bounds or Lower Confidence Bounds [159], the Thompson Sampling [2] or the Entropy Search [62].

### 1.2.2.b With constraints

For optimization problem with constraints, BO algorithms have been developed (see ([63]) to solve

$$x^* = \arg \min_{x \in \mathcal{K}} f(x) \text{ where } \mathcal{K} = \{x \in \mathcal{X}, g(x) \leq 0\}, \quad (1.13)$$

where  $g = (g_1, \dots, g_s) \leq 0$  denotes the respect of all constraints, i.e.  $\forall i \in \{1, \dots, s\}, g_i \leq 0$ . To deal with these constraints, a first idea is to transform the problem into an unconstrained optimization problem using methods such as penalty methods, Lagrangian-based methods, and performing classical BO. However, this leads to difficulties such as the need for non-stationary surrogates to fit the augmented Lagrangian [49]. An alternative is to fit each constraint function  $g_i$  with a Gaussian Process  $G_i$  and to adapt the acquisition criterion to the presence of constraints. For example, EI can be extended with constraints to define the Expected Feasible Improvement (EFI) [149, 147]. It is defined as the expectation of the feasible improvement:

$$\begin{aligned} EFI_{F^{(t)}, G^{(t)}}(x) &= \mathbb{E}[FI_{F^{(t)}, G^{(t)}}(x)] \\ &= \mathbb{E}\left[I_{F^{(t)}}(x) \mathbb{1}_{G_x^{(t)} \leq 0}\right] \\ &= EI_{F^{(t)}}(x) \mathbb{P}\left[G_x^{(t)} \leq 0\right]. \end{aligned}$$



**Figure 1.4:** Bayesian optimization of  $f_{test}$  with an initial lhs of 5 points selecting next points with Expected Improvement.

It corresponds to the constrained EI of [55, 166] if the GPs are assumed to be independent. Maximizing the EFI allows the added point to also verify the constraint with high probability.

Other acquisition functions have been adapted or developed for constrained optimization. The Integrated Expected Conditional Improvement allows to collect some non-admissible data [50]. The knowledge gradient can be adapted to the constraints as in [19] and [168]. The authors of [129] propose to reduce the expected volume of sets with low value of  $f$  where constraints are respected. Instead of coupling both objective and constraint evaluations, [46] proposes frameworks in order to

evaluate objective and constraints separately.

### 1.2.3 Robust Bayesian Optimization

In the presence of uncertain inputs modeled by random variables  $U = (U_1, \dots, U_p) \in \mathcal{U} \subset \mathbb{R}^p$  with known probability distribution  $\mathbb{P}_U$ , BO has been adapted and Robust Bayesian Optimization approaches have been developed both with and without constraints.

#### 1.2.3.a Without constraints

Initial optimization problem (without constraints) of Equation (1.11) is a deterministic optimization problem as neither the inputs  $x$  and the objective function  $f$  come with uncertainties. In presence of uncertain inputs modeled by random variable  $U$  of known probability distributions  $\mathbb{P}_U$ , the optimization problem becomes:

$$\text{Find } x \text{ which minimizes } f(x, U). \quad (1.14)$$

However, the minimization of  $f(x, U)$  is not clearly defined as  $f(x, U)$  is a random variable. Many formulations can be derived from (1.14), sometimes leading to very different optimization strategies (see [9] for a review on different robust optimization formulations). For example, we may decide to solve deterministic versions of (1.14) for fixed  $U = U(\omega)$  and then derive a robust optimum. Alternatively, robustness metrics [103] can be used to account for uncertainties in the objective function and to derive a robust optimization problem [89].

In this thesis, we will focus at minimizing  $\mathbb{E}[f(x, U)]$ , which leads to the optimization problem:

$$x^* = \arg \min_{x \in \mathcal{X}} \mathbb{E}[f(x, U)]. \quad (1.15)$$

There are several alternatives to these formulations. Common robustness metrics are the *minimax* and *minimin* for worst-case and best-case optimization [170]. The authors of [110, 57, 135] use both  $\mathbb{E}[f(x, U)]$  and  $\text{Var}[f(x, U)]$  in a multi-objective optimization, while [120, 29] use linear combinations of the two. In [167, 1, 165], the authors minimize the regret (as defined in [148]).

In the BO field, the presence of uncertain inputs has led to the development of *Robust* Bayesian optimization algorithms. The algorithms are often designed for one of the previous problem formulations, which has led to numerous robust BO approaches (see [118, 41, 15, 79, 130, 175, 139] for some examples). For the formulation (1.15), [74] proposes the EI-VAR algorithm, which we describe in detail.

In [74], the presence of uncertain inputs leads to the following changes compared to a classical EGO-like algorithm:

- After modeling  $f$  by a GP  $F$  in the joint space  $\mathcal{X} \times \mathcal{U}$ , the mean process  $Z^{(t)}$  defined by

$$Z_x^{(t)} = \mathbb{E}_U \left[ F_{(x, U)}^{(t)} \right],$$

is used to model  $\mathbb{E}[f(x, U)]$ . Note that  $Z^{(t)}$  remains Gaussian, which makes it easy enough to compute its prediction mean and covariance.

- A region of interest in the design space is found by determining  $x^{targ}$  that maximizes the classical EI but of the expected process  $Z$ :

$$\begin{aligned} x^{targ} &= \arg \max_{x \in \mathcal{X}} EI_{Z^{(t)}}(x) \\ &= \arg \max_{x \in \mathcal{X}} \mathbb{E} \left[ \max(z_{min}^{(t)} - Z_x^{(t)}, 0) \right]. \end{aligned}$$

An important difference here is that the improvement

$$I_{Z^{(t)}}(x) = \max(z_{min}^{(t)} - Z_x^{(t)}, 0),$$

is calculated with  $z_{min}^{(t)} = \min_{1 \leq i \leq t} \mathbb{E} [Z_{x^i}^{(t)}]$ , which is not observed but estimated.

- Then, the pair  $(x^{t+1}, u^{t+1})$ , is selected by minimizing the one step ahead variance at the point of interest  $x^{targ}$ , i.e., the variance if  $(x, u)$  is added to the design:

$$(x^{t+1}, u^{t+1}) = \arg \min_{x, u \in \mathcal{X} \times \mathcal{U}} \text{Var} \left[ Z_{x^{targ}}^{(t)} | F^{(t)}(x, u) = f(x, u) \right].$$

Note that this optimization doesn't require any new evaluations of  $f$  but only on computations using the regression models  $F$  and  $Z$  as explained in [74].

- Finally, at the end of the algorithm, when the entire budget is consumed, the point chosen as the best solution is the 90% quantile of  $Z$  among the evaluated points, defined as

$$x^* = \arg \min_{1 \leq i \leq t} \left( \mathbb{E} [Z_{x^i}^{(t)}] + 1.282 \sqrt{\text{Var} [Z_{x^i}^{(t)}]} \right).$$

This final criterion is necessary in the context of robust BO as no true evaluations of  $\mathbb{E} [f(x, U)]$  are done, so the uncertainty on the estimation of  $\mathbb{E} [f(x, U)]$  must be taken into account.

### 1.2.3.b With constraints: EFISUR

With constraints, most robust formulations are made in terms of the probability of satisfying the constraints, i.e., such that  $\mathbb{P} [g(x, U) \leq 0]$  is above  $\alpha$ , also called *chance constrained* optimization. Along with the expectation of the objective function, this leads the following robust constrained optimization problem

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E} [f(x, U)] \text{ where } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P} [g(x, U) \leq 0] \geq \alpha\}, \quad (1.16)$$

on which we focus in this thesis.

To solve (1.16), the authors of [33] propose an algorithm called EFISUR for Expected Feasible Improvement with Stepwise Uncertainty Reduction sampling. They adapt the previous EI-VAR algorithm to deal with constraints and propose to:

- Model  $g_i$  by independent Gaussian processes  $G_i$  in the joint space  $\mathcal{X} \times \mathcal{U}$  and derive a single constraint process  $C$  defined as

$$C_x = \alpha - \mathbb{P}_U [G_{(x, U)} \leq 0],$$

to model the probability constraint  $\mathbb{P} [g(x, U) \leq 0] \geq \alpha$ . Contrary to  $Z$ ,  $C$  is no longer Gaussian.

- Replace the EI on  $Z$  by the EFI on  $Z$  and  $C$  :

$$EFI_{Z^{(t)}, C^{(t)}}(x) = EI_{Z^{(t)}}(x) \mathbb{P} [C_x^{(t)} \leq 0],$$

to get new  $x^{targ}$  by maximizing EFI. In the expression of the improvement, the current best feasible minimum is defined as:

$$z_{min}^{feas, (t)} = \min_{1 \leq i \leq t} \mathbb{E} [Z_{x^i}^{(t)}] \text{ s.t. } \mathbb{E} [C_{x^i}^{(t)}] \leq 0.$$

- Fix  $x^{t+1} = x^{targ}$  and choose only  $u^{t+1}$  by minimizing a proxy  $S$  of the one step ahead variance of the feasible improvement:

$$u^{t+1} = \arg \min_{u \in \mathcal{U}} S(x^{targ}, u^{t+1}),$$

where  $S$  is defined as

$$S(x, u) = \text{Var} \left[ I_{Z^{(t)}}(x_{targ}) | F_{(x,u)}^{(t)} = f(x, u) \right] \\ \times \mathbb{E}_U \left[ \text{Var}_\omega \left[ \mathbb{1}_{G_{(x^{targ}, U)}^{(t)} \leq 0} | G_{(x,u)}^{(t)} = g(x, u) \right] \right].$$

In [33], this criteria is chosen as minimizing it allows to reduce the variance at  $x_{targ}$  of the improvement i.e. where  $Z$  takes low value and of the constraint feasibility.

- Trusting the algorithm to reduce the variance at selected points, the final optimum is simply taken as the best feasible minimum  $z_{min}^{feas, (t)}$ .

For the rest of the thesis, we focus on EFISUR algorithm. We detail it in Algorithm 1 given in the introduction of the second part of the thesis.

In [33], EFISUR is tested on two test cases of dimensions  $d + p$  equal to  $2 + 2$  and  $20 + 7$  and compared with the EFIRAND algorithm, which is a copy of EFISUR except that  $u^{t+1}$  is chosen at random. It provides efficient results and shows the benefits of optimizing  $S$ . Adaptation of EFISUR with a Gaussian vector  $G$  (instead of independent Gaussian variables  $G_i$ ) has been studied in [125].

#### 1.2.4 Bayesian optimization in high dimension

BO is known to struggle in high input dimension. Explained for example in [11, 95], these difficulties come from two main factors:

- Fitting GPs in high-dimensional input spaces is challenging. This difficulty arises from the increasing number of parameters that need to be estimated and the fact that covariance functions are often based on distances, which behave differently in high dimensions.
- Optimizing acquisition functions becomes more difficult in high-dimensional input spaces. This challenge stems from the complexity of optimization in high dimensions and the increased computational difficulty of estimating acquisition criteria.

To avoid these problems, several methods have been proposed. Some of them rely on adaptations to fit the GP in high dimensions using sum of low dimensional models as in [31, 77, 58, 77, 4]. For example, [4] suggests decomposing the model into a weighted sum of sub-kriging models, which simplifies both the fitting of the GPs and the optimization of the inclusion criteria. The use of alternative kernels may also be a solution as proposed in [163, 117, 69] where the authors suggest including the dimension as a parameter in the prior so that it remains usable in high dimension.

The acquisition criterion optimization can also be adapted to high dimension. For example, [179] proposes the expected coordinate improvement, which includes a coordinate selection and [133] proposes to adapt the optimization of EI by reducing the GP length scale along the optimization.

To deal with the huge size of the input space, other work rely on local search, by adaptively optimizing in subsets (of the same dimension) of the search space such as in [34, 112, 171, 115, 177].

Finally, many works assume that the optimization problem has a "low effective dimension" and propose to find a reduced number of input variables (among the initial variable or not) to simplify the BO. These approaches can be divided into two types:

- **Offline** approaches aim to reduce the dimension separately from the BO. Whether before the BO or between iterations, dimension reduction allows BO steps to be performed in low dimensions.
- **Online** approaches dwell in the BO and reduce the dimension of the acquisition function optimization.

Among the offline methods we can mention [40, 174, 12, 113, 108, 90], which choose a subspace in which the whole BO is performed. Adjusting the dimension reduction after some iterations is also done in [61, 155, 121].

Online methods have also been proposed with different variable selection methods. [91] simply studies dropout strategies while randomly selecting some inputs before optimizing the acquisition criteria. [156] improves the method by selecting inputs using goal-oriented SA techniques (presented in the next section). [172] relies on tree search to select important variables. [142] identifies influential variables by the covariance correlation lengths of the GP and the subset of influential variables is updated at each step using a doubt function.

#### Take-home message on Bayesian Optimization

##### What is Bayesian optimization?

- BO is a method for optimizing expensive black-box functions.
- It is based on Gaussian process modeling along with sequential learning obtained by optimizing well-chosen acquisition functions.
- It has been adapted to deal with both constraints and uncertainties, resulting in numerous algorithms over the past decades.
- It struggles with high input dimensions, which has led to adaptations with mainly dimension reduction methods.

##### Our contribution to Bayesian optimization :

In the thesis, we use the robust Bayesian Optimization algorithm EFISUR to solve

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E}[f(x, U)] \text{ where } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P}[g(x, U) \leq 0] \geq \alpha\}.$$

In the second part of the thesis, we propose ways to make EFISUR usable in high input dimension, relying on online and offline methods based on sensitivity analysis tools. We also propose a different validation step to overcome the GPs errors increased by the high input dimension.

## 1.3 Goal-oriented Sensitivity Analysis for (Bayesian) Optimization .....

In the previous sections, we introduced the fields of Sensitivity Analysis and robust Bayesian Optimization. In this section, we explain how SA can be used for optimization. First we present Goal-Oriented Sensitivity Analysis (GOSA) tools. Then we detail how existing work uses GOSA for (Bayesian) Optimization.

### 1.3.1 Goal-Oriented Sensitivity Analysis

The indices introduced in Section 1.1 are used to measure the influence of inputs  $(X_1, \dots, X_d)$  on an output  $Y = f(X_1, \dots, X_d)$ . In many cases, however, we are not interested in the effect of  $X_i$  on  $Y$ ,

but rather in some specific goal related to  $Y$  that we want to achieve. This has led to the framework of Goal-oriented Sensitivity Analysis (GOSA), defined in [38], in which the authors quantify the influence of inputs on a risk measure related to the probability distribution of  $Y$ . Different types of GOSA have been studied in the literature (for instance, regional SA in [158] or quantile-based SA in [81, 76]).

We focus on the framework for GOSA proposed in [97], where the authors define two types of GOSA. Given an output domain of interest  $\mathcal{C}$ ,

- Target Sensitivity Analysis (TSA), originally introduced in [157], measures the effect of inputs on the occurrence of a phenomenon defined by  $\{Y \in \mathcal{C}\}$ ,
- Conditional Sensitivity Analysis (CSA) measures the effect of the inputs on the output while it is within the range of interest, i.e., on  $Y|Y \in \mathcal{C}$ .

### 1.3.1.a Target Sensitivity Analysis

To measure the effects of inputs on the occurrence of  $\{Y \in \mathcal{C}\}$ , the natural idea, proposed in [157], is to consider the binary variable  $\mathbb{1}_{Y \in \mathcal{C}}$ . From this, any sensitivity indices usable for discretized outputs can be used by replacing the output with  $\mathbb{1}_{Y \in \mathcal{C}}$ . For example, in [93], first-order Sobol' indices are derived:

$$S_i^{\mathbb{1}} = \frac{\text{Var}[\mathbb{E}[\mathbb{1}_{Y \in \mathcal{C}} | X_i]]}{\text{Var}[Y]}.$$

However, this is criticized in [97] as it doesn't take into account the distance to  $\mathcal{C}$ . In fact, with  $\mathbb{1}_{Y \in \mathcal{C}}$ , points very close to the boundary of  $\mathcal{C}$  or very far away are treated equally. To avoid this problem, [97] propose a smooth thresholding  $w$  using as a reference  $w(y) = \exp(-d(y, \mathcal{C})/s)$ , where  $d(y, \mathcal{C})$  is a distance from  $y$  to  $\mathcal{C}$  and  $s$  is a constant parameter. Then by simply replacing  $Y$  with  $w(Y)$ , TSA Sobol' indices and TSA HSIC can be defined.

**Definition 1.9.** Given an output transformation  $w : \mathcal{Y} \mapsto \mathbb{R}$ , TSA Sobol' indices  $T\text{-}S_{i,w}$  and TSA HSIC indices  $T\text{-}HSIC_k^w(X_i, Y)$  are defined as

$$T\text{-}S_{i,w} = \frac{\text{Var}[\mathbb{E}[w(Y) | X_i]]}{\text{Var}[w(Y)]},$$

$$T\text{-}HSIC_k^w(X_i, Y) = HSIC_k(X_i, w(Y)).$$

Sobol' indices of any order, as well as normalizations of HSIC and notably HSIC-ANOVA can all be derived in the same way. Only the kernel choice for HSIC may be affected, especially when  $w(Y)$  is binary or discretized. The same estimation procedures can be used.

### 1.3.1.b Conditional Sensitivity Analysis

On the other hand, CSA requires a bit more work. To measure the effect of  $Y$  inside  $\mathcal{C}$ , ignoring what happens outside, the output of interest becomes  $Y|Y \in \mathcal{C}$ . To use this output directly in the previous indices, an i.i.d. sample  $((X_{\mathcal{C}}^1, Y_{\mathcal{C}}^1), \dots, (X_{\mathcal{C}}^N, Y_{\mathcal{C}}^N))$  of the parent distribution  $\mathbb{P}_{(X,Y)|Y \in \mathcal{C}}$  would be necessary, but is not easily affordable, especially when  $\mathcal{C}$  is a rare event.

To avoid this problem, it is proposed to estimate the Sobol' indices with the following output transformation:

$$\tilde{Y}_w = w(Y)Y + (1 - w(Y))c, \quad (1.17)$$

so that  $\tilde{Y}_w \approx Y$  in  $\mathcal{C}$ , i.e., when  $w(Y) = 1$ , and is constant equal to  $c$  otherwise. The constant  $c$  is fixed at  $\frac{\mathbb{E}[w(Y)Y]}{\mathbb{E}[w(Y)]}$ , obtained by imposing  $\mathbb{E}[\tilde{Y}_w] = c$ , i.e. that the transformation is equal to its mean when not in  $\mathcal{C}$ . This results in conditional first-order Sobol' indices.

**Definition 1.10.** Given  $\tilde{Y}_w$  defined by (1.17), Conditional first-order Sobol' indices  $C-S_{i,w}$  are defined as

$$C-S_{i,w} = \frac{\text{Var} [\mathbb{E} [\tilde{Y}_w | X_i]]}{\text{Var} [\tilde{Y}_w]}.$$

Classical estimation of Sobol' indices can be used to estimate  $C-S_{i,w}$ .

For HSIC indices, a better approach is to write the conditional probability distribution of  $Y | \{Y \in \mathcal{C}\}$  defined by for  $B \subset \mathbb{R}$  by

$$\mathbb{P}_{Y|Y \in \mathcal{C}}(B) = \frac{\int_B \mathbb{1}_{\mathcal{C}}(y) d\mathbb{P}_Y(y)}{\int_{\mathcal{C}} \mathbb{1}_{\mathcal{C}}(y) d\mathbb{P}_Y(y)}.$$

From this, [97] again proposes a "weighted conditional density", denoted  $\mathbb{P}_Y^w$ , replacing the indicator by a weighted transformation  $w$ :

$$\mathbb{P}_Y^w(B) = \frac{\int_B w(y) d\mathbb{P}_Y(y)}{\int_{\mathcal{C}} w(y) d\mathbb{P}_Y(y)} = \frac{\int_B w(y) d\mathbb{P}_Y(y)}{\mathbb{E}[w(Y)]},$$

under which  $Y$  expectation is immediate :

$$\mathbb{E}_{Y \sim \mathbb{P}_Y^w} = \mathbb{E} \left[ \frac{w(Y)Y}{\mathbb{E}[w(Y)]} \right].$$

Then, defining similarly  $\mathbb{P}_X^w$  and  $\mathbb{P}_{(X,Y)}^w$ , [97] obtains:

$$\begin{aligned} \text{C-HSIC}_k^w(X_i, Y) &= \text{HSIC}_{k, (X_i, Y) \sim \mathbb{P}_{(X_i, Y)}^w}^w(X_i, Y) \\ &= \mathbb{E} [k_{\mathcal{X}_i}(X_i, X'_i) k_{\mathcal{Y}}(Y, Y') \bar{w}(Y) \bar{w}(Y')] \\ &\quad + \mathbb{E} [k_{\mathcal{X}_i}(X_i, X'_i) \bar{w}(Y) \bar{w}(Y')] \times \mathbb{E} [k_{\mathcal{Y}}(Y, Y') \bar{w}(Y) \bar{w}(Y')] \\ &\quad - 2\mathbb{E} [\mathbb{E} [k_{\mathcal{X}_i}(X_i, X'_i) | X_i] \mathbb{E} [k_{\mathcal{Y}}(Y, Y') \bar{w}(Y) \bar{w}(Y') | Y]] \end{aligned}$$

where  $\bar{w}(Y) = \frac{w(Y)}{\mathbb{E}[w(Y)]}$ . It can then lead to an estimation similar to HSIC, also described in [97].

p-values can still be estimated with permutation-based tests. Since the conditioned inputs are no longer independent, HSIC-ANOVA cannot be used.

### 1.3.2 Offline and Online Goal-oriented Sensitivity Analysis for Optimization

Both TSA and CSA approaches can be used in constrained optimization by considering the domain of interest  $\mathcal{C}$  to be an area of low objective function value and where all constraints are respected. They then allow to measure the effect of the inputs to reach the optimum or within an area close to the optimum. It has been used for optimization in some recent work with both offline and online methods.

**OFFLINE SENSITIVITY ANALYSIS FOR OPTIMIZATION** One example of offline dimension reduction by goal-oriented SA before optimization is proposed in [157] using goal-oriented indices, which can be seen as specific cases of TSA. The authors aim at reducing the input dimension of the following problem:

$$x^* = \arg \min_{x \in \mathcal{K}} f(x) \text{ where } \mathcal{K} = \{x \in \mathcal{X}, g(x) \leq 0\}. \quad (1.18)$$

They introduce HSIC with indicator thresholding:

$$S_{q,T}^{\text{HSIC}}(X_i) = \text{HSIC}(X_i, \mathbb{1}_{f(X) \leq q, g(X) \leq T}).$$

This indices correspond to an extension of TSA-HSIC with a 2-dimensional output  $Y = (Y_1, Y_2)$  and  $w(Y) = \mathbb{1}_{Y_1 \leq q, Y_2 \leq T}$ . Then, with  $q$  a low quantile of  $f$  and  $T$  near 0,  $S_{q,T}^{\text{HSIC}}(X_i)$  allows to quantify the impact of  $X_i$  on minimizing  $f$  while  $g \leq 0$ . By keeping only the most influential inputs (less than 10% of the sum of all indices), the authors of [157] reduce the optimization dimension. The negligible inputs are either fixed at random values or at the values given by the best feasible point obtained during the sensitivity analysis. Then the optimization is performed with the COBYLA algorithm but any optimization method can then be used.

ONLINE SENSITIVITY ANALYSIS FOR OPTIMIZATION To solve

$$x^* = \arg \min_{x \in \mathcal{K}} f(x), \quad (1.19)$$

where  $f$  is a costly black-box function, the authors of [156] propose an online approach to reduce the dimension by SA in a BO algorithm. To do so, they compute  $S_{q,T}^{\text{HSIC}}(X_i)$  on the kriging mean at each step of the BO to select the influential variable with which the acquisition criterion is optimized, while the other inputs are fixed according to a different strategy.

To the best of our knowledge, no existing work proposes to use SA in robust Bayesian optimization. Therefore, inspired by the two previously presented methods, this thesis presents both online and offline methods to incorporate SA in the EFISUR algorithm.

#### Take-home message on the use of Sensitivity Analysis for (Bayesian) Optimization

##### How can sensitivity analysis be used for optimization?

- Given a region of interest for optimization, such as where the objective function is below a low value and the constraint is verified, sensitivity analysis can be goal-oriented to measure the impact of the inputs
  - to reach the region of interest with Target Sensitivity Analysis (TSA)
  - within the region of interest with Conditional Sensitivity Analysis (CSA)
- Goal Oriented Sensitivity Analysis can be used for optimization
  - with **Offline** approaches that consist in selecting influential inputs to reduce the input dimension.
  - with **Online** for Bayesian optimization by selecting influential inputs at each iteration to reduce the dimension of the acquisition function optimization.

##### Our contribution to sensitivity analysis for Bayesian optimization :

In this thesis, we propose ways to use goal-directed sensitivity analysis in the robust BO algorithm EFISUR. Both online and offline methods are adapted to the presence of uncertain inputs, with in particular the use of sensitivity indices of excursion sets proposed in the first part.



# I

## **Sensitivity Analysis of set-valued models**



# Introduction

One goal of the thesis is to quantify the impact of uncertain inputs in the context of robust optimization. In this first part, we propose to achieve this goal by studying how the uncertain inputs  $U = (U_1, \dots, U_p)$  affect excursion set  $\Gamma$  of the optimization problem, defined as

$$\Gamma = \{x \in \mathcal{X}, f(x, U) \leq q, g(x, U) \leq 0\},$$

i.e., the set of design inputs such that the objective function  $f$  takes low values, and the constraint  $g \leq 0$  is satisfied. Because of the uncertainty of  $U$ ,  $\Gamma$  is a random set on which we propose to perform a sensitivity analysis to measure the impact of each  $U_i$ . This raises the main question of this first part of **how to perform a sensitivity analysis with set-valued outputs**. Traditional sensitivity analysis methods, designed primarily for scalar outputs and extended to multiple types of outputs, fall short when dealing with set-valued models. Therefore, the next two chapters focus on advancing sensitivity analysis (SA) methods to effectively handle the intricacies of set-valued outputs, thereby providing a more comprehensive understanding of the relationships between inputs and these complex outputs.

The first chapter is entitled *Kernel-based sensitivity analysis for (excursion) sets*. The core idea revolves around using and adapting existing kernel-based SA techniques, with a particular focus on the Hilbert-Schmidt Independence Criterion (HSIC), to the context of set-valued outputs. Kernel-based SA methods provide a robust framework for handling any kind of input and output, as described in [86]. They rely mostly on having a (characteristic) kernel in the space of interest, but also on a probabilistic theory to define random elements on the space of interest. Therefore, the chapter consists essentially of the presentation of probabilistic framework for random sets adapted to excursion sets (different from the random set theory of [105]), together with the introduction of a kernel  $k_{set}$  on sets and its study. In particular, we show that  $k_{set}$  is characteristic (injectivity of the mean embedding), relying on a result of [181] that studies characteristic kernels in general spaces. While the method is initially developed with excursion sets in mind, its applicability extends to a broader range of set-valued models. This chapter not only introduces the theoretical underpinnings of the proposed method, but also demonstrates its potential to address real-world challenges associated with set-valued outputs.

In Chapter 3 entitled *Sensitivity analysis for sets: application to pollutant concentration maps*, two other types of SA indices for set-valued models are introduced. The first are inspired by Sobol' indices using Vorob'ev expectation and deviation from the random set theory of [105]. The second ones are *universal* indices of [37] designed to work in any kind of output space. The use of test functions on sets makes them usable to deal with random sets. These two indices, along with the indices introduced in the first chapter, are tested and compared on a toy function. They are then applied to a real test case of a pollutant concentration map model. This allows for a comprehensive sensitivity analysis that enhances our understanding of how different factors influence concentration pollutant maps.



# 2

## Article 1 : Kernel-based Sensitivity Analysis for (Excursion) Sets

---

|       |                                                                     |    |
|-------|---------------------------------------------------------------------|----|
| 2.1   | Introduction                                                        | 33 |
| 2.2   | Kernel-based SA in a generic space $\mathcal{Z}$                    | 34 |
| 2.2.1 | Kernel Embedding, MMD and HSIC                                      | 34 |
| 2.2.2 | HSIC-ANOVA indices                                                  | 36 |
| 2.3   | Kernel-based Sensitivity Analysis for sets                          | 38 |
| 2.3.1 | Probabilistic framework of random sets                              | 38 |
| 2.3.2 | A kernel between sets                                               | 38 |
| 2.3.3 | Estimation of HSIC-ANOVA indices for sets                           | 39 |
| 2.4   | Numerical Experiments                                               | 41 |
| 2.4.1 | Excursion sets of a toy function                                    | 41 |
| 2.4.2 | Excursion sets of the optimization of an oscillator                 | 43 |
| 2.4.3 | Sensitivity analysis for robust conception of an electrical machine | 45 |
| 2.5   | Conclusion                                                          | 46 |

---

# Kernel-based Sensitivity Analysis for (excursion) sets

N. Fellmann<sup>†,1,2</sup>, C. Helbert<sup>§,1</sup>, C. Blanchet<sup>§,1</sup>, A. Spagnol<sup>§,2</sup>, D. Sinoquet<sup>§,2</sup>

<sup>1</sup>École Centrale de Lyon, CNRS UMR 5208, Institut Camille Jordan  
{noe.fellmann,christophette.blanchet,celine.helbert}@ec-lyon.fr

<sup>2</sup>IFP Energies Nouvelles  
{noe.fellmann,delphine.sinoquet,adrien.spagnol}@ifp.fr

Fellmann, N., Blanchet-Scalliet, C., Helbert, C., Spagnol, A., & Sinoquet, D. (2024). Kernel-based Sensitivity Analysis for (Excursion) Sets. *Technometrics*, 113.  
<https://doi.org/10.1080/00401706.2024.2336537>

## Abstract

In this paper, we aim to perform sensitivity analysis of set-valued models and, in particular, to quantify the impact of uncertain inputs on feasible sets, which are key elements in solving a robust optimization problem under constraints. While most sensitivity analysis methods deal with scalar outputs, this paper introduces a novel approach to perform sensitivity analysis with set-valued outputs. Our innovative methodology is designed for excursion sets, but is versatile enough to be applied to set-valued simulators, including those found in viability fields, or when working with maps like pollutant concentration maps or flood zone maps.

We propose to use the Hilbert-Schmidt Independence Criterion (HSIC) with a kernel designed for set-valued outputs. After setting a probabilistic framework for random sets, a first contribution is the proof that this kernel is *characteristic*, an essential property in a kernel-based sensitivity analysis context. To measure the contribution of each input, we then propose to use HSIC-ANOVA indices. With these indices, we can identify which inputs should be neglected (*screening*) and we can rank the others according to their influence (*ranking*). The estimation of these indices is also adapted to the set-valued outputs. Finally, we test the proposed method on three test cases of excursion sets.

*Keywords:* HSIC-ANOVA indices, random sets, screening, ranking

## 2.1 Introduction

In many fields, it is essential to understand the input/output relationships of models that simulate complex physical systems. This knowledge can be used to simplify and optimize the model and provide valuable insights to experts. Sensitivity analysis (SA) is one response to this challenge. It quantifies how variations in inputs translate into variations in outputs, precisely measuring the impact of each input on the output. However, in certain domains, models may have highly complex outputs that traditional SA methods, originally designed for scalar outputs, may not be suitable for. Specifically, we are interested in models with set-valued outputs, where each evaluation of the model produces a subset of a larger space. Our interest lies in the need to quantify the influence of parameters on the excursion sets of optimization problems, with the goal of simplifying constrained robust optimization problems. Set-valued output models are also prevalent in other domains, including image processing, map modeling, and viability fields. Therefore, we aim to measure the impact of each input, on a set-valued output through an adapted SA approach.

Global sensitivity analysis (GSA) is a methodology used to assess the impact of input variations on the output of a system or model across the entire parameter space. A detailed review of GSA methods can be found in the book [26] or in [70]. These methods can be distinguished into two types:

- *screening-oriented* SA techniques are devoted to identifying influential and non-influential inputs;
- *ranking-oriented* SA aims to compute sensitivity indices or importance measures, which are scalars representing the effect of an input or group of inputs on the output. These indices are then used to rank the inputs by order of influence.

One well-known screening technique is the Morris method [109], which is based on one-at-a-time designs, i.e., where each input varies while the others are held fixed. Regarding ranking methods, the most commonly used indices are the Sobol' indices, which are variance-based sensitivity measures [152]. These indices quantify the portion of the output variance that can be attributed to one or a group of inputs. This variance decomposition is called the ANOVA (ANalysis Of VAriance) decomposition. However, these indices have several drawbacks. One main issue is that their estimation cost in terms of number of model evaluations is very high. For instance, the simulation budget needed for the 'pick and freeze' method [153] increases linearly with the input dimension. Rank-based estimators have recently been used to circumvent this problem [42], but at the cost of not being able to estimate the higher-order Sobol' indices. Another major drawback is that Sobol' indices quantify the input contributions to the output variance, but not to the entire output distribution. That is why other types of indices examine how the entire output distribution is affected by the input parameters. This is done by comparing the input and output probability distributions. For instance, Borgonovo's indices compare the density functions [13], Cramér-von Mises indices look at the cumulative distribution function [44], and kernel-based sensitivity indices compare the embedding of the distributions [25]. The former method relies on embedding distributions in Reproducing Kernel Hilbert Spaces (RKHS) using kernel functions, which makes it easier to compute distances. By embedding the joint input-output distribution and the product of marginal distributions in a well-chosen RKHS, indices based on the Hilbert-Schmidt Independence Criterion (HSIC) [53] can be defined. These indices quantify the dependence between an input and the output and can be used for screening. They also have a low cost because they can be estimated using a single Monte Carlo design of experiments. Under specific assumptions on the input kernels, an ANOVA-like decomposition of HSIC exists, as shown in [25], which makes them usable for ranking.

Several works in the literature have addressed SA adapted to complex outputs. However, to

the best of our knowledge, the specific case of set-valued outputs has not yet been studied. For instance, [65, 99, 127] examined spatial outputs, while [28, 28] explored spatiotemporal outputs. These methods generate sensitivity index maps that can be interpreted by connecting them to the underlying physical phenomena. However, there is a strong motivation, whether due to interpretation challenges or synthesis concerns, to find a single scalar index per input that captures its influence on the variability of the entire complex output. This is explored and presented in [43], which defines aggregated Sobol' indices for vector and functional outputs. In [45], the authors also propose a universal index that can be used in any metric space. Kernel-based indices, in particular the HSIC, can also be used with complex outputs, as proposed in [24, 32] with functional outputs. Adaptation of HSIC to complex output is in fact straightforward, kernel methods being known to be flexible with respect to the type of data (see for example [151], where kernel methods are used with several types of data such as vectors, texts, trees...). The flexibility of the latter lies in the fact that they mostly rely only on defining a kernel in the relevant space (where random elements can be defined), making them particularly permissive in accommodating different input and output types. For this flexibility, low cost, ability to quantify changes across the entire output distribution, and their utility for both screening and ranking purposes, we propose to use HSIC-ANOVA indices to conduct a SA of set-valued models.

To perform kernel-based SA on a set-valued model, there are two main requirements. First, a framework of random sets is required. Indeed, a space of sets is not an easy space to work in, and describing randomness in such a space can be challenging. For example, [105, 114] propose a complete theory of random sets. The second key to performing kernel-based SA on set-valued outputs is to have a kernel defined on sets. Given these two conditions, we can apply the kernel-based SA methodology to define sensitivity indices for set-valued outputs. This is the focus of this paper.

To this end, we first recall in Section 2.2 the general methodology for defining kernel-based sensitivity indices, and in particular the Hilbert-Schmidt Independence Criterion (HSIC), in a generic measurable output space. We then present our contributions in Section 2.3. First, we define the probabilistic framework for random sets. Then, it is shown that a generalization of the Gaussian kernel can be used to handle random sets, and the theoretical properties of this kernel are carefully studied. In particular, we show that this kernel is *characteristic*, a crucial property for screening purposes. Finally, HSIC-ANOVA indices for set-valued outputs are derived, but their estimation is made complicated by the presence of sets. Therefore, a nested Monte Carlo estimator is proposed to estimate the indices and its statistical properties are studied. Finally, numerical results obtained from two test cases related to excursion sets and from an industrial test case for electrical machine design are given in Section 2.4. Proofs of the results of Section 2.3 and some additional numerical results are given in the Supplementary Materials.

## 2.2 Kernel-based SA in a generic space $\mathcal{Z}$ .....

In this section, we explain how HSIC-ANOVA indices can be derived from kernel theory in a generic space. For a complete kernel theory, the reader can refer to the book *Support Vector Machine* of [161].

### 2.2.1 Kernel Embedding, MMD and HSIC

Let  $(\mathcal{Z}, \mathcal{A})$  be a measurable space on which  $k_{\mathcal{Z}}$  is a kernel, i.e., a symmetric and positive definite function  $k_{\mathcal{Z}} : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ . As stated in Theorem 4.21 of [161], there exists a unique Reproducing Kernel Hilbert Space (RKHS)  $\mathcal{H}_{k_{\mathcal{Z}}}$  of reproducing kernel  $k_{\mathcal{Z}}$ , i.e a kernel such that

- $\forall z \in \mathcal{Z}, k_{\mathcal{Z}}(\cdot, z) \in \mathcal{H}_{k_{\mathcal{Z}}}$
- $\forall z \in \mathcal{Z}, \forall f \in \mathcal{H}_{k_{\mathcal{Z}}}, \langle f, k_{\mathcal{Z}}(\cdot, z) \rangle_{\mathcal{H}_{k_{\mathcal{Z}}}} = f(z)$ .

Let  $\mathcal{M}_+^1(\mathcal{Z})$  be the space of probability measures on  $\mathcal{Z}$ . By Lemma 3.1 of [111], a *bounded* and *measurable* kernel  $k_{\mathcal{Z}}$  is a sufficient condition for  $\mathcal{M}_+^1(\mathcal{Z})$  to be embedded in  $\mathcal{H}_{k_{\mathcal{Z}}}$  by the kernel mean embedding defined as follows.

**Definition 2.1.** The kernel mean embedding of  $\mathcal{M}_+^1(\mathcal{Z})$  in  $\mathcal{H}_{k_{\mathcal{Z}}}$  is defined as

$$\begin{aligned} \mathcal{M}_+^1(\mathcal{Z}) &\rightarrow \mathcal{H}_{k_{\mathcal{Z}}} \\ \mu_{k_{\mathcal{Z}}} : \mathbb{P} &\mapsto \mu_{k_{\mathcal{Z}}}(\mathbb{P}) = \int_{\mathcal{Z}} k_{\mathcal{Z}}(z, \cdot) d\mathbb{P}(z). \end{aligned}$$

The distance between the kernel mean embedding of two distributions  $\mathbb{P}$  and  $\mathbb{Q}$  is called the Maximum Mean Discrepancy (MMD) and is denoted by  $\text{MMD}_{k_{\mathcal{Z}}}(\mathbb{P}, \mathbb{Q}) = \|\mu_{k_{\mathcal{Z}}}(\mathbb{P}) - \mu_{k_{\mathcal{Z}}}(\mathbb{Q})\|_{\mathcal{H}_{k_{\mathcal{Z}}}}$  (see Figure 2.1). To be a distance between the distributions  $\mathbb{P}$  and  $\mathbb{Q}$ , the kernel mean embedding  $\mu_{k_{\mathcal{Z}}}$  must be injective. In this case, the kernel  $\mu_{k_{\mathcal{Z}}}$  is said to be *characteristic*. The MMD can then

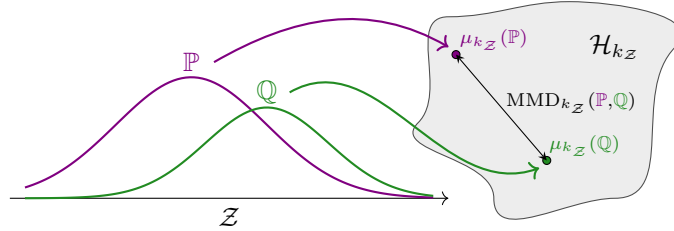


Figure 2.1: Kernel mean embedding

be used to define HSIC for the sensitivity analysis context, as proposed in [24].

Let  $\eta : \mathcal{U} \rightarrow \mathcal{Z}$  be a measurable model with respect to the Borel  $\sigma$ -algebras  $\mathcal{B}_{\mathcal{U}}$  and  $\mathcal{B}_{\mathcal{Z}}$ , where  $\mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_p \subset \mathbb{R}^p$ . Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a probability space on which each input  $U_i$  is a random variable of probability distribution  $\mathbb{P}_{U_i}$ . For any subset of indices  $A \subset \{1, \dots, p\}$ ,  $\mathbf{U}_A$  will denote the random vector  $(U_i)_{i \in A}$ . We are interested in knowing how the uncertainty of  $Z = \eta(\mathbf{U})$  can be attributed to the different inputs (or group of inputs). To do this, kernel-based sensitivity analysis relies on measuring the dependence between  $\mathbf{U}_A$  and  $Z$  by computing the MMD between the joint distribution  $\mathbb{P}_{\mathbf{U}_A, Z}$  and the product of the marginal distributions  $\mathbb{P}_{\mathbf{U}_A} \otimes \mathbb{P}_Z$ . If this distance is zero, then  $\mathbf{U}_A$  and  $Z$  are independent, i.e.  $\mathbf{U}_A$  has no effect on  $Z$ . This distance is called the Hilbert-Schmidt Independence Criterion (HSIC), first defined in [51].

**Definition 2.2.** For  $A \subset \{1, \dots, p\}$ , let  $k = k_A \otimes k_{\mathcal{Z}}$  be a kernel inducing the RKHS  $\mathcal{H}_k$ , with  $k_A$  a kernel on the space  $\mathcal{U}_A = \otimes_{i \in A} \mathcal{U}_i$ . The Hilbert-Schmidt Independence Criterion between  $\mathbf{U}_A$  and  $Z$ , denoted by  $\text{HSIC}(\mathbf{U}_A, Z)$ , is defined as

$$\text{HSIC}(\mathbf{U}_A, Z) := \text{MMD}(\mathbb{P}_{\mathbf{U}_A, Z}, \mathbb{P}_{\mathbf{U}_A} \otimes \mathbb{P}_Z)^2 = \|\mu_k(\mathbb{P}_{\mathbf{U}_A, Z}) - \mu_{k_A}(\mathbb{P}_{\mathbf{U}_A}) \otimes \mu_{k_{\mathcal{Z}}}(\mathbb{P}_Z)\|_{\mathcal{H}_k}^2.$$

HSIC can then be used to test the independence between  $\mathbf{U}_A$  and  $Z$ . For the test to be consistent, i.e.  $\text{HSIC}(\mathbf{U}_A, Z) = 0$  if and only if  $\mathbf{U}_A$  and  $Z$  are independent, it is sufficient that both  $k_A$  and  $k_{\mathcal{Z}}$  are characteristic (which doesn't necessarily mean that  $k$  is characteristic). This result can be obtained by combining Theorem 3.11 in [94] and Proposition 29 in [150] as explained in the introduction of [164]. To conduct screening, an independence test is performed for each input  $U_i$  :

$$(\mathcal{H}_0^i) : \text{HSIC}(U_i, Z) = 0 \quad \text{versus} \quad (\mathcal{H}_1^i) : \text{HSIC}(U_i, Z) > 0.$$

One of the reasons HSIC has become popular is that it can be expressed very simply in terms of kernel functions. From Lemma 1 of [51], one has:

$$\begin{aligned} \text{HSIC}(\mathbf{U}_A, Z) &= \mathbb{E}[k_A(\mathbf{U}_A, \mathbf{U}_A')k_Z(Z, Z')] \\ &\quad + \mathbb{E}[k_A(\mathbf{U}_A, \mathbf{U}_A')]\mathbb{E}[k_Z(Z, Z')] \\ &\quad - 2\mathbb{E}[\mathbb{E}[k_A(\mathbf{U}_A, \mathbf{U}_A')|\mathbf{U}_A]\mathbb{E}[k_Z(Z, Z')|Z]], \end{aligned} \quad (2.1)$$

where  $(\mathbf{U}_A', Z')$  is an independent copy of  $(\mathbf{U}_A, Z)$ . This expression makes estimating the HSIC very easy, since it can be estimated with biased or unbiased estimators based on U- and V-statistics as introduced in [51, 154]. If the sample size is large enough, asymptotic estimation of the p-value associated to the test of independence can be used, as suggested in [52]. For smaller samples, a permutation-based technique can be used (see [27]).

However, HSIC indices are not sufficient for ranking. The values  $\text{HSIC}(U_i, Z)$  must first be normalized in order to compare them and rank the inputs. In the context of sensitivity analysis, several normalizations have been studied, such as normalization by  $\sum_{i=1}^p \text{HSIC}(U_i, Z)$  [157] or by  $\sqrt{\text{HSIC}(U_i, U_i)\text{HSIC}(Z, Z)}$  [24]. These indices have been used for ranking purposes, but they are not as satisfactory as the ANOVA decomposition of Sobol' indices. However, a recent study provides an ANOVA-like decomposition that justifies the use of HSIC for ranking purposes [25].

### 2.2.2 HSIC-ANOVA indices

In [25] an ANOVA-like decomposition of the HSIC is proposed, which makes it usable to rank the inputs by influence. This decomposition requires strong assumptions, particularly the independence between the inputs and the ANOVA property of the input kernel.

**Definition 2.3** (Orthogonal and ANOVA kernel). *Let  $\mathcal{U}$  be a measurable space. A kernel  $k : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}$  is said to be orthogonal with respect to a probability measure  $\nu \in \mathcal{M}_1^+(\mathcal{U})$  if*

$$\forall u \in \mathcal{U}, \int_{\mathcal{U}} k(u, z) d\nu(z) = 0.$$

*A kernel  $K : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}$  is said to be ANOVA w.r.t.  $\nu$  if it can be decomposed as  $K = 1 + k$  where  $k$  is orthogonal w.r.t.  $\nu$ .*

**Theorem 2.1** (ANOVA decomposition of HSIC). *Assuming that:*

- i. *The inputs  $U_1, \dots, U_p$  are mutually independent;*
- ii. *Each input has an ANOVA kernel  $K_i$  w.r.t. the input distribution  $\mathbb{P}_{U_i}$ . For any group of inputs  $\mathbf{U}_A$  with  $A \subset \{1, \dots, p\}$ , the associated kernel is defined by*

$$K_A = \bigotimes_{i \in A} K_i;$$

- iii. *For any  $A \subset \{1, \dots, p\}$ ,  $\mathbb{E}[K_A(\mathbf{U}_A, \mathbf{U}_A)] < +\infty$  and  $\mathbb{E}[k_Z(Z, Z)] < +\infty$ .*

*Then the ANOVA decomposition of the HSIC is given by*

$$\text{HSIC}(\mathbf{U}, Z) = \sum_{A \subset \{1, \dots, p\}} \sum_{B \subset A} (-1)^{|A|-|B|} \text{HSIC}(\mathbf{U}_B, Z).$$

In addition to providing a decomposition of the HSIC, an ANOVA input kernel also simplifies the HSIC expression of the Equation 2.1, which becomes

$$\text{HSIC}(\mathbf{U}_A, Z) = \mathbb{E} \left[ (K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) k_Z(Z, Z') \right].$$

Given an independent and identically distributed (iid) sample  $(\mathbf{U}_A^{(1)}, Z^{(1)}), \dots, (\mathbf{U}_A^{(n)}, Z^{(n)})$  of  $(\mathbf{U}_A, Z)$ , the previous expression is easily estimated by a U-statistic

$$\widehat{\text{HSIC}}(\mathbf{U}_A, Z) = \frac{2}{n(n-1)} \sum_{i < j}^n \left( K_A \left( U_A^{(i)}, U_A^{(j)} \right) - 1 \right) k_Z \left( Z^{(i)}, Z^{(j)} \right). \quad (2.2)$$

The decomposition given in Theorem 2.1 allows SA indices to be defined in a similar way as Sobol' indices.

**Definition 2.4** (HSIC-ANOVA indices). *The first-order and total-order HSIC-ANOVA indices can be defined by*

$$\forall i \in \{1, \dots, p\} \quad S_i^{\text{HSIC}} := \frac{\text{HSIC}(U_i, Z)}{\text{HSIC}(\mathbf{U}, Z)} \quad \text{and} \quad S_{T_i}^{\text{HSIC}} := 1 - \frac{\text{HSIC}(\mathbf{U}_{-i}, Z)}{\text{HSIC}(\mathbf{U}, Z)},$$

with  $\mathbf{U}_{-i} = (U_1, \dots, U_{i-1}, U_{i+1}, \dots, U_d)$ . These indices can be generalized to groups of inputs.

These indices can be used to rank the inputs by influence by ranking their indices. It is also possible to use either first-order or total-order indices for screening, as we still have

$$S_i^{\text{HSIC}} = 0 \iff S_{T_i}^{\text{HSIC}} = 0 \iff U_i \perp \Gamma,$$

as shown in [145]. The authors also propose a test of independence that proves to be more powerful than the usual tests performed with HSIC indices.

It is important to note that the three assumptions of Theorem 2.1, and especially ii., restrict the choice of the input kernel to be ANOVA. Few kernels are known to be ANOVA. The best known are the Sobolev kernels (see [82, 25]), which correspond to the reproducing kernels of some Sobolev spaces. These kernels are ANOVA with respect to the uniform law on  $[0, 1]$  and are also characteristic (see remark 3.4 and Proposition 3.6 of [146]). It is also possible to derive ANOVA kernels from classic kernels. One of the possible transformations is suggested in [47]. For any given kernel  $k$ , its ANOVA counterpart  $K$  w.r.t.  $\nu$  is defined by

$$\begin{aligned} K(x, y) = 1 + k(x, y) - \int k(x, z) d\nu(z) - \int k(z, y) d\nu(z) \\ + \int \int k(z, z') d\nu(z) d\nu(z'). \end{aligned} \quad (2.3)$$

Given ANOVA and characteristic input kernels  $K_i$  and a characteristic output kernel  $k_Z$ , first-order and total-order HSIC-ANOVA indices can be computed and used for both screening and ranking. This approach has the strength of being very permissive about the nature of the output. Indeed, once one has a measurable and bounded characteristic kernel on an output space on which a random element can be defined, one can derive HSIC-ANOVA sensitivity indices to perform a sensitivity analysis of the complex model. We propose to do so in the next section dealing with an output space of sets.

## 2.3 Kernel-based Sensitivity Analysis for sets

In this section, we apply the methodology of the previous section to the case of set-valued outputs, i.e., where  $\mathcal{Z}$  is a space of sets. We introduce a probabilistic framework for random sets on which we propose a customized kernel and study its property so that it can be used to define HSIC-ANOVA indices for sets. We also introduce and study the estimation of these new indices. Basic concepts of functional analysis and topology useful in this section can be found, for example, in [140].

### 2.3.1 Probabilistic framework of random sets

A general theory of random sets can be found in [105]. In our case, however, we propose to define a random set as  $\Gamma = \eta(\mathbf{U})$ , where  $\eta$  is a measurable function from  $\mathcal{U}$  to a space of sets to be defined. First, we define the space of sets we are working in, and a corresponding  $\sigma$ -algebra.

Let  $\mathcal{L}(\mathcal{X})$  be the space of all Lebesgue-measurable subsets of a compact space  $\mathcal{X} \subset \mathbb{R}^d$ . Let  $\delta : \mathcal{L}(\mathcal{X}) \times \mathcal{L}(\mathcal{X}) \rightarrow \mathbb{R}$  be the Lebesgue measure of the symmetric difference defined by  $\delta(\gamma_1, \gamma_2) = \lambda(\gamma_1 \Delta \gamma_2)$  where  $\Delta$  is the symmetric difference defined by  $\gamma_1 \Delta \gamma_2 = (\gamma_1 \cup \gamma_2) \setminus (\gamma_1 \cap \gamma_2)$  and  $\lambda$  is the Lebesgue measure on  $\mathcal{X}$ .  $L^2(\mathcal{X})$  denotes the space of square Lebesgue integrable functions on  $\mathcal{X}$ , and the  $L^2$ -norm is denoted by  $\|\cdot\|_{L^2(\mathcal{X})}$ .  $\delta$  can also be seen in terms of the  $L^2$ -norm of the indicator functions of  $\gamma_1$  and  $\gamma_2$ , as given in the following lemma.

**Lemma 2.1.** *Let  $\gamma_1, \gamma_2 \in \mathcal{L}(\mathcal{X})$ , we have*

$$\delta(\gamma_1, \gamma_2) = \|\mathbb{1}_{\gamma_1} - \mathbb{1}_{\gamma_2}\|_{L^2(\mathcal{X})}^2,$$

and

$$\delta(\gamma_1, \gamma_2) = 0 \Leftrightarrow \gamma_1 = \gamma_2 \text{ } \lambda\text{-almost everywhere.}$$

Thus  $\delta$  is a pseudo-metric in  $\mathcal{L}(\mathcal{X})$ , since  $\delta(\gamma_1, \gamma_2) = 0$  only implies that  $\gamma_1 = \gamma_2$   $\lambda$ -almost everywhere. To work in a metric space, we quotient  $\mathcal{L}(\mathcal{X})$  by the equivalence relation  $\delta(\gamma_1, \gamma_2) = 0$ . The resulting quotient space is denoted by  $\mathcal{L}^*(\mathcal{X})$ .  $\delta$  remains well defined on  $\mathcal{L}^*(\mathcal{X}) = \mathcal{L}^*(\mathcal{X}) \times \mathcal{L}^*(\mathcal{X})$  and becomes a distance sometimes called the Fréchet-Nikodym-Aronszajn metric [96]. With this distance,  $\mathcal{L}^*(\mathcal{X})$  can be provided with the Borel  $\sigma$ -algebra  $\mathcal{B}_\Gamma := \mathcal{B}(\mathcal{L}^*(\mathcal{X}), \delta)$  to make it a measurable space. A random set is then simply defined as  $\Gamma = \eta(\mathbf{U})$  where  $\eta : \mathcal{U} \rightarrow \mathcal{L}^*(\mathcal{X})$  is measurable with respect to  $\mathcal{B}_\mathbf{U}$  and  $\mathcal{B}_\Gamma$ . The probability distribution of a random set  $\Gamma$  is then defined as the push-forward probability measure of  $\mathbb{P}_\mathbf{U}$  through  $\eta$ .

Examples of random set are numerous in industrial applications, especially in the context of optimization or inversion, where determining the feasible set of solutions is crucial. In the presence of uncertainties, this feasible set is random and can be called an excursion set as given in the following definition.

**Definition 2.5.** *Let  $g : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$  be a measurable function. The excursion set  $\Gamma_g$  associated with the constraint  $g \leq 0$  is defined by*

$$\Gamma_g = \{x \in \mathcal{X}, \quad g(x, \mathbf{U}) \leq 0\}.$$

### 2.3.2 A kernel between sets

Now that we have a sound framework of random sets, we want to follow the methodology of Section 2.2 but with  $\mathcal{Z} = \mathcal{L}^*(\mathcal{X})$  to define HSIC indices for sets. To do this, we first need a kernel on sets,

i.e. a symmetric and positive definite function  $k : \mathcal{L}^*(\mathcal{X}) \times \mathcal{L}^*(\mathcal{X}) \rightarrow \mathbb{R}$ . We propose the function  $k_{set}$  defined by

$$\forall \gamma_1, \gamma_2 \in \mathcal{L}^*(\mathcal{X}), \quad k_{set}(\gamma_1, \gamma_2) := e^{-\frac{\lambda(\gamma_1 \Delta \gamma_2)}{2\sigma^2}},$$

where  $\sigma$  is a positive scalar.  $k_{set}$  is inspired by the classic Gaussian kernel, since the Lebesgue measure of the symmetric difference is equal to the  $L^2$ -norm of the difference between two indicator functions, as shown in Lemma 2.1. This function was shown to be positive definite in [8] when the space of sets is an indexing collection, but the proof is similar in our case. Moreover,  $k_{set}$  is bounded and measurable.

**Proposition 2.1.** *The function  $k_{set} : \mathcal{L}^*(\mathcal{X}) \times \mathcal{L}^*(\mathcal{X}) \rightarrow \mathbb{R}$  defined by*

$$\forall \gamma_1, \gamma_2 \in \mathcal{L}^*(\mathcal{X}), \quad k_{set}(\gamma_1, \gamma_2) = e^{-\frac{\lambda(\gamma_1 \Delta \gamma_2)}{2\sigma^2}},$$

*is symmetric and positive definite for any positive scalar  $\sigma$ , which means that  $k_{set}$  is a kernel.*

**Proposition 2.2.** *The kernel  $k_{set}$  is bounded and measurable from  $\mathcal{L}_2^*(\mathcal{X}) = \mathcal{L}^*(\mathcal{X}) \times \mathcal{L}^*(\mathcal{X})$  to  $\mathbb{R}$ .*

As in Definition 2.1, this allows to define the kernel mean embedding  $\mu_{k_{set}}(\mathbb{P}_\Gamma)$  of any distribution  $\mathbb{P}_\Gamma$  describing the behaviour of a random set  $\Gamma$ . We then show that it is injective, i.e.  $k_{set}$  is characteristic.

**Proposition 2.3.** *The kernel  $k_{set}$  is characteristic.*

We now have all the ingredients to use HSIC-ANOVA indices for set-valued outputs. We can now screen the inputs through independence testing and rank them according to their influence on the random set  $\Gamma$  by using either first-order or total-order HSIC-ANOVA indices, denoted here by  $S_i^{H_{set}}$  and  $S_{T_i}^{H_{set}}$ . HSIC-ANOVA indices applied to sets provide a solution for the screening and ranking of the inputs of a set-valued model. However, to achieve these purposes, they must first be estimated, but the presence of sets raises some difficulties.

### 2.3.3 Estimation of HSIC-ANOVA indices for sets

As given in Equation 2.2,  $\text{HSIC}(U_A, \Gamma)$  can be estimated by

$$\widehat{\text{HSIC}}(U_A, \Gamma) = \frac{2}{n(n-1)} \sum_{i < j}^n \left( K_A(U_A^{(i)}, U_A^{(j)}) - 1 \right) k_{set}(\Gamma^{(i)}, \Gamma^{(j)}),$$

where  $(U_A^{(i)}, \Gamma^{(i)}), i = 1, \dots, n$  is an iid sample of  $(U_A, \Gamma)$ .

For set-valued models, it is common to have access only to the knowledge of whether a given point  $x \in \mathcal{X}$  is in an output set  $\Gamma$  or not. In terms of estimation, this means that the Lebesgue measure of the symmetric difference and thus  $k_{set}$  must be estimated.

For two random sets  $(\Gamma^{(i)}, \Gamma^{(j)})$ ,  $k_{set}(\Gamma^{(i)}, \Gamma^{(j)}) = e^{-\frac{\lambda(\Gamma^{(i)} \Delta \Gamma^{(j)})}{2\sigma^2}}$  can be written as

$$k_{set}(\Gamma^{(i)}, \Gamma^{(j)}) = \exp \left( -\frac{\lambda(\mathcal{X})}{2\sigma^2} \mathbb{E}[\mathbb{1}_{\Gamma^{(i)} \Delta \Gamma^{(j)}}(\mathbf{X}) | (\Gamma^{(i)}, \Gamma^{(j)})] \right),$$

where  $\mathbf{X} \sim \mathcal{U}(\mathcal{X})$  using that  $\lambda(\Gamma) = \lambda(\mathcal{X}) \mathbb{E}_{\mathbf{X} \sim \mathcal{U}(\mathcal{X})}[\mathbb{1}_\Gamma(\mathbf{X}) | \Gamma]$ . Then, given an iid sample  $(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)})$  of  $\mathbf{X}$ , we can estimate  $k_{set}(\Gamma^{(i)}, \Gamma^{(j)})$  by

$$\widehat{k_{set}}(\Gamma^{(i)}, \Gamma^{(j)}) = \exp \left( -\frac{\lambda(\mathcal{X})}{2\sigma^2} \frac{1}{m} \sum_{k=1}^m \mathbb{1}_{\Gamma^{(i)} \Delta \Gamma^{(j)}}(\mathbf{X}^{(k)}) \right).$$

We now need to plug this estimator into our previous one, resulting in a Nested Monte Carlo (NMC) estimator. To be compatible with the framework of NMC estimator, it is necessary for samples of  $\mathbf{X}$  to be drawn independently for every pair of indices  $(i, j)$  within the outer loop. Let  $(\mathbf{X}_{i,j}^{(k)})$  be an iid sample of  $\mathbf{X} \sim \mathcal{U}(\mathcal{X})$ , with  $k \in \{1, \dots, m\}$  and  $(i, j) \in \{1, \dots, n\}^2$  s.t.  $i < j$ . Including the previous estimation of the kernel  $k_{set}$ , the NMC estimator of  $\text{HSIC}(\mathbf{U}_A, \Gamma)$  becomes

$$\widehat{\text{HSIC}}^{nest}(\mathbf{U}_A, \Gamma) = \frac{2}{n(n-1)} \sum_{i < j}^n \left( K_A(\mathbf{U}_A^{(i)}, \mathbf{U}_A^{(j)}) - 1 \right) e^{-\frac{\lambda(\mathcal{X})}{2\sigma^2} \frac{1}{m} \sum_{k=1}^m \mathbb{1}_{\Gamma^{(i)} \Delta \Gamma^{(j)}}(\mathbf{X}_{i,j}^{(k)})}. \quad (2.4)$$

Note that this estimator is biased like any NMC estimator (see [132]). For each pair  $(\Gamma^{(i)}, \Gamma^{(j)})$ , the previous estimator requires checking whether  $\mathbf{X}_{i,j}^{(k)} \in \Gamma^{(i)} \Delta \Gamma^{(j)}$  for each  $k \in \{1, \dots, m\}$ . Each of these checks requires checking whether  $\mathbf{X}_{i,j}^{(k)} \in \Gamma^{(i)}$  and  $\mathbf{X}_{i,j}^{(k)} \in \Gamma^{(j)}$ , corresponding in the example of excursion sets, to compute  $g(\mathbf{X}_{i,j}^{(k)}, \mathbf{U}_A^{(i)})$ . This means  $n(n-1)m$  evaluations to estimate the index, which is not affordable. To solve this problem, we propose to reuse the same  $\mathbf{X}^{(k)}$  for each  $(i, j)$ . By doing so, we only need to test whether  $\mathbf{X}^{(k)} \in \Gamma^{(i)}$  for each  $k$  and  $i$ , reducing the number of evaluations to  $nm$ . The estimator is then given by

$$\widehat{\widehat{\text{HSIC}}}(\mathbf{U}_A, \Gamma) = \frac{2}{n(n-1)} \sum_{i < j}^n \left( K_A(\mathbf{U}_A^{(i)}, \mathbf{U}_A^{(j)}) - 1 \right) e^{-\frac{\lambda(\mathcal{X})}{2\sigma^2} \frac{1}{m} \sum_{k=1}^m \mathbb{1}_{\Gamma^{(i)} \Delta \Gamma^{(j)}}(\mathbf{X}^{(k)})}.$$

By simulating a single  $m$ -sample of  $\mathbf{X}$ ,  $\widehat{\widehat{\text{HSIC}}}(\mathbf{U}_A, \Gamma)$  is no longer a classic NMC estimator. Nevertheless, we show that its quadratic risk goes to 0 and we give an upper bound.

**Proposition 2.4.** *With the previous notations, we have*

$$\mathbb{E} \left( \widehat{\widehat{\text{HSIC}}}(\mathbf{U}_A, \Gamma) - \text{HSIC}(\mathbf{U}_A, \Gamma) \right)^2 \leq 2 \left( \frac{2\sigma_1^2}{n(n-1)} + \frac{4(n-2)\sigma_2^2}{n(n-1)} + \frac{L^2\sigma_3^2}{m} \right),$$

where

- $\sigma_1^2 = \text{Var}((K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) k_{set}(\Gamma, \Gamma'))$ ,
- $\sigma_2^2 = \text{Var}(\mathbb{E}[(K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) k_{set}(\Gamma, \Gamma') | (\Gamma, \mathbf{U}_A)])$ ,
- $\sigma_3^2 = \mathbb{E}[(K_A(\mathbf{U}_A, \mathbf{U}_A') - 1)^2 \text{Var}(\mathbb{1}_{\Gamma \Delta \Gamma'}(\mathbf{X}) | (\mathbf{U}_A, \mathbf{U}_A', \Gamma, \Gamma'))]$ ,
- $L = \frac{\lambda(\mathcal{X})}{2\sigma^2}$ ,

and where  $(\mathbf{U}_A', \Gamma')$  is an independent copy of  $(\mathbf{U}_A, \Gamma)$ .

The quadratic risk has a rate of  $\mathcal{O}(\frac{1}{n} + \frac{1}{m})$ , which tends to say that we should use  $n = m$ . In the case of the classic NMC estimator (see [131]), corresponding here to not reusing the same samples of  $\mathbf{X}$ , the rate is  $\mathcal{O}(\frac{1}{n} + \frac{1}{m^2})$  as shown in the Supplementary Materials. However, even if the convergence rate is better, the number of evaluations required is  $(n-1)$  times larger, which makes our choice to use the same sample of  $\mathbf{X}$  more efficient. Since the previous result is only an upper bound, we can also hope to get closer to a  $\mathcal{O}(\frac{1}{n} + \frac{1}{m^2})$  convergence rate in the application tests. That is, we can expect to have achieved convergence of our estimator without having to take high  $n$  and  $m$ . This will be highlighted in the numerical results presented in the next section. By plugging the estimator  $\widehat{\widehat{\text{HSIC}}}$ , we denote  $\hat{S}_i^{\text{H}_{set}}$  and  $\hat{S}_{T_i}^{\text{H}_{set}}$  the estimators of the first-order and total-order HSIC ANOVA indices on sets.

## 2.4 Numerical Experiments

In this part, our goal is to quantify the influence of the inputs  $\mathbf{U}$  on different excursion sets  $\Gamma_g$  (defined in Definition 2.5). To do this, we will consider three test cases. We first consider an analytically known function  $g$  defined on  $\mathcal{X} \subset \mathbb{R}^2$  and  $\mathcal{U} \subset \mathbb{R}^2$ , borrowed from [33]. In this first toy case, we will also study numerically the quadratic risk bound given in Proposition 2.4. The second test case is related to an optimization problem with a stationary harmonic oscillator, from [22], on which we want to quantify the impact of some uncertain inputs on the feasible sets. In the last case [137], a sensitivity analysis is performed in the context of a bi-objective robust optimization of a permanent magnet-assisted synchronous reluctance machine for electrical vehicle application. For each example, our sensitivity analysis is performed in two steps:

- Screening: We compute the p-values associated with the test  $\text{HSIC}(U_i, \Gamma_g) = 0$  versus  $\text{HSIC}(U_i, \Gamma_g) > 0$ . We use permutation based estimation. If the p-value is greater than a risk  $\alpha$ , the input is negligible, and influential otherwise. We use  $\alpha = 0.05$  but this value can be changed depending on the application.
- Ranking: We compute the first-order indices  $\hat{S}_i^{\text{H}_{set}}$  and the total-order indices  $\hat{S}_{T_i}^{\text{H}_{set}}$  of all inputs. The first-order indices  $\hat{S}_i^{\text{H}_{set}}$  are used to rank the inputs. With the total-order indices  $\hat{S}_{T_i}^{\text{H}_{set}}$  we can quantify the HSIC interaction effects.

The numerical implementation of the indices is done in the R language, using the `sensitivity` package and in particular the `sensiHSIC` and `testHSIC` functions, which allows to compute HSIC-ANOVA indices and p-values.

### 2.4.1 Excursion sets of a toy function

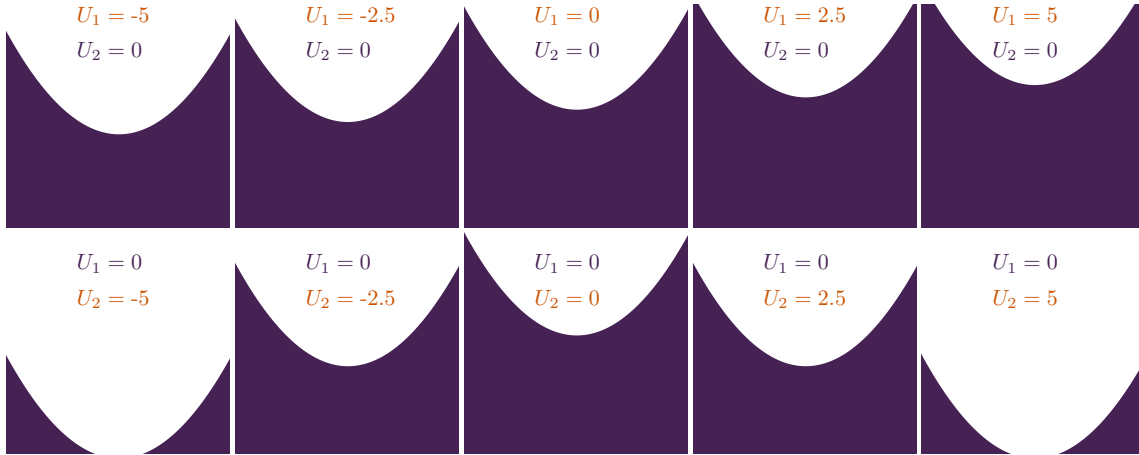
In this part, we will estimate the previous HSIC-ANOVA indices on the excursion set  $\Gamma_g$  defined by the following function  $g$  from [33],

$$\forall \mathbf{x}, \mathbf{u} \in [-5, 5]^2 \times [-5, 5]^2 \quad g(\mathbf{x}, \mathbf{u}) = -x_1^2 + 5x_2 - u_1 + u_2^2 - 1.$$

Visual conjecture (Figure 2.2) shows that  $u_2$  seems to induce more changes in  $\Gamma_g$  than  $u_1$ . To verify this assumption, we compute the p-values and the indices  $\hat{S}_i^{\text{H}_{set}}$ ,  $\hat{S}_{T_i}^{\text{H}_{set}}$  for each input  $U_i \sim \mathcal{U}([-5, 5])$  for  $i \in \{1, 2, 3\}$ , where  $U_3$  is an additional dummy input that does not appear in the function  $g$ . We first use  $n = m = 100$  and then  $n = m = 1000$ . The hyperparameter  $\sigma^2$  of the kernel  $k_{set}$  is chosen to be equal to the empirical mean of  $\lambda(\Gamma_g^{(i)} \Delta \Gamma_g^{(j)})$  with  $i > j$ . We compute the indices for five characteristic ANOVA kernels. The first is the Sobolev kernel  $K_{Sob}$  of order 1, defined by:

$$K_{Sob}(x, y) = 1 + \left(x - \frac{1}{2}\right) \left(y - \frac{1}{2}\right) + \frac{1}{2} \left[ (x - y)^2 - |x - y| + \frac{1}{6} \right].$$

The four others are obtained using the transformation given in (2.3) based on four classic kernels: the Gaussian kernel, the Laplace kernel and the Matérn 3/2 and 5/2. For these kernels, the previous transformation is known analytically and can be found in the Appendix of [47]. The hyperparameter is taken to be equal to the empirical standard deviation of the inputs. Since the inputs are not uniformly distributed on  $[0, 1]$ , we apply the inverse of the cumulative distribution function, as suggested in [25]. We repeat the estimation 20 times over 20 independent samples. The results are shown in Figure 2.3 for  $n = m = 100$  and in Figure 2.4 for  $n = m = 1000$ . The acceptance rates of the independence hypothesis are given in the table on the left side of the figure, and the variability of the indices is visualized in the boxplots on the right side.



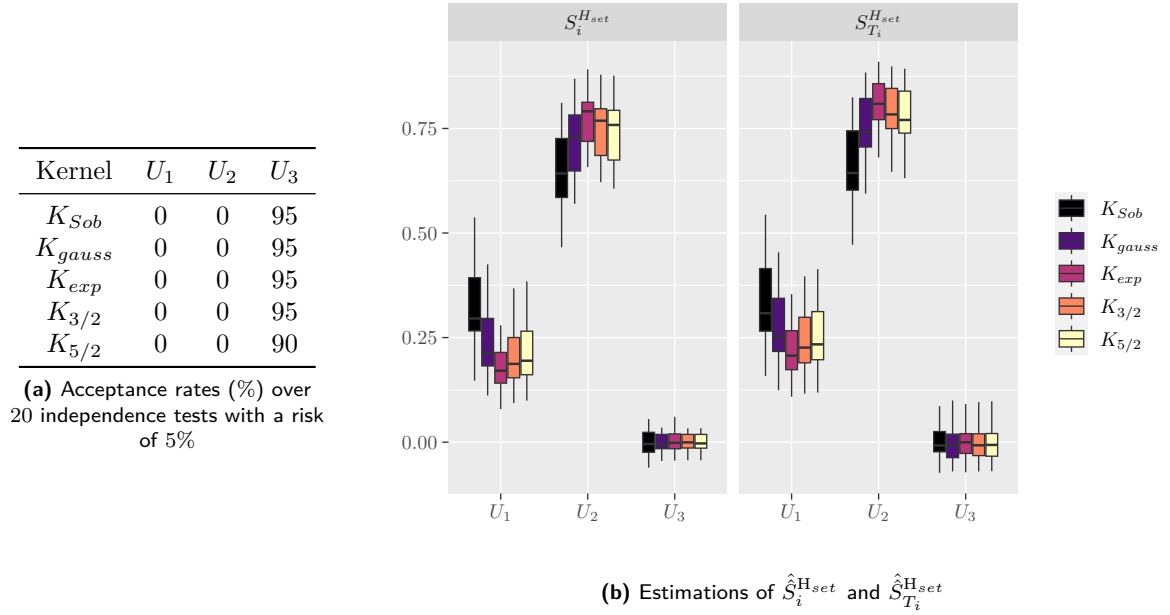
**Figure 2.2:** Excursion set of the constraint  $g \leq 0$  for  $U_1 \in \{-5, -2.5, 0, 2.5, 5\}$  and  $U_2 = 0$  (first row) and for  $U_1 = 0$  and  $U_2 \in \{-5, -2.5, 0, 2.5, 5\}$ . (second row)

With these results, we classify the inputs as influential or negligible (screening) and rank them by influence (ranking). 100% of the p-values of  $U_1$  and  $U_2$  are below the threshold of 0.05, so they are classified as influential.  $U_3$  has a p-value greater than 0.05 around 95% of the time, so it is correctly identified as negligible by the independence test. The first-order and total-order indices give the expected results that  $U_2$  has a greater influence than  $U_1$  on the excursion sets. More precisely, it shows that  $U_2$  alone explains about 70% of  $\text{HSIC}(\mathbf{U}, \Gamma_g)$  (depending on the input kernel), while  $U_1$  explains about 25%. The artificial input  $U_3$  is responsible for 0%, as expected. The rest is some kind of interaction between  $U_1$  and  $U_2$ . This is confirmed by looking at the total-order indices, whose values are a bit greater than the first-order. However, this part of the interaction is not to be interpreted as in the case of the Sobol' indices. It is actually an open question to interpret what kind of interactions are detected by the HSIC-ANOVA indices. This is studied in [146] in the case of Sobolev kernels. The effect of the input kernel on the HSIC-ANOVA indices is also an open question, and in this toy case we can say that the input kernel has a non-negligible effect on the estimates of the p-values and the indices. However, in this case, it does not change the results of the screening and the ranking. It also seems that the Sobolev kernel behaves differently from the other four kernels.

Taking  $n = m = 1000$  in Figure 2.4 reduces the variance of the indices, but was not necessary to screen and rank the inputs. To support this comment, we compute the relative quadratic risk of the estimator  $\widehat{\widehat{\text{HSIC}}}(U_1, \Gamma_g)$  defined by:

$$\mathcal{R}(\widehat{\widehat{\text{HSIC}}}(U_1, \Gamma_g)) = \mathbb{E} \left( \frac{\widehat{\widehat{\text{HSIC}}}(U_1, \Gamma_g) - \text{HSIC}(U_1, \Gamma_g)}{\text{HSIC}(U_1, \Gamma_g)} \right)^2.$$

We also compute the associated upper bound given in Proposition 2.4 and the upper bound for a classic NMC estimator (with independent  $\mathbf{X}_{i,j}^{(k)}$  for all  $1 \leq i, j \leq n$ , see Equation 2.4). The "true" value of HSIC, used to compute the quadratic risk, and the constants  $\sigma_1$ ,  $\sigma_2$ , and  $\sigma_3$  are computed for  $n = m = 3000$ . The Sobolev kernel is used as the input kernel. We plot the risk and the two bounds in the Figure 2.5 from  $n = m = 30$  to  $n = m = 500$ . We observe, as expected, that the quadratic risk of the estimator is below the upper bound of Proposition 2.4, but also below the NMC upper bound we would have by taking independent  $\mathbf{X}_{i,j}^{(k)}$ . This example confirms that we



**Figure 2.3:** Acceptance rates (a) and estimation of  $\hat{S}_i^{Hset}$  and  $\hat{S}_{T_i}^{Hset}$  (b) for the excursion set  $\Gamma_g$  computed for 5 kernels with  $n = 100$ ,  $m = 100$  and repeated 20 times

have not lost too much in terms of variance of the estimator by reusing the same  $\mathbf{X}^{(k)}$ . It also seems that we do not necessarily need to use high values of  $n = m$ . Therefore, in the next example, we will only compute the indices for  $n = m = 100$ .

## 2.4.2 Excursion sets of the optimization of an oscillator

In [22], an optimization is performed with three probabilistic constraints. We consider the first two constraints, which are defined by the following functions :

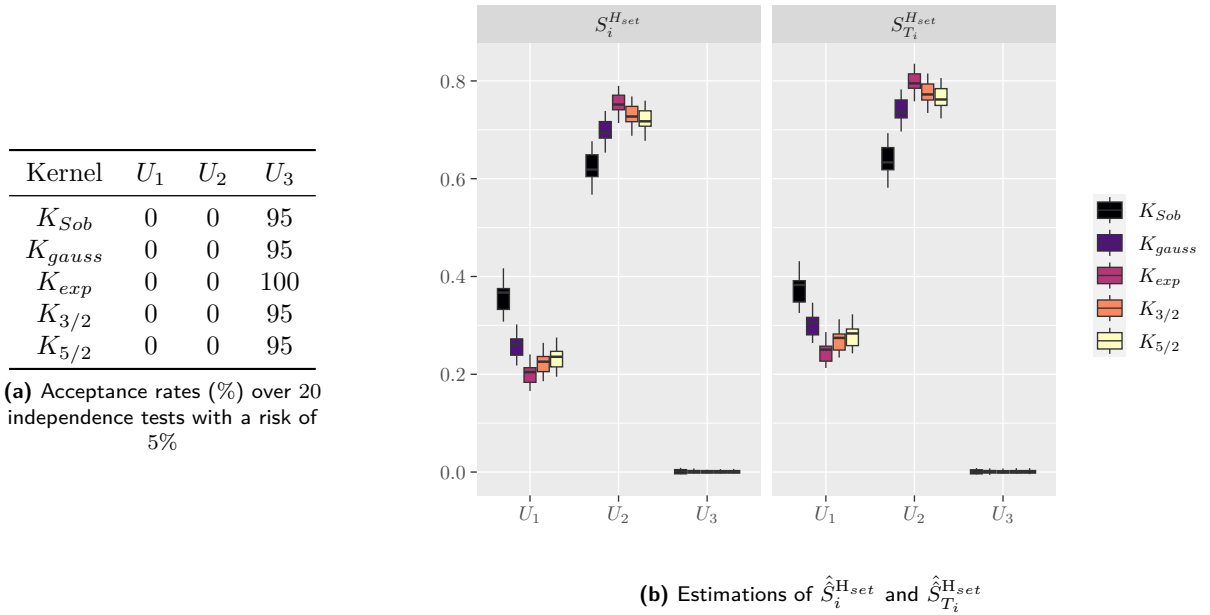
$$g_1(x_1, x_2, u_1, u_2, u_p, u_{r_1}, u_{r_2}) = u_{r_1} - \max_{t \in [0, T]} \mathcal{Y}'(x_1 + u_1, x_2 + u_2, u_p; t),$$

$$g_2(x_1, x_2, u_1, u_2, u_p, u_{r_1}, u_{r_2}) = u_{r_2} - \max_{t \in [0, T]} \mathcal{Y}''(x_1 + u_1, x_2 + u_2, u_p; t),$$

where  $\mathcal{Y}(x_1 + u_1, x_2 + u_2, u_p; t)$  is the solution of the harmonic oscillator defined by:

$$(x_1 + u_1)\mathcal{Y}''(t) + u_p\mathcal{Y}'(t) + (x_2 + u_2)\mathcal{Y}(t) = \eta(t).$$

The deterministic input domain is  $\mathcal{X} = [1, 5] \times [20, 50]$ . The uncertain input probability distributions are given in Table 2.1.  $U_{r_3}$  is initially a random input associated with a third constraint. In our case it will play the role of a dummy input to check if it is recognized as negligible. We study the impact of the uncertain inputs on the excursion sets  $\Gamma_{g_1}$  associated with the constraint  $g_1 \leq 0$  and on the excursion sets  $\Gamma_{g_2}$  associated with  $g_2 \leq 0$ . Since kernel-based methods are appropriate for vector outputs, we also consider the case where an output is the pair of the two excursion sets  $(\Gamma_{g_1}, \Gamma_{g_2})$ , each associated with a constraint. For each case, we compute  $\hat{S}_i^{Hset}$ ,  $\hat{S}_{T_i}^{Hset}$  and the associated p-values for each uncertain input with  $n = 100$  and  $m = 100$ . We again use the output kernel  $k_{set}$  and  $k_{set} \otimes k_{set}$  for the pairs of excursion sets. The same 5 ANOVA kernels are used as



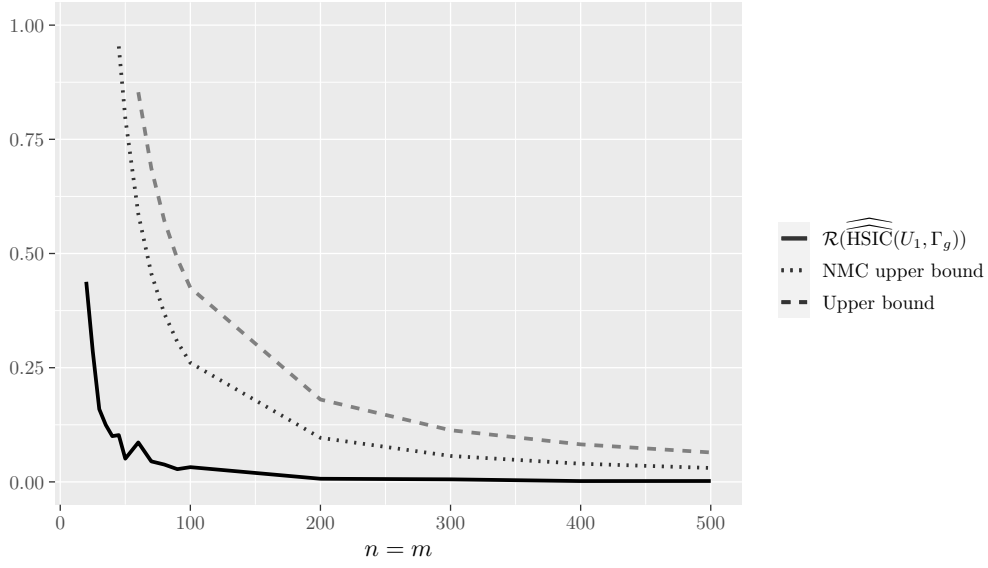
**Figure 2.4:** Acceptance rates (a) and estimation of  $\hat{S}_i^{Hset}$  and  $\hat{S}_{T_i}^{Hset}$  (b) for the excursion set  $\Gamma_g$  computed for 5 kernels with  $n = 1000$ ,  $m = 1000$  and repeated 20 times

**Table 2.1:** Definition of the uncertain inputs

| Uncertainty | Distribution             | Uncertainty | Distribution               |
|-------------|--------------------------|-------------|----------------------------|
| $U_1$       | $\mathcal{U}[-0.3, 0.3]$ | $U_{r_1}$   | $\mathcal{N}(1, 0.1^2)$    |
| $U_2$       | $\mathcal{U}[-1, 1]$     | $U_{r_2}$   | $\mathcal{N}(2.5, 0.25^2)$ |
| $U_p$       | $\mathcal{U}[0.5, 1.5]$  | $U_{r_3}$   | $\mathcal{N}(15, 3^2)$     |

in the previous example. We repeat the estimation 20 times to again obtain the acceptance rates and boxplots of the first-order indices given in Figures 2.6 to 2.8. The total-order indices are given in the Supplementary Materials.

Acceptance rates tables can be used for screening and the boxplots for ranking. For the random set associated with the constraint  $g_1 \leq 0$  (Figure 2.6), only  $U_{r_1}$  is always detected as influential.  $U_1$  is also detected as influential most of the time (with an acceptance rate of about 20%), and  $U_p$  is detected as influential only 40% of the time. Between these three inputs,  $U_{r_1}$  is much more influential, since  $\hat{S}_{r_1}^{Hset} \approx 80\%$ . Then  $U_1$  and  $U_p$  are responsible for about 10% and 3% of  $\text{HSIC}(\mathbf{U}, \Gamma_{g_1})$ . For the second constraint  $g_2 \leq 0$  (Figure 2.7), only  $U_1$  and  $U_{r_2}$  are influential with their first-order index of about 10% and 80%, respectively. If we consider the pair of the two random sets (Figure 2.8), we get a kind of compromise between the two previous cases:  $U_1$ ,  $U_{r_1}$  and  $U_{r_2}$  are always recognized as influential. In terms of ranking,  $U_{r_1}$  and  $U_{r_2}$  have almost the same influence, with their first-order index around 40% and  $U_1$  remaining around 10%. It is important to note that considering pairs of random sets is completely different from considering the random set associated with the pair of constraints, denoted  $\Gamma_{(g_1, g_2)}$ , i.e., the intersection of the two sets. The results of the latter are shown in the Supplementary Materials and are indeed very different from the results in Figure 2.8. In this test case, we observe that the choice of the input kernel has a limited impact



**Figure 2.5:** Evolution of  $\widehat{\mathcal{R}}(\widehat{\text{HSIC}}(U_1, \Gamma_g))$  and of the associated upper bounds for the excursion set  $\Gamma_g$

on the indices. Regardless of the stage of screening or ranking, the conclusions remain the same for the five kernels.

### 2.4.3 Sensitivity analysis for robust conception of an electrical machine

In [137], a sensitivity analysis is performed in the context of robust conception of an electrical machine. The studied machine is a permanent magnet assisted synchronous reluctance motor which is one of the most used machines nowadays in electrical vehicles. The purpose of the authors is to take into account the uncertainties (manufacturing and assembly tolerances) on geometric and magnetic parameters of the machine components in the optimization of the mean torque (to be maximized) and the torque ripples (to be minimized). The resulting optimization problem is the following

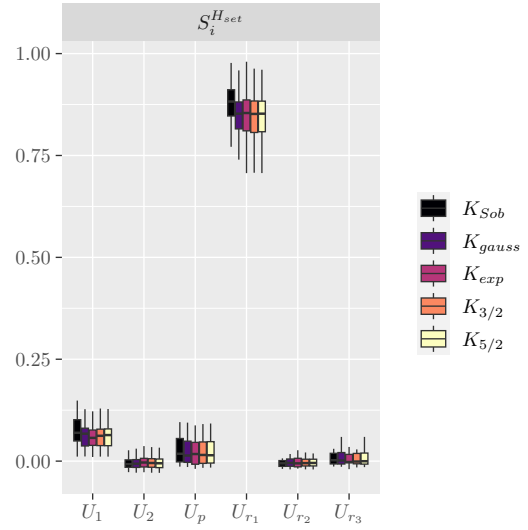
$$\min_{x \in \mathcal{X}} (\mathbb{E}[f_1(x, \mathbf{U})], \mathbb{E}[f_2(x, \mathbf{U})])$$

where  $f_1$  and  $f_2$  are two real-valued objective functions defined on  $\mathcal{X} \subset \mathbb{R}^{12}$  (respectively the opposite of the mean torque and the torque ripple) and  $\mathbf{U} \subset \mathbb{R}^{14}$ . 12 of the 14 uncertain inputs are manufacturing tolerances on each  $x$ , summarized in Table 2.2, where, for example,  $\pm 0.1^\circ$  means that the slot angle uncertainty  $U_1$  follows a uniform law on  $[-0.1, 0.1]$ . The other two,  $U_{13}$  and  $U_{14}$ , describe the magnetic material properties and follow uniform distributions on  $[-1, 1]$ .

The purpose of our study is to screen and rank the uncertain parameters according to their impact on the quantities of interest in this optimization. Since we want to minimize an objective function, we are interested in quantifying the effect of  $\mathbf{U}$  on sets where the objective function takes low values. This can be done by looking at the random set of the form  $\Gamma_{f_1} = \{x \in \mathcal{X}, f_1(x, \mathbf{U}) \leq q_1\}$ , where  $q_1$  is a threshold to be selected. It quantifies the effect of  $\mathbf{U}$  on the sets where  $f_1$  is below  $q_1$ , but does not take into account the variations of  $f_1$  within this low-valued region. For this reason, we propose to look at the effect of the uncertain inputs on the set  $\Gamma_{F_1}$  defined by  $\Gamma_{F_1} = \{(x, x_{13}) \in \mathcal{X} \times [f_1^{\min}, q_1], x_{13} \leq f_1(x, \mathbf{U})\}$  where  $f_1^{\min}$  is a lower bound to only consider sets of low values of  $f_1$ . This also corresponds to the excursion set associated with  $F_1 : \mathcal{X} \times [f_1^{\min}, q_1] \times \mathcal{U} \rightarrow \mathbb{R}$  defined

| Kernel      | $U_1$ | $U_2$ | $U_p$ | $U_{r_1}$ | $U_{r_2}$ | $U_{r_3}$ |
|-------------|-------|-------|-------|-----------|-----------|-----------|
| $K_{Sob}$   | 15    | 95    | 55    | 0         | 100       | 90        |
| $K_{gauss}$ | 15    | 100   | 60    | 0         | 100       | 95        |
| $K_{exp}$   | 25    | 90    | 60    | 0         | 100       | 95        |
| $K_{3/2}$   | 15    | 90    | 60    | 0         | 100       | 95        |
| $K_{5/2}$   | 15    | 100   | 60    | 0         | 100       | 95        |

(a) Acceptance rates (%) over 20 independence tests with a risk of 5%



(b) Estimations of  $\hat{S}_i^{Hset}$

**Figure 2.6:** Acceptance rates (a) and estimation of  $\hat{S}_i^{Hset}$  (b) for the excursion set  $\Gamma_{g_1}$  computed for 5 kernels with  $n = 100$ ,  $m = 100$  and repeated 20 times

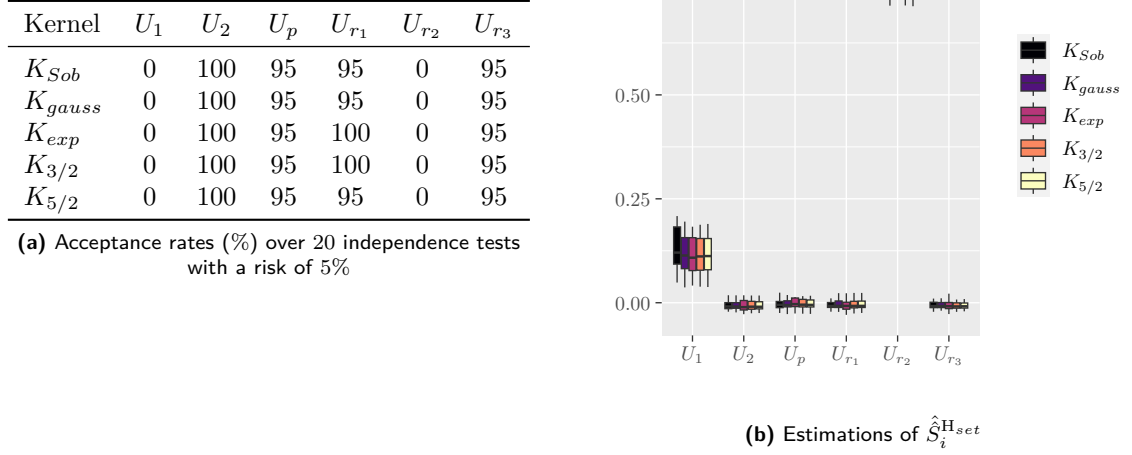
by  $F_1(\mathbf{x}, \mathbf{u}) = x_{13} - f_1(\mathbf{x}_{-13}, \mathbf{u})$ .  $\Gamma_{F_2}$  is defined similarly.

We study three different cases : sensitivity analysis on  $\Gamma_{F_1}$ , on  $\Gamma_{F_2}$ , and on the pair of excursion sets  $(\Gamma_{F_1}, \Gamma_{F_2})$ . We use the thresholds  $-q_1 = 420$  N.m and  $q_2 = 7\%$  where lies the Pareto front in Figure 9 of [137]. For these three cases, we take  $n = m = 100$  and use only the Sobolev kernel. With 20 replicates, the acceptance rates with a risk of 0.05% are given in Table 2.3 and the boxplots of the first-order indices are plotted in Figure 2.10.

Table 2.3 shows that the inputs influencing the excursion sets  $\Gamma_{F_1}$  and  $\Gamma_{F_2}$  are different. Only  $U_{13}$  and  $U_{14}$  are always tested as influential for  $\Gamma_{F_1}$  and  $U_2, U_3, U_5$  and  $U_{13}$  for  $\Gamma_{F_2}$ . They correspond to the most influential inputs in Figure 2.10 and can be ranked. For example,  $U_2$  has the largest influence on  $\Gamma_{F_2}$ , followed by  $U_{13}$ ,  $U_3$  and  $U_5$ . Detected about 95% of the time as independent of the output,  $U_2, U_4, U_6, U_7, U_9, U_{10}, U_{12}$  for  $\Gamma_{F_1}$ , and  $U_6, U_9, U_{11}$  for  $\Gamma_{F_2}$  could be removed to simplify each model. The remaining inputs have an acceptance rate between 15% and 85%, which means that they are sometimes classified as independent and sometimes not. They correspond to inputs that have a small effect on the output, as can be observed in Figure 2.10. By taking the pair of the two sets, the influential inputs are the one that were influential on at least one of the sets. This results in five dominant inputs that can be ranked by their first order index:  $U_{13} \succeq U_{14} \succeq U_2 \succeq U_3 \succeq U_5$ . The results associated to the excursion sets of the forms  $\Gamma_{f_i} = \{x \in \mathcal{X}, f_i(x, \mathbf{U}) \leq q_i\}$  are similar in this case and given in the Supplementary Materials.

## 2.5 Conclusion

In this paper, we propose a method to perform sensitivity analysis on set-valued outputs through kernel-based sensitivity analysis, which relies on the choice of a kernel between sets. We introduce the kernel  $k_{set}$ , which is based on the symmetric difference between two sets. We show that it is characteristic, which is an essential property for performing screening. We then adapt the recent



**Figure 2.7:** Acceptance rates (a) and estimation of  $\hat{S}_i^{Hset}$  (b) for the excursion set  $\Gamma_{g_2}$  computed for 5 kernels with  $n = 100$ ,  $m = 100$  and repeated 20 times

HSIC-ANOVA index to set-valued outputs and introduce an efficient estimator. Finally, we compute the indices on three test cases including a real application for robust design of electrical motor. The proposed method allows to screen and rank the uncertain inputs according to their impact on the excursion sets. For future research, it could be interesting to find and study other set-valued output kernels. Other types of approaches to perform sensitivity analysis for sets could also be investigated, such as using universal indices from [37], or by using classic random set theory from [105].

In the context of robust optimization, the presented method and especially the screening results could be used to reduce the dimension of the uncertain space by quantifying the impact of uncertain inputs on the optimization constraints. Reducing the dimension of the uncertain space can then be useful to reduce the computational cost of a joint space metamodel that could be used within a Bayesian optimization.

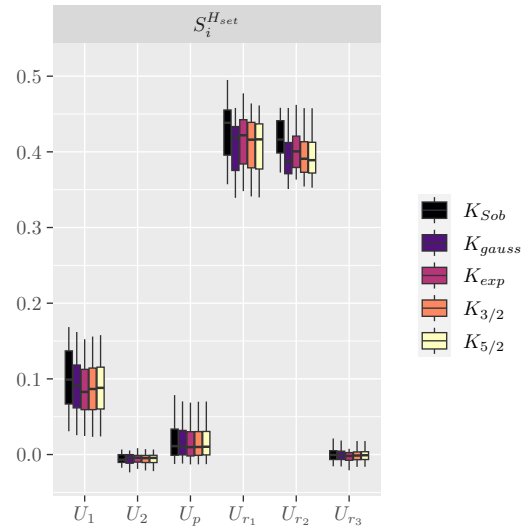
More generally, sensitivity analysis for sets can also be used when dealing with numerical codes with set-valued outputs. This occurs in several areas: in the field of viability, where the outputs are sets called viability kernels, or, for example, in flood risk, where the output is the map of flooded areas.

## Supplementary Materials

The supplementary materials given in Appendix B contain details of the proofs of Lemma 2.1 and of Proposition 2.1 to 2.4, additional numerical results, and codes to reproduce 2.3.

| Kernel      | $U_1$ | $U_2$ | $U_p$ | $U_{r_1}$ | $U_{r_2}$ | $U_{r_3}$ |
|-------------|-------|-------|-------|-----------|-----------|-----------|
| $K_{Sob}$   | 0     | 95    | 60    | 0         | 0         | 90        |
| $K_{gauss}$ | 0     | 100   | 60    | 0         | 0         | 90        |
| $K_{exp}$   | 0     | 95    | 60    | 0         | 0         | 90        |
| $K_{3/2}$   | 0     | 100   | 60    | 0         | 0         | 90        |
| $K_{5/2}$   | 0     | 100   | 60    | 0         | 0         | 90        |

(a) Acceptance rates (%) over 20 independence tests with a risk of 5%



(b) Estimations of  $\hat{S}_i^{Hset}$

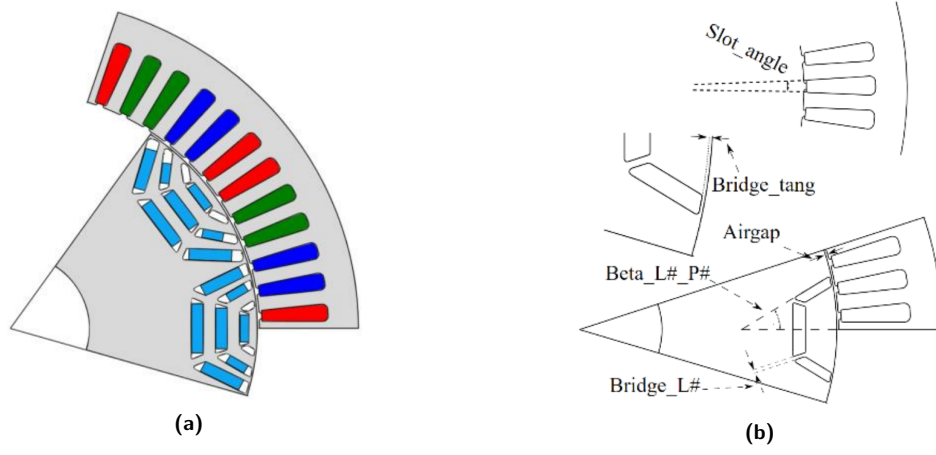
**Figure 2.8:** Acceptance rates (a) and estimation of  $\hat{S}_i^{Hset}$  (b) for the pair of excursion sets  $(\Gamma_{g_1}, \Gamma_{g_2})$  computed for 5 kernels with  $n = 100$ ,  $m = 100$  and repeated 20 times

## Acknowledgment

The authors thank Gabriel Sarazin for his numerous fruitful discussions and comments. We are also grateful to the reviewers and the associate editor for their relevant and helpful comments and to Adan Reyes Reyes for providing the model of Section 2.4.3. This research was conducted with the support of the consortium in Applied Mathematics [CIROQUO](#), gathering partners in technological research and academia in the development of advanced methods for Computer Experiments.

## Disclosure Statement

The authors report there are no competing interests to declare.



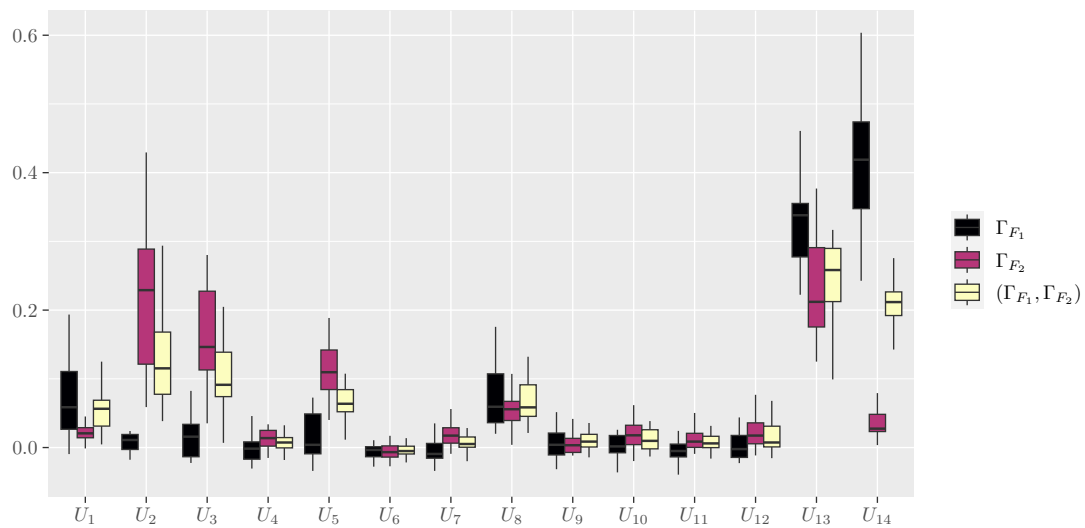
**Figure 2.9:** (a) Geometry of the permanent magnet assisted synchronous reluctance motor and (b) Design parameters for one layer (# is the number of layer)

| Input parameters | Lower bound<br>$x_{\min}$ | Upper bound<br>$x_{\max}$ | Manufacturing Tolerance $U$ |
|------------------|---------------------------|---------------------------|-----------------------------|
| Slot angle       | $2.47^\circ$              | $3.27^\circ$              | $\pm 0.1^\circ$             |
| $\beta_{L1P1}$   | $27.03^\circ$             | $29.66^\circ$             | $\pm 0.33^\circ$            |
| $\beta_{L1P2}$   | $37.03^\circ$             | $39.66^\circ$             | $\pm 0.33^\circ$            |
| $\beta_{L2P1}$   | $31.03^\circ$             | $33.66^\circ$             | $\pm 0.33^\circ$            |
| $\beta_{L2P2}$   | $47.03^\circ$             | $49.66^\circ$             | $\pm 0.33^\circ$            |
| $\beta_{L3P1}$   | $33.7^\circ$              | $37^\circ$                | $\pm 0.33^\circ$            |
| $\beta_{L3P2}$   | $59.7^\circ$              | $63^\circ$                | $\pm 0.33^\circ$            |
| Airgap           | 0.55 mm                   | 0.65 mm                   | $\pm 0.03$ mm               |
| Bridge $_{L1}$   | 2.6 mm                    | 2.98 mm                   | $\pm 0.05$ mm               |
| Bridge $_{L2}$   | 0.9 mm                    | 1.18 mm                   | $\pm 0.05$ mm               |
| Bridge $_{L3}$   | 0.5 mm                    | 0.62 mm                   | $\pm 0.03$ mm               |
| Bridge $_{tang}$ | 0.4 mm                    | 0.6 mm                    | $\pm 0.05$ mm               |

**Table 2.2:** Geometrical variables (see [137] for more details)

|                                | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ | $U_{12}$ | $U_{13}$ | $U_{14}$ |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|
| $\Gamma_{F_1}$                 | 0.40  | 0.90  | 0.75  | 0.90  | 0.75  | 1.00  | 1.00  | 0.40  | 0.90  | 0.90     | 0.85     | 1.00     | 0.00     | 0.00     |
| $\Gamma_{F_2}$                 | 0.70  | 0.00  | 0.00  | 0.75  | 0.00  | 1.00  | 0.65  | 0.15  | 0.95  | 0.70     | 0.90     | 0.60     | 0.00     | 0.40     |
| $(\Gamma_{F_1}, \Gamma_{F_2})$ | 0.25  | 0.00  | 0.05  | 0.90  | 0.05  | 1.00  | 0.95  | 0.10  | 0.85  | 0.75     | 0.80     | 0.75     | 0.00     | 0.00     |

**Table 2.3:** Acceptance rates (%) over 20 independence tests with a risk of 5% for the excursion sets  $\Gamma_{F_1}$ ,  $\Gamma_{F_2}$  and the pair  $(\Gamma_{F_1}, \Gamma_{F_2})$  with  $-q_1 = 420$  N.m and  $q_2 = 7\%$  computed with the Sobolev input kernel and with  $n = 100$ ,  $m = 100$ .



**Figure 2.10:** Estimation of  $\hat{S}_i^{H_{set}}$  for the excursion sets  $\Gamma_{F_1}$ ,  $\Gamma_{F_2}$  and the pair  $(\Gamma_{F_1}, \Gamma_{F_2})$  with  $-q_1 = 420$  N.m and  $q_2 = 7\%$  computed with the Sobolev input kernel and with  $n = 100$ ,  $m = 100$ . 20 replicates.

# 3

## Article 2 : Sensitivity Analysis for sets: application to Pollutant concentration maps

---

|       |                                                          |    |
|-------|----------------------------------------------------------|----|
| 3.1   | Introduction                                             | 53 |
| 3.2   | Motivating test case                                     | 54 |
| 3.3   | New sensitivity analysis indices for sets-valued outputs | 56 |
| 3.3.1 | Sensitivity analysis based on random set theory          | 57 |
| 3.3.2 | Universal sensitivity indices                            | 59 |
| 3.3.3 | HSIC-based sensitivity analysis                          | 60 |
| 3.4   | Results                                                  | 62 |
| 3.4.1 | Vorob'ev based indices                                   | 62 |
| 3.4.2 | Universal indices                                        | 62 |
| 3.4.3 | HSIC                                                     | 65 |
| 3.4.4 | Comparison                                               | 68 |
| 3.5   | Discussion                                               | 69 |

---

# Sensitivity analysis for sets : application to pollutant concentration maps

N. Fellmann<sup>†,1,2</sup>, M. Pasquier<sup>§,2</sup>, C. Helbert<sup>§,1</sup>, C. Blanchet<sup>§,1</sup>, A. Spagnol<sup>§,2</sup>, D. Sinoquet<sup>§,2</sup>

<sup>1</sup>École Centrale de Lyon, CNRS UMR 5208, Institut Camille Jordan  
{noe.fellmann,christophette.blanchet,celine.helbert}@ec-lyon.fr

<sup>2</sup>IFP Energies Nouvelles  
{noe.fellmann,delphine.sinoquet,adrien.spagnol,mathis.pasquier}@ifp.fr

Fellmann N, Pasquier M, Blanchet-Scalliet C, Helbert C, Spagnol A, Sinoquet D. Sensitivity analysis for sets: Application to pollutant concentration maps. Qual Reliab Eng Int. 2024; [36]

## Abstract

We are motivated by the field of air quality control, where one goal is to quantify the impact of uncertain inputs such as meteorological conditions and traffic parameters on pollutant dispersion maps. Sensitivity analysis is one answer, but the majority of sensitivity analysis methods are designed to deal with scalar or vector outputs and are badly suited to an output space of maps. To address this problem, we propose a generic approach to sensitivity analysis of set-valued models. This approach can be applied to the case of maps. We propose and study three different types of sensitivity indices. The first ones are inspired by Sobol' indices, but adapted to sets based on the theory of random sets. The second ones adapt *universal* indices defined for a general metric output space. The last set indices use kernel-based sensitivity indices adapted to sets. The proposed methods are implemented and tested to perform an uncertainty analysis for a toy excursion set problem and for time-averaged concentration maps of pollutants in an urban environment.

*Keywords:* Random sets, HSIC-ANOVA indices, universal indices, pollutant dispersion

### 3.1 Introduction

In modern scientific research, the increasing complexity of numerical models requires effective methods for analysis and simplification. Sensitivity analysis is one of the answers to both analyze and simplify the model by measuring the effect of inputs on outputs. A significant challenge arises when the nature of the model output itself is complex, such as images, functions, and sets.

This study focuses on sets, and aims to propose and compare methods for performing sensitivity analysis when dealing with set-valued outputs. Sets appear in various contexts, such as optimization and inversion with excursion sets, viability theory, and others. Sets also provide a versatile tool for generalizing a variety of output types, including functional outputs, by replacing the functional output with its hypograph (or epigraph), i.e., the sets of points below (or above) its graph.

Initially, sensitivity analysis methods focused primarily on models with scalar inputs and outputs. Standard methods such as the Morris analysis [109] and the Sobol' index [153] were widely used to evaluate the variation of outputs in response to variations in input parameters. To deal with complex outputs, new advanced sensitivity analysis techniques have emerged. For example, Sobol' indices have been generalized for vectorial and functional outputs [43]. However, the latter uses a truncated expansion of the functional output to then reuse the proposed index on vectorial outputs, which implies a loss of information. In [127], the authors propose a method to perform a global sensitivity analysis in the case of spatial output through functional principal component analysis. [141] also performs sensitivity analysis for spatial outputs by computing maps of sensitivity indices. In [37], the authors propose universal sensitivity indices that can be computed in any metric output space. The more recent indices based on the Hilbert-Schmidt Independence Criterion (HSIC) were also originally defined for scalar and vector inputs and outputs (see [51, 24]), but are much more permissive and can be extended to arbitrary outputs as long as a kernel is defined on the output space. For example, HSIC-based indices have been used for probability distribution outputs in [25]. The consideration of set-valued outputs for sensitivity analysis has received very little attention in the literature. In a previous work [35], we proposed a kernel on sets to perform kernel-based sensitivity analysis of set-valued models, noted  $H_{set}$  in the following.

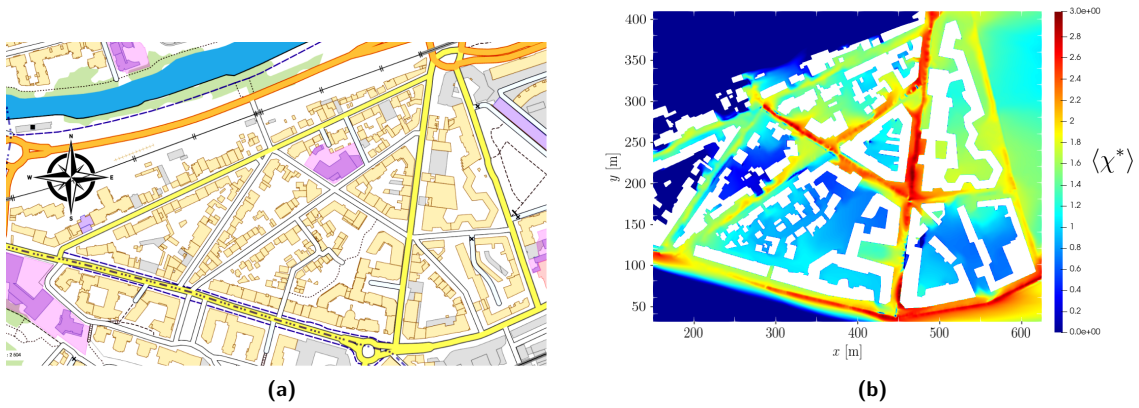
The first goal of this paper is to propose alternative indices for set-valued models. A first alternative approach is based on the universal index mentioned above, as it is designed to be usable in any output space. We propose to adapt this index to an output space of sets. Recognizing that set outputs are essentially random sets due to the randomness of the inputs, we can draw on the theory of random sets [105], which defines analogs of traditional statistical measures such as expectation, median, and variance. This theoretical framework allows us to derive a second sensitivity indices similar to the Sobol' indices, but based on the theory of random sets.

The second objective of this paper is to compare these two new indices and  $H_{set}$  in terms of interpretability, ease of use and precision. Apart from a toy function, this comparison will be made in the case of a pollutant concentration map model that produces two-dimensional maps. This model was developed in [124, 123], where the authors, for lack of relevant tools, perform a pointwise sensitivity analysis to catch effect of input randomness.

The paper is organized as follows. Section 3.2 presents the pollutant concentration map model as a motivating test case and preliminary sensitivity analysis results. Section 3.3 is dedicated to the presentation of new set-based SA indices: sensitivity analysis indices based on random set theory and universal indices from [37] adapted set-valued outputs. The  $H_{set}$  indices of [35] are also recalled. Section 3.4 presents the results of the computation of these three different indices on a toy case and on the pollutant concentration map model. Finally, the indices are compared in Section 3.5 to draw some conclusions.

## 3.2 Motivating test case

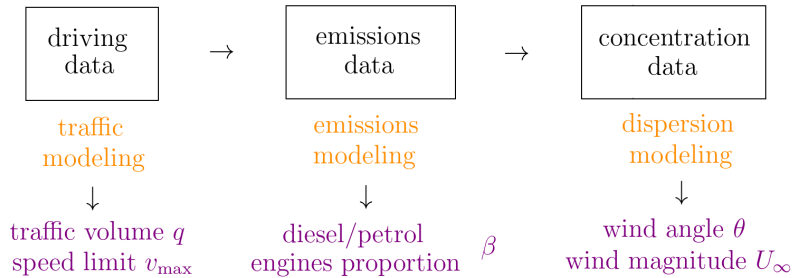
Air quality and the dispersion of atmospheric pollutants are key issues addressed by the emerging discipline of urban physics. Using Computational Fluid Dynamics (CFD) as a tool, this branch of physics simulates the dispersion of pollutants to support urban planning and emergency decision making. In [124], a complete modeling chain has been developed to simulate traffic-related pollutant dispersion at the local urban scale. The method combines a microscopic traffic simulator with a physical engine model to estimate realistic road emissions, which are used as input to a CFD code to model unsteady atmospheric dispersion and compute two-dimensional time-averaged concentration maps at ground level. A presentation of this application is shown in Figure 3.1.



**Figure 3.1:** Concentration maps from numerical CFD simulations of urban pollutant dispersion [124].

- (a) Area of interest, captured from OpenStreetMaps.  
(b) Time-averaged map of the logarithm of the concentration, denoted  $\langle \chi^* \rangle$ , obtained from a run of the CFD code with realistic road emissions. In this case, the wind comes from the left.

The modeling chain outputs two-dimensional concentration maps from a set of inputs associated with meteorological and traffic-related uncertainties. These five uncertain inputs are the incoming wind direction  $\theta$  [radians] and intensity  $U_\infty$  [meters/second], together with the volume of traffic injected into the road network in the urban geometry,  $q$  [vehicles/hour], the proportion of diesel and gasoline engines in the fleet  $\beta$  [-], and the speed limit over the network,  $v_{max}$  [kilometers/hour]. The modeling chain with the corresponding uncertain variables is shown in Figure 3.2 and the probability distributions associated with each variable are given in Table 3.1.



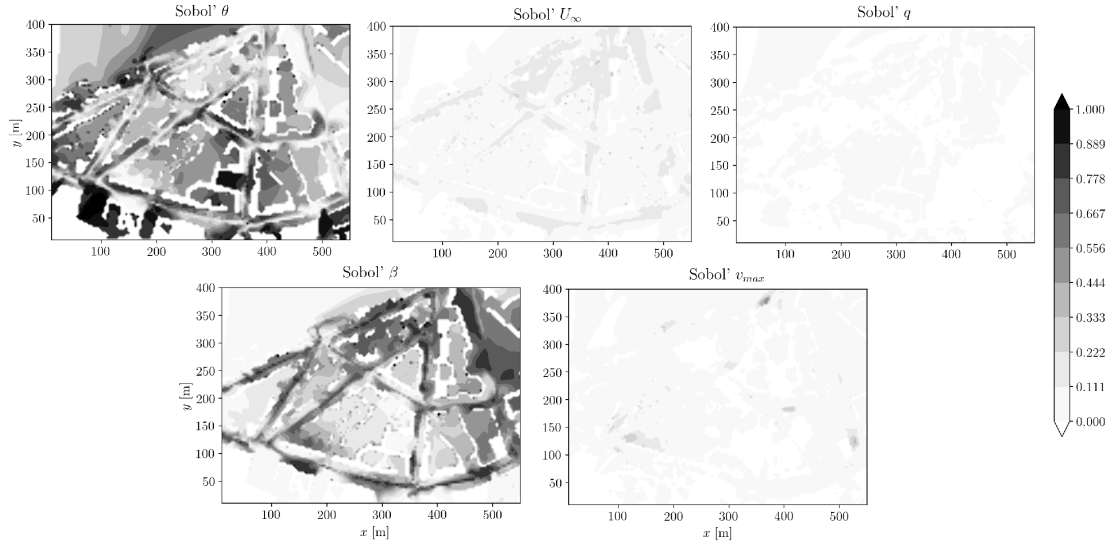
**Figure 3.2:** Whole modeling chain designed in [124] for the calculation of time-averaged concentration maps. The five uncertain parameters considered are indicated below each block.

| variable    | distribution            | bounds                 | parameters                             |
|-------------|-------------------------|------------------------|----------------------------------------|
| $\theta$    | truncated normal        | [5; 10] [rad]          | $\mu = 0.7, \sigma = 0.5$              |
| $U_\infty$  | truncated normal        | $[-0.35; 1.75]$ [m/s]  | $\mu = 8, \sigma = 2$                  |
| $q$         | truncated skewed normal | [100; 500] [vehicle/h] | $\xi = 450, \omega = 100, \alpha = -3$ |
| $\beta$     | uniform                 | [0; 1] [-]             | -                                      |
| $\nu_{max}$ | uniform                 | [30; 50] [km/h]        | -                                      |

**Table 3.1:** Probability distributions and associated parameters for the five uncertain pollutant dispersion variables

To have access to a cheaper model, we construct a metamodel of the entire modeling chain using a combination of Principal Component Analysis and Gaussian Process Regression (GPR), known in the literature as POD-GPR [56, 116, 101]. The metamodel is trained on a Latin Hypercube Sampling (LHS) design of experiments of 150 expensive CFD simulations – each simulation requires about 7 hours of computational time on 960 cores on the high performance computing facility of IFP Energies Nouvelles – and validated on a set of 85 additional simulations. A global coefficient of determination, defined as in [116], is computed to quantitatively evaluate the predictivity of the POD-GPR metamodel, which reaches about 86%. The lack of predictivity is related to residual noise in the numerical simulation results due to the difficult statistical convergence of the time-averaged concentration fields in CFD simulations.

With this cheaper model, the effect of the inputs on the output maps can be quantified by performing a sensitivity analysis of the model. The simplest approach consists in performing a pointwise sensitivity analysis, i.e. to compute the Sobol' indices (the fraction of output variance explained by an input) at each pixel of the map, resulting in the five Sobol' indices maps shown in Figure 3.3.



**Figure 3.3:** Two-dimensional maps of pointwise first-order Sobol' indices for time-averaged pollutant concentrations estimated by quasi-Monte-Carlo using  $2^{10}$  sample points and Halton sequences.

The information provided by such two-dimensional maps can be aggregated into scalar values. To do this, each Sobol index (associated with a given location) is weighted by the fraction of the total output variance explained at that location. This weighted sum results in *generalized* Sobol'

indices, which correspond to the vectorial Sobol' indices defined in [43]. The resulting indices are given in Table 3.2. They show that the wind angle ( $\theta$ ) and the proportion of diesel engines in the

| $S_{\theta}^{\text{gen}}$ | $S_{U_{\infty}}^{\text{gen}}$ | $S_q^{\text{gen}}$    | $S_{\beta}^{\text{gen}}$ | $S_{\nu_{\max}}^{\text{gen}}$ |
|---------------------------|-------------------------------|-----------------------|--------------------------|-------------------------------|
| $5.48 \times 10^{-1}$     | $4.47 \times 10^{-2}$         | $7.56 \times 10^{-4}$ | $3.22 \times 10^{-1}$    | $1.73 \times 10^{-2}$         |

**Table 3.2:** Generalized first-order Sobol' indices for each uncertain variable from two-dimensional sensitivity maps, estimated with  $2^{10}$  sample points from Halton sequences.

fleet ( $\beta$ ) contribute to most of the variance of the concentration field. The generalized indices reach about 55% and 32% for these two variables respectively, while the other indices fall below 10%, and in particular the value of  $S_q^{\text{gen}}$  is very close to zero, showing that the traffic volume  $q$  is not influential in this study. The two-dimensional Sobol' maps highlight the fact that the sensitivity to the two dominant variables varies depending on the location in the urban neighborhood: thus, if one chooses to focus the analysis on a specific part of the domain - e.g. a specific building such as a hospital or a primary school - a different hierarchy among the variables can be deduced. In [123], a second analysis is also performed by fixing  $\theta$  and  $\beta$  to their mean values and evaluating the Sobol' maps for the three remaining variables. This shows that the wind intensity dominates over the speed limit (with a generalized index of about 70% for the former and 30% for the latter) and the traffic volume remains negligible. It can also be observed that the indices sum to about 95%, which means that there are almost no interactions between the inputs.

Although this sensitivity analysis is performed point by point of the pollutant concentration map, it allows a first screening of the influencing variables and, in particular, generalized indices allow a quantitative ranking of the inputs. However, the previous approach doesn't use the geometry of the problem and treats the output as a vector. We propose an alternative approach by considering the hypographs  $\Gamma_u$  of the map-valued model  $\Phi : \mathbf{u} = (\theta, U_{\infty}, q, \beta, \nu_{\max}) \mapsto \Phi_{\mathbf{u}}(x_1, x_2)$  defined as

$$\Gamma_u = \{(x_1, x_2, c) \in \mathcal{X}, c \leq \Phi_{\mathbf{u}}(x_1, x_2)\}, \quad (3.1)$$

where  $\mathcal{X} = \mathcal{D} \times \mathcal{P}$  with  $\mathcal{D} = [x_1^{\min}, x_1^{\max}] \times [x_2^{\min}, x_2^{\max}]$  and  $\mathcal{P} = [c^{\min}, c^{\max}]$  the range of possible pollutant concentrations. The next section presents three different indices to perform a sensitivity analysis of such set-valued models.

### 3.3 New sensitivity analysis indices for sets-valued outputs .....

To remain general, let  $\Psi$  be any set-valued model, i.e. a measurable function defined by:

$$\begin{aligned} \mathcal{U} &\longrightarrow \mathcal{L}(\mathcal{X}) \\ \Psi : \mathbf{u} &\mapsto \Gamma_{\mathbf{u}} \end{aligned}$$

where  $\mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_p \subset \mathbb{R}^p$  is the uncertain input space and  $\mathcal{L}(\mathcal{X})$  is the space of Lebesgue measurable subsets of a compact subset  $\mathcal{X}$  of  $\mathbb{R}^d$ . For example, such models can represent hypographs as in Equation 3.1 or excursion sets defined as  $\Psi(\mathbf{u}) = \{\mathbf{x} \in \mathcal{X}, g(\mathbf{x}, \mathbf{u}) \leq 0\}$ . As in many sensitivity analysis frameworks, the inputs are assumed to be a random vector  $\mathbf{U} = (U_1, \dots, U_p) \in \mathcal{U}$  with independent components. It is defined on a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$  with known distributions

$\mathbb{P}_{\mathbf{U}} = \bigotimes_{i=1}^p \mathbb{P}_{U_i}$ .  $\Gamma = \Psi(\mathbf{U})$  is then a random set whose randomness comes from the inputs  $\mathbf{U}$ . For estimation purposes,  $\mathcal{X}$  is discretized with a grid of  $m$  points called  $(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$ . This allows the

volume  $\lambda(A)$  of  $A \in \mathcal{L}(\mathcal{X})$  to be estimated by

$$\hat{\lambda}_m(A) = \frac{1}{m} \sum_{k=1}^m \mathbb{1}_A(\mathbf{x}^{(k)}).$$

Let also  $(\mathbf{U}^{(i)}, \Gamma^{(i)}), i = 1, \dots, n$  be an independent and identically distributed (i.i.d.) sample of  $(\mathbf{U}, \Gamma)$ .

### 3.3.1 Sensitivity analysis based on random set theory

Working with sets instead of scalars or vectors makes the study of random elements much more difficult. For example, expectations and variance are not easy to define. In [105], a complete theory of random sets is developed with definitions of expectations, medians and deviations of random sets. In particular, based on the Vorob'ev's median and deviation, we propose to define indices inspired by Sobol' indices but adapted to random sets.

When dealing with a scalar output  $Y$ , first-order Sobol' indices  $S_i$  can be defined by :

$$S_i = 1 - \frac{\mathbb{E}[\text{Var}[Y|U_i]]}{\text{Var } Y}.$$

Here we want to define similar indices, but for set-valued outputs. To do this, we propose to replace the variance with the Vorob'ev median deviation, which is a possible adaptation of the median deviation to random sets. This requires defining the Vorob'ev median and the Vorob'ev median deviation, but also defining a Vorob'ev conditional median deviation.

**Definition 3.1** (Vorob'ev median). *The Vorob'ev median  $Q_{0.5}(\Gamma)$  of a random set  $\Gamma$  is defined as the set of points whose probability of being in  $\Gamma$  is at least 0.5, i.e.*

$$Q_{0.5}(\Gamma) = \{x \in \mathcal{X}, \mathbb{P}(x \in \Gamma) \geq 0.5\}$$

The Vorob'ev median corresponds to the set of points that are "often enough" in the random set. It is illustrated in Figure 3.4a on a toy function.

Using the Vorob'ev median, the Vorob'ev Median Deviation is defined as the Vorob'ev deviation between the random set and its median. We propose a conditional version of the Vorob'ev Median Deviation.

**Definition 3.2.** *The Vorob'ev Median Deviation of a random set  $\Gamma$ , denoted by  $\text{VMD}(\Gamma)$ , is defined by:*

$$\text{VMD}(\Gamma) = \mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})].$$

where  $\Delta$  is the symmetric difference. The Vorob'ev conditional median deviation is the random variable  $\text{VMD}(\Gamma|U_i)$  defined by:

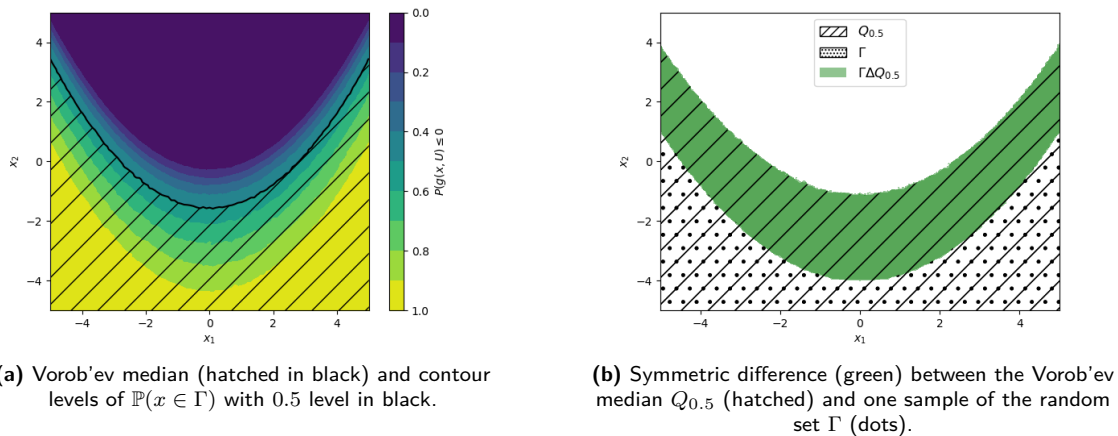
$$\text{VMD}(\Gamma|U_i) = \mathbb{E}[\lambda(\Gamma \Delta Q_{0.5}^i)|U_i]$$

where  $Q_{0.5}^i$  is the Vorob'ev Conditional median defined by:

$$Q_{0.5}^i = \{\mathbf{x} \in \mathcal{X}, \mathbb{P}(\mathbf{x} \in \Gamma|U_i) \geq 0.5\}.$$

The Vorob'ev Median Deviation calculation is based on the calculation of the symmetric difference volume between the Vorob'ev median and samples of the random set, as illustrated in Figure 3.4b.

We can now define indices that describe the part of the Vorob'ev median deviation of a random set that is due to an input  $U_i$ .



**Figure 3.4:** Example of Vorob'ev median and symmetric difference for the excursion set defined as  $\Gamma = \{(x_1, x_2) \in [-5; 5]^2, g(x_1, x_2, U_1, U_2) := -x_1^2 + 5x_2 - U_1 + U_2^2 - 1 \leq 0\}$  with  $(U_1, U_2)$  of uniform distributions in  $[-5; 5]^2$

**Definition 3.3.** Using the previous definition, we define a sensitivity index  $S_V^i$  by:

$$\begin{aligned} S_V^i &:= 1 - \frac{\mathbb{E}[\text{VMD}(\Gamma|U_i)]}{\text{VMD}(\Gamma)} \\ &= 1 - \frac{\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5}^i)]}{\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})]} \end{aligned}$$

This index quantifies the effect of  $U_i$  on the output  $\Gamma$ , since it's zero if  $\Gamma$  and  $U_i$  are independent (simply as  $Q_{0.5}^i = Q_{0.5}$  if  $\Gamma$  and  $U_i$  are independent). However, as first-order Sobol' indices, the converse doesn't hold. We show in Proposition 3.1 that it is between zero and one, as desired for sensitivity indices. The interpretation is close to a fraction of the total deviation explained by an input. However, there is no decomposition of the indices and no guarantee that their sum remains smaller than 1.

**Proposition 3.1.**  $0 \leq S_V^i \leq 1$

*Proof.* It is clear that  $S_V^i \leq 1$ . To show that it is non-negative, we use the tower property of the conditional expectation on the numerator and the denominator :

$$\frac{\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5}^i)]}{\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})]} = \frac{\mathbb{E}\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5}^i)|U_i]}{\mathbb{E}\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})|U_i]}.$$

Then we show that

$$\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5}^i)|U_i] \leq \mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})|U_i] \text{ almost surely,}$$

using that the Vorob'ev median minimises the deviation for all measurable sets (see Proposition 2.4 of [105]). Finally by taking the expectation we conclude that  $S_V^i$  is non-negative.  $\square$

The estimation of  $S_V^i$  is done by a double loop, since we do not know a cheaper estimator. Let  $U_i^{(j)}$ ,  $j \in (1, n)$  be  $n$  independent copies of a given input  $U_i$ , the conditional median  $Q_{0.5}^i(U_i^{(j)}) = \{x \in \mathcal{X}, \mathbb{P}(x \in \Gamma|U_i = U_i^{(j)}) \geq 0.5\}$  is estimated by independently drawing another

sample  $U_{-i}^{(j,l)}$ ,  $l \in (1, n)$  of  $U_{-i}$  and evaluating the output sets  $\Gamma_l(U_i^{(j)}) = \Psi(U_i^{(j)}, U_{-i}^{(j,l)})$ . The estimator of the conditional Vorob'ev median can then be derived:

$$\hat{Q}_{0.5}^i(U_i^{(j)}) := \{x \in \mathcal{X}, \frac{1}{n} \sum_{l=1}^n \mathbb{1}_{\Gamma_l(U_i^{(j)})}(x) \geq 0.5\}.$$

Then  $S_i^V$  is estimated by:

$$\tilde{S}_i^V = 1 - \frac{\sum_{j=1}^n \hat{\lambda}_m(\Gamma^{(j)} \Delta \hat{Q}_{0.5}^i(U_i^{(j)}))}{\sum_{j=1}^n \hat{\lambda}_m(\Gamma^{(j)} \Delta \hat{Q}_{0.5})}, \quad (3.2)$$

where  $\Gamma^{(j)} = \Psi(U_i^{(j)}, U_{-i}^{(j)})$ .

### 3.3.2 Universal sensitivity indices

In this second part, we propose to adapt *universal* indices from [37]. In the latter paper, the authors define indices that can be applied to any metric output space. In a context of set-valued outputs, we propose to define the following indices:

**Definition 3.4.**

$$S_i^{Univ}(\gamma, \mathbb{Q}) = \frac{\int_0^1 \text{Var} \mathbb{E}[\lambda(\Gamma \Delta \gamma_a) \mid U_i] d\mathbb{Q}(a)}{\int_0^1 \text{Var}(\lambda(\Gamma \Delta \gamma_a)) d\mathbb{Q}(a)}.$$

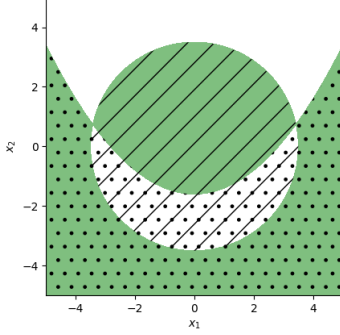
where  $(\gamma_a)_{a \in [0,1]} \subset \mathcal{X}$  is a collection of test sets parameterised by  $a \in [0, 1]$  and  $\mathbb{Q}$  is a probability distribution on  $[0, 1]$  to choose.

The idea behind these indices is to quantify the effect of inputs on multiple scalar transformations of the set-valued output, rather than on the set-valued output itself. In our case, the scalar in question is the volume of the symmetric difference between the output set and a collection of test sets. Regarding the choice of test sets, we propose to use and compare different families of sets. To stay general,  $\mathcal{X}$  is rescaled to  $[0, 1]^d$  centered on  $x^0 = (\frac{1}{2}, \dots, \frac{1}{2})$ . Four families of test sets are defined on  $[0, 1]^d$ :

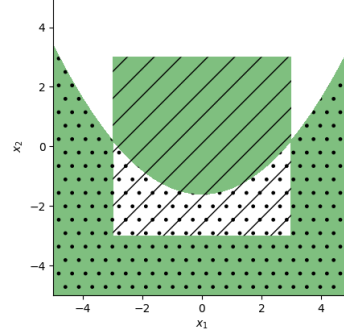
- Centered balls:  $\gamma_a = B(x^0, a)$  with  $\mathbb{Q} \sim \mathcal{U}([0, \frac{1}{2}])$
- Centered squares:  $\gamma_a = \{x \in [0, 1]^d, \|x - x^0\|_\infty \leq a\}$  with  $\mathbb{Q} \sim \mathcal{U}([0, \frac{1}{2}])$
- Slices along the  $i$ -th dimension:  $\gamma_a^i = \{x \in [0, 1]^d, x_i \leq a\}$  with  $\mathbb{Q} \sim \mathcal{U}([0, 1])$
- Vorob'ev quantiles:  $\gamma_a = \{x \in [0, 1]^d, \mathbb{P}(x \in \Gamma) \geq a\}$  with  $\mathbb{Q} \sim \mathcal{N}(\frac{1}{2}, \sigma^2)$  or  $\mathbb{Q} \sim \mathcal{U}(0, 1)$

Balls and squares are designed to be simple sets independent of models. Slices may be appropriate when a direction corresponds to a quantity of interest, as is the case in the pollutant model. Finally, Vorob'ev quantiles test sets use the geometry of the model by comparing the random sets with its quantiles. However, they must be estimated. Examples of symmetric difference with balls and sets are given in Figure 3.5.

Similar to Sobol' indices, the contribution of each input and their interactions can be decomposed to rank the inputs. The decomposition can be obtained by taking the Sobol-Hoeffding decomposition of  $\text{Var}(\lambda(\gamma \Delta \gamma_a))$  for each  $a$  and then summing for 0 to 1 with respect to the measure  $\mathbb{Q}$ . An independence test is not accessible, mainly because the family of all transformations does not necessarily characterize the whole distribution of the set-valued output. However, screening can



(a) Symmetric difference (in green) between one sample of the random set (dots) and a ball of radius 3.5 (hatched).



(b) Symmetric difference (in green) between one sample of the random set (dots) and a square of side 3 (hatched).

**Figure 3.5:** Example of symmetric differences with ball and square test sets for the excursion set defined as  $\Gamma = \{(x_1, x_2) \in [-5; 5]^2, g(x_1, x_2, U_1, U_2) := -x_1^2 + 5x_2 - U_1 + U_2^2 - 1 \leq 0\}$  with  $(U_1, U_2)$  of uniform distributions in  $[-5; 5]^2$

still be done, either by keeping only the inputs above a certain threshold, or by keeping the first inputs in the ranking.

After selecting the test sets and the law  $\mathbb{Q}$ , these indices can be estimated in a similar way to Sobol's indices. We use a rank-based estimator, which is recalled in [37]. Plugging in the volume estimation, the estimator  $\hat{S}_i^{\text{Univ}}$  of  $S_i^{\text{Univ}}$  is given by the ratio between

$$\begin{aligned} \hat{S}_{num} = \frac{1}{N_a} \sum_{l=1}^{N_a} \left[ \frac{1}{n} \sum_{j=1}^n \left( \hat{\lambda}_m(\Gamma^{(j)} \Delta \gamma_{a_l}) \right) \left( \hat{\lambda}_m(\Gamma^{(N_i(j))} \Delta \gamma_{a_l}) \right) \right] \\ - \frac{1}{N_a} \sum_{l=1}^{N_a} \left[ \frac{1}{n} \sum_{j=1}^n \left( \hat{\lambda}_m(\Gamma^{(j)} \Delta \gamma_{a_l}) \right) \right]^2 \end{aligned} \quad (3.3)$$

and,

$$\hat{S}_{den} = \frac{1}{N_a} \sum_{l=1}^{N_a} \left[ \frac{1}{n} \sum_{j=1}^n \left( \hat{\lambda}_m(\Gamma^{(j)} \Delta \gamma_{a_l}) \right)^2 \right] - \frac{1}{N_a} \sum_{l=1}^{N_a} \left[ \frac{1}{n} \sum_{j=1}^n \left( \hat{\lambda}_m(\Gamma^{(j)} \Delta \gamma_{a_l}) \right) \right]^2, \quad (3.4)$$

where  $(a_l)$  is an iid sample of the law  $\mathbb{Q}$  and  $N_i(j)$  is the index in the sample  $(U_i^l)$  that comes after  $U_i^j$  when  $(U_i^l)$  is sorted in ascending order (see [42] for details).

### 3.3.3 HSIC-based sensitivity analysis

Sensitivity analysis based on measures of dependence consists in examining the influence of an input  $X_i$  on an output  $Y$  by measuring their mutual dependence. This is done by computing a distance between the joint distribution  $\mathbb{P}_{X_i, Y}$  and the product of their marginal distributions  $\mathbb{P}_{X_i} \otimes \mathbb{P}_Y$ . If this distance is zero, then  $X_i$  and  $Y$  are independent, i.e.  $X_i$  has no effect on  $Y$ . A popular measure of dependence is the Hilbert-Schmidt Independence Criterion (HSIC). It works by selecting

appropriate input and output kernels, and then measuring the squared distance between the mean embeddings of  $\mathbb{P}_{X_i, Y}$  and  $\mathbb{P}_{X_i} \otimes \mathbb{P}_Y$  in the corresponding Reproducing Kernel Hilbert Space (RKHS), see [51] for a proper definition of the tools needed to define the HSIC. This calculation can be done with only one sample, which is one of the reasons why HSICs have gained popularity.

If a *characteristic* kernel is available on any space, i.e. a kernel whose mean embedding in the RKHS is injective, then HSIC-based indices can be used. Consequently, their use is appropriate for sensitivity analysis of set-valued models. In [35], we introduced a kernel  $k_{set}$  defined on sets, which is shown to be characteristic. It is then used to define HSIC-based indices on sets. The latter definition requires

- a characteristic kernel on  $\mathcal{L}(\mathcal{X})$ . The proposed kernel is

$$k_{set}(\gamma_1, \gamma_2) = \exp\left(-\frac{\lambda(\gamma_1 \Delta \gamma_2)}{2\sigma^2}\right) \quad \forall \gamma_1, \gamma_2 \subset \mathcal{X},$$

where  $\sigma^2$  is a normalization constant.

- $p$  characteristic kernels  $K_i$  on the input space  $\mathcal{U}_i$ . As shown in [25], an ANOVA-like decomposition of HSIC exists if the input kernels are ANOVA, i.e. that can be written as  $1 + k_i$ , where  $k_i$  is a kernel whose induced RKHS consists of zero-mean functions. Sobolev kernels are the most commonly used kernels that fulfil these requirements. Furthermore, classic characteristic kernels can be modified to become ANOVA. We will compare five different input ANOVA kernels based on the the following kernels:

- the Sobolev kernel of order 1,  $k_{sob}(x, y) = 1 + (x - \frac{1}{2})(y - \frac{1}{2}) + \frac{1}{2}[(x - y)^2 - |x - y| + \frac{1}{6}]$
- the Gaussian kernel,  $k_{rbf}(x, y) = e^{-\frac{1}{2}\left(\frac{x-y}{\sigma}\right)^2}$  with  $\sigma > 0$ ,
- the Laplace kernel,  $k_{exp}(x, y) = e^{-\frac{|x-y|}{h}}$  with  $h > 0$ ,
- the Matérn 3/2,  $k_{3/2}(x, y) = \left(1 + \sqrt{3}\frac{|x-y|}{h}\right) e^{-\sqrt{3}\frac{|x-y|}{h}}$  with  $h > 0$ ,
- the Matérn 5/2,  $k_{5/2}(x, y) = \left(1 + \sqrt{5}\frac{|x-y|}{h} + \frac{5}{3}\frac{|x-y|^2}{h^2}\right) e^{-\sqrt{5}\frac{|x-y|}{h}}$  with  $h > 0$ .

The Sobolev kernel is already ANOVA, whereas the four additional kernels are adapted to be ANOVA (see [47] for the detailed transformation).

**Definition 3.5.** *With the previous kernels, HSIC-based indices on sets are defined by*

$$\forall i \in \{1, \dots, p\}, \quad S_i^{H_{set}} := \frac{H_{set}(U_i, \Gamma)}{H_{set}(\mathbf{U}, \Gamma)},$$

where

$$\begin{aligned} H_{set}(U_i, \Gamma) &= \mathbb{E}[(K_i(U_i, U_i') - 1)k_{set}(\Gamma, \Gamma')], \\ H_{set}(\mathbf{U}, \Gamma) &= \mathbb{E}[(K(\mathbf{U}, \mathbf{U}') - 1)k_{set}(\Gamma, \Gamma')], \end{aligned}$$

with

$$K = \bigotimes_{j=1}^p K_j,$$

and  $(\mathbf{U}', \Gamma')$  an independent copy of  $(\mathbf{U}, \Gamma)$ .

Since these indices are zero if and only if the considered input  $U_i$  and the output  $\Gamma$  are independent, they are suitable for screening. Independence tests can be performed for each input  $U_i$ :

$$(\mathcal{H}_0^i) : \text{HSIC}(U_i, \Gamma) = 0 \quad \text{versus} \quad (\mathcal{H}_1^i) : \text{HSIC}(U_i, \Gamma) > 0.$$

Associated p-values can be computed to screen the inputs. Thanks to the ANOVA decomposition the inputs can also be ranked by their impact based on the value of each index.

The indices are estimated using:

$$\widehat{\widehat{H}}_{set}(U_i, \Gamma) = \frac{2}{n(n-1)} \sum_{j < l}^n \left( K_i(U_i^{(j)}, U_i^{(l)}) - 1 \right) \exp\left(-\frac{\lambda(\mathcal{X})}{2\sigma^2} \hat{\lambda}_m(\Gamma^{(j)} \Delta \Gamma^{(l)})\right). \quad (3.5)$$

P-values associated with the independence test can be computed with different estimation methods (see [154] for the example of asymptotic estimation).

## 3.4 Results

Each of the three previously introduced indices is computed, studied and analysed in two different situations. First we are interested in the excursion set  $\Gamma = \{\mathbf{x} \in [-5, 5]^2, g(\mathbf{x}, \mathbf{U}) \leq 0\}$  associated with the toy function  $g$  defined by :

$$\forall (x_1, x_2, u_1, u_2) \in [-5, 5]^4, \quad g(x_1, x_2, u_1, u_2) = -x_1^2 + 5x_2 - u_1 + u_2^2 - 1,$$

with  $U_1, U_2$  and a dummy variable  $U_3$  of uniform distributions on  $[-5, 5]$ . Then the indices are computed for the set-valued model associated with the pollutant concentration map model. For every indices, confidence intervals are estimated by bootstrap with 100 resamples.

### 3.4.1 Vorob'ev based indices

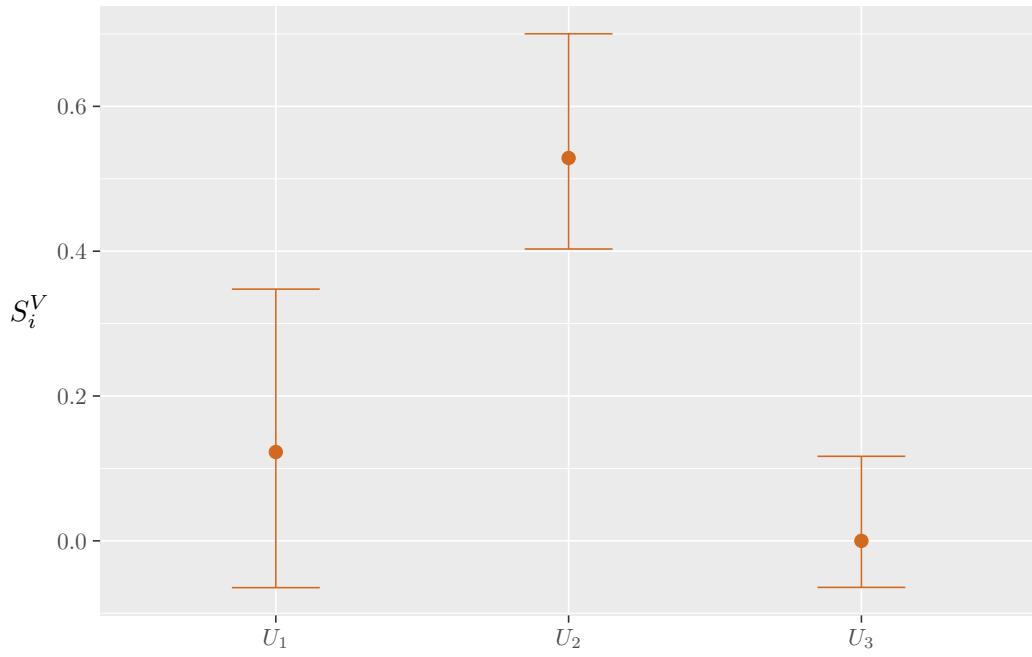
The index is estimated by  $n = 32$  model evaluations per loop, resulting in a total of  $n^2 = 1024$  model evaluations. For the toy case, the results are shown in Figure 3.6. Since these indices have no decomposition, it is difficult to justify a comparison between the value of each index. The only possible interpretation is that an index far from zero means that the input has an effect on the output. Here,  $U_2$  clearly has a strong effect on the output, while  $U_1$  and  $U_3$  are close to 0, implying that they may have no effect. The confidence intervals are wide considering the number of model evaluations. This is mainly due to the double-loop estimation method. Inspired by Sobol's indices, different estimation methods such as pick and freeze or rank-based estimation would greatly improve these indices. However, they do not seem to be applicable in this scenario because the quantity being estimated is not strictly a conditional variance.

For the pollutant case, the results are given in Figure 3.7. The intervals of  $\beta$  and  $\theta$  do not include zero, which means they have an effect. The other three, however, are close to 0, which means that they may have no effect. Again, we see that the intervals are really wide considering that 1024 model evaluations are used.

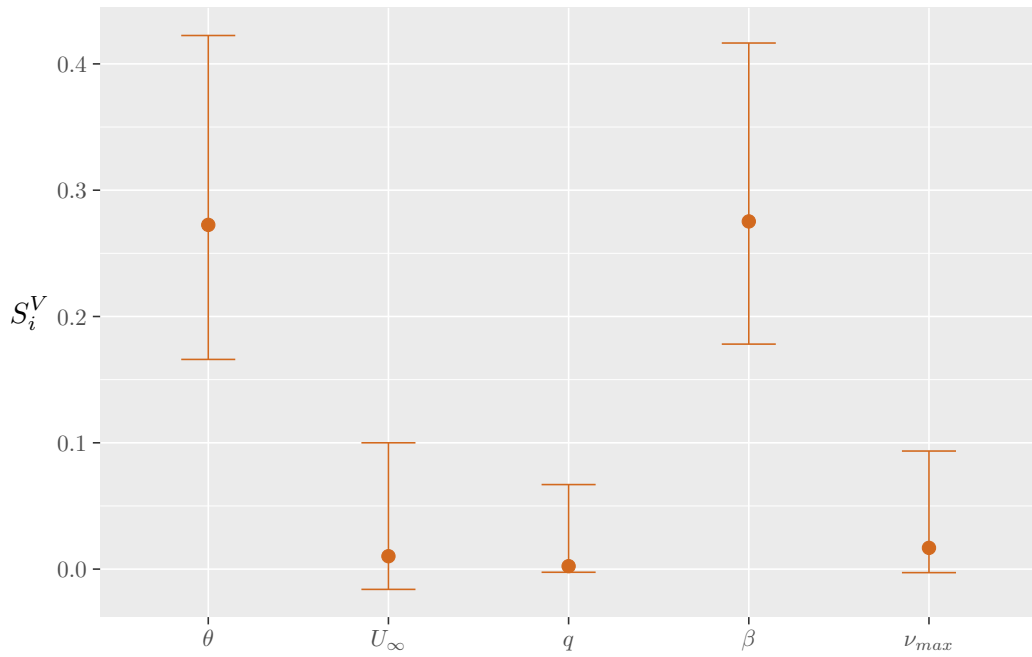
### 3.4.2 Universal indices

We compute the universal indices with  $n = 1000$  model evaluations and  $N_a = 100$  test sets. We use bootstrap samples of size  $0.8n = 800$  without replacement because replacement introduces high bias in rank-based estimation. The intervals are adjusted with a correction factor.

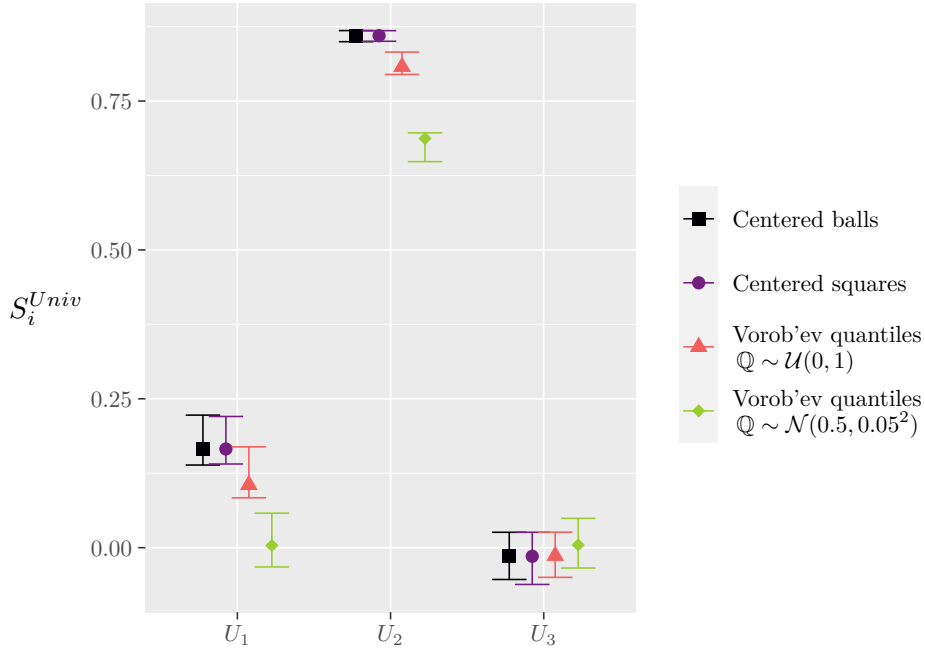
For the toy example, we use squares, balls, and Vorob'ev quantiles as test sets. The results are shown in Figure 3.8. The results are comparable when using centered balls, centered squares.  $U_2$  remains the most dominant input with an index value of about 85%, but  $U_1$  also affects the output with an index of about 15%.  $U_3$  is correctly detected as negligible. The results are similar with Vorob'ev quantiles, where the thresholds are taken uniformly at  $[0, 1]$ . The only difference



**Figure 3.6:** Estimation of  $S_V^i$  for the toy case with  $n = 100$  model evaluations per loop and a confidence interval estimated with 100 bootstrap resamples.



**Figure 3.7:** Estimation of  $S_V^i$  for the pollutant case with 1024 model evaluations and a confidence interval estimated with 100 bootstrap resamples.

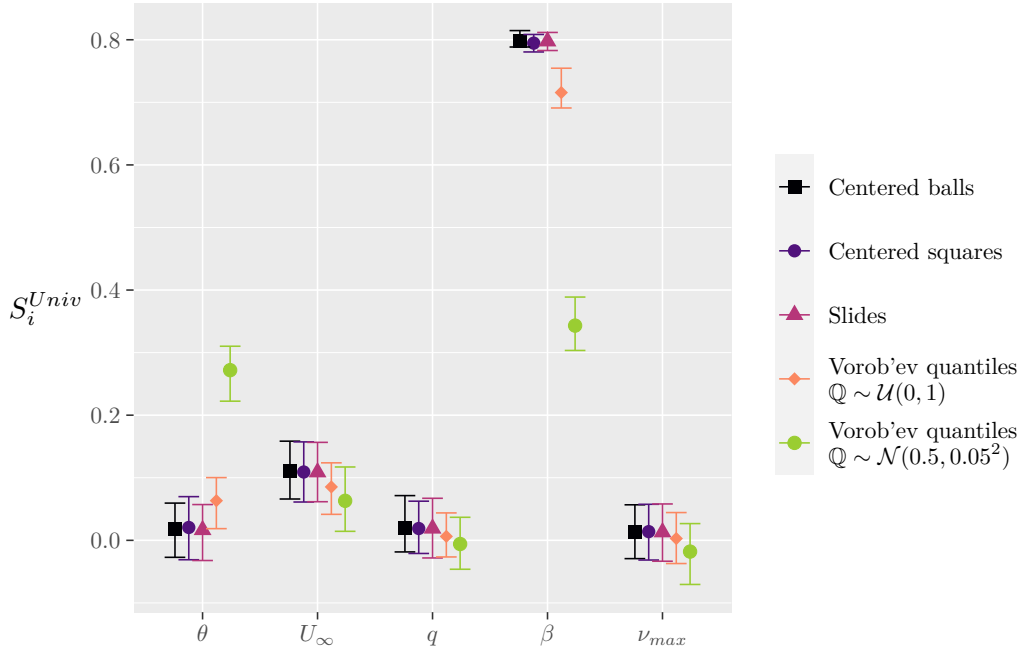


**Figure 3.8:** Estimation of the universal indices  $S_i^{Univ}$  for the toy case and for four different test sets and  $N_a = 100$ , with 1000 model evaluations. Confidence intervals are obtained with 100 bootstrap samples

is that part of the effect of  $U_1$  and  $U_2$  are interactions. In contrast, Vorob'ev quantiles with  $\mathbb{Q} = \mathcal{N}(0.5, 0.05^2)$  do not detect the effect of  $U_1$ . This difference shows that the choice of test sets is crucial, as it has a strong influence on the values of the indices. This will be discussed in comparison with the case of the pollutant concentration map. In terms of estimation, having access to the rank-based estimation allows to have a satisfactory index variability for the considered number of evaluations.

For the pollutant test case, we again use squares, balls, Vorob'ev quantiles with  $\mathbb{Q} \sim \mathcal{U}(0, 1)$  and  $\mathbb{Q} \sim \mathcal{N}(0.5, 0.05^2)$  and also slices along the third dimension to get levels of concentration test sets. The results are shown in Figure 3.9. The results are comparable when using centered balls, centered squares, slices, and Vorob'ev quantiles with uniform thresholds.  $\beta$  is the most dominant input with an index value of about 80%. The value of  $U_\infty$  is about 10%, and all other inputs have minimal impact. In contrast, with  $\mathbb{Q} \sim \mathcal{N}(0.5, 0.05^2)$ ,  $\beta$  and  $\theta$  are the most important inputs, and  $U_\infty$  also has some influence to a lesser extent. The output is clearly not affected by either  $q$  or  $\nu_{max}$ . This difference confirms that the choice of test sets is crucial, as it has a strong influence on the values of the indices.

Since the results of the Vorob'ev quantiles with  $\mathbb{Q} \sim \mathcal{N}(0.5, 0.05^2)$  are the closest to the generalized Sobol' indices of Table 3.2, this choice of test case seems more appropriate on this model. This can be explained by the fact that on this model, when the threshold is near 0.5, their geometry is similar to the realizations of  $\Gamma$ . Therefore, changes in  $\Gamma$  can be accurately detected by measuring the volume of the symmetric difference. Conversely, it may not be fully possible to detect all variations of the random set by comparing realisations of  $\Gamma$  with simple sets as squares, balls or slices. In contrast, with the toy function, simple sets seemed to be more efficient than Vorob'ev quantiles at detecting variations of the random sets. This is consistent because in this case the realizations of the random sets are simple enough and not much different in shape than squares or balls. The best



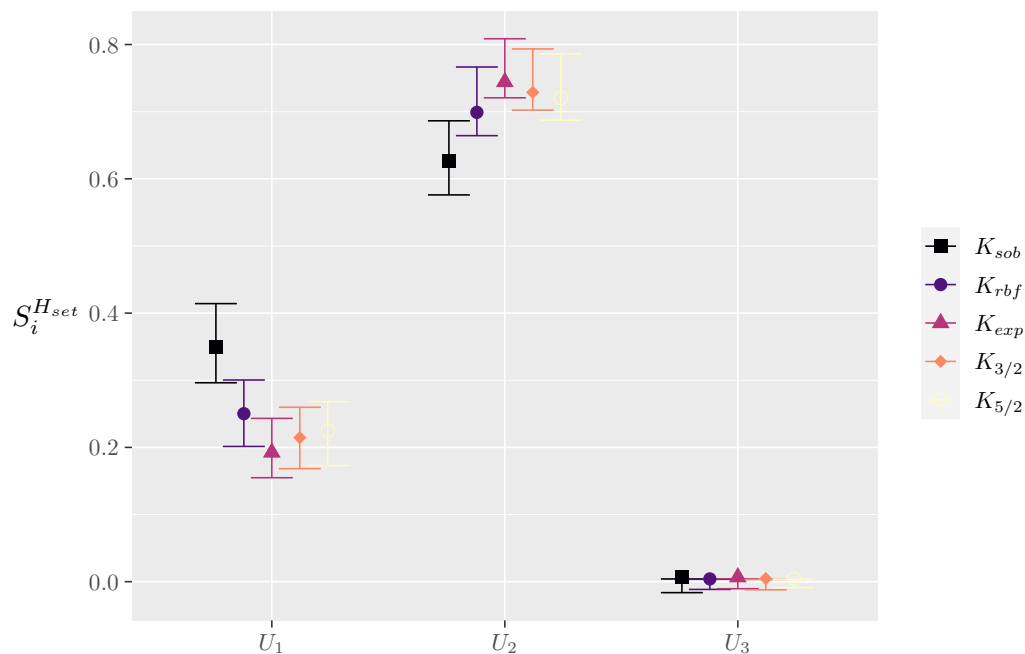
**Figure 3.9:** Estimation of the universal indices  $S_i^{\text{Univ}}$  for the pollutant case and for five different test sets and  $N_a = 100$ , with 1000 model evaluations. Confidence intervals are obtained with 100 bootstrap samples

choice of test sets seems to depend on the model, and in particular on its complexity. This makes our adaptation of the universal indices for sets very unsatisfactory to be usable for any model. To make them usable in practice, for any set-valued model, one should find the best choice of test sets (if it exists), or investigate another choice of scalar transformation (denoted  $T_a$  in [37]), different from the volume of the symmetric difference.

### 3.4.3 HSIC

We compute the indices for each input using a sample size of  $n = 1000$ . We use the five kernels suggested in Section 3.3.3. The value of  $\sigma$  is taken as the empirical mean of  $\hat{\lambda}_m(\Gamma^{(j)} \Delta \Gamma^{(l)})$  for  $j < l$ .

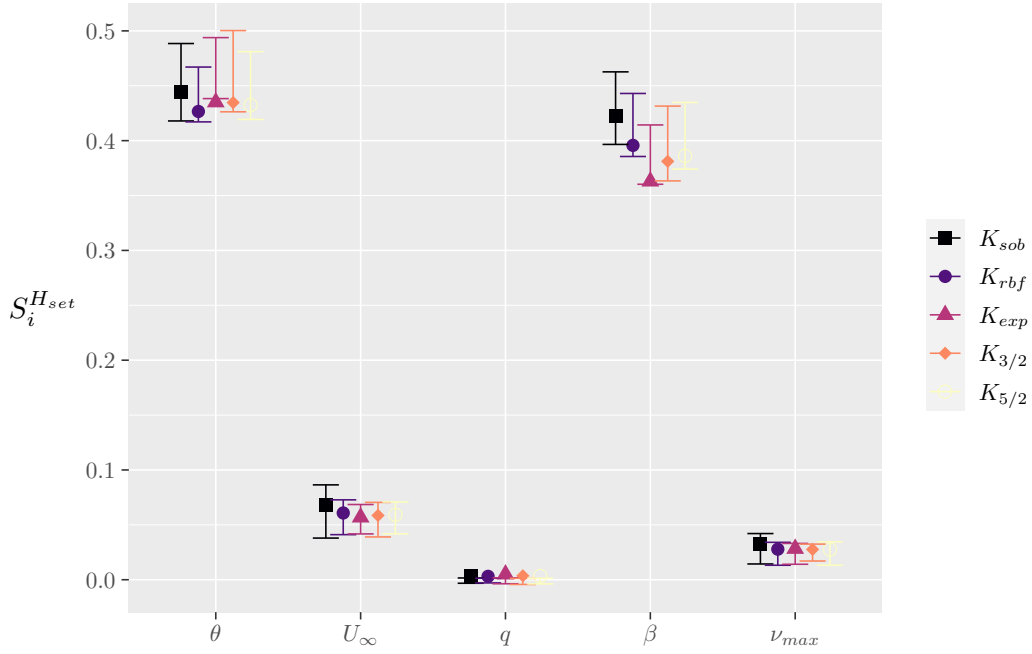
The results of the toy case are shown in Figure 3.10. We observe that the choice of the input kernel does not significantly affect the values of the indices: regardless of the kernel chosen,  $U_2$  remain the most influential variables with an index of about 70%,  $U_1$  has an index of about 25%, and  $U_3$  is correctly responsible for 0% of  $HSIC(\mathbf{U}, \Gamma)$ . These ranking results can also be used for screening purposes. For example, you can decide to keep only the most influential inputs. However, using HSIC-based indices ensures reliable screening by calculating the p-values of independence tests between an input and an output. This is done in Table 3.3, where the p-values for each input are computed by asymptotic estimation for each input kernel (see [154] for estimation details). If the p-value is greater than 0.05, then the input has no effect on the output. Specifically, in this case we observe that  $U_3$  is correctly detected as negligible. The chosen input kernel slightly changes the p-values in this case, but does not affect the screening results. In other test cases, however, the choice of kernel could affect the results, which is the main limitation of HSIC-based indices. Some research has also been done to help with kernel selection. For example, in [32], the authors suggest selecting the kernel that maximizes test performance. Without further investigation into the effects



**Figure 3.10:** Estimation of  $S_i^{H_{set}}$  for the toy case for five input kernels, 1000 model evaluations. Confidence intervals are obtained by bootstrapping with 100 resamples

|           | $U_1$        | $U_2$        | $U_3$ |
|-----------|--------------|--------------|-------|
| $K_{sob}$ | $< 10^{-16}$ | $< 10^{-16}$ | 0.62  |
| $K_{rbf}$ | $< 10^{-16}$ | $< 10^{-16}$ | 0.56  |
| $K_{exp}$ | $< 10^{-16}$ | $< 10^{-16}$ | 0.64  |
| $K_{3/2}$ | $< 10^{-16}$ | $< 10^{-16}$ | 0.58  |
| $K_{5/2}$ | $< 10^{-16}$ | $< 10^{-16}$ | 0.57  |

**Table 3.3:** Asymptotic estimation of the p-values of the HSIC-ANOVA-based independence test for the toy case with five input kernels and 1000 model evaluations and 100 bootstrap samples.



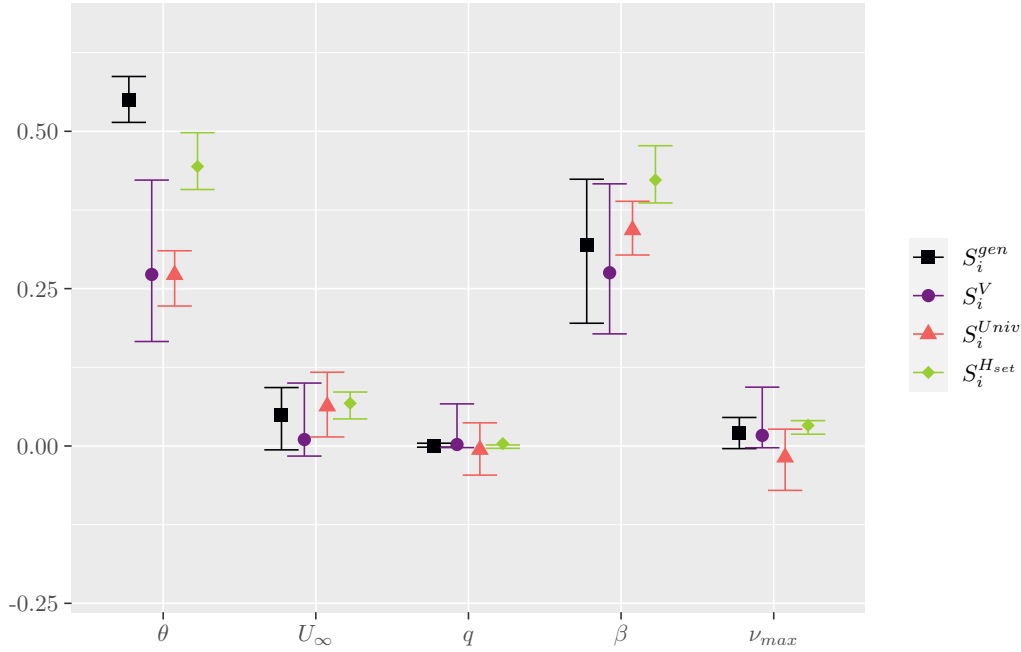
**Figure 3.11:** Estimation of  $S_i^{H_{set}}$  for the pollutant case for five input kernels, 1000 model evaluations. Confidence intervals are obtained by bootstrap with 100 resamples

of kernels, the Sobolev kernel may be preferred, as it does not require a choice of hyperparameters.

For the pollutant case, the results are plotted in Figure 3.11. We observe that the effect of the choice of the input kernel remains small again. Regardless of the kernel chosen,  $\beta$  and  $\theta$  remain the most influential variables (with about 40% each) and the other three inputs the least influential (with about 10%). In terms of screening, Table 3.4 gives the p-values for each input and each kernel. We observe that only  $q$  has a p-value above 0.05 for each kernel. In addition, a standard deviation for  $q$  is estimated by bootstrap analysis, which is shown in the second column. We confirm that despite this deviation, the p-values of  $q$  remain above 0.05. For the remaining four inputs, p-values below  $10^{-16}$  were obtained, and standard deviations of the same order indicate that these four inputs are influential. The chosen input kernel slightly alters the p-values in this case, but does not affect the screening results.

|           | $\theta$     | $U_\infty$   | $q$            | $\beta$      | $\nu_{max}$  |
|-----------|--------------|--------------|----------------|--------------|--------------|
| $K_{sob}$ | $< 10^{-16}$ | $< 10^{-16}$ | $0.74 \pm 0.6$ | $< 10^{-16}$ | $< 10^{-16}$ |
| $K_{rbf}$ | $< 10^{-16}$ | $< 10^{-16}$ | $0.76 \pm 0.6$ | $< 10^{-16}$ | $< 10^{-16}$ |
| $K_{exp}$ | $< 10^{-16}$ | $< 10^{-16}$ | $0.89 \pm 0.5$ | $< 10^{-16}$ | $< 10^{-16}$ |
| $K_{3/2}$ | $< 10^{-16}$ | $< 10^{-16}$ | $0.82 \pm 0.6$ | $< 10^{-16}$ | $< 10^{-16}$ |
| $K_{5/2}$ | $< 10^{-16}$ | $< 10^{-16}$ | $0.80 \pm 0.6$ | $< 10^{-16}$ | $< 10^{-16}$ |

**Table 3.4:** Asymptotic estimation of the p-values of the HSIC-ANOVA-based independence test for the pollutant case with five input kernels and 1000 model evaluations and 100 bootstrap samples.

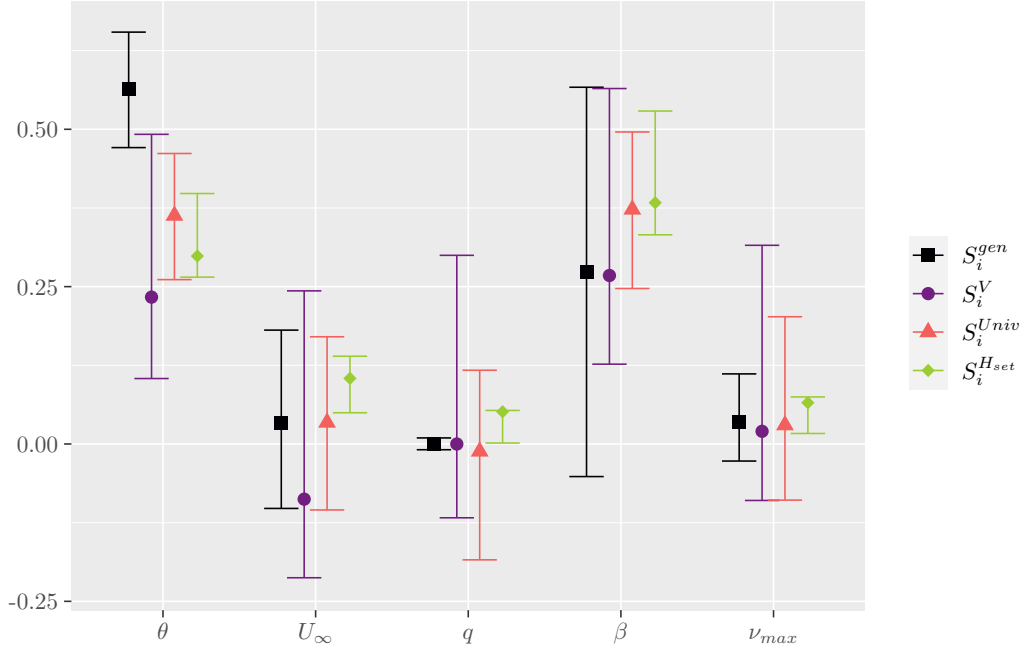


**Figure 3.12:** Comparison of the four indices with a total budget of  $n = 1000$  model evaluations. 100 bootstrap samples are used to estimate confidence intervals

### 3.4.4 Comparison

We compare the previously introduced indices on the pollutant concentration map model. The universal indices  $S_i^{Univ}$  are computed using Vorob'ev quantiles as test sets. For the HSIC-based indices, we use the Sobolev kernel. The indices are computed with a budget of about 1000 model evaluations.

In Figure 3.12, each index is plotted with its associated confidence intervals. It is important to note that we are plotting them together for comparison purposes only in terms of variability and ranking/screening results. The different indices cannot be compared in terms of values, as they all correspond to a completely different quantity. With the previous choices, the four indices rank  $\theta$  and  $\beta$  as the most influential inputs. However, there are differences in the results obtained between the indices in terms of screening and ranking. This is expected because each index quantifies a different effect. For example, HSIC-based indices quantify more than just the effect of the input on the output variance and look at the entire distributions. The HSIC is the only index that allows access to a reliable independence test, which classifies  $q$  as non-influential in the screening (according to the p-values given in Table 3.4). For the other three indices, a possible screening method is to select a threshold below which inputs are considered non-influential. For example, if a threshold of 10% is chosen, then  $U_\infty$ ,  $q$ , and  $\nu_{max}$  would be considered negligible. However, with a smaller threshold, it would be difficult to draw conclusions because some confidence intervals overlap. The four significant inputs identified by HSIC-based tests can be ranked confidently because their confidence intervals do not overlap. In contrast, the ranking of the remaining inputs varies for each index. Similar to  $S_i^{Hset}$ ,  $S_i^V$  places  $\theta$  and  $\beta$  on the same level. However,  $\theta$  is much more influential than  $\beta$  according to the generalized Sobol' indices, and the opposite is true for the universal indices. The ranking is theoretically justified for all indices except  $S_i^V$ , which lacks any decomposition.



**Figure 3.13:** Comparison of the four indices with a total budget of  $n = 100$  model evaluations. 100 bootstrap samples are used to estimate confidence intervals

Examining the confidence intervals, all four indices remain satisfactory with 1000 model evaluations, but the HSIC-based indices, and to a lesser extent the universal indices, have much less variability. To emphasize this difference, we create an identical plot on Figure 3.13 with only 100 model evaluations (10 in each loop for  $S_i^V$ ). This time, the HSIC-based indices can still be used because they have relatively small confidence intervals. The same conclusion in terms of screening and ranking can be reached with a single 100 sample. The universal indices can still be used in this case, but their variability makes it possible to have non-positive indices. As for the other two indices, it is important to note that they have not yet converged at all, making them unusable with only 100 model evaluations.

### 3.5 Discussion

In this paper, we introduced three indices adapted to set-valued outputs. They all allow to quantify the effect of uncertain inputs on a set as an output, but they are constructed with very different approaches. Based on random set theory, the first indices describe how much the inputs affect the Vorob'ev median deviation of the output. The universal indices quantify the impact of the inputs by measuring the variability of the random set by comparing it to test sets. And HSIC-based indices use kernels on sets to measure the dependence between the inputs and the output. In the case of a pollutant concentration map model, they were presented as an alternative to pointwise sensitivity analysis to reduce computational cost and implicitly account for model geometry and spatial correlations. Although we tested the indices on only two models, we can draw some conclusions.

Generalized Sobol indices ( $S_i^{gen}$ ) are characterized by their ability to rank factors by ANOVA. This method provides a clear hierarchy of importance among factors. However, its weaknesses lie

|                 | Ranking            | Screening             | Cost                                  | Comments                                      |
|-----------------|--------------------|-----------------------|---------------------------------------|-----------------------------------------------|
| $S_i^{gen}$     | ✓ ANOVA            | $\sim$ threshold      | $\sim (p+1)n$                         | ✗ Pointwise influence                         |
| $S_i^V$         | ✗ No decomposition | ✗ No screening method | ✗ $n^2$ (double loop)                 | ✓ Utilizes the geometry of the sets           |
| $S_i^{Univ}$    | ✓ ANOVA            | $\sim$ threshold      | $\sim n$ but big confidence intervals | ✗ Choice of the test sets                     |
| $S_i^{H_{set}}$ | ✓ ANOVA            | ✓ Independence test   | ✓ $n$ and small confidence intervals  | $\sim$ Choice of kernel<br>✗ Interpretability |

**Table 3.5:** Indices comparison

in its computational cost and in its pointwise influence, which may overlook nuanced geographic interactions.

Vorob'ev-based indices ( $S_i^V$ ) aims at explaining the output variability in terms of random sets but lack a decomposition property, making them unable to distinguish and compare the individual contributions of factors. Furthermore, the lack of a screening method makes them less efficient at identifying significant factors. Their current computational cost is also too high to be used. However, they fully exploit the geometry of the set-valued model, which makes them more interpretable.

Universal indices ( $S_i^{Univ}$ ) also use ANOVA for ranking and are similar to generalise Sobol' Indices in this respect. Thanks to rank-based estimation, they also require a small amount of model evaluations to be computed with sufficient precision. However, their major weakness lies in the selection of appropriate test sets, on which the results depend drastically. This is a problem for our application.

HSIC-based indices ( $S_i^{H_{set}}$ ) outperform in several aspects, including ANOVA ranking, independence tests for screening. They require a very small amount of data to reach an acceptable precision. However, they lack of interpretability, since they don't represent a part of the variance, but some part of a distance between probability distributions. Interactions between variables can be difficult to interpret in the HSIC decomposition. The choice of kernel can also be questioned, as it may have a greater impact with other models. Overall, HSIC-based index seems to be the most efficient sensitivity index for set-valued models.

The previous strengths and weaknesses of each index are summarized in Table 3.5.

The introduction of sensitivity indices for sets was originally conceived for their application to maps. However, their utility extends beyond maps and can be effectively applied to any model with set-valued outputs, such as sets of feasible points in mechanical engineering. While we specifically addressed the two-dimensional case to directly handle maps, it is also worth noting that these indices can be used with  $\mathcal{X}$  of any finite dimension.

**DATA AVAILABILITY STATEMENT** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

**FUNDING INFORMATION** This research was conducted with the support of the consortium in Applied Mathematics [CIROQUO](#), gathering partners in technological research and academia in the development of advanced methods for Computer Experiments.

**CONFLICT OF INTEREST** The authors report there are no competing interests to declare.

**ETHICAL STATEMENT** This research was conducted with integrity and adherence to ethical standards. The authors affirm that the work presented is original, and any external sources or contributions have been properly cited. The research does not involve any human or animal subjects, and thus, no ethical approval was required.

## Author biographies



**Noé Fellmann** obtained a Master's degree in Applied Mathematics from Lyon 1 University and a diploma from École Centrale de Lyon. Since 2021, Noé Fellmann has been pursuing a PhD on sensitivity analysis for robust optimization. His work focuses on set-valued outputs sensitivity analysis and the incorporation of sensitivity analysis steps in robust Bayesian optimization methods.



**Mathis Pasquier** obtained an engineering degree from École des Mines de Saint-Étienne (2017-2020) and a Master of Science in Computational Fluid Mechanics from Cranfield University (2019-2020). He also pursued master's level studies in Applied Mathematics at Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (2018-2019). From 2020 to 2023, he completed a PhD on the lattice Boltzmann method and uncertainty analysis for turbulent pollutant dispersion in urban environments.



**Celine Helbert** obtained her PhD in 2005 and worked for four years as a research engineer in Applied Mathematics at École nationale supérieure des mines de Saint-Etienne. She then joined the University of Grenoble as Assistant Professor. Since 2012, Céline Helbert teaches Probability and Statistics at École Centrale de Lyon. Her research activity focuses on the analysis and exploration of large computer codes developed in close collaboration with industrial partners (Renault, EDF, CEA).



**Christophe Blanchet-Scalliet** is an alumnus of École normale supérieure de Cachan and holds a PhD from the University of Evry - Val d'Essonne. She was Assistant Professor at the Institute of Technology in Statistics and Business Intelligence (IUT STID) from the University of Nice - Sophia Antipolis. Since 2007, she is Assistant Professor in Probability within the Department of Mathematics and Computer Science at École Centrale de Lyon. Her research mainly focuses on stochastic processes and filtration enlargement, as well as applications to mathematical finance and risk management.



**Delphine Sinoquet** holds a Master's degree in Numerical Analysis from Paris 6 University and a PhD in Applied Mathematics from Paris 13 University, specializing in the inversion problem of seismic tomography. From 1994 to 2003, she worked as a research engineer in the Geophysics Division of IFPEN, focusing on inverse problems for seismic imaging. Since 2003, she has been a research engineer in the Applied Mathematics Department of IFPEN, with a primary focus on numerical optimization and uncertainty quantification.



**Adrien Spagnol** pursued his Ph.D. from 2017 to 2020 at Ecole des Mines de St. Etienne on "Kernel-based Sensitivity Indices for High-dimensional Optimization Problems." He is now a research engineer in the Applied Mathematics Department of IFP Energies Nouvelles.

# II

## **Sensitivity Analysis for Robust Bayesian Optimization**



# Introduction

In this second part of the thesis, we return to robust optimization and aim to find  $x^*$ , defined as

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E}[f(x, U)] \text{ where } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P}[g(x, U) \leq 0] \geq \alpha\}. \quad (3.6)$$

where  $f$  and  $g$  are expensive black-box functions,  $x = (x_1, \dots, x_d) \in \mathcal{X} \subset \mathbb{R}^d$  are the design inputs,  $U = (U_1, \dots, U_p) \in \mathcal{U} \subset \mathbb{R}^p$  are uncertain inputs of known probability distribution  $\mathbb{P}_U$  and  $\alpha \in [0, 1]$ . Such a problem is hard to solve, especially in the context of a time-consuming objective function  $f$  and constraint functions  $g = (g_1, \dots, g_s)$ , which limits to a small evaluation budget (in the following fixed to 10 times the input dimension  $d + p$ ).

Bayesian optimization is an efficient solution to costly black-box optimization, relying on surrogates of  $f$  and  $g$  fitted to a design constructed along the optimization. In the case of the optimization problem of Eq. 3.6, the robust Bayesian optimization algorithm EFISUR, introduced in [33] and described in Section 1.2.3.b, allows to solve (3.6) efficiently. The algorithm is described in detail in Algorithm 1 and illustrated in Figure 3.14.

---

## Algorithm 1 EFISUR

---

- 1: Design of Experiments (DoE) in the joint space  $\mathcal{X} \times \mathcal{U}$ :  $D^{(t)} = \{(x^k, u^k), k = 1, \dots, t\}$
  - 2: Calculate simulator responses:  $f^{(t)} = (f(x^k, u^k))_{k=1}^t$  and  $g^{(t)} = (g(x^k, u^k))_{k=1}^t$
  - 3: **while**  $t \leq$  maximum budget **do**
  - 4:   Fit GPs of  $f$  and  $g$ :  $F_{(x,u)}^{(t)}$  and  $G_{(x,u)}^{(t)}$
  - 5:   Calculate  $Z_x^{(t)}$  and  $C_x^{(t)}$
  - 6:    $x^{t+1} = \arg \max_{x \in \mathcal{X}} \text{EFI}_{Z^{(t)}, C^{(t)}}(x)$
  - 7:    $u^{t+1} = \arg \min_{u \in \mathcal{U}} S(x^{t+1}, u)$
  - 8:   Calculate simulator responses at the next point  $(x^{t+1}, u^{t+1})$
  - 9:   Update the DoE and  $t \leftarrow t + 1$
  - 10: **end while**
  - 11:  $x^* = \arg \min_{x \in \mathcal{X}^{(t)}} \mathbb{E}(Z_x^{(t)})$  s.t.  $\mathbb{E}(C_x^{(t)}) \leq 0$
- 

We recall that

- $F^{(t)}$  and  $G^{(t)}$  are Gaussian Processes that approximate  $f$  and  $g$ .
- $Z_x^{(t)}$  and  $C_x^{(t)}$  are processes that approximate  $\mathbb{E}[f(x, U)]$  and  $\alpha - \mathbb{P}[g(x, U) \leq 0]$  defined as

$$Z_x^{(t)} = \mathbb{E}_U \left[ F_{(x,U)}^{(t)} \right],$$

and

$$C_x^{(t)} = \alpha - \mathbb{P}_U(G_{(x,U)}^{(t)} \leq 0).$$

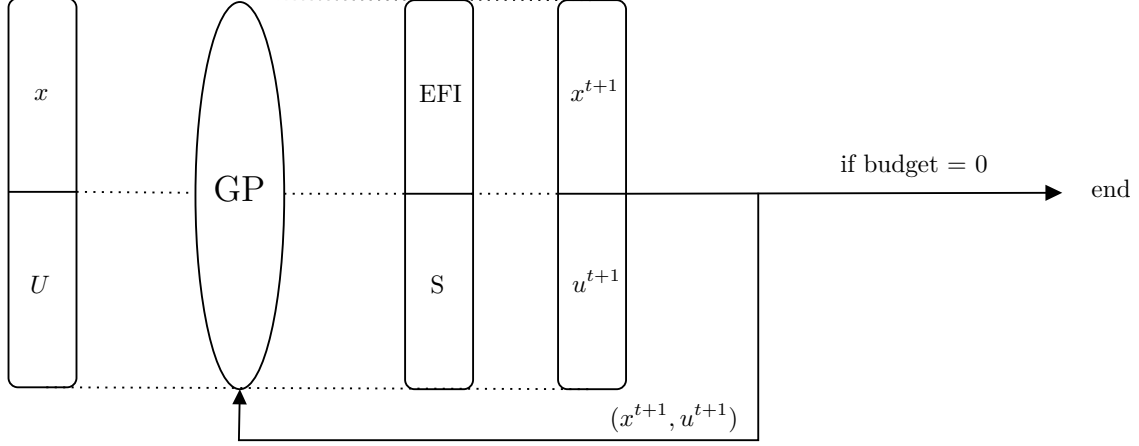


Figure 3.14: EFISUR diagram

- At each iteration, a region of interest is determined by finding  $x^{targ}$  that maximizes the Expected Feasible Improvement  $\text{EFI}_{Z^{(t)}, C^{(t)}}(x)$ , defined as

$$\text{EFI}_{Z^{(t)}, C^{(t)}}(x) = \mathbb{E} \left[ \max(z_{min}^{feas, (t)} - Z_x^{(t)}, 0) \right] \mathbb{P} \left[ C_x^{(t)} \leq 0 \right],$$

where

$$z_{min}^{feas, (t)} = \min_{1 \leq i \leq t} \mathbb{E} \left[ Z_{x^i}^{(t)} \right] \text{ s.t. } \mathbb{E} \left[ C_{x^i}^{(t)} \right] \leq 0.$$

- $x^{t+1}$  is fixed at  $x^{targ}$  and  $u^{t+1}$  is chosen by minimizing  $S(x^{t+1}, u)$ , a proxy of the feasible improvement variance at  $x^{targ}$  when adding  $(x^{t+1}, u)$  to the design, defined as

$$S(x, u) = \text{Var} \left[ \max(z_{min}^{feas, (t)} - Z_{x^{targ}}^{(t)}, 0) | F_{(x, u)}^{(t)} = f(x, u) \right] \\ \times \mathbb{E}_U \left[ \text{Var}_\omega \left[ \mathbb{1}_{G_{(x^{targ}, U)}^{(t)} \leq 0} | G_{(x, u)}^{(t)} = g(x, u) \right] \right].$$

EFISUR algorithm is a powerful solution for robust optimization when the functions  $f$  and  $g$  are expensive, as it can be done without many model evaluations. However, as the input dimension increases, EFISUR may begin to struggle. The main reasons (presented in section 1.2.4 for any BO algorithm) are the challenges of fitting GPs and optimizing acquisition criteria in high input dimension, as explained in [11] and [95]. To overcome these difficulties, we propose different ways to reduce the input dimension using sensitivity analysis methods outside or within EFISUR.

**Offline** approaches are based on dimension reductions outside EFISUR, so that the GPs are fitted in low dimension. They can be compared to the work without uncertainties of [17] and [157], where sensitivity indices (respectively interval-based sensitivity index and target HSIC-based indices) are computed to screen the inputs before carrying out the optimization in small dimension.

**Online** approaches go inside EFISUR and are based on dimension reduction by Sensitivity Analysis at each iteration so that the acquisition criteria optimizations can be done in small dimension. The proposed approaches are inspired by the work of [156], which proposes to reduce the dimension by SA at each iteration in a generic (but without uncertainty) Bayesian algorithm. In this case, however, we will see that the GPs are still fitted in the original high dimension.

Our methods, described by four algorithms, are first presented in Chapter 4. A toy function for robust optimization is then constructed and analyzed in Chapter 5. The four algorithms are finally tested and compared on this toy function and on a test case.

# 4

## Sensitivity Analysis and EFISUR : Methodology

---

|            |                                                                                |           |
|------------|--------------------------------------------------------------------------------|-----------|
| <b>4.1</b> | <b>Dimension reduction by Offline Sensitivity Analysis</b>                     | <b>78</b> |
| 4.1.1      | Off-SA : One SA step before EFISUR                                             | 78        |
| 4.1.1.a    | Method and algorithm                                                           | 78        |
| 4.1.1.b    | Discussion and perspectives                                                    | 80        |
| 4.1.2      | Off-seq-SA: Multiple SA steps outside EFISUR iterations                        | 82        |
| <b>4.2</b> | <b>Online Sensitivity Analysis in EFISUR algorithm</b>                         | <b>84</b> |
| 4.2.1      | On-SA: Dimension reduction by SA on the acquisition criteria optimization      | 84        |
| 4.2.1.a    | Method and algorithm                                                           | 85        |
| 4.2.1.b    | Discussion and perspectives                                                    | 85        |
| 4.2.2      | On-2steps-SA: Two-steps optimization of the acquisition functions guided by SA | 86        |

---

The aim of this chapter is to present adaptations of the EFISUR algorithm using sensitivity analysis to make it usable when the input dimension is large. This chapter presents two main methods. The first one is based on a dimension reduction by sensitivity analysis outside EFISUR or in-between iterations of the algorithm (Section 4.1). The second one consists in simplifying or guiding the optimization of acquisition functions by sensitivity analysis (Section 4.2).

As previously mentioned, the EFISUR algorithm, like any Bayesian optimization method based on Gaussian process modeling, suffers from the curse of dimensionality. As the dimension increases, the Gaussian process regression of the functions of interest becomes more difficult. It requires many more model evaluations to approximate the real models well enough. In EFISUR, the Gaussian modeling is done in the joint space, which makes it even more affected by the dimension increase. However, the larger the dimension, the more there are inputs that may have a negligible impact on the output or on the optimizations. Such inputs may then not be needed at all for the meta-modeling and its optimization.

In Section 4.1.1 we propose to use a goal-oriented sensitivity analysis (GOSA) to detect negligible inputs and run EFISUR in reduced dimension. This algorithm is called **Off-SA**. Since a single SA step may not be sufficient and may be risky, we also propose a second algorithm called **Off-seq-SA**

which includes several intermediate steps of dimension reduction by sensitivity analysis (Section 4.1.2).

In addition to the dimensionality issues of the Gaussian process modeling, EFISUR suffers from difficulties regarding the optimization of the acquisition functions. These optimizations require more iterations when the dimension is large and can also be very difficult to perform since acquisition functions can be hard functions to optimize. For this reason, we propose to incorporate dimension reduction inside EFISUR. In the resulting algorithm, called **On-SA**, an online sensitivity analysis is conducted at every iteration of EFISUR, before each of the two acquisition function optimizations (Section 4.2.1). The acquisition functions optimization should then be much easier in reduced dimension. The SA can however remove inputs when they seem negligible at the beginning, but that become important when approaching the optimum. To tackle this problem, **On-2steps-SA** includes another SA step on the acquisition functions to refine the selected point.

Whatever the method, we aim to keep the number of model evaluations similar to what would be necessary to perform EFISUR. The budget is roughly 10 times the input dimension, to give an idea. Thus, the algorithms are designed to keep the number of model evaluations small. Every choice made to design the algorithm will be discussed after introducing each algorithm. Choice of numerical parameters will however be left to be discussed in the next chapter.

## 4.1 Dimension reduction by Offline Sensitivity Analysis

In this section, we present two adaptations of EFISUR based on an input dimension reduction by GOSA outside the optimization loop.

### 4.1.1 Off-SA : One SA step before EFISUR

The objective and constraint functions  $f$  and  $g$  are defined in the joint input space  $\mathcal{X} \times \mathcal{U}$  of dimension  $d + p$ , potentially large. Among these design inputs  $(x_1, \dots, x_d)$  and uncertain inputs  $(U_1, \dots, U_p)$ , some can have no impact on  $f$  and  $g$  leading to a much smaller effective dimension. Moreover, some inputs may only affect the outputs without impacting the optimization results. For example, inputs that change the value of  $f$  in areas where the constraint isn't verified ( $g > 0$ ), or change the feasibility of points that always correspond to a high  $f$ , are inputs that are not necessary to keep in order to optimize the functions. Moreover, if the optimization starts with a step of metamodeling, fixing these inputs can greatly simplify the metamodeling step and its precision. For these reasons, we propose to perform a GOSA step before EFISUR, so that it can be performed in a lower dimension. Of course, this first SA step to reduce the dimension has a simulation cost and part of the computational budget needs to be used. Less model evaluations will then remain for EFISUR algorithm itself but it can be sufficient to reach the optimum in a low dimension space. Computational time gain to fit the GPs and to optimize the acquisition criteria are expected but can be relatively of more or less importance depending on the computational cost of the true model. Thus, the main question of this section is: What kind of low-cost SA on the design variables and on the uncertain variable should be used to reduce the dimension before EFISUR?

#### 4.1.1.a Method and algorithm

For both design and uncertain inputs, we propose to conduct a screening by GOSA to select the inputs to keep for the optimization. However, any SA step requires an important number of model evaluations which is not affordable in our case. Therefore, we propose to use GP regression to estimate the indices, replacing  $f$  and  $g$  by their predicted mean  $\mu_{F^{(n_{SA})}}$  and  $\mu_{G^{(n_{SA})}}$  trained

on a part of the budget denoted  $n_{SA}$ . Recalling that  $Z_x^{(n_{SA})} = \mathbb{E}_U(F_{x,U}^{(n_{SA})})$  and  $C_x^{(n_{SA})} = \alpha - \mathbb{P}_U(G_{x,U}^{(n_{SA})} \leq 0)$ , we also have access to estimates of  $\mathbb{E}[f(\cdot, U)]$  and of  $\alpha - \mathbb{P}[g(\cdot, U) \leq 0]$  through  $\mu_{Z^{(n_{SA})}}$  and  $\mu_{C^{(n_{SA})}}$ .

For the design inputs, we are interested in detecting which  $X_i$  are important to reduce  $\mathbb{E}_U[f(X, U)]$  while  $\mathbb{P}_U[g(X, U) \leq 0] \geq \alpha$ . We propose to use Target SA (TSA) indices and Conditional SA (CSA) indices (presented in Section 1.3.1 and initially introduced in [97]), and in particular T-HSIC and C-HSIC on the model defined as

$$X = (X_1, \dots, X_d) \longrightarrow (E, P) = (\mu_{Z^{(n_{SA})}}(X), \alpha - \mu_{C^{(n_{SA})}}(X)).$$

where  $\mathbb{E}_U[f(X, U)]$  and  $\mathbb{P}_U[g(X, U) \leq 0]$  are replaced by their estimation through the GPs means. This will allow to detect which  $X_i$  is necessary to reach the feasible optimum (through T-HSIC) but also those that are important within that range (with C-HSIC). Compared to [97], we have a vectorial output and some adaptation is necessary.

For the TSA, as done in [157] and mentioned in Section 1.3.2, adaptation of T-HSIC for vectors is straightforward if the goal-oriented transformation is scalar-valued. Mixing the proposed transformations of [97] and [157], we propose to use the Goal-oriented transformation  $h$  defined by

$$h(e, p) = \exp\left(-\frac{\max(e - q, 0)}{0.2\sigma_E}\right) \exp\left(-\frac{\max(\alpha - p, 0)}{0.2\sigma_P}\right), \quad (4.1)$$

where  $\sigma_E$  and  $\sigma_P$  are estimations of the standard deviations of  $E$  and  $P$  and  $q$  is fixed at the 10% quantile of  $\mu_{Z^{(n_{SA})}} \mathbb{1}_{\mu_{C^{(n_{SA})}} \leq 0}$ . The 0.2 values are chosen relying on advises given in [97].  $h$  is a goal-oriented transformation, since it will be equal to 1 in areas where  $E$  takes low values and  $P \geq \alpha$ , and will tend to 0 otherwise. This transformation is simply the product of two scalar-valued outputs transformations proposed in [97]. This also correspond to the  $L_2$ -distance to the region of interest taken equal to the sum of the distances to the regions of interest of each output. This leads to the computation of

$$\text{T-HSIC}_k^h(X_i, (E, P)) = \text{HSIC}_k(X_i, h(E, P)),$$

and associated p-values  $p_{X_i}^{\text{TSA}}$  for each  $X_i$ .  $k$  is a product of 1-d classical kernels and can be adapted (for the input kernels) if ANOVA indices are computed.

For the CSA, we will use a threshold indicator as it is the only case where p-values are accessible (see [97]). The adaptation to two outputs can then simply be done using the product of two indicators:

$$w(e, p) = \mathbb{1}_{e \leq q} \mathbb{1}_{p \geq \alpha},$$

where  $q$  is the same quantile. Then the indices are defined as

$$\text{C-HSIC}_k^w(X_i, (E, P)) = \text{HSIC}_{k, (X_i, E, P) \sim \mathbb{P}_{(X_i, E, P)}^w}^w(X_i, (E, P)),$$

where, in the case of indicators,  $\mathbb{P}_{(X_i, E, P)}^w = \mathbb{P}_{(X_i, E, P) | (E \leq q, P \geq \alpha)}$ . Associated p-values are denoted  $p_{X_i}^{\text{CSA}}$ . The 2-dimensional output kernel is taken as the product of two 1 dimensional kernels.

For the uncertain inputs, we will use the indices presented in chapters 2 and 3 to quantify effects of uncertain inputs on excursion sets. Since the most promising ones were  $S_j^{H_{set}}$  indices (i.e. HSIC-ANOVA indices of excursion sets), we propose to use them to detect which  $U_j$  should be kept for optimization. Considering the model

$$U = (U_1, \dots, U_p) \longrightarrow \Gamma_{\mu^{(t)}} = \{x \in \mathcal{X}, \mu_{F^{(n_{SA})}}(x, U) \leq q, \mu_{G^{(n_{SA})}}(x, U) \leq 0\}.$$

we compute  $S_j^{H_{set}}$  and associated p-values  $p_{U_j}$  for each  $U_j$ . Any ANOVA input kernels can be used.

Both SA steps allow to screen the design and the uncertain inputs by keeping the inputs which p-value is below a risk of type I error  $\alpha_I$  preably fixed (often at 5% or 1%) to obtain  $x_A$  and  $u_B$  influential inputs where  $A$  and  $B$  are defined as

$$A = \{i, \min(p_{X_i}^{TSA}, p_{X_i}^{CSA}) \leq \alpha_I\},$$

and

$$B = \{j, p_{U_j} \leq \alpha_I\}.$$

$x_{-A}$  and  $u_{-B}$  are then fixed at random and EFISUR is carried out in the lower dimension  $d' \times m'$  with the remaining budget, where  $d' = |A|$  and  $m' = |B|$ . The associated GPs in the reduced dimensions are denoted  $F_{AB}$ ,  $G_{AB}$ ,  $Z_A$  and  $C_A$ , along with  $S_{AB}$  the proxy  $S$  relying on the GPs of reduced dimension.  $x_{-A}$  are fixed at values drawn uniformly on their domain and  $U_{-B}$  at the value drawn according to their probability distribution  $\mathbb{P}_{U_{-A}}$ . The presented method is detailed in **Off-SA** algorithm and illustrated in Figure 4.1.

---

**Algorithm 2** SA step before EFISUR: **Off-SA**

---

- 1: Design of Experiments (DoE) in the joint space  $\mathcal{X}_d \times \mathcal{U}_p$ :  $D^{(n_{SA})} = \{(x^k, u^k), k = 1, \dots, n_{SA}\}$
  - 2: Calculate simulator responses:  $f^{(n_{SA})} = (f(x^i, u^k))_{k=1}^{n_{SA}}$  and  $g^{(n_{SA})} = (g(x^k, u^k))_{k=1}^{n_{SA}}$
  - 3: Fit GPs for  $f$  and  $g$ :  $F_{(x,u)}^{(n_{SA})}$  and  $G_{(x,u)}^{(n_{SA})}$
  - 4: Compute  $Z_x^{(n_{SA})} = \mathbb{E}_U[F_{(x,U)}^{(n_{SA})}]$  and  $C_x^{(n_{SA})} = \alpha - \mathbb{P}_U(G_{(x,U)}^{(n_{SA})} \leq 0)$
  - 5: Estimate T-HSIC<sup>*h*</sup>( $X_i, (\mu_Z(X), \mu_C(X))$ ), C-HSIC<sup>*w*</sup>( $X_i, (\mu_Z(X), \mu_C(X))$ ),  $p_{X_i}^{TSA}$ , and  $p_{X_i}^{CSA}$  for  $1 \leq i \leq d$  with  $X_i \sim \mathcal{U}(\mathcal{X}_i)$
  - 6: Estimate  $S_j^{H_{set}}$  and  $p_{U_j}$  for  $1 \leq j \leq p$  with  $U_j \sim \mathbb{P}_{U_j}$
  - 7: Screen the inputs :  $A = \{i, \min(p_{X_i}^{TSA}, p_{X_i}^{CSA}) \leq \alpha_I\}$  and  $B = \{j, p_{U_j} \leq \alpha_I\}$
  - 8: Fix  $x_{-A}$  and  $u_{-B}$  randomly
  - 9: **EFISUR in dimension  $d' \times m'$** :
  - 10: Design of Experiments (DoE) in the joint space  $\mathcal{X}_A \times \mathcal{U}_B$ :  $D_{AB}^{(t)} = \{(x_A^k, u_B^k), k = 1, \dots, t\}$
  - 11: Calculate responses:  $f^{(t)} = (f(x_A^k, x_{-A}, u_B^k, u_{-B}))_{k=1}^t$  and  $g^{(t)} = (g(x_A^k, x_{-A}, u_B^k, u_{-B}))_{k=1}^t$
  - 12: **while**  $t \leq \text{budget} - n_{SA}$  **do**
  - 13:     Fit the GPs in reduced dimension  $F_{AB}^{(t)}$  and  $G_{AB}^{(t)}$  on the design  $D_{AB}^{(t)}$
  - 14:     Calculate  $Z_A^{(t)}$  and  $C_A^{(t)}$
  - 15:      $x_A^{t+1} = \arg \min_{x_A \in \mathcal{X}_A} \text{EFI}_{Z_A^{(t)}, C_A^{(t)}}(x_A)$
  - 16:      $u_B^{t+1} = \arg \min_{u_B \in \mathcal{U}_B} S_{AB}(x_A^{t+1}, u_B)$
  - 17:     Calculate simulator responses at the next point  $(x_A^{t+1}, x_{-A}, u_B^{t+1}, u_{-B})$
  - 18:     Update the DoE and  $t \leftarrow t + 1$
  - 19: **end while**
- 

#### 4.1.1.b Discussion and perspectives

This proposed method and algorithm have been designed with several choices.

**CHOICE OF THE METAMODEL FOR THE SA** We decided to use Gaussian process modeling for the sensitivity analysis. This is mainly because using the same type of Gaussian processes as those used in the EFISUR allows an easy access to  $\mathbb{E}_U[f(X, U)]$  and  $\mathbb{P}_U[g(X, U) \leq 0]$  through the processes  $Z$  and  $C$ . However, other meta-models could be used here. Since this initial metamodeling is done

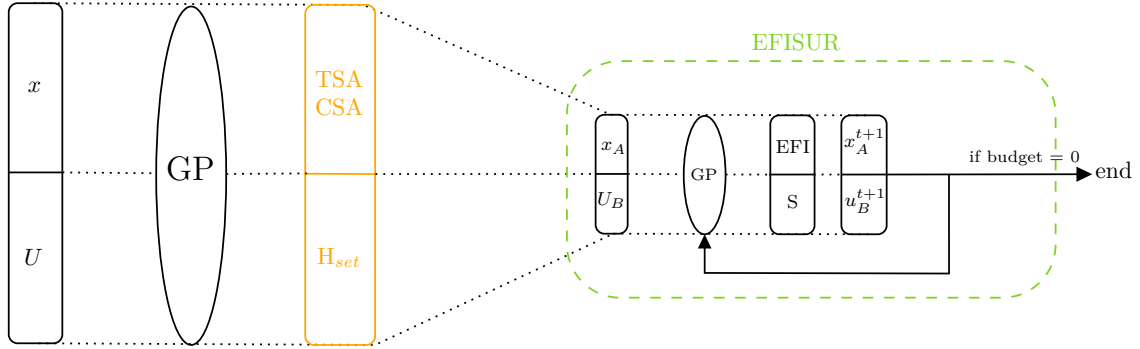


Figure 4.1: Off-SA diagram

in high dimension, it would be interesting to choose a type of metamodel that is less affected by the curse of dimensionality. As the metamodel is only used for the SA step, it may be pertinent to choose a metamodel adapted to the computation of the SA indices. For example, Sobol' indices can directly be obtained by Polynomial Chaos Expansion [162]. In [100], the use of GP for Sobol' indices estimation is studied. To our knowledge, HSIC estimation through GPs has not been developed yet, mostly because classic HSIC don't require a lot of model evaluations. In our case, mainly for simplicity, we decided to compute the indices using the GP means. Alternatives could be to compute random SA indices on the GP paths, or information about the GP uncertainty could be used to improve the estimation and robustness of the indices as done for Sobol' indices.

**CHOICE OF THE SA INDICES** For the uncertain inputs, we use  $H_{set}$  which are designed to quantify the impact of uncertain inputs on excursion sets. Alternative indices from chapter 3 can be used, but they turned out to be less efficient than the HSIC indices. For the design inputs, we decided to use TSA- and CSA-HSIC from [97], where both Sobol indices and HSIC-based indices are presented. The HSIC are preferred because they are adapted to screening. Screening with total Sobol indices could also be done. The constant  $q$  is fixed at the 10% quantile of  $\mu_{Z(n_{SA})} \mathbb{1}_{\mu_{C(n_{SA})} \leq 0}$  to have enough non-zero points so that the index can be estimated. For both SA, we decided to keep the inputs which p-value is below a fixed risk. This fixed risk should be decided whether we prefer to reduce a lot the dimension at the risk of missing influential inputs or to potentially keep negligible inputs and remain in a bigger dimension. Alternatively to screening, it is also possible to use a ranking approach. By ranking the inputs according their HSIC-ANOVA index, we can decide to keep a fixed number of inputs by taking the most influential ones. Without fixing a number of kept inputs, we can also add inputs up to a given percentage in the HSIC-ANOVA decomposition.

**CHOICE OF THE WAY TO FIX THE REMOVED INPUTS** A last choice was to fix the removed inputs  $x_{-A}$  and  $u_{-B}$  at random (uniformly for the  $x$ 's and according to their known laws for the  $u$ 's). This was justified by the fact that the SA results show these inputs to be negligible, so their value shouldn't be important. It is also possible to fix them according to some criteria. For example, in [156], the removed inputs are fixed according to different strategies, using the value at the best the point among the previous model evaluations. Fixing them to their mean or other known values of interest may also be appropriate.

A single SA step is proposed in **Off-SA** but may be insufficient. First, the SA results are computed on a GP so they can be inaccurate. Relying completely on them can be risky. Second, at the beginning of the algorithm we have no information about the domain of interest of the

optimization so the GOSA can miss inputs that become active only near the optimum. Take for example a "valley-shaped functions"  $f$  defined with three inputs  $x_1$ ,  $x_2$  and  $x_3$  where

- $x_1$  has a very strong impact on minimizing  $f$  which is minimal for  $x_1 = 0$ ,
- $x_2$  has a very limited overall impact,
- $x_3$  only affects  $f$  when  $x_1 \approx 0$ .

The proposed algorithm might only detect  $x_1$  as influential. Then, the low-dimension GP and the iterations will then only use evaluations of the form  $f(x_1^k, x_2, x_3)$ , i.e. where  $x_2$  and  $x_3$  are fixed. Then, the optimal point in the third coordinate won't be reached. Therefore, it may be interesting and efficient to perform other SA steps after some EFISUR iterations. Information provided by some EFISUR iterations could then be used to adjust the SA and would allow to more efficiently detect, inputs that become important only when approaching optimal region.

#### 4.1.2 Off-seq-SA: Multiple SA steps outside EFISUR iterations

In this part, we extend the previous method by adding multiple SA steps between series of EFISUR iterations. By doing so, the information obtained after some EFISUR iterations is used to adjust the screening of the inputs.

In the first algorithm **Off-SA**, the new model evaluations for the low-dimensional GPs and for iterations are made for fixed  $x_{-A}$  and  $u_{-B}$ , i.e. of the form  $f(x_A^k, x_{-A}, u_B^k, u_{-B})$ . If a new SA step is conducted, and inputs in  $A$  or  $B$ , say  $x_i$  and  $u_i$ , are no longer kept. Then the previous evaluations made to fit the low-dimensional GPs and made during the iterations cannot be used in the new low dimension as they are done for varying  $x_i$  and  $u_j$ . To overcome this issue, two solutions appear. The first one is to impose that previously selected inputs must be kept in all the next SA steps, and thus the GPs will be of increasing dimension. However this might increase the GPs dimension when it is not necessary. An alternative that we propose here is to use **noisy** Gaussian processes. During the whole algorithm, the negligible inputs won't be fixed but randomly sampled and their effect will be contained in the noise of reduced dimension GPs. Before presenting the algorithm, we introduce our use of noisy Gaussian processes and, in particular, how we propose to deal with the noise using the SA results.

**GAUSSIAN PROCESS WITH NOISE FITTED USING GOSA** As mentioned in section 1.2.1, noisy gaussian processes are gaussian processes with significant noise (instead of a very small nugget used to facilitate numerical computations). In our case, we have initial gaussian processes  $F^{(n_{SA})}$  and  $G^{(n_{SA})}$  trained on a design in dimension  $d \times m$ . They are not yet noisy, but we reduce the dimension by GOSA and keep only the inputs  $x_A$  and  $u_B$  and we want to fit a noisy GP in dimension  $d' + m'$  with the same model evaluations. This mean that instead of trying to fit  $f$  and  $g$ , we aim at fitting  $f_{A,B}$  and  $g_{A,B}$  defined for  $1 \leq k \leq n_{SA}$  by

$$\begin{aligned} f(x^k, u^k) &= f_{A,B}(x_A^k, u_B^k) + e_f^k \\ g(x^k, u^k) &= g_{A,B}(x_A^k, u_B^k) + e_g^k, \end{aligned}$$

where  $e_f^k$  and  $e_g^k$  are realizations of  $\epsilon_f^k$  and  $\epsilon_g^k$  with  $\epsilon_f \sim \mathcal{N}_{n_{SA}}(0, \sigma_f^2)$  and  $\epsilon_g \sim \mathcal{N}_{n_{SA}}(0, \sigma_g^2)$ . The noise corresponds to the effect of the removed inputs. This effect was measured by SA and was small in regions of interest but might be important in other regions. Therefore, an homogeneous noise (i.e. when  $\sigma_f$  and  $\sigma_g$  are constant diagonal co-variance matrix) can't be used here and a heteroscedastic one should be used. However, we don't want to add model evaluations to estimate this noise, so we propose to rely on the SA. The GO transformation  $h$  takes high values in areas of

interest and is close to zero otherwise. Therefore, we propose a variance noise proportional to  $1/h$ . This mean that we suppose  $\sigma_f = \sigma_g = \text{diag}_{n_{SA}}(\sigma^1, \dots, \sigma^{n_{SA}})$  where

$$\sigma^k \propto \frac{1}{h(\mu_Z(x^k), \mu_C(x^k))}.$$

Therefore, we impose very low noise in regions of interest and higher noise the further away from such regions. This can inadvertently create noise in regions where there is none, but since these regions are not of interest, it seems acceptable.

**SA STEPS** With noisy Gaussian processes, it is possible to perform any number of sensitivity analysis steps at no additional cost in terms of model evaluations, since all model evaluations are performed in the initial high dimensional space (i.e. without fixing any input during EFISUR iterations). Thus, we don't need anymore a budget of  $n_{SA}$  evaluations for the SA, and we can simply do the same SA than for **Off-SA** algorithm at iteration  $t$  (mostly replacing  $n_{SA}$  by  $t$ ). We propose to simply do one SA step every  $n_{iter/step}$  iteration. To extract the most information from the previous iterations (and to not only rely on the information induced by the updated GPs), we propose to conduct the sensitivity analysis steps on the  $X$ 's by replacing the uniform input laws by the discrete law of the added points. This means that we measure the impact of the  $X_i$  when they vary in a range favored by EFISUR so far. In order to remain open to other domains, the discrete law is completed with the point of the initial DoE and denoted  $\mathbb{P}_{X^{(t)}}$  at iteration  $t$ . The induced  $X_i$  are then no longer independent. The detailed method is described by **Off-seq-SA** algorithm and illustrated in Figure 4.2.

---

**Algorithm 3** SA steps outside batches of EFISUR iterations: **Off-seq-SA**

---

- 1: Design of Experiments (DoE) in the joint space  $\mathcal{X}_d \times \mathcal{U}_p$ :  $D^{(t)} = \{(x^k, u^k), k = 1, \dots, t\}$
  - 2: Calculate simulator responses:  $f^{(t)} = (f(x^k, u^k))_{k=1}^t$  and  $g^{(t)} = (g(x^k, u^k))_{k=1}^t$
  - 3: **while**  $t \leq \text{budget}$  **do**
  - 4:     Fit GPs of  $f$  and  $g$ :  $F_{(x,u)}^{(t)}$  and  $G_{(x,u)}^{(t)}$
  - 5:     Compute  $Z_x^{(t)} = \mathbb{E}_U[F_{(x,U)}^{(t)}]$  and  $C_x^{(t)} = \alpha - \mathbb{P}_U(G_{(x,U)}^{(t)} \leq 0)$
  - 6:     Estimate T-HSIC<sup>*h*</sup>( $X_i, (\mu_{Z^{(t)}}(X), \mu_{C^{(t)}}(X))$ ), C-HSIC<sup>*w*</sup>( $X_i, (\mu_{Z^{(t)}}(X), \mu_{C^{(t)}}(X))$ ),  $p_{X_i}^{TSA}$ , and  $p_{X_i}^{CSA}$  for  $1 \leq i \leq d$  with  $X \sim \mathbb{P}_{X^{(t)}}$
  - 7:     Estimate  $S_j^{H_{set}}$  and  $p_{U_j}$  for  $1 \leq j \leq p$  with  $U_j \sim \mathbb{P}_{U_j}$
  - 8:     Screen the inputs :  $A = \{i, \min(p_{X_i}^{TSA}, p_{X_i}^{CSA}) \leq \alpha_I\}$  and  $B = \{j, p_{U_j} \leq \alpha_I\}$
  - 9:     **for**  $n_{iter/step}$  iterations **do**
  - 10:         Fit **noisy** GPs in reduced dimension  $F_{AB}^{(t)}$  and  $G_{AB}^{(t)}$  on the design  $D^{(t)}$
  - 11:         Calculate  $Z_A^{(t)}$  and  $C_A^{(t)}$
  - 12:          $x_A^{t+1} = \arg \min_{x_A \in \mathcal{X}_A} \text{EFI}_{Z_A^{(t)}, C_A^{(t)}}(x_A)$
  - 13:          $u_B^{t+1} = \arg \min_{u_B \in \mathcal{U}_B} S_{AB}(x_A^{t+1}, u_B)$
  - 14:         Calculate simulator responses at the next point  $(x_A^{t+1}, x_{-A}, u_B^{t+1}, u_{-B})$  where  $x_{-A}$  and  $u_{-B}$  are drawn randomly
  - 15:         Update  $D^{(t)}$  and  $t \leftarrow t + 1$
  - 16:     **end for**
  - 17: **end while**
-

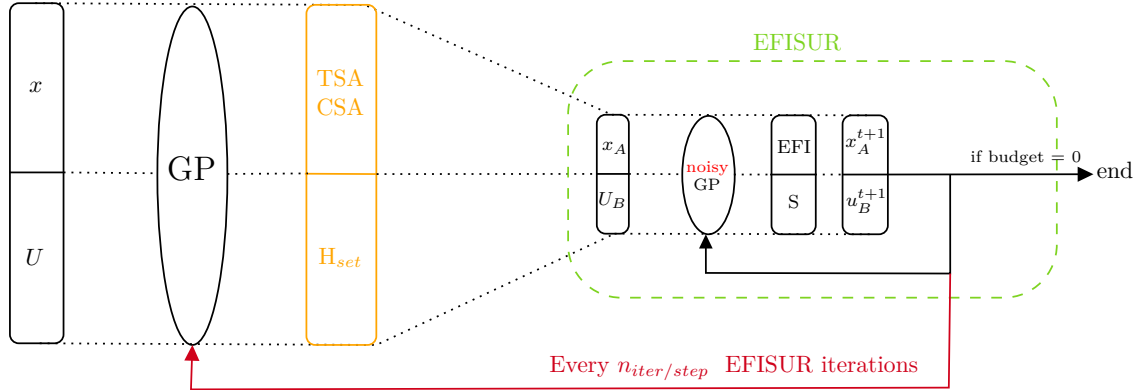


Figure 4.2: Off-seq-SA diagram

As **Off-seq-SA** is based on the previous **Off-SA** algorithm 2, most mentioned choices in Section 4.1.1.b are common to both algorithms. The main difference is the use of noisy gaussian processes that we fit with SA. Alternatives to fit the noise can be to estimate the output noise at each point of the design using the fitted gaussian process in the initial dimension. Estimations of  $Var[\mu_{Z(t)}(x_A^k, X_{-A})]$  for the objective function for instance could be appropriate. However, this is still an estimation which can be very different from the real noise as it relies on the GP. We can also think of alternative ways than using a noisy GP. An example would be to separate the initial GP into two gaussian processes. One first GP associated to the kept inputs fitted with care, and a second GP corresponding to the impact of the rest of the inputs fitted with less precision. Instead of trying to fit the output  $f$ , it is also possible to aim at fitting  $\mathbb{E}(f(X, U)|(X_A, U_A))$  as done in [98].

This first approach aimed at making EFISUR usable in higher dimensions by performing one or multiple SA steps to reduce the input dimension in order to simplify the Gaussian process regression. In a second part, we will see how SA can also be used inside EFISUR to simplify the optimization of the acquisition functions.

## 4.2 Online Sensitivity Analysis in EFISUR algorithm

In this part, we are interested in getting inside EFISUR iterations and in particular in the two optimization steps within EFISUR, i.e. the maximization of the expected feasible improvement and the minimization of the proxy of the one-step-ahead feasible improvement variance. In higher dimension, these two optimizations can become difficult because the optimization requires more iterations and the time computation of the two acquisition criteria explode. To overcome this difficulty, we aim at using sensitivity analysis to facilitate these two optimization steps.

### 4.2.1 On-SA: Dimension reduction by SA on the acquisition criteria optimization

First, at each EFISUR iteration, we reduce the dimension of  $\mathcal{X}$  before optimizing the EFI and of  $\mathcal{U}$  before optimizing  $S$ .

#### 4.2.1.a Method and algorithm

In this part, we want to reduce the dimension only for the optimization of the EFI. So we propose to measure the effects of the design inputs on the EFI itself. We consider

$$X = (X_1, \dots, X_d) \longrightarrow \text{EFI}_{Z^{(t)}, C^{(t)}}(X),$$

to compute  $\text{HSIC}(X_i, \text{EFI}_{Z^{(t)}, C^{(t)}}(X))$  and the associated p-values  $p_{X_i}^{\text{EFI}}$  for  $i \in \{1, \dots, d\}$ , to select only the inputs that affect the EFI. After fixing the negligible ones, the optimization of the EFI can be done in a smaller dimension. The negligible  $x_{-A}$  are fixed at random values as the SA reveals that their value doesn't change the EFI.

Similarly, we propose to measure the effect of the uncertain inputs on  $S$  as we want to minimize it. We are already at a fixed  $x$  (fixed by the maximization of the EFI at  $x^{t+1}$ ), so the output is also a scalar. We consider

$$U = (U_1, \dots, U_p) \longmapsto S(x^{t+1}, U),$$

to compute  $\text{HSIC}(U_j, S(x^{t+1}, U))$  and the associated p-value  $p_{U_j}$  for  $j \in \{1, \dots, p\}$  to detect which  $U_j$  affects  $S$ . The negligible  $U_{-B}$  are also fixed at random values according to  $\mathbb{P}_{U_{-B}}$ . Then the optimization of  $S(x^{t+1}, \cdot, u_{-B})$  is performed in the reduced dimension.

**On-SA** algorithm and Figure 4.3 details EFISUR with these two additional steps.

---

**Algorithm 4** EFISUR with SA on acquisition functions: **On-SA**


---

- 1: Design of Experiments (DoE) in the joint space  $\mathcal{X} \times \mathcal{U}$ :  $D^{(t)} = \{(x^k, u^k), k = 1, \dots, t\}$
  - 2: Calculate simulator responses:  $f^{(t)} = (f(x^k, u^k))_{k=1}^t$  and  $g^{(t)} = (g(x^k, u^k))_{k=1}^t$
  - 3: **while**  $t \leq \text{budget}$  **do**
  - 4:   Fit GPs of  $f$  and  $g$ :  $F_{(x,u)}^{(t)}$  and  $G_{(x,u)}^{(t)}$
  - 5:   Calculate  $Z_x^{(t)}$  and  $C_x^{(t)}$
  - 6:   **SA step on the  $x$ 's:**
  - 7:     Estimate  $\text{HSIC}(X_i, \text{EFI}_{Z^{(t)}, C^{(t)}}(X))$ ,  $p_{X_i}^{\text{EFI}}$  for  $1 \leq i \leq d$  with  $X_i \sim \mathcal{U}(\mathcal{X}_i)$ .
  - 8:     Screen the  $x_i$ :  $A = \{i, p_{X_i} \leq \alpha_I\}$  and fix the others randomly at  $x_{-A}$
  - 9:      $x_A^{t+1} = \arg \max_{x_A \in \mathcal{X}_A} \text{EFI}_{Z^{(t)}, C^{(t)}}(x_A, x_{-A})$
  - 10:   Set  $x^{t+1} = (x_A^{t+1}, x_{-A})$
  - 11:   **SA step on the  $U$ 's:**
  - 12:     Estimate  $\text{HSIC}(U_j, S(x^{t+1}, U))$  and  $p_{U_j}^S$  for  $1 \leq j \leq p$  with  $U_j \sim \mathbb{P}_{U_j}$
  - 13:     Screen the  $U_j$ :  $B = \{j, p_{U_j} \leq \alpha_I\}$  and fix the others randomly at  $u_{-B}$
  - 14:      $u_B^{t+1} = \arg \min_{u_B \in \mathcal{U}_B} S(x^{t+1}, u_B, u_{-B})$
  - 15:   Set  $u^{t+1} = (u_B^{t+1}, u_{-B})$
  - 16:   Calculate simulator responses at the next point  $(x^{t+1}, u^{t+1})$
  - 17:   Update the DoE and  $t \leftarrow t + 1$
  - 18: **end while**
- 

#### 4.2.1.b Discussion and perspectives

The main choice in **On-SA** is the method for conducting the two sensitivity analysis steps before each optimization of the acquisition function. A second important choice is to fix the removed inputs at random.

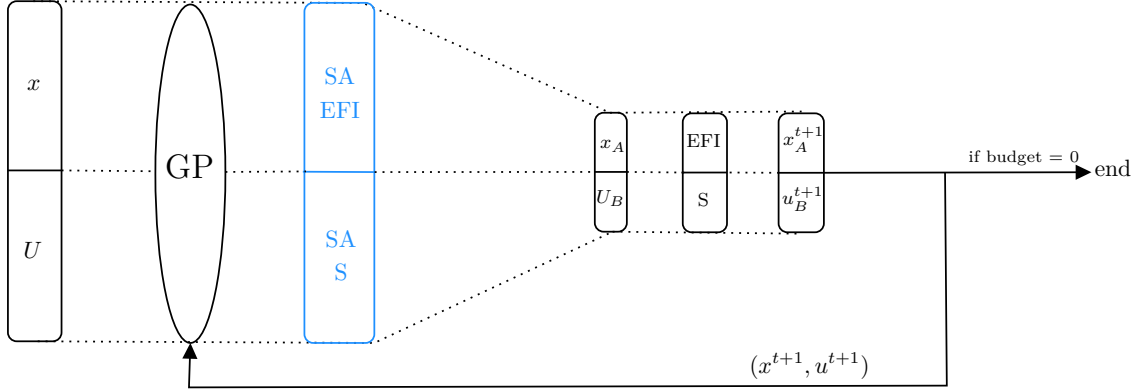


Figure 4.3: On-SA diagram

**SA STEPS** For the SA, we decided to select the influential inputs by computing  $\text{HSIC}(X_i, \text{EFI}_{Z^{(t)}, C^{(t)}}(X))$  and  $\text{HSIC}(U_i, S(x^{t+1}, U))$  without adding goal-oriented transformations, as it seems redundant since the EFI and  $S$  are already goal-oriented transformations. Alternatively, we could apply the SA to the optimization problem itself, i.e., the minimization of  $f$  while  $g \leq 0$ , using the same  $h$  transformation as in Section 4.1.1. For the uncertain inputs for instance,  $\text{HSIC}(U_i, S(x^{t+1}, U))$  can be replaced by  $\text{HSIC}(U_i, h(\mu_{F^{(t)}}(x^{t+1}, U), \mu_{G^{(t)}}(x^{t+1}, U)))$ . In terms of cost, this is also much simpler since the one-step-ahead proxy of the output variance is expensive to compute. However, inputs that have an impact on  $h(\mu_{F^{(t)}}(x^{t+1}, U), \mu_{G^{(t)}}(x^{t+1}, U))$  are not necessarily the same than the input that impact  $S$  and for this reason, measuring the inputs effect on  $S$  is preferred in this work. The same choice is made for the EFI rather than computing  $\text{HSIC}(X_i, h(\mu_{F^{(t)}}(X), \mu_{Z^{(t)}}(X)))$ .

**FIXING THE NEGLIGIBLE INPUTS** With the same reasoning as in Section 4.1.1.b, the negligible inputs are fixed at random, since they are determined to not affect the quantity to be optimized. Moreover, the chosen value of the inputs is directly used to fit the Gaussian process, so it shouldn't be fixed at a deterministic value since it would reduce the quality of the GP to have too many identical points in some dimensions. An interesting point here is that drawing some of the  $U$ 's at random consists of mixing EFISUR and EFIRAND (which is the same as EFISUR with the  $u^{t+1}$  always taken randomly). In the case that all  $U_i$  are found to be negligible, some EFIRAND steps actually happen in our algorithm.

Reducing dimension before optimizing acquisition functions reduces the computational time, but may induce a worse optimum if SA fails to detect inputs that affect the optimum region. However, it is not dramatic since this probably not occur at each EFISUR iteration. To increase the chance of not missing an influential input, an improved version of **On-SA** is presented in the next part.

#### 4.2.2 On-2steps-SA: Two-steps optimization of the acquisition functions guided by SA

The idea of this algorithm is to add another step of SA to guide the optimization of the acquisition function. In particular, we want to be able to detect inputs that become important locally.

To guide the optimization of the acquisition functions, we propose a methodology in two steps detailed here for the EFI but the principle is the same for  $S$  :

- A SA step on the EFI is conducted to obtain  $x_A$  influential inputs, and the optimization of EFI is carried in  $\mathcal{X}_A$  at fixed  $x_{-A}$  to obtain an optimum  $(x_A^{t+1}, x_{-A})$ . This corresponds to **On-SA** algorithm.
- Then, we consider the model

$$\begin{aligned}\mathcal{X}_{-A} &\longrightarrow \mathbb{R} \\ x_{-A} &\longmapsto \text{EFI}_{Z^{(t)}, C^{(t)}}(x_A^{t+1}, x_{-A})\end{aligned},$$

on which we conduct a second SA step to obtain  $x_{A'}$  influential inputs with  $A' \subset -A$ . Then, EFI is optimized in  $\mathcal{X}_{A'}$  at fixed  $x_A^{t+1}$  and  $x_{-A \cup A'}$ . The final optimum is then  $x^{t+1} = (x_A^{t+1}, x_{A'}^{t+1}, x_{-A \cup A'})$ .

With this added step, the optimization in big dimension of the acquisition functions is replaced by two small dimension optimizations guided by SA.

Instead of keeping fixed  $x_A$  during the second optimization, we also propose to unfreeze it, meaning that the second optimization is done in  $\mathcal{X}_{A \cup A'}$  instead of  $\mathcal{X}_{A'}$ . To use the result of the first optimization,  $x_A^{t+1}$  is used as a starting point (on the corresponding dimensions) for the second optimization.

The resulting **On-2steps-SA** algorithm details the additional steps with a **different color for the unfreezed version**. Figure 4.4 give a summarized version.

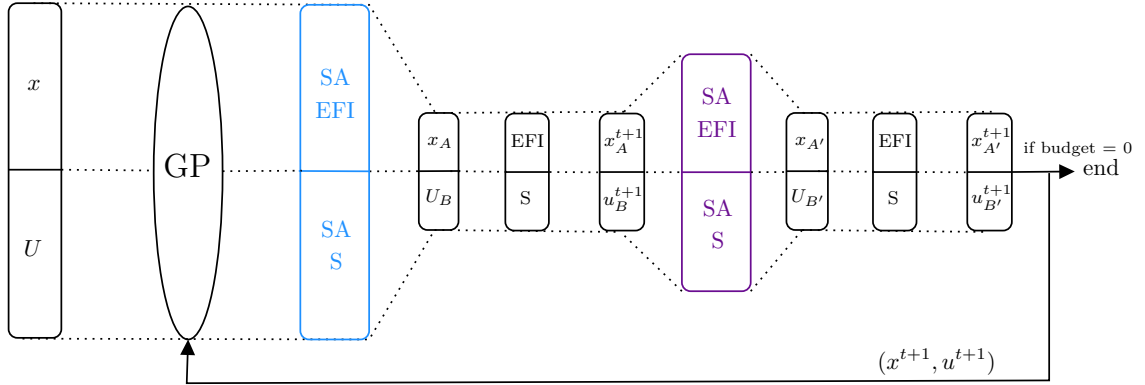


Figure 4.4: On-2steps-SA diagram

Compared to **On-SA**, we add in **On-2steps-SA** another optimization in a small dimension among the remaining inputs not selected at the first step. We can easily think of adding others paired steps of SA and optimization. This could be compared to a block coordinate descent [10], where each inner optimization is the acquisition function optimization and where the coordinate selection is made by SA. However, in our case, a selected input cannot be selected again, so the maximal number of step would be equal to the initial input dimension. Adding steps have a computational cost and may be not necessary as the added inputs are less and less influential on the optimum. For this reason, we consider only these two steps in the following. We still provide **On-seq-SA** algorithm in Appendix that adapts the previous algorithm with multiple steps of SA which consists simply to replace the two SA steps by while loop.

Both **On-SA** and **On-2steps-SA** aim at simplifying the acquisition function optimization but they still rely on gaussian processes in the initial dimension which are difficult and time expensive

---

**Algorithm 5** EFISUR with two SA steps on acquisition functions: **On-2steps-SA-free**


---

- 1: Design of Experiments (DoE) in the joint space  $\mathcal{X} \times \mathcal{U}$ :  $D^{(t)} = \{(x^i, u^i), i = 1, \dots, t\}$
  - 2: Calculate simulator responses:  $f^{(t)} = (f(x^i, u^i))_{i=1}^t$  and  $g^{(t)} = (g(x^i, u^i))_{i=1}^t$
  - 3: **while**  $t \leq$  maximum budget **do**
  - 4:   Fit GPs of  $f$  and  $g$ :  $F_{(x,u)}^{(t)}$  and  $G_{(x,u)}^{(t)}$
  - 5:   Calculate  $Z_x^{(t)}$  and  $C_x^{(t)}$
  - 6:   **1st SA step on the  $x$ 's**
  - 7:     Estimate  $\text{HSIC}(X_i, \text{EFI}_{Z^{(t)}, C^{(t)}}(X))$  and  $p_{X_i}$  for  $1 \leq i \leq d$  with  $X_i \sim \mathcal{U}(\mathcal{X}_i)$
  - 8:     Screen the  $x_i$ :  $A = \{i, p_{X_i}^{\text{EFI}} \leq \alpha_I\}$  and fix the others randomly at  $x_{-A}$
  - 9:      $x_A^{t+1} = \arg \max_{x_A \in \mathcal{X}_A} \text{EFI}_{Z^{(t)}, C^{(t)}}(x_A, x_{-A})$
  - 10:   **2nd SA step on the  $x$ 's**
  - 11:     Estimate  $\text{HSIC}(X_i, \text{EFI}_{Z^{(t)}, C^{(t)}}(X_{-A}, x_A^{t+1}))$  and  $p_{X_i}$  for  $i \in -A$  with  $X_i \sim \mathcal{U}(\mathcal{X}_i)$ .
  - 12:     Screen the  $x_i$ :  $A' = \{i \in -A, p_{X_i}^{\text{EFI}} \leq \alpha_I\}$  and fix the others randomly at  $x_{-(A \cup A')}$
  - 13:      $x_{A'}^{t+1} = \arg \max_{x_{A'} \in \mathcal{X}_{A'}} \text{EFI}_{Z^{(t)}, C^{(t)}}(x_{A'}, x_A^{t+1}, x_{-(A \cup A')})$
  - 14:     **Or** :  $x_{A \cup A'}^{t+1} = \arg \max_{x_{A \cup A'} \in \mathcal{X}_{A \cup A'}} \text{EFI}_{Z^{(t)}, C^{(t)}}(x_{A \cup A'}, x_{-(A \cup A')})$  started at  $x_A = x_A^{t+1}$
  - 15:     Set  $x^{t+1} = (x_A^{t+1}, x_{A'}^{t+1}, x_{-(A \cup A')})$  or  $x^{t+1} = (x_{A \cup A'}^{t+1}, x_{-(A \cup A')})$
  - 16:   **1st SA step on the  $U$ 's**
  - 17:     Estimate  $\text{HSIC}(U_i, S(x^{t+1}, U))$  and  $p_{U_j}^S$  for  $1 \leq j \leq p$  with  $U_j \sim \mathbb{P}_{U_j}$
  - 18:     Screen the  $U_j$ :  $B = \{j, p_{U_j} \leq \alpha_I\}$  and fix the others randomly at  $u_{-B}$
  - 19:      $u_B^{t+1} = \arg \min_{u_B \in \mathcal{U}_B} S(x^{t+1}, u_B, u_{-B})$
  - 20:   **2nd SA step on the  $U$ 's**
  - 21:     Estimate  $\text{HSIC}(U_i, S(x^{t+1}, U))$  and  $p_{U_j}^S$  for  $i \in -B$  with  $U_j \sim \mathbb{P}_{U_j}$
  - 22:     Screen the  $U_j$ :  $B' = \{j \in -B, p_{U_j} \leq \alpha_I\}$  and fix the others randomly at  $u_{-(B \cup B')}$
  - 23:      $u_{B'}^{t+1} = \arg \min_{u_{B'} \in \mathcal{U}_{B'}} S(x^{t+1}, u_{B'}, u_B^{t+1}, u_{-(B \cup B')})$
  - 24:     **Or** :  $u_{B \cup B'}^{t+1} = \arg \min_{u_{B \cup B'} \in \mathcal{U}_{B \cup B'}} S(x^{t+1}, u_{B \cup B'}, u_{-(B \cup B')})$  started at  $u_B = u_B^{t+1}$
  - 25:     Set  $u^{t+1} = (u_B^{t+1}, u_{B'}^{t+1}, u_{-(B \cup B')})$  or  $u^{t+1} = (u_{B \cup B'}^{t+1}, u_{-(B \cup B')})$
  - 26:     Calculate simulator responses at the next point  $(x^{t+1}, u^{t+1})$
  - 27:     Update the DoE and  $t \leftarrow t + 1$
  - 28: **end while**
-

to fit. For this reason, they might remain strongly affected by the increase of the dimension. This particular point is studied numerically in the next chapter.

To conclude this chapter dedicated to the presentation of different algorithms that use SA to help EFISUR in large dimension, we can add that one of the algorithms of Section 4.1 can easily be combined with one of Section 4.2. This would allow to reduce both the dimension of the fitted Gaussian processes and then the dimension of the acquisition criterion optimization in the inner loops. This might be appropriate when the function to be optimized is defined in high dimension, with an "effective target dimension" that is still too high to easily optimize the EFI and S.



# 5

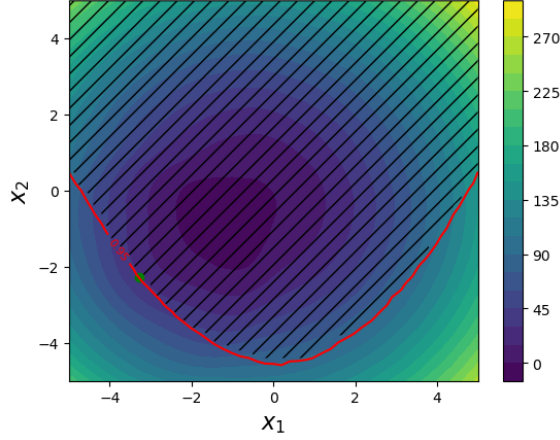
## Sensitivity Analysis and EFISUR : Numerical study

---

|            |                                                                                           |            |
|------------|-------------------------------------------------------------------------------------------|------------|
| <b>5.1</b> | <b>Construction of a toy function for robust optimization and EFISUR results</b>          | <b>92</b>  |
| 5.1.1      | Toy functions $f_{Glob}^d$ and $g_{Glob}^d$ with global impact far from the optimum       | 93         |
| 5.1.1.a    | Definition of the test case                                                               | 93         |
| 5.1.1.b    | Sensitivity Analysis of $f_{Glob}^{10}$ and $g_{Glob}^{10}$                               | 94         |
| 5.1.1.c    | Optimization results with EFISUR on the toy functions $f_{Glob}^{10}$ and $g_{Glob}^{10}$ | 96         |
| 5.1.2      | Toy functions $f_{Loc}^d$ and $g_{Loc}^d$ with local influence near the optimum           | 100        |
| 5.1.2.a    | Definition of the test case                                                               | 100        |
| 5.1.2.b    | Sensitivity Analysis of $f_{Loc}^{10}$ and $g_{Loc}^{10}$                                 | 100        |
| 5.1.2.c    | EFISUR on the toy function                                                                | 102        |
| <b>5.2</b> | <b>Offline Sensitivity Analysis in EFISUR: numerical results on the toy functions</b>     | <b>102</b> |
| 5.2.1      | Off-SA: One SA step before EFISUR                                                         | 102        |
| 5.2.2      | Off-seq-SA: Multiple SA steps outside EFISUR iterations                                   | 107        |
| 5.2.2.a    | Off-seq-SA with two SA steps                                                              | 107        |
| 5.2.2.b    | Off-seq-SA with multiple SA steps                                                         | 109        |
| <b>5.3</b> | <b>Online Sensitivity Analysis in EFISUR: numerical results on the toy functions</b>      | <b>111</b> |
| 5.3.1      | On-SA: Dimension reduction by SA on the acquisition functions                             | 111        |
| 5.3.2      | On-2steps-SA: Two-steps dimension reduction by SA on the acquisition functions            | 115        |
| 5.3.2.a    | On-2steps-SA                                                                              | 115        |
| 5.3.2.b    | On-2steps-SA-free                                                                         | 115        |
| <b>5.4</b> | <b>Numerical comparison of the algorithms on the toy functions</b>                        | <b>118</b> |
| <b>5.5</b> | <b>Application: Optimization of an electric machine</b>                                   | <b>119</b> |
| 5.5.1      | Validation                                                                                | 121        |

---

In this chapter, we perform a numerical study of the four algorithms presented in Chapter 4. Since the main motivations is to make EFISUR usable and efficient in higher dimensions, we first introduce a toy function whose input dimension can be chosen. The construction of the toy function is presented in Section 5.1. Locally influential inputs are added to complicate the optimization and to highlight the efficiency or failure of the different algorithms. Related Sensitivity analysis results



**Figure 5.1:** Contour plot of  $\mathbb{E}[f_1(x, U)]$  and hatched area of  $\mathbb{P}[g_1(x, U)] < 0.95$ . Estimated on a  $50 \times 50$  grid with  $U$ -samples of size 10000. The green dot is the global optimum.

on this toy function are also presented and EFISUR is tested on this toy function. In a second (Section 5.2) and third time (Section 5.3) we present and analyze the results of all the algorithms on this toy function. A numerical comparison of the methods is then made in Section 5.4. Finally, they are tested on a real test case of electrical machine design (Section 5.5).

## 5.1 Construction of a toy function for robust optimization and EFISUR results

Initially, EFISUR has been designed and tested in [33] with the toy functions  $f_1$  and  $g_1$  defined by:

$$f_1(\mathbf{x}, \mathbf{u}) = 5(x_1^2 + x_2^2) - (u_1^2 + u_2^2) + x_1(u_2 - u_1 + 5) + x_2(u_1 - u_2 + 3),$$

$$g_1(\mathbf{x}, \mathbf{u}) = -x_1^2 + 5x_2 - u_1 + u_2^2 - 1,$$

with  $\mathbf{x} \in [-5, 5]^2$  and  $\mathbf{U} \sim \mathcal{U}([-5, 5]^2)$ . The corresponding robust optimization problem defined by

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E}[f_1(x, U)] \text{ where } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P}[g_1(x, U) \leq 0] \geq \alpha\}.$$

for  $\alpha = 0.95$  leads to a reference optimum  $x^* = (x_1^*, x_2^*) = (-3.28, -2.27)$  and a minimum  $e^* = \mathbb{E}[f_1(x^*, U)] = 39.44$  obtained numerically with a large number of  $f_1$  and  $g_1$  evaluations. The contour levels of  $\mathbb{E}[f_1(x, U)]$  and the hatched non-feasible domain where  $\mathbb{P}[g_1(x, U) \leq 0] < \alpha$  are plotted in Figure 5.1, estimated with crude Monte Carlo with samples of size 10000 at each point of a  $50 \times 50$  grid. The same numerical estimation parameters are used for the others contour plots. As shown in [33], EFISUR performs well on this function. However, this is not a very difficult toy function since it's only in dimension  $2 + 2$  and the feasible region is wide and connected. To increase the difficulty of the optimization, we propose an adaptation of this toy function by adding inputs. Our goal is to propose a toy function whose dimension can be increased arbitrarily, but whose number of variables required to find the optimum remains small. A first simple possibility is to add dummy inputs, i.e. inputs that don't affect the output values. Similarly, it is possible to add inputs that have a very small effect on the output values. In these two cases, classical dimension reduction methods work easily enough. Therefore, we prefer to add either inputs that affect  $f_1$  and

$g_1$  **globally** but don't modify  $x^*$  and  $e^*$  (Section 5.1.1 with the toy functions  $f_{Glob}^d$  and  $g_{Glob}^d$ ), or inputs that **locally** affect  $x^*$  and  $e^*$  and whose global effect on  $f_1$  and  $g_1$  remains small (Section 5.1.2 with the toy functions  $f_{Loc}^d$  and  $g_{Loc}^d$ ).

### 5.1.1 Toy functions $f_{Glob}^d$ and $g_{Glob}^d$ with global impact far from the optimum

#### 5.1.1.a Definition of the test case

Adding inputs that affect  $f_1$  and  $g_1$  without changing  $x^*$  and  $e^*$  means that no matter what values the additional inputs take, having  $(x_1, x_2) = (x_1^*, x_2^*)$  is enough to reach  $e^*$ . We modify the toy functions so that the optimal points are all points of the hypervolume  $\mathcal{H}^* = \{(x_1^*, x_2^*, x_3, \dots, x_d), x_{3:d} \in [-5.5]^{d-2}\}$ . To do this, we increase together both dimension  $d$  and  $p$  and we define  $f_{Glob}^{d,1}$  and  $g_{Glob}^{d,1}$  for  $d \geq 3$  and  $p = d$  as

$$\begin{aligned} f_{Glob}^{d,1}(x, u) &= f_1(x_1, x_2, u_1, u_2) \\ &+ \sum_{i=3}^d \left( \beta_{i-2} \left( x_i^2 + \frac{u_i + 5}{10} \right) \right) \mathbb{1}_{\|x_{1:2} - x^*\|_2 > 3} \\ &+ \sum_{i=3}^d \left( \beta_{i-2} \left( x_i^3 + \frac{u_i + 5}{10} \right) \right) \mathbb{1}_{\|x_{1:2} - x_{far}\|_2 < 3}, \end{aligned}$$

$$\begin{aligned} g_{Glob}^{d,1}(x, u) &= g_1(x_1, x_2, u_1, u_2) \\ &+ \sum_{i=3}^d \left( \beta'_{i-2} \left( x_i^2 + \frac{u_i + 5}{10} \right) \right) \mathbb{1}_{\|x_{1:2} - x^*\|_2 > 3}, \\ &- \sum_{i=3}^d (\beta'_{i-2} x_i^3) \mathbb{1}_{\|x_{1:2} - x_{far}\|_2 < 3} \end{aligned}$$

where  $x^* = (-3.28, -2.27)$ ,  $x_{far} = (3, 3)$ ,  $\beta_3 \geq \beta_4 \geq \dots \geq \beta_d$  and  $\beta'_3 \geq \beta'_4 \geq \dots \geq \beta'_d$ .  $f_{Glob}^{d,1}$  and  $g_{Glob}^{d,1}$  are constructed so that the added inputs have no impact in the region of the optimum, since  $\mathbb{1}_{\|x_{1:2} - x^*\|_2 > 3}$  and  $\mathbb{1}_{\|x_{1:2} - x_{far}\|_2 < 3}$  are zero near  $\mathcal{H}^*$ . The minimum remains  $e^* = \mathbb{E} \left[ f_{Glob}^{d,1}(x^*, U) \right] = 39.44$  for any  $x^* \in \mathcal{H}^*$ . The second sum allows to increase the effect of the added inputs on the objective function in the whole domain and to also add a non-feasible minimum of  $\mathbb{E} \left[ f_{Glob}^{d,1}(x, U) \right]$  (near  $x_{far}$ ).  $\beta_i$  and  $\beta'_i$  are decreasing with  $i$ , so the effect of the added inputs decreases with dimension. Thus,  $x_3$  and  $U_3$  should be the added inputs that affect the objective and constraint functions the most. To preserve the initial smoothness of the toy function, we use hyperbolic

tangents which mimics indicator functions:

$$\begin{aligned}
f_{Glob}^d(x, u) &= f_1(x_1, x_2, u_1, u_2) \\
&\quad + \sum_{i=3}^d \left( \beta_{i-2} \left( x_i^2 + \frac{u_i + 5}{10} \right) \right) (\tanh(2(\|x_{1:2} - x^*\|_2 - 3)) + 1) \\
&\quad + \sum_{i=3}^d \left( \beta_{i-2} \left( x_i^3 + \frac{u_i + 5}{10} \right) \right) \left( 1 - \frac{1}{2} (\tanh(2(\|x_{1:2} - x_{far}\|_2 - 3)) + 1) \right), \\
g_{Glob}^d(x, u) &= g_1(x_1, x_2, u_1, u_2) \\
&\quad + \sum_{i=3}^d \left( \beta'_{i-2} \left( x_i^2 + \frac{u_i + 5}{10} \right) \right) (\tanh(2(\|x_{1:2} - x^*\|_2 - 3)) + 1) \\
&\quad - \sum_{i=3}^d (\beta'_{i-2} x_i^3) \left( 1 - \frac{1}{2} (\tanh(2(\|x_{1:2} - x_{far}\|_2 - 3)) + 1) \right).
\end{aligned}$$

For the rest of the chapter, we will mostly work with  $d = 10$  and  $p = d = 10$ . Contour plots of  $\mathbb{E}[f_{Glob}^{10}(x, U)]$  and hatched non-feasible domain where  $\mathbb{P}[g_{Glob}^{10}(x, U) \leq 0]$  are given in Figure 5.2. The plots are displayed in the first two dimensions, and the added inputs  $x_3$  to  $x_{10}$  are changed from  $-5$  to  $5$  at the same time.

#### 5.1.1.b Sensitivity Analysis of $f_{Glob}^{10}$ and $g_{Glob}^{10}$

To analyze the impact of the inputs, we perform a SA on  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . To compare the effects of the inputs on  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  in the entire domain and on  $e^*$  and  $x^*$ , we propose to do a "classical" SA without fixed optimization goal and a Goal-Oriented Sensitivity analysis.

The SA on the design inputs is made on the input-output model

$$X \longrightarrow \begin{pmatrix} E \\ P \end{pmatrix} = \begin{pmatrix} \mathbb{E}_U [f_{Glob}^{10}(X, U)] \\ \mathbb{P}_U [g_{Glob}^{10}(X, U) \leq 0] \end{pmatrix}$$

HSIC-ANOVA with a product of Gaussian kernels for the output are computed for the "classical" SA. GOSA is done by computing T-HSIC<sup>h</sup>( $X_i, (E, P)$ ) and C-HSIC<sup>h</sup>( $X_i, (E, P)$ ) as done in Section 4.1.1.a.

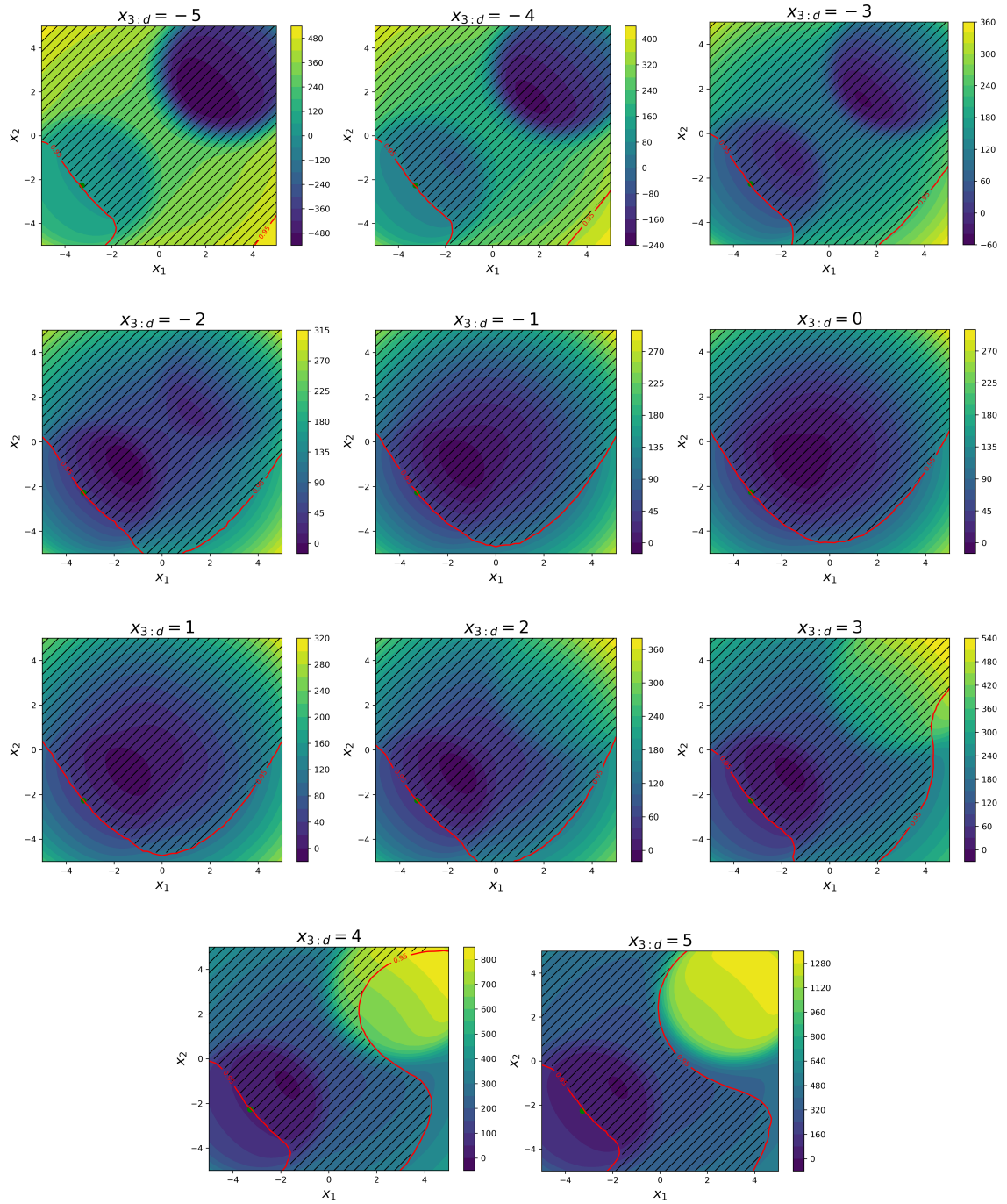
For uncertain inputs, we use HSIC of sets indices.  $S_j^{H_{set}}$  with

$$\Gamma = \{x \in \mathcal{X}, f_{Glob}^{10}(x, U) \leq q, g_{Glob}^{10}(x, U) \leq 0\},$$

where  $q$  is a low quantile of  $f_{Glob}^{10}(x, U)$ , are by definition goal-oriented, as they measure the effect on the  $U$  on minimizing the objective function and respecting the constraint. They are then computed only for the GOSA. For the "classical" SA, we refer to the end of Section 3.2. in which we added the output as a design variable so that pollutant concentration maps could be treated as sets. By doing so,  $S_j^{H_{set}}$  is no longer goal-oriented. Here, we have two outputs  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  so we need to add two new variables denoted  $x_f$  and  $x_g$  varying in a range of values  $[x_f^{min}, x_f^{max}]$  and  $[x_g^{min}, x_g^{max}]$  containing all possible values of  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . This defines excursion sets by :

$$\Gamma = \{(x, x_f, x_g) \in \mathcal{X} \times [x_f^{min}, x_f^{max}] \times [x_g^{min}, x_g^{max}], f_{Glob}^{10}(x, U) \leq x_f, g_{Glob}^{10}(x, U) \leq x_g\}$$

for which  $S_j^{H_{set}}$  are computed.



**Figure 5.2:** Contour plots of  $\mathbb{E}[f_{Glob}^{10}(x, U)]$  and hatched area of  $\mathbb{P}[g_{Glob}^{10}(x, U)] < 0.95$  for fixed  $x_3$  to  $x_{10}$ . Estimated on a  $50 \times 50$  grid with  $U$ -samples of size 10000. Red curve corresponds to  $\mathbb{P}[g_{Glob}^{10}(x, U)] = 0.95$ . The green dot is the global optimum.

For twenty repetitions, rejection rates (i.e. the percentage of the repetitions an input is kept) are computed for all SA methods, with sample of size 1000 and a risk of 5%. The results are given in Table 5.1 for the design inputs and in Table 5.2 for the uncertain inputs. Corresponding boxplots of the indices estimations are given in Figures E.1 and E.2 in Appendix.

|          | $X_1$       | $X_2$       | $X_3$       | $X_4$       | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|----------|-------------|-------------|-------------|-------------|-------|-------|-------|-------|-------|----------|
| HSIC     | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>0.80</b> | 0.10  | 0.05  | 0.05  | 0.10  | 0.05  | 0.00     |
| HSIC-TSA | <b>1.00</b> | <b>1.00</b> | 0.05        | 0.05        | 0.00  | 0.05  | 0.00  | 0.05  | 0.00  | 0.00     |
| HSIC-CSA | 0.15        | <b>0.65</b> | 0.10        | 0.00        | 0.00  | 0.00  | 0.00  | 0.00  | 0.10  | 0.15     |

**Table 5.1:** Rejection rates on 20 repetitions of the design inputs obtained with classic, target and conditional HSIC for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ .

| HSIC of sets | $U_1$       | $U_2$       | $U_3$       | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ |
|--------------|-------------|-------------|-------------|-------|-------|-------|-------|-------|-------|----------|
| SA           | <b>1.00</b> | <b>1.00</b> | <b>0.70</b> | 0.15  | 0.00  | 0.20  | 0.05  | 0.00  | 0.00  | 0.10     |
| GOSA         | <b>1.00</b> | <b>1.00</b> | 0.00        | 0.10  | 0.10  | 0.05  | 0.15  | 0.10  | 0.00  | 0.15     |

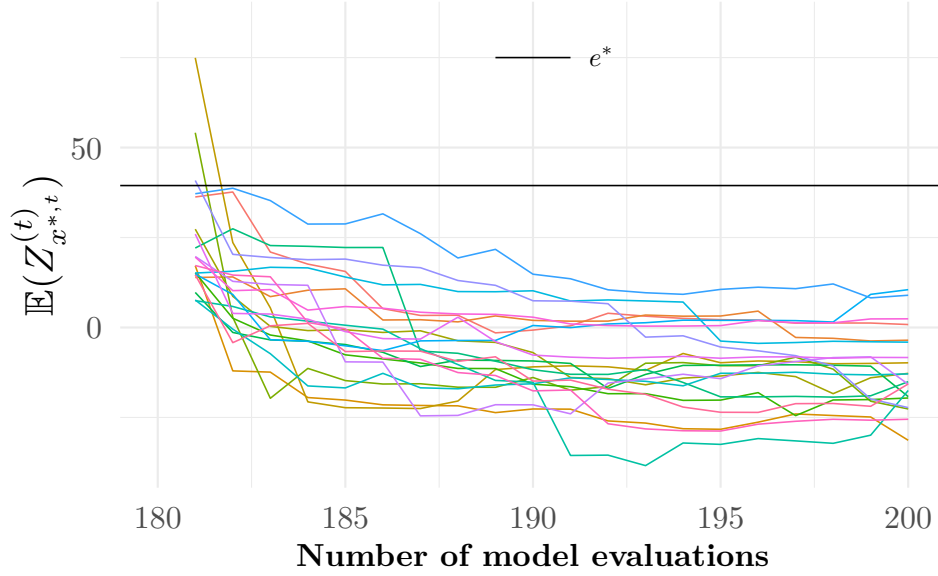
**Table 5.2:** Rejection rates on 20 repetitions of the uncertain inputs obtained with goal-oriented and not goal-oriented  $S_j^{H_{set}}$  for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ .

As expected, "classical" SA on the  $X$ 's and on the  $U$ 's show that  $x_1, x_2, U_1$  and  $U_2$  are influential but also other inputs which effect decreases with their index. Both  $X_3$  and  $U_3$  are (almost always) influential. Then  $X_4$  is often found to be influential as well. Then the rest of the inputs are nearly always found to be negligible, meaning that the constant  $\beta_i$  reduces drastically their influence. With the GOSA we get different results. For the  $X$ 's, the TSA allows to always detect  $X_1$  and  $X_2$ , while the rest of the inputs are almost never detected as influential. This confirms that only  $X_1$  and  $X_2$  are necessary to reach the optimum. With CSA,  $X_2$  is often detected as important (65%) and  $X_1$  only 15%. This reveals that  $X_2$  is more influential than  $X_1$  near the optimum. For uncertain inputs, GOSA confirms that only  $U_1$  and  $U_2$  affect the excursion sets.

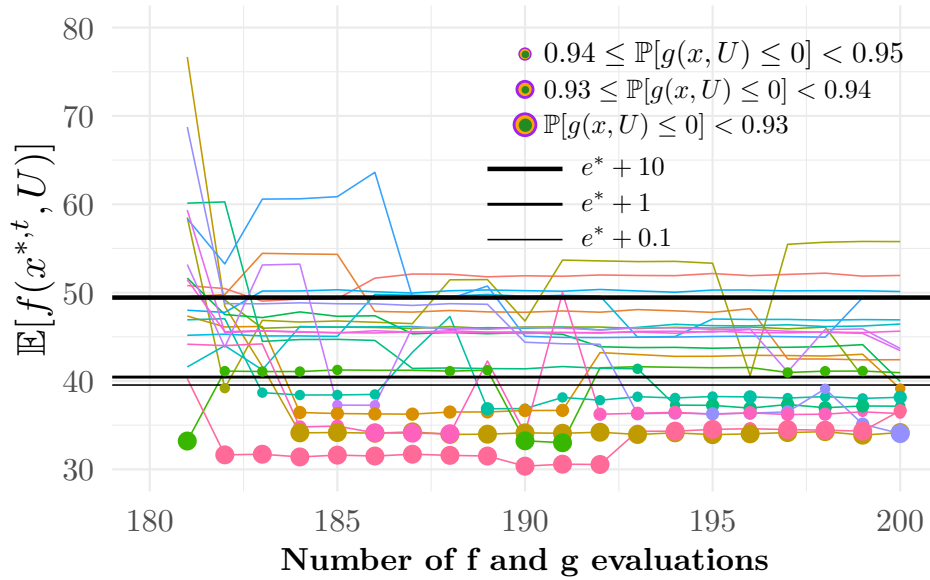
### 5.1.1.c Optimization results with EFISUR on the toy functions $f_{Glob}^{10}$ and $g_{Glob}^{10}$

EFISUR is run  $N = 20$  times with a budget of  $10 \times (d + p) = 200$  evaluations of  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . This budget is divided into the number of points for the initial design and the number of iterations. We first test a budget allocation of  $180 + 20$ . To plot EFISUR results, a first possibility is to plot the evolution of the obtained feasible minimum  $\mathbb{E}(Z_{x^*,t}^{(t)})$  at the current optimum  $x^{*,t}$  along the iterations. This is done in Figure 5.3. We observe, as expected, that  $\mathbb{E}(Z_{x^*,t}^{(t)})$  is decreasing. However, it is far below the true minimum  $e^*$  (black line), highlighting important errors in the models. To see if the algorithm really minimizes  $\mathbb{E}[f_{Glob}^{10}(x, U)]$  while satisfying the constraint, we replace  $\mathbb{E}(Z_{x^*,t}^{(t)})$  with an accurate estimate of  $\mathbb{E}[f_{Glob}^{10}(x^{*,t}, U)]$  done with crude MC and 10000 points. We also check if the constraint is satisfied by estimating  $\mathbb{P}[g_{Glob}^{10}(x^{*,t}, U) \leq 0]$ . The resulting plot is shown in Figure 5.4 with some levels of objective and constraints defined in the next paragraph. This confirms that despite GP errors, the algorithm still manages to find low values of  $\mathbb{E}[f_{Glob}^{10}(x, U)]$ , but some of the proposed points don't verify the constraint. Such plots illustrate well EFISUR efficiency but are not suitable for comparison purposes. Instead, we propose to plot *data profiles*.

**DATA PROFILES** In the context of deterministic optimization [107], data profiles allow to measure and compare performances of optimization algorithms. They consist in measuring how fast (in terms



**Figure 5.3:** Evolution of the estimated current feasible minimum during EFISUR iterations for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  with an initial design of 180 points and 20 iterations for 20 optimization runs with different initial lhs designs.



**Figure 5.4:** Evolution of the true current feasible minimum during EFISUR iterations for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  with an initial design of 180 points and 20 iterations for 20 optimization runs and different initial lhs designs.

of model evaluations) each run of the algorithm reaches a fixed goal. In practice, for each model evaluation, the proportion of runs that reach the goal is calculated. The plot of this proportion as a function of the number of model evaluations is called a data profile. This allows to consider both the speed of convergence of the algorithm (at which iteration it reaches the goal) and its robustness (how many runs reach the goal). In our case, given generic objective and constraint functions  $f$  and  $g$ , we consider two goals defined by

- $\mathbb{E}[f(x, U)] \leq e$  where  $e$  is a low objective function value.
- $\mathbb{P}[g(x, U) \leq 0] \geq p$  where  $p$  is a fixed threshold.

The goals are considered to be reached if at least one point of the current dataset of evaluations  $D_i^{(t)}$  satisfy the previous conditions. For  $N$  runs and a fixed goal  $(e, p)$ , the data profile  $dp_{e,p}$  is defined by

$$dp_{e,p} : t \in \mathbb{N} \mapsto \frac{\text{card}\{i \in \{1, \dots, N\}, \exists x \in D_i^{(t)}, \mathbb{E}[f(x, U)] \leq e, \mathbb{P}[g(x, U) \leq 0] \geq p\}}{N} \quad (5.1)$$

where  $D_i^{(t)}$  is the design of the  $i$ -th run containing  $t$  first search points (initial points and points added by the algorithm). Computing data profiles is possible with a toy function as accurately estimates  $\mathbb{E}[f(x, U)]$  and  $\mathbb{P}[g(x, U) \leq 0]$  are affordable.

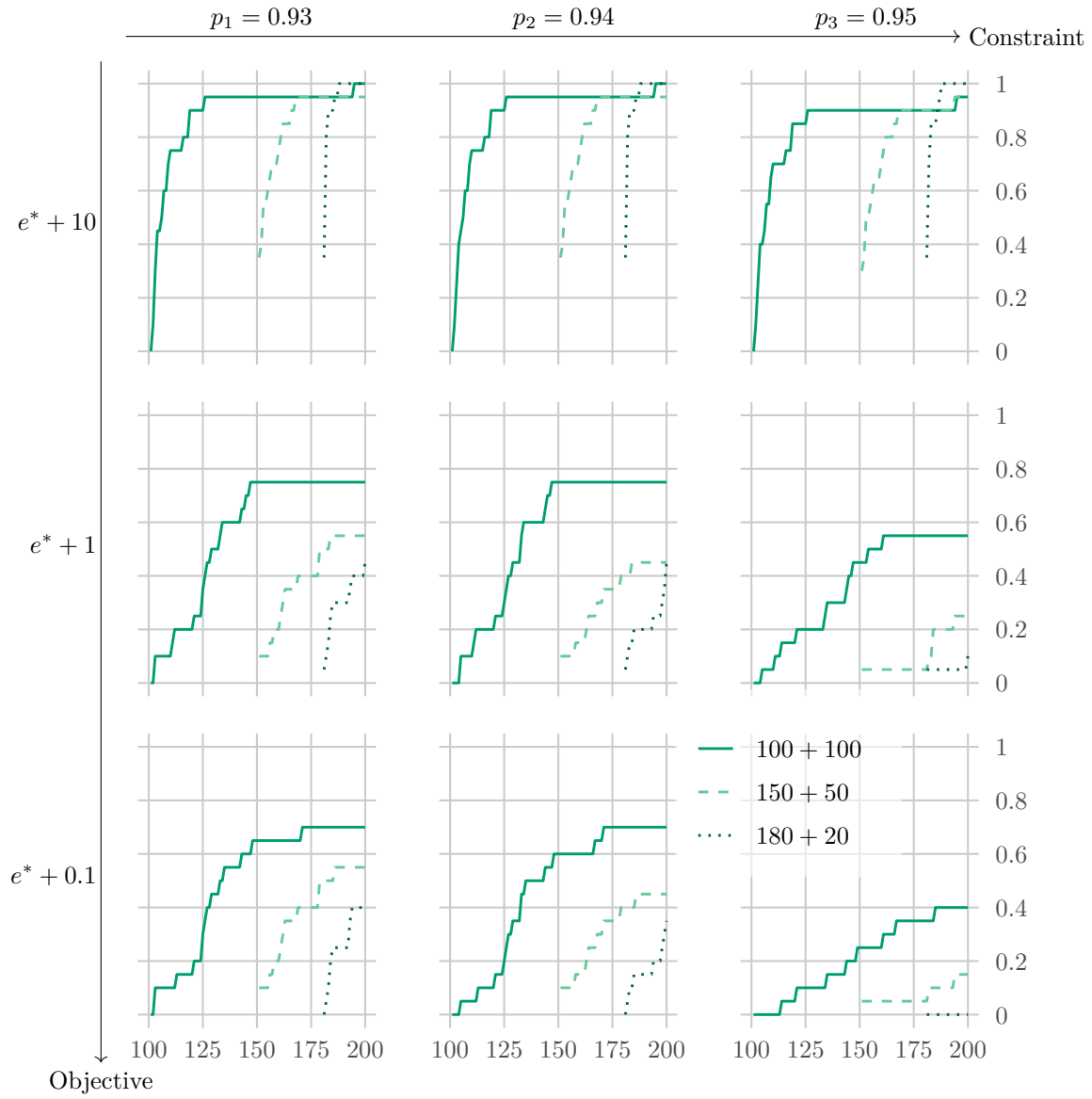
For all numerical results, we consider three accuracy levels of both both goals  $e$  and  $p$

- $e \in (e^* + 10, e^* + 1, e^* + 0.1)$  define three reference values for the objective.
- $p \in (\alpha, \alpha - 0.1, \alpha - 0.2) = (0.95, 0.94, 0.93)$  define three constraint thresholds.

This results in nine possible goals defined by all the pairs  $(e_i, p_j)$  for  $(i, j) \in \{1, 2, 3\}^2$ .

We compute data profiles resulting from 20 runs of EFISUR on functions  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . Three different simulation budget allocations are tested:  $180 + 20$ ,  $150 + 50$ , and  $100 + 100$  (size of the initial DoE + number of iteration). These budgets are identical in terms of the number of model evaluations, but a small number of iterations should be preferred for two reasons. First, it reduces the number of algorithm iterations and thus the algorithm running time, which can be important especially in high dimensions. But more importantly, the model evaluations during the iterations cannot be done in parallel as the new point depends directly on the updated processes, while those done during an initial design can be done simultaneously, which can greatly reduce the total computation time if the model is time consuming. These different configurations are compared for each of the nine goals, resulting in nine data profiles, plotted in Figure 5.5. On this first toy function  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ , EFISUR produces satisfactory results with a small budget (only 200 evaluations of  $f$  and  $g$ ). For the three allocations, we manage to reach  $e^* + 10$  for at least 90% of the runs whatever the level of constraint  $p$  (top plots).  $180 + 20$  allocation even manages to catch up after only 8 EFISUR iterations. For the other goals  $e^* + 1$  and  $e^* + 0.1$ , we observe that the more iterations, the better the results. However, even with 100 iterations, from 25% to 60% of the runs don't reach the goals. Relaxing the threshold constraint (from right to left) shows that the algorithm manages to find good points with respect to the objective function value, for which the probability of satisfying the constraint remains high even if it is below the initial threshold of 0.95. To improve these results, we should use more model evaluations, but since our goal is to have a reference to compare with our proposed methods, we don't necessarily aim to do enough model evaluations so that 100% of the runs reach the desired objectives.

With this toy function, all added inputs have a global effect but do not change the optimum. We will now see how to modify the function to make it more complex by adding an input that has a small global effect but which changes the optimum.



**Figure 5.5:** Data profiles of EFISUR for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

### 5.1.2 Toy functions $f_{Loc}^d$ and $g_{Loc}^d$ with local influence near the optimum

#### 5.1.2.a Definition of the test case

With the previous functions  $f_{Glob}^d$  and  $g_{Glob}^d$ , only two design and uncertain inputs ( $x_1, x_2, U_1$  and  $U_2$ ) are needed to find the optimum, and the rest can be neglected. To complicate the problem a bit, we propose to add a last design input  $x_d$  and uncertain input  $U_d$ , which have a small impact on the optimized functions when considering the whole domain, but whose impact increases when approaching the optimum. They are then be essential for finding the optimum. To do so, we modify the objective function by changing the effects of the inputs  $x_d$  and  $U_d$ . The new objective function is defined for  $d \geq 3$  and  $p = d$  as

$$\begin{aligned} f_{Loc}^d(x, u) = & f_1(x_1, x_2, u_1, u_2) \\ & + \sum_{i=3}^{d-1} \left( \beta_{i-2} \left( x_i^2 + \frac{u_i + 5}{10} \right) \right) (\tanh(2(\|x_{1:2} - x^*\| - 3)) + 1) \\ & + \sum_{i=3}^{d-1} \left( \beta_{i-2} \left( x_i^3 + \frac{u_i + 5}{10} \right) \right) \left( 1 - \frac{1}{2} (\tanh(2(\|x_{1:2} - x_{far}\| - 3)) + 1) \right) \\ & - \left( 5 - \frac{(x_d - 4)^2}{5} \right) (u_p + 5) \left( 1 - \frac{1}{2} (\tanh(2(\|x_{1:2} - x^*\| - 1)) + 1) \right). \end{aligned}$$

In this function,  $x_d$  and  $U_d$  are only active when  $\|x_{1:2} - x^*\| < 1$ , i.e. near the optimum where the value  $x_d = 4$  allows to decrease the value of  $f_{Loc}^d$  (and of  $\mathbb{E}[f_{Loc}^d(x, U)]$ ). The minimum is now  $e^* = \mathbb{E}[f_{Loc}^d(x, U)] = 14.89$  obtained with  $x_{1:2} = x^*$  and  $x_d = 4$ , instead of  $e^* = 39.44$  for  $f_{Glob}^d$ . The constraint is not modified, but we denote  $g_{Loc}^d = g_{Glob}^d$  for simplicity. Contour plots in  $10 + 10$  dimension are given in Figure 5.6.

#### 5.1.2.b Sensitivity Analysis of $f_{Loc}^{10}$ and $g_{Loc}^{10}$

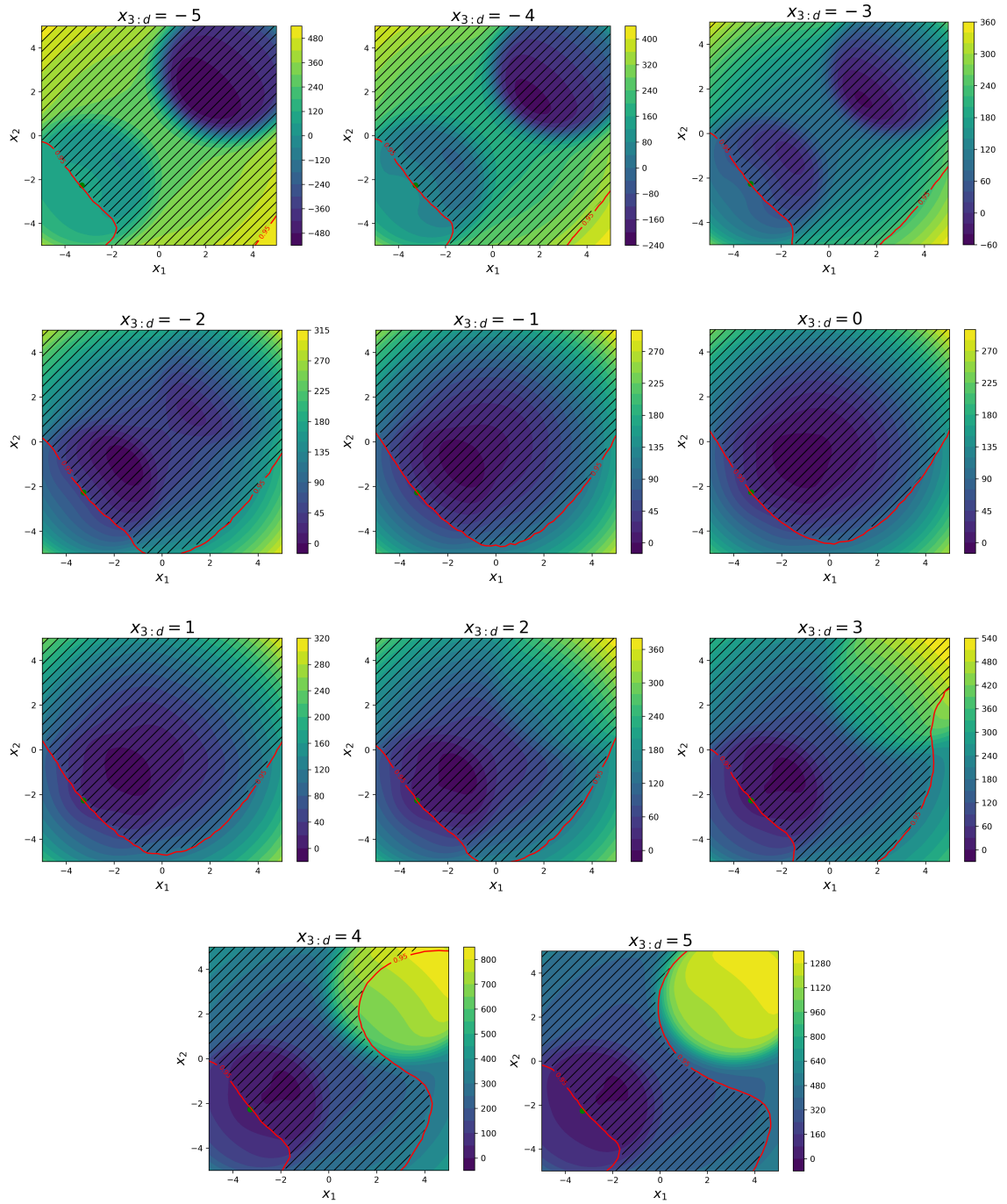
We perform the same sensitivity analysis than in Section 5.1.1.b to quantify the effects of the inputs, especially those of  $x_d$  and  $U_m$ . The rejection rates are given in Table 5.3 for the design inputs and in Table 5.4 for the uncertain inputs. Corresponding indices estimations boxplots are given in Figure E.3 and E.4 in Appendix.

| HSIC    | $X_1$       | $X_2$       | $X_3$       | $X_4$       | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$     |
|---------|-------------|-------------|-------------|-------------|-------|-------|-------|-------|-------|--------------|
| Classic | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>0.85</b> | 0.15  | 0.00  | 0.00  | 0.10  | 0.00  | 0.05         |
| TSA     | <b>1.00</b> | <b>1.00</b> | 0.05        | 0.05        | 0.05  | 0.05  | 0.05  | 0.05  | 0.05  | <b>0.10</b>  |
| CSA     | 0.10        | <b>0.70</b> | 0.00        | 0.10        | 0.10  | 0.00  | 0.10  | 0.15  | 0.15  | <b>0.125</b> |

**Table 5.3:** Rejection rates on 20 repetitions of the design inputs obtained with classic, target and conditional HSIC for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

| HSIC of sets | $U_1$       | $U_2$       | $U_3$       | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$    |
|--------------|-------------|-------------|-------------|-------|-------|-------|-------|-------|-------|-------------|
| SA           | <b>1.00</b> | <b>1.00</b> | <b>0.85</b> | 0.25  | 0.00  | 0.00  | 0.15  | 0.05  | 0.05  | 0.00        |
| GOSA         | <b>1.00</b> | <b>1.00</b> | 0.10        | 0.10  | 0.15  | 0.10  | 0.10  | 0.10  | 0.15  | <b>1.00</b> |

**Table 5.4:** Rejection rates on 20 repetitions of the uncertain inputs obtained with goal-oriented and not goal-oriented  $S_j^{H_{set}}$  for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .



**Figure 5.6:** Contour plots of  $\mathbb{E}[f_{Loc}^{10}(x, U)]$  and hatched area of  $\mathbb{P}[g_{Loc}^{10}(x, U)] < 0.95$  for fixed  $x_3$  to  $x_{10}$ . Estimated on a  $50 \times 50$  grid with  $U$ -samples of size 10000. Red curve corresponds to  $\mathbb{P}[g_{Loc}^{10}(x, U)] = 0.95$ . The green dot is the global optimum.

We again manage to detect  $X_1$  and  $X_2$  separately from the other inputs with the TSA. However, we observe that the TSA and the CSA detect  $X_{10}$  only 10% and 25% of the repetitions. Since we know that  $X_{10}$  has a strong influence on the optimum, which goal-oriented SA does not detect, this confirms that the optimization is difficult. Using our knowledge that  $x_{10}$  is only important near the optimum, we restrict the domain by imposing  $x_1 \leq 0$  and  $x_2 \leq 0$  (southwest quadrant), which contain the optimum. The volume of the design space is then divided by four, which should make it easier to detect  $X_{10}$ . This is indeed the case, as shown in Table 5.5, where both the CSA and the TSA (and even a bit of the classical SA) detect  $X_{10}$ . Associated boxplots are given in Figure E.5 in Appendix.

| HSIC    | $X_1$       | $X_2$       | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$    |
|---------|-------------|-------------|-------|-------|-------|-------|-------|-------|-------|-------------|
| Classic | <b>1.00</b> | <b>1.00</b> | 0.25  | 0.00  | 0.05  | 0.15  | 0.05  | 0.10  | 0.05  | 0.40        |
| TSA     | <b>1.00</b> | <b>1.00</b> | 0.05  | 0.05  | 0.00  | 0.05  | 0.00  | 0.10  | 0.05  | <b>1.00</b> |
| CSA     | <b>0.70</b> | 0.55        | 0.10  | 0.00  | 0.10  | 0.15  | 0.05  | 0.05  | 0.00  | <b>1.00</b> |

**Table 5.5:** Rejection rates on 20 repetitions of the design inputs obtained with classic, target and conditional HSIC for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  zoomed in the south-west quadrant ( $x_1 \leq 0$  and  $x_2 \leq 0$ ).

For the uncertain inputs,  $U_{10}$  is directly identified as influential by the goal-oriented SA, without the need for zooming.

### 5.1.2.c EFISUR on the toy function

We run EFISUR on this harder case for the same budget allocations as for functions  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  (180 + 20, 150 + 50 and 100 + 100). The different goals are kept the same by adjusting  $e^*$ . The resulting data profiles are plotted in Figure 5.7. As expected, we observe that the algorithm produces worse results. The optimization is more difficult for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  and EFISUR struggles more to reach the optimum. This time, even for  $e^* + 10$ , only 100 + 100 allocation always reach the optimum. When the constraint is a bit relaxed (two left columns), some of the runs with 100 iterations manage to reach even the hardest objective function goal of  $e^* + 0.1$ . However, with a threshold constraint of 0.95, almost no run achieves both  $e^* + 1$  and  $e^* + 0.1$  with only 200 model evaluations.

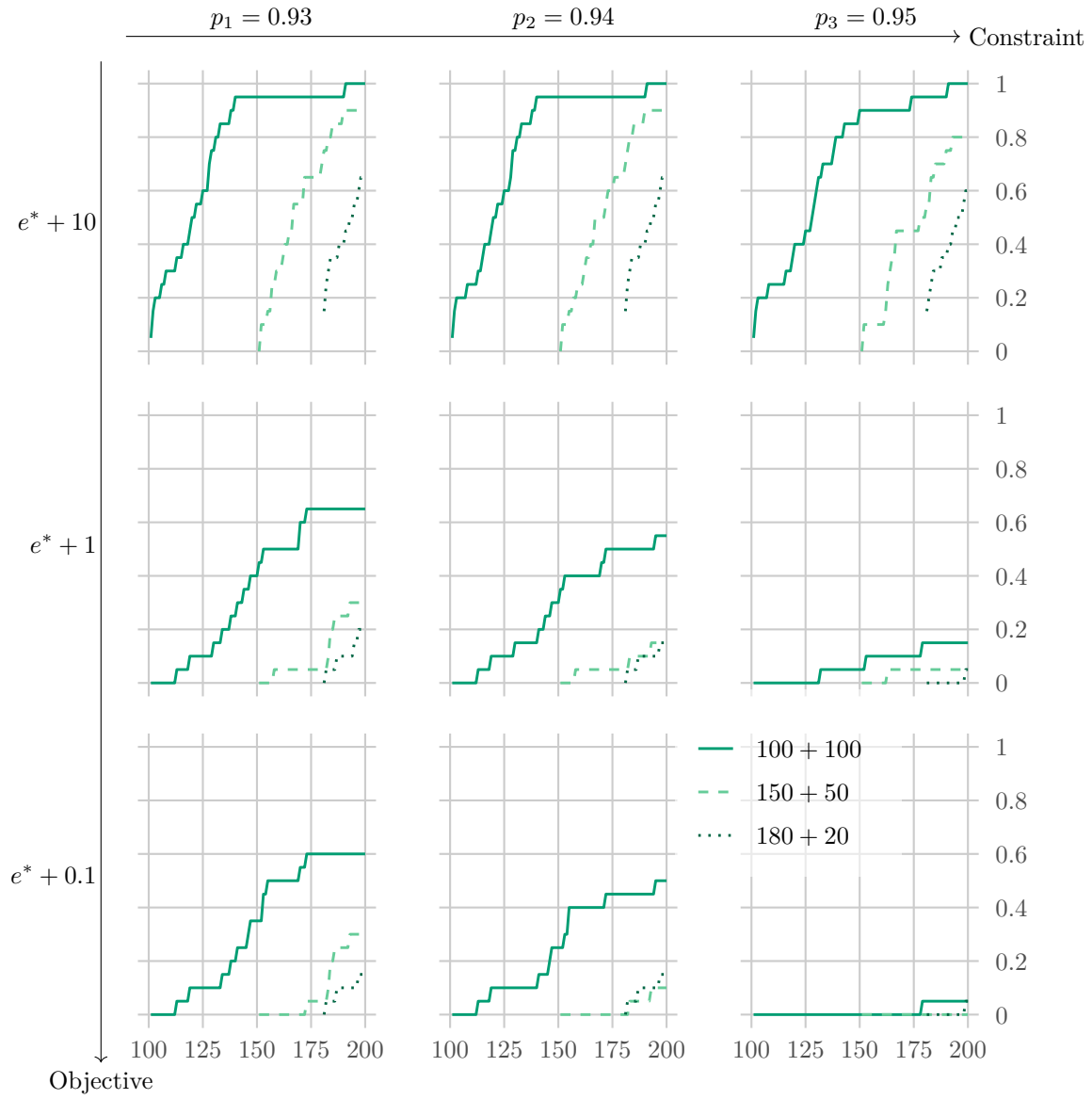
Both Figures 5.5 and 5.7 now serve as reference results of EFISUR on our proposed toy functions. In the next parts we test the algorithms of Chapter 4 on these two toy functions to see if we manage to get better, similar or worse results than classical EFISUR.

## 5.2 Offline Sensitivity Analysis in EFISUR: numerical results on the toy functions

In this part, we test the Offline approach, corresponding to the **Off-SA** and **Off-seq-SA** (algorithms 1 and 2).

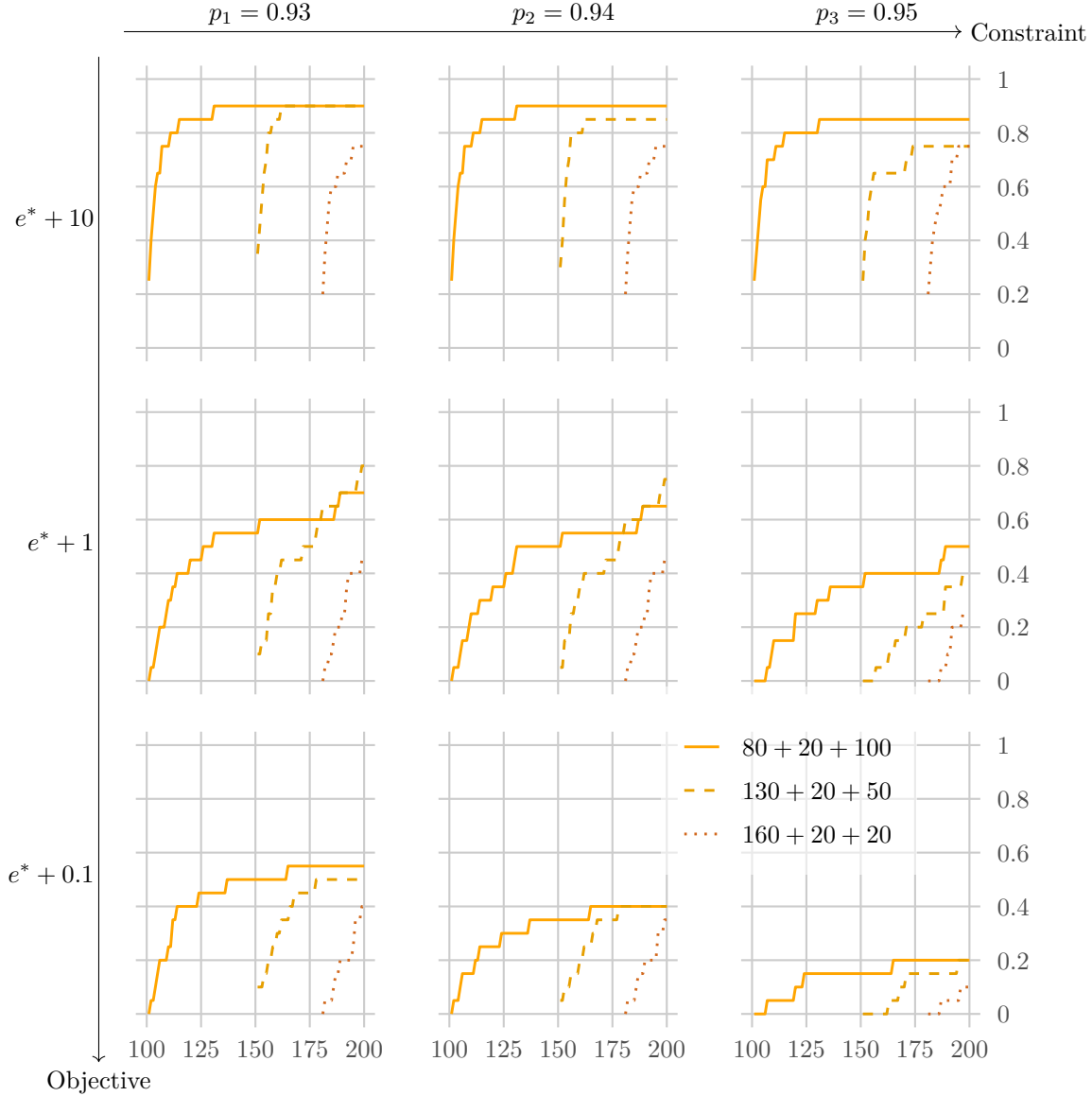
### 5.2.1 Off-SA: One SA step before EFISUR

**Off-SA** (algorithm 1), summarized in Scheme 4.1, is based on a dimension reduction before EFISUR. The dimension reduction is done by SA on the GPs in the initial dimension and trained with a part of the budget of  $f$  and  $g$  evaluations. This part of the budget is important as the GP in initial dimension need to be sufficiently reliable for the SA. Then, the remaining budget is shared between



**Figure 5.7:** Data profiles of EFISUR for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

the initialization step in low dimension and the EFISUR iterations. We test the following budget allocations:  $160 + 20 + 20$ ,  $130 + 20 + 50$ , and  $80 + 20 + 100$  (size of the DoE in initial dimension to fit the GP for the SA step + size of the initial DoE in small dimension + number of EFISUR iterations in small dimension). The screening is done with a risk of  $\alpha_I = 0.05$ . The data profiles for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  are shown in Figure 5.8 obtained for 20 runs of the algorithm. The obtained



**Figure 5.8:** Data profiles of **Off-SA** algorithm for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

results are a bit degraded compared to EFISUR, but they are still satisfactory. To understand why, we can look at the result of the SA steps i.e. at the percentage of runs each input is kept for the optimization. Table 5.6 shows that regardless of the budget allocation, the algorithm keeps only  $X_1$  and  $X_2$  for all runs. In table 5.6 we see that  $U_1$  is not always kept, other inputs being sometimes

preferred. This can be a reason for the degradation of the results. This is accentuated when the initial budget is small, because the SA results rely on GPs that are less predictive. This points the main difficulty of the algorithm, which is to decide how much budget to allocate to the SA so that the SA is reliable meanwhile enough budget remains to conduct EFISUR in small dimension.

| Budget apportionment | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 160 + 20 + 20        | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
| 130 + 20 + 50        | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
| 80 + 20 + 100        | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |

**Table 5.6:** Percentage of kept  $X$ 's on 20 runs in the SA step of **Off-SA** algorithm for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ .

| Budget allocation | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 160 + 20 + 20     | 0.75  | 1     | 0     | 0.05  | 0.1   | 0     | 0.05  | 0.05  | 0     | 0        |
| 130 + 20 + 50     | 0.8   | 1     | 0.05  | 0.05  | 0.1   | 0     | 0     | 0     | 0     | 0        |
| 80 + 20 + 100     | 0.65  | 1     | 0.1   | 0.1   | 0     | 0.05  | 0     | 0     | 0.05  | 0.05     |

**Table 5.7:** Percentage of kept  $U$ 's on 20 runs in the SA step of **Off-SA** algorithm for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ .

Being able to at least always detect  $X_1$  and  $X_2$  even with a small number of initial points is reassuring, but not a very difficult task, as we saw in the SA results of Section 5.1.1 that  $X_1$  and  $X_2$  were responsible for most of the output variability. This should be more difficult for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ , where  $X_{10}$  is difficult to detect. The data profiles for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  and corresponding SA results are shown in Figure 5.9 and Tables 5.8 and 5.9.

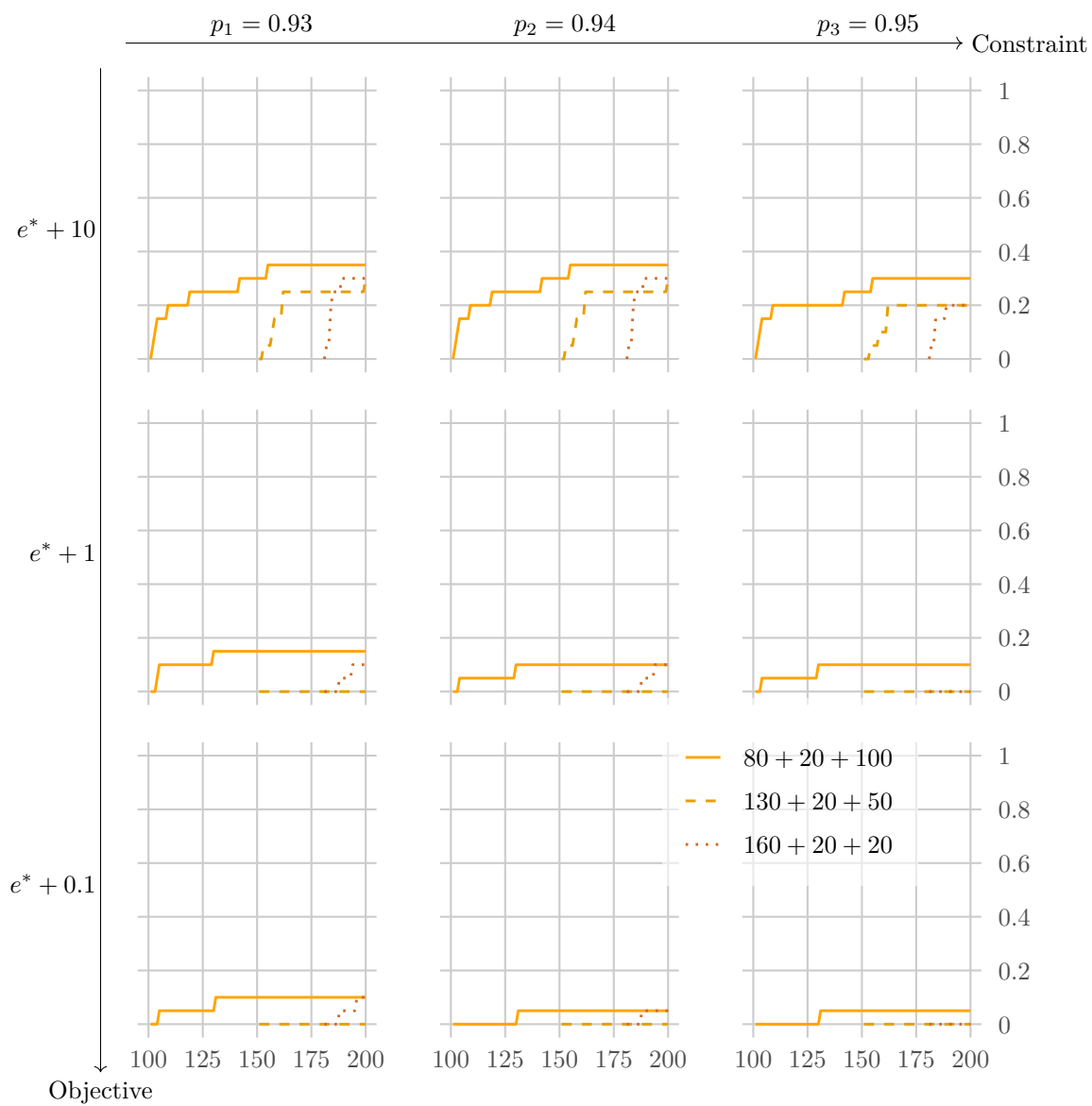
| Budget allocation | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 160 + 20 + 20     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
| 130 + 20 + 50     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
| 80 + 20 + 100     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |

**Table 5.8:** Percentage of kept  $X$ 's on 20 runs in the SA step of **Off-SA** algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

| Budget allocation | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 160 + 20 + 20     | 0.95  | 1     | 0.05  | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
| 130 + 20 + 50     | 0.75  | 1     | 0.05  | 0     | 0.05  | 0     | 0     | 0.05  | 0.05  | 0.05     |
| 80 + 20 + 100     | 0.45  | 1     | 0.15  | 0.05  | 0.2   | 0     | 0.05  | 0     | 0.05  | 0.05     |

**Table 5.9:** Percentage of kept  $U$ 's on 20 runs in the SA step of **Off-SA** algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

The obtained data profiles show a really bad performance of the algorithm on the functions  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ . Even the simplest goal is reached between 20% and 35% of the runs. The reason for this lies in the screening table, where we see that  $X_{10}$  is never detected as influential and so EFISUR is run in the first two dimensions with the other inputs, in particular  $x_{10}$ , fixed at random. Therefore, unless  $x_{10}$  is luckily fixed at a value close to four, the algorithm has no chance to reach the true optimum. Since we don't manage to detect  $x_{10}$  with the largest possible initial part of the budget (here 160), there is no hope the other allocations. The failure to detect  $x_{10}$  was expected, since in Section 5.1.2 with real functions evaluations (without any GP approximations) we had to zoom on the southwest quadrant to be able to detect  $X_{10}$ . For the uncertain inputs, we were able to detect



**Figure 5.9:** Data profiles of Off-SA algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

$U_{10}$  easily enough with the true functions, but this is not the case at all with the GPs. Even  $U_1$  is not often retained when the GPs are fit to the smallest budget of 80 points. These difficulties in detecting inputs that locally affect the optimization are reasons to add SA steps along the EFISUR iterations.

## 5.2.2 Off-seq-SA: Multiple SA steps outside EFISUR iterations

In this section, we now test **Off-seq-SA** (algorithm 2), summarized in Scheme 4.2, which consists of performing several SA steps between series of EFISUR iterations to modify the kept inputs for the next iterations. A main difference here is the use of noisy GPs. They allow to consider the impact of the removed inputs on the optimized function as noise of the evolving reduced model. An important choice in the algorithm is the number of SA steps. It seems appropriate to perform enough EFISUR iterations between two SA steps, so that the algorithm has time to find a better optimum. Since the main goal of the method is to be able to detect the added inputs  $x_{10}$  and  $U_{10}$ , we test the algorithm only on  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

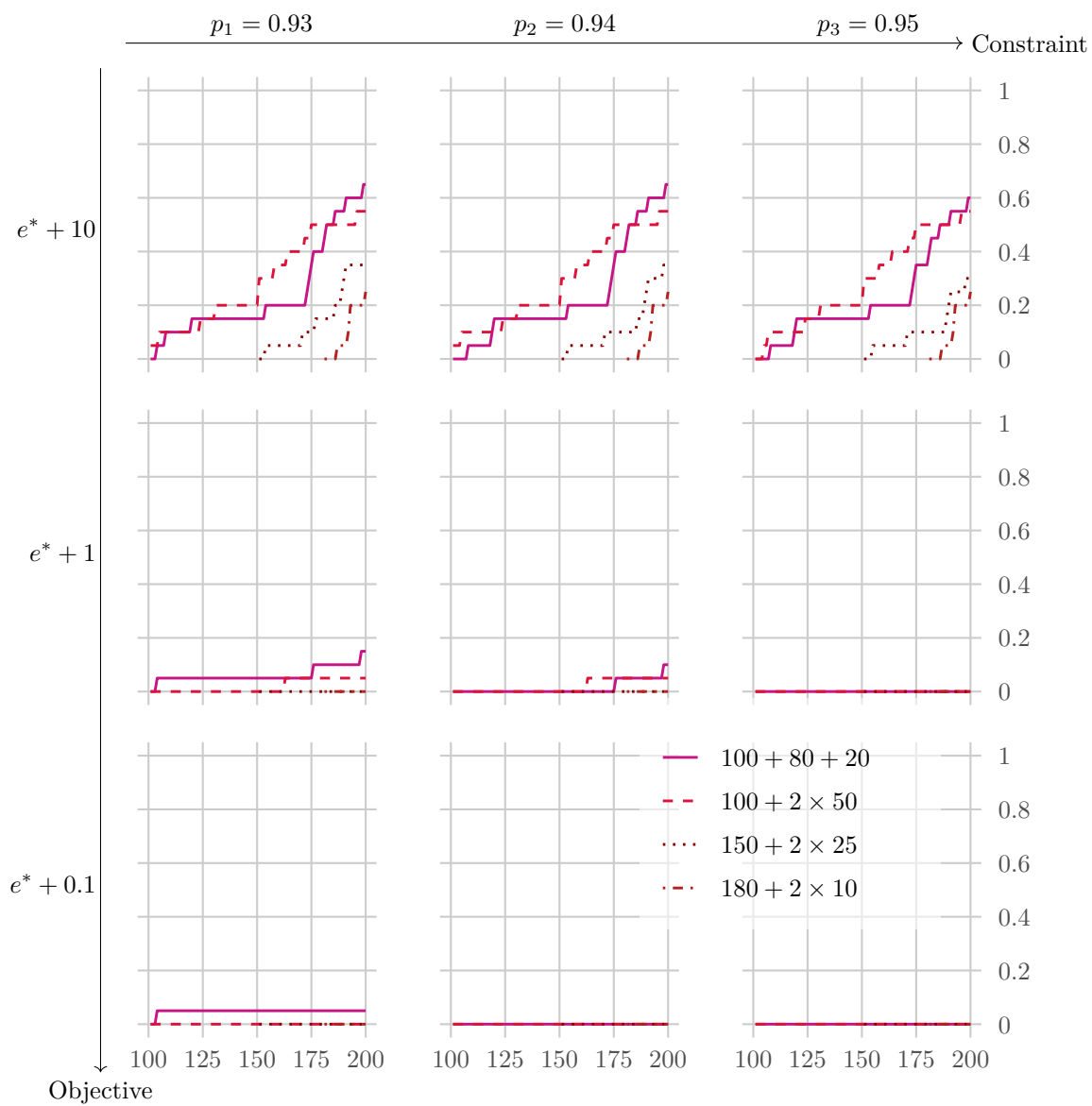
### 5.2.2.a Off-seq-SA with two SA steps

We first propose to only do two SA steps. The first one aims at reducing a lot the dimension to reach areas of interest. Then a finer SA step aims at also detecting locally influential inputs.

With noisy GPs, evaluations used to fit the GPs in high dimension and perform the SA can be reused to fit the noisy GPs in low dimension. No additional budget is then required for small dimension initialization. We test four budget allocations. Three of them by dividing the iterations budget in two equal parts:  $180 + 2 \times 10$ ,  $150 + 2 \times 25$ ,  $100 + 2 \times 50$ . And one,  $100 + 80 + 20$  where more iterations (80) are done after the 1st SA step than after the second (20). As the first SA step is intended to reduce a lot the dimension, we limit the number of selected inputs by keeping only the two most influential ones according to their rank in the ANOVA decomposition. Data profiles are plotted in Figure 5.10 for the four budget allocations and the screening results of both steps are given in Tables 5.10 and 5.11.

| Budget allocation   | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$    |
|---------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------------|
| $100 + 80 + 20$     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0           |
|                     | 1     | 1     | 0.1   | 0.2   | 0.05  | 0.1   | 0.15  | 0.15  | 0     | <b>0.35</b> |
| $100 + 2 \times 50$ | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0           |
|                     | 1     | 1     | 0.2   | 0.15  | 0.05  | 0.05  | 0.1   | 0.2   | 0     | <b>0.3</b>  |
| $150 + 2 \times 25$ | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0           |
|                     | 1     | 1     | 0.2   | 0     | 0.05  | 0     | 0.1   | 0.2   | 0.1   | <b>0.15</b> |
| $180 + 2 \times 10$ | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0           |
|                     | 1     | 1     | 0.2   | 0.05  | 0.1   | 0.15  | 0.1   | 0.05  | 0.05  | <b>0.15</b> |

**Table 5.10:** Percentage of kept  $X$ 's on 20 runs in the first SA step (1st row) and second SA step (2nd row) of **Off-seq-SA** algorithm with two SA steps for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .



**Figure 5.10:** Data profiles of **Off-seq-SA** algorithm with two SA steps for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with four budget allocations for 20 runs associated to different initial lhs design.

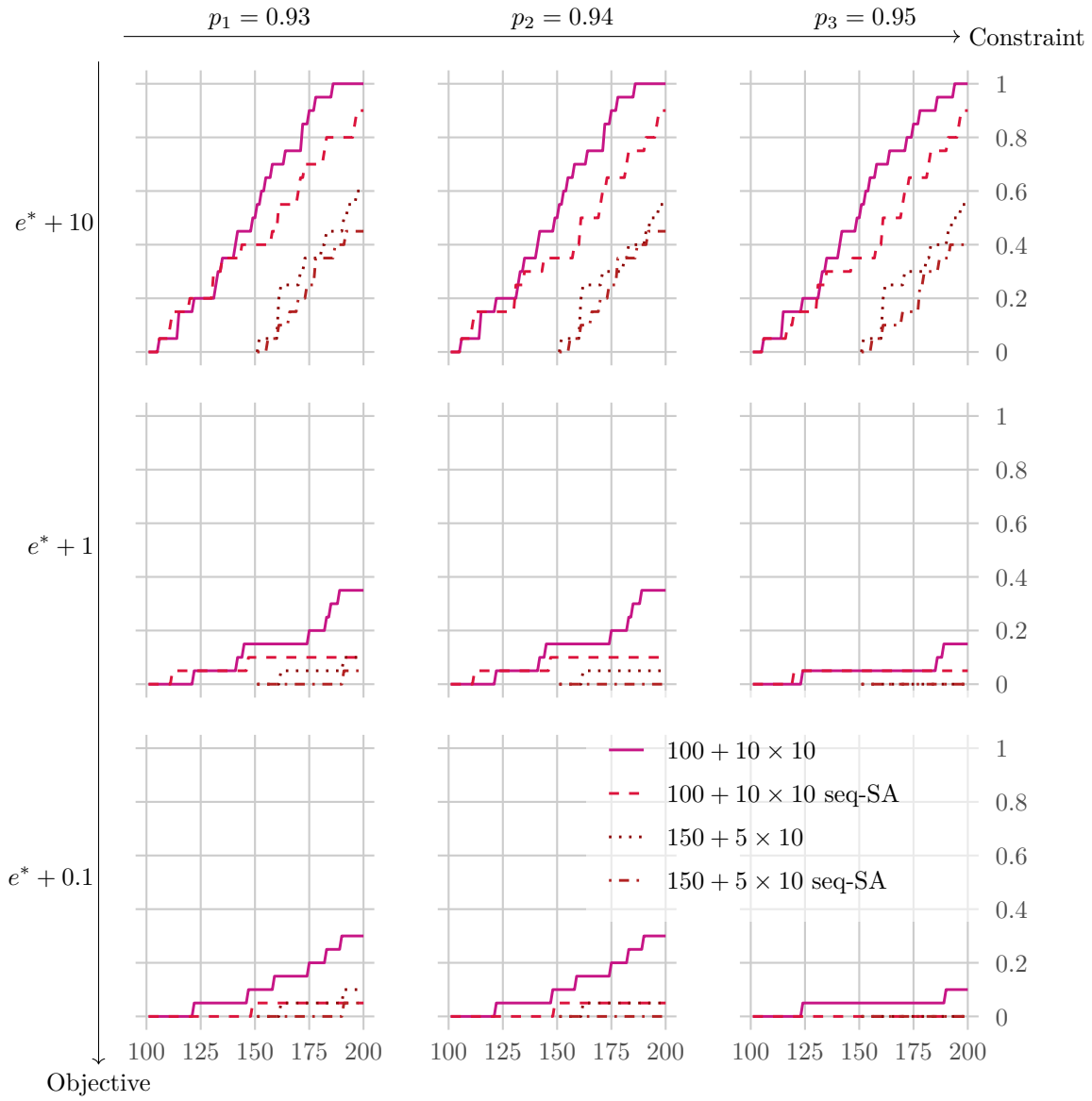
| Budget allocation | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$    |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------------|
| 100 + 80 + 20     | 0.45  | 1     | 0.1   | 0.05  | 0.05  | 0     | 0.1   | 0.1   | 0     | 0.15        |
|                   | 0.8   | 0.95  | 0.45  | 0.2   | 0.15  | 0.2   | 0.15  | 0.2   | 0.1   | <b>0.35</b> |
| 100 + 2 × 50      | 0.7   | 1     | 0.1   | 0.05  | 0.05  | 0     | 0     | 0.05  | 0     | 0.05        |
|                   | 0.9   | 1     | 0.2   | 0.2   | 0.1   | 0.15  | 0.25  | 0.25  | 0.15  | <b>0.3</b>  |
| 150 + 2 × 25      | 0.95  | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0.05  | 0           |
|                   | 0.65  | 0.9   | 0.3   | 0.1   | 0.2   | 0.1   | 0.1   | 0.1   | 0.2   | <b>0.1</b>  |
| 180 + 2 × 10      | 0.95  | 1     | 0     | 0.05  | 0     | 0     | 0     | 0     | 0     | 0           |
|                   | 0.85  | 0.9   | 0.2   | 0.2   | 0.15  | 0.15  | 0.2   | 0.1   | 0.1   | <b>0.2</b>  |

**Table 5.11:** Percentage of kept  $U$ 's on 20 runs in the first SA step (1st row) and second SA step (2nd row) of **Off-seq-SA** algorithm with two SA steps for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

The obtained tables are better than the tables with only one SA step. For the  $X$ 's, we again always keep  $X_1$  and  $X_2$ , and we also sometimes manage to keep  $X_{10}$  in the second SA step, even if it remains rare. The more iterations are done after the 1st SA step, the more runs are able to keep  $X_{10}$  after the second SA step. Similar observations can be made for the uncertain inputs. The data profiles (Figure 5.10) are slightly better than with only one SA step for the goal  $e^* + 10$ . For the other goals, they are very unsatisfactory and even worse than with only one SA step. For example, even reaching  $e^* + 1$  with  $p_2 = 0.94$  is achieved in less than 10% of runs, regardless of the budget allocation. As the SA results are better, it means that the degradation of the results comes from the use of noisy GPs. Even if the algorithm keep  $X_{10}$  and  $U_{10}$  to refine the optimum, the GP noise seems to hide small potential improvements. However, if all influential inputs are retained, the noise should be small enough that it doesn't interfere with the search for better points. This means that we need to improve the way we deal with noise. Even if noisy GPs were perfectly fitted, a major problem with using only two SA steps is that there is only one chance to detect inputs that are influential only when approaching the optimum. If it is not detected, there are no more steps to find it. On the other hand, if it is detected, there is no chance to forget it in a future SA step.

### 5.2.2.b Off-seq-SA with multiple SA steps

In this part, **Off-seq-SA** is tested with one SA step every 10 EFISUR iterations. We test initial design of 150 and 100 points resulting in budgets allocations denoted  $150 + 5 \times 10$  and  $100 + 10 \times 10$ . First the SA steps are made without limiting the number of kept inputs. Then we also propose to limit the number of inputs kept during the algorithm. We propose to do it sequentially along the SA steps. For an initial budget of 150, we limit the number of possible retained inputs to 2 at the first iteration and increase it by 2 at each SA step, so that all inputs might be kept in the last step. The corresponding runs are denoted  $150 + 5 \times 10$  - seqSA. Similarly,  $100 + 10 \times 10$  - seqSA is done by increasing the number of possible kept inputs by 1 one at each SA step. Since we are mostly interested in the SA result of  $X_{10}$  and  $U_{10}$ , the screening results for these two inputs only are given in Table 5.12. Complete screening tables can be found in Tables E.1 and E.2 in the Appendix. The data profiles are plotted in Figure 5.11.



**Figure 5.11:** Data profiles of **Off-seq-SA** algorithm with a SA step every 10 iterations for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals, two budget allocations, with and without limiting the number of kept inputs for 20 optimization runs.

| Budget allocation     |          | Sensitivity Analysis steps |      |      |      |      |      |      |      |      |      |
|-----------------------|----------|----------------------------|------|------|------|------|------|------|------|------|------|
| 100 + 10 × 10         | $X_{10}$ | 0                          | 0.15 | 0.15 | 0.4  | 0.65 | 0.75 | 0.85 | 0.95 | 1    | 1    |
|                       | $U_{10}$ | 0.05                       | 0.4  | 0.55 | 0.4  | 0.35 | 0.4  | 0.4  | 0.7  | 0.7  | 0.6  |
| 100 + 10 × 10 - seqSA | $X_{10}$ | 0                          | 0.15 | 0.2  | 0.2  | 0.4  | 0.5  | 0.55 | 0.6  | 0.8  | 0.95 |
|                       | $U_{10}$ | 0                          | 0.1  | 0.05 | 0.15 | 0.1  | 0.25 | 0.35 | 0.3  | 0.5  | 0.5  |
| 150 + 5 × 10          | $X_{10}$ | 0                          |      | 0.15 |      | 0.25 |      | 0.4  |      | 0.8  |      |
|                       | $U_{10}$ | 0.05                       |      | 0.35 |      | 0.3  |      | 0.2  |      | 0.1  |      |
| 150 + 5 × 10 - seqSA  | $X_{10}$ | 0                          |      | 0.1  |      | 0.1  |      | 0.35 |      | 0.6  |      |
|                       | $U_{10}$ | 0.1                        |      | 0.15 |      | 0.15 |      | 0.3  |      | 0.15 |      |

**Table 5.12:** Percentage of kept  $X_{10}$  and  $U_{10}$  on 20 runs in **Off-seq-SA** algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  along 5 or 10 SA steps.

We observe in Table 5.12 that  $X_{10}$  is kept more and more often during the algorithm, regardless of the budget allocation. For example, with an initial budget of 100 points and 10 SA steps, we manage to always keep  $X_{10}$  in the last two sensitivity analysis steps. Unexpectedly,  $U_{10}$  seems to be much harder to detect than  $X_{10}$ , with e.g. at most 70% of the runs managing to keep it. This wasn't expected, since the SA results with the true functions of Section 5.1.2 showed that  $U_{10}$  is easier to detect than  $X_{10}$ . The sensitivity analysis on the sets seems to be more hindered by the Gaussian process approximation than the SA on the  $X$ 's. This improvement of the screening steps results in improvement in the data profiles. Figure 5.11 reveal an increase of the number of runs that achieve any goal. Again, the more iterations, the better the results. Sequentially limiting the number of retained inputs seems to degrade the results a bit in both the data profiles and the SA results.

A strength of the proposed methods is that the GPs used to perform EFISUR are trained in a small dimension. This means that the algorithm can be used even if the initial dimension is very large. Only the GP regressions for the SA steps are done in the initial dimension and may become difficult as the dimension increases.

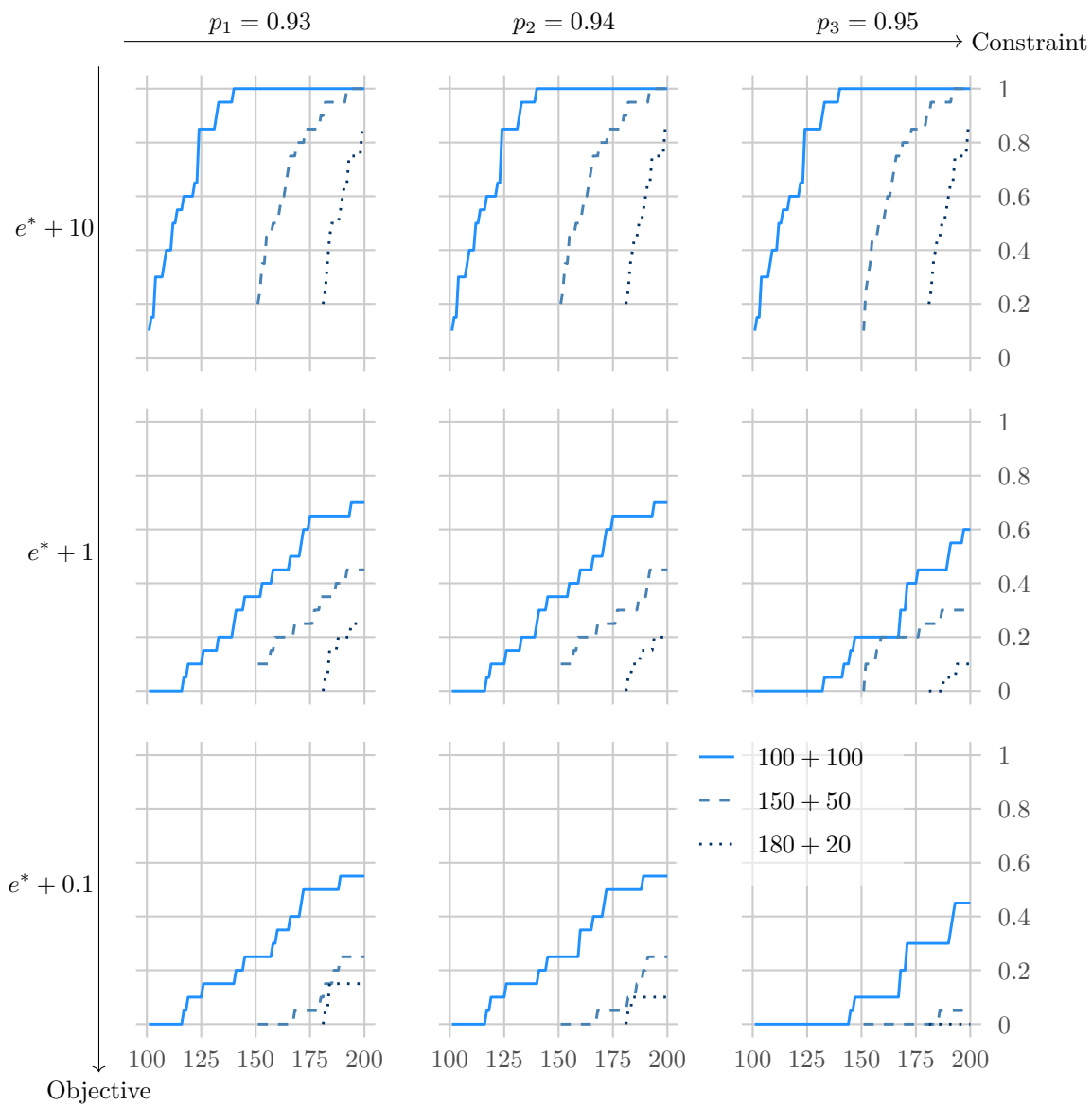
The previous sections showed how **Off-SA** and **Off-seq-SA** algorithms behave on two test cases of different difficulties. The algorithms give promising results, which opens up the use of EFISUR in higher dimensions. The algorithms can clearly be improved, notably in the use of noisy GPs, but they give a first answer to the use of SA in EFISUR. Now, instead of reducing the dimension by SA outside the iterations of EFISUR, we add SA steps inside the optimizer iterations.

### 5.3 Online Sensitivity Analysis in EFISUR: numerical results on the toy functions

In this part we test the second method, i.e. **On-SA** and **On-2steps-SA** (algorithms 3 and 4).

#### 5.3.1 On-SA: Dimension reduction by SA on the acquisition functions

**On-SA** (algorithm 3) consists of a classical EFISUR with an additional SA step before each optimization of the two acquisition functions. Each SA step is performed either on  $\mathcal{E}$  or on  $\mathcal{S}$  to reduce the dimension of  $\mathcal{X}$  and  $\mathcal{U}$ . This allows some time gain in computation, since the optimization of the acquisition criteria should take less time to converge. The quality of the found optimum of the acquisition functions may also be improved, since a smaller dimension reduces the chance of missing the global optimum. The same budget allocations as for EFISUR are tested. We first run the algorithm with  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . The resulting data profiles are shown in Figure 5.12. They



**Figure 5.12:** Data profiles of On-SA algorithm for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

show that the algorithm performs quite well on  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ , which means that reducing the dimension of the acquisition function optimization didn't degrade the convergence. The screening results given in Table 5.13 confirm that  $X_1$ ,  $X_2$  are always kept for the optimization of the EFI. However, the results are very different for the SA steps on  $S$  given in Table 5.14. We observe that each uncertain input is sometimes conserved without much difference for  $U_1$  and  $U_2$ . Even if some uncertain inputs should not affect the output in areas of interest, they are recognized as important to reduce the variance of the feasible improvement at the added point. Interestingly, we observe in the last column, called EFIRAND, that neglecting all inputs and randomly drawing the next  $U$  happens a significant number of times during the runs (about 50% on the three budget allocations). In Table 5.15 we detail the evolution of this last column during the iterations. We observe that the number of times no  $U_i$  is kept increases with the iterations. This means that during the first iterations of the algorithm, it is crucial to choose the new  $U$  wisely (by optimizing  $S$ ). However, this importance decreases drastically with iterations. This is an indicator that optimizing  $S$  is not always necessary and can be a waste of time, which our algorithm avoids.

| Budget allocation | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 100 + 100         | 1.00  | 1.00  | 0.36  | 0.15  | 0.09  | 0.08  | 0.10  | 0.10  | 0.08  | 0.09     |
| 150 + 50          | 1.00  | 1.00  | 0.31  | 0.14  | 0.10  | 0.08  | 0.08  | 0.11  | 0.07  | 0.07     |
| 180 + 20          | 1.00  | 1.00  | 0.30  | 0.13  | 0.08  | 0.10  | 0.09  | 0.10  | 0.09  | 0.09     |

**Table 5.13:** Percentage of kept  $X$ 's for **On-SA** algorithm for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ , averaged on 20 runs and on every iterations.

| Budget allocation | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | EFIRAND |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|---------|
| 100 + 100         | 0.31  | 0.24  | 0.22  | 0.20  | 0.21  | 0.21  | 0.21  | 0.21  | 0.21  | 0.21     | 0.58    |
| 150 + 50          | 0.21  | 0.32  | 0.24  | 0.20  | 0.23  | 0.26  | 0.24  | 0.16  | 0.20  | 0.25     | 0.57    |
| 180 + 20          | 0.23  | 0.35  | 0.18  | 0.21  | 0.17  | 0.24  | 0.25  | 0.19  | 0.26  | 0.21     | 0.44    |

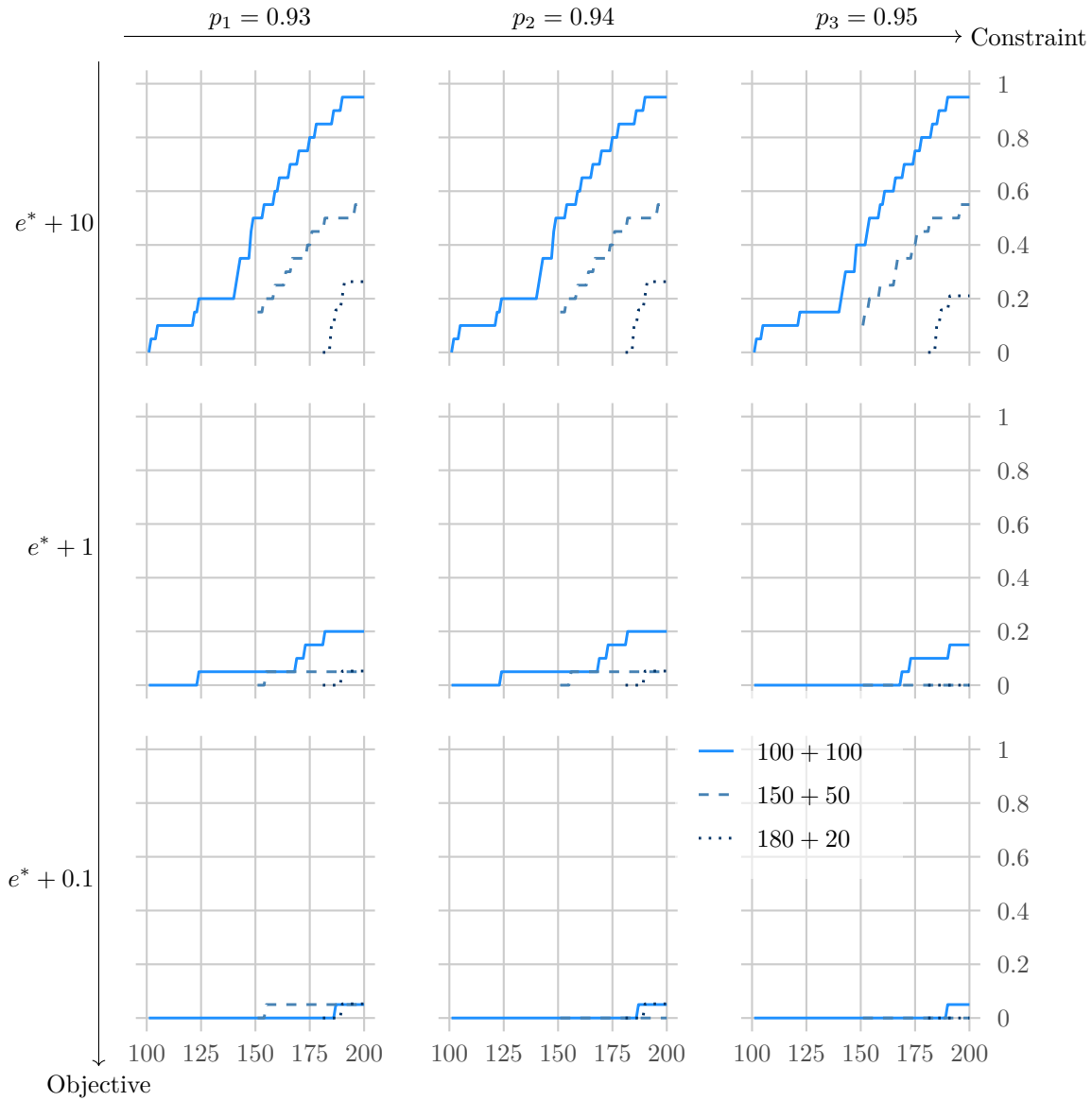
**Table 5.14:** Percentage of kept  $U$ 's for **On-SA** algorithm for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ , averaged on 20 runs and on every iterations. EFIRAND column correspond to the iterations where none  $U_i$  is kept.

| Budget allocation    | 100 + 100 | 150 + 50 | 180 + 20 |
|----------------------|-----------|----------|----------|
| 1 to 10 iterations   | 0.26      | 0.285    | 0.345    |
| 11 to 20 iterations  | 0.41      | 0.595    | 0.54     |
| 21 to 30 iterations  | 0.45      | 0.595    |          |
| 31 to 40 iterations  | 0.57      | 0.655    |          |
| 41 to 50 iterations  | 0.57      | 0.695    |          |
| 51 to 60 iterations  | 0.67      |          |          |
| 61 to 70 iterations  | 0.71      |          |          |
| 71 to 80 iterations  | 0.68      |          |          |
| 81 to 90 iterations  | 0.70      |          |          |
| 91 to 100 iterations | 0.76      |          |          |

**Table 5.15:** Evolution of the averaged times when none uncertain input is kept after the SA on  $S$  for **On-SA** algorithm run 20 times for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$

In Figure 5.13, the data profiles obtained for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  show that the efficiency of the algorithm decreases in the presence of locally influential inputs. The algorithm doesn't even manage

to reach the intermediate objective of  $e^* + 1$ . Interestingly, the results don't seem to change when the constraint is relaxed (from right to left). In this algorithm, the GPs are built in the initial dimension, so the failure of the algorithm can only be explained by failures of the SA steps in keeping  $X_{10}$  for the EFI optimization. To verify this, the evolution of the screening results for  $X_{10}$  is given in Table 5.16. We observe that  $X_{10}$  is kept to optimize the EFI only after several iterations. With budget allocations of  $150 + 50$  and  $180 + 20$ , the algorithm has no time to detect  $X_{10}$ , which explains the results in these cases. Even with a hundred iterations, the algorithm keeps  $X_{10}$  only 75% of the time in the last 10 iterations, where we would hope  $X_{10}$  to be always kept. Adding another SA step should aid to keep  $X_{10}$ .



**Figure 5.13:** Data profiles of **On-SA** algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

| Budget allocation    | 100 + 100 | 150 + 50 | 180 + 20 |
|----------------------|-----------|----------|----------|
| 1 to 10 iterations   | 0.14      | 0.07     | 0.12     |
| 11 to 20 iterations  | 0.15      | 0.21     | 0.14     |
| 21 to 30 iterations  | 0.22      | 0.24     |          |
| 31 to 40 iterations  | 0.30      | 0.25     |          |
| 41 to 50 iterations  | 0.43      | 0.40     |          |
| 51 to 60 iterations  | 0.47      |          |          |
| 61 to 70 iterations  | 0.65      |          |          |
| 71 to 80 iterations  | 0.66      |          |          |
| 81 to 90 iterations  | 0.76      |          |          |
| 91 to 100 iterations | 0.75      |          |          |

**Table 5.16:** Evolution of the averaged times  $X_{10}$  is kept after the SA on the EFI for **On-SA** algorithm ran 20 times for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

### 5.3.2 On-2steps-SA: Two-steps dimension reduction by SA on the acquisition functions

#### 5.3.2.a On-2steps-SA

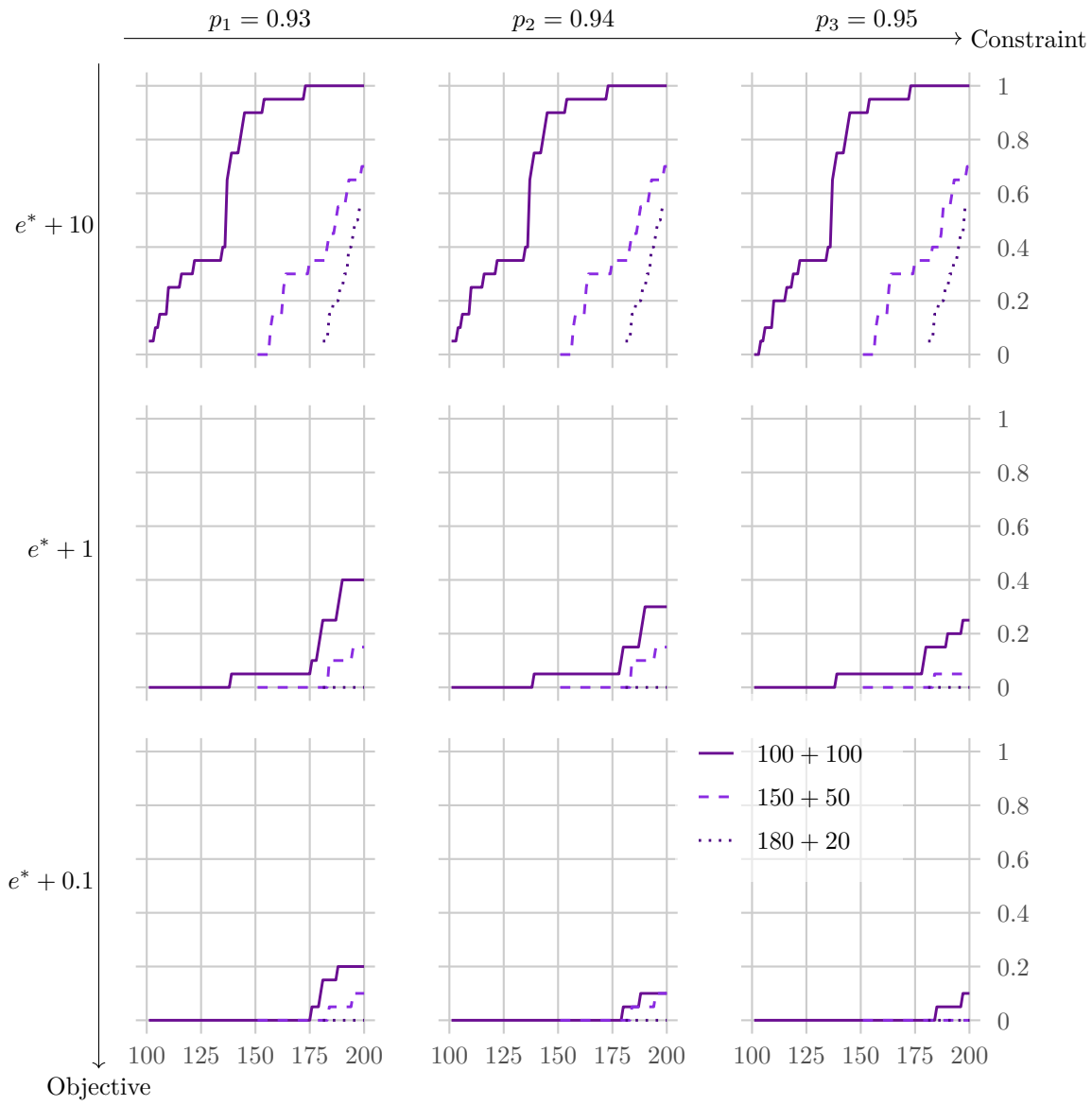
**On-2steps-SA** (algorithm 4) consists in adding another SA step in the optimization of the acquisition functions to detect locally influential variables. Since one SA step was sufficient for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ , the algorithm is tested only for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ , and the resulting data profiles are shown in Figure 5.14. The added SA step greatly increases the efficiency of the algorithm. For example, with 100 initial points, 90% of the runs reach  $e^* + 10$  for any constraint level after only 50 iterations, while the previous algorithm reached it after iteration 90. This is also the case for larger initial design. Table 5.17 confirms that the added SA step greatly increases the number of times  $X_{10}$  is kept.

| Budget allocation    | 100 + 100 | 150 + 50 | 180 + 20 |
|----------------------|-----------|----------|----------|
| 1 to 10 iterations   | 0.25      | 0.21     | 0.21     |
| 11 to 20 iterations  | 0.42      | 0.50     | 0.44     |
| 21 to 30 iterations  | 0.66      | 0.73     |          |
| 31 to 40 iterations  | 0.71      | 0.79     |          |
| 41 to 50 iterations  | 0.79      | 0.83     |          |
| 51 to 60 iterations  | 0.86      |          |          |
| 61 to 70 iterations  | 0.88      |          |          |
| 71 to 80 iterations  | 0.94      |          |          |
| 81 to 90 iterations  | 0.93      |          |          |
| 91 to 100 iterations | 0.89      |          |          |

**Table 5.17:** Evolution of the averaged times  $X_{10}$  is kept with two SA steps on the EFI for **On-2steps-SA** algorithm ran 20 times for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

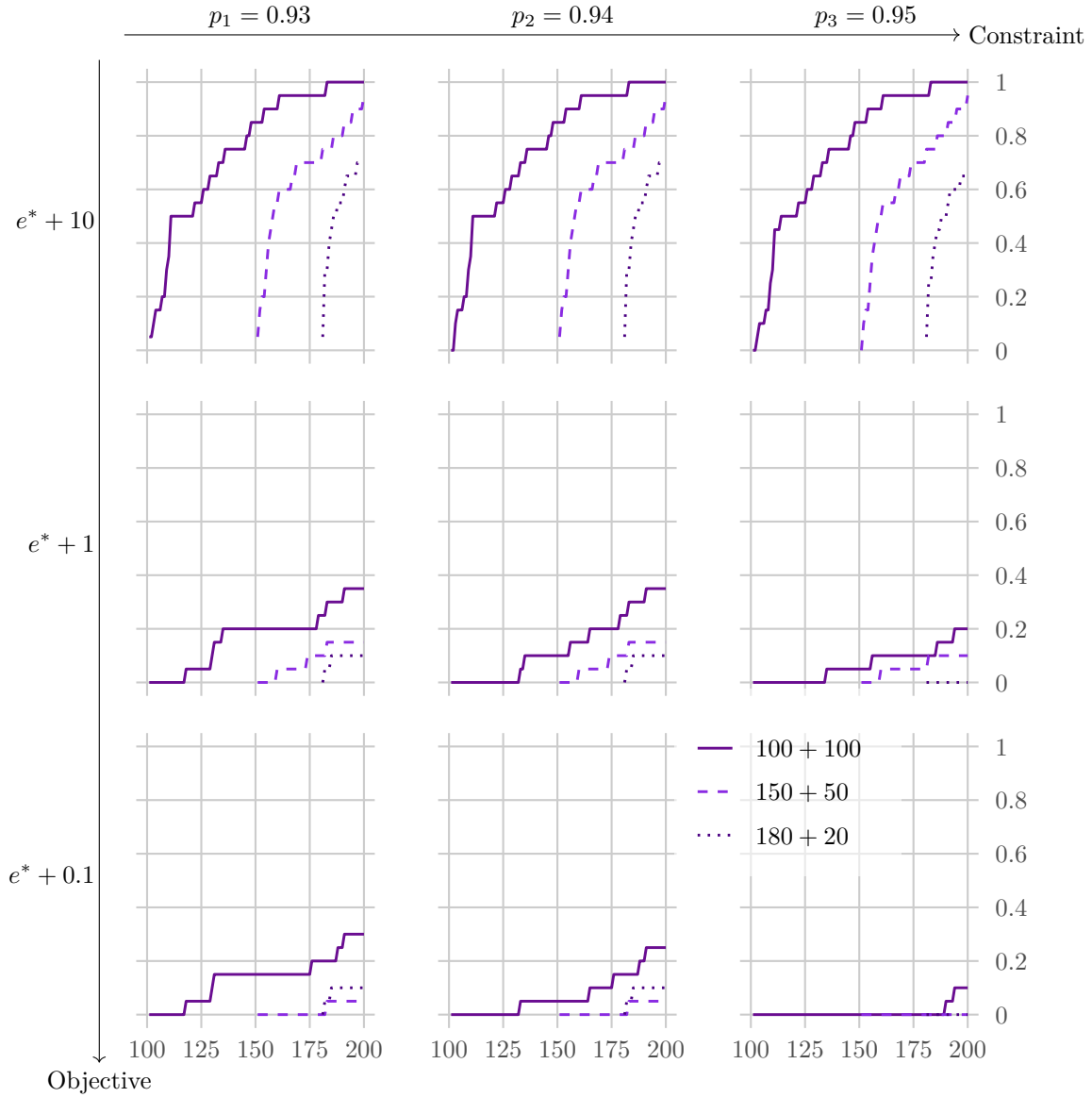
#### 5.3.2.b On-2steps-SA-free

One of the reasons the results remain moderate, especially for hard objectives, is that with this approach  $x_{10}$  is often not simultaneously optimized with  $x_1$  and  $x_2$ . Indeed,  $x_1$  and  $x_2$  are mostly optimized after the first SA step while  $x_{10}$  is only detected in the second step. This can have a negative impact on the optimization, especially if the inputs interact with each other. To overcome this problem, we test **On-2steps-SA-free**, the variant of the previous algorithm, in which the



**Figure 5.14:** Data profiles of On-2steps-SA algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

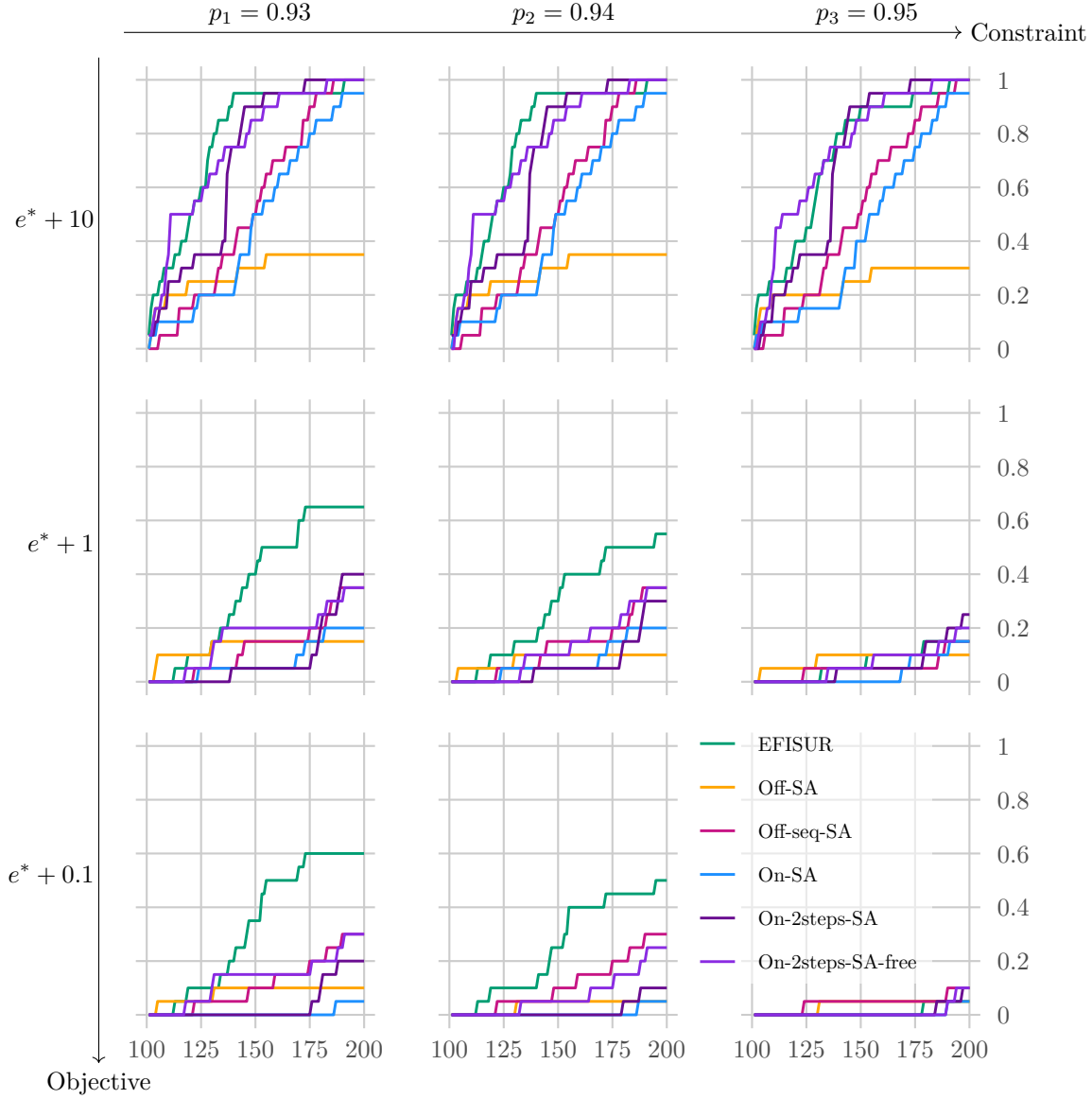
first kept inputs are no longer fixed but are again optimized in the second optimization step. This second optimization is done in a higher dimension than the first, but remains small compared to the initial dimension. To use the result of the first optimization step, the found optimum is used as a starting point for the second optimization. Associated data profiles are plotted in Figure 5.15 and we observe that, at the cost of a second optimization of the acquisition function in a larger but small dimension, we manage to obtain slightly better results.



**Figure 5.15:** Data profiles of On-2steps-SA-free algorithm for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with three budget allocations for 20 runs associated to different initial lhs design.

## 5.4 Numerical comparison of the algorithms on the toy functions

In this part we propose a comparison of the algorithms both in terms of result quality and computation time. Figure 5.16 compares all methods for 100 initial points and 100 iterations, as it gave the best results in terms of final goals reached. Except for **Off-SA**, all algorithms manage to reach



**Figure 5.16:** Data profiles of all algorithms for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and 100 iterations for 20 runs associated to different initial lhs design.

$e^* + 10$  on every run. EFISUR and both versions of **On-2steps-SA** do it a little faster than the others. For the goals  $e^* + 1$  and  $e^* + 0.1$  with  $p = 0.94$  or  $p = 0.93$ , EFISUR beats all others, followed by **Off-seq-SA** and **On-2steps-SA-free**. For the initial constraint threshold  $p_3 = 0.95$ , these two algorithms are slightly better than EFISUR. The same comparison with only 50 and 20

iterations is shown in figures E.6 and E.7 in Appendix, where **On-2steps-SA-free** is almost always better than EFISUR.

These comparisons are made only by measuring the efficiency of the algorithms in terms of the number of model evaluations. Each algorithm also requires a different computation time, which should be taken into account especially if is comparable to the time computation of  $f$  and  $g$ . The running times of each algorithm for the functions  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  are given in Table 5.18.

| Duration of       | Off-SA | GP   | EFI | S     | SA-x | SA-u | rest | total |
|-------------------|--------|------|-----|-------|------|------|------|-------|
| EFISUR            | 0.00   | 1400 | 729 | 10200 | 0.00 | 0.00 | 45.2 | 12400 |
| Off-SA            | 76.4   | 16.8 | 132 | 353   | 0.00 | 0.00 | 0.24 | 502   |
| Off-seq-SA        | 177    | 317  | 193 | 699   | 0.00 | 0.00 | 252  | 1460  |
| On-SA             | 0.00   | 1370 | 306 | 800   | 127  | 164  | 46.0 | 2810  |
| On-2steps-SA      | 0.00   | 1390 | 481 | 949   | 225  | 164  | 46.9 | 3260  |
| On-2steps-SA-free | 0.00   | 1380 | 603 | 912   | 226  | 159  | 47.4 | 3330  |

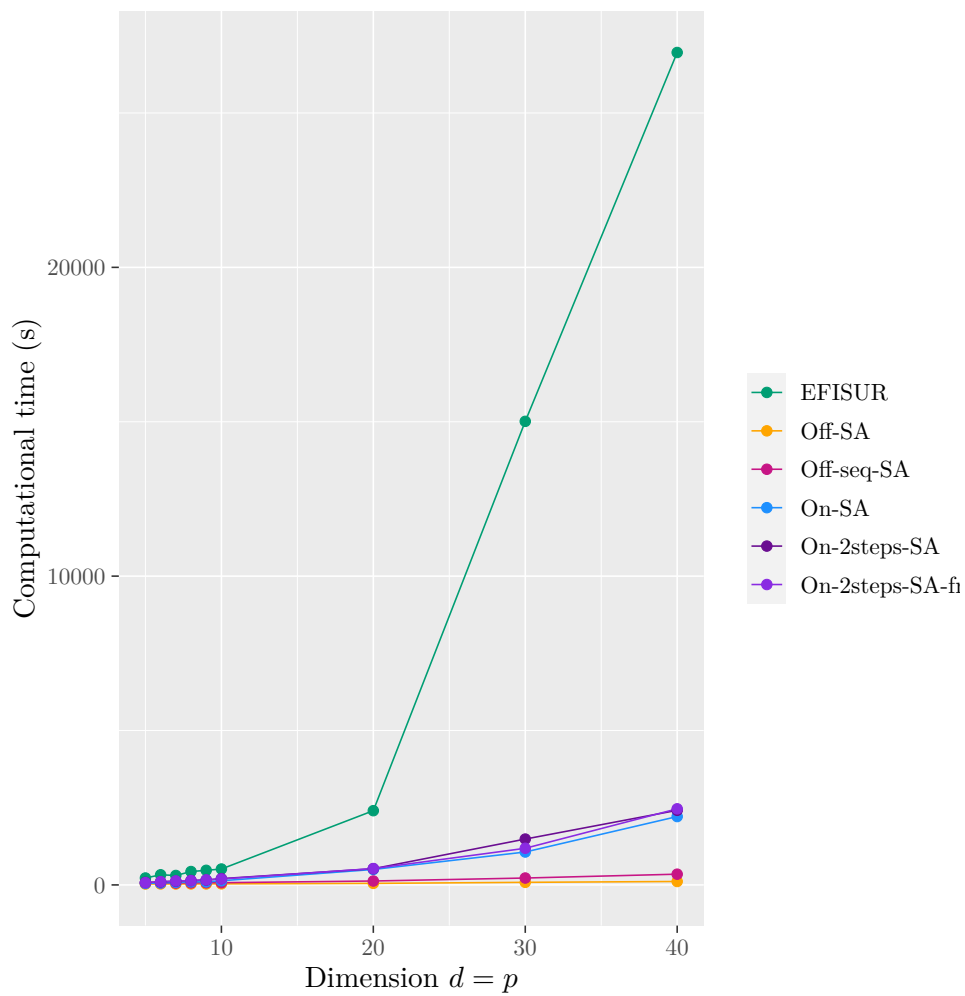
**Table 5.18:** Averaged computation time in seconds for each algorithm and for each inner step for 180 starting points and 20 iterations. From left to right, the inner steps are the SA steps outside EFISUR, the Gaussian process modeling, the optimization of EFI and S, the inner SA steps, and some remaining costs for 20 optimization runs.

We observe that EFISUR requires four times the computation time of online SA algorithms. The gain is mainly due to the optimization of  $S$ , which consumes most of the time of EFISUR. For offline algorithms, the reduced dimension of the GP allows to gain time both in the fitting of the GPs and in the optimization of EFI and  $S$ , which leads to a huge time gain. However, in the case of expensive  $f$  and  $g$ , these gains may be small compared to the time required to compute  $f$  and  $g$ .

The time computations of a single step of each algorithm are computed in Figure 5.17 as the input dimension increases. For the two Online-SA algorithms, we simply add the time computation of the SA step. For the Offline-SA algorithms, as the SA steps are used for multiple EFISUR iterations, we divide the SA computation time by 10, which is the minimum number of iterations for which one SA step is used. The time difference was already important in the dimension  $d = p = 10$ , but it increases a lot with the dimension. Both **Off-SA** and **Off-seq-SA** keep a very small computation time, since the dimension increase only affects the time of the SA steps. However, for these steps to be trustworthy, the initial modeling of the true functions must be good, which becomes more difficult in high dimension. For the Offline SA algorithms, we gain a lot compared to EFISUR with about 10 times less computation time in dimension  $30 + 30$  when it was only four times in dimension  $10 + 10$ . This gain is explained by the fact that the computation time of the acquisition functions, and also the number of evaluations required for their optimization, explode with the dimension. This difference is mitigated by the SA analysis steps of the algorithm, but the latter can be parallelized while the optimization of the acquisition functions cannot.

## 5.5 Application: Optimization of an electric machine

In this last part, we apply our algorithms to the test case previously presented in Section 2.4.3, from [137]. In [137], a sensitivity analysis is performed in the context of the robust conception of an electrical machine. The machine studied is a permanent magnet synchronous reluctance motor, which is one of the most commonly used machines in electric vehicles today. The aim of the authors is to take into account the uncertainties (manufacturing and assembly tolerances) on the geometrical and magnetic parameters of the machine components in the optimization of the mean torque (to



**Figure 5.17:** Computation time of one step of each algorithm measured for increasing input dimensions  $d = p$  for  $f_{Loc}^d$  and  $g_{Loc}^d$ .

be maximized) and the torque ripple (to be minimized). The resulting optimization problem is

$$\min_{x \in \mathcal{X}} (\mathbb{E}[f_{ripp}(x, U)], \mathbb{E}[-g_{torq}(x, U)]), \quad (5.2)$$

where  $f_{ripp}$  is the torque ripple and  $g_{torq}$  is the mean torque both defined on  $\mathcal{X} \subset \mathbb{R}^{12}$  and  $\mathcal{U} \subset \mathbb{R}^{14}$ .  $U_1$  to  $U_{12}$  are manufacturing tolerances at each  $x$ , while  $U_{13}$  and  $U_{14}$  describe the magnetic material properties. The range of the  $x$  and the laws of the  $U$  were given in Figure 2.2 of Chapter 2. After performing a sensitivity analysis on the uncertain inputs using  $S_j^{H_{set}}$  of the excursion sets of the optimization problem, we obtained in Chapter 2 that the most influential uncertain inputs are, in this order,  $U_{13}$ ,  $U_{14}$ ,  $U_2$ ,  $U_3$ , and  $U_5$  (see Figure 2.3) and then, to a lesser extent,  $U_8$  and  $U_1$ . The rest were found to be negligible in the multi-objective optimization problem.

In our case, the algorithms are made for single objective optimization with constraints. The problem (5.2) is adapted and we aim to minimize the expectation of the torque ripple  $f_{ripp}$ , while the probability of the mean torque  $g_{torq}$  being higher than 420 N.m is above 95% as defined in the following equation:

$$x^* = \arg \min_{x \in \mathcal{K}} \mathbb{E}[f_{ripp}(x, U)] \text{ where } \mathcal{K} = \{x \in \mathcal{X}, \mathbb{P}[420 - g_{torq}(x, U) \leq 0] \geq 0.95\}.$$

For this problem, we run each algorithm a single time on a budget of  $10 \times (12 + 14) = 260$  model evaluations. We only test them with an initial design half the size of the budget (130 points) because it gave the best results on the toy functions. In Figure 5.18 we plot the evolution of the current minimum for each algorithm. At first it seems that EFISUR beats every algorithm. But the obtained points minimize the objective function and respect the constraint only through their metamodels  $Z$  and  $C$ . The latter have potentially large errors, especially if the GPs are in large dimension. On the toy functions, we could estimate the true probability and the true expectation at the current optimum, using important number of  $f$  and  $g$  evaluations. However, in this case and in real cases, this is not affordable, so a more clever validation step is needed.

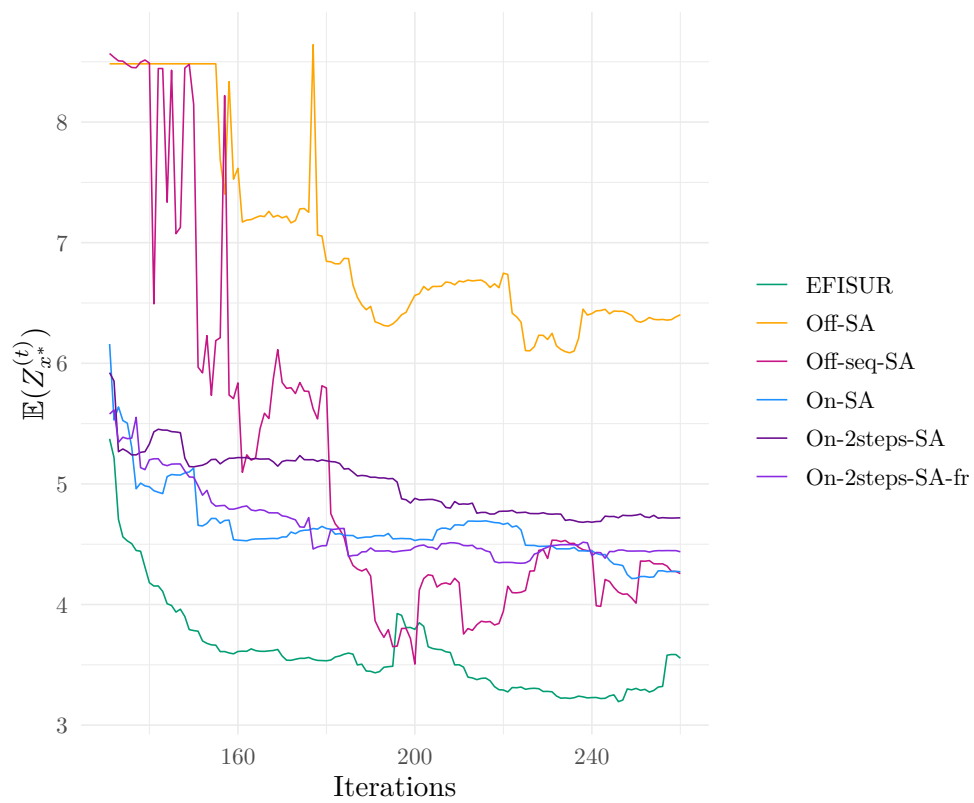
### 5.5.1 Validation

Without uncertainties, we can simply evaluate some of the best points given by the algorithm on the real model and keep the best ones. With uncertainties, we don't have access to true evaluations of  $\mathbb{E}[f_{ripp}(x, U)]$  and  $\mathbb{P}[420 - g_{torq}(x, U) \leq 0]$ , but only to evaluations of  $f_{ripp}(x, u)$  and  $g_{torq}(x, u)$ . An important validation budget is then required so that  $\mathbb{E}[f_{ripp}(x, u)]$  and  $\mathbb{P}[420 - g_{torq}(x, u) \leq 0]$  can be better estimated for some points of interest. We call these better estimates *high fidelity estimates* and estimate them with additional evaluations of  $f_{ripp}$  and  $g_{torq}$ . Since this can be an important budget compared to the optimization budget, we want to strongly limit the number of tested points, denoted  $k_{max}$ , and to choose them very carefully. In EFISUR the best point is given by:

$$x^* = \arg \min_{x \in \mathcal{X}^{(t)}} \mathbb{E}(Z_x^{(t)}) \text{ s.t. } \mathbb{E}(C_x^{(t)}) \leq 0. \quad (5.3)$$

The obtained point  $x^*$  minimizes the mean objective function while satisfying the constraint, relying on the surrogate models  $Z$  and  $C$ . However, they may have significant errors, especially in high dimension. Selecting only the best points according to this criterion may lead to points that don't respect the true constraint and may have a higher than expected mean objective function. To overcome this problem, we propose a validation methodology. However, this is a preliminary work and many changes and improvements are possible.

Our validation method consists in using the previously tested high-fidelity estimates to judge the quality of the GPs and to adjust the choice of the next point. If the GPs are trustworthy, the next



**Figure 5.18:** Value of the estimated current feasible minimum at each iterations for all algorithms with a budget of 260 model evaluations with 130 points for the initial DoE.

point is selected as the next best point using the same GPs. However, if we observe large errors in the GPs, the GPs are adjusted before selecting the next points.

In practice, we want feasible points, so we decide to focus on the constraint. To do so, we define modified constraint processes  $C_x^c$  by

$$C_x = \alpha - \mathbb{P}_U(G_{(x,U)} \leq c),$$

where the constraint threshold  $c$  can be modified. Then, to select the points where high fidelity estimates are computed, denoted  $(x_1^*, \dots, x_{k_{max}}^*)$ , we propose Algorithm 6.

---

**Algorithm 6** Validation
 

---

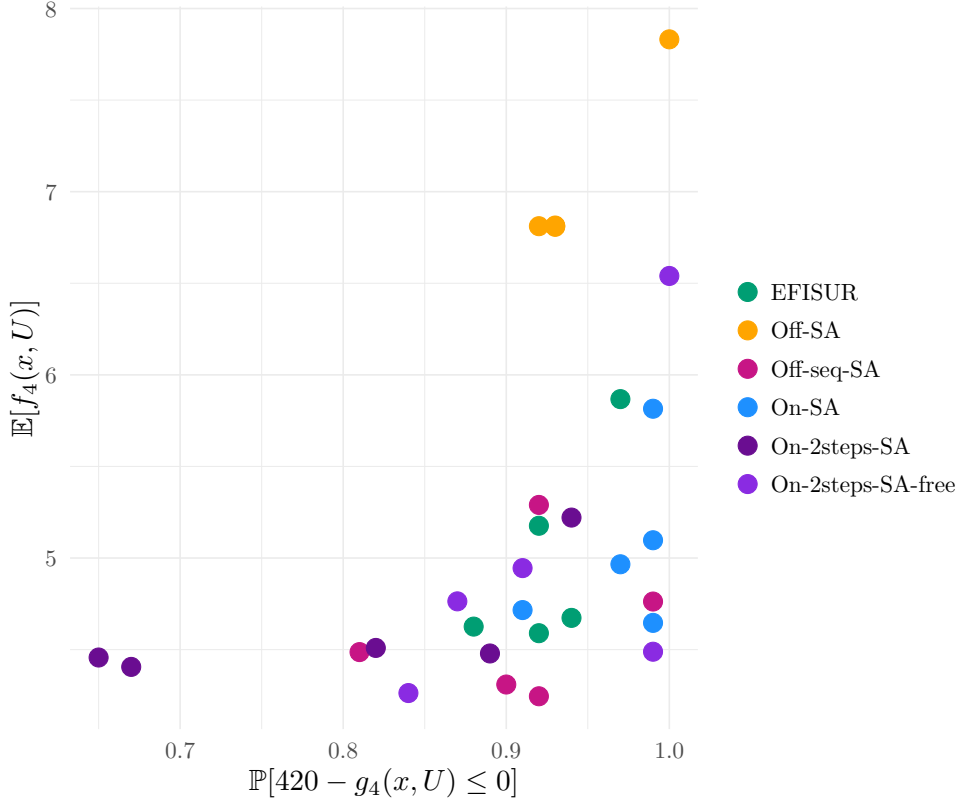
```

1:  $k \leftarrow 1, c \leftarrow 0$ 
2:  $x_k^* = \arg \min_{x \in \mathcal{X}^{(t)}} \mathbb{E}(Z_x^{(t)})$  such that  $\mathbb{E}(C_x^{c,(t)}) \leq 0$ .
3: High fidelity estimate :  $p_k = \mathbb{P}[420 - g_{torq}(x_k^*, U) \leq 0]$  with evaluations of  $g_{torq}$  of the validation budget
4: while  $k \leq k_{max}$  do
5:   if  $|p_k - 0.95| < \epsilon$  then
6:      $x_{k+1}^* = \arg \min_{x \in \mathcal{X}^{(t)} \setminus \{x_i^*\}_{i=1}^k} \mathbb{E}(Z_x^{(t)})$  such that  $\mathbb{E}(C_x^{c,(t)}) \leq 0$ .
7:   else
8:      $x_{temp}^* \leftarrow x_k^*$ 
9:     while  $x_{temp}^* \in \{x_i^*\}_{i=1}^k$  do
10:      Set  $s$  the index such that  $x_{temp}^* = x_s^*$ 
11:      if  $p_s < 0.95$  then
12:        Decrease  $c$ .
13:      else
14:        Increase  $c$ .
15:      end if
16:       $x_{temp}^* = \arg \min_{x \in \mathcal{X}^{(t)}} \mathbb{E}(Z_x^{(t)})$  such that  $\mathbb{E}(C_x^{c,(t)}) \leq 0$ .
17:    end while
18:     $x_{k+1}^* \leftarrow x_{temp}^*$ 
19:  end if
20:  High fidelity estimate :  $p_{k+1} = \mathbb{P}[420 - g_{torq}(x_{k+1}^*, U) \leq 0]$  with evaluations of  $g_{torq}$  of the validation budget
21:   $k \leftarrow k + 1$ 
22: end while
23: High fidelity estimates :  $\mathbb{E}[f_{ripp}(x, U)]$  for all  $x \in \{x_i^*\}_{i=1}^{k_{max}}$  with evaluations of  $f_{ripp}$  of the validation budget.
```

---

The main choice in the validation algorithm is the criterion to decide whether  $c$  should be increased or decreased. We propose a criterion based on the value of the high fidelity estimate of  $p_k \approx \mathbb{P}[420 - g_{torq}(x_k^*, U) \leq 0]$ :

- If  $p_k \approx 0.95$ , then the next point  $x_{k+1}^*$  is taken as the second best point (according to (5.3)) without changing the constraint process by modifying  $c$ . We justify this choice by assuming that, with active constraints, the (true) optimum  $x^*$  should be at the bound  $\mathbb{P}[420 - g_{torq}(x^*, U) \leq 0] = 0.95$ . Then  $p_k \approx 0.95$  means that the criterion (5.3) that relies on the processes has indeed provided a point near the boundary. The GPs can then be trusted (at least in this domain), and we won't change them.

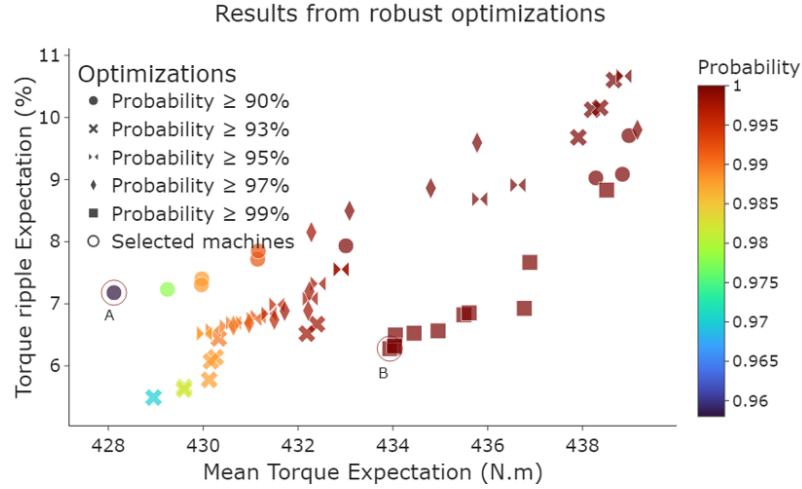


**Figure 5.19:** Best five points for each algorithm obtained after validation with a budget of 100 evaluations of  $f_{ripp}$  and  $g_{torq}$  for each point.

- On the other hand, if  $p_k$  is far from 0.95 (either in non-feasible regions  $p_k \ll 0.95$  or in non-active constraint regions  $p_k = 1$ ), the GPs don't seem to provide relevant points, so we change the constraint threshold  $c$  to compensate for their error:
  - If  $p_k \ll 0.95$ ,  $c$  is decreased. Indeed,  $p_k \ll 0.95$  implies that  $\mathbb{P}_U(G_{(x,U)} \leq c)$  overestimates  $p_k \approx \mathbb{P}_U(420 - g_{torq}(x, U) \leq 0)$ , and should then be reduced which is done by decreasing  $c$ .
  - Similarly,  $p_k \gg 0.95$ ,  $c$  is increased.

We use  $k_{max} = 5$  points with high fidelity estimates done by crude MC with 100 points for a total of 500 evaluations of  $f_{ripp}$  and  $g_{torq}$ . The resulting optima for each algorithm are shown in Figure 5.19 by plotting  $\mathbb{E}[f_{ripp}(x, U)]$  against  $\mathbb{P}[420 - g_{torq}(x, U) \leq 0]$ . Contrary to the results of Figure 5.18, EFISUR does not necessarily lead to the best optima. This means that the GP errors in EFISUR are higher than in other algorithms and can lead to worst points. By reducing the dimension, our algorithms allow to reduce these errors and to obtain better points.

For comparison, the multiobjective problem (5.2), with also the constraint  $\mathbb{P}[420 - g_{torq}(x, U) \leq 0]$ , is solved in [138], using an adapted EFISUR version for multiobjective, after an initial dimension reduction by SA using the results of [137] (which can be compared to our **Off-SA** algorithm). In [138], all design inputs except  $X_8$  are kept, and only  $U_{13}$ ,  $U_{14}$ ,  $U_1$ ,  $U_2$ ,  $U_3$ , and  $U_5$  for the uncertain



**Figure 5.20:** Non-dominated machines proposed by each optimization for  $\alpha \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$ . Color stands for the predicted  $\mathbb{P}[420 - g_{torq}(x, U) \leq 0]$ . Figure taken from [138].

inputs, which is almost the same as our results in Chapter 2 (where SA was done only on the uncertain inputs). Another difference is that the uncertainties are considered under an  $x + U$  form. With 235 iterations (140 for the initial DoE and 19 iterations with 5 added points at each iteration), the authors of [138] obtain the Pareto front shown in Figure 5.20. We observe that their best  $\mathbb{E}[f_{ripp}(x, U)]$  with  $\mathbb{P}[420 - g_{torq}(x, U) \leq 0]$  are between 6% and 7%. In our case (Figure 5.19) we get smaller values of  $\mathbb{E}[f_{ripp}(x, U)]$  with 4.5% and 99% for the best point of **On-2steps-SA-free**. However, we are not trying to maximize  $\mathbb{E}[g_{torq}(x, U)]$  but only to verify the constraint, so our obtained points may have a lower  $\mathbb{E}[g_{torq}(x, U)]$  than the best points of Figure 5.20.

In terms of SA results, we can compare if our algorithms keep the same inputs as those of [138] and of Chapter 2. We don't necessarily expect the same kept inputs as [138] since we don't perform the same optimization. Also, the online methods clearly don't look at the same effect of the inputs as the online methods. We still provide Tables 5.19 and 5.20 showing which inputs are kept for [138] and **Off-SA**, and what percentage each input is kept on all SA steps for the other algorithms.

For the design inputs, only  $X_8$  was removed in [138]. For our algorithm,  $X_8$  is also removed except for the **Off-seq-SA** algorithm. We also almost never keep  $X_{10}$ ,  $X_{11}$ , and  $X_{12}$ , and to a lesser extent  $X_4$  and  $X_6$ . On the other hand,  $X_1$ ,  $X_2$ ,  $X_3$ , and  $X_5$  are always kept regardless of the method, while  $X_7$  and  $X_9$  are only detected with the second version of the online and offline algorithms.

For the uncertain inputs, the results are very different depending on the method. We still recover that for the SA method of [138], and offline algorithms,  $U_{13}$  and  $U_{14}$  are kept. For example, we also keep  $U_8$ ,  $U_{10}$ , and  $U_{12}$ , but this is logical since we measure the effect of  $U$  differently. Interestingly, the online methods show a strong effect of  $U_2$ , which seems to be important to reduce  $S$ , i.e. the variability of the one step ahead feasible improvement.

To conclude this chapter, we can say that the numerical results obtained both on toy functions and on a real test case show that the algorithms proposed in chapter 4 open the use of EFISUR to higher dimension, producing comparable if not sometimes better results for a reduced computational time.

| Method            | $X_1$       | $X_2$       | $X_3$       | $X_4$       | $X_5$       | $X_6$       | $X_7$       | $X_8$       | $X_9$       | $X_{10}$    | $X_{11}$    | $X_{12}$    |
|-------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Multi-obj [138]   | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.00        | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> |
| Off-SA            | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.00        | <b>1.00</b> | 0.00        | 0.00        | 0.00        | 0.00        | 0.00        | 0.00        | 0.00        |
| Off-seq-SA        | <b>1.00</b> | <b>1.00</b> | <b>0.85</b> | <b>0.38</b> | <b>0.69</b> | <b>0.31</b> | <b>0.62</b> | <b>0.62</b> | <b>0.62</b> | 0.08        | 0.08        | 0.00        |
| On-SA             | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.07        | <b>0.92</b> | 0.01        | 0.06        | 0.09        | 0.22        | 0.07        | 0.03        | 0.01        |
| On-2steps-SA      | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.21        | <b>1.00</b> | 0.02        | <b>0.78</b> | 0.15        | <b>1.00</b> | 0.21        | 0.03        | 0.05        |
| On-2steps-SA-free | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.11        | <b>0.99</b> | 0.06        | <b>0.81</b> | 0.05        | <b>0.80</b> | 0.02        | 0.06        | 0.04        |

**Table 5.19:** Percentage of kept design inputs during the SA steps of each algorithm, on the electric machine test case.

| Method            | $U_1$       | $U_2$       | $U_3$       | $U_4$       | $U_5$       | $U_6$       | $U_7$       | $U_8$       | $U_9$ | $U_{10}$    | $U_{11}$ | $U_{12}$    | $U_{13}$    | $U_{14}$    |
|-------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------|-------------|----------|-------------|-------------|-------------|
| Multi-obj [138]   | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.00        | 0.00        | 0.00        | 0.00  | 0.00        | 0.00     | 0.00        | <b>1.00</b> | <b>1.00</b> |
| Off-SA            | <b>1.00</b> | <b>1.00</b> | 0.00        | <b>1.00</b> | 0.00        | 0.00        | 0.00        | <b>1.00</b> | 0.00  | <b>1.00</b> | 0.00     | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> |
| Off-seq-SA        | <b>0.54</b> | 0.15        | <b>0.92</b> | 0.15        | <b>0.77</b> | 0.08        | 0.00        | <b>0.69</b> | 0.08  | <b>0.62</b> | 0.15     | <b>0.54</b> | <b>0.92</b> | <b>1.00</b> |
| On-SA             | 0.12        | <b>0.86</b> | 0.08        | 0.05        | 0.07        | 0.02        | 0.05        | 0.12        | 0.05  | 0.05        | 0.05     | 0.07        | 0.05        | 0.11        |
| On-2steps-SA      | 0.04        | <b>0.99</b> | 0.05        | 0.03        | 0.04        | 0.12        | 0.04        | 0.03        | 0.06  | 0.03        | 0.04     | 0.03        | 0.02        | 0.03        |
| On-2steps-SA-free | 0.10        | <b>0.99</b> | 0.10        | <b>0.48</b> | 0.12        | <b>0.41</b> | <b>0.17</b> | 0.05        | 0.15  | <b>0.22</b> | 0.13     | 0.08        | 0.09        | 0.12        |

**Table 5.20:** Percentage of kept uncertain inputs during the SA steps of each algorithm, on the electric machine test case.

# 6

## Conclusion and perspectives

---

|       |                                   |     |
|-------|-----------------------------------|-----|
| 6.1   | Summary of the main contributions | 127 |
| 6.2   | Summary of the perspectives       | 128 |
| 6.2.1 | Part 1: SA of set-valueds models  | 128 |
| 6.2.2 | Part 2 : SA and EFISUR            | 129 |

---

In this last chapter, we conclude the thesis by first summarizing the main contributions and then detailing perspectives of research.

### 6.1 Summary of the main contributions

The first main contribution, corresponding to the first part of the thesis, is to propose ways to perform SA of set-valued models. Chapter 2 introduces sensitivity indices corresponding to HSIC-ANOVA indices but for set-valued outputs. They rely on the strength of kernels for comparing elements of complex types, which are used here to compare sets with a kernel  $k_{set}$  shown to be characteristic. An estimator adapted to sets is also introduced and its convergence is studied. Since  $H_{set}$  doesn't have any rivals to compare with, we propose in Chapter 3 two other types of indices. The first is inspired by Sobol' indices, but adapted to sets using random set theory. The second is based on universal indices, which are existing indices designed to be flexible enough to be used in any metric output space. With these three indices, a comparison is made both on a toy function and on a real test case of a pollutant map concentration model.

The need to perform a SA of sets originally arose from the question of quantifying the effect of uncertain inputs on a robust optimization problem. This leads to the second main contribution of the thesis, which is to propose adaptations of EFISUR, a robust Bayesian optimization algorithm, to make it usable in higher dimensions. These adaptations, based on SA steps at different points of the algorithm, are first presented in Chapter 4. Two different approaches are introduced. The first one is based on dimension reduction before or between EFISUR iterations. The second one aims at reducing the dimension of the acquisition criteria optimization at each EFISUR iterations. Then,

in Chapter 5, all these methods are tested and compared on toy functions. The toy functions are built so that the dimension can be chosen and are of different complexity. Finally, the algorithms are tested on a real test of the robust design of an electrical machine. These various comparisons show that the algorithm performs well enough and allows important computational time gains, especially as the dimension increases. However, a lot of work remains to perfect the algorithm, and some ideas are given in the perspectives.

## 6.2 Summary of the perspectives

In this section, we use the same plan as the manuscript and we list different potential upgrades or studies for each part of the thesis.

### 6.2.1 Part 1: SA of set-valued models

**HSIC-BASED SA INDICES FOR SET-VALUED MODELS** In the second and third chapters, we propose and test HSIC-ANOVA indices for set-valued indices based on a kernel  $k_{set}$  that is shown to be characteristic. We also introduce a nested Monte Carlo estimator of HSIC-ANOVA, of which we compute the quadratic risk.

- Our proposed nested MC estimator of  $\text{HSIC}(U_i, \Gamma)$  has only a quadratic risk convergence rate of  $\frac{1}{n} + \frac{1}{m}$ . We show in Section B.1.6 of Appendix B that it becomes  $\frac{1}{n} + \frac{1}{m^2}$  in the case of independent m-samples. However, the latter case is not profitable, since it greatly increases the number of model evaluations. The question then is whether we can approach the rate of  $\frac{1}{n} + \frac{1}{m^2}$  without adding model evaluations. One idea is to use a well chosen  $m$  sample (e.g. quasi-MC sequences), which might improve the convergence rate in  $m$  and produce a convergence rate between  $\frac{1}{m}$  and  $\frac{1}{m^2}$ .
- We proposed only one kernel for sets. To have other kernels, one idea is to draw inspiration from kernels in the case of scalar outputs. For example, if we see  $k_{set}$  as a Gaussian kernel, adding a square root would make a Laplace kernel, and equivalents of matérn kernels can be proposed by replacing  $|x - y|^2$  by the volume of the symmetric difference. It would then be interesting to adapt the proofs of 2.1 and 2.3, i.e. that it is indeed a kernel, and that it is characteristic. However, the adapted proofs don't seem straightforward.
- We have only shown that  $k_{set}$  is characteristic, which is not enough to make  $k_{set}$  characteristic in the context of a pair of set-valued outputs. This is a sufficient condition (along with a characteristic input kernel) to have independence if and only if HSIC is zero. It would then be interesting to show whether  $k_{set} \otimes k_{set}$  and even  $\bigotimes_{i=1}^r k_{set}$  are characteristic kernels.

Although we have not attempted the proof, we think that the proof that  $k_{set}$  is characteristic can be generalized for products of  $k_{set}$ , especially since the product of the exponential is the exponential of the sum.

- We have restricted HSIC of sets to the ANOVA case, i.e., with ANOVA input kernels, which greatly simplifies the HSIC expression, leading to only one expectation to be estimated instead of the sum of 3. It has then become much easier to prove Proposition 2.4, i.e. the convergence of the risk of the nested estimator. The corresponding nested estimator of classical HSIC is straightforward, using a nested estimator for each 3 term. It would then be interesting to compute the corresponding quadratic risk and see if its convergence rate remains similar.

**VOROB'EV BASED INDICES** The proposed Vorob'ev index lack good "sensitivity indices" property such as decomposition and are costly to estimate. Detailed in Appendix C, alternative formulations also based on Vorob'ev expectation or median can be used, but don't seem to provide easier estimation nor better properties. One line of research may be to use other random set expectations such as the Selection or Debreu expectations presented in Section A.2.1 of Appendix A. They don't have a direct deviation, but the volume of the symmetric difference can still be used. However, even if this leads to indices with nice properties, their estimation would probably be too costly.

**UNIVERSAL INDICES** Since we have seen that the effect of the test sets is very strong, it would be interesting to try to study this effect in detail, both from a theoretical perspective and through numerical tests on different examples.

## 6.2.2 Part 2 : SA and EFISUR

**OFFLINE SA** Offline approaches rely on dimension reduction steps by SA between series of EFISUR iterations. The SA is performed on the predicted mean of GPs in high dimension to reduce the dimension on which noisy GPs are used for EFISUR iterations.

- The SA is done on the predicted mean of the GPs, either on  $\mu_F$  and  $\mu_G$  for HSIC of sets for the uncertain inputs, and on  $\mu_Z$  and  $\mu_C$  for the GOSA indices for the  $x$ . It would be interesting and appropriate to adapt the SA to the fact that we are using GP, e.g. inspired by the work of [100], where Sobol' indices have been adapted when calculated on a GP in order to integrate GP modeling error in their estimation.
- Even if the choice of a GP modeling for the SA steps seems logical, since it is the same one used in EFISUR, it might be more appropriate to use other surrogates for the SA steps, possibly more suitable for a sensitivity analysis. This would mean having two different surrogates, one for the SA and GPs for EFISUR, which is possible but only for offline methods.
- In Off-seq-SA it would be interesting and crucial to improve the way we deal with noise. We proposed to use a **noisy** GP so that the temporarily removed inputs are not evaluated at fixed values. The noisy GP is fitted with a noise determined by the GOSA, with the intuitive idea that the noise should be low in regions of interest determined by the goal-oriented transformation. From our point of view, it would be interesting to study and propose more general goal-oriented approaches to deal with GP noise. It may also be appropriate to treat the noise differently. In our case, inspired from [5], we could use a sum of two GPs, using a high fidelity one for the selected inputs and a low fidelity one for the removed inputs. It might also be interesting to modify the EFISUR acquisition criteria to adapt them to noise, as in [54], where the expected improvement is penalized by the noise, so that areas of strong noise are treated differently from noise-free areas.
- In **Off-seq-SA**, we simply propose to perform a SA step every  $n_{iter/step}$  iterations. It would be interesting to define a criterion to decide when and if a new SA step should be performed. This criterion could be based on a measure of the convergence of the algorithm in the current small dimension: if the algorithm has converged, it means that it might be necessary to add (or remove) some dimension to explore different areas.

**ONLINE SA** Online approaches reduce the dimension of the acquisition criteria optimizations at each iteration.

- In online algorithms, the SA of the proxy  $S$  has shown that  $S$  is often optimized for nothing, in the sense that no  $U$  affects  $S$ . This happens especially at the end of the algorithm, i.e.,

after several iterations. Therefore, it may be worthwhile to explore alternative optimization criteria to improve EFISUR algorithm.

- We decided to perform a SA on the acquisition criteria, specifically on  $\text{EFI}$  and  $S$ . A sensitivity analysis directly on the optimization functions  $f$  and  $g$  (using the GP means), similar to what is done in offline algorithms, could also be considered. This approach is justified because while we are optimizing  $\text{EFI}$  and  $S$ , the primary objective is still to minimize  $f$  while ensuring that  $g \leq 0$ . Moreover, such a SA would be less costly, since predictions based on GP means are much cheaper compared to computations of  $\text{EFI}$  and  $S$ .
- For **On-2steps-SA**, generalized with **On-seq-SA** with sequentially coupled "SA/optimization in small dimension", it would be appropriate to define a criterion for stopping or continuing with "SA/optimization" steps. This criterion could be based on the SA results, e.g. deciding to stop if no input influences the output sufficiently.

**OFFLINE AND ONLINE SA** The two presented approaches can easily be merged. In a context of very high dimension, it may be appropriate to reach a reasonable dimension with offline approaches and to keep reducing the dimension before each acquisition function optimization.

**GENERALIZATION OF THE APPROACHES** Both the offline and online algorithms presented are designed for the EFISUR algorithm. However, their general idea can be adapted to any robust Bayesian optimization algorithm. It would be important to clearly distinguish what is specific to EFISUR and what is not, in order to write these four algorithms (possibly reunited in one or two algorithms) in a context of generic BO, especially with arbitrary acquisition criteria.

**VALIDATION** The suggested validation step is preliminary work. It can be modified with :

- Changing the criterion for selecting points.
- Changing the criterion to decide to adjust the process  $C$  by modifying the threshold  $c$ . It would be interesting to construct a criterion that directly uses the error at the point of interest to decide how to adjust the GPs. The criterion should also use the objective process  $Z$ .
- During the validation algorithm, the evaluations  $f$  and  $g$  used for high fidelity estimates should be used to increase the accuracy of the current GPs.

# A

## Appendix of chapter 1: Some elements of random set theory

### A.1 A bit of random set theory

A goal of the thesis is to propose ways to perform a SA of excursion sets  $\Gamma$  defined by

$$\Gamma = \{x \in \mathcal{X}, f(x, U) \leq q, g(x, U) \leq 0\}, \quad (\text{A.1})$$

to quantify the effects of the uncertain inputs. This implies the use of a random set theory. Some elements of the random set theory of [105] are presented in the following.

#### A.1.1 Random set definition

Let  $\mathcal{X}$  be a locally compact Hausdorff second countable topological space (think of any compact subset of  $\mathbb{R}^d$ , like  $[0, 1]^d$ ). Since the power set  $\mathcal{P}(\mathcal{X})$ , i.e., the space of all subsets of  $\mathcal{X}$ , is too large, the notion of random sets is often defined for certain subsets of  $\mathcal{P}(\mathcal{X})$ . The most common choice is to consider only random closed sets, defined as random elements in the space of closed subsets  $\mathcal{F}(\mathcal{X})$ . A random closed set is then defined as a  $\mathcal{F}(\mathcal{X})$ -valued random variable, i.e. a measurable map from  $(\Omega, \mathcal{F})$  to  $\mathcal{F}(\mathcal{X})$  with a  $\sigma$ -algebra to choose. Instead of using the classical definition of measurability, [105] proposes a more practical definition of random sets:

**Definition A.1** (Definition of a random closed set). *A map  $\Gamma : \Omega \mapsto \mathcal{F}(\mathcal{X})$  is called a random closed set if,*

$$\forall K \in \mathcal{K}(\mathcal{X}) \quad \{\omega \in \Omega, \Gamma(\omega) \cap K \neq \emptyset\} \in \mathcal{F}.$$

where  $\mathcal{K}(\mathcal{X})$  denotes the compact subsets of  $\mathcal{X}$ .

A random set  $\Gamma$  is thus defined in such a way that one can always say whether a  $\Gamma$  realization hits any compacts or not. This definition does correspond to a classical definition of measurability,

when endowing  $\mathcal{F}(\mathcal{X})$  with the  $\sigma$ -algebra  $\mathcal{B}(\mathcal{F}(\mathcal{X}))$  generated by  $\{F \in \mathcal{F}(\mathcal{X}), F \cap K \neq \emptyset\}$  for  $K \in \mathcal{K}(\mathcal{X})$ , called Effros'  $\sigma$ -algebra.

**Example A.1.** For  $\mathcal{X} \subset \mathbb{R}^d$ , the closed ball (for the  $\|\cdot\|_2$  norm)  $B(0, R)$  of random radius a random variable  $R$  is a random closed set.

*Proof.* The proof uses the continuity of  $\|\cdot\|_2$  and the existence of a dense countable subset of any compact  $K \subset \mathbb{R}^d$  (as it is a compact subset of a separable space).

Let  $K \in \mathcal{K}(\mathcal{X})$ . Let us show that  ${}^c\{\omega \in \Omega, B(0, R)(\omega) \cap K \neq \emptyset\} \in \mathcal{F}$  where  ${}^cA$  denotes the complement of  $A$ .

$$\begin{aligned} {}^c\{\omega \in \Omega, B(0, R)(\omega) \cap K \neq \emptyset\} &= \{\omega \in \Omega, \forall x \in K, \|x\|_2 \geq R(\omega)\} \\ &= \bigcap_{x \in K} R^{-1}([0, \|x\|_2]) \end{aligned}$$

Let then  $K_{dense}$  be a dense countable subset of  $K$ . Let us show that

$$\bigcap_{x \in K} R^{-1}([0, \|x\|_2]) = \bigcap_{x \in K_{dense}} R^{-1}([0, \|x\|_2]).$$

- $\subset$  : True as  $K_{dense} \subset K$ .
- $\supset$  : Let  $\omega \in \bigcap_{x \in K_{dense}} R^{-1}([0, \|x\|_2])$ , i.e.,  $\forall x \in K_{dense} \|x\|_2 \geq R(\omega)$ . Let  $x \in K$  and  $x_n \rightarrow x$  with  $x_n \in K_{dense}$  i.e.,  $\|x_n\| \geq R(\omega)$ . By continuity of the norm, we obtain  $\|x\| \geq R(\omega)$ . Finally,  $\forall x \in K, \|x\| \geq R(\omega)$ , i.e.,  $\omega \in \bigcap_{x \in K} R^{-1}([0, \|x\|_2])$ .

We can then conclude as  $R^{-1}([0, \|x\|_2]) \in \mathcal{F}$  so complement and the countable union remains in  $\mathcal{F}$ .  $\square$

**Example A.2.** Under the assumption that  $g(\cdot, u)$  is continuous on  $\mathcal{X}$  for all  $u \in \mathcal{U}$  and  $g(x, \cdot)$  is measurable on  $\mathcal{U}$  for all  $x \in \mathcal{X}$ , the excursion sets of (A.1) are random closed sets.

*Proof.* The proof is similar than the previous proof.  $\square$

Given this definition, classical notions associated with random variables, such as the probability distribution function and the expectation, can be derived. For example, the capacity functional is the counterpart of the pdf, and the selection expectation corresponds to a classical notion of expectation. However, they do not provide good numerical approximations and are not easy to interpret. For this reason, in this thesis we focus on another expectation, the Vorob'ev expectation, which is the most widely used in the literature.

### A.1.2 Vorob'ev expectation and deviation

Given a random set  $\Gamma$ , the Vorob'ev expectation relies on a coverage function that measures the probability of each  $x \in \mathcal{X}$  to be in  $\Gamma$ :

**Definition A.2** (Coverage function). The coverage function  $p_\Gamma$  of a random set  $\Gamma$  is defined for all  $x \in \mathcal{X}$  by

$$p_\Gamma(x) = \mathbb{P}(x \in \Gamma).$$

From  $p_\Gamma$ , quantiles can be defined:

**Definition A.3** (Vorob'ev quantiles). For  $\alpha \in [0, 1]$ , the  $\alpha$ -th Vorob'ev quantile  $Q_\alpha$  is given by:

$$Q_\alpha = \{x \in \mathcal{X}, \quad p_\Gamma(x) \geq \alpha\}.$$

$Q_{0.5}$  is called the Vorob'ev median.

The Vorob'ev expectation is then defined as the Vorob'ev quantile whose volume is closest to the mean volume of  $\Gamma$ . Before introducing it, we need to justify that the mean volume of  $\Gamma$  is defined. This is based on Robbins' theorem proved in [105].

**Theorem A.1** (Robbins' theorem ([105])). Let  $\Gamma$  be a random closed set in a Polish space  $\mathcal{X}$ . If  $\lambda$  is a locally finite measure on  $\mathcal{B}(\mathcal{X})$ , then  $\lambda(\Gamma)$  is a random variable and:

$$\mathbb{E}(\lambda(\Gamma)) = \int_{\mathcal{X}} p_\Gamma(x) \lambda(dx).$$

Given  $\lambda$  s.t.  $\mathbb{E}(\lambda(\Gamma))$  is defined (e.g. take Lebesgue measure if  $\mathcal{X} \subset \mathbb{R}^d$ ), Vorob'ev expectation can be introduced.

**Definition A.4** (Vorob'ev expectation). Assuming that  $\mathbb{E}(\lambda(\Gamma)) < \infty$ , the Vorob'ev expectation  $\mathbb{E}^V(\Gamma)$  of a random set  $\Gamma$  is defined by:

$$\mathbb{E}^V(\Gamma) = \{x \in \mathcal{X}, \quad p_\Gamma(x) \geq \alpha^*\} = Q_{\alpha^*},$$

where  $\alpha^*$  determined by

$$\mathbb{E}(\lambda(\Gamma)) = \lambda(\mathbb{E}^V(\Gamma)), \tag{A.2}$$

or by

$$\forall \alpha > \alpha^* \quad \lambda(Q_\alpha) \leq \mathbb{E}(\lambda(\Gamma)) \leq \lambda(\mathbb{E}^V(\Gamma)).$$

if (A.2) has no solution.

**Example A.3.** Let  $\Gamma = B(0, R)$  a random closed centered ball in  $\mathcal{X} = \mathbb{R}^d$  (for the  $\|\cdot\|_2$  norm) with a random radius  $R$  being an integrable random variable. We have

$$\mathbb{E}^V(\Gamma) = B(0, \mathbb{E}(R^d)^{\frac{1}{d}}).$$

*Proof.* • **d = 2** : The mean volume of  $\Gamma$  is  $\mathbb{E}(\lambda(\Gamma)) = \pi \mathbb{E}(R^2)$ . The coverage function writes

$$\mathbb{P}(x \in \Gamma) = \mathbb{P}(\|x\|_2 \leq R) = 1 - F_R(\|x\|_2),$$

where  $F_R$  is the cdf of  $R$ . The volume of a Vorob'ev quantile of order  $\alpha$  is:

$$\begin{aligned} \lambda(Q_\alpha) &= \lambda(\{x \in \mathcal{X}, \quad 1 - F_R(\|x\|_2) \geq \alpha\}) \\ &= \lambda(\{x \in \mathcal{X}, \quad \|x\|_2 \leq F_R^{-1}(1 - \alpha)\}) \\ &= \lambda(B(0, F_R^{-1}(1 - \alpha))) \\ &= \pi F_R^{-1}(1 - \alpha)^2. \end{aligned}$$

Then,  $\lambda(Q_{\alpha^*}) = \mathbb{E}(\lambda(\Gamma))$  gives  $\alpha^* = 1 - F_R(\sqrt{\mathbb{E}(R^2)})$  and finally:

$$\mathbb{E}^V(\Gamma) = Q_{\alpha^*} = B(0, \mathbb{E}(R^2)^{\frac{1}{2}}).$$

- $d \geq 2$  : In this case the only difference is the volume of the hypersphere, whose expression is known and gives a mean volume of  $\mathbb{E}(\lambda(\Gamma)) = \frac{\pi^{d/2} \mathbb{E}(R^d)}{\Gamma_{fct}(\frac{d}{2} + 1)}$  where  $\Gamma_{fct}$  is the Gamma function.

We still have :

$$\lambda(Q_\alpha) = \lambda(B(0, F_R^{-1}(1 - \alpha))) = \frac{\pi^{d/2} (F_R^{-1}(1 - \alpha))^d}{\Gamma_{fct}(\frac{d}{2} + 1)},$$

which gives the result  $\mathbb{E}^V(\Gamma) = B(0, \mathbb{E}(R^d)^{\frac{1}{d}})$ .

□

To measure a variability of the random set, comparisons between sets are necessary. One way to compare sets is to use the symmetric difference.

**Definition A.5** (Symmetric difference). *For  $A, B \subset \mathcal{X}$ , the symmetric difference is defined as*

$$A \Delta B = A \cup B - A \cap B.$$

Then the volume of the symmetric difference  $\lambda(A \Delta B)$  is a pseudo-distance and can be used to compare the sets  $A$  and  $B$ . From this, the Vorob'ev expectation minimizes a deviation, which defines the Vorob'ev deviation [105].

**Proposition A.1** (Vorob'ev deviation). *For all borel set  $M$  s.t.  $\lambda(M) = \mathbb{E}(\lambda(\Gamma))$ ,*

$$\mathbb{E}[\lambda(\Gamma \Delta \mathbb{E}^V(\Gamma))] \leq \mathbb{E}[\lambda(\Gamma \Delta M)].$$

*The Vorob'ev deviation  $Var^V(\Gamma)$  is then defined as:*

$$Var^V(\Gamma) = \mathbb{E}[\lambda(\Gamma \Delta \mathbb{E}^V(\Gamma))].$$

**Example A.4.** *For  $\Gamma = B(0, R) \subset \mathbb{R}^d$ ,*

$$Var^V(\Gamma) = \frac{\pi^{d/2}}{\Gamma_{fct}(\frac{d}{2} + 1)} \mathbb{E}[|R^d - \mathbb{E}(R^d)|].$$

By removing the assumption  $\lambda(M) = \mathbb{E}(\lambda(\Gamma))$ , the minimizer becomes the Vorob'ev median:

**Proposition A.2** (Vorob'ev median deviation). *For all borel set  $M$  s.t.  $\lambda(M) < \infty$ ,*

$$\mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})] \leq \mathbb{E}[\lambda(\Gamma \Delta M)].$$

*The Vorob'ev median deviation  $VMD(\Gamma)$  is then defined as:*

$$VMD(\Gamma) = \mathbb{E}[\lambda(\Gamma \Delta Q_{0.5})].$$

Estimation of Vorob'ev expectations and deviations have been studied in [106], [60] and by [20], [7] for excursion sets.

In the thesis (Chapter 3), one way to perform a SA on sets is to measure inputs impact on these deviations.

#### Take-home message on random sets

- Random closed sets are defined as random variables in a space of closed sets.
- Classical expectation of random sets is not practical to use. A commonly used alternative is the Vorob'ev expectation.
- Through symmetric difference, deviations allow to measure the variability of a random set.

## A.2 A bit more of random set theory

This part has no direct interest for the rest of the thesis, but provides a justification of the tools used on random sets. In particular, we present alternative definitions of random set expectations, and we justify why we chose to use the Vorob'ev expectation. All the following definitions are taken from [105].

### A.2.1 Alternative expectations of random closed sets

The most general definition of the expectation of a random variable  $X \in \mathbb{R}$  is given by  $\mathbb{E}[X] = \int_{\Omega} X(\omega) d\mathbb{P}(\omega)$  if  $\int_{\Omega} |X(\omega)| d\mathbb{P}(\omega) < +\infty$ . It is not straightforward to use this definition when  $X$  is a random set.

For simple random variables (i.e., discrete, but with a finite number of values), the expectation is simply the weighted sum  $\mathbb{E}[X] = \sum_i x_i \mathbb{P}(X = x_i)$ . The random set counterparts of simple random variables are called simple random sets, defined as random sets that take a finite number of values. Applying the previous expectation implies to use a sum between sets. The Minkowski sum is the most common sum between two sets.

**Definition A.6** (Minkowski sum and scalar product). *Given two sets  $A, B \subset \mathcal{X}$ , their Minkowski sum  $A + B$  is defined by:*

$$A + B = \{a + b, a \in A, b \in B\}.$$

For  $\lambda \in \mathbb{R}$ ,  $\lambda A$  is defined by

$$\lambda A = \{\lambda a, a \in A\} \text{ for } \lambda \in \mathbb{R}.$$

With this sum, the expectation of a simple random closed set  $\Gamma$  of values  $(\gamma_i)_i$  can be defined as

$$\mathbb{E}[\Gamma] = \sum_i \gamma_i \mathbb{P}(\Gamma = \gamma_i).$$

**Example A.5.** Let  $\Gamma = B(0, R)$ , where  $\mathbb{P}(R = r_1) = p$  and  $\mathbb{P}(R = r_2) = 1 - p$  with  $r_1, r_2 \geq 0$ . Then, we have

$$\mathbb{E}[\Gamma] = B(0, pr_1 + (1 - p)r_2) = B(0, \mathbb{E}[R]).$$

This is easily shown by relying on the result that the Minkowski sum of two centered balls of radius  $r_1$  and  $r_2$  is the centered ball of radius  $r_1 + r_2$ . This is shown using the triangle inequality for one inclusion and writing  $x = \frac{r_1}{r_1 + r_2}x + \frac{r_2}{r_1 + r_2}x$  for the other inclusion.

This expectation can be extended to random sets that are the limit of simple sets, with the Debreu expectation.

#### A.2.1.a Debreu Expectation

Before defining the Debreu expectation, we need to define the limit of sets, which requires a metric definition. Since the previously introduced volume of the symmetric difference is not a metric (but only a pseudo-metric), the Hausdorff metric is preferred.

**Definition A.7** (Hausdorff metric). *The Hausdorff metric (or distance) between two subsets  $\gamma_1, \gamma_2$  of a metric space  $(\mathcal{X}, \delta)$  is defined as:*

$$d_H(\gamma_1, \gamma_2) := \max \left\{ \sup_{y \in \gamma_2} \delta(\gamma_1, y), \sup_{x \in \gamma_1} \delta(x, \gamma_2) \right\} = \max \left\{ \sup_{y \in \gamma_2} \inf_{x \in \gamma_1} \delta(x, y), \sup_{x \in \gamma_1} \inf_{y \in \gamma_2} \delta(x, y) \right\}.$$

Then we say that  $\Gamma = \lim_{n \rightarrow +\infty} \gamma_n$  iff  $d_H(\gamma_n, \gamma) \xrightarrow{n \rightarrow +\infty} 0$  and a random set is called *Hausdorff approximable* if it is an almost sure limit for the Hausdorff metric of simple random sets.

**Definition A.8** (Debreu expectation). *For a Hausdorff approximable random convex bounded closed set  $\Gamma = \lim_{n \rightarrow +\infty} \Gamma_n$  where  $(\Gamma_n)$  is a sequence of simple random set of values  $F_1^n, \dots, F_{k(n)}^n$  with probabilities  $p_1^n, \dots, p_{k(n)}^n$ , then the Debreu expectation of  $\Gamma$  is :*

$$\mathbb{E}^D(\Gamma) = \lim_{n \rightarrow +\infty} \overline{p_1^n F_1^n + \dots + p_{k(n)}^n F_{k(n)}^n}. \quad (\text{A.3})$$

**Example A.6.** For  $\Gamma = B(0, R)$ , where  $R$  is a positive integrable continuous random variable, we have

$$\mathbb{E}^D(\Gamma) = B(0, \mathbb{E}(R)).$$

*Proof.* Let  $R_n$  be a sequence of simple random variables with values  $r_1^n, \dots, r_{k(n)}^n$  and probabilities  $p_1^n, \dots, p_{k(n)}^n$  such that  $R_n \xrightarrow{\text{a.s.}} R$  (which exists, for example take  $r_k^n = \frac{k}{2^n}$  if  $\frac{k}{2^n} \leq R < \frac{k+1}{2^n}$  for  $k = 0, 1, 2, \dots, 2^n - 1$ , with  $p_k^n = \frac{1}{2^n}$ ). We have  $B(0, R_n) \xrightarrow{\text{a.s.}} B(0, R)$  for the Hausdorff metric as  $d_H(B(0, R), B(0, R_n)) = |R - R_n|$ . So by denoting  $\Gamma_n = B(0, R_n)$  the simple random set of values  $B(0, r_1^n), \dots, B(0, r_{k(n)}^n)$  with probabilities  $p_1^n, \dots, p_{k(n)}^n$  we can compute the

Debreu expectation of  $\Gamma$  which is  $\mathbb{E}^D(\Gamma) = \lim_{n \rightarrow +\infty} \overline{\sum_{j=1}^{k(n)} p_j^n B(0, r_j^n)}$ . First we compute  $\sum_{j=1}^{k(n)} p_j^n B(0, r_j^n)$ :

$$\begin{aligned} \sum_{j=1}^{k(n)} p_j^n B(0, r_j^n) &= \sum_{j=1}^{k(n)} B(0, p_j^n r_j^n) \\ &= B(0, \sum_{j=1}^{k(n)} p_j^n r_j^n) \\ &= B(0, \mathbb{E}(R_n)) \end{aligned}$$

The closure doesn't change the set as it's already a closed ball. Finally  $B(0, \mathbb{E}(R_n)) \rightarrow B(0, \mathbb{E}(R))$  for the Hausdorff metric as  $\mathbb{E}(R_n) \rightarrow \mathbb{E}(R)$ .  $\square$

Definition A.8 implies an approximation of Debreu expectation by a computation of the minkowski sum of Equation (A.3) for a high  $n$ . In terms of estimations, if the space  $\mathcal{X}$  is discretized in  $\{x^1, \dots, x^N\}$ , it implies the computation of  $k(n)^N$  classical sums, which is a lot.

The previous expectation requires that the random set be convex and Hausdorff approximable. An alternative expectation that doesn't require these assumptions is called the selection expectation (and coincides with the Debreu expectation for Hausdorff-approximable and a.s. weakly compact convex random sets [105]).

### A.2.1.b Selection expectation

The selection expectation is much easier to define, but more complex to interpret. It requires the definition of selections first.

**Definition A.9** (Selection). *A random variable  $X \in \mathcal{X}$  is called a selection of  $\Gamma$  if  $X(\omega) \in \Gamma(\omega)$  for almost all  $\omega \in \Omega$ . The family of all selections is denoted  $S(\Gamma)$ .*

**Definition A.10** (Selection expectation). *The selection expectation  $\mathbb{E}^S[\Gamma]$  of  $\Gamma$  is defined as*

$$\mathbb{E}^S[\Gamma] = \overline{\{\mathbb{E}(X), X \in S(\Gamma) \cap L^1\}},$$

*i.e., as the closure of the set of all expectations of integrable selections.*

**Example A.7.** *For  $\Gamma = B(0, R)$  where  $R$  is a positive integrable random variable, we have*

$$\mathbb{E}^S(\Gamma) = B(0, \mathbb{E}(R)).$$

*Proof.*  $X$  is a selection of  $\Gamma$  iff

$$\|X\| \leq R \text{ almost surely.}$$

We have then

$$\forall X \in S(\Gamma) \quad \|\mathbb{E}(X)\| \leq \mathbb{E}(\|X\|) \leq \mathbb{E}(R),$$

which means that  $\{\mathbb{E}(X), X \in S(\Gamma) \cap L^1\} \subset B(0, \mathbb{E}(R))$ . Finally as  $B(0, \mathbb{E}(R))$  is closed, we have  $\mathbb{E}^S(\Gamma) \subset B(0, \mathbb{E}(R))$ .

Reciprocally for any  $x \in B(0, \mathbb{E}(R))$ ,  $X = R \frac{x}{\mathbb{E}(R)}$  is a selection of  $\Gamma$  as  $\|X\| = R \frac{\|x\|}{\mathbb{E}(R)} \leq R$  and we have  $\mathbb{E}(X) = x$  so  $x \in \{\mathbb{E}(X), X \in S(\Gamma)\} \subset \mathbb{E}^S(\Gamma)$ .  $\square$

This definition is simple and useful for theoretical proofs. It also verifies a strong law of large numbers (see [105]), which more or less says that for high  $N$

$$\mathbb{E}^S(\Gamma) \approx \frac{1}{N} \sum_{i=1}^N \Gamma^i$$

where  $(\Gamma^1, \dots, \Gamma^N)$  is an i.i.d. sample of  $\Gamma$ . This might seem easy to estimate if we forget that the sum is a Minkowski sum. For example, if  $\mathcal{X}$  is discretized by  $\{x^1, \dots, x^n\}$ , then the Minkowski sum requires  $n^N$  classical sums to compute, which is a lot.

In this part we have presented two alternatives of random set expectations, but they both lack a reasonable estimation cost. Moreover, they don't present classical deviations, i.e., quantities that are minimized at the expectation and which then allow to measure the random set variability. For these reasons, we only consider the Vorob'ev expectation in Chapter 3.



# B

## Appendix of chapter 2: Proofs and additional plots

### B.1 Proofs

#### B.1.1 Proof of Lemma 2.1

**Lemma 2.1.** *Let  $\gamma_1, \gamma_2 \in \mathcal{L}(\mathcal{X})$ , we have*

$$\delta(\gamma_1, \gamma_2) = \|\mathbb{1}_{\gamma_1} - \mathbb{1}_{\gamma_2}\|_{L^2(\mathcal{X})}^2,$$

*and*

$$\delta(\gamma_1, \gamma_2) = 0 \Leftrightarrow \gamma_1 = \gamma_2 \text{ } \lambda\text{-almost everywhere.}$$

*Proof.*

$$\begin{aligned} \delta(\gamma_1, \gamma_2) &= \lambda(\gamma_1 \Delta \gamma_2) \\ &= \int_{\mathcal{X}} \mathbb{1}_{\gamma_1 \Delta \gamma_2} d\lambda \\ &= \int_{\mathcal{X}} (\mathbb{1}_{\gamma_1 \cup \gamma_2} - \mathbb{1}_{\gamma_1 \cap \gamma_2}) d\lambda \\ &= \int_{\mathcal{X}} (\mathbb{1}_{\gamma_1} + \mathbb{1}_{\gamma_2} - 2\mathbb{1}_{\gamma_1 \cap \gamma_2}) d\lambda \\ &= \int_{\mathcal{X}} (\mathbb{1}_{\gamma_1} - \mathbb{1}_{\gamma_2})^2 d\lambda \\ &= \|\mathbb{1}_{\gamma_1} - \mathbb{1}_{\gamma_2}\|_{L^2(\mathcal{X})}^2 \end{aligned}$$

$$\begin{aligned} \delta(\gamma_1, \gamma_2) = 0 &\Leftrightarrow \|\mathbb{1}_{\gamma_1} - \mathbb{1}_{\gamma_2}\|_{L^2(\mathcal{X})}^2 = 0 \\ &\Leftrightarrow \mathbb{1}_{\gamma_1}(x) = \mathbb{1}_{\gamma_2}(x) \text{ for } \lambda\text{-almost every } x \in \mathcal{X} \\ &\Leftrightarrow \gamma_1 = \gamma_2 \text{ } \lambda\text{-almost everywhere.} \end{aligned}$$

□

### B.1.2 Proof of Proposition 2.1

**Proposition 2.1.** *The function  $k_{set} : \mathcal{L}^*(\mathcal{X}) \times \mathcal{L}^*(\mathcal{X}) \rightarrow \mathbb{R}$  defined by*

$$\forall \gamma_1, \gamma_2 \in \mathcal{L}^*(\mathcal{X}), \quad k_{set}(\gamma_1, \gamma_2) = e^{-\frac{\lambda(\gamma_1 \Delta \gamma_2)}{2\sigma^2}},$$

*is symmetric and positive definite for any positive scalar  $\sigma$ , which means that  $k_{set}$  is a kernel.*

*Proof.* The proof is similar as the one of Lemma 2.1. in [8] but is recalled and adapted here.

Let  $\alpha_1, \dots, \alpha_n \in \mathbb{R}$  and  $\gamma_1, \dots, \gamma_n \in \mathcal{L}^*(\mathcal{X})$ . Let's show that

$$\sum_{kl} e^{-\frac{\lambda(\gamma_k \Delta \gamma_l)}{2\sigma^2}} \alpha_k \alpha_l \geq 0.$$

First let's use Lemma 2.1 to write:

$$\sum_{kl} e^{-\frac{\lambda(\gamma_k \Delta \gamma_l)}{2\sigma^2}} \alpha_k \alpha_l = \sum_{kl} e^{-\frac{\|\mathbb{1}_{\gamma_k} - \mathbb{1}_{\gamma_l}\|_{L^2(\mathcal{X})}^2}{2\sigma^2}} \alpha_k \alpha_l.$$

Then we use the Bochner-Milos theorem (Theorem 1.1 of [64]), which says that there exists a random variable  $G \in L^2(\mathcal{X})$  such that

$$\forall f \in L^2(\mathcal{X}), \quad \mathbb{E}(e^{i\langle f, G \rangle_{L^2(\mathcal{X})}}) = e^{-\frac{1}{2}\|f\|_{L^2(\mathcal{X})}^2}.$$

This allows to derive :

$$\begin{aligned} \sum_{kl} e^{-\frac{\lambda(\gamma_k \Delta \gamma_l)}{2\sigma^2}} \alpha_k \alpha_l &= \sum_{kl} e^{-\frac{\|\mathbb{1}_{\gamma_k} - \mathbb{1}_{\gamma_l}\|_{L^2(\mathcal{X})}^2}{2\sigma^2}} \alpha_k \alpha_l \\ &= \sum_{kl} \mathbb{E} e^{i\langle \frac{\mathbb{1}_{\gamma_k} - \mathbb{1}_{\gamma_l}}{\sqrt{2}\sigma}, G \rangle_{L^2(\mathcal{X})}} \alpha_k \alpha_l \\ &= \mathbb{E} \sum_{kl} e^{\frac{i}{\sqrt{2}\sigma} \langle \mathbb{1}_{\gamma_k}, G \rangle_{L^2(\mathcal{X})}} \alpha_k e^{-\frac{i}{\sqrt{2}\sigma} \langle \mathbb{1}_{\gamma_l}, G \rangle_{L^2(\mathcal{X})}} \alpha_l \\ &= \mathbb{E} \left| \sum_k e^{\frac{i}{\sqrt{2}\sigma} \langle \mathbb{1}_{\gamma_k}, G \rangle_{L^2(\mathcal{X})}} \alpha_k \right|^2 \geq 0. \end{aligned}$$

□

### B.1.3 Proof of Proposition 2.2

**Proposition 2.2.** *The kernel  $k_{set}$  is bounded and measurable from  $\mathcal{L}_2^*(\mathcal{X}) = \mathcal{L}^*(\mathcal{X}) \times \mathcal{L}^*(\mathcal{X})$  to  $\mathbb{R}$ .*

*Proof.*  $k_{set}$  is clearly bounded by 1. In term of measurability, we are studying the measurability of  $k_{set}$  with respect to the two Borels  $\sigma$ -algebras  $\mathcal{B}(\mathcal{L}_2^*(\mathcal{X}), \delta_2)$  and  $\mathcal{B}(\mathbb{R}, |\cdot|)$  with  $\delta_2((\gamma_1^A, \gamma_1^A), (\gamma_2^B, \gamma_2^B)) = \delta(\gamma_1^A, \gamma_1^B) + \delta(\gamma_2^A, \gamma_2^B)$ . As  $\delta : \mathcal{L}_2^*(\mathcal{X}) \rightarrow \mathbb{R}^+$  is a distance on  $\mathcal{L}^*(\mathcal{X})$ , it is a continuous function from  $\mathcal{L}_2^*(\mathcal{X})$  to  $\mathbb{R}$ .  $k_{set}$  is then continuous as exp is continuous. Finally the continuity of  $k_{set}$  implies measurability as we are working in two Borel  $\sigma$ -algebras. □

### B.1.4 Proof of Proposition 2.3

**Proposition 2.3.** *The kernel  $k_{\text{set}}$  is characteristic.*

*Proof.* The proof is based on the Proposition 5.2. of [181] which is recalled here with our notations.

**Proposition B.1.** *Let  $\mathcal{P}$  be a Polish space,  $H$  a separable Hilbert space,  $T$  a measurable and injective mapping from  $\mathcal{P}$  to  $H$ , and  $\varphi$  the Laplace transform of a finite Borel measure  $\nu$  on  $[0, +\infty)$  such that  $\nu \neq 0$  and  $\text{supp}(\nu) \neq \{0\}$ . Then, the kernel  $k$  on  $\mathcal{P} \times \mathcal{P}$  defined by*

$$k(x, x') := \varphi \left( \|T(x) - T(x')\|_H^2 \right), \quad (x, x') \in \mathcal{X}^2$$

*is integrally strictly positive definite with respect to  $\mathcal{M}(\mathcal{X})$ .*

Using the previous Proposition, we will show that  $k_{\text{set}}$  is integrally strictly positive which is a sufficient condition to be characteristic (see Theorem 7 in [160]).

To apply the Proposition, we first introduce the notation,

$$\mathcal{F}_{\text{bin}}^* = \{f \in L^2(\mathcal{X}) : \exists \gamma \in \mathcal{L}(\mathcal{X}) \text{ such that } f = \mathbb{1}_\gamma \text{ } \lambda\text{-almost everywhere}\}.$$

We want to apply the proposition with  $\mathcal{P} = \mathcal{L}^*(\mathcal{X})$ ,  $H = L^2(\mathcal{X})$ ,  $\varphi(\cdot) = \exp(-\frac{\cdot}{2\sigma^2})$  and  $T$  defined by

$$\begin{aligned} T : \mathcal{L}^*(\mathcal{X}) &\longrightarrow \mathcal{F}_{\text{bin}}^* \subset H \\ [\gamma]_\delta &\longmapsto [\mathbb{1}_\gamma]_2, \end{aligned}$$

where  $[\cdot]_\delta$  and  $[\cdot]_2$  denote equivalence classes in  $\mathcal{L}(\mathcal{X})$  and  $\mathcal{F}_{\text{bin}}$  respectively. We need to show that  $\mathcal{L}^*(\mathcal{X})$  is Polish, that  $L^2(\mathcal{X})$  is a separable Hilbert space, and that  $T$  is measurable and injective.

- Let's first show that is  $T$  a well-defined measurable and injective mapping from  $\mathcal{P}$  to  $H$ :
  - $T$  is well-defined and in the same time injective as for any  $\gamma_1, \gamma_2 \in \mathcal{L}(\mathcal{X})$ ,

$$\begin{aligned} [\gamma_1]_\delta = [\gamma_2]_\delta &\Leftrightarrow \delta(\gamma_1, \gamma_2) = 0 \\ &\Leftrightarrow \lambda(\gamma_1 \Delta \gamma_2) = 0 \\ &\Leftrightarrow \|\mathbb{1}_{\gamma_1} - \mathbb{1}_{\gamma_2}\|_{L^2(\mathcal{X})} = 0 \\ &\Leftrightarrow [\mathbb{1}_{\gamma_1}]_2 = [\mathbb{1}_{\gamma_2}]_2. \end{aligned}$$

- $T$  is measurable (with respect to the Borel  $\sigma$ -algebras  $\mathcal{B}(\mathcal{L}^*(\mathcal{X}), \delta)$  and  $\mathcal{B}(\mathcal{F}_{\text{bin}}^*, \|\cdot\|_{L^2(\mathcal{X})})$ ) because it is continuous as it is an isometry from  $(\mathcal{L}^*(\mathcal{X}), \sqrt{\delta})$  to  $(\mathcal{F}_{\text{bin}}^*, \|\cdot\|_{L^2(\mathcal{X})})$ .
- Let us show that  $T$  is also surjective which will be useful later on. Let  $[f]_2 \in \mathcal{F}_{\text{bin}}^*$ . There exist  $\gamma \in \mathcal{L}(\mathcal{X})$  such that  $f = \mathbb{1}_\gamma$   $\lambda$ -almost everywhere which implies that  $\|f - \mathbb{1}_\gamma\|_{L^2(\mathcal{X})} = 0$  i.e.  $[f]_2 = [\mathbb{1}_\gamma]_2 = T([\gamma]_\delta)$ .
- $L^2(\mathcal{X})$  is a separable Hilbert space as  $\mathcal{X}$  is compact so separable.
- Let us now show that  $\mathcal{L}^*(\mathcal{X})$  is Polish i.e. a topological space homeomorphic to a separable complete metric space. Luckily, as  $T$  is a surjective isometry, it is an homeomorphism from  $\mathcal{L}^*(\mathcal{X})$  to  $\mathcal{F}_{\text{bin}}^*$  which is a metric space. Then, it only remains to prove that  $\mathcal{F}_{\text{bin}}^*$  is both complete and separable.
  - $\mathcal{F}_{\text{bin}}^* \subset L^2(\mathcal{X})$  which is separable so  $\mathcal{F}_{\text{bin}}^*$  is separable.

- As  $L^2(\mathcal{X})$  is complete, having  $\mathcal{F}_{\text{bin}}^*$  closed is a sufficient condition for  $\mathcal{F}_{\text{bin}}^*$  to be complete. Let us show then that it is closed. Let  $\mathbb{1}_{\gamma_n} \xrightarrow[n \rightarrow +\infty]{L_2} f$  with  $\gamma_n \in \mathcal{L}(\mathcal{X}) \forall n$ . The  $L_2$  convergence implies that there is a sub-sequence  $(\mathbb{1}_{\gamma_{\phi(n)}})_n$  that converges almost everywhere pointwise to  $f$  (as stated in Theorem 3.13 of [140]). It means that there exists a  $\lambda$ -null set  $\mathcal{N}$  s.t.  $\forall x \notin \mathcal{N}, \mathbb{1}_{\gamma_{\phi(n)}}(x) \xrightarrow[n \rightarrow +\infty]{} f(x)$ . As  $\mathbb{1}_{\gamma_{\phi(n)}}(x)$  is a sequence of 0 and 1, we have that its limit,  $f(x)$ , belong to  $\{0, 1\}$ . So,  $f = \mathbb{1}_{f^{-1}(\{1\})}$   $\lambda$ -almost everywhere and  $f^{-1}(\{1\}) \in \mathcal{L}(\mathcal{X})$  as  $f$  is measurable. So  $f \in \mathcal{F}_{\text{bin}}^*$ .  $\mathcal{F}_{\text{bin}}^*$  is thus closed.

Thus  $k_{\text{set}}$  is integrally strictly positive definite with respect to  $\mathcal{M}(\mathcal{L}^*(\mathcal{X}))$ , the set of signed measure on  $\mathcal{L}^*(\mathcal{X})$ , which implies that it is characteristic.  $\square$

### B.1.5 Proof of Proposition 2.4

**Proposition 2.4.** *With the previous notations, we have*

$$\mathbb{E} \left( \widehat{\text{HSIC}}(\mathbf{U}_A, \Gamma) - \text{HSIC}(\mathbf{U}_A, \Gamma) \right)^2 \leq 2 \left( \frac{2\sigma_1^2}{n(n-1)} + \frac{4(n-2)\sigma_2^2}{n(n-1)} + \frac{L^2\sigma_3^2}{m} \right),$$

where

- $\sigma_1^2 = \text{Var}((K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) k_{\text{set}}(\Gamma, \Gamma'))$ ,
- $\sigma_2^2 = \text{Var}(\mathbb{E}[(K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) k_{\text{set}}(\Gamma, \Gamma') | (\Gamma, \mathbf{U}_A)])$ ,
- $\sigma_3^2 = \mathbb{E}[(K_A(\mathbf{U}_A, \mathbf{U}_A') - 1)^2 \text{Var}(\mathbb{1}_{\Gamma \Delta \Gamma'}(\mathbf{X}) | (\mathbf{U}_A, \mathbf{U}_A', \Gamma, \Gamma'))]$ ,
- $L = \frac{\lambda(\mathcal{X})}{2\sigma^2}$ ,

and where  $(\mathbf{U}_A', \Gamma')$  is an independent copy of  $(\mathbf{U}_A, \Gamma)$ .

*Proof.* Let  $f(\mathbf{u}, z) = g(\mathbf{u})h(z)$  with  $g(\mathbf{u}) = (K_A(\mathbf{u}_1, \mathbf{u}_2) - 1)$ ,  $h(z) = e^{-\frac{\lambda(\mathcal{X})}{2\sigma^2} z}$ , and let  $\Phi(\mathbf{x}, \gamma_1, \gamma_2) = \mathbb{1}_{\gamma_1 \Delta \gamma_2}(x)$ . Let us have i.i.d. samples  $(\mathbf{U}_A^{(i)}, \Gamma^{(i)})$ ,  $i = 1, \dots, n$  of  $(\mathbf{U}_A, \Gamma)$  and  $(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)})$  of  $\mathbf{X} \sim \mathcal{U}(\mathcal{X})$ . We denote

$$\mathbf{H} = \text{HSIC}(\mathbf{U}_A, \Gamma) = \mathbb{E}[f(\mathbf{U}, \mathbf{U}', \mathbb{E}[\Phi(\mathbf{X}, \Gamma, \Gamma') | (\Gamma, \Gamma')])],$$

$$\mathbf{H}_{n,m} = \widehat{\text{HSIC}}_u(\mathbf{U}_A, \Gamma) = \frac{2}{n(n-1)} \sum_{i < j}^n f(\mathbf{U}_A^{(i)}, \mathbf{U}_A^{(j)}, \frac{1}{m} \sum_{k=1}^m \Phi(\mathbf{X}^{(k)}, \Gamma^{(i)}, \Gamma^{(j)})),$$

and

$$\mathbf{H}_n = \frac{2}{n(n-1)} \sum_{i < j}^n f(\mathbf{U}_A^{(i)}, \mathbf{U}_A^{(j)}, \mathbb{E}[\Phi(\mathbf{X}, \Gamma^{(i)}, \Gamma^{(j)}) | (\Gamma^{(i)}, \Gamma^{(j)})]).$$

First we split the risk into two terms:

$$\mathbb{E}|\mathbf{H}_{n,m} - \mathbf{H}|^2 \leq 2\mathbb{E}|\mathbf{H}_n - \mathbf{H}|^2 + 2\mathbb{E}|\mathbf{H}_n - \mathbf{H}_{n,m}|^2$$

The first term is the variance of a classic U-statistic of order 2:

$$\mathbb{E}|\mathbf{H}_n - \mathbf{H}|^2 = \frac{2\sigma_1^2}{n(n-1)} + \frac{4(n-2)\sigma_2^2}{n(n-1)}.$$

The second term can be developed:

$$\mathbb{E} |H_n - H_{nm}|^2 = \frac{4}{n^2(n-1)^2} \left( \sum_{i < j}^n \sum_{p < l}^n \mathbb{E} (E_{ij} - E_{ij,m}) (E_{pl} - E_{pl,m}) \right),$$

with

$$E_{ij} = f(\mathbf{U}_A^{(i)}, \mathbf{U}_A^{(j)}, \mathbb{E}[\Phi(\mathbf{X}, \Gamma^{(i)}, \Gamma^{(j)}) | (\Gamma^{(i)}, \Gamma^{(j)})])$$

and

$$E_{ij,m} = f(\mathbf{U}_A^{(i)}, \mathbf{U}_A^{(j)}, \frac{1}{m} \sum_{k=1}^m \Phi(\mathbf{X}^{(k)}, \Gamma^{(i)}, \Gamma^{(j)}).$$

As  $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)}$  are common to each  $E_{ij,m}$ , the terms  $E_{ij,m}$  and  $E_{pl,m}$  are not independent even if  $i, j, p, l$  are pairwise distinct. We can still bound them but we will lose one order of convergence in  $m$ . We first have

$$|\mathbb{E} (E_{ij} - E_{ij,m}) (E_{pl} - E_{pl,m})| \leq (\mathbb{E} |E_{ij} - E_{ij,m}|^2 \mathbb{E} |E_{pl} - E_{pl,m}|^2)^{\frac{1}{2}} \quad (\text{B.1})$$

$$= (\mathbb{E} |E_{12} - E_{12,m}|^2) \quad (\text{B.2})$$

$$\leq \frac{L^2}{m} \mathbb{E} \left[ (K_A(\mathbf{U}_A, \mathbf{U}_A') - 1)^2 \text{Var}(\Phi(\mathbf{X}, \Gamma, \Gamma') | (\mathbf{U}_A, \mathbf{U}_A', \Gamma, \Gamma')) \right],$$

using Cauchy-Schwarz inequality in (B.1) and that  $h$  is  $L$ -lipschitz. Summing each term, we obtain

$$\mathbb{E} |H_n - H_{nm}|^2 \leq \frac{L^2}{m} \mathbb{E} \left[ (K_A(\mathbf{U}_A, \mathbf{U}_A') - 1)^2 \text{Var}(\Phi(\mathbf{X}, \Gamma, \Gamma') | (\mathbf{U}_A, \mathbf{U}_A', \Gamma, \Gamma')) \right].$$

Putting all results together, we get

$$\mathbb{E} |H_{n,m} - H|^2 \leq 2 \left( \frac{2\sigma_1^2}{n(n-1)} + \frac{4(n-2)\sigma_2^2}{n(n-1)} + \frac{L^2\sigma_3^2}{m} \right).$$

□

### B.1.6 Convergence rate in the case of Independent $m$ sample

If an  $m$ -sample  $\mathbf{X}_{ij}^{(k)}$  is drawn independently for each  $(i, j)$ , we can obtain an asymptotic rate of  $\mathcal{O}(\frac{1}{n} + \frac{1}{m^2})$ . Indeed in (B.1) we use the independence between  $(\mathbf{X}_{ij}^{(k)})_k$  and  $(\mathbf{X}_{pl}^{(k)})_k$  for  $i, j, p, l$  pairwise distinct:

$$\begin{aligned} |\mathbb{E} (E_{ij} - E_{ij,m}) (E_{pl} - E_{pl,m})| &= |\mathbb{E} (E_{ij} - E_{ij,m}) \mathbb{E} (E_{pl} - E_{pl,m})| \\ &= |\mathbb{E} (E_{12} - E_{12,m})|^2 \end{aligned}$$

Then by applying Taylor's theorem (with Lagrange remainder) to  $h$  with  $a = \mathbb{E}(\Phi(\mathbf{X}, \Gamma^{(1)}, \Gamma^{(2)}) | (\Gamma^{(1)}, \Gamma^{(2)}))$

and  $b = \frac{1}{m} \sum_{k=1}^m \Phi(\mathbf{X}^{(k)}, \Gamma^{(1)}, \Gamma^{(2)})$ , we have the existence of  $\theta \in \mathbb{R}^+$  such that,

$$\begin{aligned} E_{12,m} - E_{12} &= (K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) (h(b) - h(a)) \\ &= (K_A(\mathbf{U}_A, \mathbf{U}_A') - 1) \left[ h'(a)(b - a) + \frac{h''(\theta)}{2} (b - a)^2 \right]. \end{aligned}$$

Then we take the expectation and use the tower property and we use that Monte Carlo estimators are unbiased (i.e.  $\mathbb{E}(b|(\Gamma^{(1)}, \Gamma^{(2)})) = a$ ) which leads to

$$\begin{aligned}\mathbb{E}(E_{12,m} - E_{12}) &= \mathbb{E}[\mathbb{E}(E_{12,m} - E_{12} | \mathbf{U}_A, \mathbf{U}'_A, \Gamma, \Gamma')] \\ &= \mathbb{E}\left[\left(K_A(\mathbf{U}_A, \mathbf{U}'_A) - 1\right) \mathbb{E}\left(\frac{h''(\theta)}{2}(b-a)^2 | \mathbf{U}_A, \mathbf{U}'_A, \Gamma, \Gamma'\right)\right].\end{aligned}$$

Then, as  $h''$  is bounded by  $L$ ,

$$|\mathbb{E}(E_{12,m} - E_{12})| \leq \frac{L^2}{2m} \mathbb{E}\left[\left|K_A(\mathbf{U}_A, \mathbf{U}'_A) - 1\right| \text{Var}(\Phi(\mathbf{X}, \Gamma, \Gamma') | (\mathbf{U}_A, \mathbf{U}'_A, \Gamma, \Gamma'))\right],$$

which finally leads to

$$|\mathbb{E}(E_{ij} - E_{ij,m})(E_{pl} - E_{pl,m})| \leq \frac{L^4}{4m^2} \sigma_4^4,$$

where

$$\sigma_4^2 = \mathbb{E}\left[\left|K_A(\mathbf{U}_A, \mathbf{U}'_A) - 1\right| \text{Var}(\Phi(\mathbf{X}, \Gamma, \Gamma') | (\mathbf{U}_A, \mathbf{U}'_A, \Gamma, \Gamma'))\right].$$

If  $i = p$  or  $i = l$  or  $j = p$  or  $j = l$ , we lost the independence so we use the previous result that

$$|\mathbb{E}(E_{ij} - E_{ij,m})(E_{pl} - E_{pl,m})| \leq \frac{L^2}{m} \sigma_3^2.$$

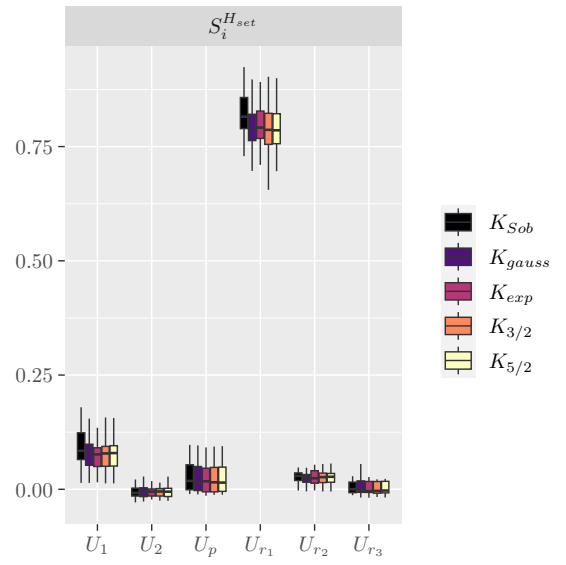
There are  $\frac{n(n-1)(n-2)(n-3)}{4}$   $i, j, p, l$  pairwise distinct with  $i < j$  and  $p < l$  so we finally obtain that,

$$\begin{aligned}\mathbb{E}|H_{n,m} - H|^2 &\leq 2\left(\frac{2\sigma_1^2}{n(n-1)} + \frac{4(n-2)\sigma_2^2}{n(n-1)} + \frac{L^2 2(2n-3)\sigma_3^2}{n(n-1)m} + \frac{L^4(n-2)(n-3)\sigma_4^4}{4n(n-1)m^2}\right) \\ &= \mathcal{O}\left(\frac{1}{n} + \frac{1}{m^2}\right).\end{aligned}$$

## B.2 Additional plots

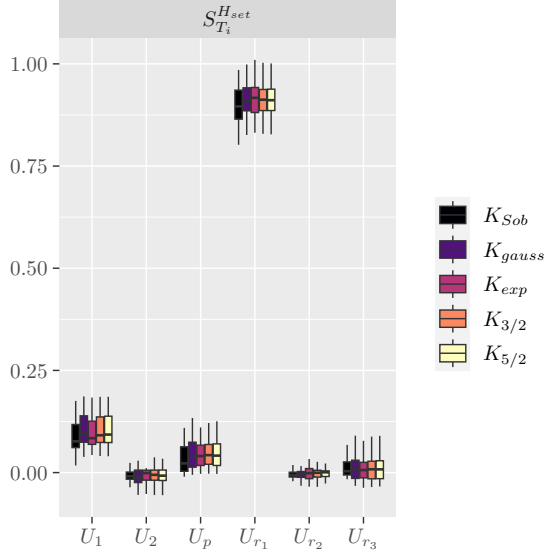
|           | $U_1$ | $U_2$ | $U_p$ | $U_{r_1}$ | $U_{r_2}$ | $U_{r_3}$ |
|-----------|-------|-------|-------|-----------|-----------|-----------|
| $K_{Sob}$ | 5     | 95    | 55    | 0         | 75        | 90        |
| $K_{Sob}$ | 5     | 100   | 60    | 0         | 75        | 95        |
| $K_{Sob}$ | 15    | 95    | 55    | 0         | 65        | 95        |
| $K_{Sob}$ | 10    | 95    | 55    | 0         | 65        | 95        |
| $K_{Sob}$ | 5     | 100   | 55    | 0         | 70        | 95        |

(a) Acceptance rate (%) over 20 independence tests with a risk of 5%

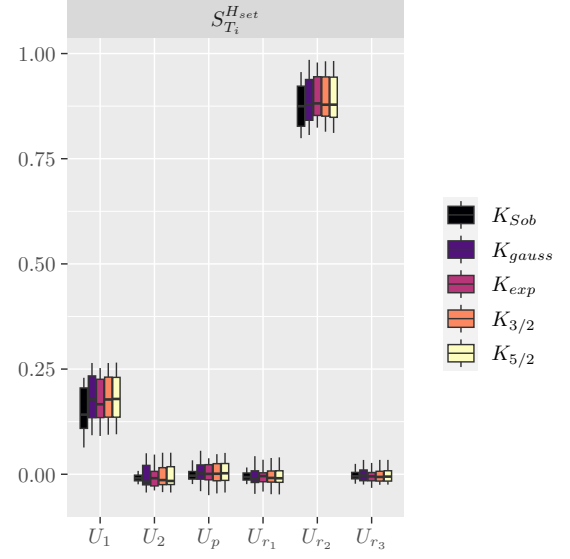


(b) Estimations of  $\hat{S}_i^{H_{set}}$

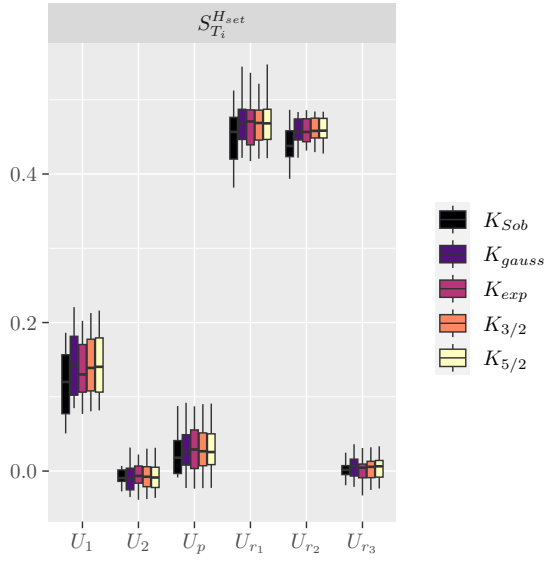
**Figure B.1:** Acceptance rate (a) and estimation of  $\hat{S}_i^{H_{set}}$  (b) for the excursion set  $\Gamma_{(g_1, g_2)}$  computed for 5 kernels with  $n = 100$ ,  $m = 100$  and repeated 20 times



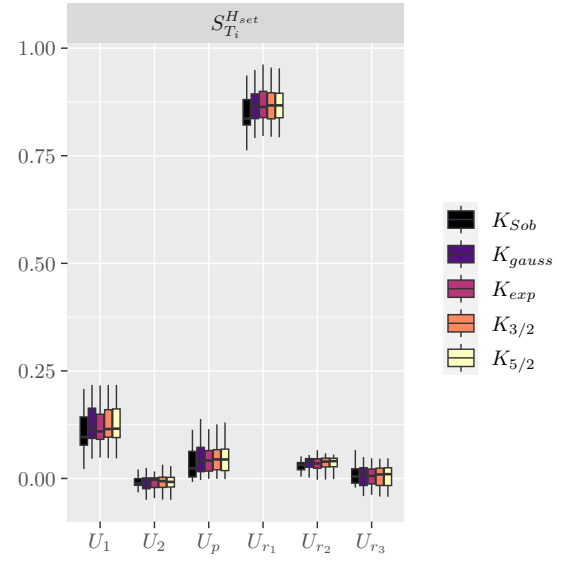
(a)  $\Gamma_{g_1}$



(b)  $\Gamma_{g_2}$

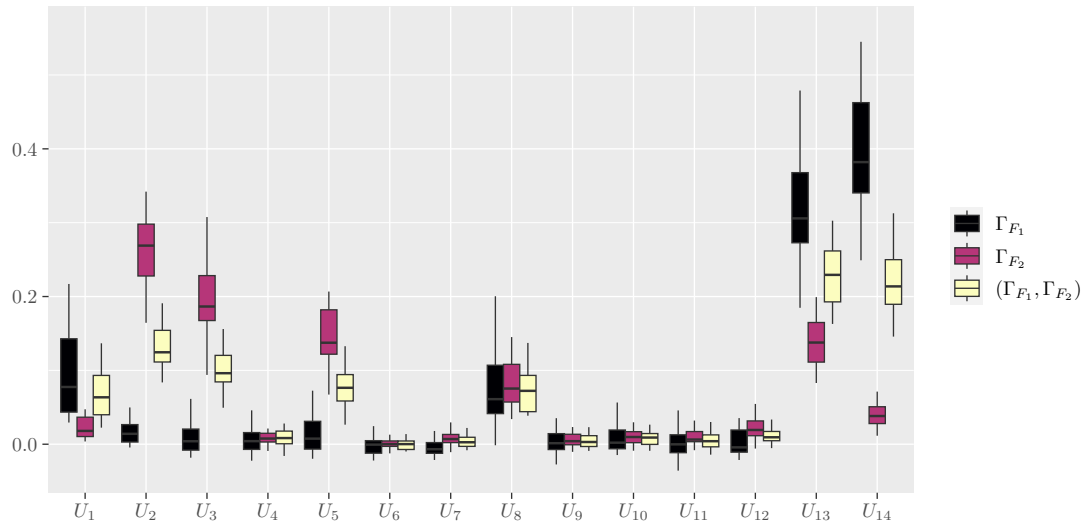


(c)  $(\Gamma_{g_1}, \Gamma_{g_2})$



(d)  $\Gamma_{(g_1, g_2)}$

**Figure B.2:** Estimation of the total-order indices  $\hat{S}_{T_i}^{H_{set}}$  for the oscillator case computed for 5 kernels with  $n = 100$ ,  $m = 100$  and repeated 20 times



**Figure B.3:** Estimation of  $\hat{S}_i^{H_{set}}$  for the excursion sets  $\Gamma_{f_1}$ ,  $\Gamma_{f_2}$  and the pair  $(\Gamma_{f_1}, \Gamma_{f_2})$  with  $-q_1 = 420$  N.m and  $q_2 = 7\%$  computed with the Sobolev input kernel and with  $n = 100$ ,  $m = 100$ . 20 replicates.



# C

## Appendix of chapter 3: Alternative Vorob'ev-based indices

In Chapter 3 we propose the Vorob'ev-based index

$$S_i^V = 1 - \frac{\mathbb{E}[\text{VMD}(\Gamma|U_i)]}{\text{VMD}(\Gamma)}.$$

It is inspired by Sobol's indices. But to get the result of Proposition 3.1 that  $S_i^V$  lies between 0 and 1, we had to define  $S_i^V$  as 1 minus the total index associated with the indices  $-i$ , and use the Vorob'ev Median Deviation instead of the Vorob'ev deviation. Indices that correspond directly to Sobol' indices (i.e., as part of the variance of the conditional expectation) can also be defined as the Vorob'ev deviation of the conditional Vorob'ev expectation:

$$\begin{aligned}\tilde{S}_i^V &= \frac{\text{Var}^V(\mathbb{E}^V(\Gamma|U_i))}{\text{Var}^V(\Gamma)} \\ &= \frac{\mathbb{E}[\lambda(\mathbb{E}^V(\Gamma|U_i)\Delta\mathbb{E}^V(\Gamma|U_i))]}{\mathbb{E}[\lambda(\Gamma\Delta\mathbb{E}^V(\Gamma))]}.\end{aligned}$$

or of the conditional Vorob'ev median:

$$\begin{aligned}\bar{S}_i^V &= \frac{\text{VMD}(Q_{0.5}^i)}{\text{VMD}(\Gamma)} \\ &= \frac{\mathbb{E}[\lambda(Q_{0.5}^i\Delta Q_{0.5}(\Gamma))]}{\mathbb{E}[\lambda(\Gamma\Delta Q_{0.5}(\Gamma))]}.\end{aligned}$$

In this case, however, we have not been able to show that the indices are less than 1, even though numerical tests seem to verify it. The difficulties arise mainly from the fact that, unlike the classical case of random variables, we don't have either

$$\mathbb{E}^V[\mathbb{E}^V(\Gamma|U_i)] = \mathbb{E}^V(\Gamma)$$

nor

$$Q_{0.5}(Q_{0.5}^i) = Q_{0.5}.$$

Even if we can show that they are less than 1, their estimation is much more difficult due to the presence of the Vorob'ev median or expectation of a conditional median or expectation.

# D

## Appendix of chapter 4: On-seq-SA algorithm

---

**Algorithm 7** EFISUR with chained SA steps on acquisition functions: **On-seq-SA**


---

```

1: Fix a number of maximum SA steps
2: Design of Experiments (DoE) in the joint space  $\mathcal{X} \times \mathcal{U}$ :  $D^{(t)} = \{(x^k, u^k), k = 1, \dots, t\}$ 
3: Calculate simulator responses:  $f^{(t)} = (f(x^k, u^k))_{k=1}^t$  and  $g^{(t)} = (g(x^k, u^k))_{k=1}^t$ 
4: while  $t \leq$  maximum budget do
5:   Fit GPs of  $f$  and  $g$ :  $F_{(x,u)}^{(t)}$  and  $G_{(x,u)}^{(t)}$ 
6:   Calculate  $Z_x^{(t)}$  and  $C_x^{(t)}$ 
7:    $k \leftarrow 0$  and  $x_A^{t+1} \leftarrow ()$ 
8:   while ( $k <$  number of maximum SA steps) and  $\dim(x_A^{t+1}) < d$  do
9:     Estimate  $\text{HSIC}(X_i, \text{EFI}_{Z^{(t)}, C^{(t)}}(X_{-A}, x_A^{t+1}))$  and  $p_{X_i}$  for  $i \in -A$  with  $X_i \sim \mathcal{U}(\mathcal{X}_i)$ .
10:    Screen the  $x_i$ :  $A' = \{i \in -A, p_{X_i}^{\text{EFI}} \leq \alpha_I\}$  and fix the others randomly at  $x_{-(A \cup A')}$ .
11:     $x_{A'}^{t+1} = \arg \max_{x_{A'} \in \mathcal{X}_{A'}} \text{EFI}_{Z^{(t)}, C^{(t)}}(x_{A'}, x_A^{t+1}, x_{-(A \cup A')})$ 
12:     $x_{A \cup A'}^{t+1} \leftarrow (x_A^{t+1}, x_{A'}^{t+1})$ ,  $A \leftarrow A \cup A'$  and  $k \leftarrow k + 1$ 
13:  end while
14:  Set  $x^{t+1} = (x_A^{t+1}, x_{-A})$ 
15:   $k = 0$  and  $u_B^{t+1} \leftarrow ()$ 
16:  while ( $k <$  number of maximum SA steps) and  $\dim(u_B^{t+1}) < m$  do
17:    Estimate  $\text{HSIC}(U_i, S(x^{t+1}, U))$  and  $p_{U_j}^S$  for  $i \in -B$  with  $U_j \sim \mathbb{P}_{U_j}$ 
18:    Screen the  $U_j$ :  $B' = \{j \in -B, p_{U_j} \leq \alpha_I\}$  and fix the others randomly at  $u_{-(B \cup B')}$ 
19:     $u_{B'}^{t+1} = \arg \min_{u_{B'} \in \mathcal{U}_{B'}} S(x^{t+1}, u_{B'}, u_B^{t+1}, u_{-(B \cup B')})$ 
20:     $u_{B \cup B'}^{t+1} \leftarrow (u_B^{t+1}, u_{B'}^{t+1})$ ,  $B \leftarrow B \cup B'$  and  $k \leftarrow k + 1$ 
21:  end while
22:  Set  $u^{t+1} = (u_B^{t+1}, u_{-B})$ 
23:  Calculate simulator responses at the next point  $(x^{t+1}, u^{t+1})$ 
24:  Update the DoE and  $t \leftarrow t + 1$ 
25: end while

```

---

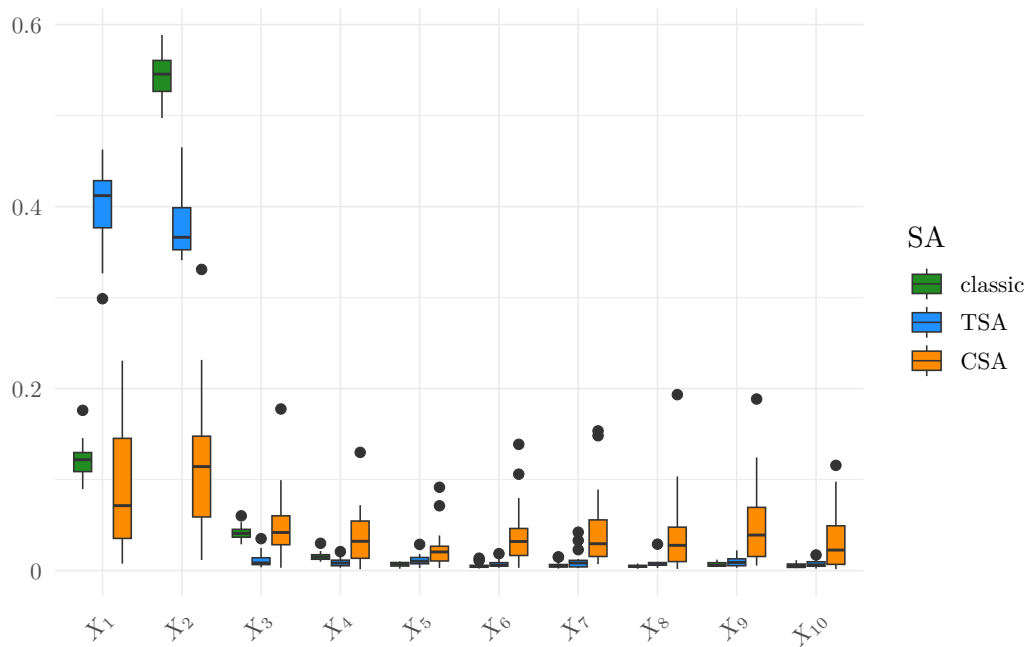




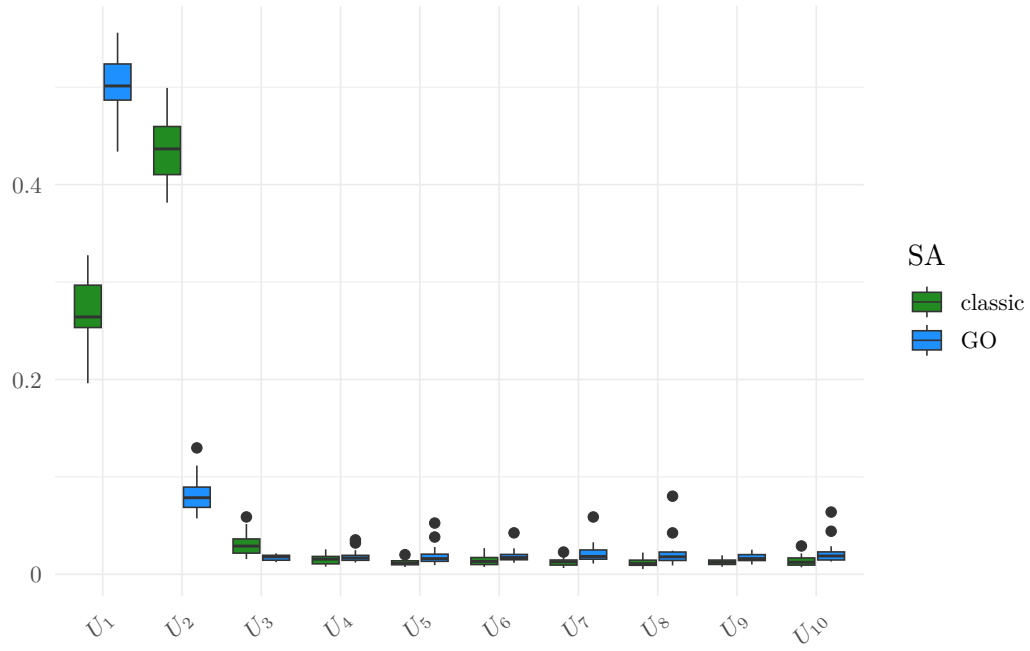
# E

## Appendix of chapter 5: Additional numerical results

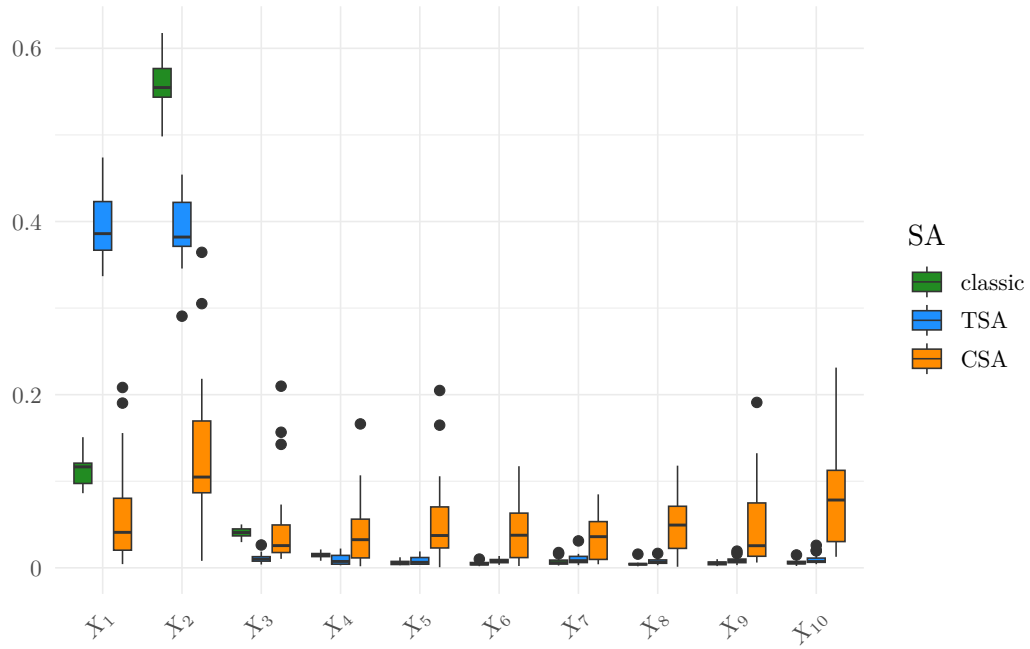
### E.1 Additional Sensitivity Analysis plots



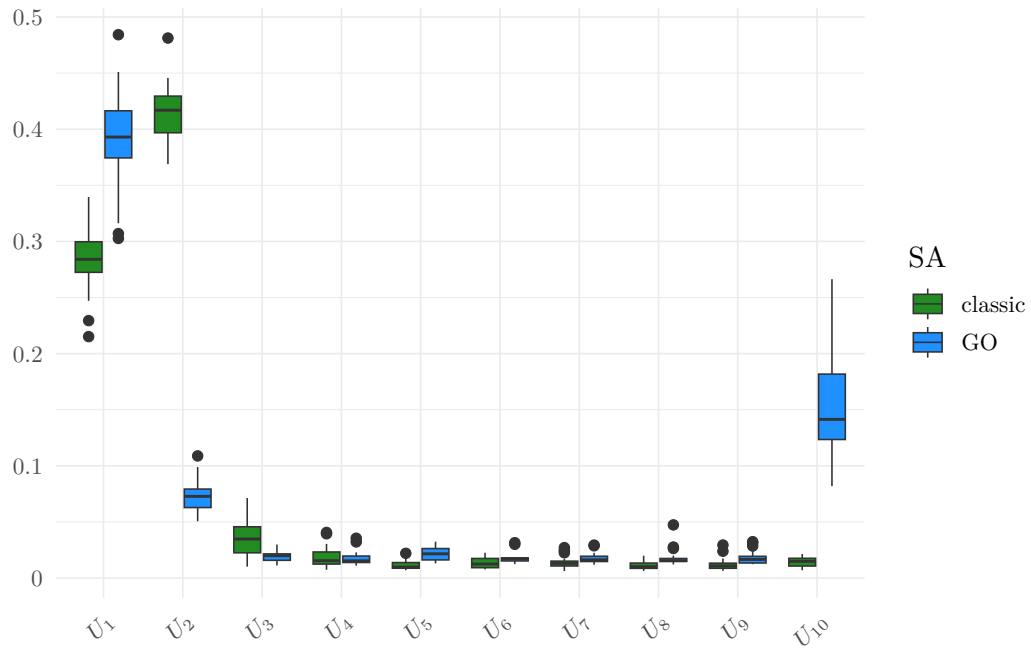
**Figure E.1:** Classic, target and conditional HSIC on the design inputs for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . Boxplots are obtained with 20 repetitions.



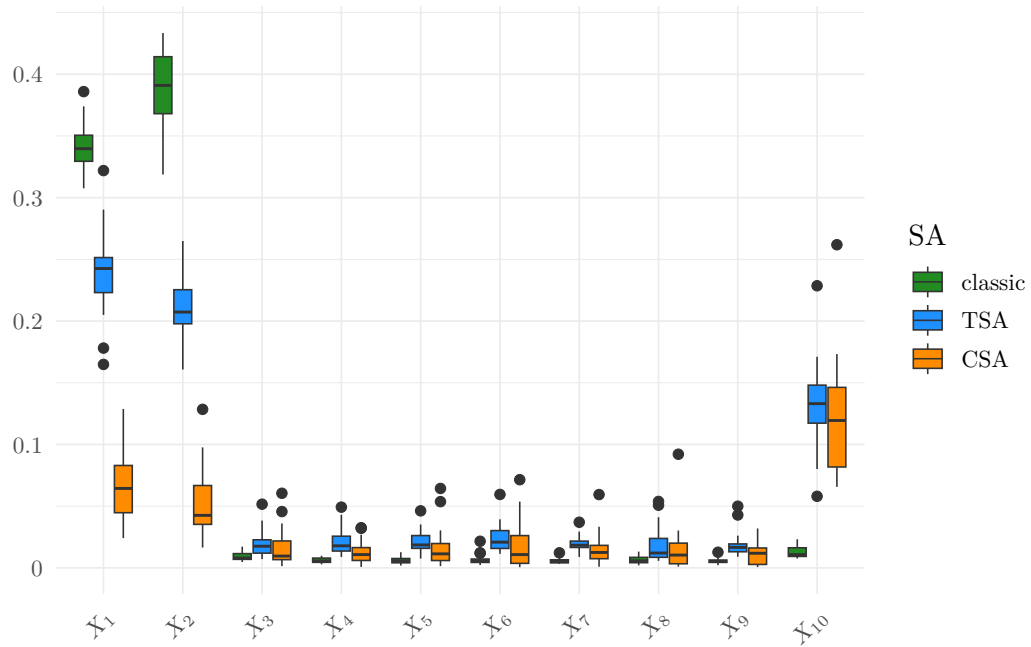
**Figure E.2:** First order HSIC of sets on the uncertain inputs for  $f_{Glob}^{10}$  and  $g_{Glob}^{10}$ . Boxplots are obtained with 20 repetitions.



**Figure E.3:** Classic, target and conditional HSIC on the design inputs for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ . Boxplots are obtained with 20 repetitions.



**Figure E.4:** First order HSIC of sets (goal-oriented and not goal-oriented) on the uncertain inputs for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ . Boxplots are obtained with 20 repetitions.



**Figure E.5:** Classic, target and conditional HSIC on the design inputs, zoomed around the optimum, for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ . Boxplots are obtained with 20 repetitions.

## E.2 Complete screening tables of Off-seq-SA

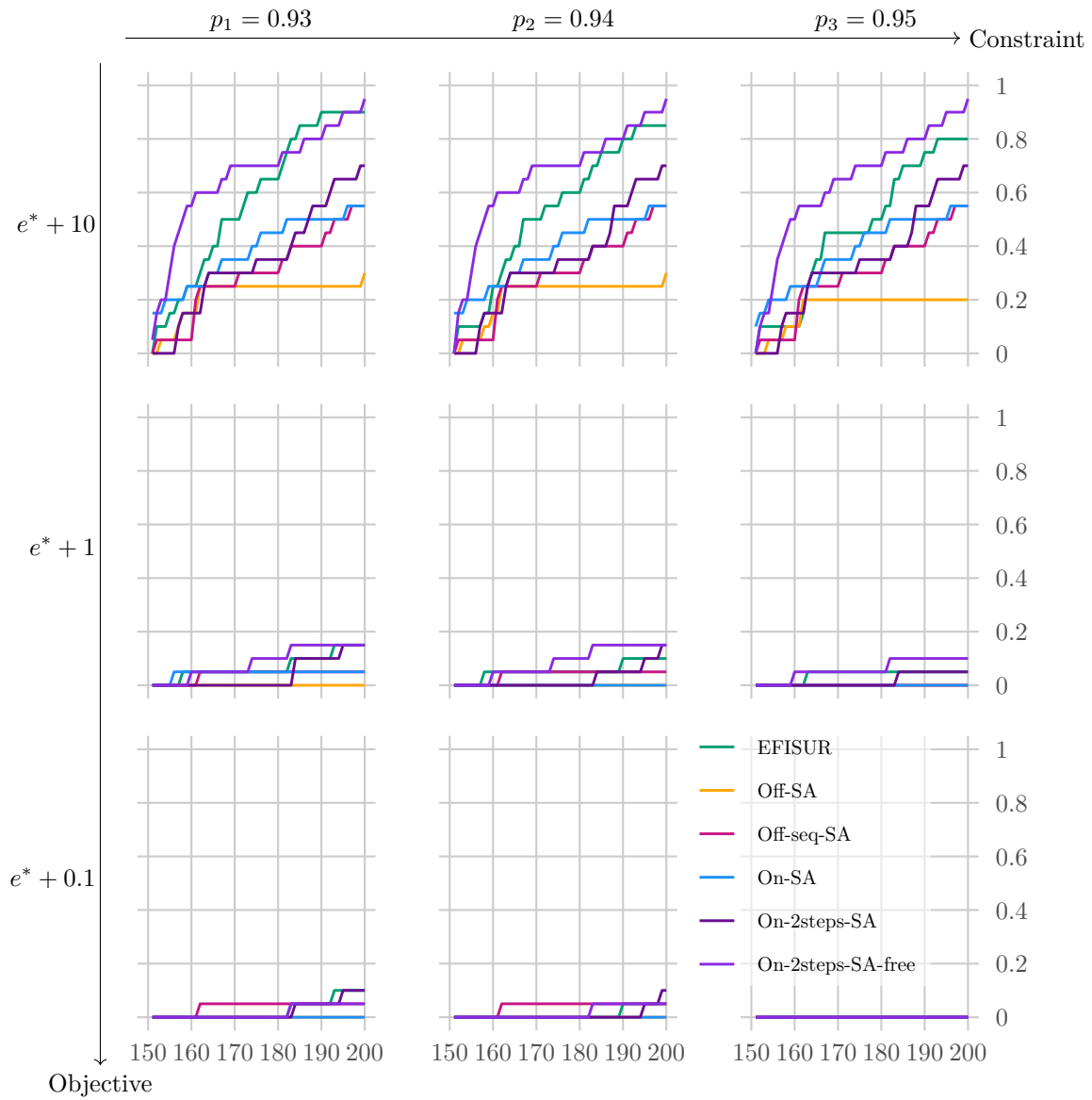
| Budget allocation     | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 100 + 10 × 10         | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
|                       | 1     | 1     | 0.05  | 0.1   | 0.1   | 0.05  | 0.1   | 0.1   | 0.2   | 0.15     |
|                       | 1     | 1     | 0.05  | 0.15  | 0.25  | 0.1   | 0.05  | 0.1   | 0.1   | 0.15     |
|                       | 1     | 1     | 0.2   | 0.2   | 0.1   | 0     | 0.15  | 0.2   | 0.15  | 0.4      |
|                       | 1     | 1     | 0.15  | 0.25  | 0.1   | 0.05  | 0.15  | 0.25  | 0.1   | 0.65     |
|                       | 1     | 1     | 0.25  | 0.25  | 0.1   | 0.1   | 0.1   | 0.2   | 0.15  | 0.75     |
|                       | 1     | 1     | 0.3   | 0.25  | 0.2   | 0.2   | 0.1   | 0.25  | 0.1   | 0.85     |
|                       | 1     | 1     | 0.3   | 0.4   | 0.1   | 0.15  | 0.2   | 0.15  | 0.3   | 0.95     |
|                       | 1     | 1     | 0.4   | 0.25  | 0.05  | 0.2   | 0.2   | 0.1   | 0.3   | 1        |
|                       | 1     | 1     | 0.45  | 0.2   | 0.1   | 0.1   | 0.45  | 0.2   | 0.1   | 1        |
| 100 + 10 × 10 - seqSA | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
|                       | 1     | 1     | 0.1   | 0.1   | 0.15  | 0.15  | 0     | 0.05  | 0.1   | 0.15     |
|                       | 1     | 1     | 0.15  | 0.1   | 0.1   | 0.3   | 0.05  | 0.05  | 0.2   | 0.2      |
|                       | 1     | 1     | 0.2   | 0.15  | 0.2   | 0.2   | 0.1   | 0.05  | 0.1   | 0.2      |
|                       | 1     | 1     | 0.1   | 0.3   | 0.15  | 0.25  | 0.1   | 0     | 0.1   | 0.4      |
|                       | 1     | 1     | 0.15  | 0.2   | 0.1   | 0.15  | 0.05  | 0.1   | 0.05  | 0.5      |
|                       | 1     | 1     | 0.25  | 0.05  | 0.15  | 0.15  | 0.1   | 0     | 0.15  | 0.55     |
|                       | 1     | 1     | 0.15  | 0.3   | 0.15  | 0.15  | 0.1   | 0.05  | 0.2   | 0.6      |
|                       | 1     | 1     | 0.3   | 0.1   | 0.1   | 0.05  | 0.2   | 0.1   | 0.2   | 0.8      |
|                       | 1     | 1     | 0.5   | 0.15  | 0.1   | 0.2   | 0.15  | 0.15  | 0.15  | 0.95     |
| 150 + 5 × 10          | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
|                       | 1     | 1     | 0.05  | 0.1   | 0.1   | 0.05  | 0.1   | 0.3   | 0     | 0.15     |
|                       | 1     | 1     | 0.1   | 0.2   | 0.1   | 0.05  | 0.05  | 0.25  | 0.15  | 0.25     |
|                       | 1     | 1     | 0.15  | 0.15  | 0.05  | 0.15  | 0.2   | 0.25  | 0.1   | 0.4      |
|                       | 1     | 1     | 0.15  | 0.2   | 0.1   | 0.1   | 0.15  | 0.1   | 0.15  | 0.8      |
| 150 + 5 × 10 - seqSA  | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0        |
|                       | 1     | 1     | 0.15  | 0     | 0     | 0.15  | 0.05  | 0.1   | 0.05  | 0.1      |
|                       | 1     | 1     | 0.1   | 0.05  | 0.05  | 0.1   | 0.1   | 0     | 0.1   | 0.1      |
|                       | 1     | 1     | 0.15  | 0.1   | 0.2   | 0.15  | 0.1   | 0     | 0.15  | 0.35     |
|                       | 1     | 1     | 0.1   | 0     | 0.1   | 0.1   | 0.1   | 0.15  | 0.25  | 0.6      |

**Table E.1:** Screening results of the  $X$ 's on 20 runs for **Off-seq-SA** algorithm with a SA step every 10 iterations for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

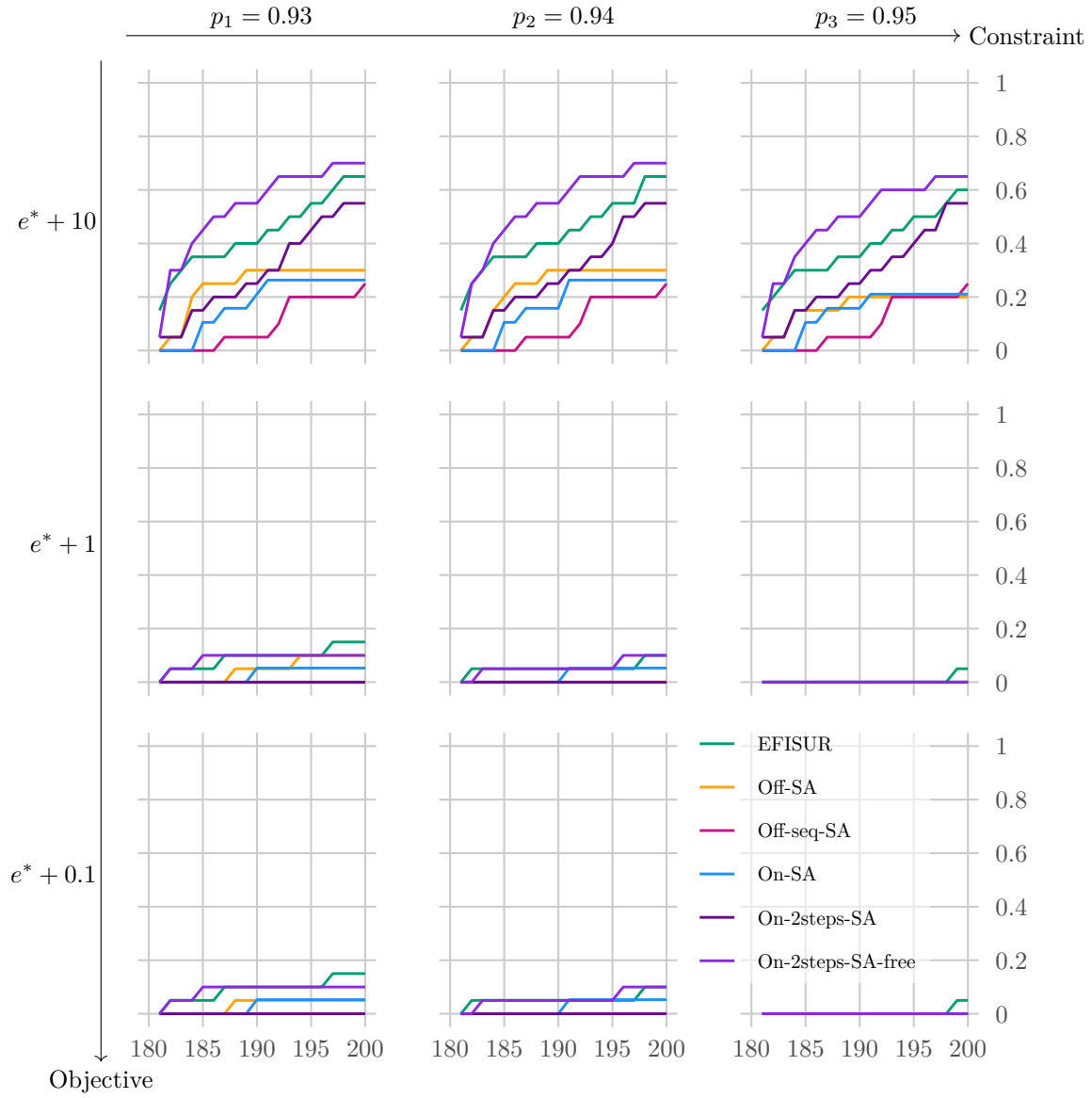
| Budget allocation     | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 100 + 10 × 10         | 0.7   | 1     | 0.15  | 0     | 0     | 0     | 0.1   | 0     | 0     | 0.05     |
|                       | 0.85  | 1     | 0.4   | 0.15  | 0.05  | 0.25  | 0.35  | 0.15  | 0     | 0.4      |
|                       | 0.75  | 1     | 0.35  | 0.3   | 0.25  | 0.15  | 0.15  | 0.2   | 0.1   | 0.55     |
|                       | 0.75  | 1     | 0.3   | 0.05  | 0.3   | 0.2   | 0.25  | 0.15  | 0.15  | 0.4      |
|                       | 0.85  | 1     | 0.15  | 0.25  | 0.3   | 0.15  | 0.25  | 0.25  | 0.1   | 0.35     |
|                       | 0.85  | 1     | 0.35  | 0.15  | 0.25  | 0.2   | 0.35  | 0.1   | 0.2   | 0.4      |
|                       | 0.85  | 1     | 0.2   | 0.3   | 0.1   | 0.25  | 0.25  | 0.1   | 0.15  | 0.4      |
|                       | 0.85  | 1     | 0.2   | 0.1   | 0.15  | 0.2   | 0.3   | 0.15  | 0.15  | 0.7      |
|                       | 0.85  | 1     | 0.3   | 0.15  | 0.1   | 0.1   | 0.25  | 0.05  | 0.05  | 0.7      |
|                       | 0.85  | 1     | 0.2   | 0.2   | 0.1   | 0.15  | 0.1   | 0.15  | 0.15  | 0.6      |
| 100 + 10 × 10 - seqSA | 0.8   | 1     | 0     | 0     | 0     | 0.05  | 0.15  | 0     | 0     | 0        |
|                       | 0.6   | 1     | 0.25  | 0.25  | 0.1   | 0.15  | 0.2   | 0.15  | 0.15  | 0.1      |
|                       | 0.6   | 1     | 0.2   | 0.2   | 0.25  | 0.3   | 0.15  | 0.25  | 0.35  | 0.05     |
|                       | 0.8   | 1     | 0.3   | 0.4   | 0.2   | 0.3   | 0.15  | 0.15  | 0.25  | 0.15     |
|                       | 0.85  | 1     | 0.35  | 0.4   | 0.15  | 0.35  | 0.15  | 0.15  | 0.4   | 0.1      |
|                       | 0.95  | 1     | 0.2   | 0.4   | 0.25  | 0.2   | 0.3   | 0.15  | 0.3   | 0.25     |
|                       | 0.95  | 1     | 0.4   | 0.2   | 0.3   | 0.2   | 0.15  | 0.1   | 0.4   | 0.35     |
|                       | 1     | 1     | 0.45  | 0.25  | 0.25  | 0.3   | 0.2   | 0.1   | 0.2   | 0.3      |
|                       | 1     | 1     | 0.2   | 0.4   | 0.15  | 0.15  | 0.2   | 0.15  | 0.15  | 0.5      |
|                       | 0.95  | 1     | 0.4   | 0.2   | 0.25  | 0.15  | 0.1   | 0.15  | 0.15  | 0.5      |
| 150 + 5 × 10          | 0.9   | 1     | 0.05  | 0     | 0     | 0     | 0     | 0     | 0     | 0.05     |
|                       | 0.85  | 0.95  | 0.25  | 0.1   | 0.05  | 0.1   | 0.25  | 0.15  | 0.25  | 0.35     |
|                       | 0.9   | 1     | 0.35  | 0.15  | 0.2   | 0.1   | 0.4   | 0.2   | 0.25  | 0.3      |
|                       | 0.85  | 0.95  | 0.35  | 0.1   | 0.2   | 0.25  | 0.05  | 0.05  | 0.2   | 0.2      |
|                       | 0.85  | 1     | 0.35  | 0.2   | 0.15  | 0.15  | 0.25  | 0.15  | 0.3   | 0.1      |
| 150 + 5 × 10 - seqSA  | 0.85  | 1     | 0     | 0.05  | 0     | 0     | 0     | 0     | 0     | 0.1      |
|                       | 0.85  | 0.9   | 0.35  | 0.2   | 0.3   | 0.15  | 0.2   | 0.3   | 0.3   | 0.15     |
|                       | 0.85  | 1     | 0.3   | 0.15  | 0.4   | 0.3   | 0.3   | 0.15  | 0.25  | 0.15     |
|                       | 0.75  | 0.9   | 0.2   | 0.2   | 0.25  | 0.1   | 0.35  | 0.35  | 0.3   | 0.3      |
|                       | 0.9   | 1     | 0.15  | 0.35  | 0.3   | 0.2   | 0.35  | 0.2   | 0.35  | 0.15     |

**Table E.2:** Screening results of the  $U$ 's on 20 runs for **Off-seq-SA** algorithm with a SA step every 10 iterations for  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$ .

### E.3 Comparison of the algorithms for different budget allocations



**Figure E.6:** Data profiles of all algorithms  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with 50 iterations for 20 runs associated to different initial lhs design.



**Figure E.7:** Data profiles of all algorithms  $f_{Loc}^{10}$  and  $g_{Loc}^{10}$  with nine goals and with 20 iterations for 20 runs associated to different initial lhs design.



## Bibliography

- [1] Alekh Agarwal and Tong Zhang. “Minimax regret optimization for robust machine learning under distribution shift”. *Conference on Learning Theory*. 2022. DOI.
- [2] Shipra Agrawal and Navin Goyal. “Analysis of thompson sampling for the multi-armed bandit problem”. *Conference on learning theory*. JMLR Workshop and Conference Proceedings. 2012. DOI.
- [3] Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. “Unexpected improvements to expected improvement for bayesian optimization”. *Advances in Neural Information Processing Systems*. 2023. DOI.
- [4] Tanguy Appriou, Didier Rulli  re, and David Gaudrie. “High-dimensional Bayesian Optimization with a Combination of Kriging models”. *Structural and Multidisciplinary Optimization*. 2024. DOI.
- [5] Rub  n Conde Arenzana, Andr  s F L  pez-Lopera, Sylvain Mouton, Nathalie Bartoli, and Thierry Lefebvre. “Multi-fidelity Gaussian process model for CFD and wind tunnel data fusion”. 2021. URL.
- [6] Nachman Aronszajn. “Theory of reproducing kernels”. *Transactions of the American mathematical society*. 1950. DOI.
- [7] Dario Azzimonti. “Two types of Bayesian excursion set estimates based on Gaussian process models”. *21st European Young Statisticians Meeting*. 2019. URL.
- [8] Paul Balan  a and Erick Herbin. “A set-indexed Ornstein-Uhlenbeck process”. *Electronic Communications in Probability*. 2012. DOI.
- [9] Hans-Georg Beyer and Bernhard Sendhoff. “Robust optimization—a comprehensive survey”. *Computer methods in applied mechanics and engineering*. 2007. DOI.
- [10] JC Bezdek, RJ Hathaway, RE Howard, CA Wilson, and MP Windham. “Local convergence analysis of a grouped variable version of coordinate descent”. *Journal of Optimization Theory and Applications*. 1987. DOI.
- [11] Mickael Binois and Nathan Wyc  ff. “A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization”. *ACM Transactions on Evolutionary Learning and Optimization*. 2022. DOI.
- [12] Micka  l Binois, David Ginsbourger, and Olivier Roustant. “On the choice of the low-dimensional domain for global optimization via random embeddings”. *Journal of global optimization*. 2020. DOI.
- [13] Emanuele Borgonovo. “A new uncertainty importance measure”. *Reliability Engineering & System Safety*. 2007. DOI.

- [14] Emanuele Borgonovo and Elmar Plischke. “Sensitivity analysis: A review of recent advances”. *European Journal of Operational Research*. 2016. DOI.
- [15] Sait Cakmak, Raul Astudillo Marban, Peter Frazier, and Enlu Zhou. “Bayesian optimization of risk measures”. *Advances in Neural Information Processing Systems*. 2020. DOI.
- [16] Karen Chan, Andrea Saltelli, and E Marian Scott. *Sensitivity analysis*. 2000.
- [17] Qi Chang, Changcong Zhou, Matthias GR Faes, and Marcos A Valdebenito. “Design optimization with variable screening by interval-based sensitivity analysis”. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*. 2024. DOI.
- [18] Sourav Chatterjee. “A new coefficient of correlation”. *Journal of the American Statistical Association*. 2021. DOI.
- [19] Wenjie Chen, Shengcai Liu, and Ke Tang. “A new knowledge gradient-based method for constrained bayesian optimization”. *preprint*. 2021. DOI.
- [20] Clément Chevalier, David Ginsbourger, Julien Bect, and Ilya Molchanov. “Estimating and quantifying uncertainties on level sets using the Vorobev expectation and deviation with Gaussian process models”. *mODa 10–Advances in Model-Oriented Design and Analysis*. Springer. 2013. DOI.
- [21] Moon-Hyun Chun, Seok-Jung Han, and Nam-IL Tak. “An uncertainty importance measure using a distance metric for the change in a cumulative distribution function”. *Reliability Engineering & System Safety*. 2000. DOI.
- [22] Alexis Cousin, Josselin Garnier, Martin Guiton, and Miguel Munoz Zuniga. “A two-step procedure for time-dependent reliability-based design optimization involving piece-wise stationary Gaussian processes”. *Structural and Multidisciplinary Optimization*. 2022. DOI.
- [23] Imre Csiszár. “On information-type measure of difference of probability distributions and indirect observations”. *Studia Sci. Math. Hungar.* 1967.
- [24] Sébastien Da Veiga. “Global sensitivity analysis with dependence measures”. *Journal of Statistical Computation and Simulation*. 2015. DOI.
- [25] Sébastien Da Veiga. “Kernel-based ANOVA decomposition and Shapley effects–Application to global sensitivity analysis”. *Preprint*. 2021. DOI.
- [26] Sébastien Da Veiga, Fabrice Gamboa, Bertrand Iooss, and Clémentine Prieur. *Basics and trends in sensitivity analysis: Theory and practice in R*. 2021. DOI.
- [27] Matthias De Lozzo and Amandine Marrel. “New improvements in the use of dependence measures for sensitivity analysis and screening”. *Journal of Statistical Computation and Simulation*. 2016. DOI.
- [28] Matthias De Lozzo and Amandine Marrel. “Sensitivity analysis with dependence and variance-based measures for spatio-temporal numerical simulators”. *Stochastic environmental research and risk assessment*. 2017. DOI.
- [29] Xiaoping Du, Agus Sudjianto, and Wei Chen. “An integrated framework for optimization under uncertainty using inverse reliability strategy”. *J. Mech. Des.* 2004. DOI.
- [30] Nicolas Durrande, David Ginsbourger, Olivier Roustant, and Laurent Carraro. “ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis”. *Journal of Multivariate Analysis*. 2013. DOI.
- [31] David K Duvenaud, Hannes Nickisch, and Carl Rasmussen. “Additive gaussian processes”. *Advances in neural information processing systems*. 2011. DOI.

- [32] Mohamed Reda El Amri and Amandine Marrel. “More powerful HSIC-based independence tests, extension to space-filling designs and functional data”. *International Journal for Uncertainty Quantification*. 2024. DOI.
- [33] Reda El Amri, Rodolphe Le Riche, Céline Helbert, Christophette Blanchet-Scalliet, and Sébastien Da Veiga. “A sampling criterion for constrained Bayesian optimization with uncertainties”. *The SMAI Journal of computational mathematics*. 2023. DOI.
- [34] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. “Scalable global optimization via local Bayesian optimization”. *Advances in neural information processing systems*. 2019. DOI.
- [35] Noé Fellmann, Christophette Blanchet-Scalliet, Céline Helbert, Adrien Spagnol, and Delphine Sinoquet. “Kernel-based sensitivity analysis for (excursion) sets”. *Technometrics*. 2024. DOI.
- [36] Noé Fellmann, M. Pasquier, C. Blanchet-Scalliet, C. Helbert, A. Spagnol, and D. Sinoquet. “Sensitivity analysis for sets: Application to pollutant concentration maps”. *Quality and Reliability Engineering International*. DOI.
- [37] Jean-Claude Fort, Thierry Klein, and Agnès Lagnoux. “Global sensitivity analysis and Wasserstein spaces”. *SIAM/ASA Journal on Uncertainty Quantification*. 2021. DOI.
- [38] Jean-Claude Fort, Thierry Klein, and Nabil Rachdi. “New sensitivity analysis subordinated to a contrast”. *Communications in Statistics-Theory and Methods*. 2016. DOI.
- [39] Peter I Frazier, Warren B Powell, and Savas Dayanik. “A knowledge-gradient policy for sequential information collection”. *SIAM Journal on Control and Optimization*. 2008. DOI.
- [40] Nando de Freitas and Ziyu Wang. “Bayesian optimization in high dimensions via random embeddings”. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. 2013. DOI.
- [41] Lukas Fröhlich, Edgar Klenke, Julia Vinogradskaya, Christian Daniel, and Melanie Zeilinger. “Noisy-input entropy search for efficient robust Bayesian optimization”. *International Conference on Artificial Intelligence and Statistics*. 2020. DOI.
- [42] Fabrice Gamboa, Pierre Gremaud, Thierry Klein, and Agnès Lagnoux. “Global sensitivity analysis: A novel generation of mighty estimators based on rank statistics”. *Bernoulli*. 2022. DOI.
- [43] Fabrice Gamboa, Alexandre Janon, Thierry Klein, and Agnès Lagnoux. “Sensitivity analysis for multidimensional and functional outputs”. *Electronic Journal of Statistics*. 2014. DOI.
- [44] Fabrice Gamboa, Thierry Klein, and Agnès Lagnoux. “Sensitivity analysis based on Cramér–von Mises distance”. *SIAM/ASA Journal on Uncertainty Quantification*. 2018. DOI.
- [45] Fabrice Gamboa, Thierry Klein, Agnès Lagnoux, and Leonardo Moreno. “Sensitivity analysis in general metric spaces”. *Reliability Engineering and System Safety*. 2021. DOI.
- [46] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. “Bayesian optimization with unknown constraints”. *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. 2014. DOI.
- [47] David Ginsbourger, Olivier Roustant, Dominic Schuhmacher, Nicolas Durrande, and Nicolas Lenz. “On ANOVA decompositions of kernels and Gaussian random field paths”. 2016. DOI.
- [48] Paul Goldberg, Christopher Williams, and Christopher Bishop. “Regression with input-dependent noise: A Gaussian process treatment”. *Advances in neural information processing systems*. 1997. URL.

- [49] Robert B Gramacy, Genetha A Gray, Sébastien Le Digabel, Herbert KH Lee, Pritam Ranjan, Garth Wells, and Stefan M Wild. “Modeling an augmented Lagrangian for blackbox constrained optimization”. *Technometrics*. 2016. DOI.
- [50] Robert B. Gramacy and Herbert K. H. Lee. “Optimization Under Unknown Constraints”. *Bayesian Statistics*. 2010. DOI.
- [51] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. “Measuring statistical dependence with Hilbert-Schmidt norms”. *International conference on algorithmic learning theory*. 2005. DOI.
- [52] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. “A kernel statistical test of independence”. *Advances in neural information processing systems*. 2007. URL.
- [53] Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, and Bernhard Schölkopf. “Kernel methods for measuring independence”. *Journal of Machine Learning Research*. 2005. URL.
- [54] Ryan-Rhys Griffiths, Alexander A Aldrick, Miguel Garcia-Ortega, Vidhi Lalchand, et al. “Achieving robustness to aleatoric uncertainty with heteroscedastic Bayesian optimisation”. *Machine Learning: Science and Technology*. 2021. DOI.
- [55] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. “Constrained Bayesian optimization for automatic chemical design using variational autoencoders”. *Chemical science*. 2020. DOI.
- [56] Mengwu Guo and Jan S. Hesthaven. “Reduced order modeling for nonlinear structural analysis using Gaussian process regression”. *Computer Methods in Applied Mechanics and Engineering*. 2018. DOI.
- [57] Caner Hamarat, Jan H Kwakkel, Erik Pruyt, and Erwin T Loonen. “An exploratory approach for adaptive policymaking by using multi-objective robust optimization”. *Simulation Modelling Practice and Theory*. 2014. DOI.
- [58] Eric Han, Ishank Arora, and Jonathan Scarlett. “High-dimensional Bayesian optimization via tree-structured additive models”. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. DOI.
- [59] Joseph L Hart, Alen Alexanderian, and Pierre A Gremaud. “Efficient computation of Sobol’ indices for stochastic models”. *SIAM Journal on Scientific Computing*. 2017. DOI.
- [60] Philippe Heinrich, Radu S Stoica, and Viet Chi Tran. “Level sets estimation and Vorobev expectation of random compact sets”. *Spatial Statistics*. 2012. DOI.
- [61] Erik Orm Hellsten, Carl Hvarfner, Leonard Papenmeier, and Luigi Nardi. “High-dimensional Bayesian optimization with group testing”. *ICLR 2024*. 2023. DOI.
- [62] Philipp Hennig and Christian J Schuler. “Entropy search for information-efficient global optimization.” *Journal of Machine Learning Research*. 2012. DOI.
- [63] José Miguel Hern, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, Zoubin Ghahramani, et al. “A general framework for constrained Bayesian optimization using information-based search”. *Journal of Machine Learning Research*. 2016. DOI.
- [64] Takeyuki Hida, Hui-Hsiung Kuo, Jürgen Potthoff, and Ludwig Streit. *White noise: an infinite dimensional calculus*. 2013. DOI.
- [65] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. “Computer model calibration using high-dimensional output”. *Journal of the American Statistical Association*. 2008. DOI.

- [66] Wassily Hoeffding. “A class of statistics with asymptotically normal distribution”. *The Annals of Mathematical Statistics*. 1948. DOI.
- [67] Toshimitsu Homma and Andrea Saltelli. “Importance measures in global sensitivity analysis of nonlinear models”. *Reliability Engineering & System Safety*. 1996. DOI.
- [68] Xiaodan Hong, Yongsheng Ding, Lihong Ren, Lei Chen, and Biao Huang. “A weighted heteroscedastic Gaussian Process Modelling via particle swarm optimization”. *Chemometrics and Intelligent Laboratory Systems*. 2018. DOI.
- [69] Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. “Vanilla Bayesian optimization performs great in high dimension”. *ICML 2024*. 2024. DOI.
- [70] Bertrand Iooss and Paul Lemaître. “A review on global sensitivity analysis methods”. *Uncertainty management in simulation-optimization of complex systems: algorithms and applications*. 2015. DOI.
- [71] Bertrand Iooss and Mathieu Ribatet. “Global sensitivity analysis of computer models with functional inputs”. *Reliability Engineering & System Safety*. 2009. DOI.
- [72] A Jahanbakhshi and K Kheiralipour. “Carrot sorting based on shape using image processing, artificial neural network, and support vector machine”. *Journal of Agricultural Machinery*. 2019.
- [73] Alexandre Janon, Thierry Klein, Agnes Lagnoux, Maëlle Nodet, and Clémentine Prieur. “Asymptotic normality and efficiency of two Sobol index estimators”. *ESAIM: Probability and Statistics*. 2014. DOI.
- [74] Janis Janusevskis and Rodolphe Le Riche. “Simultaneous kriging-based estimation and optimization of mean response”. *Journal of Global Optimization*. 2013. DOI.
- [75] Donald R Jones, Matthias Schonlau, and William J Welch. “Efficient global optimization of expensive black-box functions”. *Journal of Global optimization*. 1998. DOI.
- [76] Zdenk Kala. “Quantile-oriented global sensitivity analysis of design resistance”. *Journal of Civil Engineering and Management*. 2019. DOI.
- [77] Kirthivasan Kandasamy, Jeff Schneider, and Barnabás Póczos. “High dimensional Bayesian optimisation and bandits via additive models”. *International conference on machine learning*. 2015. DOI.
- [78] Kristian Kersting, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard. “Most likely heteroscedastic Gaussian process regression”. *Proceedings of the 24th international conference on Machine learning*. 2007. DOI.
- [79] Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. “Distributionally robust Bayesian optimization”. *International Conference on Artificial Intelligence and Statistics*. 2020. DOI.
- [80] Esther van der Knaap. “Shaping the harvest: What determines crop shape?” *Research Outreach*. 2020.
- [81] Sergei Kucherenko, Shufang Song, and Lu Wang. “Quantile based global sensitivity measures”. *Reliability Engineering & System Safety*. 2019. DOI.
- [82] F Kuo, I Sloan, Grzegorz Wasilkowski, and Henryk Woniakowski. “On decompositions of multivariate functions”. *Mathematics of computation*. 2010. DOI.
- [83] Dorota Kurowicka and Roger M Cooke. *Uncertainty analysis with high dimensional dependence modelling*. 2006. DOI.

- [84] Harold J Kushner. “A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise”. *Journal of fluids engineering*. 1964. DOI.
- [85] Guerlain Lambert, Céline Helbert, and Claire Lauvernet. “Quantization-based LHS for dependent inputs: application to sensitivity analysis of environmental models”. *Applied Stochastic Models in Business and Industry*. 2024. DOI.
- [86] Matieyendou Lamboni. “Kernel-based sensitivity indices for any model behavior and screening”. *Socio-Environmental Systems Modelling*. 2023. DOI.
- [87] Matieyendou Lamboni, David Makowski, Simon Lehuger, Benoit Gabrielle, and Hervé Monod. “Multivariate global sensitivity analysis for dynamic crop models”. *Field Crops Research*. 2009. DOI.
- [88] Quoc V Le, Alex J Smola, and Stéphane Canu. “Heteroscedastic Gaussian process regression”. *Proceedings of the 22nd international conference on Machine learning*. 2005. DOI.
- [89] Nicolas Lelièvre, Pierre Beaurepaire, Cécile Mattrand, Nicolas Gayton, and Abdelkader Otsmane. “On the consideration of uncertainty in design: optimization-reliability-robustness”. *Structural and Multidisciplinary Optimization*. 2016. DOI.
- [90] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. “Re-examining linear embeddings for high-dimensional Bayesian optimization”. *Advances in neural information processing systems*. 2020. DOI.
- [91] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. “High dimensional Bayesian optimization using dropout”. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2018. DOI.
- [92] Zhenxing Li, Xiaodan Hong, Kuangrong Hao, Lei Chen, and Biao Huang. “Gaussian process regression with heteroscedastic noisesA machine-learning predictive variance approach”. *Chemical Engineering Research and Design*. 2020. DOI.
- [93] Li Luyi, Lu Zhenzhou, Feng Jun, and Wang Bintuan. “Moment-independent importance measure of basic variable and its state dependent parameter solution”. *Structural Safety*. 2012. DOI.
- [94] Russell Lyons. “Distance covariance in metric spaces”. *The Annals of Probability*. 2013. DOI.
- [95] Mohit Malu, Gautam Dasarathy, and Andreas Spanias. “Bayesian optimization in high-dimensional spaces: A brief survey”. *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE. 2021. DOI.
- [96] Edward Marczewski and Hugo Steinhaus. “On a certain distance of sets and the corresponding distance of functions”. *Colloquium Mathematicum*. 1958. URL.
- [97] Amandine Marrel and Vincent Chabridon. “Statistical developments for target and conditional sensitivity analysis: application on safety studies for nuclear reactor”. *Reliability Engineering and System Safety*. 2021. DOI.
- [98] Amandine Marrel, Bertrand Iooss, Sébastien Da Veiga, and Mathieu Ribatet. “Global sensitivity analysis of stochastic computer models with joint metamodels”. *Statistics and Computing*. 2012. DOI.
- [99] Amandine Marrel, Bertrand Iooss, Michel Jullien, Béatrice Laurent, and Elena Volkova. “Global sensitivity analysis for models with spatially dependent outputs”. *Environmetrics*. 2011. DOI.
- [100] Amandine Marrel, Bertrand Iooss, Beatrice Laurent, and Olivier Roustant. “Calculations of Sobol indices for the Gaussian process metamodel”. *Reliability Engineering & System Safety*. 2009. DOI.

- [101] Amandine Marrel, Nadia Pérot, and Clementine Mottet. “Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators”. *Stochastic Environmental Research and Risk Assessment*. 2014. DOI.
- [102] Georges Matheron. “Principles of geostatistics”. *Economic geology*. 1963. DOI.
- [103] Cameron McPhail, Holger R Maier, Jan H Kwakkel, Matteo Giuliani, Andrea Castelletti, and Seth Westra. “Robustness metrics: How are they calculated, when should they be used and why do they give different results?” *Earth’s Future*. 2018. DOI.
- [104] Jonas Mokus. “On Bayesian methods for seeking the extremum”. *Optimization techniques IFIP technical conference: Novosibirsk, July 1–7, 1974*. Springer. 1975. DOI.
- [105] Ilya Molchanov. *Theory of random sets*. 2005. DOI.
- [106] IS Molchanov. “Empirical estimation of distribution quantiles of random closed sets”. *Theory of Probability & Its Applications*. 1991. DOI.
- [107] Jorge J Moré and Stefan M Wild. “Benchmarking derivative-free optimization algorithms”. *SIAM Journal on Optimization*. 2009. DOI.
- [108] Riccardo Moriconi, KS Sesh Kumar, and Marc Peter Deisenroth. “High-dimensional Bayesian optimization with projections using quantile Gaussian processes”. *Optimization Letters*. 2020. DOI.
- [109] Max D Morris. “Factorial sampling plans for preliminary computational experiments”. *Technometrics*. 1991. DOI.
- [110] Zissimos P Mourelatos and Jinghong Liang. “A methodology for trading-off performance and robustness under uncertainty”. *Journal of mechanical design*. 2006. DOI.
- [111] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. “Kernel mean embedding of distributions: A review and beyond”. *Foundations and Trends in Machine Learning*. 2017. DOI.
- [112] Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. “Local policy search with Bayesian optimization”. *Advances in Neural Information Processing Systems*. 2021. DOI.
- [113] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. “A framework for Bayesian optimization in embedded subspaces”. *International Conference on Machine Learning*. 2019. URL.
- [114] Hung T Nguyen. *An introduction to random sets*. 2006. URL.
- [115] Quan Nguyen, Kaiwen Wu, Jacob Gardner, and Roman Garnett. “Local Bayesian optimization via maximizing probability of descent”. *Advances in neural information processing systems*. 2022. DOI.
- [116] B. Nony, Mélanie Rochoux, Thomas Jaravel, and Didier Lucor. “Reduced-order modeling for parameterized large-eddy simulations of atmospheric pollutant dispersion”. *Stochastic Environmental Research and Risk Assessment*. 2023. DOI.
- [117] ChangYong Oh, Efstratios Gavves, and Max Welling. “BOCK: Bayesian optimization with cylindrical kernels”. *International Conference on Machine Learning*. 2018. DOI.
- [118] Rafael Oliveira, Lionel Ott, and Fabio Ramos. “Bayesian optimisation under uncertain inputs”. *The 22nd international conference on artificial intelligence and statistics*. 2019. DOI.
- [119] Art B Owen. “Randomly permuted (t, m, s)-nets and (t, s)-sequences”. *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. Springer. 1995. DOI.

- [120] Manolis Papadrakakis, Nikos D Lagaros, and Vagelis Plevris. “Design optimization of steel structures considering uncertainties”. *Engineering Structures*. 2005. DOI.
- [121] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. “Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces”. *Advances in Neural Information Processing Systems*. 2022. DOI.
- [122] Chang K Park and Kwang-Il Ahn. “A new approach for measuring uncertainty importance and distributional sensitivity in probabilistic safety assessment”. *Reliability Engineering & System Safety*. 1994. DOI.
- [123] Mathis Pasquier. “Quantification d’incertitudes pour la dispersion turbulente de polluants liés au trafic routier à l’échelle micro-urbaine”. fr. thesis. Aix-Marseille, 2023.
- [124] Mathis Pasquier, Stéphane Jay, Jérôme Jacob, and Pierre Sagaut. “A Lattice-Boltzmann-based modelling chain for traffic-related atmospheric pollutant dispersion at the local urban scale”. *Building and Environment*. 2023. DOI.
- [125] Julien Pelamatti, Rodolphe Le Riche, Céline Helbert, and Christophette Blanchet-Scalliet. “Coupling and selecting constraints in Bayesian optimization under uncertainties”. *Optimization and Engineering*. 2024. DOI.
- [126] Guillaume Perrin and Rodolphe Le Riche. “Bayesian optimization with derivatives acceleration”. *Transactions on Machine Learning Research Journa*. 2023. URL.
- [127] TVE Perrin, Olivier Roustant, Jérémy Rohmer, Olivier Alata, JP Naulin, Déborah Idier, Rodrigo Pedreros, D Moncoulon, and P Tinard. “Functional principal component analysis for global sensitivity analysis of model with spatial output”. *Reliability Engineering & System Safety*. 2021. DOI.
- [128] Francesca Pianosi and Thorsten Wagener. “A simple and efficient method for global sensitivity analysis based on cumulative distribution functions”. *Environmental Modelling & Software*. 2015. DOI.
- [129] Victor Picheny. “A stepwise uncertainty reduction approach to constrained global optimization”. *Artificial intelligence and statistics*. 2014. URL.
- [130] Victor Picheny, Henry Moss, Léonard Torossian, and Nicolas Durrande. “Bayesian quantile and expectile optimisation”. *Uncertainty in Artificial Intelligence*. 2022. DOI.
- [131] Tom Rainforth, Rob Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. “On nesting monte carlo estimators”. *International Conference on Machine Learning*. 2018. DOI.
- [132] Tom Rainforth, Robert Cornish, Hongseok Yang, and Frank Wood. “On the pitfalls of nested Monte Carlo”. *Advances in Approximate Bayesian Inference*. 2016. DOI.
- [133] Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. “High dimensional Bayesian optimization with elastic Gaussian process”. *International conference on machine learning*. 2017.
- [134] Alfred Rappaport. “Sensitivity analysis in decision making”. *The Accounting Review*. 1967. URL.
- [135] Vijay Rathod, Om Prakash Yadav, Ajay Rathore, and Rakesh Jain. “Optimizing reliability-based robust design model using multi-objective genetic algorithm”. *Computers & Industrial Engineering*. 2013.
- [136] Rommel G Regis and Christine A Shoemaker. “Parallel radial basis function methods for the global optimization of expensive functions”. *European journal of operational research*. 2007. DOI.

- [137] Adán Reyes Reyes, André Nasr, Delphine Sinoquet, and Sami Hlioui. “Study on the impact of uncertain design parameters on the performances of a permanent magnet assisted synchronous reluctance motor”. *Sci. Tech. Energ. Transition*. 2024. DOI.
- [138] Adán Reyes-Reyes, André Nasr, Delphine Sinoquet, and Sami Hlioui. “An adapted constrained multi-objective Bayesian optimization under uncertainties: application on a permanent magnet assisted synchronous reluctance motor”. 2024. DOI.
- [139] Mickael Rivier and Pietro Marco Congedo. “Surrogate-assisted bounding-box approach applied to constrained multi-objective optimisation under uncertainty”. *Reliability Engineering & System Safety*. 2022. DOI.
- [140] Walter Rudin. *Real and complex analysis*. 1921.
- [141] Nathalie Saint-Geours, Jean-Stéphane Bailly, Frédéric Grelot, and Christian Lavergne. “Multi-scale spatial sensitivity analysis of a model for economic appraisal of flood risk management policies”. *Environmental Modelling & Software*. 2014. DOI.
- [142] Malek Ben Salem, François Bachoc, Olivier Roustant, Fabrice Gamboa, and Lionel Tomaso. “Gaussian process-based dimension reduction for goal-oriented sequential design”. *SIAM/ASA Journal on uncertainty quantification*. 2019. DOI.
- [143] A Saltelli. *Global Sensitivity Analysis: the Primer*. 2008. DOI.
- [144] Andrea Saltelli. “Making best use of model evaluations to compute sensitivity indices”. *Computer physics communications*. 2002. DOI.
- [145] Gabriel Sarazin, Amandine Marrel, Sébastien Da Veiga, and Vincent Chabridon. “Test d’indépendance basé sur les indices HSIC-ANOVA d’ordre total”. *53èmes Journées de Statistique de la SFdS*. 2022. URL.
- [146] Gabriel Sarazin, Amandine Marrel, Sébastien da Veiga, and Vincent Chabridon. “New insights into the feature maps of Sobolev kernels: application in global sensitivity analysis”. *preprint*. 2023. URL.
- [147] Michael J Sasena, Panos Papalambros, and Pierre Goovaerts. “Exploration of metamodeling sampling criteria for constrained global optimization”. *Engineering optimization*. 2002. DOI.
- [148] Leonard J Savage. “The theory of statistical decision”. *Journal of the American Statistical association*. 1951. DOI.
- [149] Matthias Schonlau, William J Welch, and Donald R Jones. “Global versus local search in constrained optimization of computer models”. *Lecture notes-monograph series*. 1998. URL.
- [150] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. “Equivalence of distance-based and RKHS-based statistics in hypothesis testing”. *The Annals of Statistics*. 2013. DOI.
- [151] John Shawe-Taylor and Nello Cristianini. *Basic kernels and kernel types*. 2004. DOI.
- [152] I M Sobol’. “Sensitivity estimates for nonlinear mathematical models”. *Math. Model. Comput. Exp.*. 1993. URL.
- [153] Il’ya M. Sobol. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. *Mathematics and Computers in Simulation*. 2001. DOI.
- [154] Le Song, Alexander Smola, Arthur Gretton, Karsten Borgwardt, and Justin Bedo. “Supervised feature selection via dependence estimation”. *Proceedings of the 24th international conference on Machine learning*. 2007. DOI.

- [155] Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. “Monte carlo tree search based variable selection for high dimensional bayesian optimization”. *Advances in Neural Information Processing Systems*. 2022. DOI.
- [156] Adrien Spagnol, Rodolphe Le Riche, and Sébastien Da Veiga. “Bayesian optimization in effective dimensions via kernel-based sensitivity indices”. *ICASP13*. 2019. DOI.
- [157] Adrien Spagnol, Rodolphe Le Riche, and Sébastien Da Veiga. “Global sensitivity analysis for optimization with variable selection”. *SIAM/ASA Journal on Uncertainty Quantification*. 2019. DOI.
- [158] Robert C Spear and George M Hornberger. “Eutrophication in peel inletII. Identification of critical uncertainties via generalized sensitivity analysis”. *Water research*. 1980. DOI.
- [159] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. “Gaussian process optimization in the bandit setting: No regret and experimental design”. *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2009. DOI.
- [160] Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R.G. Lanckriet. “Universality, characteristic kernels and RKHS embedding of measures”. *Journal of Machine Learning Research*. 2011. DOI.
- [161] I Steinwart and A Christmann. “Kernels and reproducing kernel hilbert spaces”. *Support Vector Machines*. 2008. DOI.
- [162] Bruno Sudret. “Global sensitivity analysis using polynomial chaos expansions”. *Reliability engineering & system safety*. 2008. DOI.
- [163] Kevin Swersky, David Duvenaud, Jasper Snoek, Frank Hutter, and Michael A Osborne. “Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces”. *NIPS Workshop on Bayesian Optimization in Theory and Practice*. 2014. DOI.
- [164] Zoltán Szabó and Bharath K. Sriperumbudur. “Characteristic and universal tensor product kernels”. *Journal of Machine Learning Research*. 2018. DOI.
- [165] Nima Taheri and Mir Saman Pishvaei. “A regret-based robust optimization model for municipal water distribution network redesign under disruption risks”. *Computers & Chemical Engineering*. 2024. DOI.
- [166] Anh Tran, Mike Eldred, Scott McCann, and Yan Wang. “srMO-BO-3GP: A sequential regularized multi-objective constrained Bayesian optimization for design applications”. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. 2020. DOI.
- [167] Victor Trappier, Élise Arnaud, Arthur Vidard, and Laurent Debreu. “Robust calibration of numerical models based on relative regret”. *Journal of Computational Physics*. 2021. DOI.
- [168] Juan Ungredda and Juergen Branke. “Bayesian optimisation for constrained problems”. *ACM Transactions on Modeling and Computer Simulation*. 2024. DOI.
- [169] Ky Khac Vu, Claudia d’Ambrosio, Youssef Hamadi, and Leo Liberti. “Surrogate-based methods for black-box optimization”. *International Transactions in Operational Research*. 2017. DOI.
- [170] Abraham Wald. “Statistical decision functions”. *The Annals of Mathematical Statistics*. 1949. DOI.
- [171] Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. “Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces”. *ICML 2021*. 2021. DOI.

- [172] Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. “Learning search space partition for black-box optimization using monte carlo tree search”. *Advances in Neural Information Processing Systems*. 2020. DOI.
- [173] Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. “Recent advances in Bayesian optimization”. *ACM Computing Surveys*. 2023. DOI.
- [174] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Freitas. “Bayesian optimization in a billion dimensions via random embeddings”. *Journal of Artificial Intelligence Research*. 2016. DOI.
- [175] Dorina Weichert and Alexander Kister. “Bayesian optimization for min max optimization”. *ICML 2020 Workshop*. 2021. DOI.
- [176] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. 2006. DOI.
- [177] Kaiwen Wu, Kyurae Kim, Roman Garnett, and Jacob Gardner. “The behavior and convergence of local bayesian optimization”. *Advances in neural information processing systems*. 2024. DOI.
- [178] George Wynne and Andrew B Duncan. “A kernel two-sample test for functional data”. *Journal of Machine Learning Research*. 2022. URL.
- [179] Dawei Zhan. “Expected coordinate improvement for high-dimensional Bayesian optimization”. *Swarm and Evolutionary Computation*. 2024. DOI.
- [180] Dawei Zhan and Huanlai Xing. “Expected improvement for expensive optimization: a review”. *Journal of Global Optimization*. 2020. DOI.
- [181] Johanna Ziegel, David Ginsbourger, and Lutz Dümbgen. “Characteristic kernels on Hilbert spaces, Banach spaces, and on sets of measures”. *Bernoulli*. 2024. DOI.
- [182] AG Zilinskas. “Single-step Bayesian search method for an extremum of functions of a single variable”. *Cybernetics*. 1975. DOI.



# Analyse de sensibilité pour l'optimisation bayésienne de boîtes noires coûteuses sous contraintes et sous incertitudes. Analyse de sensibilité d'ensembles aléatoires.

**Résumé :** Cette thèse explore l'utilisation d'outils d'Analyse de Sensibilité (AS) dans le cadre de l'optimisation bayésienne contrainte de boîtes noires coûteuses, en présence d'incertitudes. Ces entrées incertaines complexifient grandement l'optimisation et la rendent propice à des simplifications. Des approches d'AS permettent alors de quantifier l'influence des variables incertaines et de réduire la dimension du problème. L'impact des entrées incertaines est mesuré par une AS des ensembles d'excursion du problème d'optimisation. Pour ce faire, trois approches distinctes d'AS pour des sorties ensemblistes sont développées. La première repose sur des méthodes à noyaux et adapte en particulier le HSIC en introduisant un noyau sur des ensembles. La deuxième propose des indices « à la Sobol », reposant sur des concepts issus de la théorie des ensembles aléatoires. La troisième approche utilise des indices dits universels, conçus pour s'adapter à tout type d'espace de sortie. Ces trois méthodes permettent non seulement de quantifier l'impact des entrées incertaines sur les ensembles d'excursion, mais ouvrent également le champ de l'AS aux modèles ensemblistes. Des étapes d'AS sont ensuite intégrées à un algorithme d'optimisation bayésienne robuste appelé EFISUR. Ces étapes permettent de sélectionner les entrées les plus influentes et ainsi de réduire la dimension à différentes étapes de l'algorithme. Cela entraîne une réduction significative du coût de calcul, rendant l'algorithme plus adapté à la grande dimension.

**Mots clés :** Analyse de sensibilité, optimisation robuste, HSIC, ensembles aléatoires, réduction de dimension, noyau, optimisation bayésienne, krigeage.

*Sensitivity analysis for black-box Bayesian optimization under constraints and in presence of uncertainties. Sensitivity Analysis of random sets.*

**Abstract :** This thesis explores the use of Sensitivity Analysis (SA) tools in the context of constrained Bayesian optimization of expensive black-box functions in the presence of uncertainties. These uncertain inputs greatly complicate the optimization process, making it prone to simplifications. SA approaches then allow to quantify the influence of uncertain variables and reduce the dimension. The impact of uncertain inputs is measured by SA on excursion sets of the optimization problem. To achieve this, three different SA approaches for set-valued outputs are developed. The first is based on kernel methods, in particular adapting HSIC by introducing a kernel on sets. The second proposes Sobol-like indices based on concepts from random set theory. The third approach uses so-called universal indices, which are designed for any type of output space. These three methods not only quantify the impact of uncertain inputs on the excursion sets, but also extend the scope of SA to set-valued models. The SA steps are then integrated into a robust Bayesian optimization algorithm called EFISUR. These steps allow the selection of the most influential inputs, thereby reducing the dimension at different stages of the algorithm. This results in a significant reduction in computational cost, making the algorithm more suitable for high-dimensional problems.

**Keywords :** Sensitivity analysis, robust optimization, HSIC, random sets, dimension reduction, kernel, Bayesian optimization, kriging.

**Image en couverture :** Kernel embedding of random sets : Are tomatoes and carrots shapes related ? Images from [72,80].

