High-Dimensional Bayesian Multi-Objective Optimization

David Gaudrie

N° d'ordre NNT : 2019LYSEM026

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
L'École des Mines de Saint-Étienne

École Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat : Mathématiques Appliquées
Discipline : Science des Données

Soutenue publiquement le 28/10/2019, par :

David Gaudrie

High-Dimensional Bayesian Multi-Objective Optimization

Optimisation bayésienne multi-objectif en haute dimension

Devant le jury composé de :

Villon, Pierre	*Professeur, Université de Technologie de Compiègne, France*	*Président*
Duvigneau, Régis	*Chargé de recherche, INRIA Sophia Antipolis, France*	*Rapporteur*
Emmerich, Michael	*Professeur associé, Leiden University, Pays-Bas*	*Rapporteur*
Brockhoff, Dimo	*Chargé de recherche, INRIA Saclay, France*	*Examinateur*
Villon, Pierre	*Professeur, Université de Technologie de Compiègne, France*	*Examinateur*
Le Riche, Rodolphe	*Directeur de recherche, CNRS LIMOS à EMSE, France*	*Directeur*
Picheny, Victor	*Chargé de recherche, Prowler.io, Royaume-Uni*	*Co-directeur*
Enaux, Benoît	*Ingénieur de recherche, Groupe PSA, France*	*Invité*
Herbert, Vincent	*Ingénieur de recherche, Groupe PSA, France*	*Invité*

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	SCIENCES DES IMAGES ET DES FORMES	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	N. Absi, Maitre de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	Ph. Lalevée, Professeur

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

ABSI	Nabil	MR	Génie industriel	CMP
AUGUSTO	Vincent	CR	Image, Vision, Signal	CIS
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	PR	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
CAMEIRAO	Ana	MA(MDC)	Génie des Procédés	SPIN
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	MR	Sciences des Images et des Formes	SPIN
DEGEORGE	Jean-Michel	MA(MDC)	Génie industriel	Fayol
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
BERGER-DOUCE	Sandrine	PR1	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
DUTERTRE	Jean-Max	MA(MDC)		CMP
EL MRABET	Nadia	MA(MDC)		CMP
FAUCHEU	Jenny	MA(MDC)	Sciences et génie des matériaux	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Sciences de la Terre	SPIN
GAVET	Yann	MA(MDC)	Sciences des Images et des Formes	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GONZALEZ FELIU	Jesus	MA(MDC)	Sciences économiques	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NAVARRO	Laurent	CR		CIS
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR1	Génie des Procédés	SPIN
O CONNOR	Rodney Philip	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PINOLI	Jean Charles	PR0	Sciences des Images et des Formes	SPIN
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROUSSY	Agnès	MA(MDC)	Microélectronique	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
SANAUR	Sébastien	MA(MDC)	Microélectronique	CMP
SERRIS	Eric	IRD		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR0	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

Mise à jour : 30/08/2018

Supervisor
Rodolphe Le Riche, Research Director, CNRS LIMOS at EMSE, France

Co-supervisors
Victor Picheny, Research Fellow, Prowler.io, United-Kingdom
Benoît Enaux, Research Engineer, Groupe PSA, France
Vincent Herbert, Research Engineer, Groupe PSA, France

Jury
Régis Duvigneau, Permanent Researcher, INRIA Sophia Antipolis, France
Michael Emmerich, Associate Professor, Leiden University, The Netherlands
Dimo Brockhoff, Research Fellow, INRIA Saclay, France
Pierre Villon, Professor, Université de Technologie de Compiègne, France

Opponent
David Gaudrie

Contact information

Department of Mathematics and Industrial Engineering
Henri Fayol Institute, Mines Saint-Étienne
158 cours Fauriel, F-42023 Saint-Étienne cedex 2, France

Email address: webmaster@emse.fr
URL: https://www.mines-stetienne.fr/
Telephone: +33 (0)4 77 42 01 23
Fax: +33 (0)4 77 42 00 00

Remerciements

Au cours de ces trois dernières années, à travers mon passage à l'INRA à Auzeville-Tolosane, à l'École des Mines de Saint-Étienne, chez PSA à Vélizy, ou durant les congrès scientifiques auxquels j'ai participé, j'ai eu le plaisir de côtoyer un grand nombre de personnes que je souhaiterais remercier pour leur aide, les discussions, et les conseils prodigués.

En premier lieu, je tiens à remercier mon directeur de thèse, Rodolphe Le Riche, ainsi que mes co-encadrants, Victor Picheny, Benoît Enaux et Vincent Herbert pour leur disponibilité, leur enthousiasme, leurs conseils, leur soutien, et outre leurs qualités scientifiques, pour leurs qualités humaines. Vous m'avez énormément apporté, et c'est grâce à vous que je garderai un si bon souvenir de ces trois années. Ce fut un plaisir de travailler avec vous et pour ceux que je quitte à l'issue de cette aventure, j'espère vous revoir dans l'avenir.

I would like to thank Régis Duvigneau and Michael Emmerich for having accepted to review this manuscript and for their detailed reports and suggestions. I am also grateful to my jury members, Dimo Brockhoff and Pierre Villon.

Je remercie tous les collègues croisés dans les différents environnements dans lesquels j'ai eu le bonheur d'évouler. Ce fut un plaisir de vous rencontrer. En particulier je souhaiterais remercier mes "frères de thèse" Léonard et Adrien, mes co-bureaux stéphanois Jean-Charles et Vincent, Andrés (merci pour tes conseils en LaTeX), Élodie (merci pour tes succulents gateaux), Xavier (merci pour les sorties vélo en Haute-Loire). Merci aux stéphanois, Audrey, Amine, Christine, Damien, Didier, Éric, Isabelle, Jean-François, Mahdi, Mireille, Nicolas, Nihad, Nilou, Olivier, Paolo, Sawsen, Serena pour les moments partagés ensemble, votre aide, et votre accueil. J'ai passé une année très agréable au sein du département GMI, et Saint-Étienne va réellement me manquer. Merci aux collègues de PSA et notamment à Clément, Gentien, Guillaume, Laurent et Sébastien qui m'ont accompagné au cours de la dernière année. Je suis ravi de poursuivre mon chemin à vos côtés.

Merci aussi à mes amis Alexandre, Alexis, Amandine, Baptiste, Corentin, Cyril, Edouard, Frédéric, Hadrien, Hamid, Jean, Léo, Nicolas, Victor, Walid d'avoir partagé ces dernières années, fut-ce autour d'un verre ou sur un vélo. Enfin, je remercie mes parents Astrid et Thierry, mes frères Benjamin et Félix, et ma grand-mère Doris, pour leur soutien durant cette aventure.

Optimisation bayésienne multi-objectif en haute dimension

David Gaudrie

Génie Mathématique et Industriel
Institut Fayol, Mines Saint-Étienne
158 cours Fauriel, F-42023 Saint-Étienne cedex 2, France
david.gaudrie@emse.fr

Introduction

Comme de nombreux produits conçus dans divers domaines de l'ingénierie, un véhicule est constitué d'un grand nombre de systèmes, intéragissant entre eux et avec l'environnement, comme le moteur, les suspensions, le chassis, la forme extérieure du véhicule, les composants électroniques, etc. Afin de garantir la performance, la sécurité et le confort passager, ces systèmes doivent être optimisés. Au cours des dernières décennies, l'augmentation de la puissance de calcul des ordinateurs a permis de substituer de nombreuses expériences physiques par des codes de calcul, et il est ainsi possible de simuler le comportement d'une voiture dans de nombreux domaines tels que la combustion, l'aérodynamique, l'aéro-acoustique, la vibro-acoustique, l'électro-magnétique, réduisant les coûts de prototypage et le temps de conception de nouveaux véhicules de manière drastique. Une reproduction fidèle de la réalité nécessite néanmoins l'usage de simulateurs numériques coûteux en temps de calcul. Dans des applications aérodynamiques ou en combustion, la simulation de l'écoulement autour d'un véhicule ou de l'inflammation au sein du moteur nécessitent la résolution de systèmes d'équations aux dérivées partielles (EDP) hautement non-linéaires. En raison de la complexité des phénomènes physiques mis en jeu (turbulence, spray), les codes de mécanique des fluides numérique nécessitent un maillage du système (c'est-à-dire la forme extérieure de la voiture ou le moteur) à un niveau de résolution très fin. Les méthodes classiques telles que les volumes finis ou les éléments finis considèrent la résolution du système d'EDP en chacun des n_{el} éléments du maillage via des solveurs itératifs. Cette opération est numériquement coûteuse en raison du grand nombre de mailles (plusieurs dizaines de millions), et une seule simulation numérique dure en général entre 12 et 24 heures.

Plus que la reproduction fidèle du comportement de la voiture, c'est l'optimisation des systèmes à travers la simulation numérique qui est recherchée par les ingénieurs. Dans les applications industrielles, l'optimisation vise à proposer de nouveaux designs capables de combiner le respect de normes toujours plus exigeantes et l'attractivité du produit. Elle a également pour but d'être un outil d'aide à la décision en proposant des solutions et des règles de conception de systèmes complexes que l'humain ne peut pas intuiter. L'optimisation de systèmes repose généralement sur la Conception Assistée par Ordinateur (CAO) : les designs considérés sont restreints à une classe de systèmes paramétrés par d variables $x_1, \dots, x_d$, $\mathbf{x} \in X$, déterminant la forme associée, $\Omega_{\mathbf{x}}$. Ces variables correspondent à des caractéristiques diverses du système : elles incluent des descriptions générales telles que des tailles (hauteur, largeur, longueur du système) ainsi que des détails (rayons, angles, ajustements locaux, ...). Selon le degré de précision de la CAO, le nombre de tels paramètres peut être grand, $d \gtrsim 50$. Dans le but d'obtenir la configuration optimale du système, les paramètres CAO $\mathbf{x}^*$ qui minimisent une fonction objectif $f(\mathbf{x})$ sont recherchés.

En raison du coût associé à chaque simulation numérique, ces optimisations sont budgetées : en fonction du planning projet et de la durée d'une simulation, un nombre prescrit d'évaluations de la fonction objectif, un *budget* (typiquement une centaine de simulations), est autorisé. Des méthodes d'optimisation largement répandues telles que les algorithmes évolutionnaires (Deb, 2001; Eiben and Smith, 2003; Michalewicz, 2013) nécessitent un grand nombre d'évaluations de fonction avant d'atteindre l'optimum. Les méthodes de gradient (Liu and Nocedal, 1989) sont plus rapides mais requièrent $\nabla f(\mathbf{x})$ qui n'est en général pas connu, et surtout, ces méthodes ne convergent que vers un optimum local dont la qualité est dépendante du point d'initialisation. Des techniques de multistart permettent de se rapprocher de l'optimum global au prix d'un plus grand nombre d'évaluations de la fonction objectif. Ces deux types de méthodes ne sont donc pas adaptés aux fonctions "boîtes noires coûteuses" que nous considérons, et pour lesquelles le lien entre un système $\mathbf{x}$ et sa réponse $y = f(\mathbf{x})$ est uniquement accessible à travers une simulation numérique. Une approximation de $\nabla f(\mathbf{x})$ par différences finies nécessiterait d évaluations supplémentaires et n'est pas envisagée. Optimiser une fonction basse fidélité $\tilde{f}(\mathbf{x})$ plus rapide à évaluer n'est pas une solution en général dans la mesure où des phénomènes physiques critiques pourraient être omis. Dans cette thèse, nous nous intéressons à des algorithmes d'optimisation ne nécessitant qu'un nombre restreint d'évaluations de $f(\cdot)$ pour proposer une solution. De telles méthodes (Jones, 2001) reposent sur l'utilisation d'un modèle de substitution (ou méta-modèle, Loshchilov et al., 2010; Rasmussen and Williams, 2006; Sudret, 2008) du code de calcul, peu coûteux à évaluer, construit à partir des simulations passées et utilisé pour déterminer itérativement une séquence de designs $\mathbf{x}^{(n+1)}, \mathbf{x}^{(n+2)}, \dots, \mathbf{x}^{(budget)}$ à évaluer pour diriger rapidement la recherche vers $\mathbf{x}^*$. L'efficacité des méthodes assistées par méta-modèles a été démontrée pour approcher l'optimum en un nombre restreint d'itérations sur un vaste éventail d'applications (Forrester and Keane, 2009; Shahriari et al., 2015).

La performance de telles méthodes se détériore néanmoins quand le nombre de variables considérées $x_1, \dots, x_d$ est grand. Ce phénomène, connu comme le *fléau de la dimension* (Bellman, 1961) rend l'optimisation de formes paramétrées difficile. De plus, bien qu'ils soient intuitifs pour un designer pour automatiser la génération de formes, les paramètres CAO x_i n'ont pas vocation à satisfaire des propriétés mathématiques, et ne sont pas forcément la représentation la plus pertinente de l'objet sous-jacent. Il existe souvent des corrélations entre des paires ou des groupes de x_i, et certaines variables décrivent la forme de manière globale au contraire d'autres qui raffinent le système localement. Une paramétrisation reliée à la forme entière, et dont l'impact sur Ω soit quantifiable serait plus judicieuse que celle reposant sur des modifications ponctuelles du système.

Les ingénieurs souhaitent souvent optimiser leurs systèmes vis-à-vis de multiples objectifs antagonistes, et peuvent être amenés à spécifier des contraintes de faisabilité. Au lieu d'un problème mono-objectif, ce sont des solutions optimales d'un problème multi-objectif contraint qui sont cherchées, c'est-à-dire le front/l'ensemble de Pareto. Les approches avec méta-modèles ont été étendues pour répondre à cette problématique (Binois, 2015; Emmerich et al., 2006; Ponweiser et al., 2008) et pour converger rapidement vers le front de Pareto. Il n'est cependant pas possible d'approcher précisément le front en un nombre restreint d'évaluations de fonctions, notamment lorsque plus de 2 ou 3 objectifs sont considérés, en raison de l'augmentation exponentielle de la taille du front avec le nombre d'objectifs. De plus, une grande partie du front possède peu d'intérêt en pratique, et les méthodes avec méta-modèles deviennent elles aussi plus coûteuses lorsque davantage d'objectifs sont considérés. Au lieu de tenter de découvrir (le plus souvent en vain, en raison du budget limité) la totalité du front sans tenir compte des ressources disponibles, il est préférable d'améliorer la convergence vers des solutions réellement pertinentes dans la limite du budget imparti.

Résumé par chapitres

Dans le Chapitre 2, les notions essentielles et les techniques de l'état de l'art en processus gaussiens, optimisation bayésienne et optimisation multi-objectif, utilisées tout au long de cette thèse, sont exposées.

Dans le Chapitre 3, le cas test MetaNACA est présenté. Il s'agit d'un problème multi-objectif, de dimension d et à nombre d'objectifs m variables. Le MetaNACA est un émulateur d'un simulateur numérique de l'écoulement aérodynamique autour d'un profil d'aile d'avion NACA. Il a été construit en alliant des techniques de méta-modélisation (processus gaussiens) à des méthodes d'enrichissement séquentiel : un grand nombre de profils NACA (≈ 1000) ont été évalués par le simulateur, lesquels ont été utilisés pour construire un méta-modèle suffisamment précis pour remplacer la simulation. L'intérêt du MetaNACA réside dans son temps d'exécution négligeable, ainsi que dans les phénomènes physiques qu'il modélise : ce cas test est un représentant typique des problèmes d'optimisation de fonctions physiques réellement rencontrés et pour lesquels nous souhaitons développer des optimiseurs multi-objectifs. Par conséquent, il est préférable de s'étalonner sur ce problème pour tester et comparer les méthodes développées plutôt que sur des fonctions analytiques (Zitzler et al., 2000), peu représentatives de problèmes réels (typiquement trop multi-modales). Les objectifs à optimiser sont la traînée et la portance de ce profil, à angle d'incidence $\alpha_I = 0°$ ou

$\alpha_I = 8°$. Des problèmes avec $m = 2$, 3 ou 4 objectifs peuvent ainsi être formulés. La géométrie d'un profil NACA est déterminée par $d = 3$ paramètres, $\mathbf{x} = (M, P, T)^\top$. Pour traiter des problèmes en plus grande dimension, nous avons créé des géométries avec $d = 8$ et $d = 22$ paramètres de forme, en ajoutant respectivement 5 ou 19 bosses de hauteur L_i le long de l'intrados et de l'extrados. Dans le Chapitre 3, le MetaNACA est utilisé pour comparer des fonctions d'acquisition de la littérature (Emmerich et al., 2006; Picheny, 2015; Ponweiser et al., 2008; Svenson and Santner, 2010), étudier l'influence de la dimension et du nombre d'objectifs sur les résultats de l'optimisation, ainsi que la répartition du *budget* d'optimisation entre les n évaluations constituant le plan d'expériences initial et les p évaluations pilotées par la fonction d'acquisition. Il a été observé que les meilleurs résultats en termes de convergence vers le front de Pareto sont obtenus en utilisant la majorité du *budget* lors de la phase d'ajouts séquentiels. Au cours des chapitres suivants, le MetaNACA est employé pour analyser et comparer les diverses méthodes et algorithmes d'optimisation multi-objectif et de réduction de dimension développés (d'autres cas tests le compléteront).

Le Chapitre 4 est consacré au développement d'une nouvelle méthode d'optimisation multi-objectif assistée par méta-modèles. Dans le cas de budgets d'optimisation fortement restreints et/ou d'un grand nombre d'objectifs, il n'est pas possible de converger précisément vers le front de Pareto entier. La taille de ce dernier augmente en effet exponentiellement avec le nombre d'objectifs. De plus, toutes les solutions Pareto-optimales n'intéressent généralement pas le décideur. Si ce dernier est capable d'exprimer ses préférences à travers un point de référence $\mathbf{R}$, employé comme une cible à atteindre ou à dépasser, l'algorithme R-EHI vise à prioriser la partie du front de Pareto correspondante. Si le décideur n'est pas en mesure d'exprimer ses préférences, l'algorithme C-EHI, dans lequel le centre du front de Pareto est la préférence par défaut, est utilisé. R-EHI et C-EHI reposent sur une nouvelle lecture du critère d'acquisition Expected Hypervolume Improvement (EHI), un critère qui respecte la Pareto-optimalité. Le centre du front de Pareto est aussi l'une des contributions de cette thèse. Il s'agit d'un point d'équilibre entre les m objectifs. Sa définition, ses bonnes propriétés et des moyens robustes pour l'estimer via le méta-modèle sont exposés dans le Chapitre 4. C-EHI et R-EHI se concentrent dans un premier temps sur l'atteinte d'un point bien précis du front de Pareto. Il est possible que ce dernier soit atteint avant épuisement du *budget*. Lorsque tel est le cas, cibler exclusivement ce point est une perte de ressources ; il vaut mieux se concentrer sur une zone plus large du front, mais pas trop vaste pour pouvoir être découverte de manière précise au cours des b itérations restantes. Pour cela, dans le Chapitre 4, un critère de convergence locale vers le front de Pareto est défini. Il repose sur l'incertitude du front de Pareto dans la zone d'intérêt. Lorsque ce dernier est déclenché, C-EHI et R-EHI déterminent une nouvelle zone d'intérêt plus large, dans laquelle une bonne convergence en fin d'optimisation est prédite. La détermination de cette région se fait en anticipant virtuellement le comportement de l'algorithme au cours des b itérations restantes, pour diverses parties du front de Pareto de taille croissante. La région la plus grande pour laquelle suffisamment peu d'incertitude est prédite en fin d'optimisation est choisie pour finalement être ciblée au cours des évaluations restantes. C-EHI et R-EHI proposent ainsi une approximation du front de Pareto s'adaptant au *budget* disponible, tout en insistant sur les solutions les plus pertinentes (fournies par l'utilisateur, ou bien le centre du front de Pareto par défaut). Par rapport à des méthodes bayésiennes multi-objectif classiques (par exemple EHI, Emmerich et al., 2006), ou des algorithmes évolutionnaires (NSGA-II, Deb et al., 2002), une meilleure convergence vers la partie critique du front et une atteinte plus rapide de solutions désirées sont obtenues dans le cas de budgets limités.

Dans le Chapitre 5, des extensions de C-EHI et R-EHI, renforçant leur intérêt pratique, sont proposées. Dans un premier temps, les fonctions d'acquisition utilisées par ces algorithmes pour décider des paramétrisations à évaluer sont étendues à des groupes de points : au lieu de proposer un point $\mathbf{x}^{(t+1)}$, q designs $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}$ prometteurs sont retournés. Cela présente un grand intérêt dans le cas où q simulateurs ou noeuds de calculs d'un cluster sont disponibles puisque des lots de q designs peuvent être évalués simultanément. Pour un même temps horloge, q fois plus de simulations numériques pourront être effectuées, améliorant la convergence vers le front de Pareto et le temps de restitution de solutions désirées par l'utilisateur. Dans ce chapitre, les algorithmes sont également adaptés pour la prise en compte de contraintes d'optimisation. Les étapes de C-EHI et R-EHI sont modifiées en conséquence. La propriété de ciblage est également étudiée dans un autre contexte : celui de problèmes fortement contraints, dans lesquels il est difficile de trouver une paramétrisation satisfaisant les m_c contraintes. D'autres modifications et utilisations possibles de C-EHI et de R-EHI sont discutées pour améliorer le temps d'exécution de l'algorithme, ou pour utiliser les mécanismes de ciblage dans d'autres situations.

Enfin, dans le Chapitre 6, motivés par le problème d'optimisation de formes à grand nombre de paramètres, nous proposons une méthode permettant d'éviter le fléau de la dimension (Bellman, 1961) quand d est grand. Au lieu de considérer les paramètres CAO $\mathbf{x}$ qui définissent la forme associée, $\Omega_{\mathbf{x}}$, une base de données de formes est analysée de manière non-supervisée, par Analyse en Composantes Principales (ACP) après plongement (non-linéaire) dans un espace de formes en très grande dimension. Cela permet de faire émerger les motifs les plus fréquents, les "formes propres" du design. Au contraire des paramètres CAO, ces dernières ont un impact global sur la forme. Chaque design $\mathbf{x}$ peut être décomposé dans cette nouvelle base via ses coefficients de reconstruction de forme, $\boldsymbol{\alpha}$. Les formes propres sont hiérarchisées : elles sont triées par ordre décroissant d'importance, et il est ainsi aisé de n'en conserver qu'un nombre restreint pour effectuer l'optimisation bayésienne en dimension réduite. Cela étant, l'ACP classe les formes propres selon leur importance d'un point de vue géométrique. Il n'est néanmoins pas garanti que les plus grandes variations géométriques d'une forme soient majoritairement responsables des fluctuations de la sortie y. Une procédure de maximum de vraisemblance pénalisée vise à déterminer les quelques formes propres $\mathbf{v}^j$ les plus influentes sur la sortie. Ces variables actives, $\boldsymbol{\alpha}^a$, sont priorisés au sein d'un processus gaussien additif qui prend également en compte les variables non-sélectionnées, $\boldsymbol{\alpha}^{\overline{a}}$, de manière plus grossière

: les variables actives sont décrites par un processus anisotrope, celles inactives par un processus isotrope. Cette structuration des variables permet à la fois de réduire la dimension en considérant l'impact de $\boldsymbol{\alpha}^{\overline{a}}$ sur y comme un effet résiduel, et de conduire l'optimisation bayésienne en dimension réduite à travers la priorisation des variables actives. Cette démarche permet d'obtenir des designs optimaux bien plus rapidement que par l'approche classique reposant sur l'optimisation des paramètres CAO, $\mathbf{x}$.

Le cas des formes comportant plusieurs éléments est également étudié dans ce chapitre. Des méthodes permettant de prendre en compte les symétries apparaissant dans de tels problèmes sont proposées et analysées. Elles reposent sur la modification des formes propres ou sur la propagation de la symétrie dans l'espace des $\boldsymbol{\alpha}$, et permettent d'accroître la précision du méta-modèle.

Conclusions

Les développements poursuivis dans ce manuscrit ont visé à étendre les méthodes existantes en optimisation multi-objectif bayésienne pour mieux contrôler la qualité de la convergence lorsque le nombre d'appels aux fonctions coûts est fortement restreint. Tout d'abord, les méthodes de l'état de l'art en optimisation multi-objectif assistée par méta-modèles (Emmerich et al., 2006; Jones et al., 1998) ont été améliorées. Une nouvelle façon d'appréhender le problème multi-objectif a été proposée au travers des algorithmes C-EHI et R-EHI. Il s'agit de prioriser la découverte de solutions optimales pertinentes, puis, en fonction du budget à disposition au moment de la convergence locale vers le front de Pareto, d'étendre la zone de recherche afin de proposer un éventail de solutions plus large, mais toujours pertinent pour le décideur. Lorsque ce dernier n'a pas ou ne sait pas exprimer ses préférences, le centre du front de Pareto, défini dans cette thèse, est la région d'intérêt par défaut. Il s'agit d'un point du front particulièrement intéressant dans la mesure où il équilibre les objectifs, ce qui est l'essence même d'un problème multi-objectif. Les techniques employées dans ces algorithmes ont été complétées pour les rendre plus pratiques : une version multi-point de la fonction d'acquisition permet d'évaluer plusieurs designs en parallèle ; elle a également été étendue pour traiter des problèmes multi-objectifs avec contraintes.

Dans un second temps, inspirée par l'optimisation de formes paramétrées, une méthode permettant de contourner le fléau de la dimension a été développée. Elle se base sur la construction d'un espace de variables auxiliaires plus à même de décrire les formes globalement, et dont un nombre limité impacte la réponse du simulateur. Ces variables sont priorisées au sein d'un processus gaussien additif qui ne néglige pas totalement les variables moins influentes, et qui est utilisé au cours d'une procédure d'optimisation bayésienne en dimension réduite.

Dans l'esprit de la SIR (Li, 1991) ou de la PLS (Frank and Friedman, 1993), la construction d'une base orthonormale en dimension réduite dépendante des m objectifs faciliterait la procédure de sélection des variables actives et pourrait améliorer la méta-modélisation. L'utilisation de méta-modèles plus flexibles tels que les processus de Student (Shah et al., 2014) ou les récents Deep Gaussian Processes (Bui et al., 2016; Damianou and Lawrence, 2013) est également une perspective attrayante pour inférer le lien entre la sortie y et les variables d'entrée $\mathbf{x}$ (ou $\phi(\mathbf{x})$). Enfin, des paires ou groupes de fonctions à optimiser étant souvent corrélés (Shah and Ghahramani, 2016), la réduction du nombre d'objectifs (Brockhoff and Zitzler, 2006b; Deb and Saxena, 2006) ou la méta-modélisation jointe de ceux-ci (Fricker et al., 2013; Svenson, 2011) constituent des directions de recherche importantes pour l'amélioration des solutions restituées.

Table des Matières

Contents

List of Figures

List of Tables

Scientific Contributions

Results throughout this thesis are based on diverse scientific contributions including publications in international journals, proceedings in international conferences, preprints and conferences. The scientific contributions are listed below.

Publications, conference proceedings, submitted papers and pre-prints

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. (2018). Budgeted Multi-Objective Optimization with a Focus on the Central Part of the Pareto Front - Extended Version. `arXiv` pre-print: https://arxiv.org/abs/1809.10482.

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. (2018). Targeting Well-Balanced Solutions in Multi-Objective Bayesian Optimization Under a Restricted Budget. *Proceedings of the 12th International Conference on Learning and Intelligent Optimization, LION 12 (2018)*, Lecture Notes in Computer Science, vol 11353, Springer. DOI 10.1007/978-3-030-05348-2_15.

- Gaudrie, D., Le Riche, R., Picheny, V., Enaux B. and Herbert V. (2019). Targeting solutions in Bayesian Multi-Objective Optimization: Sequential and Batch Versions. *Annals of Mathematics and Artificial Intelligence*. DOI 10.1007/s10472-019-09644-8.

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. (2019). From CAD to Eigenshapes for Surrogate-Based Optimization. *Proceedings of the 13th World Congress of Structural and Multidisciplinary Optimization, WCSMO 13 (2019).*

- Gaudrie, D., Le Riche, R., Picheny, V., Enaux B. and Herbert V. (2020). Modeling and Optimization with Gaussian Processes in Reduced Eigenbases. Submitted to *Structural and Multidisciplinary Optimization* (under minor revision). `arXiv` preprint (extended version): https://arxiv.org/abs/1908.11272.

Conferences and workshops

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Targeting Well-Balanced Solutions in Multi-Objective Bayesian Optimization Under a Restricted Budget, in PGMO Days 2017, Saclay, France, Nov. 13-14, 2017.

- (Poster) Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Targeting Well-Balanced Solutions in Multi-Objective Bayesian Optimization Under a Restricted Budget, in Journées de la Chaire Oquaido 2017, Orléans, France, Nov. 23-24, 2017.

- (Poster) Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Targeting Well-Balanced Solutions in Multi-Objective Bayesian Optimization Under a Restricted Budget, in MASCOT-NUM Annual Conference 2018, Nantes, France, Mar. 21-23, 2018.

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Targeting Well-Balanced Solutions in Multi-Objective Bayesian Optimization Under a Restricted Budget, in 12th International Conference on Learning and Intelligent Optimization (LION 12), Kalamata, Greece, June 11-15, 2018.

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Achievable Goals in Bayesian Multi-Objective Optimization, in Journées du GdR MOA 2018, Pau, France, Oct. 17-19, 2018.

- (Poster) Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Dimension Reduction for Shape Optimization, in Journées de la Chaire Oquaido 2018, Cadarache, France, Nov. 22-23, 2018.

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Dimension Reduction for the Bayesian Optimization of Shapes, in MASCOT-NUM Annual Conference 2019, Rueil-Malmaison, France, Mar. 18-20, 2019 (best oral presentation award).

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. From CAD to Eigenshapes for Surrogate-based Optimization, in 13th World Congress of Structural and Multidisciplinary Optimization (WCSMO 13), Beijing, China, May 20-24, 2019.

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Budgeted Bayesian Multiobjective Optimization, in Journées de la Chaire Oquaido 2019, Saint-Étienne, France, November 27-28, 2019 (accepted for presentation).

- Gaudrie D., Le Riche R., Picheny V., Enaux B. and Herbert V. Bayesian Optimization in Reduced Eigenbases, in PGMO Days 2019, Saclay, France, December 3-4, 2019 (accepted for presentation).

Notations

The notations, symbols and acronyms employed throughout this thesis are introduced in this section. Vectors and matrices are given in bold symbols, and the $(\cdot)$ notation indicates the handled object is a function.

$\preceq$	Pareto domination.
$\mathbf{0}_n$	n-dimensional vector of zeros.
$\mathbf{1}_n$	n-dimensional vector of ones.
$\mathbb{1}_A$	Indicator function of the event A.
a	General purpose threshold.
$\mathcal{A}$	Manifold of $\boldsymbol{\alpha}$'s for which $\exists \mathbf{x} \in X$: $\mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) = \boldsymbol{\alpha}$.
$\mathcal{A}_N$	Empirical manifold of $\boldsymbol{\alpha}$'s which are the coordinates of the $\phi(\mathbf{x}^{(i)})$'s in the eigenbasis.
$\boldsymbol{\alpha}$	Coordinates of a design in the eigenshape basis.
$\boldsymbol{\alpha}^a$	Active components of $\boldsymbol{\alpha}$.
$\boldsymbol{\alpha}^{\overline{a}}$	Inactive components of $\boldsymbol{\alpha}$.
$\boldsymbol{\alpha}^{(i)}$	i-th design (among the N designs in $\boldsymbol{\Phi}$, or the t in $\mathcal{D}_t$) in the eigenvector basis.
$\boldsymbol{\alpha}^{(1:t)}$	Set of t designs in the eigenshape basis, $\boldsymbol{\alpha}^{(1:t)} = \{\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(t)}\}$.
$\boldsymbol{\alpha}_{1:\delta}$	δ first components of $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}_{1:\delta} = (\alpha_1, \dots, \alpha_\delta)^\top$.
α_j	j-th eigenbasis component of $\boldsymbol{\alpha}$.
α_I	Angle of incidence of the NACA airfoil.
$\overline{\alpha}$	Coordinate along a random line in the $\boldsymbol{\alpha}^{\overline{a}}$ space.
α_{LCB}	Lower confidence bound optimization parameter.
b	Remaining budget once the local convergence criterion is triggered.
β	Constant mean function of the GP.
$\widehat{\beta}$	GLS estimate of β.
$budget$	Number of allowed function evaluations during an optimization (with single-point infill criterion), $budget = n + p$.
$c(\cdot, \cdot)$	Conditional covariance function, $c(\mathbf{x}, \mathbf{x}) = s^2(\mathbf{x})$.
C	Number of candidate reference points.
$\mathbf{C}$	Center of the Pareto front.
$\widehat{\mathbf{C}}$	Estimated center of the Empirical Pareto front.
$\mathbf{C}_{\boldsymbol{\Phi}}$	Empirical covariance matrix of $\boldsymbol{\Phi}$.

d	Number of (CAD) parameters.
δ	Number of chosen/selected components for dimension reduction.
D	Dimension of the high-dimensional shape representation.
$\mathcal{D}_t$	Set of t input/output pairs, $\mathcal{D}_t = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\} = \{\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\}$.
e	Number of elements in a multi-element shape.
$f(\cdot)$	Objective function.
$\widehat{f}(\cdot)$	Surrogate to $f(\cdot)$.
$f_j(\cdot)$	j-th objective function in a multi-objective problem, $j = 1, \dots, m$.
$\mathbf{f}(\cdot)$	Multi-objective objective function, $\mathbf{f}(\cdot) = (f_1(\cdot), \dots, f_m(\cdot))^\top$.
$f_{\min}$	Smallest observed objective function value.
$\mathcal{F}_X$	Feasible part of X.
$g_j(\cdot)$	j-th constraint, $j = 1, \dots, m_c$.
$\mathbf{g}(\cdot)$	Vector of m_c constraints.
$G_j(\cdot)$	Gaussian Process model for the constraint $g_j(\cdot)$.
$\mathbf{G}(\cdot)$	(Independent) Gaussian Process model for the m_c constraints.
$\mathbf{\Gamma}$	Conditional covariance matrix.
$\mathcal{H}_k$	Reproducing Kernel Hilbert Space of $k(\cdot, \cdot)$.
i, j, k, l	Indices. Most often, i refers to a design while j refers to a parameter or to an objective function. k is employed for number the n_{sim} or N_{MC} simulated GPs.
$I(\cdot)$	Improvement function.
I_H	Hypervolume indicator.
$\mathcal{I}_{\mathbf{R}}$	m-dimensional space dominated by $\mathbf{R}$, $\mathcal{I}_{\mathbf{R}} = \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y} \preceq \mathbf{R}\}$.
$\mathbf{I}$	Ideal point of $\mathcal{P}_{\mathcal{Y}}$, $\mathbf{I} \in \mathbb{R}^m$.
$\widehat{\mathbf{I}}$	Estimated Ideal point.
$\overline{\mathbf{I}}$	Empirical Ideal point.
$k(\cdot, \cdot)$	Gaussian Process kernel.
$k_\phi(\cdot, \cdot)$	Kernel PCA kernel.
$\mathbf{K}$	Kernel matrix whose entries are $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.
$l(\cdot)$	Log-likelihood function.
λ	Penalized (log-) likelihood parameter.
λ_j	j-th PCA eigenvalue.
L	Likelihood function.
$\mathcal{L}$	Ideal-Nadir line.
$\mathcal{L}'$	Broken line joining $\mathbf{I}$, the user-provided $\mathbf{R}$ and $\mathbf{N}$.
$\widehat{\mathcal{L}}$	Estimation of $\mathcal{L}$.
m	Number of objectives.
$m(\cdot)$	GP mean function.
m_c	Number of constraints.
$\mathbf{M}$	Maximum objective value in Y, $\mathbf{M} \in \mathbb{R}^m$.
$\overline{\mathbf{M}}$	Empirical maximum objective value.

n	Number of designs in the initial DoE $\mathcal{D}_n$.
n_{sim}	Number of Gaussian Process simulations.
N	Number of shapes in the $\mathbf{\Phi}$ database.
N_{DOE}	Number of designs to build the MetaNACA.
N_{MC}	Number of Monte Carlo samples for estimating the q-EHI and q-mEI criteria.
$\mathbf{N}$	Nadir point of $\mathcal{P_Y}$, $\mathbf{N} \in \mathbb{R}^m$.
$\widehat{\mathbf{N}}$	Estimated Nadir point.
$\overline{\mathbf{N}}$	Empirical Nadir point.
$\mathcal{N}$	Standard normal distribution.
$\boldsymbol{\nu^j}$	j-th extreme point.
$\overline{\boldsymbol{\nu^j}}$	j-th extreme point of the empirical Pareto front.
$\Omega_{\mathbf{x}}$	Shape induced by the $\mathbf{x}$ parameterization.
p	Number of allowed calls to the infill criterion. With single-point criteria, p is the number of additional designs evaluated after the initial DoE.
$p(\cdot)$	Probability of domination of $\mathbf{y} \in Y$.
$pl(\cdot)$	Penalized log-likelihood.
$\mathcal{P_X}$	Pareto set.
$\mathcal{P_Y}$	Pareto front.
$\widehat{\mathcal{P_Y}}$	Empirical Pareto front (from the set $\mathbf{y}^{(1:t)}$).
$\widetilde{\mathcal{P_Y}}$	Simulated Pareto front.
$\Pi_A(\cdot)$	Projection operator onto A.
$\varphi_{\mathcal{N}}(\cdot)$	Density function of the standard Gaussian $\mathcal{N}(0, 1)$
$\phi_{\mathcal{N}}(\cdot)$	Cumulative distribution function of the standard Gaussian $\mathcal{N}(0, 1)$.
$\phi(\cdot)$	High-dimensional shape mapping, $\phi : X \mapsto \Phi$
Φ	Space of shape discretizations, $\Phi \subset \mathbb{R}^D$.
$\boldsymbol{\phi}$	High-dimensional shape representation of one design ($\boldsymbol{\phi} \in \mathbb{R}^D$).
$\overline{\boldsymbol{\phi}}$	Mean shape in the $\mathbf{\Phi}$ database.
$\mathbf{\Phi}$	Shape database ($N \times D$ matrix whose i-th row is $\phi(\mathbf{x}^{(i)})$).
q	Number of designs returned by multi-point infill criteria (batch size).
$\mathbf{R}$	EHI or mEI's reference point. $\mathbf{R}$ is also used as the user-provided reference point in R-EHI.
$\mathbf{R}_{\boldsymbol{\theta}}$	Correlation matrix released from σ^2, $\sigma^2 \mathbf{R}_{\boldsymbol{\theta}} = \mathbf{K}$.
$\widehat{\mathbf{R}}$	Updated reference point.
$\widetilde{\mathbf{R}}$	Attainable reference point (on $\mathcal{P_Y}$).
s	Number of designs $\mathbf{x}$ where Gaussian Process simulations are performed.
$s^2(\cdot)$	Kriging variance, $s^2(\cdot) : X \to \mathbb{R}^+$. $s(\cdot) = \sqrt{s^2(\cdot)}$ is alternatively employed.
$s_j^2(\cdot)$	Kriging variance of the metamodel of the j-th objective.
$\mathbf{s}^2(\cdot)$	Kriging variance of the m objective functions, $\mathbf{s}^2(\cdot) = (s_1^2(\cdot), \ldots, s_m^2(\cdot))^\top$.

σ Permutation in Φ such that the discretizations $\sigma \circ \phi(\mathbf{x})$ and $\phi(\mathbf{x})$ correspond to the same $\Omega_{\mathbf{x}}$. Additionally, $\sigma \circ \phi(\mathbf{x}) = \phi(\tau \circ \mathbf{x})$ for one specific τ.

σ^2 Variance parameter of the GP.

$\widehat{\sigma^2}$ Estimated variance of the GP.

t Current number of evaluated designs during a Bayesian optimization, $n \leq t \leq budget$.

τ Permutation of a design under which the shape is invariant, $\Omega_{\mathbf{x}} = \Omega_{\tau \circ \mathbf{x}}$.

τ_ε GP nugget effect.

$\boldsymbol{\theta}$ Vector of GP length-scales, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)$

ϑ GP hyperparameters including $\boldsymbol{\theta}$ and $\widehat{\sigma^2}$.

$U(\cdot)$ Uncertainty measure.

$\mathbf{v}^j$ j-th eigenvector of the covariance matrix of $\boldsymbol{\Phi}$. $\mathbf{v}^j \in \mathbb{R}^D$, $j = 1, \dots, D$.

$\mathbf{V}$ $D \times D$ matrix whose columns are the $\mathbf{v}^j$'s.

$\mathcal{V}$ Eigenvector basis.

$\mathbf{x}$ Design vector in the space of parameters, $\mathbf{x} \in X$.

$\mathbf{x}^{(i)}$ i-th observed value.

x_j j-th parameter of $\mathbf{x}$.

$\mathbf{x}^{(1:t)}$ Set of t designs, $\mathbf{x}^{(1:t)} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}\} \subset X$.

$\boldsymbol{\xi}^j$ j-th extreme design, $\boldsymbol{\xi}^j \in X$ and $f_j(\boldsymbol{\xi}^j) = N_j$.

X Original search space (of CAD parameters), $X \subset \mathbb{R}^d$.

$\mathbb{X}$ Set of designs, $\mathbb{X} \subset X$.

y Scalar output of $f(\cdot)$.

$y^{(i)}$ i-th observed (scalar) output.

y_j Scalar output of $f_j(\cdot)$.

$y^{(1:t)}$ Set of t observed scalar outputs, $y^{(1:t)} = \{y^{(1)}, \dots, y^{(t)}\}$.

$\mathbf{y}$ m-dimensional observed value in a multi-objective problem, $\mathbf{y} \in Y$.

$\mathbf{y}^{(i)}$ i-th observed (vectorial) output, $\mathbf{y}^{(i)} \in \mathbb{R}^m$.

$\mathbf{y}^{(1:t)}$ Set of t observed vectorial outputs, $\mathbf{y}^{(1:t)} = \{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(t)}\} \subset Y$.

$\widehat{y}(\cdot)$ Kriging mean predictor, $\widehat{y}(\cdot) : X \to \mathbb{R}$.

$\widehat{y_j}(\cdot)$ Kriging mean predictor of the metamodel of the j-th objective.

$\widehat{\mathbf{y}}(\cdot)$ Vectorial kriging mean predictor, $\widehat{\mathbf{y}}(\cdot) = (\widehat{y_1}(\cdot), \dots, \widehat{y_m}(\cdot))^\top$.

Y objective space, $Y \subset \mathbb{R}^m$.

$Y(\cdot)$ Gaussian Process model for the function $f(\cdot)$.

$\mathbf{Y}(\cdot)$ (Independent) Gaussian Process models for the multi-objective function $\mathbf{f}(\cdot)$, $\mathbf{Y}(\cdot) = (Y_1(\cdot), \dots, Y_m(\cdot))^\top$.

AddGP	Additive Gaussian Process.
C-EHI	Centered Expected Hypvervolume Improvement.
CAD	Computer Aided Design.
CFD	Computational Fluid Dynamics.
DoE	Design of Experiments.
EGO	Efficient Global Optimization.
EHI	Expected Hypervolume Improvement (also known as EHVI).
EI	Expected Improvement.
EMOA	Evolutionary Multi-objective Optimization Algorithm.
EMI	Expected Maximin Improvement.
EV	Expected Violation.
GLS	Generalized Least Squares.
GP	Gaussian Process.
IGD	Inverted Generational Distance.
KB	Kriging Believer.
KPCA	Kernel Principal Component Analysis.
LCB	Lower-Confidence Bound.
LHS	Latin Hypercube Sampling.
mEI	Multiplicative Expected Improvement.
ND	Non-dominated.
NSGA-II	Non-dominated Sorting Genetic Algorithm II.
PCA	Principal Component Analysis.
PF	Pareto Front.
PI	Probability of Improvement.
PLS	Partial Least Squares.
PoF	Probability of Feasibility.
R-EHI	Reference-point based Expected Hypvervolume Improvement.
REMBO	Random Embedding Bayesian Optimization.
RKHS	Reproducing Kernel Hilbet Space.
SIR	Sliced Inverse Regression.
SMS	$\mathcal{S}$-Metric Selection.
SUR	Stepwise Uncertainty Reduction.
WEHI	Weighted Expected Hypervolume Improvement.

Chapter 1

Introduction

As is common in design engineering, a vehicle is made of several systems interacting together, such as the engine, the suspensions, the bodystructure, electrical devices. To guarantee performance, reliability, user-comfort and to comply with certifications, these systems need to be optimized. Over the last decades, computer codes have increasingly replaced physical experiments and one is capable to simulate the behavior of the car in various fields such as combustion, aerodynamics, aeroacoustics, noise vibration and harshness (NVH), electromagnetics, etc., reducing the costs of prototyping and the time required for designing new cars. A high-fidelity simulation nevertheless requires computationally demanding numerical simulators. In applications such as combustion or aerodynamics for instance, highly non-linear systems of Partial Differential Equations (PDE) need to be solved for simulating the ignition inside the combustion chamber or the flow around the vehicle. Because of the complex modeling of the underlying physical phenomena (spray, turbulence), Computational Fluid Dynamics (CFD) codes require the meshing of the system (i.e. the engine or the external shape of the vehicle) at a very fine resolution to guarantee precision. Standard techniques such as finite volumes or finite elements consist in a meshing of the shape in n_{el} elements on which the PDE is addressed. Iterative solvers aim at finding the solution at any vertex by solving a system of n_{el} equations. This operation is numerically expensive due to the large number of nodes in the mesh (several decades of millions), and a single simulation time typically ranges between 12 and 24 hours.

More than the accurate prediction of the vehicle's behavior, it is the possibility to optimize the systems through numerical simulation which is aimed at. In industrial applications, optimization aims at proposing new attractive designs that comply with more binding regulations (EURO, 2016, CAFE, 2011). It also supports decision makers by providing not only solutions but also insights in the non-intuitive design of complex systems. The classical approach to design optimization relies on Computer Aided Design (CAD): shapes of interest are restricted to a class of designs parameterized by d variables $x_1, \dots, x_d$, $\mathbf{x} \in X$, which define the associated shape $\Omega_{\mathbf{x}}$. These variables stand for various characteristics of the design: they include macro descriptions such as sizes (height, width, length of the design) as well as smaller details (radii, angles, local adjustments, ...). Depending on the level of refinement, there may be a large number of variables, $d \gtrsim 50$.

To obtain the optimal configuration of the system, parametric shape optimization aims at finding the CAD parameters $\mathbf{x}^*$ which minimize a physical objective function $f(\mathbf{x})$.

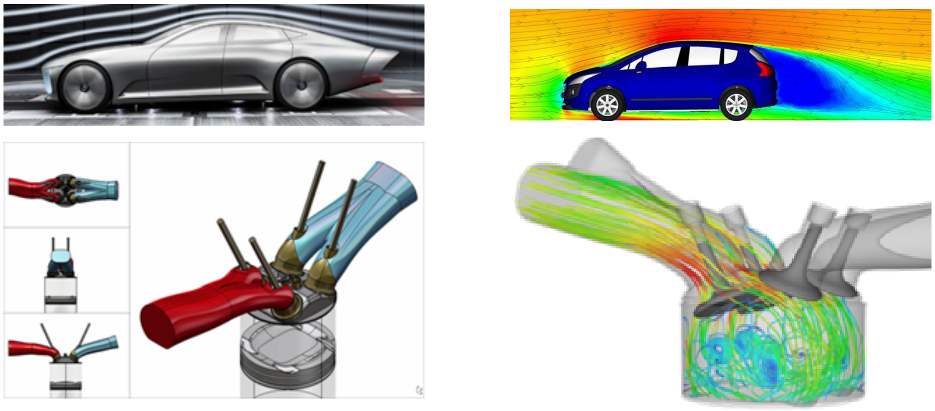


Figure 1.1: Typical systems evaluated through CFD and which are optimized.

Because of the expense associated to one numerical simulation, optimizations are budgeted: depending on project schedules and on the duration of one simulation, a prescribed number of function evaluations, the *budget* (typically of the order of 100), is allowed. Among these few evaluated designs, the best observed one[1] is chosen. Standard optimization methods such as evolutionary algorithms (Deb, 2001; Eiben and Smith, 2003; Michalewicz, 2013) require a large amount of function evaluations before finding the optimum. Gradient-based methods (Liu and Nocedal, 1989) are faster but require $\nabla f(\mathbf{x})$ which is generally unknown, and above all, only converge to a local optimum whose quality depends on the starting design. Multistart methods are a step towards global optimization at the expense of a larger number of function evaluations. These methods are not adapted to the "expensive black-box" objective functions we consider, for which the link between a design $\mathbf{x}$ and its associated output $y = f(\mathbf{x})$ is exclusively available through a computer experiment, and for which an approximation of $\nabla f(\mathbf{x})$ via finite differences would require d additional simulations. Optimizing a faster lower-fidelity function $\widetilde{f}(\mathbf{x})$ is not a solution in most cases since critical physical phenomena may be omitted. Instead, this thesis focuses on optimization algorithms which only require a small amount of function evaluations to propose a solution. Such methods (Jones, 2001) hinge on a cheap surrogate model (or metamodel, e.g., Gaussian Processes in Rasmussen and Williams, 2006, Support Vector Regression in Loshchilov et al., 2010, or polynomial chaos in Sudret, 2008) to the computer code built upon past simulations which is employed for the iterative construction of a sequence of promising designs $\mathbf{x}^{(n+1)}, \mathbf{x}^{(n+2)}, \dots, \mathbf{x}^{(budget)}$, to rapidly direct the search towards optimal designs. Surrogate-based approaches have proven their effectiveness in locating $\mathbf{x}^*$ within a few iterations on a wide range of applications (Forrester and Keane, 2009; Shahriari et al., 2015).

The performance of such methods nonetheless degrades when the number of considered design variables $x_1, \dots, x_d$ is large. This phenomenon, known as the *curse of dimension-*

[1] or one among the set of best designs in a multi-objective problem.

ality (Bellman, 1961), makes it difficult to optimize parameterized shapes. Moreover, even though they are intuitive to a designer to automate shape generation, the CAD parameters x_i are not intended to satisfy any mathematical property, and may not be the most relevant way to characterize the underlying object. Correlations typically exist between some pairs or groups of x_i's, and some variables describe the design globally as opposed to others which locally refine the shape. A parameterization related to the entire shape instead of marginal details and whose impact on the shape is quantifiable would be preferable.

Engineers often would like to optimize systems with regard to multiple conflicting objectives $f_1(\cdot), \dots, f_m(\cdot)$, and potentially wish to specify feasibility constraints. Instead of a mono-objective problem, optimal trade-off solutions to a (constrained) multi-objective problem known as the Pareto set are sought. Surrogate-based approaches have been extended to this setting (Binois, 2015; Emmerich et al., 2006; Wagner et al., 2010) to rapidly locate the Pareto set/front. However, it is not possible to approximate it accurately within a restricted number of function evaluations, especially when more than 2 or 3 objectives are considered because the size of the Pareto set grows exponentially with the number of objectives. Moreover, a large part of the Pareto front/set has a limited interest in applications, and surrogate-based optimization methods become computationally more demanding as m increases. One would benefit from enhancing convergence towards relevant solutions instead of trying to uncover the whole front most often in vain considering the budget limitations.

Following existing works, Bayesian multi-objective optimization methods are further developed in this thesis. State-of-the-art techniques and concepts in Gaussian Processes, Bayesian optimization, and multi-objective optimization are reviewed in Chapter 2.

Chapter 3 introduces a benchmark test problem made of real-world aerodynamic simulations, the MetaNACA. The MetaNACA benchmark is often employed, in addition to classical academic functions, for testing and evaluating multi-objective optimizers, dimension reduction techniques, as well as other methods developed throughout this thesis. It has tunable dimensions, $d = 3, 8, 22$ parameters, and several number of objectives, $m = 2, 3, 4$.

Chapter 4 is devoted to the R-EHI algorithm, a new Bayesian multi-objective optimizer. Contrarily to existing methods, this algorithm uncovers the Pareto front in steps. It prioritizes convergence towards user-desired solutions during its first phase by revisiting the Expected Hypervolume Improvement (EHI, Emmerich et al., 2006) acquisition criterion. Once a convergence criterion has detected the attainment of the Pareto front in that preferred part, in its second phase, R-EHI aims at unveiling a broader region of the Pareto front. The breadth of this new targeted area is determined by forecasting the width of the Pareto front that can be accurately discovered during the remaining iterations. R-EHI assumes user-preferences have been provided. If this is not the case, as non-compromising designs usually have little interest in applications, C-EHI chooses the center of the Pareto front as a default region where to seek solutions first. The concept of Pareto front center is a contribution of this thesis defined in Chapter 4 together with properties and estimation methods relying on Gaussian Processes.

Chapter 5 extends the C-EHI/R-EHI algorithm to exploit parallel computing possibilities of the objective functions, and to consider optimization constraints. The acquisition function which dictates the designs to be evaluated by the simulator is modified to yield a batch of promising designs per iteration and to consider the constraints.

Chapter 6 is devoted to the dimension reduction of parametric shapes. First, the non-supervised learning of a shape database through a Principal Component Analysis (PCA) permits to build a new basis which describes the shapes globally. The axes that contribute the most to the output's variation are selected through a regularized likelihood maximization, and are emphasized inside an additive GP of lower dimension which does not completely disregard the less important directions. Bayesian optimization is carried out in the smaller dimensional space of active components, complemented by a random embedding (Wang et al., 2013) in the space of remaining components, to address the optimization in reduced dimension. Finally in Chapter 7, the contributions of this thesis are summarized, and directions for future research are proposed.

Chapter 2

Basics in Gaussian Processes and Multi-Objective Optimization

Contents

We consider the framework of expensive and/or time-consuming experiments. The underlying phenomenon is considered as a black-box function, $f : \mathbb{R}^d \to \mathbb{R}$ and the link between the d real valued parameters (or variables) $\mathbf{x} \in X \subset \mathbb{R}^d$ and the scalar output $y = f(\mathbf{x})$ is only available through a costly experiment. In this setting, it is customary to replace $f(\cdot)$ by a cheap surrogate model $\widehat{f}(\cdot)$, alternatively called meta-model, or response surface. Different types of surrogate models have been employed in the literature, e.g. Generalized Linear Models (Hastie et al., 2005; McCullagh and Nelder, 1989), Radial Basis Functions (Broomhead and Lowe, 1988), Gaussian Processes (Cressie, 1992; Rasmussen and Williams, 2006; Stein, 1999), Artificial Neural Networks (Hastie et al., 2005; Zurada, 1992), Support Vector Regression (Boser et al., 1992; Drucker et al., 1997; Scholkopf and Smola, 2001; Vapnik and Chervonenkis, 1974), Regression Trees (Breiman et al., 1984), Lasso (Tibshirani, 1996) or ridge regression (Hastie et al., 2005), etc. They differ in the nature, quantity and dimension of the data they can handle, but share a common point: evaluating $\widehat{f}(\mathbf{x})$ is much cheaper than running $f(\mathbf{x})$.

In the following, only Gaussian Process (GP) surrogate models are employed. They were first proposed by Krige (Krige, 1951) for geophysical applications before being formalized by Matheron under the name *kriging* (Matheron, 1962, 1969).

We chose GPs mainly for two reasons. First, they provide a distribution probability $\pi(\mathbf{x})$ at any $\mathbf{x}$, instead of solely returning a prediction $\widehat{f}(\mathbf{x})$. This is particularly appealing for global optimization because GPs are equipped with a built-in exploitation/exploration mechanism (Jones et al., 1998). The strong mathematical background they hinge on (Cressie, 1992; Rasmussen and Williams, 2006; Stein, 1999) also facilitates their construction, interpretability, and flexibility. Second, these methods are particularly efficient in the framework of small data: accurate predictions are achieved even when only limited observations $\mathcal{D}_n := \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})\}$, $y^{(i)} = f(\mathbf{x}^{(i)})$ are available. Other methods may be more appropriate when n goes beyond 1000-2000 observations, albeit making GPs tractable for larger datasets is an ongoing field of research (Rullière et al., 2018; Titsias, 2009). But since the problems we consider are expensive, acquiring 100 or 200 experiments is anyway an approximate upper bound for n.

2.1 Gaussian Processes

In this part, definitions, properties, and practical techniques to handle Gaussian Processes are introduced.

2.1.1 Definition

Definition 2.1. *(Gaussian Process) A Gaussian Process $Z : X \subset \mathbb{R}^d \to \mathbb{R}$ is a collection of random variables such that $\forall n \in \mathbb{N}$, $\forall \mathbf{x}^{(i)} \in X$, $Z(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)})$ is a Gaussian vector.*

A Gaussian Process is a random function, entirely characterized by its mean function $m : X \to \mathbb{R}$, $m(\mathbf{x}) = \mathbb{E}[Z(\mathbf{x})]$ and its covariance function, a.k.a. its *kernel* $k : X \times X \to \mathbb{R}$, $k(\mathbf{x}, \mathbf{x}') = Cov(Z(\mathbf{x}), Z(\mathbf{x}'))$. We employ the notation $Z(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$. $m(\cdot)$ corresponds to a long term behavior. One may choose a relevant basis (e.g. a polynomial basis) $\mathbf{F}(\cdot)$ and fit some regression coefficients $\boldsymbol{\beta}$, $m(\mathbf{x}) = \mathbf{F}(\mathbf{x})^\top \boldsymbol{\beta}$ (Universal Kriging, see Forrester and Keane, 2009), but a common practice that we follow in this thesis is the use of a constant mean, $m(\mathbf{x}) = \beta$ (Ordinary Kriging, see Forrester and Keane, 2009), learned from the data. The covariance function has to be a symmetric semi-positive definite function, i.e.

$$\forall \mathbf{x}, \mathbf{x}' \in X, k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$$

$$\forall n \in \mathbb{N}, \forall \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)} \in X, \forall \boldsymbol{\alpha} \in \mathbb{R}^n, \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \geq 0$$

Therefore, the Gram matrix (or kernel matrix) $\mathbf{K} \in \mathbb{R}^{n\times n}$ whose elements are $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is symmetric semi-positive definite too.

A wide variety of kernels can be found in the literature (Rasmussen and Williams, 2006). They measure the dependence between $\mathbf{x}, \mathbf{x}' \in X$. Four usual one-dimensional kernels are given in Example 2.1. They include some hyperparameters ϑ which account for modifications of the original kernel, an horizontal or vertical scaling in the following example where $\vartheta = (\theta, \sigma^2)$. Those hyperparameters let themselves interpret as the length-scale of the GP and the variance of the GP.

Example 2.1. *(Usual kernels).*

- *Exponential kernel* $k(x,x') = \sigma^2 \exp\left(-\frac{|x-x'|}{\theta}\right)$
- *Matérn 3/2 kernel* $k(x,x') = \sigma^2 \left(1 + \frac{\sqrt{3}|x-x'|}{\theta}\right) \exp\left(-\frac{\sqrt{3}|x-x'|}{\theta}\right)$
- *Matérn 5/2 kernel* $k(x,x') = \sigma^2 \left(1 + \frac{\sqrt{5}|x-x'|}{\theta} + \frac{5|x-x'|^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}|x-x'|}{\theta}\right)$
- *Squared-exponential kernel* $k(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$

The kernel is the main ingredient of the Gaussian Process. It defines a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ (Gretton, 2013) of functions with particular regularity and/or some other features. The mean predictor $\widehat{y}(\cdot)$ of a GP $Y(\cdot)$ with kernel $k(\cdot,\cdot)$ belongs to $\mathcal{H}_k$. The RKHS's of kernels in Example 2.1 are the space of $\mathcal{C}^0$, $\mathcal{C}^1$, $\mathcal{C}^2$ or $\mathcal{C}^\infty$ functions, respectively. Figures 2.1, 2.2 and 2.3 show the effect of varying θ, σ^2, or the kernel of GPs with covariance function $k(\cdot,\cdot)$.

Many algebraic operations preserve the semi-positive definiteness. If $k_1(\cdot,\cdot)$ and $k_2(\cdot,\cdot)$ are semi-positive definite kernels, the sum $k_1(\mathbf{x},\mathbf{x}') + k_2(\mathbf{x},\mathbf{x}')$ or the tensor sum $k_1(\mathbf{x}^a, \mathbf{x}'^a) + k_2(\mathbf{x}^b, \mathbf{x}'^b)$ (where $\mathbf{x}^a$ and $\mathbf{x}^b$ form a partition of $\mathbf{x}$) are semi-positive functions, hence valid covariance functions too. This also applies to the product and tensor product, warping of the input space, etc. (see Rasmussen and Williams, 2006, for more details).

A kernel is said to be stationary if $k(x,x') = \tilde{k}(x-x')$, meaning that the covariance between $Z(x)$ and $Z(x')$ only depends on the distance between x and x'. It further implies that the distribution of the GP is insensitive to translations. The kernels in Example 2.1 and those which will be used throughout this thesis are all stationary kernels. In such cases, $k(x,x) = \tilde{k}(0) = \sigma^2$, the variance of the GP, and the regularity of the GP only depends on the regularity of $\tilde{k}(0)$ (Stein, 1999).

With multi-dimensional inputs $\mathbf{x} \in \mathbb{R}^d$, the distance between $\mathbf{x}$ and $\mathbf{x}'$ is measured by the weighted Euclidean distance,

$$\|\mathbf{x} - \mathbf{x}'\|_{\boldsymbol{\theta}} = \left(\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{\theta_i^2}\right)^{1/2}.$$

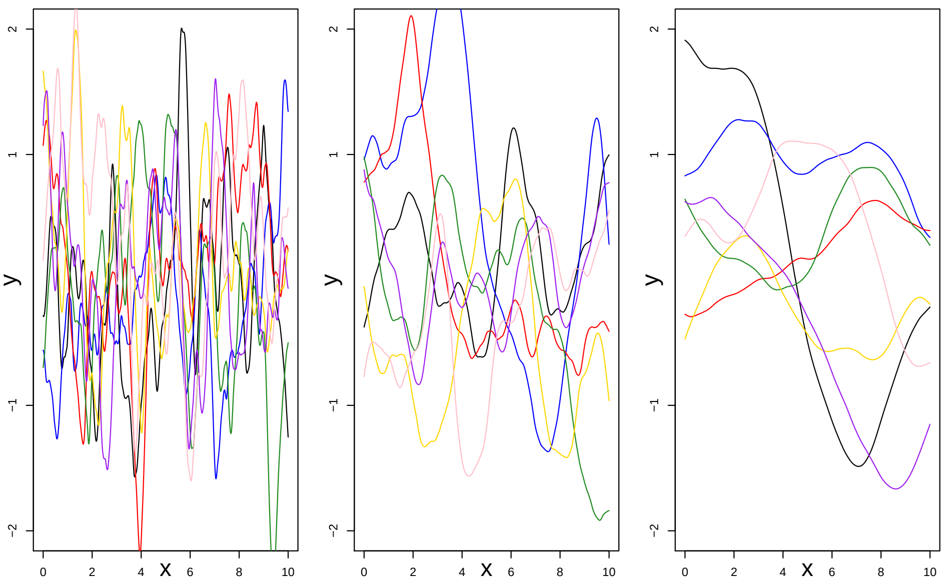


Figure 2.1: Effect of changing the length-scale of the GP. Left: $\theta = 0.3$. Center: $\theta = 1$. Right: $\theta = 3$.

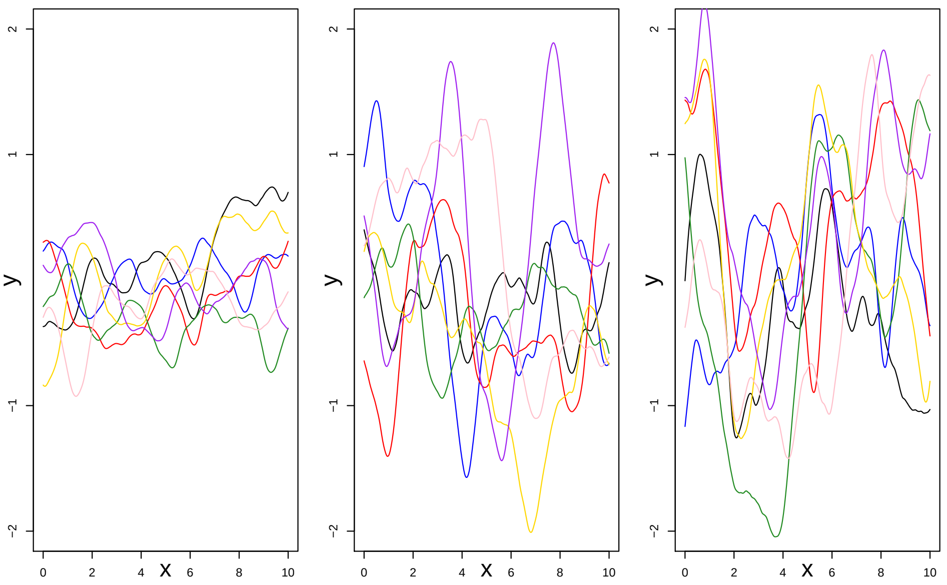


Figure 2.2: Effect of changing the variance of the GP. Left: $\sigma^2 = 0.1$. Center: $\sigma^2 = 0.5$. Right: $\sigma^2 = 1$.

$\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)^\top \in \mathbb{R}^d$ is a vector of length-scales associated to each dimension. If $\theta_i = \theta$ $\forall i = 1, \dots, d$ the kernel is isotropic, otherwise, an anisotropic behavior is implemented, which is of interest when the GP needs to vary differently according to the direction. By

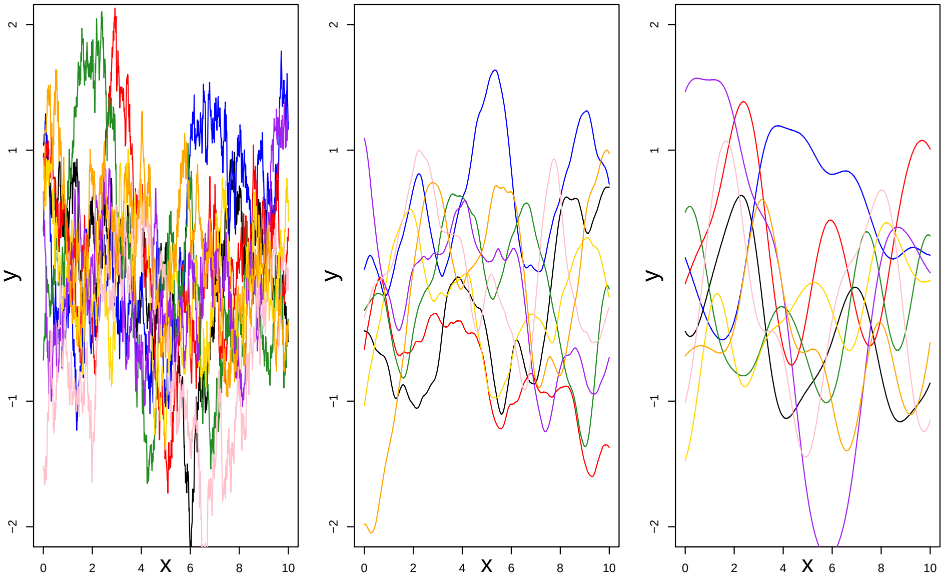


Figure 2.3: Effect of changing the kernel of the GP. Left: exponential kernel. Center: Matérn 5/2 kernel. Right: squared-exponential kernel.

learning these hyperparameters from the data, an Automatic Relevance Determination (ARD, Rasmussen and Williams, 2006) is obtained.

2.1.2 Gaussian conditioning

Definition 2.1 states that $\forall \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)} \in X$,

$$\begin{pmatrix} Z(\mathbf{x}^{(1)}) \\ \vdots \\ Z(\mathbf{x}^{(n)}) \end{pmatrix} \sim \mathcal{N}_n\left(\mathbf{1}_n\beta, \mathbf{K}\right)$$

where $\mathcal{N}_n$ is the n-dimensional Gaussian distribution and $\mathbf{K}$ is the covariance matrix with elements $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

Let $\mathcal{D}_n := \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})\} = \{\mathbf{x}^{(1:n)}, y^{(1:n)}\}$ be n observations of $f(\cdot)$. By applying the Gaussian conditioning formulae (Eaton, 1983), $Z(\cdot)$ can be conditioned by $\{Z(\mathbf{x}^{(1)}) = y^{(1)}, \dots, Z(\mathbf{x}^{(n)}) = y^{(n)}\}$, and the conditional GP

$$Y(\cdot) := [Z(\cdot)|\mathcal{D}_n] \sim \mathcal{GP}(\widehat{y}(\cdot), c(\cdot,\cdot)), \tag{2.1}$$

is obtained, where

$$\widehat{y}(\mathbf{x}) = \widehat{\beta} + k(\mathbf{x}, \mathbf{x}^{(1:n)})^\top \mathbf{K}^{-1}(y^{(1:n)} - \mathbf{1}_n\widehat{\beta}) \tag{2.2}$$

is the conditional mean function (a.k.a., the kriging mean predictor) and

$$c(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, \mathbf{x}^{(1:n)})^\top \mathbf{K}^{-1} k(\mathbf{x}', \mathbf{x}^{(1:n)}) + \frac{(1 - \mathbf{1}_n^\top \mathbf{K}^{-1} k(\mathbf{x}, \mathbf{x}^{(1:n)}))(1 - \mathbf{1}_n^\top \mathbf{K}^{-1} k(\mathbf{x}', \mathbf{x}^{(1:n)}))}{\mathbf{1}_n^\top \mathbf{K}^{-1} \mathbf{1}_n} \tag{2.3}$$

is the conditional covariance function, from which stems the conditional variance, $s^2(\mathbf{x}) = c(\mathbf{x}, \mathbf{x})$ (see e.g. Rasmussen and Williams, 2006, for details). Remark that the latter does not depend on the observations $y^{(1:n)}$ but solely on a distance metric induced by $k(\cdot, \cdot)$.

In particular, for any new $\mathbf{x}$, $Y(\mathbf{x}) \sim \mathcal{N}(\widehat{y}(\mathbf{x}), s^2(\mathbf{x}))$. $\widehat{y}(\mathbf{x})$ and $s^2(\mathbf{x})$ let themselves interpret as the kriging mean predictor, and the uncertainty of this prediction at any unknown $\mathbf{x}$. $\widehat{y}(\mathbf{x})$ is known to be the Best Linear Unbiased Predictor (BLUP) at $\mathbf{x}$ and $s^2(\mathbf{x})$ its mean squared error (Sacks et al., 1989; Stein, 1999). From the RKHS point of view (Gretton, 2013), $\widehat{y}(\cdot)$ is the function $h(\cdot) \in \mathcal{H}_k$ with minimal (RKHS) norm, $\langle h, h \rangle_{\mathcal{H}_k}$, which interpolates the observations, i.e. $h(\mathbf{x}^{(i)}) = y^{(i)}$, $i = 1, \ldots, n$.

Following Definition 2.1 and (2.1), the distribution of $Y(\cdot)$ at a set of untested points $\{\mathbf{x}^{(n+1)}, \ldots, \mathbf{x}^{(n+s)}\}$ is an s-dimensional Gaussian vector:

$$\begin{pmatrix} Y(\mathbf{x}^{(n+1)}) \\ \vdots \\ Y(\mathbf{x}^{(n+s)}) \end{pmatrix} \sim \mathcal{N}_s \left(\begin{pmatrix} \widehat{y}(\mathbf{x}^{(n+1)}) \\ \vdots \\ \widehat{y}(\mathbf{x}^{(n+s)}) \end{pmatrix}, \boldsymbol{\Gamma} \right) \tag{2.4}$$

where $\Gamma_{ij} = c(\mathbf{x}^{(n+i)}, \mathbf{x}^{(n+j)})$. The knowledge of the distribution of $(Y(\mathbf{x}^{(n+1)}), \ldots, Y(\mathbf{x}^{(n+s)}))^\top$ enables to draw samples of the conditional GP at these locations (Binois et al., 2015a) as shown on the left part of Figure 2.5.

The computational bottleneck for GP prediction and uncertainty quantification is the inversion of $\mathbf{K}$ in (2.2) and (2.3). $\mathbf{K}$ is symmetric semi positive definite and its Cholesky factorization (Johnson and Horn, 1985) accelerates the inversion. The complexity of the procedure is $\mathcal{O}(n^3)$. The remaining operations are matrix-vector products of complexity $\mathcal{O}(n^2)$[1], and can be accelerated by judicious algebraic tricks (Roustant et al., 2012). Likewise, for simulating conditional GPs (2.4), the inversion (or Cholesky factorization) of the $s \times s$ matrix $\boldsymbol{\Gamma}$ is required. Gaussian Processes are therefore well-suited for "small-data" problems where few observations ($n \approx$ 100-200) are available. An approximate upper bound for n and s is 5000 points. Notice that the estimation of hyperparameters described in Section 2.1.3 involves several inversions of $\mathbf{K}$, hence is already expensive when $n \approx 1000$.

Example 2.2. *(Gaussian Process conditioning: prediction, uncertainty quantification, and simulation).*

Let $Z(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot))$ where $k(\cdot, \cdot)$ is a Matérn 5/2 kernel with variance $\sigma^2 = 0.2$ and length-scale $\theta = 1$. Figure 2.4 shows 10 random functions with this prior.

The function $f(x) = \sin(x)\exp(-x^2/40)$ to be learned is observed at $\{x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}\} = \{0, 2, 4, 6, 8, 10\}$, $y^{(i)} = f(x^{(i)})$, $i = 1, \ldots, 6$, shown in Figure 2.4.

The conditional GP $Y(\cdot) = [Z(\cdot) | \{Z(x^{(1)}) = y^{(1)}, \ldots, Z(x^{(6)}) = y^{(6)}\}]$ interpolates the data. Ten realizations of $Y(\cdot)$ are shown on the left hand side of Figure 2.5. The kriging mean predictor $\widehat{y}(x) = \mathbb{E}[Z(x)]$ and its variance $s^2(x) = Var(Z(x))$ are the mean and the variance over these random curves. They can be computed analytically via (2.2) and

[1] $\mathcal{O}(ln^2)$ if predicting at l locations.

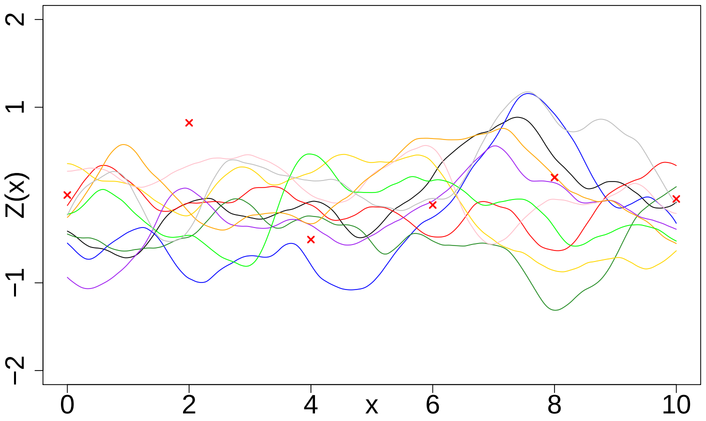


Figure 2.4: Ten samples of the initial (unconditioned) GP $Z(\cdot)$ and observations $(x^{(i)}, y^{(i)})$ (red crosses).

(2.3) and are shown on the right hand side of Figure 2.5. Notably, $\widehat{y}(x^{(i)}) = y^{(i)}$ and $s^2(x^{(i)}) = 0 \ \forall i = 1, \dots, 6$.

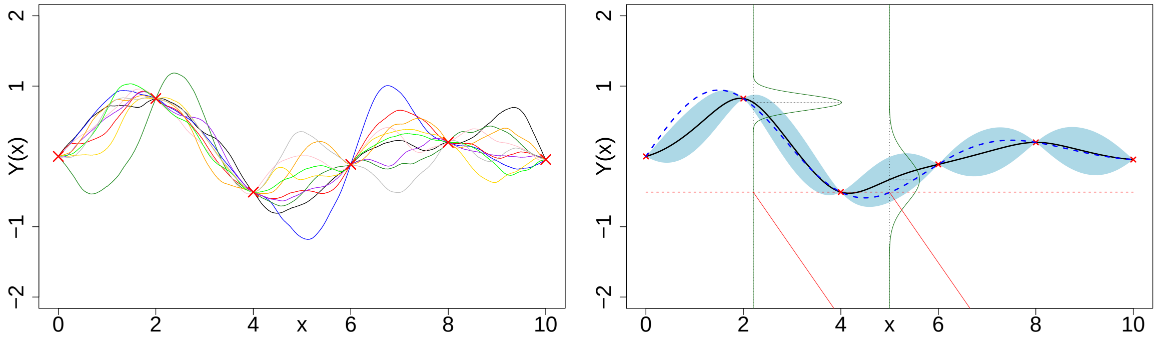


Figure 2.5: Left: samples of the conditional GP which interpolate the data. Right: kriging mean predictor (black curve) and $[\widehat{y}(x) - s(x), \widehat{y}(x) + s(x)]$ confidence interval (light blue envelope). The dotted blue line is the true function. The distribution at two untested designs $x = 2.2$ and $x = 5$, $Y(2.2) \sim \mathcal{N}(\widehat{y}(2.2), s^2(2.2))$, $Y(5) \sim \mathcal{N}(\widehat{y}(5), s^2(5))$, is the vertical green density.

Noisy measurements

We deal with GPs that interpolate the data, i.e. $Y(\mathbf{x}^{(i)}) = y^{(i)}$, $\forall i = 1, \dots, n$. GPs can also handle noisy measurements where the observations $y^{(i)}$ are corrupted by noise $\varepsilon^{(i)}$ with variance τ_ε^2, i.e. $y^{(i)} = f(\mathbf{x}^{(i)}) + \varepsilon^{(i)}$ are observed. Using $\mathbf{K} + \tau_\varepsilon^2 \mathbf{I}_n$ instead of the Gram matrix $\mathbf{K}$ incorporates this uncertainty inside the GP, and $\widehat{y}(x^{(i)}) \neq y^{(i)}$, $s^2(x^{(i)}) \neq 0$. τ_ε^2 is also called the nugget effect (Cressie, 1988; Rasmussen and Williams, 2006; Roustant et al., 2012), and has the additional effect of improving the conditioning of the Gram matrix. GP regression with nugget effect is related to ridge regression (Hastie et al., 2005) in the RKHS framework (Gretton, 2013).

2.1.3 Hyperparameter estimation

The kernels in Example 2.1 include hyperparameters which may be difficult to hand-tune. The mean of the GP, β, has also to be chosen. One way to determine them is to resort to the probabilistic nature of GPs. Since $(Y(\mathbf{x}^{(1)}), \dots, Y(\mathbf{x}^{(n)}))^\top \sim \mathcal{N}_n(\beta, \mathbf{K})$, the likelihood of $Y(\cdot)$ is

$$L(\vartheta; \mathbf{x}^{(1:n)}, y^{(1:n)}) = f_{Y(\mathbf{x}^{(1)}),\dots,Y(\mathbf{x}^{(n)})}(\vartheta; y^{(1:n)}),$$

the density of the Gaussian vector $(Y(\mathbf{x}^{(1)}), \dots, Y(\mathbf{x}^{(n)}))^\top$ at the observations $y^{(1:n)}$ when $Y(\cdot)$ has the hyperparameters ϑ. L has closed-form expression (Roustant et al., 2012):

$$L(\vartheta; \mathbf{x}^{(1:n)}, y^{(1:n)}) = \frac{1}{(2\pi)^{n/2}|\mathbf{K}|^{1/2}} \exp\left(-\frac{1}{2}(y^{(1:n)} - \mathbf{1}_n\beta)^\top \mathbf{K}^{-1}(y^{(1:n)} - \mathbf{1}_n\beta)\right) \quad (2.5)$$

where $|\mathbf{K}|$ stands for the determinant. The hyperparameters $\vartheta = (\theta_1, \dots, \theta_d, \sigma^2)$ are contained in $\mathbf{K}$ through $k(\cdot, \cdot)$. Maximizing L with respect to ϑ means finding the hyperparameters under which the observed data is the most likely to have been generated by a GP $Y(\cdot)$ possessing this ϑ.

Usually, the log-likelihood $l(\vartheta; \mathbf{x}^{(1:n)}, y^{(1:n)}) = \log(L(\vartheta; \mathbf{x}^{(1:n)}, y^{(1:n)}))$ is maximized instead (Rasmussen and Williams, 2006; Roustant et al., 2012). To break the dependence between β, σ^2 and the θ_j's in (2.5), by setting their partial derivatives to 0, β and σ^2 are estimated by

$$\widehat{\beta} = \frac{\mathbf{1}_n^\top \mathbf{R}_{\boldsymbol{\theta}}^{-1} y^{(1:n)}}{\mathbf{1}_n^\top \mathbf{R}_{\boldsymbol{\theta}}^{-1} \mathbf{1}_n}$$

and

$$\widehat{\sigma^2} = \frac{1}{n}(y^{(1:n)} - \mathbf{1}_n\widehat{\beta})^\top \mathbf{R}_{\boldsymbol{\theta}}^{-1}(y^{(1:n)} - \mathbf{1}_n\widehat{\beta})$$

where $\mathbf{R}_{\boldsymbol{\theta}}$ is the correlation matrix with entries $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ released from the GP variance σ^2 and which only depends on $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)^\top$, $\sigma^2\mathbf{R}_{\boldsymbol{\theta}} = \mathbf{K}$. $\widehat{\beta}$ is the Generalized Least Squares estimate of β (Hastie et al., 2005; Roustant et al., 2012) and this approach extends to universal kriging where $\boldsymbol{\beta}$ is a vector of regressors. Both terms are plugged in (2.5), and finally the concentrated log-likelihood,

$$\widehat{l}(\vartheta; \mathbf{x}^{(1:n)}, y^{(1:n)}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\widehat{\sigma^2}) - \frac{1}{2}\log(|\mathbf{R}_{\boldsymbol{\theta}}|) - \frac{n}{2} \quad (2.6)$$

is maximized to find $\vartheta^* = (\theta_1^*, \dots, \theta_d^*, \widehat{\sigma^2})$. (2.6) is differentiable with respect to θ_j, $j = 1, \dots, d$ (see Roustant et al., 2012, for the formula), and is maximized by means of standard gradient-based techniques, such as BFGS (Liu and Nocedal, 1989) with multistart.

In a purely Bayesian framework, ϑ are instead given a prior distribution and ϑ^* is sampled from the posterior distribution. However, such approaches usually rely on expensive MCMC procedures, which make them much harder to use in practice. Other techniques to set-up ϑ include cross-validation (Bachoc et al., 2017). This is a common approach to fit the hyperparameters of surrogate models which do not possess a likelihood of ϑ given the observations, such as Radial Basis Functions or Neural Networks.

2.2 Bayesian Optimization

Global optimization aims at finding $\mathbf{x}^* = \arg\min_{\mathbf{x}\in X} f(\mathbf{x})$. Common techniques in optimizaton (Allaire, 2005; Nocedal and Wright, 2006) hinge on assumptions such as the convexity of $f(\cdot)$, Lipschitz continuity, and/or the knowledge of its gradient, whose opposite is employed as a descent direction. Nonetheless, for a black-box function, the convexity of $f(\cdot)$ is an overly strong assumption and though $f(\cdot)$ might be sufficiently regular, its Lipschitz constant is obviously unknown. Gradient methods are not well-suited to find $f(\cdot)$'s global minimizer since they only converge to a local optimum. Additionally, $\nabla f(\cdot)$ is usually unknown and d additional function evaluations would be required to estimate it via finite differences (Smith, 1985). Last but not least, these methods need a large number of function evaluations which we cannot afford. Evolutionary Algorithms (Goldberg, 1989) are standard tools for global optimization but require many calls to $f(\cdot)$ too.

Bayesian optimization rather exploits a cheap surrogate model by placing a Gaussian Process prior over the function to be minimized: a GP $Y(\cdot)$ is fitted to the t previous observations $\{\mathbf{x}^{(1:t)}, y^{(1:t)}\}$. Instead of $f(\cdot)$, information provided by $Y(\cdot)$ is gathered to conduct the optimization. A review of Bayesian optimization methods can be found in Shahriari et al. (2015).

2.2.1 Design of Experiments

The first step of Bayesian optimization is the fitting of the GP (see Section 2.1.2) to a Design of Experiments (DoE). It is advocated (Jones et al., 1998; Loeppky et al., 2009) to pick up $n = 10d$ points $\mathbf{x}^{(1:n)} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\} \subset X$ where to evaluate $f(\cdot)$. To gather as much information as possible about $f(\cdot)$ in the limit of these n evaluations, $\mathbf{x}^{(1:n)}$ needs to be space-filling. Different such designs have been discussed in the literature. Sobol (Sobol', 1967) and Halton (Halton, 1960) designs are analytical sequences (hence quickly computable) which cover the design space X. Latin Hypercube Designs (McKay et al., 1979; Stein, 1987) are a class of DoEs which put restrictions on the location of the $\mathbf{x}^{(i)}$'s. Basically, X is divided in n^d hypercubes which contain at most one $\mathbf{x}^{(i)}$. Two $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ cannot share the same hyperrow or hypercolumn. Usually, a Latin Hypercube Sample (LHS) is generated and optimized with regard to an infill measure using an heuristic algorithm, such as simulated annealing (Van Laarhoven and Aarts, 1987). The most employed criterion is the maximin (Pronzato, 2017), which tends to maximize the smallest distance between two designs $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, $i, j = 1, \ldots, n$, $i \neq j$. The minimax (Pronzato, 2017) measures the maximal distance of any $\mathbf{x} \in X$ to its nearest neighbor within $\mathbf{x}^{(1:n)}$, and has to be minimized, but is much more cumbersome. Entropy, discrepancy or uniformity measures (Fang et al., 2005) of $\mathbf{x}^{(1:n)}$ can also be used for the DoE optimization.

Adaptive infill criteria

Once fitted to $\{\mathbf{x}^{(1:n)}, y^{(1:n)}\}$ adaptive criteria aim at improving $Y(\cdot)$ by sampling new designs (Picheny et al., 2010). Typically, the variance of the estimator is decreased by enriching the DoE with $\mathbf{x}^*$, the solution of D-optimality, A-optimality, E-optimality, G-optimality or I-optimality problems (Sacks et al., 1989), given the previous observations.

2.2.2 Infill criteria for optimization

Rather than enhancing the precision of $f(\cdot)$'s predictor, optimization aims at finding the minimum of $f(\cdot)$. The objective of adaptive optimization criteria is to generate a sequence of designs $\mathbf{x}^{(n+1)}, \ldots, \mathbf{x}^{(n+p)}$ such that $\min\limits_{i=1,\ldots,n+p} y^{(i)}$ gets as close as possible to the true minimum, $\min\limits_{\mathbf{x} \in X} f(\mathbf{x})$, where n is the number of designs in the initial DoE and p the number of additional infills.

The outline of a Bayesian optimization algorithm is depicted in Figure 2.6. In the Efficient Global Optimization (EGO) algorithm (Jones et al., 1998), once a GP has been fitted to the initial DoE $\mathcal{D}_n$, at each iteration $n \leq t < n+p$ a cheap infill criterion which relies on $Y(\cdot)$ is maximized (or minimized). Its optimum $\mathbf{x}^{(t+1)}$ is then evaluated by the simulator which returns $y^{(t+1)} = f(\mathbf{x}^{(t+1)})$, and $Y(\cdot)$ is updated. This step is repeated until the $budget := n + p$ is exhausted. To exploit parallel computing possibilities, e.g. when $f(\cdot)$ can be evaluated simultaneously on different computers or on several nodes of a cluster, infill criteria returning a batch of q designs have been proposed (Ginsbourger et al., 2010; Schonlau, 1997) and are the topic of Section 5.1. During the update step, the covariance parameters are re-estimated and the additional evaluation $(\mathbf{x}^{(t+1)}, y^{(t+1)})$ taken into account, which modifies the conditional mean and covariance of the GP. If the hyperparameters are unchanged, update formulae (Chevalier et al., 2014) enable the fast recomputation of $\widehat{y}(\cdot)$ and $s^2(\cdot)$. At the end of the procedure, the best observed design and its performance, $\mathbf{x}^* := \arg\min\limits_{i=1,\ldots,budget} f(\mathbf{x}^{(i)})$, $y^* := f(\mathbf{x}^*)$, are returned. To a certain extent, Bayesian optimization transforms the minimization of $f(\cdot)$ into the optimization of an acquisition function which solely relies on the cheap surrogate $\widehat{f}(\cdot)$ and only uses the expensive $f(\cdot)$ to evaluate the infill criterion-promoted design.

Both mean and variance of the GP predictor are differentiable almost everywhere[2]. They have closed form expressions which depend on the kernel and its gradient, see Stein (1999) for instance. This property is very appealing since the gradient of most infill

[2] With a stationary kernel, they are differentiable $\forall \mathbf{x} \in X$ except at the $\mathbf{x}^{(i)}$'s if the kernel is not regular enough.

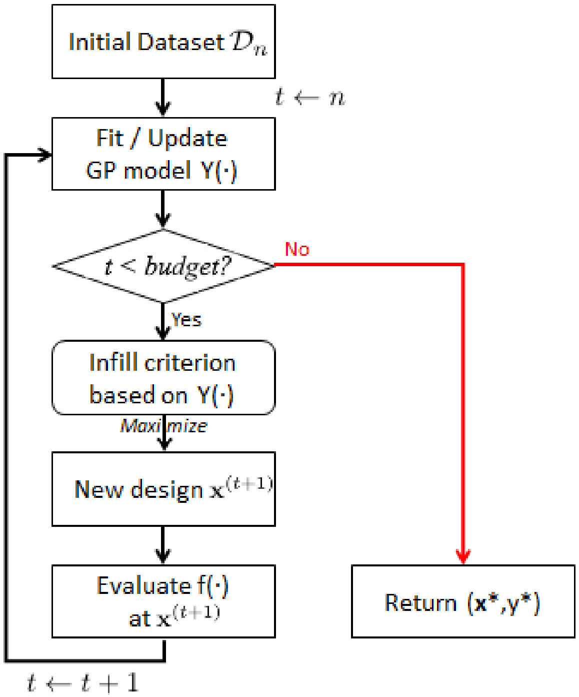

Initial Dataset $\mathcal{D}_n$
$t \leftarrow n$
Fit / Update GP model Y(·)
$t < budget?$
No
Yes
Infill criterion based on Y(·)
Maximize
New design $\mathbf{x}^{(t+1)}$
Evaluate f(·) at $\mathbf{x}^{(t+1)}$
Return ($\mathbf{x}^*$,y^*)
$t \leftarrow t+1$

Figure 2.6: Outline of a Bayesian optimization algorithm.

criteria (and in particular the ones detailed in this section) can be derived too. Global search methods such as evolutionary algorithms (Goldberg, 1989; Mebane Jr et al., 2011) can therefore be combined with gradient-based (Liu and Nocedal, 1989) techniques to accelerate and improve the optimization of the infill criterion.

Besides the metamodel, the acquisition function is the main ingredient of Bayesian optimization since it determines the sequence of designs that are evaluated. Several have been proposed in the literature (Jones, 2001; Jones et al., 1998). The most straightforward is to sample $f(\cdot)$ at the minimizer of the predictor mean, $\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x} \in X} \widehat{y}(\mathbf{x})$. But such a method gets rapidly stuck at a local optimum (Jones, 2001).

The probabilistic framework of GPs enables the use of the prediction uncertainty, $s^2(\mathbf{x})$, inside the infill criterion. A popular infill criterion is the probability of improvement (Kushner, 1964; Schonlau, 1997) over a target a, usually set as $f_{\min} := \min_{i=1,\dots,t} y^{(i)}$, the minimal value observed during the t function evaluations. It aims at finding the design which is the most likely to achieve better performance than the current best observed solution. Since $Y(\mathbf{x}) \sim (\widehat{y}(\mathbf{x}), s^2(\mathbf{x}))$, it is computable in closed form:

$$\mathrm{PI}(\mathbf{x}; a) = \mathbb{P}(Y(\mathbf{x}) \leq a) = \phi_{\mathcal{N}} \left(\frac{a - \widehat{y}(\mathbf{x})}{s(\mathbf{x})} \right), \tag{2.7}$$

where $\phi_{\mathcal{N}}$ stands for the normal cumulative distribution function. PI is a global criterion since it both promotes designs with small $\widehat{y}(\mathbf{x})$, or with large $s^2(\mathbf{x})$ when $a < \widehat{y}(\mathbf{x})$. However, it was found to get stuck at local optima for a long time before visiting other promising parts of X (Jones, 2001). In Figure 2.5, PI corresponds to the integral of the (vertical) green density for y under the red $f_{\min}$ dotted line.

Instead of solely considering the probability of improving a, the Expected Improvement (EI, Mockus, 1975) measures the magnitude of progress (over a, usually chosen as $f_{\min}$) which is expected, $\text{EI}(\mathbf{x};a) := \mathbb{E}[I(\mathbf{x};a)] = \mathbb{E}[(a - Y(\mathbf{x}))_+]$ where the *improvement function* $I(\mathbf{x};a) = (a - Y(\mathbf{x}))_+$ is a random variable measuring the progress at $\mathbf{x}$, and $(z)_+ := \max(z, 0)$. Likewise, it has closed-form expression:

$$\text{EI}(\mathbf{x};a) = (a - \widehat{y}(\mathbf{x}))\phi_{\mathcal{N}}\left(\frac{a - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\varphi_{\mathcal{N}}\left(\frac{a - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}\right) \tag{2.8}$$

where $\varphi_{\mathcal{N}}$ stands for the normal density function. The EI favors designs whose prediction improves over a (first part of 2.8) and/or for which the prediction variance is large (second part of 2.8). It is naturally equipped with an “exploitation-exploration” mechanism (Jones, 2001): parts of X in the vicinity of $\mathbf{x}^{(i)}$'s with good $f(\mathbf{x}^{(i)})$ value are promoted, but under-sampled regions of X with a large $s^2(\mathbf{x})$ too: when the predictor is very uncertain, prediction errors may be large and $f(\cdot)$ might be much smaller than $\widehat{y}(\cdot)$ there. Such areas should therefore not be disregarded and are episodically promoted by the EI. Contrarily to PI, promising under-sampled areas get visited by the EI more rapidly (Jones, 2001). In Figure 2.5, EI corresponds to the integral of the (vertical) green density for y under the red $f_{\min}$ dotted line multiplied by the red improvement line.

The EI is differentiable and has closed-form expression, see for instance Roustant et al. (2012),

$$\nabla\text{EI}(\mathbf{x};a) = -\nabla\widehat{y}(\mathbf{x}) \times \phi_{\mathcal{N}}(z(\mathbf{x})) + \nabla s(\mathbf{x}) \times \varphi_{\mathcal{N}}(z(\mathbf{x})), \tag{2.9}$$

where $z(\mathbf{x}) = (a - \widehat{y}(\mathbf{x}))/s(\mathbf{x})$. This property[3] is appealing for the efficient maximization of (2.8). $\nabla\widehat{y}(\mathbf{x})$ and $\nabla s(\mathbf{x})$ require the gradient of $Y(\cdot)$'s kernel $k(\cdot,\cdot)$ at $\mathbf{x}$, with the past observations $\mathbf{x}^{(1:t)}$, i.e. $\nabla k(\mathbf{x}, \mathbf{x}^{(1:t)})$, which is analytically computable. $\nabla s^2(\mathbf{x}) = 2s(\mathbf{x})\nabla s(\mathbf{x})$ helps computing $s(\mathbf{x})$'s gradient.

Another popular approach that achieves an exploitation-exploration trade-off is the minimization of a Lower Confidence Bound (LCB, Brochu et al., 2010; Srinivas et al., 2009). Instead of $\widehat{y}(\mathbf{x})$ which leads to over-exploitation, the minimizer of $\widehat{y}(\mathbf{x}) - \alpha_{LCB}s(\mathbf{x})$ is sought and used in the next iteration. α_{LCB} controls the exploration/exploitation trade-off. While theoretical results (Srinivas et al., 2009) propose specific increasing sequences of values, they are in practice hugely over-conservative, and in practice α_{LCB} is usually set to 1 or 2 (Emmerich et al., 2020; Ponweiser et al., 2008).

Figure 2.7 illustrates and compares these four infill criteria on a simple example. A GP has been fitted to $(x^{(1)}, y^{(1)}), \dots, (x^{(4)}, y^{(4)})$ (red crosses). The predictor $\widehat{y}(x)$ (Equation 2.2) is the black curve which aims at predicting the true function (dotted blue curve) at any untested x. $x^{(2)}$ is the current minimizer and $f_{\min} = y^{(2)}$ is the threshold for the EI (2.8) and the PI (2.7), which are the red and blue curves at the bottom. The green curve

[3] which is not limited to the EI, the gradient of other acquisition functions such as PI have closed-form expression too.

is the LCB, $\widehat{y}(x) - 2s(x)$. The next iterate selected by these four infill criteria differs. The maximizer of PI (blue triangle) and minimizer of $\widehat{y}(x)$ (black overlapped triangle) exploit too much previous observations and provide an $x^{(t+1)}$ close to the current minimizer. The exploitation/exploration mechanism of EI and LCB is highlighted: their maximizer (respectively minimizer) take the uncertainty of $\widehat{y}(x)$ into account and designs which are farther from $x^{(2)}$ though promising get promoted.

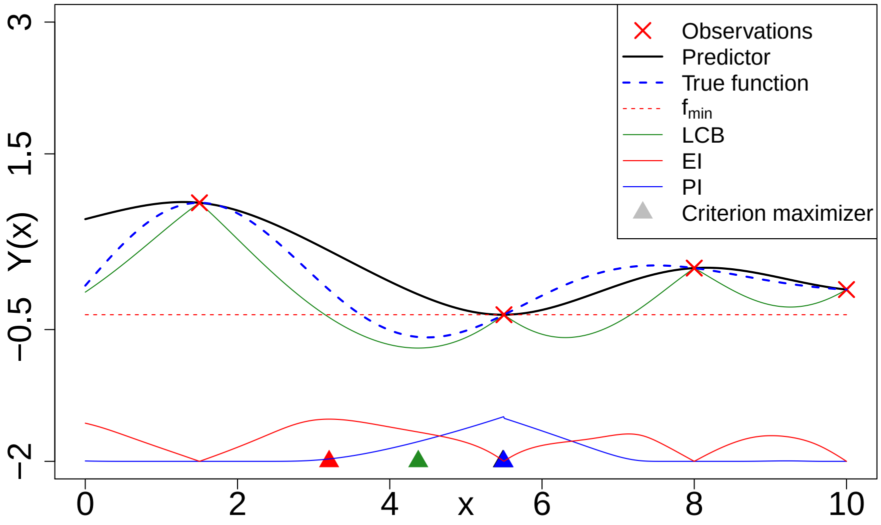


Figure 2.7: Comparison of four common Bayesian optimization acquisition functions. The new promoted iterate, $x^{(4+1)}$ (triangle) differs depending on the infill criterion.

Bayesian optimization is not limited to those criteria. Entropy Search (Hernández-Lobato et al., 2014; Villemonteix et al., 2009) is a Stepwise Uncertainty Reduction (SUR) strategy (Picheny, 2014) where $\mathbf{x}^{(t+1)}$ is the design which reduces the most the uncertainty about $\mathbf{x}^*$, $f(\cdot)$'s minimizer. In Benassi et al. (2011), a fully Bayesian infill criterion was developed. More than the kriging prediction uncertainty, the latter also accounts for hyperparameter uncertainties while searching $\mathbf{x}^*$. This additionally improves their estimation and the precision of the surrogate. Convergence of Bayesian optimization algorithms has been proven for infill criteria which factorize in a certain form and respect some regularity and monotonicity conditions (Bect et al., 2016). This is the case for PI, EI, LCB, as well as for other acquisition functions. However, no convergence rate has been found for these heuristic methods.

Contrarily to Gaussian Processes, other types of surrogate models which are not accompanied by a built-in uncertainty quantification do not directly let themselves use for global optimization. Since they have no natural $s^2(\mathbf{x})$ measure of uncertainty, the EI or other infill criteria cannot be employed directly. Even though measures of uncertainty can eventually be derived by varying some hyperparameters (Snoek et al., 2015), or via bootstrap techniques, the ability to provide a prediction uncertainty is one major advantage of GPs for surrogate-based global optimization.

2.3 Multi-Objective Optimization

Often, the minimization of not only one objective is considered. Indeed, the worth of a design is frequently measured with several criteria and subject to constraints, which corresponds to a constrained multi-objective optimization problem. While the handling of constraints is rapidly discussed in Sections 2.4 and 5.2, the focus of this thesis is on multi-objective problems, introduced in this section,

$$\min_{\mathbf{x}\in X}(f_1(\mathbf{x}),\dots,f_m(\mathbf{x})). \tag{2.10}$$

2.3.1 Definitions

In (2.10), $f_j(\cdot)$, $j=1,\dots,m$ are the m real-valued objective functions. Since these goals are generally competing, there does not exist a single solution $\mathbf{x}^*$ minimizing every function in (2.10), but several trade-off solutions.

Definition 2.2. *(Domination). $\mathbf{a}\in\mathbb{R}^m$ is said to Pareto-dominate $\mathbf{b}\in\mathbb{R}^m$, $\mathbf{a}\preceq\mathbf{b}$, if and only if $a_j\leq b_j$ $\forall j=1,\dots,m$ and $a_i<b_i$ for at least one i.*
$\mathbf{b}\in\mathbb{R}^m$ is dominated by the set $\mathcal{A}\subset\mathbb{R}^m$ (written $\mathcal{A}\preceq\mathbf{b}$) if $\exists\mathbf{a}\in\mathcal{A}:\mathbf{a}\preceq\mathbf{b}$.
"$\preceq$" is a partial ordering since $\mathbf{a}\npreceq\mathbf{b}$ and $\mathbf{b}\npreceq\mathbf{a}$ may simultaneously occur. In strict Pareto dominance ("$\prec$"), $a_j<b_j$ must hold $\forall j$.

Definition 2.3. *(Non-domination). A solution $\mathbf{a}\in\mathcal{A}\subset\mathbb{R}^m$ is said to be non-dominated (ND) within $\mathcal{A}$ if there exists no $\mathbf{a}'\in\mathcal{A}$ such that $\mathbf{a}'\preceq\mathbf{a}$.*
$\mathbf{a}\in\mathbb{R}^m$ is non-dominated by the set $\mathcal{B}\subset\mathbb{R}^m$ (written $\mathcal{B}\npreceq\mathbf{a}$) if $\nexists\mathbf{b}\in\mathcal{B}:\mathbf{b}\preceq\mathbf{a}$.

Definition 2.4. *(Set-domination). The set $\mathcal{A}\subset\mathbb{R}^m$ is said to Pareto-dominate $\mathcal{B}\subset\mathbb{R}^m$ written $\mathcal{A}\preceq\mathcal{B}$ if and only if $\forall\mathbf{b}\in\mathcal{B},\exists\mathbf{a}\in\mathcal{A}:\mathbf{a}\preceq\mathbf{b}$*

Figure 2.8 illustrates these concepts. On the left, **A** and **B** dominate **C** since the latter belongs to their the dominance cone (upper right part of the objective space from these vectors). **A**, **B** and **D** are mutually non-dominated. Remark that **D** belongs to the non-dominated set even if it does not dominate **C** whereas **A** and **B** do. On the right, $\mathcal{A}\preceq\mathcal{B}$ since any red circle $\mathbf{b}$ is dominated by at least one black cross $\mathbf{a}\in\mathcal{A}$. $\mathcal{A}$ and $\mathcal{C}$ cannot be directly compared using the sole domination concept because $\mathcal{A}\npreceq\mathcal{C}$ ($\mathcal{A}\npreceq\mathbf{c}^1$) and $\mathcal{C}\npreceq\mathcal{A}$. They are non comparable. The same remark applies to $\mathcal{B}$ and $\mathcal{C}$ ($\mathcal{B}\npreceq\mathbf{c}^1$ or $\mathbf{c}^2$), even though some points of $\mathcal{C}$ are dominated by $\mathcal{B}$ ($\mathcal{B}\preceq\mathbf{c}^3$), and vice versa.

Definition 2.5. *(Pareto set). The optimal solutions to (2.10) form the Pareto set $\mathcal{P}_\mathcal{X}$. They correspond to an optimal compromise in the sense that it is not possible to find a competitor being better in all objectives simultaneously, $\mathcal{P}_\mathcal{X}=\{\mathbf{x}\in X:\nexists\mathbf{x}'\in X,\mathbf{f}(\mathbf{x}')\preceq\mathbf{f}(\mathbf{x})\}$, where $\mathbf{f}(\mathbf{x}):=(f_1(\mathbf{x}),\dots,f_m(\mathbf{x}))^\top$.*

Definition 2.6. *(Pareto front). The Pareto front $\mathcal{P}_\mathcal{Y}$ is the image of the Pareto set and contains only non-dominated solutions: $\mathcal{P}_\mathcal{Y}=\mathbf{f}(\mathcal{P}_\mathcal{X})=\{\mathbf{y}\in Y:\nexists\mathbf{y}'\in Y,\mathbf{y}'\preceq\mathbf{y}\}$, with*

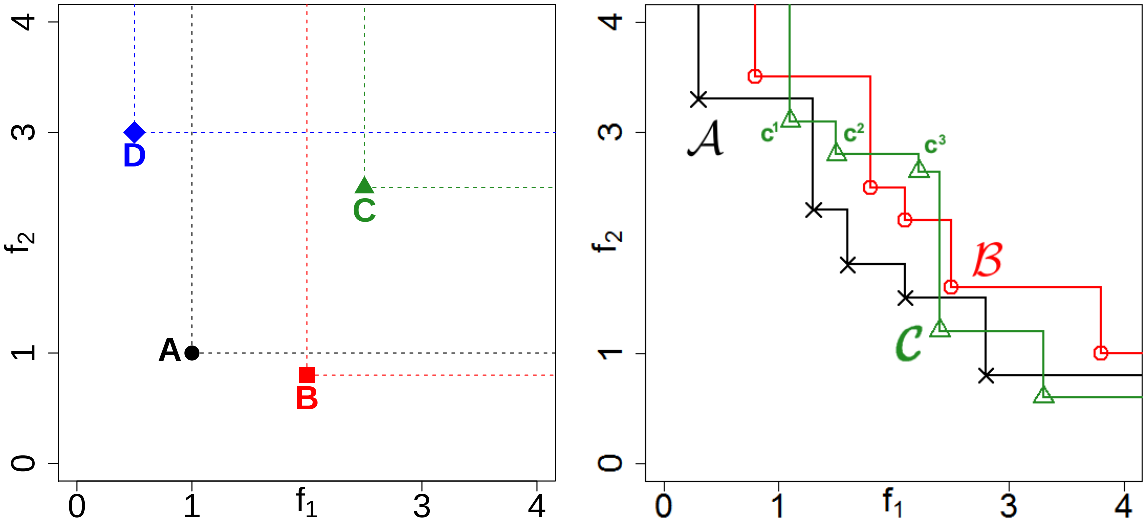


Figure 2.8: Domination relation among vectors (left) and among sets (right) in a problem with $m = 2$ objectives.

$Y = \mathbf{f}(X) \subset \mathbb{R}^m$ the image of the design space through the objectives called the objective space.

Definition 2.7. *(Ideal point). The Ideal point* **I** *of a Pareto front $\mathcal{P}_\mathcal{Y}$ is its component-wise minimum,* $\mathbf{I} = (\min\limits_{\mathbf{y}\in\mathcal{P}_\mathcal{Y}} y_1, \dots, \min\limits_{\mathbf{y}\in\mathcal{P}_\mathcal{Y}} y_m)$*.*

The Ideal point also corresponds to the vector composed of each objective function minimum. Obviously, there exists no $\mathbf{y}$ better in all objectives than the minimizer of objective j. As a consequence, the latter belongs to $\mathcal{P}_\mathcal{Y}$ and $\min\limits_{\mathbf{y}\in\mathcal{P}_\mathcal{Y}} y_j = \min\limits_{\mathbf{y}\in Y} y_j$, $j = 1, \dots, m$. **I** can therefore be alternatively defined as $(\min\limits_{\mathbf{x}\in X} f_1(\mathbf{x}), \dots, \min\limits_{\mathbf{x}\in X} f_m(\mathbf{x}))$. The decomposition on each objective does not hold for the Nadir point, which depends on the structure of the Pareto front:

Definition 2.8. *(Nadir point). The Nadir point* **N** *of a Pareto front $\mathcal{P}_\mathcal{Y}$ is the component-wise maximum of the Pareto front,* $\mathbf{N} = (\max\limits_{\mathbf{y}\in\mathcal{P}_\mathcal{Y}} y_1, \dots, \max\limits_{\mathbf{y}\in\mathcal{P}_\mathcal{Y}} y_m)$*.*

I and **N** are virtual points, that is to say that there generally does not exist an $\mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{I}$ or **N**. They are bounding points for the Pareto front, as every $\mathbf{y} \in \mathcal{P}_\mathcal{Y}$ is contained in the hyperbox defined by these points. Usually **N** is different from the maximum point **M**.

Definition 2.9. *(Maximum point). The Maximum point* **M** *of (2.10) is component-wise maximum of the objective space,* $\mathbf{M} = (\max\limits_{\mathbf{y}\in Y} y_1, \dots, \max\limits_{\mathbf{y}\in Y} y_m) = (\max\limits_{\mathbf{x}\in X} f_1(\mathbf{x}), \dots, \max\limits_{\mathbf{x}\in X} f_m(\mathbf{x}))$*.*

Definition 2.10. *(Extreme points). An extreme point for the j-th objective, $\boldsymbol{\nu}^j$, is an m-dimensional vector that belongs to the Pareto front, $\boldsymbol{\nu}^j \in \mathcal{P}_{\mathcal{Y}}$, and such that $\nu^j_j = N_j$. The Nadir point can thus be rewritten as $\mathbf{N} = (\nu^1_1, \ldots, \nu^m_m)$. A j-th extreme design point is $\boldsymbol{\xi}^j \in X$ such that $\mathbf{f}(\boldsymbol{\xi}^j) = \boldsymbol{\nu}^j$, i.e., $f_j(\boldsymbol{\xi}^j) = \nu^j_j = N_j$.*

The reader interested in additional concepts and theory in multi-objective optimization is referred to Collette and Siarry (2002); Deb (2001); Gal et al. (1999); Miettinen (1998); Sawaragi et al. (1985).

Figure 2.9 shows examples of Pareto fronts with $m = 2$ or 3 objectives. The Ideal point, Nadir point and the 2 (or 3) extreme points are also shown. The Pareto front is not necessarily convex nor continuous. Remark that in the case $m = 2$, $\nu^j_i = I_i, i \neq j$: the other coordinate of the j-th extreme point is the minimum in the other objective. This is generally not the case when $m > 2$; $\boldsymbol{\nu}^j$ has simply to be non-dominated in the remaining dimensions $\{1, \ldots, m\} \backslash \{j\}$.

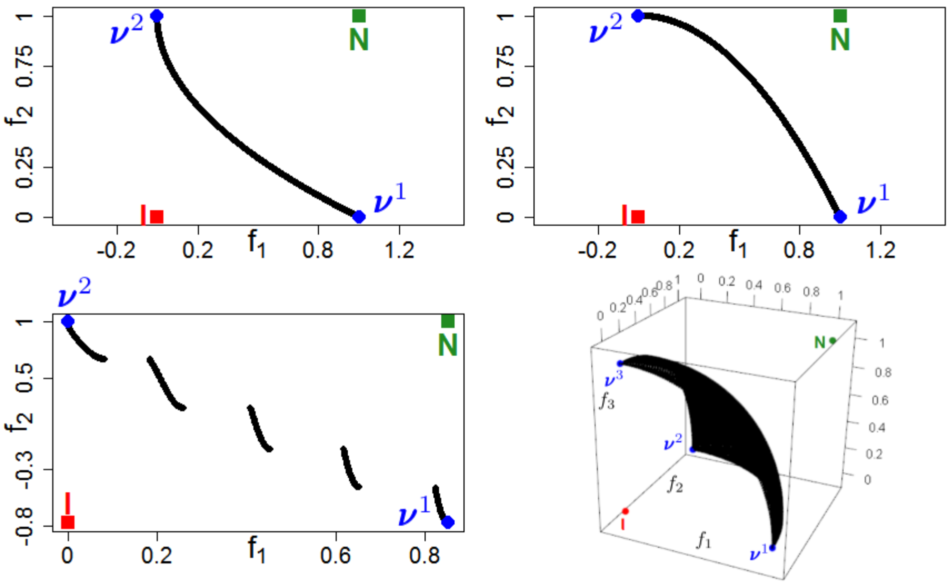


Figure 2.9: Example of Pareto fronts (black), Ideal (red square), Nadir (green square) and Extreme points (blue dots).

2.3.2 Performance metrics

Multi-objective optimizers aim at finding an approximation front $\widehat{\mathcal{P}_{\mathcal{Y}}}$ to $\mathcal{P}_{\mathcal{Y}}$ built upon the past observations $\mathbf{y}^{(1:t)} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)}\}$, $\widehat{\mathcal{P}_{\mathcal{Y}}} = \{\mathbf{y} \in \mathbf{y}^{(1:t)} : \nexists \mathbf{y}' \in \mathbf{y}^{(1:t)}, \mathbf{y}' \preceq \mathbf{y}\}$. The empirical Ideal, empirical Nadir, empirical Maximum and empirical extreme points are the Ideal, Nadir, Maximum and extreme points of $\widehat{\mathcal{P}_{\mathcal{Y}}}$, denoted $\overline{\mathbf{I}}$, $\overline{\mathbf{N}}$, $\overline{\mathbf{M}}$ and $\overline{\boldsymbol{\nu}}^j$, respectively (the $\widehat{\cdot}$ notation is kept for estimators of the true $\mathbf{I}$ and $\mathbf{N}$ in Chapter 4). $\widehat{\mathcal{P}_{\mathcal{Y}}}$ should come with some properties such as convergence or diversity. Comparing the performance of algorithms and the approximation fronts they return is not as easy as in the single-objective case, where the lowest value is a straightforward metric. Since the solutions of an approximation front $\mathcal{A}$ may generally contain solutions that dominate those of another approximation $\mathcal{B}$ and vice-versa (see right part of Figure 2.8), more sophisticated indicators need to be employed. In this section, common multi-objective performance metrics (Deb, 2001; Miettinen, 1998) which enable the comparison of two non-dominated sets $\mathcal{A}$ and $\mathcal{B} \subset \mathbb{R}^m$, are introduced.

Definition 2.11. *(Hypervolume indicator). The hypervolume indicator (Zitzler, 1999; Zitzler and Thiele, 1998, also known as $\mathcal{S}$-metric) $I_H(\mathcal{A};\mathbf{R})$ of a non-dominated set $\mathcal{A} \subset \mathbb{R}^m$ is the m-dimensional volume upper-bounded by a reference point $\mathbf{R} \in \mathbb{R}^m$, which is dominated by at least one $\mathbf{a} \in \mathcal{A}$, $I_H(\mathcal{A};\mathbf{R}) = Vol\left(\bigcup_{\mathbf{a}\in\mathcal{A}}\{\mathbf{z} : \mathbf{a} \preceq \mathbf{z} \preceq \mathbf{R}\}\right)$.*

The hypervolume indicator is a unary indicator since it does not need to be compared with another front or with a reference front. It complies with Pareto-dominance since $\mathcal{A} \preceq \mathcal{B} \Rightarrow I_H(\mathcal{A};\mathbf{R}) > I_H(\mathcal{B};\mathbf{R})$, whatever $\mathbf{R}$. However, I_H depends on the scaling of the objectives and on the reference point. If $I_H(\mathcal{A};\mathbf{R}) > I_H(\mathcal{B};\mathbf{R})$, there might exist an $\mathbf{R}'$ such that $I_H(\mathcal{B};\mathbf{R}') > I_H(\mathcal{A};\mathbf{R}')$, or a different scaling of the objectives which inverts the ordering relation (Knowles and Corne, 2002, 2003).

Definition 2.12. *(Additive ε-indicator). The additive ε-indicator (Zitzler et al., 2002) $I_{\varepsilon^+}(\mathcal{A},\mathcal{B})$ measures to what extent $\mathcal{A}$ needs to be improved to dominate $\mathcal{B}$. By denoting $\mathcal{A} - z\mathbf{1}_m = \{\mathbf{a} - z\mathbf{1}_m, \mathbf{a} \in \mathcal{A}\}$, the additive ε-indicator is $\mathbf{I}_{\varepsilon^+}(\mathcal{A},\mathcal{B}) = \min\limits_{\varepsilon \geq 0} \varepsilon : \mathcal{A} - \varepsilon\mathbf{1}_m \preceq \mathcal{B}$.*

I_{ε^+} is a binary measure since it compares $\mathcal{A}$ with $\mathcal{B}$. Except if $\mathcal{A} \preceq \mathcal{B}$ (or vice versa), in which case $I_{\varepsilon^+}(\mathcal{A},\mathcal{B}) = 0$, $I_{\varepsilon^+}(\mathcal{A},\mathcal{B}) > 0$ and $I_{\varepsilon^+}(\mathcal{B},\mathcal{A}) > 0$ simultaneously. $\mathcal{A}$ is better than $\mathcal{B}$ if $I_{\varepsilon^+}(\mathcal{A},\mathcal{B}) < I_{\varepsilon^+}(\mathcal{B},\mathcal{A})$. The additive ε-indicator can also be used in its unary version if a reference front (such as the true Pareto front of the problem), $\mathcal{R}$, is provided. In this case, $\mathcal{A}$ is better than $\mathcal{B}$ if $I_{\varepsilon^+}(\mathcal{A},\mathcal{R}) < I_{\varepsilon^+}(\mathcal{B},\mathcal{R})$. Since isotropic improvements $-\varepsilon\mathbf{1}_m$ are considered, the $f_j(\cdot)$'s should have the same magnitude.

Definition 2.13. *(Inverted Generational Distance). The Inverted Generational Distance (IGD, Coello and Cortés, 2005) measures the average distance between the closest point of a non-dominated set $\mathcal{A}$ to the points of a reference set $\mathcal{R}$, $IGD(\mathcal{A},\mathcal{R}) = \frac{1}{|\mathcal{R}|}\sqrt{\sum_{\mathbf{r}\in\mathcal{R}} \min\limits_{\mathbf{a}\in\mathcal{A}} \|\mathbf{r} - \mathbf{a}\|_2^2}$.*

It is a unary indicator which requires the knowledge of a reference set (e.g. the true Pareto front) and promotes the good coverage and convergence to $\mathcal{R}$. $\mathcal{A}$ is better than $\mathcal{B}$ if $\text{IGD}(\mathcal{A},\mathcal{R}) < \text{IGD}(\mathcal{B},\mathcal{R})$, even though the IGD is not monotonic with respect to the domination ordering.

Definition 2.14. *(Attainment time). The attainment time of a target $\mathbf{r} \in \mathbb{R}^m$ corresponds to the number of function evaluations required by an algorithm to dominate $\mathbf{r}$. Supposing the elements of $\mathcal{A}$ are sorted chronologically, $\mathcal{T}(\mathcal{A};\mathbf{r}) = \min\{i : \mathbf{a}^{(i)} \preceq \mathbf{r}\}$.*

Figure 2.10 illustrates these indicators. More multi-objective performance metrics exist and can be found e.g. in Collette and Siarry (2002).

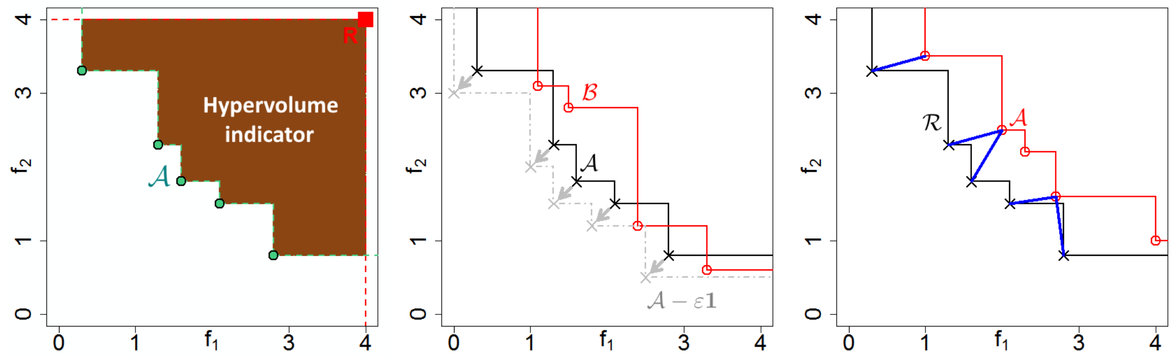


Figure 2.10: Left: hypervolume indicator (brown area) of a non-dominated set $\mathcal{A}$. Center: additive ε-indicator of $\mathcal{A}$ (black set) with respect to $\mathcal{B}$ (red set). Right: IGD indicator of $\mathcal{A}$ (red set), with respect to the reference set $\mathcal{R}$ (black set).

2.3.3 Standard techniques and Evolutionary Multi-Objective Optimization Algorithms

Standard techniques (Gal et al., 1999; Miettinen, 1998) consider the aggregation of objective functions via weights $w_j > 0$, $\sum_{j=1}^{m} w_j = 1$ and the optimization of $\sum_j w_j f_j(\mathbf{x})$. The solution to this problem is Pareto-optimal. Solving the scalarized problem for different convex combinations of the objectives provides a wider variety of solutions belonging to $\mathcal{P}_\mathcal{Y}$. These approaches are nonetheless capable of finding the whole Pareto front in problems where $\mathcal{P}_\mathcal{Y}$ is convex only, and the distribution of solutions may be poor. Adaptive methods (Kim and de Weck, 2005) exist to refine the Pareto front approximation in regions where it lacks of diversity. Other aggregations which come up with better properties are weighted Tchebycheff functions (Steuer and Choo, 1983).

Lexicographic methods solve the m problems sequentially in significance order. Each objective function $f_j(\cdot)$ is minimized under the constraint that the solution of all previous objectives $f_i(\cdot)$, $i < j$ is not worsened. In the same spirit, ε-constraint methods put constraints on all objectives but one which is optimized.

The Normal Boundary Intersection method (Das and Dennis, 1998) is a way to produce a well-distributed Pareto front. It consists in a series of single-objective optimizations starting from points located on the line (or hyperplane) joining anchor points (such as extreme points $\boldsymbol{\nu}^j$). The optimization is conducted from each starting point by solving a scalarized problem whose weights are adjusted to direct the optimization in the normal direction to the hyperplane.

Other methods to solve (2.10) can be found e.g. in Marler and Arora (2004); Miettinen (1998); Sawaragi et al. (1985).

Evolutionary Multi-Objective Optimization Algorithms (EMOA) are another class of techniques to solve (2.10). Being population-based methods, they are well-suited to the plurality of solutions and have proven their benefits for solving multi-objective problems (Coello et al., 2007; Deb, 2001). Various algorithms have been proposed. VEGA (Schaffer, 1985) alternates the objective to be optimized. SPEA2 (Zitzler et al., 2001) and NSGA-II (Deb et al., 2002) are dominance-based algorithms in the sense that they count the number of dominated solutions or rank the non-domination of solutions and incorporate this information in the fitness. Indicator-Based Evolutionary Algorithms such as SMS-EMOA (Beume et al., 2007), HypE (Bader and Zitzler, 2011), R2-IBEA (Phan and Suzuki, 2013) are driven by performance metrics such as those defined in Section 2.3.2 to assign the fitness of solutions and to conduct the optimization.

In the absence of a model to the objective functions, EMOAs are however not adapted to expensive objectives because they need a large number of function evaluations.

Preference incorporation in Multi-Objective Optimization

Targeting special parts of the objective space has been largely discussed within the multi-objective optimization literature, see for example Rachmawati and Srinivasan (2006) or Bechikh et al. (2015) for a review. Preference-based methods incorporate user-supplied information to guide the search towards specific parts of the Pareto front (Wierzbicki, 1980, 1999). The preference can be expressed either as an aggregation of the objectives (e.g., Bowman, 1976; Miettinen, 1998), or an aspiration level (also known as reference point) to be attained or improved upon (Wierzbicki, 1980), the distance to which is measured by a specific metric (e.g. L^1, L^2 or L^∞ norms). It can also appear as a ranking of solutions or objectives (Fonseca and Fleming, 1995), or via the modification of the dominance relation (Branke et al., 2004b).

For instance, in Wierzbicki (1980), achievement scalarizing problems are defined. They employ a user-provided reference point $\mathbf{R} \in \mathbb{R}^m$ which reflects some preferences and aim at minimizing $\max_{j=1,\dots,m} \frac{f_j(\mathbf{x})-R_j}{N_j-I_j} + \rho\sum_{j=1}^m \frac{f_j(\mathbf{x})-R_j}{N_j-I_j}$. Instead of the max, another $L^p, p < \infty$ norm can be used alternatively (Wierzbicki, 1999).

Using same ingredients, EMOAs have also been developed with the aim of taking preferences into-account (Bechikh et al., 2015; Deb and Sundar, 2006; Rachmawati and Srinivasan, 2006).

2.4 Bayesian Multi-Objective Optimization

To avoid the slow convergence of evolutionary algorithms, Bayesian methods have been extended to perform Efficient Global Optimization (Jones et al., 1998) in a multi-objective setting. In general, m GPs $Y_j(\cdot)$ are fitted to each objective $f_j(\cdot)$ independently even though different approaches such as the one described in Loshchilov et al. (2010), where the Pareto dominance relation is modeled by one surrogate, exist. Svenson (2011) has considered the m GPs to be (negatively) correlated in a bi-objective case, without noticing significant benefits. The GP framework enables both the prediction of the objective functions, $\widehat{y_j}(\mathbf{x})$, and the quantification of the uncertainties, $s_j^2(\mathbf{x}), \forall \mathbf{x} \in X$.

All Bayesian multi-objective methods conform to the outline of Figure 2.6, excepted that m surrogates $Y_1(\cdot), \ldots, Y_m(\cdot)$ and m objective functions are now considered, and that an empirical Pareto set $\widehat{\mathcal{P}_{\mathcal{X}}}$ and Pareto front $\widehat{\mathcal{P}_{\mathcal{Y}}}$ are returned. As in the single-objective case, the problem is cast into the sequential optimization of an acquisition function used for determining $\mathbf{x}^{(t+1)} \in X$, the most promising next iterate to be evaluated. In some approaches, the m surrogates are aggregated or use an aggregated form of EI (Jeong and Obayashi, 2005; Knowles, 2006; Liu et al., 2007; Zhang et al., 2009). Other methods use a multi-objective infill criterion relying on $\mathbf{Y}(\cdot) := (Y_1(\cdot), \ldots, Y_m(\cdot))^\top$ for taking into account all the metamodels simultaneously (Wagner et al., 2010). This is the case of the Expected Hypervolume Improvement (EHI, also called EHVI, Emmerich et al., 2005, 2011, 2006), the Expected Maximin Improvement (EMI, Svenson, 2011; Svenson and Santner, 2010), and Keane (2006)'s Euclidean-based improvement, multi-objective infill criteria that boil down to EI when facing a single objective. $\mathcal{S}$-metric selection (SMS, Ponweiser et al., 2008) is based on an LCB strategy, and SUR (Picheny, 2015) considers the stepwise uncertainty reduction on the Pareto set. These infill criteria aim at providing new non-dominated points while balancing exploitation and exploration, and eventually have the goal of approximating the Pareto front entirely. Most acquisition functions redefine the single-objective improvement over the best solution by the increase of a multi-objective metric (see Section 2.3.2): in EHI and SMS a growth of the hypervolume indicator is considered, while EMI focuses on the expected growth of the additive ε-indicator. Contrarily to the EI, these acquisition functions and/or their gradient are not necessarily known in closed-form, which complicates their computation and maximization. The hypervolume-based criteria additionally suffer from the exponentially growing complexity of the latter in the number of objectives. Being the basis for the multi-objective infill criterion developed in Chapter 4, more details about EHI (Emmerich et al., 2006; Emmerich and Klinkenberg, 2008), one of the most popular Bayesian multi-objective infill criterion, as well as adaptations which enable it to incorporate preferences and to target parts of the objective space will be given in Chapter 4.

Example 2.3. *(Bayesian Multi-Objective Optimization).*

The following simple example in dimension $d = 1$ and with $m = 2$ objectives gives insights into EHI's logic: $\min_{x\in[0,1]} (f_1(x), f_2(x))$ where $f_1(x) = 0.6x^2 - 0.24x + 0.1$ and $f_2(x) = x^2 - 1.8x + 1$. Both minima are respectively 0.2 and 0.9. The multi-objective optimality conditions (Miettinen, 1998) show that the Pareto set is $\mathcal{P}_{\mathcal{X}} = [0.2, 0.9]$ and the Pareto front $\mathcal{P}_{\mathcal{Y}} = \{\mathbf{y} = (f_1(x), f_2(x))^\top, x \in [0.2, 0.9]\}$. The functions are sampled at $x^{(1:3)} = \{0.1, 0.5, 0.9\}$ (black dots on the left plot of Figure 2.11). The EHI is maximal at $x^* = 0.67$ (right plot of Figure 2.11). This design leads to the largest expected growth of the hypervolume indicator. In the left plot, the prediction of $Y_1(\cdot)$ and $Y_2(\cdot)$ at x^* are shown (blue cross), as well as the hypervolume increase brought by $(\widehat{y_1}(x^*), \widehat{y_2}(x^*))^\top$. The prediction uncertainty $(s_1(x^*), s_2(x^*))$ is also shown (blue ellipse); the latter is taken into account since the expectation of the hypervolume improvement is considered.

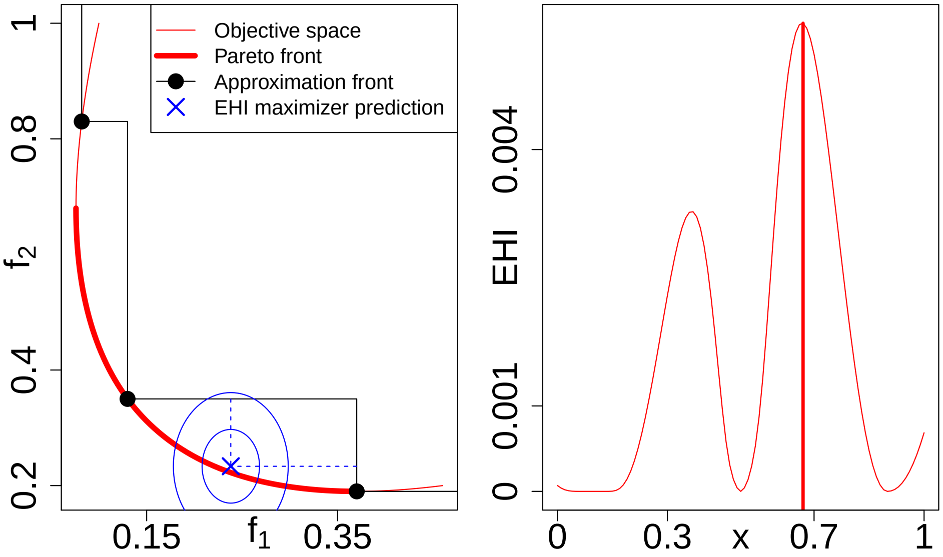


Figure 2.11: Example of Bayesian Multi-Objective Optimization with EHI. 3 designs (left figure, black dots) have been observed. EHI is maximal at $x = 0.67$ (right figure) where the expected growth of the hypervolume indicator is the largest.

Handling constraints in Bayesian optimization

A closely related problem is that of (mono-objective) constrained optimization,

$$\min_{\substack{\mathbf{x}\in X \\ g_1(\mathbf{x})\leq 0 \\ \vdots \\ g_{m_c}(\mathbf{x})\leq 0}} \quad f(\mathbf{x}) \tag{2.11}$$

where $g_j(\cdot)$, $j = 1, \dots, m_c$ are m_c continuous expensive-to-evaluate constraints the design has to satisfy. Here, we only consider inequality constraints, $g_j(\cdot) \leq 0$. Gaussian Processes cannot account for equality constraints directly, which deserve special attention (Picheny et al., 2016). Binary constraints can be handled through classifiers (Rasmussen and Williams, 2006; Vapnik, 1995). The issue of non-evaluable designs for which $f(\mathbf{x}) = \emptyset$ (e.g. non-convergence or crash of the simulation) is a different topic discussed in Bachoc et al. (2019); Basudhar et al. (2012); Sacher et al. (2018).

Standard approaches to solve (2.11) (Coello Coello, 2016; Forrester and Keane, 2009; Gardner et al., 2014; Parr et al., 2012a,b; Schonlau, 1997) rely on independent[4] GPs $Y(\cdot)$ for $f(\cdot)$ and $G_j(\cdot)$ for $g_j(\cdot)$, $j = 1, \dots, m_c$. The probabilistic framework of GPs and the independence enables the calculation of the probability of feasibility (PoF):

[4] which may be an excessively strong assumption in cases such as box constraints, $g_1(\mathbf{x}) \leq b, g_2(\mathbf{x}) = -g_1(\mathbf{x}) \leq a$ (Jiao et al., 2019).

$$\mathrm{PoF}(\mathbf{x}) = \mathbb{P}(G_1(\mathbf{x}) \leq 0, \ldots, G_{m_c}(\mathbf{x}) \leq 0) = \prod_{j=1}^{m_c} \mathbb{P}(G_j(\mathbf{x}) \leq 0) = \prod_{j=1}^{m_c} \mathrm{PI}_j(\mathbf{x}; 0) \tag{2.12}$$

where $\mathrm{PI}_j(\mathbf{x}; 0)$ is the Probability of Improvement over the value 0 for the GP $G_j(\cdot)$, defined in (2.7). The infill criterion needs to be adapted to cope with the constraints. The EI (2.8) can be maximized under a constraint of feasibility (Sacher et al., 2018; Schonlau, 1997), $\max\limits_{\substack{\mathbf{x} \in X \\ \mathrm{PoF}(\mathbf{x}) \geq \beta_p}} \mathrm{EI}(\mathbf{x})$, where β_p is a feasibility threshold. Another option is to maximize a constrained infill criterion, $\max\limits_{\mathbf{x} \in X} \mathrm{EIPF}(\mathbf{x})$ where EIPF is the product of both criteria, $\mathrm{EIPF}(\mathbf{x}; a) = \mathrm{EI}(\mathbf{x}; a) \times \mathrm{PoF}(\mathbf{x})$. Under the independence hypothesis, the justification is it equals $\mathbb{E}[(a - Y(\mathbf{x}))_+ \mathbb{1}_{G_1(\mathbf{x}) \leq 0, \ldots, G_{m_c}(\mathbf{x}) \leq 0}]$: it is the expectation of a constrained improvement function which equals 0 if $\mathbf{x}$ is unfeasible. Similarly to the EI, PoF's gradient has closed-form expression which is advantageous for EIPF's maximization.

In constrained multi-objective problems, the multiplication of a multi-objective infill criterion (e.g., EHI) with PoF is commonplace (Feliot, 2017; Feliot et al., 2017; Hussein and Deb, 2016; Parr, 2013; Singh et al., 2014). The multi-objective infill criterion developed in Chapter 4 which is extended to constrained problems in Section 5.2 follows this logic too. The targeting properties and fast attainment it exhibits may also be appealing property for highly-constrained multi-objective problems (Feliot, 2017; Jiao et al., 2019, 2018) introduced in Section 5.2.2, where finding feasible designs is challenging.

Chapter 3

MetaNACA: a practical aerodynamic test bed for multi-objective optimizers

Contents

When designing algorithms, it is worth benchmarking and enhancing them on the class of problems for which they are intended (Stork et al., 2020). The "no free lunch theorem" (Ho and Pepyne, 2002; Wolpert et al., 1997) states that it is not possible to design a method which performs best on any problem. Instead of evaluating our techniques only on artificial multi-objective test functions (Deb et al., 2005; Zitzler et al., 2000), which may contain irrelevant features in comparison with the considered physical objective functions (Stork et al., 2020), we aim at testing our algorithms on "real-world like" multi-objective shape optimization problems. The developed algorithms should behave well on the class of physical functions (which should be quite regular), with parametric designs (see Chapter 1). For benchmarking purposes and extensive comparison, the evaluation time of test problems should nonetheless be negligible, contrarily to real cases where evaluating the simulator is cumbersome.

Pursuing these goals for designing and comparing well-suited algorithms (Stork et al., 2020), we have built the "MetaNACA" test suite. It stems from 2D aerodynamic simulations and returns the lift coefficient (Cz) and the drag coefficient (Cx) of a parameterized airfoil. The evaluation time of such a simulation is relatively small (20 minutes) but nonetheless prohibitive for conducting a large amount of simulations and optimization experiments. Therefore, using (adaptive) DoE techniques (Gramacy and Lee, 2009), we have fitted GPs to a large number of simulations. The created metamodel is an emulator of both physical quantities and used as a benchmark problem. The approach of creating a surrogate to a computer code to benchmark algorithm was already pursued in the MOPTA test case for constrained optimization (Jones, 2008) and in Eggensperger et al. (2015) for hyperparameters optimization.

To study the effect of the dimension d and of the number of objectives m, modified parameterizations of the airfoil, as well as additional objective functions have been designed. The final test bed has instances in $d = 3, 8, 22$ dimensions, with $m = 2, 3$ or 4 objectives to be optimized simultaneously.

3.1 NACA airfoil and aerodynamic simulation

NACA profiles (Anderson Jr, 1984) are airfoil shapes described by 4 digits. They correspond to three parameters, M, P and T which define the geometry of a shape. M is the maximum of camber, P the position of this maximum, and T the thickness of the airfoil. The parametric shapes $\{(x_u, y_u), (x_l, y_l)\}$ of the upper part (extrados) and of the lower part (intrados) depend on (M, P, T) and are calculated analytically (Jacobs et al., 1933). A NACA profile is shown in Figure 3.1.

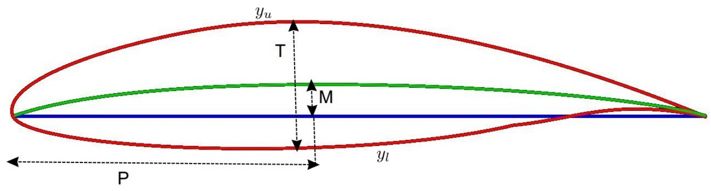


Figure 3.1: Example of a NACA airfoil with parameters (M, P, T).

The flow around the profile is simulated using a commercial Computational Fluid Dynamics (CFD) software. The Reynolds Averaged Navier Stokes (RANS) equations with a $k - \varepsilon$ turbulence model (Launder and Spalding, 1983) and standard wall functions are solved. The angle of attack is $\alpha_I = 0°$, the chord $c = 960$mm and the freestream velocity V_∞ is 40m/s to perform the simulation at a typical Reynolds number for aeronautics. The NACA profile is placed at 10 chord lengths upstream, 20 chord lengths downstream and 15 chord lengths of the top and the bottom walls of a fixed domain meshed in approximately 100,000 cells. The boundary layer of the NACA is composed of 10 unstructured layers of quadratic elements in order to capture the essential physical phenomena near the airfoil. A second order finite volume scheme returns the pressure and velocity field of the flow around the airfoil, from which the lift and drag forces are computed (Anderson Jr, 1984) and averaged over the converged last 30 iterations[1], to obtain the drag coefficient (Cx)

[1] among 1500 iterations.

and lift coefficient (Cz) of the NACA profile.

The NACA simulator is a typical blackbox function: a parametric design $\mathbf{x} = (M, P, T)^\top \in \mathbb{R}^3$ is given, and two physical outputs $f_1 \equiv Cz$ and $f_2 \equiv Cx$ are returned. ∇f_1 and ∇f_2 are unknown. For these reasons, it belongs to the class of optimization problems considered throughout this thesis. The multi-objective optimization of the lift and drag coefficient of an airfoil is a typical real-world problem, which was also considered for benchmarking optimizers in Yang et al. (2019c).

3.1.1 Towards higher-dimensional shapes

Typical parametric shapes may nonetheless hinge on a larger number of dimensions d. We therefore aim at creating NACA airfoils parameterized by more than 3 variables. To this aim, we design the "NACA 8" and "NACA 22" shapes in $d = 8$ and $d = 22$ dimensions, respectively. The latter are modifications of the standard NACA, to which small bumps are added. More precisely, the 5 and 19 additional parameters are heights of evenly distributed perturbations along the airfoil, $L_i > 0$, which modify the shape. Figure 3.2 shows a NACA 8 shape (left) and a NACA 22 shape (right). The dotted line is the original NACA profile, and L_i the size of the i-th bump. To keep the shapes smooth, a spline is fitted to the L_i's corresponding to the extrados and added to y_u. Likewise, a second spline is fitted to the L_i's corresponding to the intrados and subtracted from y_l.

The magnitude of these bumps is smaller than the typical dimensions of a NACA airfoil, $L_i \in [0.1, 15]$ mm while the airfoil's chord (blue line on Figure 3.1) is 960mm. The M, P, T parameters therefore have a larger impact on the shape than the L_i's, as can be seen at the bottom of Figure 3.2. This is a common setting in real-world Computer Aided Design (CAD) shapes: some parameters such as the length, the width, etc., are a macro description of the shape, while others correspond to smaller refinements.

The NACA simulator is extended to both cases and returns the Cx and Cz of an airfoil parameterized by $d = 8$ or $d = 22$ parameters.

3.1.2 Additional objective functions

A typical multi-objective optimization problem is the simultaneous maximization of the lift coefficient and minimization of the drag coefficient of the NACA airfoil,

$$\min_{\mathbf{x} \in X} \left(f_1(\mathbf{x}), f_2(\mathbf{x})\right) \tag{3.1}$$

where $X \subset \mathbb{R}^d$ is a hypercubic domain depending on the considered instance of the NACA airfoil, $f_1(\mathbf{x}) = -Cz(\mathbf{x})$ (a minus sign is employed to maximize the lift coefficient), $f_2(\mathbf{x}) = Cx(\mathbf{x})$.

Bi-objective problems are nonetheless not the only class of problems we are interested in. Moreover, some properties of multi-objective problems (see Chapter 2) slightly differ whether $m = 2$ or $m > 2$. We therefore look for additional criteria to increase the dimension of (3.1) in terms of objectives.

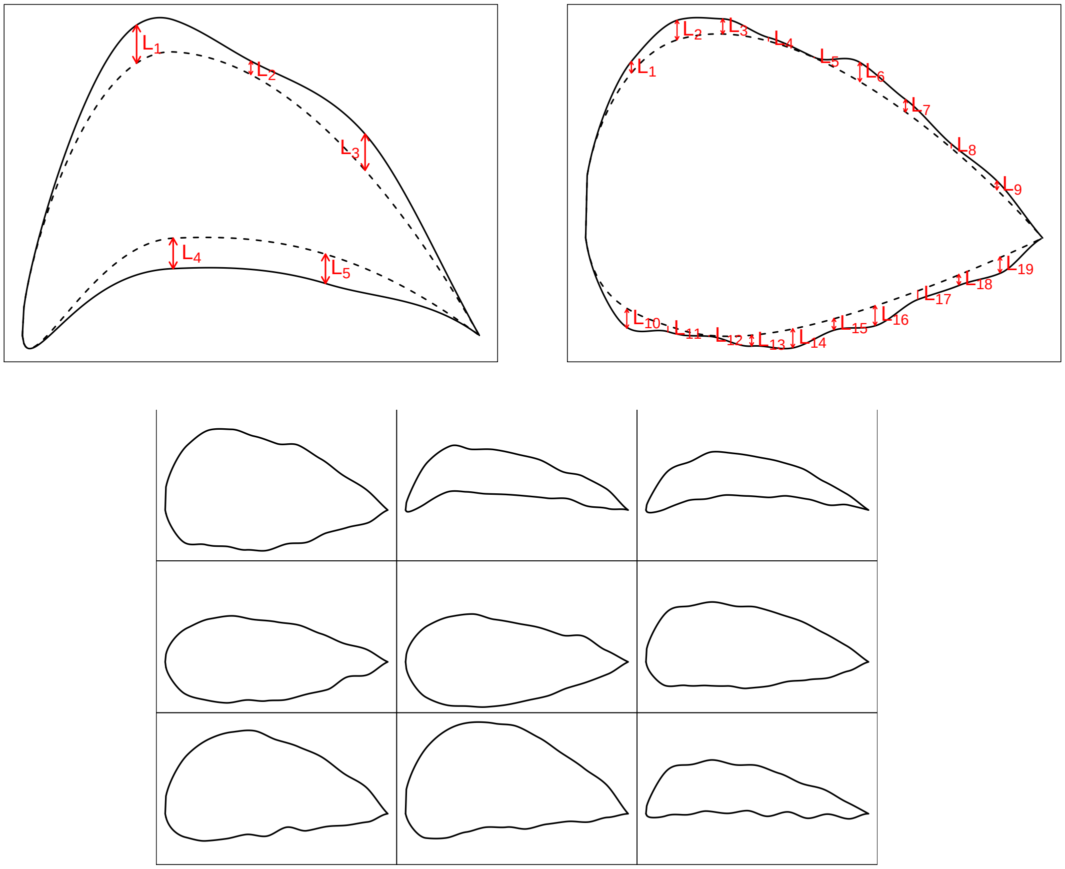


Figure 3.2: Top: example of a NACA 8 profile (left) and of a NACA 22 profile (right). Bottom: nine different NACA 22 airfoils corresponding to different parameterizations $\mathbf{x} = (M, P, T, L_1, \ldots, L_{19})^\top$.

Two additional aerodynamic objectives are defined. They are the drag coefficient Cx and the lift coefficient Cz at a different angle of incidence with the chord, $\alpha_I = 8°$, shown in Figure 3.3.

$f_3(\mathbf{x}) = -Cz(\mathbf{x})|_{\alpha_I=8^\circ}$ and $f_4(\mathbf{x}) = Cx(\mathbf{x})|_{\alpha_I=8^\circ}$ are obtained by running the NACA simulator on a design $\mathbf{x} \in \mathbb{R}^d$ where the aerodynamic flow around the rotated airfoil (Figure 3.3) is simulated. Notice that these functions are highly correlated to $f_1(\mathbf{x}) = -Cz(\mathbf{x})|_{\alpha_I=0^\circ}$ and to $f_2(\mathbf{x}) = Cx(\mathbf{x})|_{\alpha_I=0^\circ}$, respectively. This is nonetheless not an issue, since real-world problems with *many* objectives may consider similar (or even the same) objective functions under different operating conditions. The correlation in objectives may also open discussions about the necessity of all objectives, and about objective space dimension reduction.

These extensions of the original NACA problem enable to define multi-objective problems with $d = 3, 8, 22$ parameters and with $m = 2, 3, 4$ objectives.

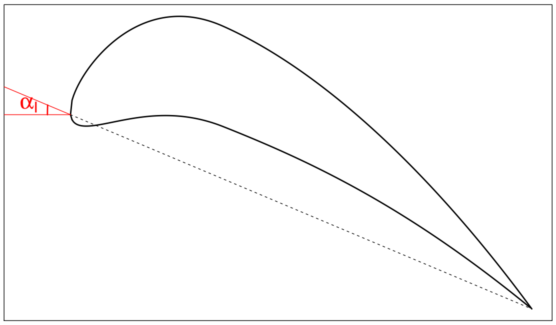


Figure 3.3: NACA airfoil with angle of attack $\alpha_I = 8°$.

3.2 MetaNACA: a metamodel of the NACA problems

The benefits of the NACA problem reside in its parametric shape associated to multiple physical objective functions, with tunable dimension and number of objectives. Our interest in using it is the investigation and the benchmarking of algorithmic behaviors and settings, since it is a typical representative of the class of problems we eventually aim at solving.

The $f_j(\cdot)$'s defined in Section 3.1 evaluated by the NACA simulator with inputs in $X \subset \mathbb{R}^d$ could be directly used for this purpose. To further accelerate the evaluation time of the objective functions, surrogates to the $f_j(\cdot)$'s, $\widehat{f_j}(\cdot)$, $j = 1, \ldots, m$, will be used instead in the optimization experiments. Once built, the evaluation time of $\widehat{f_j}(\mathbf{x})$ will be negligible (smaller than 1s). Remark the assumption of Bayesian optimization is verified by considering the $\widehat{f_j}(\cdot)$'s: the functions to optimize are the realization of a Gaussian Process.

The two following parts of this section detail the construction of the surrogate models, called “MetaNACA”.

3.2.1 Design of Experiments

True evaluations of the NACA simulator are required for building the MetaNACAs. To substitute the NACA by an emulator, the surrogate model should be as accurate as possible. To this aim, in reason of the moderate cost of the computer code and of the possibility of distributing calculations on different processors, we build a large Design of Experiments (DoE) of $N_{DOE} \approx 1000$ designs for each dimension of the NACA.

To enhance the predictivity of $\widehat{f_j}(\cdot)$, the DoE $\mathbf{x}^{(1:N_{DOE})} \subset X$ needs to be space-filling. In reason of the curse of dimensionality (Bellman, 1961), 1000 points in $X \subset \mathbb{R}^3$ cover the space much better than in $\mathbb{R}^8$ or in $\mathbb{R}^{22}$. Two different strategies are therefore completed regarding d.

3.2.1.1 Factorial design in dimension 3

In dimension $d = 3$, the objective functions are evaluated on a factorial design (Fang et al., 2005): the $f_j(\cdot)$'s are computed on a regular grid of $10^3 = 1000$ designs. X is a hyper-rectangular domain, with bounds for M, P, T [0,0.09], [0.1,0.5] and [0.05,0.25]. They correspond to the maximum camber, position of this maximum, and maximum thickness, divided by the chord, respectively. These values are normalized the $[0,1]^3$ hypercube, and the grid values of the factorial design are $x_j = 0.05, 0.15, \dots, 0.95$, in each dimension $j = 1, \dots, d$.

Next, the $\widehat{f}_j(\cdot)$'s are simply the mean predictor of a GP fitted to the 1000 $(\mathbf{x}^{(i)}, f_j(\mathbf{x}^{(i)}))$, $i = 1, \dots, N_{DOE}$ observations (see Chapter 2). To avoid issues related to the magnitude of the objective functions, the observations $f_j(\mathbf{x}^{(i)})$ are centered and scaled to unit variance before.

The 1000 observations and the empirical Pareto front are shown in black in Figure 3.4. An evolutionary algorithm (NSGA-II, Deb et al., 2002) was applied to obtain the Pareto front and the Pareto set of the $\min\limits_{\mathbf{x} \in X} (\widehat{f}_1(\mathbf{x}), \widehat{f}_2(\mathbf{x}))$ problem. $\mathcal{P}_{\mathcal{Y}}$ is shown in red on the left, and $\mathcal{P}_{\mathcal{X}}$ is shown in the design space on the right.

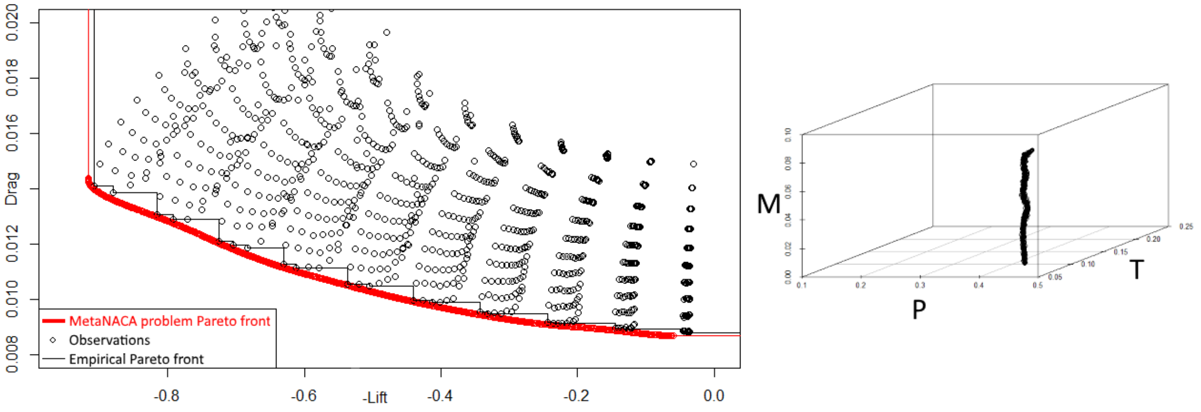


Figure 3.4: MetaNACA 3 DoE. Left: observed values (circles) and Pareto front (red) of the MetaNACA 3 problem. Right: Pareto set of the MetaNACA problem.

3.2.1.2 Space-filling LHS and adaptive infilling in dimension 8 and 22

Evaluating 10^8 or 10^{22} designs is not affordable. 1000 points in a 8 or 22 dimensional design space may nonetheless not reflect enough of the $f_j(\cdot)$'s features, and we could potentially miss some fluctuations.

A two step strategy is therefore considered for the construction of the MetaNACA 8 and MetaNACA 22. First, an LHS design (McKay et al., 1979) of size $N_{DOE} = 1000$ optimized by the maximin criterion (Pronzato, 2017), is evaluated. At this step, each $f_j(\cdot)$ is normalized (so that the N_{DOE} sample mean is 0 and has variance 1) and a GP $\widehat{f}_j(\cdot)$ is fitted to $(\mathbf{x}^{(i)}, f_j(\mathbf{x}^{(i)}))_{i=1,\dots,N_{DOE}}$.

Following, an adaptive infill criterion is used for enriching the DoE with 100 supplementary designs. Rather than using a predictivity-oriented acquisition function which aims at improving the quality of $\widehat{f_j}(\cdot)$ in the whole X, we use a multi-objective Bayesian optimization infill criterion (see Section 2.4). The MetaNACA test functions being designed for multi-objective problems, optimization experiments are likely to ask for the evaluation of designs related to Pareto-optimality. It is worth improving $\widehat{f_j}(\cdot)$ in good trade-off areas of X, rather than in an arbitrary undersampled part of X. A classical multi-objective Bayesian optimization procedure (Figure 2.6) is applied to $(f_1(\mathbf{x}), f_2(\mathbf{x}))$ and to $(f_3(\mathbf{x}), f_4(\mathbf{x}))$[2]: given the surrogate models $\widehat{f_j}(\cdot)$, 100 new designs $\mathbf{x}^* \in \mathbb{R}^d$ are sequentially promoted and evaluated by the NACA simulator. The metamodels $\widehat{f_j}(\cdot)$ are updated at each iteration. Figure 3.5 depicts this procedure.

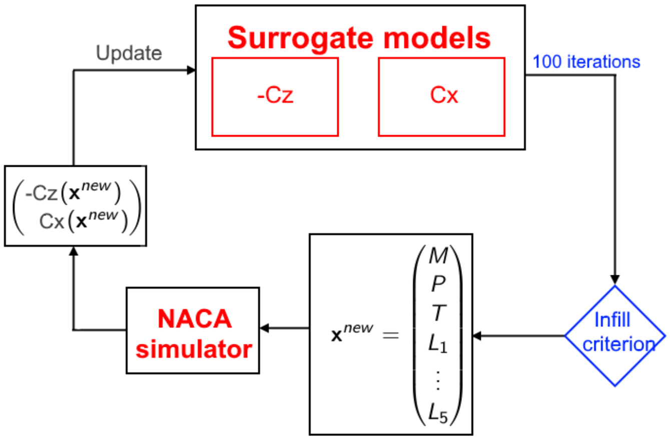


Figure 3.5: Sequential infill procedure for the construction of the MetaNACA 8.

Finally, to prevent from any artifact of alignment in the high-dimensional design space due to some space-filling properties, 100 randomly chosen designs $\mathbf{x}^{(i)} \sim \mathcal{U}(X)$ are evaluated. While these last points will help in improving the accuracy, they are mainly useful in removing any artificial periodicity in the design space due to space-filling properties which might hinder the estimation of correlation parameters. The final surrogate model is built over these 1200 evaluations, and its predictive capabilities have been enhanced in optimal parts of X.

Figure 3.6 highlights the benefits of this procedure for the MetaNACA 8 and 22 on the $(f_1(\cdot), f_2(\cdot))$ problem. The blue dots are the objective values of the starting DoE (N_{DOE} designs). The red triangles correspond to both the sequential and the random infills.

[2] such a procedure selects designs which are promising in the $(f_1(\mathbf{x}), f_2(\mathbf{x}))$ and in the $(f_3(\mathbf{x}), f_4(\mathbf{x}))$ bi-objective problems. Applying the routine to $(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x}))$ could have been better for choosing designs which are critical for all 4 objectives simultaneously. Our approach is justified by the fact that the $\widehat{f_j}(\cdot)$ metamodels at $\alpha_I = 0°$ and at $\alpha_I = 8°$ were not created at the same moment. Eventually, since the objectives are correlated, the metamodels have been enriched in similar regions of the design space. Last but not least, this infill criterion is devoted to bi-objective optimizations which are widely investigated.

These red points have clearly enhanced the empirical Pareto front. The Pareto fronts of $\min_{\mathbf{x}\in X} (\widehat{f}_1(\mathbf{x}), \widehat{f}_2(\mathbf{x}))$ found by NSGA-II are shown in both problems. The blue one is the Pareto front of the $\widehat{f}_j(\cdot)$'s fitted to 1000 designs only, and the red one corresponds to the Pareto front of the $\widehat{f}_j(\cdot)$'s fitted to all 1200 designs. The latter is much more accurate because the metamodel has gained in precision in Pareto optimal parts of X. In the $d = 22$ case (right), it has clearly corrected the over-optimistic estimation of the Pareto front.

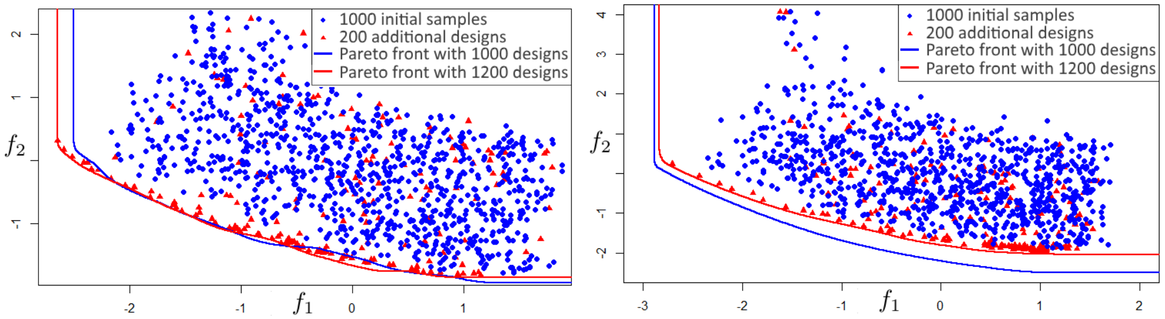


Figure 3.6: Observations and estimated Pareto front (obtained by means of an Evolutionary Algorithm applied to the predictor) of the MetaNACA fitted to the 1000 first points (blue), or to all 1200 points (red). Left $d = 8$, right $d = 22$.

Remark 3.1. *Depending on the designs and on the instance of the NACA airfoil, some simulations did not converge, led to outliers, or even crashed. These simulations were of course not taken into account for building the surrogate models, which in the end contain slightly less than 1000 or 1200 observations.*

3.2.2 Validation and Pareto front analysis

3.2.2.1 Validation

To substitute the true NACA simulators by the MetaNACAs for the purpose of optimization experiments, the latter need to be validated. This is achieved by means of prediction on a different test set and leave-one-out cross validation. Comparison metrics include the Normalized RMSE (NRMSE), the Mean Absolute Error and the coefficient of determination R2. For the sake of brevity, only the leave-one-out R2 of $\widehat{f}_1(\cdot)$ and $\widehat{f}_2(\cdot)$ are given (for $d = 3, 8, 22$) in Table 3.1, because the same conclusions were obtained from other indicators and for the other MetaNACAs. By denoting $y^{(i)}$ the i-th observation, $\bar{y} = \frac{1}{N_{DOE}} \sum_{i=1}^{N_{DOE}} y^{(i)}$ (N_{DOE} it the total number of simulations ≈ 1200), and $\widehat{y}^{(-i)}(\mathbf{x}^{(i)})$ the leave-one-out prediction at $\mathbf{x}^{(i)}$, the R2 is defined as

$$\mathrm{R2} = 1 - \frac{\sum_{i=1}^{N_{DOE}} (y^{(i)} - \widehat{y}^{(-i)}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^{N_{DOE}} (y^{(i)} - \bar{y})^2}. \tag{3.2}$$

d	$\widehat{f_1}(\cdot)$	$\widehat{f_2}(\cdot)$
3	0.99984	0.98822
8	0.99985	0.98634
22	0.99708	0.97699

Table 3.1: Leave-one-out R2 coefficient for $\widehat{f_1}(\cdot)$ and $\widehat{f_2}(\cdot)$, MetaNACAs in dimension $d = 3, 8, 22$.

Figure 3.7 details the leave-one-out residuals and shows a QQ-plot for $d = 8$. As confirmed by the latter, the model correctly predicts unobserved designs and the standardized residuals $r^{(i)} := \frac{y^{(i)}-\widehat{y}^{(-i)}(\mathbf{x}^{(i)})}{s^{(-i)}(\mathbf{x}^{(i)})}$, where $s^{2(-i)}(\mathbf{x}^{(i)})$ is the leave-one-out variance at $\mathbf{x}^{(i)}$, are normally distributed, in accordance with the theory (Rasmussen and Williams, 2006).

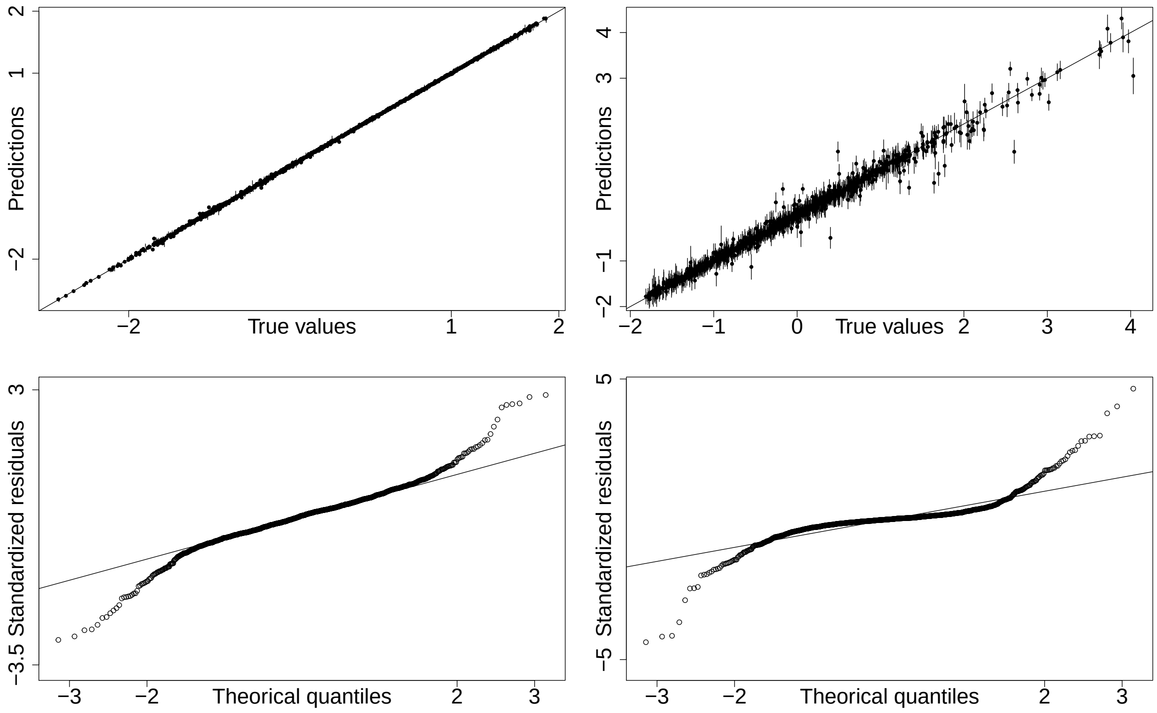


Figure 3.7: Leave-one-out predictions and 95% confidence interval (top) and QQ-plots (bottom) to validate the MetaNACA 8. Left: negative lift, right: drag.

The R2 confirms the excellent goodness-of-fit. The same conclusions were obtained by validation on additional test designs. The MetaNACAs are therefore an accurate enough emulator of the NACA simulator and can be used as a substitute to benchmark optimizers.

Since they have second-order importance, it was analyzed whether the additional L_i variables could simply be ignored. If yes, the MetaNACA would no longer be a $d = 8$

or $d = 22$ dimensional problem. Validation indicators which take the complexity (i.e. dimension) of the metamodel into account, such as the adjusted R2 coefficient (Draper and Smith, 1998), nonetheless favored the MetaNACA in dimension $d = 8$ or $d = 22$ over metamodels considering x_1, x_2, x_3 only; these L_i's are not dummy variables which can be disregarded.

An interesting validation test related to the DoE enriching (Section 3.2.1.2) is the prediction at the 200 additional points by the "initial DoE MetaNACA", i.e. the $\widehat{f}_j(\cdot)$'s before incorporation of these points. As was already pointed out by the blue Pareto frontier in the right plot of Figure 3.6, the initial metamodel was too optimistic. This is confirmed by the validation test shown in Figure 3.8. For small values (lower left part) of the negative lift (left plot) or drag (right plot), the designs are consistently predicted lower than their true value. This highlights the benefits of enriching the DoE in Pareto-optimal regions, and the effect of higher dimensional input spaces.

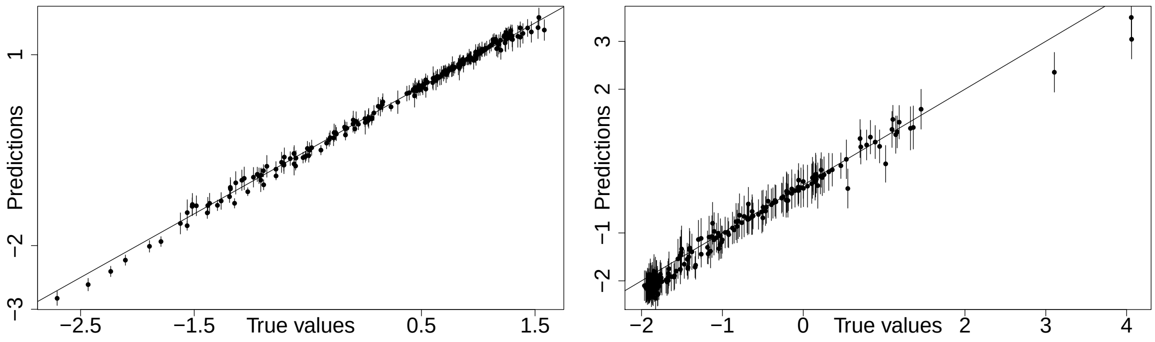


Figure 3.8: Evaluation of the "initial MetaNACA 22" built over the first 1000 points, on the 200 additional designs (to be further incorporated in this model). Predicted values and 95% confidence bounds are plotted against the true value of these designs. Left: negative lift, right: drag.

3.2.2.2 MetaNACA Pareto front

Since the MetaNACA emulators have a negligible execution time, for all dimensions $d = 3, 8, 22$, the Pareto fronts of the problems $\min_{\mathbf{x} \in \mathbb{R}^d} (\widehat{f}_1(\mathbf{x}), \widehat{f}_2(\mathbf{x}))$, $\min_{\mathbf{x} \in \mathbb{R}^d} (\widehat{f}_1(\mathbf{x}), \widehat{f}_2(\mathbf{x}), \widehat{f}_4(\mathbf{x}))$, $\min_{\mathbf{x} \in \mathbb{R}^d} (\widehat{f}_1(\mathbf{x}), \widehat{f}_2(\mathbf{x}), \widehat{f}_3(\mathbf{x}), \widehat{f}_4(\mathbf{x}))$ have been obtained by brute-force. The evolutionary multi-objective optimization algorithm NSGA-II (Deb et al., 2002) was run with a large population and for enough generations to return an approximation front $\mathcal{P}_\mathcal{Y}$ to be considered as the "true MetaNACA Pareto front". The knowledge of the latter enables to compute binary metrics (Section 2.3.2) depending on a reference set, as well as the analysis and improvement of the algorithms developed throughout this thesis. Figure 3.9 shows the Pareto front of the $(\widehat{f}_1(\cdot), \widehat{f}_2(\cdot))$ problem for $d = 3, 8, 22$, as well as the Pareto front of $(\widehat{f}_1(\cdot), \widehat{f}_2(\cdot), \widehat{f}_4(\cdot))$ in dimension $d = 8$. 100,000 points randomly drawn in X are evaluated to illustrate the objective space Y of these problems. Remark that it gets more difficult to randomly obtain Pareto optimal designs with the increase in dimension.

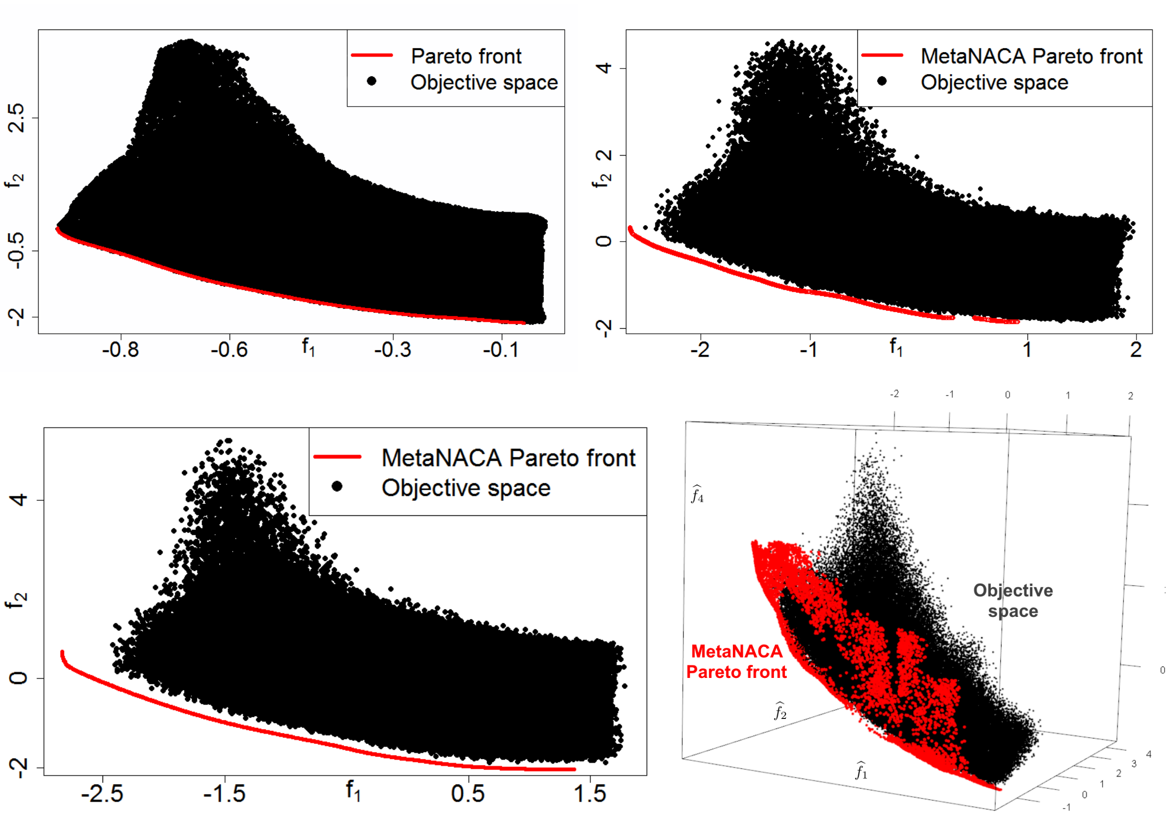


Figure 3.9: Pareto fronts (red curve/dots) for some MetaNACA problems (top left $d = 3$, top right $d = 8$, bottom left $d = 22$, bottom right $d = 8$) with two or three objectives. The black dots correspond to other randomly sampled designs.

An analysis of the Pareto set $\mathcal{P}_{\mathcal{X}}$ highlights the most critical variables, as well as relations between parts of $\mathcal{P}_{\mathcal{Y}}$ and regions of $\mathcal{P}_{\mathcal{X}}$, but is not detailed here for the sake of brevity. As an example, in the (-lift, drag) optimization in dimension 3 (see Figure 3.4), Pareto optimal designs have large P and small T (thin airfoil where the maximum of camber is far away from the leading edge). The variation along the Pareto front, from high lift-high drag airfoils to profiles with low lift and low drag is caused by the decrease of M, the maximum camber. In this case the Pareto front and set are continuous, but this is not necessarily the case in other problems.

3.3 Benchmarking of Bayesian Multi-Objective Optimizers

From now on, the MetaNACA are considered as black-boxes. Given the dimension of the problem, a design $\mathbf{x} \in \mathbb{R}^d$ is associated to its outputs $\widehat{f}_j(\mathbf{x})$ (where the j's are the considered problems).

In applications, multi-objective problems

$$\min_{\mathbf{x}\in X} \; (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \tag{3.3}$$

have to be solved within a prescribed schedule, which determines the allowed computational *budget* in the Bayesian optimization procedure (Figure 2.6). Questions nonetheless remain regarding the optimizer setting. Having a benchmark with tunable dimension and number of objectives enables the analysis of the behavior of Bayesian multi-objective optimizers (Figure 3.10).

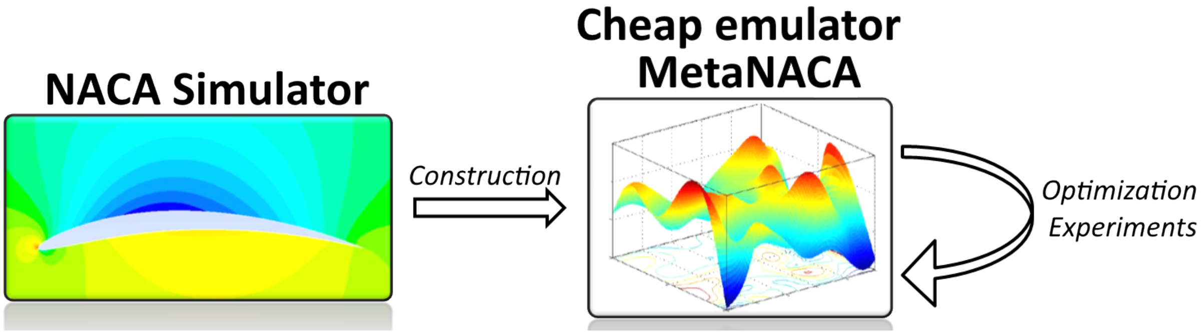


Figure 3.10: MetaNACA: a benchmark for Bayesian multi-objective optimizers, built from real-world data using surrogate modeling techniques.

The GPareto (Binois and Picheny, 2015) package used during this thesis includes four Bayesian multi-objective infill criteria introduced in Section 2.4: EHI (Emmerich et al., 2006), EMI (Svenson and Santner, 2010), SMS (Ponweiser et al., 2008) and SUR (Picheny, 2015). In this section, optimization experiments are carried out on the MetaNACA test suite to answer following questions:

- Given a *budget* $= n + p$, how to choose n and p? Should the emphasis be given to a large DoE, in order to have an accurate initial metamodel? Or should the budget favor the infill criterion, to drive the optimization towards the Pareto front as rapidly as possible, at the risk of being mislead by an insufficiently precise initial metamodel?

- Is there an acquisition function which consistently outperforms or gets outperformed by the others? How do they drive the optimization and how does the empirical front evolve during the successive iterations? Do they lead to similar Pareto front approximations?

- Does the increase in dimension d make the problem harder? How does it impact the optimizers?

- Is the increase in the number of objectives m a supplementary difficulty? May it be worth to consider less objectives?

For the purpose of answering these questions, optimizations were run of the problems defined in Section 3.2.2.2, for all dimensions $d = 3, 8, 22$.

More designs are needed to fit an initial DoE with increasing dimension. It is also expected to wait for more iterations before covering the Pareto front in 22 dimensions than in 3. For these reasons, we took $budget = 60$ for $d = 3$, $budget = 100$ for $d = 8$, and $budget = 200$ for $d = 22$. Even though $n = 10d$ is advised for the initial DoE in many studies (Jones et al., 1998; Loeppky et al., 2009), we tried surrogate models with even less observations. The smallest DoEs contain $n \approx 2d + 4$ designs, similar to the size $n = 3d$ in Feliot (2017). The following $n + p$ combinations are used to investigate the allocation of *budget*.

d	*budget*	Allocation $(n + p)$
3	60	10+50 30+30 50+10
8	100	20+80 40+60 60+40 80+20
22	200	50+150 100+100 150+50

Table 3.2: *budget* distribution in the MetaNACA experiments.

All the experiments were started from 10 different space-filling DoEs of size n to provide statistically significant results. Some results and convergence figures are given in the Appendix A. SUR was only run for $d = 3$ and $m = 2$ due to the integration in the X space. The general conclusions are the following:

- SMS slightly outperforms the other acquisition functions.

- EHI uncovers $\mathcal{P}_\mathcal{Y}$ slightly more at the borders of the front than other criteria. The approximation needs a little more iterations to attain the central part of $\mathcal{P}_\mathcal{Y}$.

- Regarding the *budget* allocation, it was evidenced that it is worth assigning an as large proportion as possible to p. Even though the first surrogate models may lack of precision, the approximation front is enhanced during the first iterations. Additionally, such a procedure improves the GP in the part of the design space related to Pareto-optimality, instead of making it more accurate in the whole, but majorly non-critical X. Recall that the exploitation-exploration mechanism may anyway episodically promote undersampled regions of X. An illustration of two runs with same budget but different allocations, 20+80 and 80+20, $(d = 8)$ is proposed in Figure 3.11. The n DoE points are shown in black, and the p sequential infills are the blue triangles. In terms of hypervolume indicator, the black Pareto front (i.e. the Pareto front considering only the DoE observations) is better when $n = 80$ (right) than when $n = 20$ (left). However, for the same computational budget ($budget = n + p = 100$), the final Pareto front (blue) is enhanced in the 20+80 (left) case, as highlighted by the larger hypervolume (computed up to the red square, cyan area).

- Initial fronts (i.e. approximation fronts after the sole n DoE designs have been evaluated) are only slightly improved with larger n's.

- The progress in the indicators is the largest during the first iterations. Even if the initial metamodel might lack of precision, Pareto front enhancements are quickly obtained. However, the hypervolume indicator does not reach 1 at the exhaust of the budget and its slope (see Appendix A) is not null in the last iterations, i.e. *budget* is not enough to unveil the whole Pareto front accurately. This appears to be even more true when $m = 4$ as the size of the Pareto is larger.

- The quality of Pareto front approximations are quite comparable regarding d (keeping in mind the budgets are augmented with d). However, more clusters of similar $\mathbf{y}$ values have been observed for larger d's. Indeed, due to the L_i's which do not impact $\mathbf{y}$ substantially, some designs appear to be very distant in X although they lead to approximately the same output.

- At an equal budget, the increase in number objectives leads to approximately the same ratio of covered hypervolume. However, if considering the restriction of four-objectives optimizations to two objectives (i.e. $(\widehat{f_1}(\mathbf{x}), \widehat{f_2}(\mathbf{x}), \widehat{f_3}(\mathbf{x}), \widehat{f_4}(\mathbf{x}))$ restricted to $(\widehat{f_1}(\mathbf{x}), \widehat{f_2}(\mathbf{x})))$, the front in the latter two objectives is not as good as the front in the purely bi-objective problem: a worsened marginal optimality is the price to pay when optimizing more objectives. Vice-versa, optimizations focusing on two objectives do not produce an as good four-objectives hypervolume as four-objectives optimizations. Another remark is that the infill criteria become much more expensive to compute and optimize, especially for criteria that consider an expectation, i.e. EHI and EMI, and the number of non-dominated points grows.

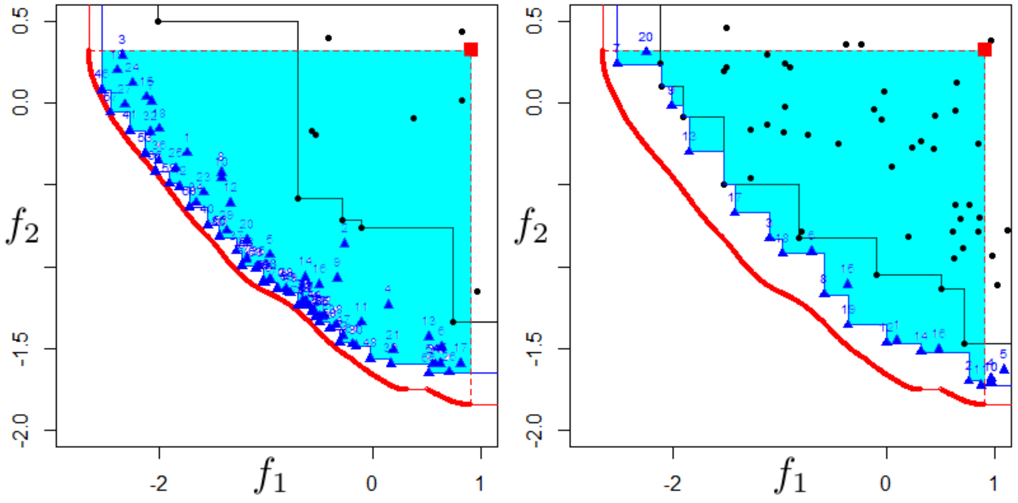


Figure 3.11: Impact of the *budget* $= n + p$ allocation. Left: EHI optimization with $n + p = 20 + 80$. Right: EHI optimization with 80+20 budget allocation.

A supplementary experiment was conducted with non-normalized (in the objective space) MetaNACAs $\widehat{f}_j(\cdot)$. The optimizations were only lightly degraded for all infill criteria, except for EMI, whose convergence to the Pareto front became clearly poorer because this acquisition function relies on a max criterion among the objectives, which is not adapted to objective functions with different magnitudes.

In the following chapters of this thesis, together with other test problems, the MetaNACA test bed is employed for benchmarking various proposed concepts and techniques. It is also used in experiments in Chapters 4 and 5 to compare and analyze the behavior of the developed algorithms with regard to state-of-the-art implementations.

The method developed in Chapter 4 hinges on the EHI criterion. Even though it was slightly outperformed by SMS, it performed well on the MetaNACA instances. Additionally, its reference point, originally thought of as a second-order hyperparameter, lets itself interpret as a focus operator which will be managed to control the optimization.

Chapter 4

Targeting Solutions in Bayesian Multi-Objective Optimization: the C-EHI/R-EHI algorithm

Contents

In multi-objective optimization, when the number of experiments is severely restricted and/or when the number of objectives increases, uncovering the whole set of Pareto optimal solutions is out of reach, even for surrogate-based approaches: the proposed solutions are sub-optimal or do not cover the front well. As all optimal solutions do not have the same worth, in this chapter, we prioritize the search of solutions that reflect the decision maker's preferences, expressed through a target objective. Following, a Bayesian multi-objective optimization method for directing the search towards this preferred part of $\mathcal{P}_\mathcal{Y}$ is proposed by tailoring the well-known EHI (Emmerich et al., 2006) infill criterion. If no preference indication is given, we start by searching solutions which are close to the Pareto front center, as non-compromising solutions have usually little point in applications. We define and characterize this center, which is defined for any type of front. Targeting a subset of the Pareto front allows an improved optimality of the solutions and a better coverage of this zone, which is our main concern. A criterion for detecting local convergence to the Pareto front is described. Once the criterion is triggered, a widened part of the Pareto front, where sufficiently accurate convergence is forecasted within the remaining budget, is targeted. Numerical experiments show how the resulting algorithm, C-EHI or R-EHI (whether the central region or a user-supplied area is desired), better attains the preferred part of the Pareto front when compared to state-of-the-art Bayesian algorithms.

4.1 Introduction

We consider the multi-objective optimization problem

$$\min_{\mathbf{x}\in X}(f_1(\mathbf{x}),\dots,f_m(\mathbf{x})) \tag{4.1}$$

where $X \subset \mathbb{R}^d$ is the parameter space, and $f_j(\cdot)$, $j = 1,\dots,m$ are the m objective functions. The latter are the outputs of a computationally expensive computer code (several hours to days for one evaluation), so that only a small number of experiments can be carried out. Under this restriction, Bayesian optimization methods (see Section 2.2) have proven their effectiveness in single objective problems, and have been extended to the multi-objective setting (Bautista, 2009; Binois, 2015; Emmerich et al., 2006; Keane, 2006; Knowles, 2006; Picheny, 2015; Ponweiser et al., 2008; Svenson and Santner, 2010). In the case of very narrow budgets (about a hundred evaluations), obtaining an accurate approximation of the Pareto front remains out of reach, even for Bayesian approaches. This issue gets worse with increasing number of criteria. The chapter provides illustrations of this phenomenon in Section 4.7. Looking for the entire front can anyway seem useless as the Pareto set will contain many irrelevant solutions from an end-user's point of view.

In this chapter, instead of trying to approximate the entire front, we search for a well-chosen part of it. The practitioner is asked to express its desires via the specification of an aspiration point $\mathbf{R}$ which should be attained and improved if possible. If no specific information about the preferences of the decision maker is given, we assume that solutions which are well-balanced are the most interesting ones. More than returning only relevant designs at the end of the procedure, we argue that convergence at these solutions should be enhanced by the specifically targeting them. Restricting the search to parts of the objective space according to user-supplied information is a common practice in multi-objective optimization, see Section 2.3. More recently, preferences have also been included in Bayesian multi-objective optimization and a more detailed review of related works is given in Section 4.2.2.

The first contribution of this chapter is the definition of a criterion for targeting specific parts of the Pareto front (i.e. the user-provided target or equilibrated solutions). The second contribution is the formal definition of "well-balanced solutions" via the concept of Pareto front center. The latter is automatically determined by processing the GPs and defines an implicitly preferred region when no external information is supplied. Other contributions are the description of a local convergence criterion to the Pareto front, and the management of the preference region according to the remaining computational budget once the criterion is triggered.

An overview of the proposed method, which we name the C-EHI algorithm (for Centered Expected Hypervolume Improvement), is sketched in Figure 4.1. It uses the concept of Pareto front center defined in Section 4.4.2. C-EHI iterations are made of three steps. First, an estimation of the Pareto front center is carried out, as described in Section 4.4.2 and sketched in Figure 4.1a. Second, the estimated center allows to target well-balanced parts of the Pareto front by a modification of the EHI criterion (cf. Section 4.3). Figure 4.1b illustrates the idea. Third, to avoid wasting computations once the center is attained, the part of the Pareto front that is searched for is broadened in accordance with the remaining budget. To this aim, a criterion to test convergence to the center is introduced in Section 4.5. When triggered (see Figure 4.1c), a new type of iteration starts until the budget is exhausted (see Figure 4.1d). Section 4.6 explains how the new goals are determined.

The R-EHI algorithm (for Reference point based Expected Hypervolume Improvement) operates exactly in the same manner, except that the preferred region to be targeted has no longer to be estimated, since it is externally supplied by the decision maker.

The methodology is tested on popular test functions (ZDT1, ZDT3, Zitzler et al., 2000, and P1, Parr, 2013), and on the MetaNACA (Chapter 3). The results are presented in Section 4.7. The default test case that illustrates the algorithm concepts before numerical testing (Figures 4.12 to 4.23) is the MetaNACA with $m = 2$ objectives and $d = 8$ variables.

4.2 Deeper Insights in Bayesian Multi-Objective Optimization

4.2.1 EHI: a multi-objective optimization infill criterion

The EHI (Expected Hypervolume Improvement, Emmerich et al., 2005, 2011, 2006) is one of the most competitive (Emmerich et al., 2020; Shimoyama et al., 2013; Yang et al.,

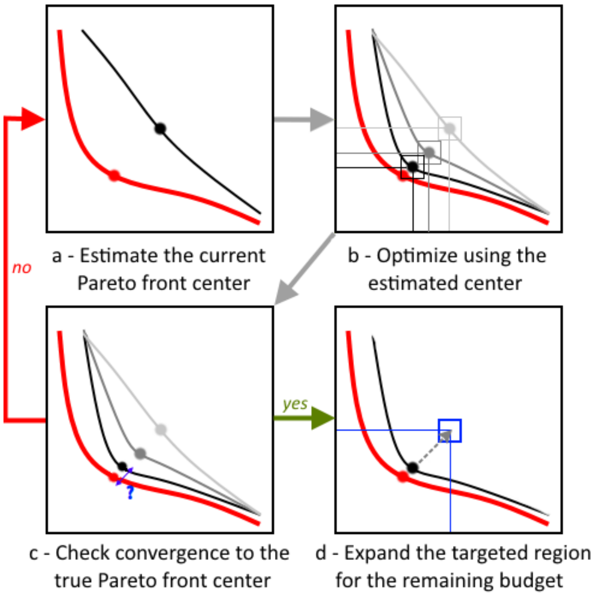

no
a - Estimate the current Pareto front center
b - Optimize using the estimated center
?
yes
c - Check convergence to the true Pareto front center
d - Expand the targeted region for the remaining budget

Figure 4.1: Sketch of the proposed C-EHI algorithm for targeting equilibrated solutions. The Pareto front center properties (a) are discussed in Section 4.4.2; How to guide the optimization (b) is the topic of Section 4.3; Section 4.5 details how convergence to the Pareto front center is tested (c); How to widen the search within the remaining budget (d), is presented in Section 4.6.

2015; Zuhal et al., 2019) multi-objective infill criteria. It rewards the expected growth of the hypervolume indicator (Emmerich et al., 2005; Zitzler, 1999), corresponding to the hypervolume dominated by the approximation front up to a reference point $\mathbf{R}$ (see Section 2.4 and Figure 4.2), when adding a new observation $\mathbf{x}$. The hypervolume indicator of a set $\mathcal{A}$ is

$$I_H(\mathcal{A};\mathbf{R}) = \bigcup_{\mathbf{y}\in\mathcal{A}} \int_{\mathbf{y}\preceq\mathbf{z}\preceq\mathbf{R}} d\mathbf{z} = Vol\left(\bigcup_{\mathbf{y}\in\mathcal{A}} \{\mathbf{z} : \mathbf{y} \preceq \mathbf{z} \preceq \mathbf{R}\}\right)$$

and the hypervolume improvement induced by $\mathbf{y} \in \mathbb{R}^m$ to the set $\mathcal{A}$ is $I(\mathbf{y};\mathbf{R}) = I_H(\mathcal{A} \cup \{\mathbf{y}\};\mathbf{R}) - I_H(\mathcal{A};\mathbf{R})$. In particular, if $\mathcal{A} \preceq \{\mathbf{y}\}$, or if $\mathbf{y} \npreceq \mathbf{R}$, $I(\mathbf{y};\mathbf{R}) = 0$. For a design $\mathbf{x}$, $\text{EHI}(\mathbf{x};\mathbf{R})$ is

$$\text{EHI}(\mathbf{x};\mathbf{R}) := \mathbb{E}[I(\mathbf{Y}(\mathbf{x});\mathbf{R})]. \tag{4.2}$$

The EHI possesses appealing theoretical properties (Emmerich et al., 2011; Shimoyama et al., 2012; Wagner et al., 2010) and is a refinement of the Pareto dominance (see Section 2.3). As the hypervolume improvement induced by a dominated solution equals zero, EHI maximization intrinsically leads to Pareto optimality. It also favors well-spread solutions, as the hypervolume increase is small when adding a new value close to an already observed one in the objective space (Auger et al., 2009c, 2012).

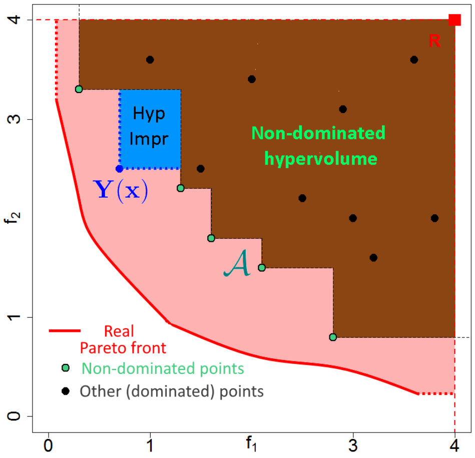


Figure 4.2: The hypervolume indicator of the non-dominated set (green points) corresponds to the area dominated by it, up to $\mathbf{R}$ (in brown). The blue rectangle is the hypervolume improvement brought by $\mathbf{Y}(\mathbf{x})$, $I(\mathbf{Y}(\mathbf{x}); \mathbf{R})$.

Several drawbacks should be mentioned. First, EHI requires the computation of m-dimensional non rectangular hypervolumes. Even though the development of efficient algorithms for computing the hypervolume (Beume et al., 2009; Chan, 2013; Couckuyt et al., 2014; Jaszkiewicz, 2018; Lacour et al., 2017; Russo and Francisco, 2014; While et al., 2012) to temper the computational burden of EHI is an active field of research especially in bi-objective (Emmerich et al., 2011, 2016) and three objectives problems (Yang et al., 2017; Zhao et al., 2018), the complexity grows exponentially with the number of objectives and linearly with the number of non-dominated points. Very recently only, a formula for computing EHI has been found for any number of objectives m (Yang et al., 2019a). This avoids expensive Monte Carlo that were previously required to compute the EHI (Binois, 2015; Binois and Picheny, 2015; Emmerich et al., 2006). The complexity of EHI's calculation is nonetheless growing exponentially in m as the non-dominated hypervolume computation is an NP hard problem (in the number of objectives, Yang et al., 2019a). An analytic expression of its gradient has been discovered recently only, and is limited to the bi-objective case (Yang et al., 2019b). Second, the hypervolume indicator is less relevant for many-objective optimization, as the amount of non-dominated solutions rises with m, and more and more solutions contribute to the growth of the non-dominated hypervolume; in a many-objective setting, this metric is less able to distinguish truly relevant from non-informative solutions. Last, the choice of the reference point $\mathbf{R}$ is unclear and influences the optimization results, as will be highlighted in Example 4.1 and in Section 4.3.

$\mathbf{R}$ was originally seen as an arbitrary second order hyperparameter with default values chosen so that all Pareto optimal points are valued in the EHI (Beume et al., 2007; Emmerich et al., 2005; Knowles and Corne, 2003). Several studies (e.g., Ponweiser et al., 2008) suggest taking $\overline{\mathbf{N}} + \mathbf{1}$, or $\overline{\mathbf{N}} + 0.1(\overline{\mathbf{N}} - \overline{\mathbf{I}})$, which is employed in Binois and Picheny (2015).

Later, the effect of $\mathbf{R}$ has received some attention. Auger et al. (2009c, 2012) have theoretically and experimentally investigated the μ-optimal distribution on the Pareto front induced by the choice of $\mathbf{R}$. Ishibuchi et al. (2010) have noticed a variability in the solutions given by an EMO algorithm when $\mathbf{R}$ changes. Feliot (2017) has also observed that $\mathbf{R}$ impacts the approximation front and recommends $\mathbf{R}$ to be neither too far away nor too close to $\mathcal{P}_{\mathcal{Y}}$. By calculating EHI restricted to areas dominated by "goal points", Parr (2013) implicitly acted on $\mathbf{R}$ and noticed fast convergence when the goal points were taken on $\widehat{\mathcal{P}_{\mathcal{Y}}}$. In Li et al. (2018b), an alternative to the hypervolume improvement is proposed. In essence, it is a sum of EHI's with different non-dominated reference point $\mathbf{R}$'s which eases the computations when compared to EHI in a similar fashion to the criterion proposed in Section 4.3.2.

The choice of $\mathbf{R}$ is further discussed in Section 4.3. The following example highlights its omnipresence inside the EHI infill criterion.

Example 4.1. *Let us consider the m-objective degenerate problem,*

$$\min_{\mathbf{x}\in X\subset\mathbb{R}^d} \underbrace{(f(\mathbf{x}),\dots,f(\mathbf{x}))}_{m\ times}, \tag{4.3}$$

observed at $\mathbf{x}^{(1:n)}\subset X$ *with corresponding values* $\mathbf{y}^{(1:n)}=\{y^{(1)}\mathbf{1}_m,\dots,y^{(n)}\mathbf{1}_m\}\subset Y$*. Let* $f_{\min}=\min(y^{(1)},\dots,y^{(n)})$*. (4.3) has a unique minimum and the empirical Pareto front is* $\widehat{\mathcal{P}_{\mathcal{Y}}}=\{\mathbf{f}_{min}\}$*, with* $\mathbf{f}_{\min}=\mathbf{1}_m f_{\min}$*.*

Consider EHI with an arbitrary reference point $\mathbf{R}$ *for which* $\widehat{\mathcal{P}_{\mathcal{Y}}}\preceq\mathbf{R}$ *(i.e.,* $f_{\min}\le R$*) with* $R_j=R\ \forall j=1,\dots,m$ *(such a dominated* $\mathbf{R}$ *eases the calculations and is adapted to this problem, but the remark applies to any dominated* $\mathbf{R}$*). We consider* $\mathbf{y}$*'s of the form* $\mathbf{1}_m y$ *which stem from (4.3) and by definition, EHI*$(\mathbf{x},\mathbf{R})=\mathbb{E}[I(\mathbf{Y}(\mathbf{x});\mathbf{R})]$ *where* $I(\mathbf{y};\mathbf{R})=I_H(\widehat{\mathcal{P}_{\mathcal{Y}}}\cup\{\mathbf{y}\};\mathbf{R})-I_H(\widehat{\mathcal{P}_{\mathcal{Y}}};\mathbf{R})=I_H(\{\min(\mathbf{f}_{min},\mathbf{y})\};\mathbf{R})-I_H(\widehat{\mathcal{P}_{\mathcal{Y}}};\mathbf{R})$*. Since* $\mathbf{R}=\mathbf{1}_m R$ *and* $\mathbf{y}=\mathbf{1}_m y$*,* $I_H(\{\mathbf{y}\};\mathbf{R})=(R-y)_+^m$*,* $I_H(\widehat{\mathcal{P}_{\mathcal{Y}}};\mathbf{R})=(R-f_{\min})_+^m$*, and finally*

$$I(\mathbf{y};\mathbf{R})=\sum_{k=0}^{m-1}C_m^k(f_{min}-y)_+^{m-k}(R-f_{min})_+^k. \tag{4.4}$$

While $\mathbf{R}$ *should not impact the optimization and be a second-order hyperparameter, it is evidenced in (4.4) that the improvement brought by* $\mathbf{y}$ *depends on* $\mathbf{R}$ *through the* $(R-f_{\min})_+^k$ *term, see Figure 4.3.*

This example highlights that pre-assigned reference points, or $\mathbf{R}$'s chosen as $\overline{\mathbf{N}}+1$ or $\mathbf{R}=\overline{\mathbf{M}}$ are not neutral and influence the optimization, due to the last term in (4.4).

Here, $\mathbf{R}$'s bias can be removed by suppressing the terms depending on R which occur for $k=1,\dots,m-1$ in (4.4), i.e. if $(R-f_{\min})_+=0$, hence $R=f_{\min}$, in which case $I(\mathbf{y};\mathbf{R})=(f_{\min}-y)_+^k=(R-y)_+^k$: it is a product of improvements over $f_{\min}$ (or R), and by independence of the metamodels the EHI boils down to a product of Expected Improvements. As (4.3) is in reality a mono-objective problem, this is the formulation one would wish to retrieve.

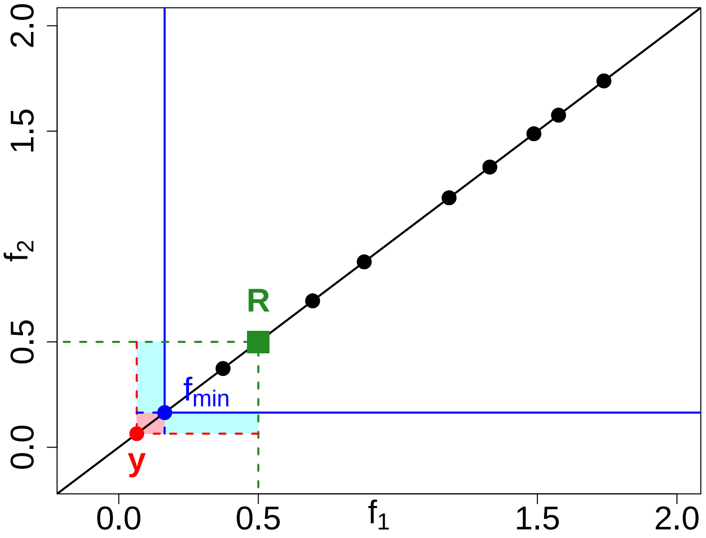


Figure 4.3: m-degenerate problem (4.3). The improvement brought by $\mathbf{y} \in Y$ is measured by the improvement over $\mathbf{f}_{\min}$ (light red area), but also by a term depending on $\mathbf{R}$ (light blue area).

(4.4) holds for $\widehat{\mathcal{P}_{\mathcal{Y}}} \preceq \mathbf{R}$, hence does not apply to $R < f_{\min}$, because $\mathbf{R} \prec \widehat{\mathcal{P}_{\mathcal{Y}}}$ (strictly) in this case. y's such that $R < y < f_{\min}$ would have null improvement in this case, even though they improve over $f_{\min}$. With such an $\mathbf{R}$, $I(\mathbf{y};\mathbf{R}) = (R-y)_+^m$: the improvement exclusively depends on $\mathbf{R}$. Even though it corresponds to $\mathbf{R} \prec \widehat{\mathcal{P}_{\mathcal{Y}}}$ which is clearly not the standard setting (Beume et al., 2007), $R < f_{\min}$ eventually also makes sense as it refers to a different EI threshold (Equation 2.8) investigated in Jones (2001), namely $a = R$. In this case, the improvement is uniquely but explicitly controlled by the choice of R and no longer by $f_{\min}$.

In truly multi-objective problems, $\mathbf{R}$ biases the improvement of extreme points only. The remarks nonetheless highlight the impact of $\mathbf{R}$ and suggest possible directions to articulate it (Auger et al., 2009c) through alternative settings such as $\mathbf{R} \in \widehat{\mathcal{P}_{\mathcal{Y}}}$ or non-dominated $\mathbf{R}$'s, that, to the best of our knowledge, have never been investigated[1]. The proper choice of $\mathbf{R}$ constitutes the foundations for the mEI criterion introduced in Section 4.3 and is further discussed in Section 5.3.1.

Remark 4.1. *In Example 4.1, setting the reference point* $\mathbf{R} = \overline{\mathbf{N}} + r(\overline{\mathbf{N}} - \overline{\mathbf{I}})$, $r \geq 0$, *which is recommended in Ishibuchi et al. (2018) and is the default setting in* `GPareto` *(Binois and Picheny, 2015) with* $r = 0.1$, *removes the* $(R - f_{min})_+^k$ *term in (4.4) since* $\overline{\mathbf{I}} = \overline{\mathbf{N}}$, *and eventually* $\mathbf{R} = \mathbf{f}_{\min}$.

Remark 4.2. *If the EMI infill criterion (Svenson and Santner, 2010, Section 2.4) is*

[1] In vertue of Proposition 4.1, some approaches proposed in Parr (2013); Zhan et al. (2017) are implicitly equivalent to choosing such particular $\mathbf{R}$'s, which however was not the original aim of the authors.

considered for solving (4.3), the improvement $I(\mathbf{y})$ *is measured by* $\max_{j=1,\dots,m}(f_{\min j} - y_j)_+ = (f_{\min} - y)_+$. *The criterion is not biased by an external hyperparameter and the EI formulation is directly retrieved, but it is surprising to find a mono-objective improvement inside a multi-objective infill criterion. As discovered through the experiments of Chapter 3, this acquisition function indeed tends to consider improvements in the* m *functions separately.*

4.2.2 Past work on targeted Bayesian multi-objective optimization

Targeting special parts of the objective space has been largely discussed within the multi-objective optimization literature (see Section 2.3.3). The benefits of targeting a part of the Pareto front instead of trying to unveil it entirely go beyond reflecting the user's preferences: as will be shown by the experiments of Section 4.7, it allows an enhanced distribution of the proposed solutions within this area.

Previous works in Bayesian Multi-Objective Optimization have also targeted particular areas of the objective space thanks to ad-hoc infill criteria. The Weighted Expected Hypervolume Improvement (WEHI, Auger et al., 2009a,b; Brockhoff et al., 2013; Feliot et al., 2018; Zitzler et al., 2007) is a variant of EHI that emphasizes given parts of the objective space through a user-defined weighting function. In Palar et al. (2018); Yang et al. (2016a,b), a Truncated EHI criterion is studied where the Gaussian distribution is restricted to a user-supplied hyperbox in which new solutions are sought.

4.3 An infill criterion to target parts of the Pareto front

4.3.1 Targeting with the reference point

Our approach starts from the observation that any region of the objective space can be targeted with EHI solely by controlling the reference point $\mathbf{R}$. Indeed, as $\mathbf{y} \npreceq \mathbf{R} \Rightarrow I(\mathbf{y};\mathbf{R}) = 0$, the choice of $\mathbf{R}$ is instrumental in deciding the combination of objectives for which improvement occurs, the *improvement region*:

$$\mathcal{I}_{\mathbf{R}} := \{\mathbf{y} \in Y : \mathbf{y} \preceq \mathbf{R}\}.$$

As illustrated in Figure 4.4, the choice of $\mathbf{R}$ defines the region in objective space where $I > 0$ and where the maximum values of EHI are expected to be found. The choice of $\mathbf{R}$ is crucial as it defines the region in objective space that is highlighted. To our knowledge, $\mathbf{R}$ has always been chosen to be dominated by the whole approximation front (that is, $\mathbf{R}$ is at least the empirical Nadir point, which corresponds to the case of **R1** in Figure 4.4). The targeting ability of $\mathbf{R}$ can and should however be taken into account: for example,

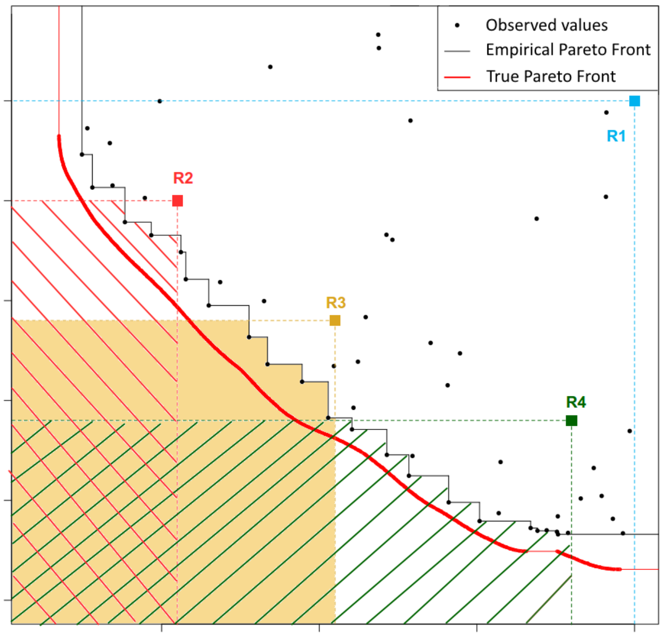


Figure 4.4: Different reference points and the areas $\mathcal{I}_{\mathbf{R}}$ that are targeted.

solutions belonging to the left part of the Pareto front in Figure 4.4 can be aimed at using EHI$(\cdot;\mathbf{R2})$ instead of the more general EHI$(\cdot;\mathbf{R1})$.

Because of the extremely limited number of possible calls to the objective functions, we would like to prioritize the search by first looking for a well-chosen part of $\mathcal{P}_{\mathcal{Y}}$: we implicitly prefer this area over other solutions. This is implemented simply by setting the reference point $\mathbf{R}$ adequately, in contrast to other works that set the reference point at levels dominated by all Pareto optimal points, and by maximizing EHI$(\mathbf{x};\mathbf{R})$.

4.3.2 mEI, a computationally efficient proxy to EHI

We define the mEI criterion for multiplicative Expected Improvement

Definition 4.1. *(mEI criterion) The multiplicative Expected Improvement is the product of Expected Improvements in each objective defined in Equation (2.8),*

$$mEI(\cdot;\mathbf{R}) := \prod_{j=1}^{m} EI_j(\cdot;R_j), \tag{4.5}$$

where EI$_j$ is the EI operating on the j-th metamodel, $Y_j(\cdot)$. mEI is a natural extension of the mono-objective Expected Improvement, as the utility function $(f_{min}-y)_+$ is replaced by $\prod_j (R_j - y_j)_+$. A large part of the motivation for using mEI is that it is naturally designed for promoting $\mathcal{I}_{\mathbf{R}}$. Under some hypothesis, it is equivalent to EHI and therefore shares the appealing properties of the latter (Emmerich et al., 2011; Knowles and Corne, 2002; Wagner et al., 2010), while being less computationally demanding (the complexity grows linearly in m) and easier to be maximized.

First, it is able to target a part of the objective space via $\mathbf{R}$ as the improvement function it is built over differs from zero only in $\mathcal{I}_\mathbf{R}$ and therefore favors designs which dominate $\mathbf{R}$. Conversely, as it does not take the shape of the current approximation front into account, mEI cannot help in finding well-spread Pareto optimal solutions.

Second, when $\widehat{\mathcal{P}_\mathcal{Y}} \nprec \mathbf{R}$, mEI is equivalent to EHI but it is much easier to compute. Contrarily to EHI, mEI does not imply the computation of a costly m-dimensional hypervolume (cf. Section 4.2.1). Its formula is analytical (substitute Equation 2.8 into Equation 4.5) and can easily be parallelized on different processors.

Proposition 4.1. *(EHI-mEI equivalence). Let $Y_1(\cdot), \dots, Y_m(\cdot)$ be independent GPs fitted to the observations $\{\mathbf{x}^{(1:t)}, \mathbf{y}^{(1:t)}\}$, with empirical Pareto front $\widehat{\mathcal{P}_\mathcal{Y}}$. If $\widehat{\mathcal{P}_\mathcal{Y}} \nprec \mathbf{R}$, EHI$(\cdot;\mathbf{R}) =$ mEI$(\cdot;\mathbf{R})$.*

Proof. Let $\widehat{\mathcal{P}_\mathcal{Y}} \nprec \mathbf{R}$. For such a reference point, the hypervolume improvement is

$$I(\mathbf{y};\mathbf{R}) = I_H(\widehat{\mathcal{P}_\mathcal{Y}} \cup \{\mathbf{y}\};\mathbf{R}) - I_H(\widehat{\mathcal{P}_\mathcal{Y}};\mathbf{R}) = I_H(\{\mathbf{y}\};\mathbf{R}) = \begin{cases} \prod_{j=1}^m (R_j - y_j) & \text{if } \mathbf{y} \preceq \mathbf{R} \\ 0 & \text{else} \end{cases}$$

With the $(.)_+$ notation, $I(\mathbf{y};\mathbf{R}) = \prod_{j=1}^m (R_j - y_j)_+$ and EHI$(\mathbf{x};\mathbf{R})$ reduces to $\mathbb{E}[\prod_{j=1}^m (R_j - Y_j(\mathbf{x}))_+] = \prod_{j=1}^m \mathbb{E}[(R_j - Y_j(\mathbf{x}))_+]$ as the $Y_j(\cdot)$ are independent. This is the product of m Expected Improvements with thresholds R_j. □

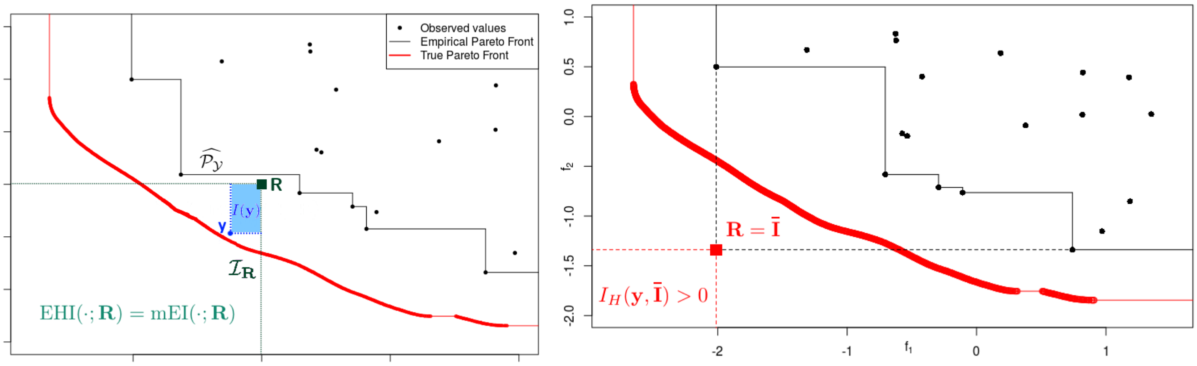


Figure 4.5: Left: when using a non-dominated reference point $\widehat{\mathcal{P}_\mathcal{Y}} \nprec \mathbf{R}$, EHI and mEI are equivalent. The area in blue corresponds to a sample of both the product of improvements w.r.t. R_j and the hypervolume improvement. Right: a product of Expected Improvements with respect to each $f_j(\cdot)$'s best observed value is equivalent to searching an hypervolume increase in $\mathcal{I}_{\bar{\mathbf{I}}}$ (lower left corner), hence a much too optimistic setting.

Thus, choosing a (weakly) non-dominated point as reference point allows to define a criterion that can replace EHI for targeted optimization at a much lower computational cost. The complexity of accounting for the empirical Pareto front is carried over from mEI's calculation to the location of the reference point.

Third, being a product of Expected Improvements, $\nabla \mathrm{mEI}(\mathbf{x};\mathbf{R})$ is computable as

$$\nabla \mathrm{mEI}(\mathbf{x};\mathbf{R}) = \sum_{i=1}^{m} \left[\nabla \mathrm{EI}_i(\mathbf{x};R_i) \prod_{\substack{j=1 \\ j\neq i}}^{m} \mathrm{EI}_j(\mathbf{x};R_j) \right] \tag{4.6}$$

where $\nabla \mathrm{EI}_i(\mathbf{x};\mathbf{R})$ has closed form, see Roustant et al. (2012) for instance. This offers the additional possibility of combining global optimization with gradient based methods when maximizing $\mathrm{mEI}(\cdot;\mathbf{R})$. In comparison, EHI's gradient has been discovered recently only (Yang et al., 2019b), and is limited to $m = 2$ objectives.

As we shall soon observe with the numerical experiments in Section 4.7, mEI is an efficient infill criterion for attaining the Pareto front provided that $\mathbf{R}$ is taken in the non-dominated neighborhood of the Pareto front. It is important that $\mathbf{R}$ is (weakly) not dominated, not only for the equivalence with EHI to hold. Indeed, mEI with a dominated $\mathbf{R}$ may lead to clustering: let $\mathbf{y}^{i_0} = \mathbf{f}(\mathbf{x}^{i_0}) \in \widehat{\mathcal{P}_\mathcal{Y}}$ such that $\mathbf{y}^{i_0} \prec \mathbf{R}$. Then, because improvement over $\mathbf{R}$ is certain at $\mathbf{x}^{i_0}$, $\mathrm{mEI}(\cdot;\mathbf{R})$ will be large and often maximal in the vicinity of $\mathbf{x}^{i_0}$. Clustering in both the objective and the design space will be a consequence, leading to ill-conditioned covariance matrices. Taking a weakly non-dominated reference point instead will diminish this risk as $\prod_{j=1}^{m}(R_j - y_j)_+ = 0 \ \forall \mathbf{y} \in \widehat{\mathcal{P}_\mathcal{Y}}$, and no already observed solution will attract the search. If the reference point is too optimistic, the mEI criterion makes the search exploratory as the only points $\mathbf{x}$ where progress is achieved during GP sampling are those with a large associated uncertainty $\mathbf{s}^2(\mathbf{x})$. A clear example of a too optimistic reference point comes from the straightforward generalization of the default single objective $\mathrm{EI}(\cdot;f_{\min})$ to multiple objectives: it is the criterion $\prod_{j=1}^{m} \mathrm{EI}_j(\cdot;f_{\min_j}) = \mathrm{mEI}(\cdot;\bar{\mathbf{I}})$, that is, the mEI criterion with the empirical Ideal as a reference (right part of Figure 4.5). In non-degenerated problems where the Ideal is unattainable, sequentially maximizing $\mathrm{mEI}(\cdot;\bar{\mathbf{I}})$ will be close to sequentially maximizing $\mathbf{s}^2(\mathbf{x})$.

4.4 Targeting preferred regions

In our method, while mEI is a simple criterion, the emphasis is put on the management of the reference point, described for two situations. Section 4.4.1 assumes the reference point $\mathbf{R}$ expresses the initial goal of the search and an updated target $\widehat{\mathbf{R}}$, which adapts to the current Pareto front approximation $\widehat{\mathcal{P}_\mathcal{Y}}$, controls the next iterates through $\mathbf{x}^{(t+1)} = \arg\max_{\mathbf{x}\in X} \mathrm{mEI}(\mathbf{x};\widehat{\mathbf{R}})$. In the absence of explicitly provided user preferences, the central part of the Pareto front is targeted. The center of the Pareto front, $\mathbf{C}$, is defined in Section 4.4.2 and is as a default preference since it balances the objectives. The estimated center $\widehat{\mathbf{C}}$ corresponds to the center of the current approximation front, and at each iteration of the algorithm, improvements over it are sought by evaluating $\mathbf{x}^{(t+1)} = \arg\max_{\mathbf{x}\in X} \mathrm{mEI}(\mathbf{x};\widehat{\mathbf{C}})$.

4.4.1 User-provided aspiration point

In this section, a user-supplied target $\mathbf{R}$ is provided. Two situations occur and are shown in Figure 4.6. Either this goal can be reached, i.e. there are points of the true Pareto front that also belong to the dominance cone $\mathcal{I}_{\mathbf{R}}$ (left plot), in which case we want to find any of these performance points as fast as possible. Since it is possible to find solutions better than $\mathbf{R}$, a more ambitious goal is defined: $\widetilde{\mathbf{R}}$ is the point belonging to the Ideal-$\mathbf{R}$ line that is the closest to the true Pareto front; this goal is the intersection of $\mathbf{IR}$ with $\mathcal{P}_{\mathcal{Y}}$ if it exists. Or the initial aspiration point is too ambitious, no point of the Pareto front dominates $\mathbf{R}$ (right plot), in which case the new achievable goal $\widetilde{\mathbf{R}}$ to reach is taken as the point belonging to the $\mathbf{R}$-Nadir line that is the closest to the true Pareto front.

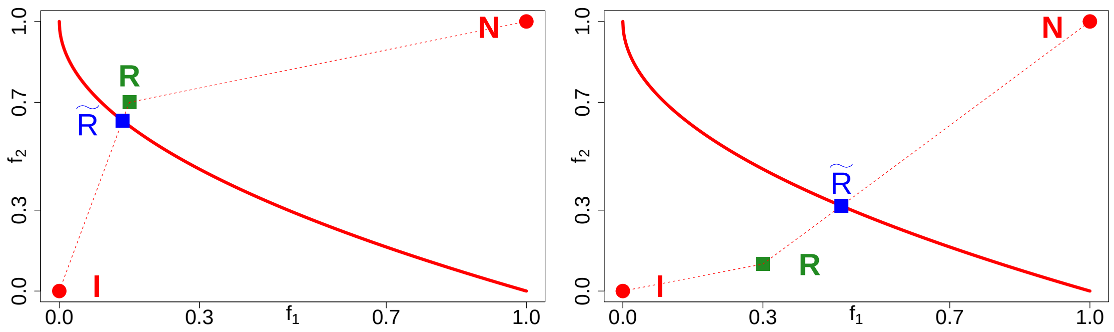


Figure 4.6: Achievable point $\widetilde{\mathbf{R}} \in \mathcal{P}_{\mathcal{Y}}$ arising from the target $\mathbf{R}$. In case of non-continuous fronts, $\widetilde{\mathbf{R}}$ is taken as the closest point to $\mathcal{P}_{\mathcal{Y}}$, see Figure 4.7.

The updated target $\widehat{\mathbf{R}}$ used inside mEI is controlled to reach the attainable target $\widetilde{\mathbf{R}}$ while avoiding the two pitfalls of global optimization: too much search intensification in already sampled regions, and too much exploration of low potential regions. Excessive intensification is associated with $\widehat{\mathbf{R}}$ dominated by already sampled points while superfluous exploration comes from a too ambitious $\widehat{\mathbf{R}}$. A compromise is to adapt $\widehat{\mathbf{R}}$ to the current approximation front $\widehat{\mathcal{P}_{\mathcal{Y}}}$ and to determine it as illustrated in Figure 4.7: if $\mathbf{R}$ dominates at least a point of the empirical Pareto front, $\widehat{\mathbf{R}}$ is the point of the $\mathbf{R}$-estimated Nadir ($\widehat{\mathbf{N}}$) line that is the closest in Euclidean distance to a point of $\widehat{\mathcal{P}_{\mathcal{Y}}}$; vice versa, if $\mathbf{R}$ is dominated by at least one calculated point, $\widehat{\mathbf{R}}$ is the point of the estimated Ideal ($\widehat{\mathbf{I}}$)-$\mathbf{R}$ line that is the closest to $\widehat{\mathcal{P}_{\mathcal{Y}}}$; finally, in more general cases where $\mathbf{R}$ is non-dominated, $\widehat{\mathbf{R}}$ is set at the point of the broken line $\widehat{\mathcal{L}}'$ joining $\widehat{\mathbf{I}}$, $\mathbf{R}$ and $\widehat{\mathbf{N}}$ that is the closest to $\widehat{\mathcal{P}_{\mathcal{Y}}}$. $\widehat{\mathbf{R}}$ progresses along $\widehat{\mathcal{L}}'$ during the optimization and smoothly drives the search towards $\mathbf{R}$ (more precisely towards the attainable target $\widetilde{\mathbf{R}}$). Being critical in the definition of the center of the Pareto front, the estimation of the Ideal and Nadir point is detailed in Section 4.4.2.3. In the rare cases where $\widehat{\mathbf{R}}$ is dominated after the projection, it is moved on the $\widehat{\mathbf{I}}\mathbf{R}\widehat{\mathbf{N}}$ segments towards $\widehat{\mathbf{I}}$ until it becomes non dominated. Thus $\widehat{\mathbf{R}}$ is non-dominated which has the theoretical advantage that mEI$(\mathbf{x};\widehat{\mathbf{R}})$ is equivalent to EHI$(\mathbf{x};\widehat{\mathbf{R}})$.

During the optimization, as $\widehat{\mathcal{P}_{\mathcal{Y}}}$ progresses towards $\mathcal{P}_{\mathcal{Y}}$, $\widehat{\mathbf{R}}$ moves along the $\widehat{\mathbf{I}}\mathbf{R}\widehat{\mathbf{N}}$ line. In the limit of a converged $\widehat{\mathcal{P}_{\mathcal{Y}}}$, the updated reference point will correspond to the $\widetilde{\mathbf{R}}$ of

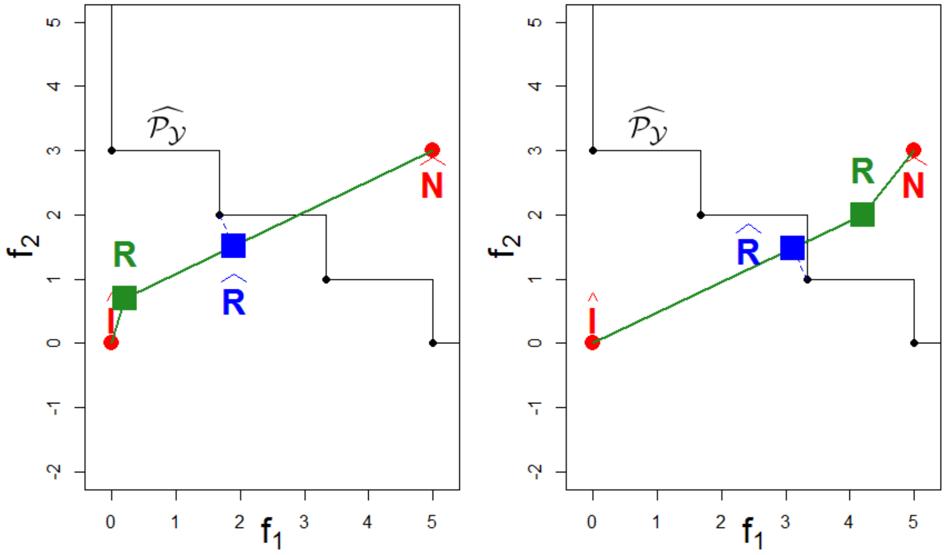


Figure 4.7: To adapt to $\widehat{\mathcal{P}_\mathcal{Y}}$, the user-supplied $\mathbf{R}$ is updated to $\widehat{\mathbf{R}}$. Left: $\mathbf{R}$ is too optimistic, it dominates some of the points of $\widehat{\mathcal{P}_\mathcal{Y}}$, and $\widehat{\mathbf{R}}$ is the closest orthogonal projection of a non-dominated point onto $\mathbf{R}\widehat{\mathbf{N}}$. Right: the user-provided target has been attained and a more ambitious $\widehat{\mathbf{R}}$ is used instead, the closest orthogonal projection of a point of $\widehat{\mathcal{P}_\mathcal{Y}}$ onto $\widehat{\mathbf{I}}\mathbf{R}$.

Figure 4.6. mEI is therefore a local criterion in the Y space, since it eventually targets $\widetilde{\mathbf{R}}$. It is nonetheless worth mentioning that mEI is only local in the Y space. Similarly to EI or to EHI, it is naturally equipped with the exploitation/exploration mechanism and searches for promising[2] new designs in the entire X space.

A flow chart of this Bayesian targeting search is given in Algorithm 1. In the absence of preferences expressed through $\mathbf{R}$, the default implementation uses the center of the front as target, $\mathbf{R} = \mathbf{C}$, which is the subject of the next section.

4.4.2 Center of the Pareto front: definition, properties and estimation

In this section, no preferences are expressed by the decision maker, and as a default setting, the unveiling of well-balanced solutions of $\mathcal{P}_\mathcal{Y}$ is prioritized.

There has been attempts to characterize parts of the Pareto front where objectives are "visually" equilibrated. In Wierzbicki (1999), the neutral solution is defined as the closest point in the objective space to the Ideal point in a (possibly weighted) L^p norm and is located "somewhere in the middle" of the Pareto front. The point of the Pareto front which minimizes the distance to the Ideal point is indeed a commonly preferred

[2] In the sense that they improve over $\widehat{\mathbf{R}}$.

```
Data: Create and evaluate an initial DoE of $n$ designs;
Initialize $m$ GPs $Y_j(\cdot)$ for each objective $f_j(\cdot), j = 1, \dots, m$;
$t = n$; budget;
while $t < budget$ do
    Estimate the Ideal and Nadir point, $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}}$;
    if $\mathbf{R}$ given ;                /* adapt $\widehat{\mathbf{R}}$ to the current Pareto front */
     then
        Compute $\widehat{\mathbf{R}}$ as the closest point from the broken line joining $\widehat{\mathbf{I}}$, $\mathbf{R}$ and $\widehat{\mathbf{N}}$
         to $\widehat{\mathcal{P}_\mathcal{Y}}$;
    else
        /* no $\mathbf{R}$ given, default to center                              */
        Estimate the center of the Pareto front $\widehat{\mathbf{C}}$, and set $\widehat{\mathbf{R}} = \widehat{\mathbf{C}}$;
    end
    $\mathbf{x}^{(t+1)} = \arg\max_{\mathbf{x} \in X} \text{mEI}(\mathbf{x}; \widehat{\mathbf{R}})$;
    Evaluate $f_j(\mathbf{x}^{(t+1)})$, $j = 1, \dots, m$, update the GPs and $\widehat{\mathcal{P}_\mathcal{Y}}$;
    $t = t + 1$;
end
```

Algorithm 1: The **R**/**C**-mEI Bayesian targeting Algorithm.

solution (Zeleny, 1976). In Buchanan and Gardiner (2003), not only the closest to the Ideal point, but also the farthest solution to the Nadir point (see definitions in Section 2.3.1) are brought out, in terms of a weighted Tchebycheff norm. Note that the weights depend on user-supplied aspiration points. Other appealing points of the Pareto front are knee points as defined in Branke et al. (2004a). They correspond to parts of the Pareto front where a small improvement in one objective goes with a large deterioration in at least one other objective, which makes such points stand out as kinks in the Pareto front. When the user's preferences are not known, the authors claim that knee points should be emphasized and propose methods for guiding the search towards them.

Continuing the same effort, we propose a definition of the Pareto front center that depends only on the geometry of the Pareto front.

4.4.2.1 Definition

Definition 4.2. *(Pareto front center) The center of a Pareto front* $\mathbf{C}$ *is the closest point in Euclidean distance to* $\mathcal{P}_\mathcal{Y}$ *on the Ideal-Nadir line* $\mathcal{L}$*.*

In the field of Game Theory, our definition of the center of a Pareto front corresponds to a particular case of the Kalaï-Smorodinsky equilibrium[3] (Binois et al., 2019; Kalai and Smorodinsky, 1975), taking the Nadir as disagreement point $\mathbf{d} \equiv \mathbf{N}$. This equilibrium aims at equalizing the ratios of maximal gains of the players, which is the appealing

[3] convexity of the objective space is also assumed for the KS solution.

property for the center of a Pareto front as an implicitly preferred point. Recently, it has been used for solving many-objective problems in a Bayesian setting (Binois et al., 2019). In general, **C** is different from the neutral solution (Wierzbicki, 1999) and from knee points (Branke et al., 2004a). They coincide in particular cases, e.g. a symmetric and convex front with scaled objectives and a non-weighted norm.

In the case where the Pareto front is an $m-1$-dimensional continuous hypersurface, **C** corresponds to the intersection between $\mathcal{P}_{\mathcal{Y}}$ and $\mathcal{L}$. In a more general case the Pareto front may not be continuous, or may contain some lower dimensional hypersurfaces. This is in particular the case for the empirical front $\widehat{\mathcal{P}_{\mathcal{Y}}}$, a set of non-dominated points. **C** is then the projection of the closest point belonging to $\mathcal{P}_{\mathcal{Y}}$ on $\mathcal{L}$.

The computation of this point remains cheap even for a large m in comparison with alternative definitions involving e.g. the computation of a barycenter in high-dimensional spaces. Some examples for two-dimensional fronts are shown in Figure 4.8. The center of the Pareto front has also some nice properties that are detailed in following section. It exists even if $\mathcal{P}_{\mathcal{Y}}$ is discontinuous (top right front) or convoluted.

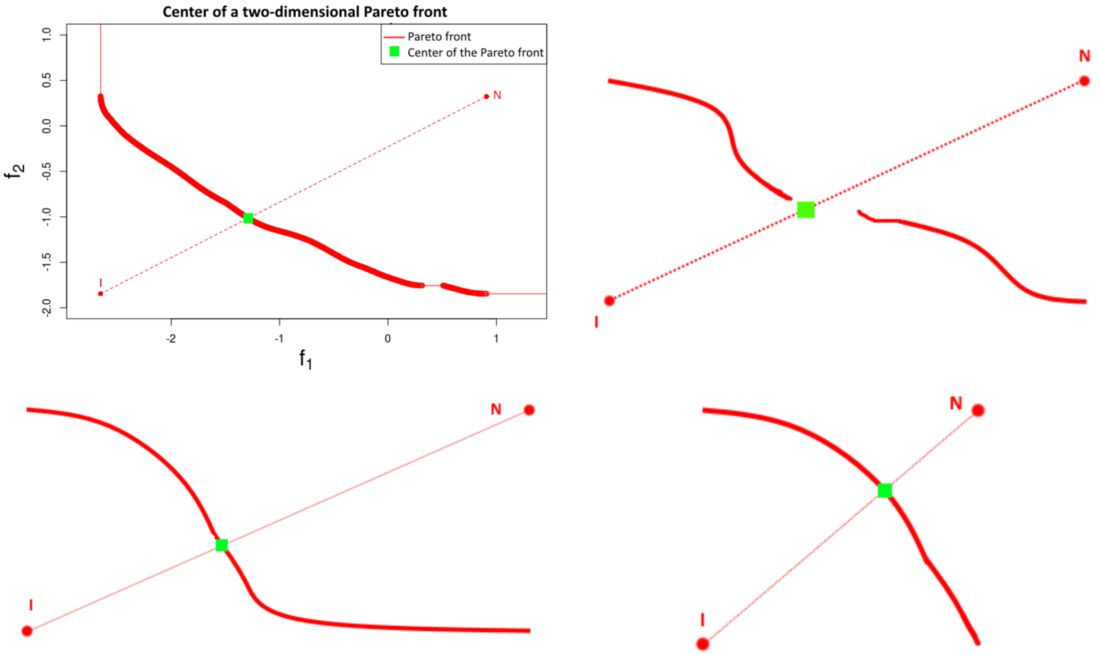


Figure 4.8: Examples of two-dimensional Pareto fronts and their center. Notice that on the bottom, the left and the right fronts are the same, except that the left is substantially extended in the x direction. However, the center has been only slightly modified.

4.4.2.2 Properties

Invariance to a linear scaling of the objectives

The Kalai-Smordinsky solution has been proved to verify a couple of properties, such as invariance to linear scaling[4] (Kalai and Smorodinsky, 1975), which hold in our case. We extend here the linear invariance to the case where there is no intersection between $\mathcal{P}_{\mathcal{Y}}$ and $\mathcal{L}$.

Proposition 4.2. *(Center invariance to linear scaling, intersection case) When $\mathcal{P}_{\mathcal{Y}}$ intersects $\mathcal{L}$, the intersection is unique and is the center of the Pareto front. Furthermore, in that case, the center is invariant after a linear scaling $S : \mathbb{R}^m \to \mathbb{R}^m$ of the objectives: $S(\mathbf{C}(\mathcal{P}_{\mathcal{Y}})) = \mathbf{C}(S(\mathcal{P}_{\mathcal{Y}}))$.*

The proof is given in Appendix B.1. In the bi-objective case ($m = 2$), we also show that a linear scaling applied to the objective space does not change the order of Euclidean distances to $\mathcal{L}$. When $\mathcal{P}_{\mathcal{Y}} \cap \mathcal{L} = \emptyset$, the closest $\mathbf{y} \in \mathcal{P}_{\mathcal{Y}}$ to $\mathcal{L}$, whose projection on $\mathcal{L}$ produces $\mathbf{C}$, remains the closest after any linear scaling of the objective space.

Proposition 4.3. *(Center invariance to linear scaling, 2D case) Let $\mathbf{y}, \mathbf{y}' \in Y \subset \mathbb{R}^2$, and $\mathcal{L}$ be a line in $\mathbb{R}^2$ passing through the two points $\mathbf{I}$ and $\mathbf{N}$. Let $\Pi_{\mathcal{L}}$ be the projection on $\mathcal{L}$. If $\|\mathbf{y} - \Pi_{\mathcal{L}}(\mathbf{y})\| \leq \|\mathbf{y}' - \Pi_{\mathcal{L}}(\mathbf{y}')\|$, then $\mathbf{y}$ remains closer to $\mathcal{L}$ than $\mathbf{y}'$ after having applied a linear scaling $S : \mathbb{R}^2 \to \mathbb{R}^2$ to Y.*

The proof is given in Appendix B.2. This property is of interest as the solutions in the approximation front $\widehat{\mathcal{P}_{\mathcal{Y}}}$ will generally not belong to $\mathcal{L}$. Applying a linear scaling to Y in a bi-objective case does not change the solution in $\widehat{\mathcal{P}_{\mathcal{Y}}}$ that generates $\widehat{\mathbf{C}}$. However, exceptions may occur for $m \geq 3$ as the closest $\mathbf{y} \in \mathcal{P}_{\mathcal{Y}}$ to $\mathcal{L}$ may not remain the same after a particular affine transformation of the objectives, as shown in Appendix B.3.

Low sensitivity to Ideal and Nadir variations

Another positive property is the low sensitivity of $\mathbf{C}$ with regard to extreme points (see Section 2.3.1 for definitions). This property is appealing because the Ideal and the Nadir will be estimated with errors at the beginning of the search (cf. Section 4.4.2.3) and having a stable target $\mathbf{C}$ prevents dispersing search efforts.

Under mild assumptions, the following proposition expresses the low sensitivity in terms of the norm of the gradient of $\mathbf{C}$ with respect to $\mathbf{N}$.

Proposition 4.4. *(Stability of the center to perturbations in Ideal and Nadir) Let $\mathcal{P}_{\mathcal{Y}}$ be locally continuous and $m-1$ dimensional around its center $\mathbf{C}$. Then, $|\frac{\partial C_i}{\partial N_j}| < 1,\ i, j =$*

[4] in Game Theory, given a feasible agreement set $F \subset \mathbb{R}^m$ (Y in our context) and a disagreement point $\mathbf{d} \in \mathbb{R}^m$ ($\mathbf{N}$ here), a KS solution $f \in F$ (the center $\mathbf{C}$) satisfies the four following requirements: Pareto optimality, symmetry with respect to the objectives, invariance to affine transformations (proven in Proposition 4.2) and, contrarily to a Nash solution, monotonicity with respect to the number of possible agreements in F.

$1, \dots, m$ where $\mathbf{N}$ *is the Nadir point, and the variation* $\Delta\mathbf{C}$ *of* $\mathbf{C}$ *induced by a small variation* $\Delta\mathbf{N}$ *in* $\mathbf{N}$ *verifies* $\|\Delta\mathbf{C}\|_2 < \|\Delta\mathbf{N}\|_2$*. A similar relation stands for small Ideal point variations,* $\|\Delta\mathbf{C}\|_2 < \|\Delta\mathbf{I}\|_2$*.*

The proof is given in Appendix B.4. Proposition 4.4 is a local stability result. Without formal proof, it is observed that the center will be little affected by larger errors in Ideal and Nadir positions when compared to alternative definitions of the center. A typical illustration is as follows: the Nadir point is moved by a large amount in one objective (see Figure 4.9). The center is shifted by a relatively small amount and will continue to correspond to an area of equilibrium between all objectives. Other definitions of the center, typically those based on the barycenter of $\mathcal{P_Y}$ would lead to a major displacement of $\mathbf{C}$. In Figure 4.9, the barycenter on $\mathcal{P_Y}$ signaled by $\mathbf{B}$ and $\mathbf{B}'$ has $B'_2 \approx I_2$, which does not correspond to an equilibrated solution as the second objective would almost be at its minimum.

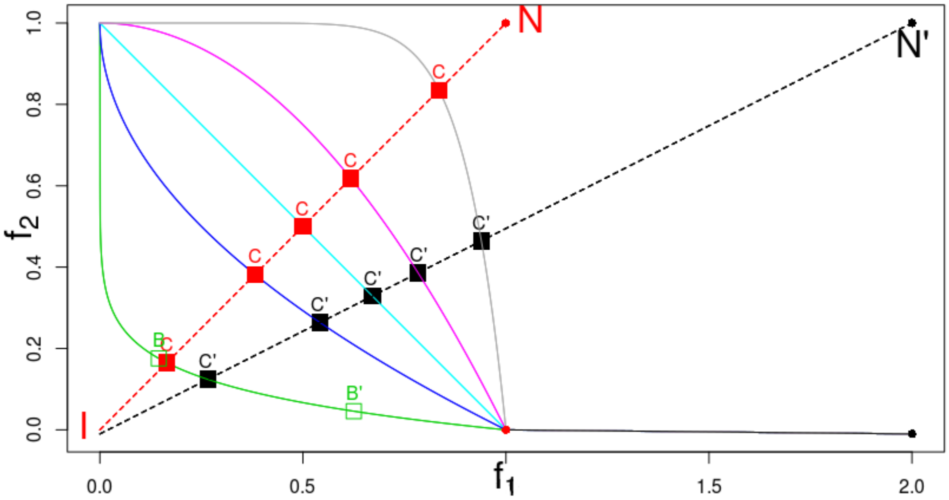


Figure 4.9: Illustration of the global stability of the center in 2D: adding the black part to the colored Pareto fronts will highly modify them and $\mathbf{N}'$ becomes the new Nadir point. The new center $\mathbf{C}'$ is relatively close to $\mathbf{C}$ despite this major $\mathbf{N}$ modification. $\mathbf{B}$, a barycenter-based center would be much more affected, and would no longer correspond to an equilibrium.

4.4.2.3 Estimation of the Pareto front center using Gaussian Processes

Now that we have given a definition of $\mathbf{C}$ relying on $\mathcal{P_Y}$ through $\mathbf{I}$ and $\mathbf{N}$, let us discuss the estimation of $\mathbf{C}$. The *real* front $\mathcal{P_Y}$ is obviously unknown and at any stage of the algorithm, we solely have access to an approximation front $\widehat{\mathcal{P_Y}}$. The empirical Ideal and Nadir points (computed using $\widehat{\mathcal{P_Y}}$) could be weak estimates in the case of a biased approximation front. Thus, we propose an approach using the GPs $Y_j(\cdot)$ to better estimate $\mathbf{I}$ and $\mathbf{N}$ through conditional simulations.

Estimating I and N with GP simulations

Estimating the Ideal and the Nadir point accurately is a difficult task. Indeed, obtaining $\mathbf{I}$ is equivalent to finding the minimum of each $f_j(\cdot), j = 1, \ldots, m$, which corresponds to m classical mono-objective optimization problems. Prior to computing $\mathbf{N}$, the whole Pareto front has to be unveiled but this is precisely our primary concern. Estimating $\mathbf{N}$ before running the multi-objective optimization has been proposed in Bechikh et al. (2010); Deb et al. (2010) using modified EMOAs to emphasize extreme points. We aim at obtaining sufficiently accurate estimators $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}}$ of $\mathbf{I}$ and $\mathbf{N}$ rather than solving these problems exactly. The low sensitivity of $\mathbf{C}$ with regard to $\mathbf{I}$ and $\mathbf{N}$ discussed previously suggests that the estimation error should not be a too serious issue for estimating $\mathbf{C}$. As shown in Section 2.1.2, given s simulation points $\mathbf{x}^{(t+1)}, \ldots, \mathbf{x}^{(t+s)}$, possible responses at those locations can be obtained through the conditional GPs $Y_j(\cdot), j = 1, \ldots, m$. The simulated responses are filtered by Pareto dominance to get n_{sim} simulated fronts $\widetilde{\mathcal{P}_{\mathcal{Y}}}^{(k)}$. The Ideal and Nadir points are then estimated by $\widehat{I_j} = \underset{k=1,\ldots,n_{sim}}{\text{median}} \left(\min_{\mathbf{y} \in \widetilde{\mathcal{P}_{\mathcal{Y}}}^{(k)}} y_j \right)$; $\widehat{N_j} = \underset{k=1,\ldots,n_{sim}}{\text{median}} \left(\max_{\mathbf{y} \in \widetilde{\mathcal{P}_{\mathcal{Y}}}^{(k)}} y_j \right)$, $j = 1, \ldots, m$.

Notice that the definition of $\mathbf{I}$ is not based on the Pareto front. Hence the estimation of I_j does not require m-dimensional simulated fronts, but only single independently simulated responses $\widetilde{Y_j}^{(k)}$. By contrast, as the Nadir hinges on a front, simulated fronts $\widetilde{\mathcal{P}_{\mathcal{Y}}}^{(k)}$ are mandatory for estimating $\mathbf{N}$.

GP simulations are attractive for estimating extrema because they not only provide possible responses of the objective functions but also take into account the surrogate's uncertainty. It would not be the case by applying a (multi-objective) optimizer to a deterministic surrogate such as the conditional mean functions. Even so, they rely on the choice of simulation points $\mathbf{x}^{(t+i)}, i = 1, \ldots, s$ (in a d-dimensional space). For technical reasons (Cholesky or spectral decomposition of $\mathbf{\Gamma}_j$ required for sampling from the posterior), the number of points is restricted to $s \lessapprox 5000$. $\mathbf{x}^{(t+i)}$ have thus to be chosen in a smart way to make the estimation as accurate as possible. In order to estimate $\mathbf{I}$ or $\mathbf{N}$, GP simulations are performed at $\mathbf{x}$'s that have a large probability of contributing to one component of those points: first, the kriging mean and variance of a very large sample $\mathbb{X} \subset X$ is computed. The calculation of $\widehat{y_j}(\mathbb{X})$ and $s_j^2(\mathbb{X})$ is indeed tractable for large samples contrarily to GP simulations. Next, s designs are picked up from $\mathbb{X}$ using these computations. In order to avoid losing diversity, the selection is performed using an importance sampling procedure (Bect et al., 2017), based on the probability of contributing to the components I_j or N_j.

As $I_j = \min_{\mathbf{x} \in X} f_j(\mathbf{x})$ good candidates are $\mathbf{x}$'s such that $\mathbb{P}(Y_j(\mathbf{x}) < a_j)$ is large. To account for new evaluations of $f_j(\cdot)$, a typical value for a_j is the minimum observed value in the j-th objective, $\min_{i=1,\ldots,t} f_j(\mathbf{x}^{(i)})$. According to the surrogate, such points have the greatest probability of improving over the currently best value if they were evaluated.

Selecting candidates for estimating $\mathbf{N}$ is more demanding. As seen in the Definition 2.8, N_j is not the maximum value over the whole objective space Y but over the unknown $\mathcal{P}_\mathcal{Y}$, i.e., each N_j arises from a ND point. Thus the knowledge of an m-dimensional front is mandatory for estimating $\mathbf{N}$. The best candidates for $\mathbf{N}$'s estimation are extreme design points (Definition 2.10). Quantifying which points are the most likely to contribute to the Nadir components, in other terms produce extreme points, is a more difficult task than its pendant for the Ideal. Good candidates are $\mathbf{x}$'s such that the sum of probabilities $\mathbb{P}(Y_j(\mathbf{x}) > \overline{\nu}_j^j, \mathbf{Y}(\mathbf{x}) \text{ ND}) + \mathbb{P}(\mathbf{Y}(\mathbf{x}) \preceq \overline{\boldsymbol{\nu}}^j)$ is large, where $\overline{\boldsymbol{\nu}}$ are the extreme points of the empirical Pareto front $\widehat{\mathcal{P}_\mathcal{Y}}$. For reasons of brevity, the procedure is detailed in Appendix C. An illustration of the selected $\mathbf{x}$'s where the GP simulations are performed is given in Appendix D.

Since the optimization is directed towards the center of the Pareto front, the metamodel may lack precision at extreme points. It might be tempting to episodically target these parts of the Pareto front to improve $\mathbf{I}$ and $\mathbf{N}$'s estimation. But this goes against the limited budget of calls to $\mathbf{f}(\cdot)$ and it is not critical since the center is quite stable with respect to $\mathbf{I}$ and $\mathbf{N}$'s inaccuracies (Proposition 4.4). Since the optimality of solutions is favored over the attainment of the exact center of the Pareto front, this option has not been further investigated.

Ideal-Nadir line and estimated center

To estimate $\mathbf{I}$ and $\mathbf{N}$, we first select $s = 5000$ candidates from a large space-filling DoE (Halton, 1960; Sobol', 1967), $\mathbb{X} \subset X$, with a density proportional to their probability of generating either a I_j or a N_j as discussed before. $s/2m$ points are selected for the estimation of each component of $\mathbf{I}$ and $\mathbf{N}$. n_{sim} conditional GP simulations are then performed at those $\mathbf{x}^{(t+i)}, i = 1, \dots, s$ in order to generate simulated fronts, whose Ideal and Nadir points are aggregated through the medians to produce the estimated $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}}$. The resulting simulated fronts are biased towards particular parts of the Pareto front (extreme points, individual minima). Finally, the estimated center $\widehat{\mathbf{C}}$ is the projection of the closest point of $\widehat{\mathcal{P}_\mathcal{Y}}$ on the estimated Ideal-Nadir line, $\widehat{\mathcal{L}}$.

Linearly extending the Pareto front approximation following the approach of Hartikainen et al. (2012) and taking the intersection with $\widehat{\mathcal{L}}$ was originally considered for defining $\widehat{\mathbf{C}}$. However, when $m > 2$, the prolongation of $\widehat{\mathcal{P}_\mathcal{Y}}$ is a collection of polytopes of dimension at most $m - 1$ (Singh et al., 2016). To preserve non-domination among all members, the concept of inherent non-dominance was defined in Hartikainen et al. (2011a,b). As a result, some polytopes of dimension $m - 1$ are removed and others of dimension lower than $m - 1$ belong to the extension. Therefore, $\widehat{\mathcal{L}}$ does not necessarily cross the extended Pareto front approximation.

Experiments have shown significant benefits over methodologies that choose the simulation points $\mathbf{x}^{(t+i)}$ according to their probability of being not dominated by the whole approximation front, or that use s points from a space-filling DoE (Morris and Mitchell,

1995) in X. Figure 4.10 compares the component estimation of $\mathbf{I}$ and $\mathbf{N}$ for different techniques during one optimization run of the MetaNACA with $m = 3$ objectives. X.IN (blue curve) corresponds to our methodology. The other curves stand for competing methodologies: X.LHS (green) selects the $\mathbf{x}^{(t+i)}$ from a space-filling design, and X.ND (red) chooses them according to their probability of being non-dominated with respect to the entire front. NSGA-II (gold) does not select design points $\mathbf{x}^{(t+i)}$ to perform GP simulations but rather uses the Ideal and Nadir point found by one run of the NSGA-II (Deb et al., 2002) multi-objective optimizer applied to the kriging predictors $\widehat{y}_j(\cdot)$, $j = 1, \dots, m$. The black dashed line corresponds to the component of the current empirical front ($\bar{\mathbf{I}}$ and $\overline{\mathbf{N}}$), a computationally much cheaper estimator. The bold dashed line shows $\mathbf{I}$ and $\mathbf{N}$'s true components.

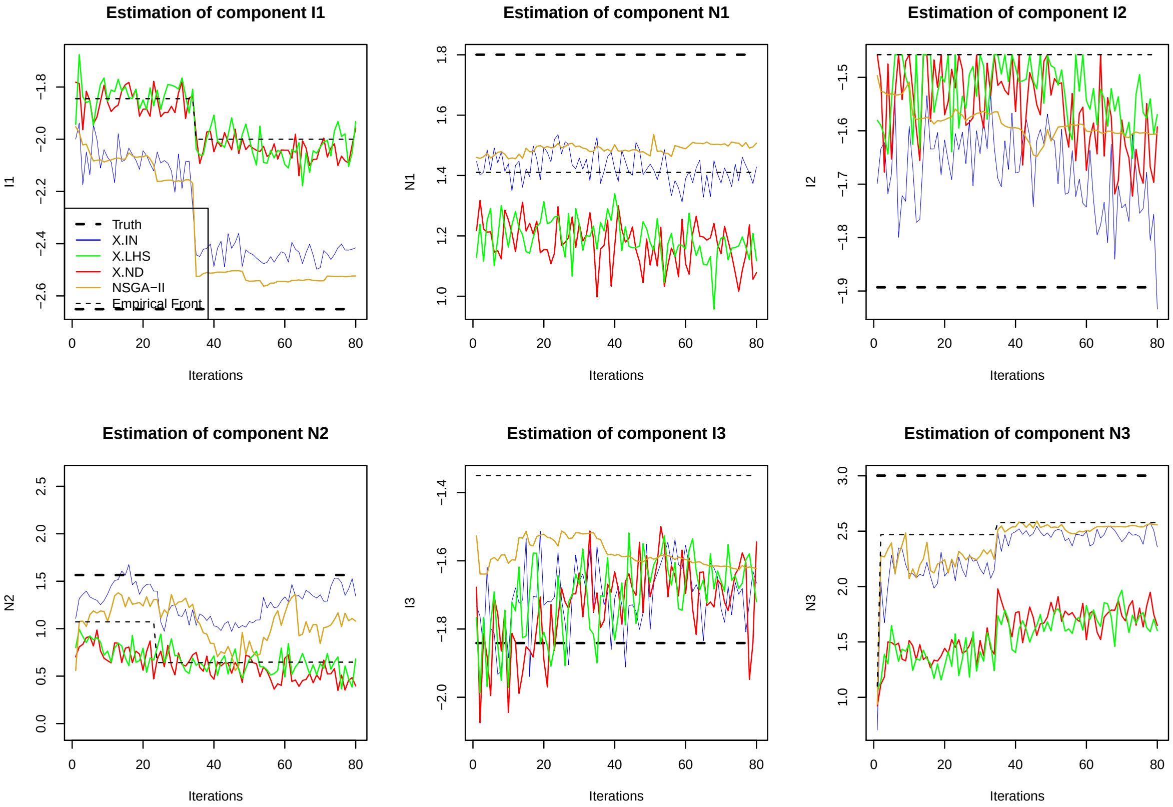


Figure 4.10: Estimation of $\mathbf{I}$ and $\mathbf{N}$ using different techniques. The proposed methodology (blue) is able to consistently produce close estimates to $\mathbf{I}$'s and $\mathbf{N}$'s components (bold black dashed line).

Our methodology outperforms the two other simulation techniques, because they do not perform the simulations specifically at locations that are likely to correspond to an extreme design point or to a single-objective minimizer. Benefits are also observed compared with the empirical Ideal and Nadir points, that are sometimes poor estimators (for example for I_1, I_2 and N_2). Using the output of a multi-objective optimizer (here

NSGA-II) applied to the kriging mean functions is also a promising approach but has the drawback of not considering any uncertainty in the surrogates (that may be large at the extreme parts of the Pareto front). It also suffers from classical EMOA's disadvantages, e.g. several runs would be required for more reliable results and convergence can not be guaranteed. Note that as these methods rely on the surrogates, they are biased by the earlier observations: the change of the empirical Ideal or Nadir point has an impact on the estimation. However, the X.IN, X.LHS and X.ND estimators compensate by considering the GPs uncertainty to reduce this bias.

As we are in fine not interested in the Ideal and the Nadir point but in the Pareto front center, we want to know if these estimations lead to a good $\widehat{\mathbf{C}}$. Proposition 4.4 suggests that the small Ideal and Nadir estimation error should not be a too serious concern. This is confirmed by Figure 4.11, where the center estimation error is low with respect to the range of the Pareto front.

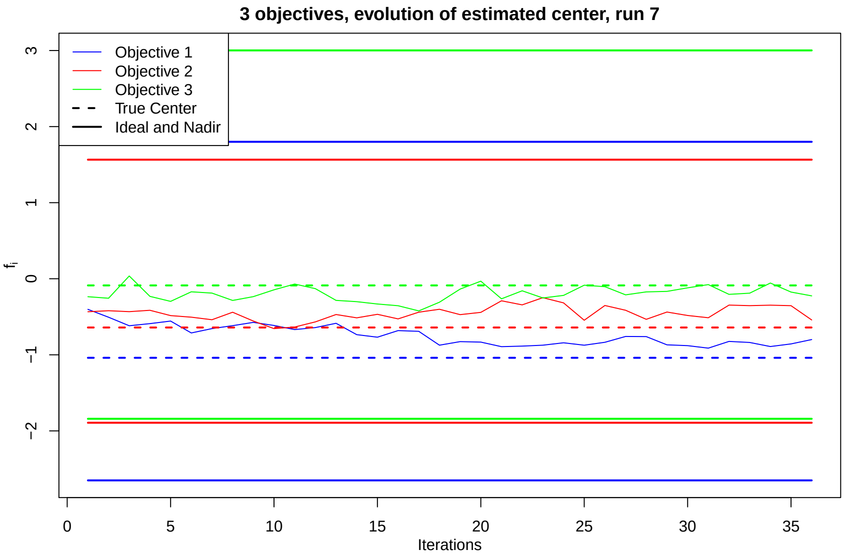


Figure 4.11: Evolution of the estimated center during one run (using $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}}$ from Figure 4.10): $\widehat{\mathbf{C}}$'s components are close to the true ones in regard of the range of $\mathcal{P}_{\mathcal{Y}}$ (between the horizontal bars).

Figure 4.12 shows an example of one GP simulation targeting the extreme points of the Pareto front. Notice the difference between the current empirical Pareto front (in blue) and the simulated front for **I** and **N** (in black): the extreme points which are simulated go well beyond those already observed.

4.4.3 Experiments: targeting with the mEI criterion

This section illustrates the targeting capabilities of mEI (4.5), both when the attainment of a target or the unveiling of the Pareto front center is aimed at. Examples on the MetaNACA benchmark (Chapter 3) illustrate the logic of the criterion as well as the

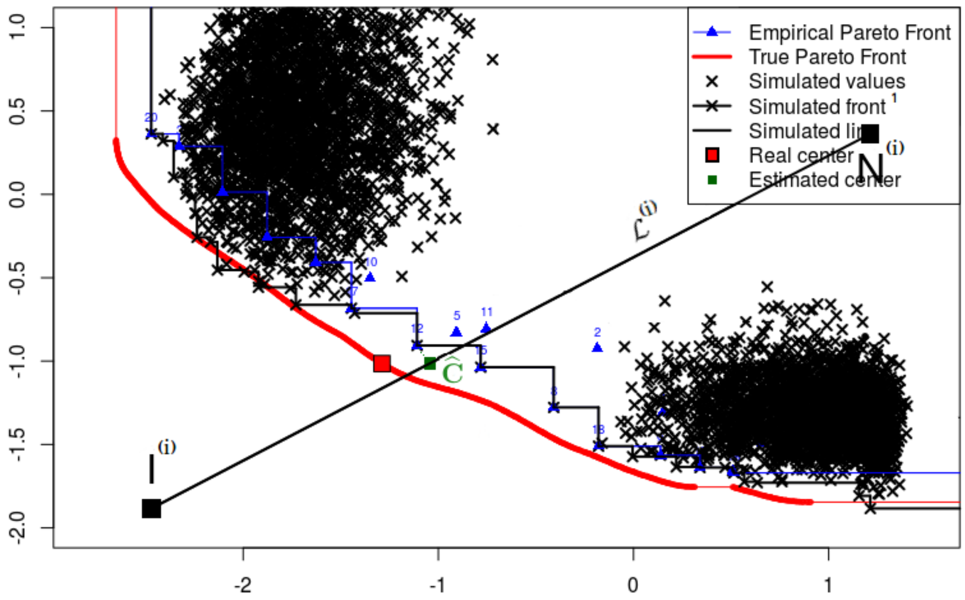


Figure 4.12: One (among the n_{sim}) GP simulation targeting the extreme points of the Pareto front to enhance the estimation of $\mathbf{I}$ and $\mathbf{N}$. The projection of the closest non-dominated point to $\mathcal{L}$ on it is the estimated center (in green). The real center (in red) lies close to the estimated center and to the estimated Ideal-Nadir line.

reference point update mechanism. The mEI targeting capabilities are compared with state-of-the-art multi-objective optimizers (the Bayesian EHI, Emmerich et al., 2006, and the evolutionary algorithm NSGA-II, Deb et al., 2002) on the MetaNACA and on the P1 (Parr, 2013) and ZDT3 (Zitzler et al., 2000) test problems.

4.4.3.1 Targeting a user-defined region

The proposed methodology is applied to the MetaNACA benchmark. The chosen version of the problem is the one with $d = 8$ dimensions and $m = 2$ objectives, the negative lift and the drag, to be minimized. The target $\mathbf{R} = (-1.7, 0)^\top$ is provided to explicitly target the associated region $\mathcal{I}_{\mathbf{R}}$. A sample convergence of the $\mathbf{R}$-mEI algorithm is shown in Figure 4.13 through the sampled $\mathbf{f}(\mathbf{x}^{(i)})$'s (blue triangles) and Figure 4.14 gives the associated updated aspiration points $\widehat{\mathbf{R}}$. mEI$(\cdot, \widehat{\mathbf{R}})$ effectively guides the search towards the region of progress over $\mathbf{R}$. Upon closer inspection, it is seen that the points are not spread within $\mathcal{I}_{\mathbf{R}}$ as they would be with EHI$(\cdot, \mathbf{R})$ because the mEI criterion targets a single point $(\widetilde{\mathbf{R}})$ on the Pareto front.

For a more significant analysis of the ability of mEI to attain a region of the Pareto front defined through a target $\mathbf{R}$, two popular analytical test functions for multi-objective optimization are considered. The first one is the ZDT3 (Zitzler et al., 2000) function which is represented in Figure 4.15. The Pareto set and front of this bi-objective problem consist of five disconnected parts, and we target the second sub-front by setting $\mathbf{R}$ to its Nadir,

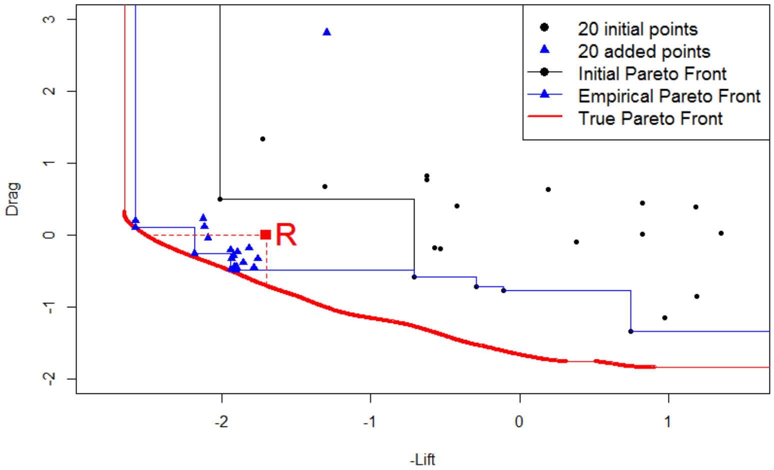


Figure 4.13: Optimization run targeting an off-centered part of the Pareto front through $\mathbf{R}$. After 20 iterations, the Pareto front approximation has been improved in the left part, as specified by $\mathbf{R}$. The successive reference points $\widehat{\mathbf{R}}$ used by mEI are shown in Figure 4.14.

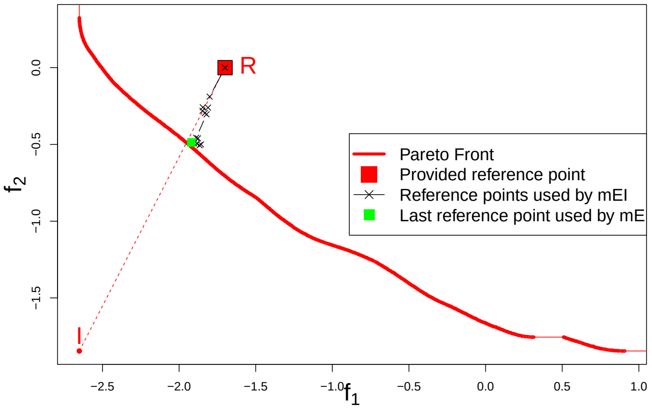


Figure 4.14: Reference points $\widehat{\mathbf{R}}$ successively used for directing the search during the run of Figure 4.13, where $\mathbf{R} = (-1.7, 0)^\top$ is provided. $\widehat{\mathbf{R}}$ adjusts to the current approximation front to direct the algorithm in a region of the Pareto front that dominates $\mathbf{R}$.

$\mathbf{R} = (0.258, 0.670)^\top$. In the $d = 4$ dimensional version of ZDT3 that we consider in the following experiments, less than $V_{\mathbf{R}} = 0.003\%$ of the input space $X = [0, 1]^d$ overshoots this target.

In the second experiment, we consider the P1 benchmark problem of Parr (2013) which is also plotted in Figure 4.15. It has $d = 2$ dimensions, and we target the part of the objective space such that $f_1(\mathbf{x}) \leq 10$ and $f_2(\mathbf{x}) \leq -23$ by setting $\mathbf{R} = (10, -23)^\top$. This corresponds to approximately $V_{\mathbf{R}} = 0.9\%$ of the design space, $X = [0, 1]^2$.

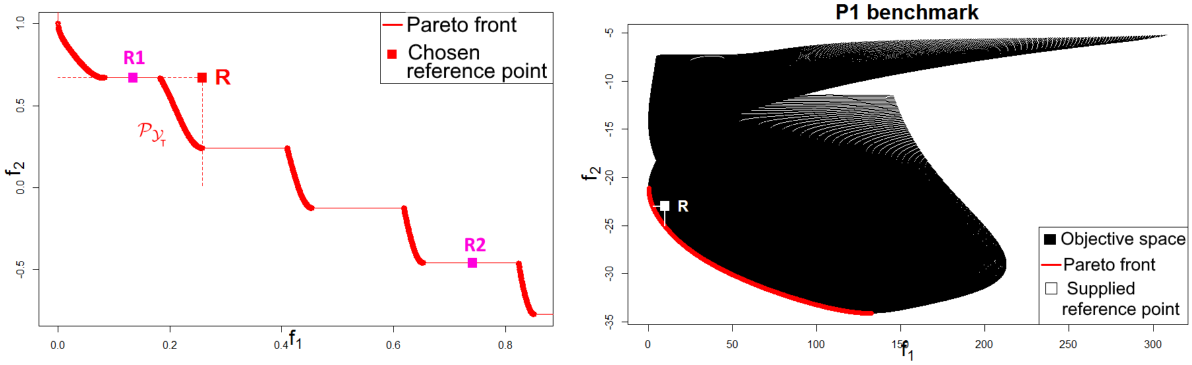


Figure 4.15: Left: Pareto front of the ZDT3 problem and chosen $\mathbf{R}$. $\mathcal{P}_{\mathcal{Y}_T}$ is targeted. Right: objective space and Pareto front of the P1 problem, and targeted region defined through $\mathbf{R}$.

$\mathbf{R}$-mEI's ability to produce user-desired solutions is compared with the EHI acquisition function (Emmerich et al., 2006), and with the evolutionary algorithm NSGA-II (Deb et al., 2002). The default choice in `GPareto` (Binois and Picheny, 2015) is used for EHI's reference point, i.e. $1.1\overline{\mathbf{N}} - 0.1\overline{\mathbf{I}}$ (re-computed at each iteration). For ZDT3, the frequently advised (Emmerich et al., 2020; Yang et al., 2019c) reference point $(11, 11)^\top$ is also investigated. In the ZDT3 problem, $\mathbf{R}$-mEI and EHI start with an initial design of experiments of size $n = 20$ and are run for $p = 20$ additional iterations. For NSGA-II, a population of 20 individuals is used, and the results are shown after the second generation (i.e., after 20 additional function evaluations, for comparison at the total budget of mEI and EHI), and after 19 supplementary generations. Runs of NSGA-II with different number of generations and population sizes have also been investigated, but since they do not change our conclusions and for the sake of brevity, they are not shown here.

In the P1 problem, the size of the initial design of experiments is $n = 8$, and $\mathbf{R}$-mEI and EHI are run for $p = 12$ iterations. NSGA-II is run for 1 and for 11 supplementary generations, with a population of 12 individuals. Table 4.1 summarizes the configuration of both experiments.

Test function	User target $\mathbf{R}$	$V_{\mathbf{R}}$ (%)	d	n	p
P1	$(10, -23)^\top$	0.9	2	8	12
ZDT3	$(0.258, 0.670)^\top$	0.003	4	20	20

Table 4.1: Benchmark problem configurations.

The attainment time (Definition 2.14) of $\mathbf{R}$ including the n initial function evaluations is the first comparison metric. × indicates that no run was able to attain $\mathbf{R}$ within the prescribed budget. An estimator for the expected runtime (Auger and Hansen, 2005)[5] is given in red if at least one run did not reach this target, together with the number of

[5] A rough estimator for the expected runtime is $\overline{T}_s/p_s$ where $\overline{T}_s$ and p_s correspond to the runtime of successful runs and to the proportion of successful runs respectively

successful runs in brackets. The second metric is the hypervolume indicator (Definition 2.11). The latter is computed up to $\mathbf{R}$ which restricts it to the part of the Pareto front which dominates $\mathbf{R}$. I_H is normalized by the hypervolume of the true Pareto front of each problem $\mathcal{P}_{\mathcal{Y}}$ such that the upper bound for the indicator is 1. A third metric is the number of obtained solutions that dominate $\mathbf{R}$, i.e., that fall in the preferred region. These indicators are averaged over 10 runs starting from different initializations, and their standard deviations given in brackets in Tables 4.2 and 4.3. In this table, NSGA-II's subscript corresponds to the number of additional generations that have been evaluated.

	mEI	EHI	EHI$_{(11,11)}$	NSGA-II$_1$	NSGA-II$_{19}$
$\#\mathbf{f}(\cdot)$	20+20	20+20	20+20	20+20	20+380
Attainment time	24.2 (2.6)	45.3 [7]	103.3 [3]	×	341.5 [7]
Hypervolume	0.634 (0.078)	0.218 (0.353)	0.112 (0.211)	0	0.248 (0.253)
Solutions $\prec \mathbf{R}$	4.1 (1.8)	1.1 (1.9)	0.3 (0.5)	0	4.2 (4.1)

Table 4.2: Comparison of the algorithms on the ZDT3 function, with respect to the three metrics. The results are averaged over 10 runs, and the standard deviation is shown in brackets. The number of function evaluations for each method can be found in the row $\#\mathbf{f}(\cdot)$ (initial design + additional function evaluations).

	mEI	EHI	NSGA-II$_1$	NSGA-II$_{11}$
$\#\mathbf{f}(\cdot)$	8+12	8+12	12+12	12+132
Attainment time	12.6 (3.5)	25.6 [5]	120 [1]	67.1 [8]
Hypervolume	0.620 (0.165)	0.163 (0.213)	0.043 (0.136)	0.394 (0.295)
Solutions $\prec \mathbf{R}$	6.5 (2.5)	0.6 (0.7)	0.2 (0.6)	2.8 (2.4)

Table 4.3: Comparison of the algorithms on the P1 function, with respect to the three metrics. The results are averaged over 10 runs, and the standard deviation is shown in brackets. The number of function evaluations for each method can be found in the row $\#\mathbf{f}(\cdot)$ (initial design + additional function evaluations).

These results confirm that mEI is able to consistently produce solutions in the user-defined part of the Pareto front within a limited number of iterations: all mEI runs attain $\mathcal{I}_{\mathbf{R}}$ contrarily to EHI, which attains the region 7 times out of 10 on ZDT3, and 5 times out of 10 on P1. At the same budget, NSGA-II almost never attains $\mathcal{I}_{\mathbf{R}}$, and some runs still do not reach it despite larger budgets, on both problems. In both experiments mEI takes the least function evaluations to attain $\mathbf{R}$.

In comparison with EHI, mEI better converges to $\mathcal{P}_{\mathcal{Y}}$ in $\mathcal{I}_{\mathbf{R}}$ as confirmed by the larger hypervolume (even though mEI's logic is solely to overshoot $\mathbf{R}$ and not necessarily a good distribution in $\mathcal{I}_{\mathbf{R}}$, as highlighted by Figure 4.14). The smaller standard deviations confirm that the results of mEI are more repetitive. Indeed some EHI runs converge to $\mathcal{P}_{\mathcal{Y}_T}$ while other runs do not. Remark that EHI with $(11, 11)^\top$ as reference point leads to even less points inside $\mathcal{I}_{\mathbf{R}}$ than the variant where EHI's reference point is computed accordingly to $\bar{\mathbf{I}}$ and $\overline{\mathbf{N}}$. Even for much larger budgets, the hypervolume indicator in $\mathcal{I}_{\mathbf{R}}$ obtained by NSGA-II is much smaller.

Last, more solutions dominating $\mathbf{R}$ are produced by mEI than by the other algorithms. Even though this indicator may not be as meaningful as the others since it does not measure the fast attainment of $\mathbf{R}$, the convergence to $\mathcal{P}_{\mathcal{Y}}$ inside $\mathcal{I}_{\mathbf{R}}$, nor the solution's diversity, it evidences that mEI is a criterion capable of directing the optimization towards desired and difficult-to-attain solutions.

4.4.3.2 Targeting a hole

To prove that the mEI acquisition function is able to cope with any kind of Pareto front, let us continue with ZDT3 and target $\mathbf{R1} = (0.133, 0.665)^\top$ and $\mathbf{R2} = (0.738, -0.465)^\top$. These vectors correspond to the middle of the hole between ZDT3's first and second front, and to the middle between ZDT3's fourth and fifth front, respectively (see Figure 4.15). They are utopian since $\nexists \mathbf{x} \in X : \mathbf{f}(\mathbf{x}) \preceq \mathbf{R1}$ or $\mathbf{R2}$. Even worse, solutions in their vicinity are not Pareto-optimal. However, mEI naturally adapts and rapidly drives the search towards the closest Pareto optimal solutions in the neighborhood of $\mathbf{R}$ as figured out in Figure 4.16. Because of the shape of ZDT3's objective space ($f_2 \approx 1$ when $f_1 = 0.738 = R2_1$, hence much more than $R2_2 = -0.465$, whereas f_1 is only slightly larger than $R2_1$ when $f_2 = -0.465 = R2_2$), the right-hand side optimization solely attains the fourth front while the left-hand side run reaches both the first and the second sub-front.

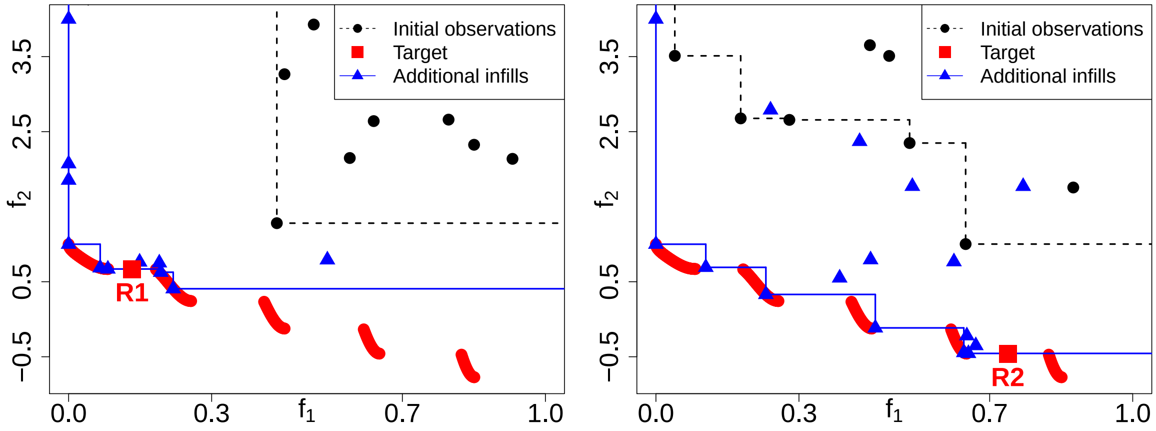


Figure 4.16: When the target $\mathbf{R}$ is at a discontinuity of $\mathcal{P}_{\mathcal{Y}}$, mEI uncovers the closest Pareto optimal solutions.

4.4.3.3 Targeting the center of the Pareto front

When no preferences are given, the center of the Pareto front becomes the implicit target, $\mathbf{R} = \mathbf{C}$. In the following experiments, the center of the MetaNACA benchmark (Figure 4.18) is targeted. Figure 4.17 shows that, compared with standard techniques, the proposed methodology automatically leads to a faster and a more precise convergence to the central part of the Pareto front at the cost of a narrower covering of the front. Figure 4.18 indicates how $\mathbf{R} = \widehat{\mathbf{C}}$ evolves to direct the search to the unknown center of the true Pareto front, $\mathbf{C}$.

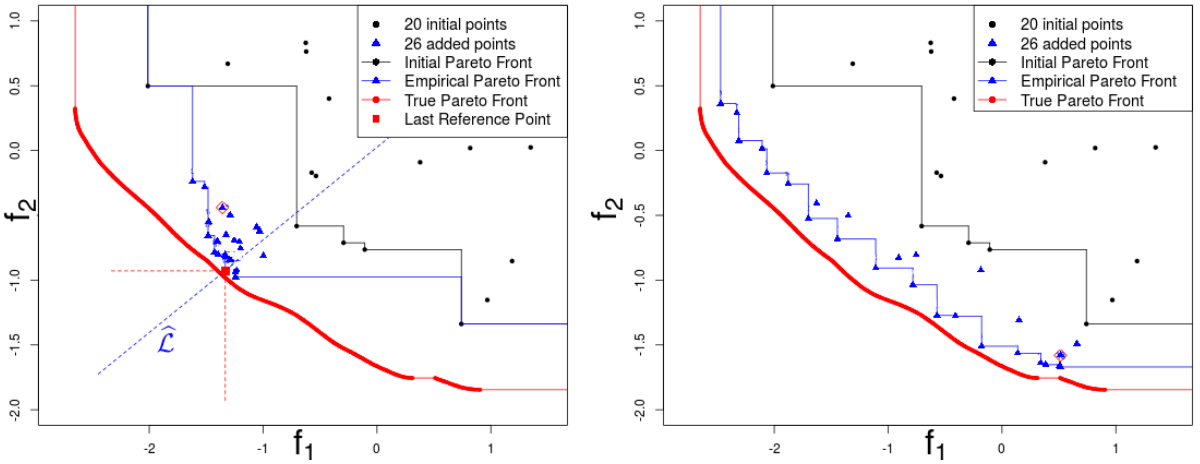


Figure 4.17: Bi-objective optimization with the **C**-mEI algorithm (left). The initial approximation (black) has mainly been improved around the center. Compared with a standard EHI (right), the proposed methodology achieves convergence to the central part of the front. EHI considers more compromises between objectives, but cannot converge within the given budget.

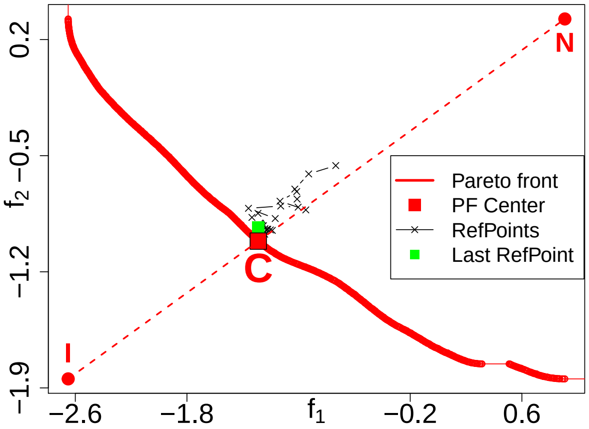


Figure 4.18: Reference points **R** successively used for directing the search during the **C**-mEI run of Figure 4.17. They lie close to the dashed Ideal-Nadir line (**IN**) and lead the algorithm to the center of the Pareto front (**C**).

For more significant results, the ability of **C**-mEI to automatically drive the optimization towards the center of the Pareto front is investigated on the MetaNACA benchmark

(Chapter 3). Again, mEI is compared with EHI and NSGA-II by means of the attainment time and the hypervolume indicator. Since the latter require a target or a reference point, we use $\mathbf{R}_w := (1-w)\mathbf{C}+w\mathbf{N}$, where $\mathbf{C}$ is the center of the Pareto front, and $\mathbf{N}$ the Nadir point of the problem. We take $w = 0.1$ which means that the hypervolume is calculated only for the points that are in a small vicinity[6] of $\mathbf{C}$. Figure 4.27 shows the part of $\mathcal{P}_{\mathcal{Y}}$ to which $\mathbf{R}_w$ refers.

Table 4.4 reports the averages and standard deviations of the performance metrics, calculated over 10 independent runs for the MetaNACA problems with $d = 8$ and 22 parameters, and $m = 2$ objectives (lift and drag). The optimizations start with $n = 20$ or 50 observations and are run for $p = 20$ or 50 supplementary iterations, respectively. NSGA-II is run with a population of 20 individuals, and results are shown after 1 and 19 additional generations when $d = 8$, and after 4 and 24 when $d = 22$ (generations are given in subscript), to compare it with the Bayesian approaches both at the same number of function evaluations, and for 5 times more calls to $\mathbf{f}(\cdot)$. Other population/generation configurations have been tested but are not reported for the sake of brevity since they led to same conclusions. × indicates that no run was able to attain $\mathbf{R}_{0.1}$, and the empirical runtime is reported in red if at least one run did not reach this target, with the number of successful runs in brackets.

	$d = 8$				$d = 22$			
Criterion	mEI	EHI	NSGA-II$_1$	NSGA-II$_{19}$	mEI	EHI	NSGA-II$_4$	NSGA-II$_{24}$
$\#\mathbf{f}(\cdot)$	20+20	20+20	20+20	20+180	50+50	50+50	20+80	20+480
Attainment time	28.4 (5.4)	66.8 [5]	×	261.9 [6]	56.3 (7.2)	71.4 (13.9)	×	191.9 [9]
Hypervolume	0.256 (0.09)	0.025 (0.04)	0	0.044 (0.08)	0.222 (0.12)	0.153 (0.09)	0	0.106 (0.07)

Table 4.4: Comparison of the different infill criteria and algorithms for the MetaNACA. The metrics are averaged over 10 runs, and the standard deviation is shown in brackets. The number of function evaluations for each technique can be found in the row $\#\mathbf{f}(\cdot)$.

These empirical results indicate that mEI is able to automatically direct the optimization towards the center of the Pareto front. It attains $\mathcal{I}_{0.1} := \mathcal{I}_{\mathbf{R}_{0.1}}$ in each run and in the smallest number of function evaluations. mEI better converges towards the central part of $\mathcal{P}_{\mathcal{Y}}$ than EHI as measured by the hypervolume indicator in $\mathcal{I}_{0.1}$. EHI attempts to uncover the whole Pareto front but does not get as close to the true Pareto front's center, and in the problem in dimension $d = 8$, 5 EHI runs out of 10 did not reach $\mathcal{I}_{0.1}$ within the prescribed budget. At the same budget, NSGA-II is not able to produce solutions within $\mathcal{I}_{0.1}$, and some runs are not able to reach it even for 5 times larger budgets. The MetaNACA in dimension $d = 22$ gives the same conclusions.

[6] In case of a linear Pareto front, $\mathcal{I}_{0.1}$ corresponds to the 10% most central solutions

4.5 Detecting local convergence to the Pareto front

The Pareto front may be locally reached before depletion of the computational resources. If the algorithm continues targeting the same region, no improvement can be obtained, and the infill criterion will tend to favor the most uncertain parts of the design space. As sketched in Figure 4.1, the aim of our optimization algorithm is to first converge towards relevant solutions (the attainable target $\widetilde{\mathbf{R}}$ or the center $\mathbf{C}$), before unveiling a broader part of $\mathcal{P}_\mathcal{Y}$ around the initial goal. It is necessary to detect local convergence to the Pareto front so that a wider search can be conducted in the remaining iterations, as will be explained in Section 4.6. In this section, we propose a novel method for checking convergence to the center. It does not utilize the mEI value which was found too unstable (since $\mathbf{R}$ fluctuates at each iteration) to yield a reliable stopping criterion. Instead, the devised test relies on a measure of local uncertainty.

To test the convergence to a local part of the Pareto front, we define the probability of domination in the Y space[7], $p(\mathbf{y})$, as the probability that there exists $\mathbf{y}' \in Y : \mathbf{y}' \preceq \mathbf{y}$. $\mathbf{y}$'s for which $p(\mathbf{y})$ is close to 0 or to 1 have a small or large probability, respectively, that there exist objective vectors dominating them. On the contrary, $p(\mathbf{y})$ close to 0.5 indicates no clear knowledge about the chances to find better vectors than $\mathbf{y}$. $p(\mathbf{y})$ measures how certain domination or non-domination of $\mathbf{y}$ is. Formally, the domination $d(\mathbf{y})$ is a binary variable that equals 1 if $\exists \mathbf{x} \in X : \mathbf{f}(\mathbf{x}) \preceq \mathbf{y}$ and 0 otherwise. The Pareto front being a boundary for domination, $d(\cdot)$ can also be expressed in the following way

$$d(\mathbf{y}) = \begin{cases} 1 \text{ if } \mathcal{P}_\mathcal{Y} \preceq \mathbf{y}, \\ 0 \text{ otherwise.} \end{cases}$$

$d(\mathbf{y})$ can be thought of as a binary classifier between dominated and non-dominated vectors whose frontier is the Pareto front and which is only known for previous observations ($d(\mathbf{y}^{(i)}) = 1$ if $\mathbf{y}^{(i)} \notin \widehat{\mathcal{P}_\mathcal{Y}}$). The metamodeling of $d(\cdot)$ is the approach followed by Loshchilov et al. (2010) where rather than the objective functions, the Pareto dominance of $\mathbf{x}$ is modeled. The probability that a design is Pareto optimal is also utilized within the acquisition function in Davins-Valldaura et al. (2017).

We now consider an estimator $D(\mathbf{y})$ of $d(\mathbf{y})$ that has value 1 when the random Pareto front of the GPs, $\mathcal{P}_{\mathbf{Y}(\cdot)}$, dominates $\mathbf{y}$, and has value 0 otherwise,

$$D(\mathbf{y}) = \mathbb{1}(\mathcal{P}_{\mathbf{Y}(\cdot)} \preceq \mathbf{y}).$$

The reader interested in theoretical background about the random set $\mathcal{P}_{\mathbf{Y}(\cdot)}$ is referred to Binois (2015); Molchanov (2005).

$D(\mathbf{y})$ is a Bernoulli variable closely related to the domination probability through $p(\mathbf{y}) = \mathbb{P}(D(\mathbf{y}) = 1) = \mathbb{E}[D(\mathbf{y})]$. If $p(\mathbf{y})$ goes quickly from 0 to 1 as $\mathbf{y}$ crosses the Pareto front, the front is precisely known around this $\mathbf{y}$.

[7] The probability of domination is also called “attainment function” in Binois (2015).

As the $Y_j(\cdot)$'s are independent, it is easy to calculate the probability of domination for a specific $\mathbf{x}$, $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \preceq \mathbf{y}) = \prod_{j=1}^m \phi_{\mathcal{N}}\left(\frac{y_j - \widehat{y}_j(\mathbf{x})}{s_j(\mathbf{x})}\right)$, a product of probabilities of improvement (Equation 2.7). In contrast, the probability of dominating $\mathbf{y}$ at any $\mathbf{x}$ by $\mathbf{Y}(\mathbf{x})$, $\mathbb{P}(\exists \mathbf{x} \in X : \mathbf{Y}(\mathbf{x}) \preceq \mathbf{y})$, has no closed-form as many overlapping cases occur. Even for a discrete set $\mathbb{X} = \{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+s)}\}$, $\mathbb{P}(\exists \mathbf{x} \in \mathbb{X} : \mathbf{Y}(\mathbf{x}) \preceq \mathbf{y})$ has to be estimated by numerical simulation because of the correlations in the Gaussian vector $\mathbf{Y}(\mathbb{X})$.

To estimate the probability $p(\mathbf{y})$ that an objective vector $\mathbf{y}$ can be dominated, we exploit the probabilistic nature of the GPs conditioned by previous observations: Pareto fronts $\widetilde{\mathcal{P}_\mathcal{Y}}^{(k)}, k = 1, \dots, n_{sim}$ are extracted from n_sim simulated GPs. $D^{(k)}(\cdot)$ is a realization of the estimator and random variable $D(\mathbf{y})$,

$$D^{(k)}(\mathbf{y}) = \mathbb{1}(\widetilde{\mathcal{P}_\mathcal{Y}}^{(k)} \preceq \mathbf{y}) = \begin{cases} 1 \text{ if } \exists \mathbf{z} \in \widetilde{\mathcal{P}_\mathcal{Y}}^{(k)} : \mathbf{z} \preceq \mathbf{y}, \\ 0 \text{ otherwise.} \end{cases}$$

Therefore, $p(\mathbf{y})$ which is the mean of $D(\mathbf{y})$ can be estimated by averaging the realizations,

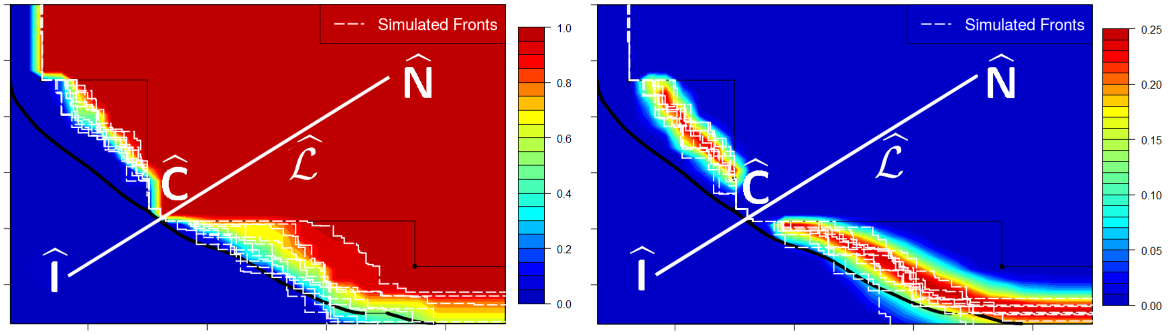
$$p(\mathbf{y}) = \lim_{n_{sim} \to \infty} \widehat{p}(\mathbf{y}) \qquad \text{where} \qquad \widehat{p}(\mathbf{y}) = \frac{1}{n_{sim}} \sum_{k=1}^{n_{sim}} D^{(k)}(\mathbf{y}) \ .$$

One can easily check that $\widehat{p}(\mathbf{y})$ is monotonic with domination: if $\mathbf{y}' \preceq \mathbf{y}$, then every $\widetilde{\mathcal{P}_\mathcal{Y}}^{(k)}$ dominating $\mathbf{y}'$ will also dominate $\mathbf{y}$ and $\widehat{p}(\mathbf{y}') \leq \widehat{p}(\mathbf{y})$.

As discussed in Section 4.4.2.3, the choice of points $\mathbf{x}^{(t+i)} \in X, i = 1, \dots, s$ where the GP simulations are performed is crucial. Here, as the simulated Pareto fronts aim at being possible versions of the true front, the $\mathbf{x}$'s are chosen according to their probability of being non-dominated with regard to the current approximation $\widehat{\mathcal{P}_\mathcal{Y}}$, $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND})$, in a roulette wheel selection procedure (Deb, 2001) to maintain both diversity and a selection pressure. An illustration of the selected $\mathbf{x}$'s where the GP simulations are performed is given in Appendix D. Simulating the GPs on a space-filling DoE (Halton, 1960; Morris and Mitchell, 1995; Sobol', 1967) leads to less dominating simulated fronts hence an under-estimated probability of dominating $\mathbf{y}$. Another advantage of the estimation of $p(\cdot)$ through GP simulations is that the computational burden resides in the $\mathbf{x}$ selection procedure and the simulation of the GPs. Once the simulated fronts have been generated, $p(\cdot)$ can be estimated for many $\mathbf{y}$'s $\in Y$ without significant additional effort.

The variance of the Bernoulli variable $D(\mathbf{y})$ is $p(\mathbf{y})(1 - p(\mathbf{y}))$ and can be interpreted as a measure of uncertainty about dominating $\mathbf{y}$. When $p(\mathbf{y}) = 1$ or 0, no doubt subsists regarding the fact that $\mathbf{y}$ is dominated or non-dominated, respectively. When half of the simulated fronts dominate $\mathbf{y}$, $p(\mathbf{y}) = 0.5$ and $p(\mathbf{y})(1 - p(\mathbf{y}))$ is maximal: uncertainty about the domination of $\mathbf{y}$ is at its highest.

Here, we want to check convergence to the Pareto front in the preferred part of $\mathcal{P}_\mathcal{Y}$ (the center, or the user-desired region) which is located on the estimated Ideal-Nadir line $\widehat{\mathcal{L}}$ or the estimated Ideal-Target-Nadir line $\widehat{\mathcal{L}'}$, respectively (see Figure 4.7 or Figure 4.12). For the sake of brevity, $\widehat{\mathcal{L}}$ is used in the following, but the approach is identical for $\widehat{\mathcal{L}'}$. We



(a) $\widehat{p}$ in Y space

(b) $\widehat{p}(1-\widehat{p})$ in Y space

Figure 4.19: Detection of convergence to the Pareto front center using simulated fronts. Five of the $n_{sim} = 200$ simulated fronts are shown in white. The approximation $\widehat{\mathcal{P}_\mathcal{Y}}$ (thin black line) has converged towards $\mathcal{P}_\mathcal{Y}$ (thick black curve) at the center of the front (intersection with $\widehat{\mathcal{L}}$). Consequently, $p(\mathbf{y})$ grows very fast from 0 to 1 along $\widehat{\mathcal{L}}$ and the domination uncertainty on the right plot $p(\mathbf{y})(1-p(\mathbf{y}))$ is null.

consider the uncertainty measure $(p(\mathbf{y})(1-p(\mathbf{y})))$ for $\mathbf{y}$ varying along $\widehat{\mathcal{L}}$, convergence at the center being equivalent to a sufficiently small uncertainty of $D(\mathbf{y})$ along $\widehat{\mathcal{L}}$. This leads to saying that convergence to the center has occurred if the *line uncertainty* is below a threshold, $U(\widehat{\mathcal{L}}) < \varepsilon$, where the line uncertainty is defined as

$$U(\widehat{\mathcal{L}}) := \frac{1}{|\widehat{\mathcal{L}}|} \int_{\widehat{\mathcal{L}}} p(\mathbf{y})(1-p(\mathbf{y}))d\mathbf{y}. \tag{4.7}$$

$|\widehat{\mathcal{L}}|$ is the (Euclidean) distance between the estimated Ideal and Nadir points and ε is a small positive threshold. Figure 4.19 illustrates a case of detection of convergence to the Pareto front center. On the left plot, when moving along $\widehat{\mathcal{L}}$ from $\widehat{\mathbf{I}}$ to $\widehat{\mathbf{N}}$, $p(\cdot)$ goes quickly from 0 to 1 when crossing the estimated and real Pareto fronts. The variability between the simulated Pareto fronts is low in the central part, as seen on the right plot: $p(\mathbf{y})(1-p(\mathbf{y}))$ equals 0 (up to estimation precision) all along $\widehat{\mathcal{L}}$ and in particular near the center of the approximation front where sufficiently many points $\mathbf{f}(\mathbf{x}^{(i)})$ have been observed and no further improvement can be achieved.

If $p(\mathbf{y})$ equals either 0 or 1 along $\widehat{\mathcal{L}}$, all n_{sim} simulated fronts are intersected at the same location by $\widehat{\mathcal{L}}$, thus convergence is assumed in this area. To set the threshold ε, we consider that convergence has occurred in the following limit scenarios: as there are 100 integration points on $\widehat{\mathcal{L}}$ for the computation of the criterion (4.7), $p(\mathbf{y})$ jumps successively from 0 to 0.05 and 1 (or from 0 to 0.95 and 1); or $p(\mathbf{y})$ jumps successively from 0 to 0.025, 0.975 and 1. This rule leads to a threshold $\varepsilon = 10^{-3}$.

Remark 4.3. *To select the* $\mathbf{x}$*'s where to simulate the GPs, it may be cumbersome to evaluate* $\mathbb{P}(\mathbf{Y}(\mathbf{x})$ *ND) on a large space-filling design when the number of non-dominated solutions is large and/or when* m *increases. Since we are mostly interested in* $p(\mathbf{y})$ *for* $\mathbf{y}$*'s which dominate* $\mathbf{R}$ *(on the* $\widehat{\mathcal{L}}$ *line), other relevant selection operators are the* *probability of dominating the reference point $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \preceq \mathbf{R})$ or the mEI$(\mathbf{x};\mathbf{R})$. These criteria are analytically tractable and their complexity only grows linearly in m and does not depend on the cardinality of $\widehat{\mathcal{P}_{\mathcal{Y}}}$. Other measures of local convergence such as the median improvement of the simulated fronts along $\widehat{\mathcal{L}}$ (normalized by the square root of the sum of the GP variances), or the number of simulated fronts that improve $\widehat{\mathcal{P}_{\mathcal{Y}}}$ sufficiently enough in $\mathcal{I}_{\mathbf{R}}$ have also been investigated. In both cases, similar results than those obtained by (4.7) have been observed.*

4.6 Expansion of the approximation front within the remaining budget

If local convergence on the preferred part of the Pareto front is detected and the objective functions *budget* is not exhausted, i.e., b calls to $\mathbf{f}(\cdot)$ are still allowed, the goal is no longer to search at this location where no direct progress is possible, but to investigate a wider part of the Pareto front. A second phase of the algorithm is started during which a new fixed reference point $\mathbf{R}$ is set for the EHI infill criterion. To continue targeting the preferred part of the Pareto front, the new $\mathbf{R}$ has to be located on $\widehat{\mathcal{L}}$ (or $\widehat{\mathcal{L}}'$). The more distant $\mathbf{R}$ is from $\mathcal{P}_{\mathcal{Y}}$, the broader the targeted area in the objective space will be, as $\mathcal{I}_{\mathbf{R}} \subset \mathcal{I}_{\mathbf{R}'}$ if $\mathbf{R} \preceq \mathbf{R}'$. As shown in Figure 4.20, $\mathbf{R}$ is instrumental in deciding in which area solutions are sought. After having spent the b remaining calls to the objective functions, we would like to have (i) an approximation front $\widehat{\mathcal{P}_{\mathcal{Y}}}$ as broad as possible, and (ii) which has converged to $\mathcal{P}_{\mathcal{Y}}$ in the entire targeted area $I_{\mathbf{R}}$. These goals are conflicting: at a fixed remaining budget, the larger the targeted area, the least $\mathcal{P}_{\mathcal{Y}}$ will be well described. The reference point leading to the best trade-off between convergence to the Pareto front and width of the final approximation front is sought.

To choose the best reference point for the remaining b iterations, we anticipate the behavior of the algorithm and the final approximation front obtained with a given $\mathbf{R}$. Candidate reference points $\mathbf{R}^c, c = 1, \dots, C$, are uniformly distributed along $\widehat{\mathcal{L}}$ starting from $\mathbf{R}^0 = \widehat{\mathbf{C}}$ or $\widehat{\mathbf{R}}$, and $\mathbf{R}^C = \widehat{\mathbf{N}}$. Each $\mathbf{R}^c$ is related to an area in the objective space it targets, $\mathcal{I}_{\mathbf{R}^c}$. Departing from the current GPs $\mathbf{Y}(\cdot)$, C virtual optimization scenarios are anticipated by sequentially maximizing EHI b times for each candidate reference point $\mathbf{R}^c$. We use a Kriging Believer (Ginsbourger et al., 2010) strategy in which the metamodel is augmented at each virtual iteration using the kriging mean functions $\widehat{\mathbf{y}}(\mathbf{x}^{*(i)})$, $\mathbf{x}^{*(i)}$ being the maximizer of EHI$(\cdot;\mathbf{R}^c)$ at one of the virtual steps $i = 1, \dots, b$. Such a procedure does not modify the posterior mean $\widehat{\mathbf{y}}(\cdot)$, but it changes the posterior variance $\mathbf{s}^2(\cdot)$. The conditional GPs $\mathbf{Y}(\cdot)$ augmented by these b Kriging Believer steps are denoted as $\mathbf{Y}^{KB}(\cdot)$.

The optimizations for the $\mathbf{R}^c$'s are independent and parallel computing can be exploited (in our implementation, it has been done through the `foreach` `R` package). At the end, C different final Kriging Believer GPs $\mathbf{Y}_c^{KB}(\cdot)$ are obtained that characterize the associated $\mathbf{R}^c$. $\mathbf{R}$'s close to $\mathbf{R}^0 = \widehat{\mathbf{C}}$ or $\widehat{\mathbf{R}}$ produce narrow and densely sampled final fronts whereas distant $\mathbf{R}$'s lead to more extended and sparsely populated fronts, as can be seen in Figure 4.21.

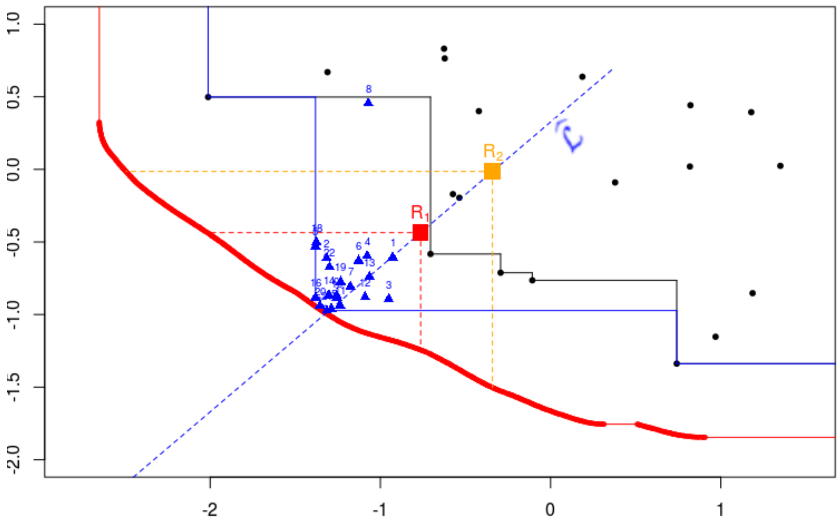


Figure 4.20: Two possible reference points $\mathbf{R}_1$ and $\mathbf{R}_2$ located on $\widehat{\mathcal{L}}$, and the part of the Pareto front they allow to target when used within EHI.

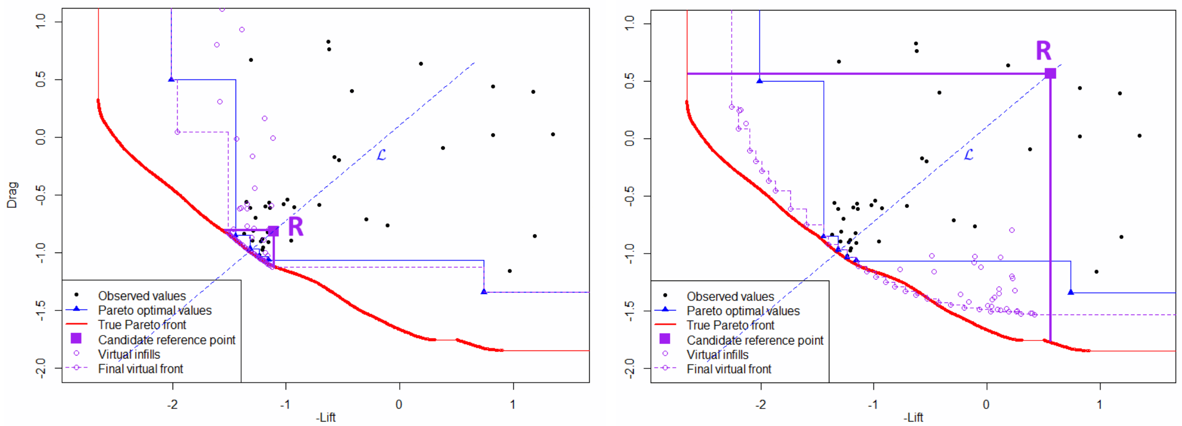


Figure 4.21: Virtual infills (purple circles) obtained by sequentially maximizing EHI$(\cdot;\mathbf{R})$ b times, for two different reference points (purple squares). The shape and sampling density of the final virtual front depends on $\mathbf{R}$.

To measure how much is known about the Pareto front, we generalize the line uncertainty of Equation (4.7) to the volume $\mathcal{I}_{\mathbf{R}}$ and define the *volume uncertainty*, $U(\mathbf{R};\mathbf{Y}(\cdot))$ of the GPs $\mathbf{Y}(\cdot)$. The volume uncertainty is the average domination uncertainty $p(\mathbf{y})(1-p(\mathbf{y}))$ in the volume that dominates $\mathbf{R}$ bounded by the Ideal point where $p(\mathbf{y})$ is calculated for $\mathbf{Y}(\cdot)$,

$$U(\mathbf{R};\mathbf{Y}(\cdot)) := \frac{1}{Vol(\mathbf{I},\mathbf{R})}\int_{\mathbf{I}\preceq\mathbf{y}\preceq\mathbf{R}} p(\mathbf{y})(1-p(\mathbf{y}))d\mathbf{y}\ . \tag{4.8}$$

In practice, the estimated Ideal $\widehat{\mathbf{I}}$ is substituted for the Ideal. $U(\mathbf{R};\mathbf{Y}(\cdot))$ quantifies the convergence to the estimated Pareto front in the progress region delimited by $\mathbf{R}$. It is a more rigorous uncertainty measure than others based on the density of points in the Y space as it accounts for the possibility of having many inverse images $\mathbf{x}$ to $\mathbf{y}$.

The optimal reference point is the one that creates the largest and sufficiently well populated Pareto front. The concepts of augmented GPs and volume uncertainty to measure convergence allow to define the *optimal reference point*,

$$\mathbf{R}^* := \mathbf{R}^{c^*} \quad \text{where} \quad c^* = \max_{c=1,\dots,C} c \quad \text{such that } U(\mathbf{R}^c; \mathbf{Y}_c^{KB}(\cdot)) < \varepsilon \tag{4.9}$$

Note that the uncertainty is calculated with the augmented GPs $\mathbf{Y}_c^{KB}(\cdot)$, i.e., the domination probabilities $p(\mathbf{y})$ in Equation (4.8) are obtained with $\mathbf{Y}_c^{KB}(\cdot)$. Associated to $\mathbf{R}^*$ is the *optimal improvement region*, $\mathcal{I}_{\mathbf{R}^*}$ that will be the focus of the search in the second phase. The same threshold $\varepsilon = 10^{-3}$ as in Equation (4.7) is applied.

The procedure for selecting $\mathbf{R}$ after local convergence is illustrated in Figures 4.22 and 4.23. In this example, the initial DoE is made of $n = 20$ observations, and convergence to the center is detected after 26 added points, leaving $b = 54$ calls to $\mathbf{f}(\cdot)$ in the second phase of the algorithm for a total *budget* of 100 $\mathbf{f}(\cdot)$ evaluations. Figure 4.22 shows the final virtual Pareto fronts obtained for two different reference points, as well as simulated fronts sampled from the final virtual posterior (those fronts are used for measuring the uncertainty). On the left, the area targeted by $\mathbf{R}$ is small, and so is the remaining uncertainty ($U(\mathbf{R}; \mathbf{Y}_c^{KB}(\cdot)) = 3 \times 10^{-6} < 10^{-3}$). On the right, a farther $\mathbf{R}$ leads to a broader approximation front, but to higher uncertainty ($U(\mathbf{R}; \mathbf{Y}_c^{KB}(\cdot)) = 0.0015 > 10^{-3}$). Figure 4.23 represents the final approximation front obtained with the optimal $\mathbf{R}^*$ ($U(\mathbf{R}^*; \mathbf{Y}_{c^*}^{KB}(\cdot)) = 9.4 \times 10^{-4}$) of Equation (4.9) for the b remaining iterations. A complete covering of $\mathcal{P}_{\mathcal{Y}}$ in the targeted area is observed. As the remaining budget after local convergence was important in this example (54 iterations), the Pareto front has been almost entirely unveiled. When less resources remain (e.g. 14 iterations), an $\mathbf{R}^*$ much closer to $\mathcal{P}_{\mathcal{Y}}$ is determined.

4.7 Algorithm implementation and testing

4.7.1 Implementation of the C-EHI algorithm

The concepts and methods defined in Sections 4.3 to 4.6 are put together to make the C-EHI/R-EHI algorithm which stands for Centered or Reference point-based Expected Hypervolume Improvement, depending if a used supplied target $\mathbf{R}$ has been specified. The `R` package `DiceKriging` has been used for building the Gaussian Processes and additional implementations were written in the `R` language. The C-EHI algorithm which was sketched in Figure 4.1 is further detailed in Algorithm 2. The R-EHI algorithm follows exactly the same outline (replace $\widehat{\mathbf{C}}$ by $\widehat{\mathbf{R}}$ and $\widehat{\mathcal{L}}$ by $\widehat{\mathcal{L}}'$). The integral for $U(\widehat{\mathcal{L}})$ is estimated numerically using $N_{\mathcal{L}} = 100$ points regularly distributed along $\widehat{\mathcal{L}}$. $U(\mathbf{R})$ is computed by means of Monte Carlo techniques with $N_{MC} = 10^5$ samples.

```
Inputs: uncertainty limit $\varepsilon$, $budget$;
Data: Create and evaluate an initial DoE of $n$ designs;
Initialize $m$ GPs $Y_j(\cdot)$ for each objective $f_j(\cdot)$, $j = 1, \dots, m$;
$t = n$; $U(\widehat{\mathcal{L}}) = +\infty$;          /* $U(\widehat{\mathcal{L}})$ line uncertainty, Equation (4.7) */
/* First phase: optimization towards the center, see Algorithm 1 */
while $t < budget$ and $U(\widehat{\mathcal{L}}) > \varepsilon$ do
    Estimate the Ideal and Nadir point, $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}} \Rightarrow \widehat{\mathcal{L}}$ and $\widehat{\mathbf{C}}$;   /* see Section 4.4.2 */
    $\mathbf{x}^{(t+1)} = \arg\max_{\mathbf{x} \in X} \mathrm{mEI}(\mathbf{x}; \widehat{\mathbf{C}})$;          /* see Section 4.3 */
    Evaluate $\mathbf{f}(\mathbf{x}^{(t+1)})$ and update the GPs; Compute $U(\widehat{\mathcal{L}})$;   /* see Section 4.5 */
    $t = t + 1$;
end
/* If remaining budget after convergence: second phase */
/* Determine widest accurately attainable area and target it, Section 4.6 */
if $t < budget$ then
    Choose $\mathbf{R}^*$ solution of Equation (4.9);          /* see Section 4.6 */
    $\mathbf{R}^* = \arg\min_{\substack{\mathbf{R} \in \widehat{\mathcal{L}} \\ \text{s.t. } U(\mathbf{R}; \mathbf{Y}^{KB}(\cdot)) < \varepsilon}} \|\mathbf{R} - \widehat{\mathbf{N}}\|$;
end
while $t < budget$ do
    $\mathbf{x}^{(t+1)} = \arg\max_{\mathbf{x} \in X} \mathrm{EHI}(\mathbf{x}; \mathbf{R}^*)$;  /* Target larger improvement region $\mathcal{I}_{\mathbf{R}^*}$ */
    Evaluate $f_j(\mathbf{x}^{(t+1)})$ and update the GPs;
    $t = t + 1$;
end
return final DoE, final GPs, and approximation front $\widehat{\mathcal{P}_{\mathcal{Y}}}$;
```

Algorithm 2: C-EHI (Centered Expected Hypervolume Improvement) Algorithm.

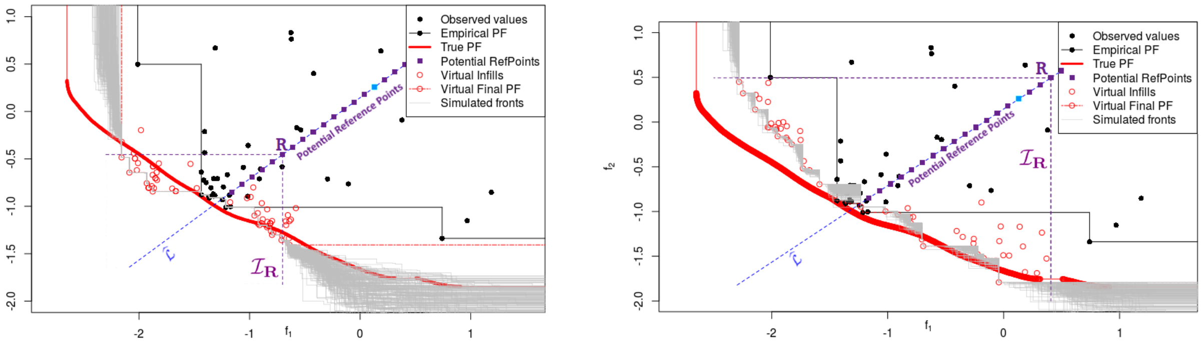


Figure 4.22: Uncertainty quantification through final virtual fronts. The anticipated remaining uncertainty can be visualized as the grey area within $\mathcal{I}_\mathbf{R}$ roamed by the sampled fronts. It is small enough for the $\mathbf{R}$ used on the left and too important for the $\mathbf{R}$ on the right. The blue reference point on $\widehat{\mathcal{L}}$ is $\mathbf{R}^*$, the farthest point that leads to a virtual front with low enough uncertainty.

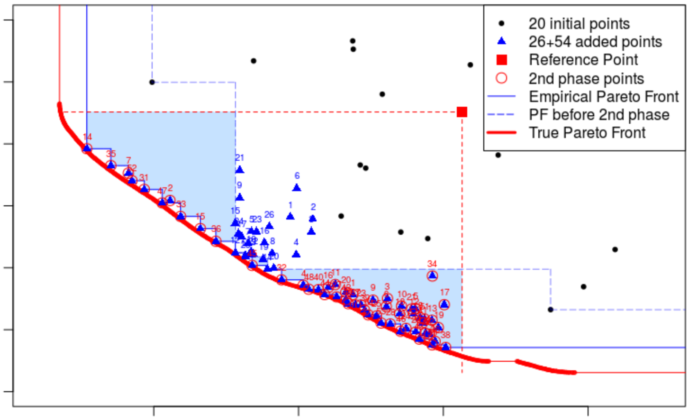
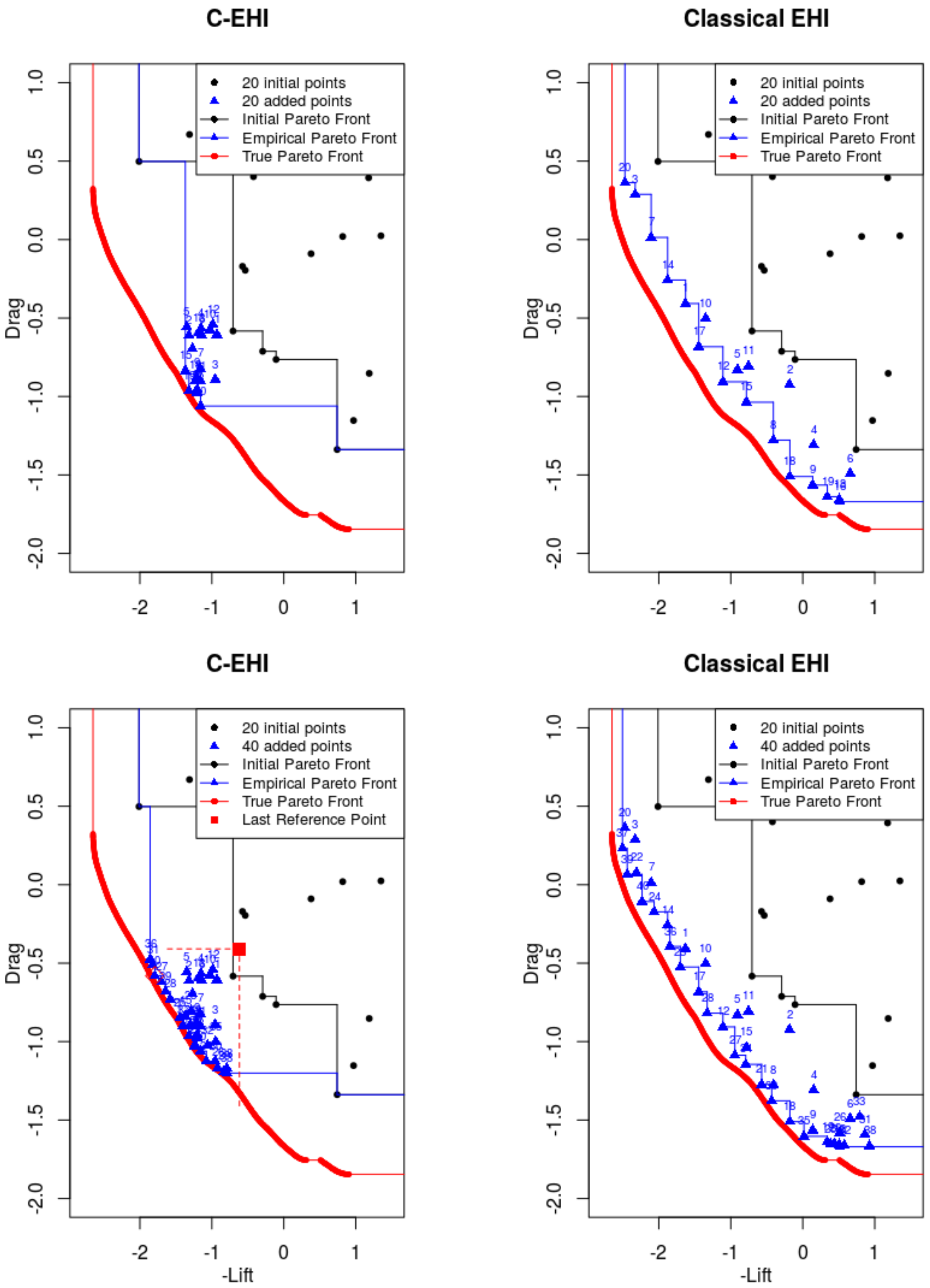


Figure 4.23: Final approximation of the Pareto front by C-EHI with, as a red square, the reference point of the second phase chosen as a solution to problem (4.9), $\mathbf{R} = \mathbf{R}^*$. The objective values added during the second phase of the algorithm are circled in red. Compared to the initial front obtained when searching for the center (other blue triangles), the final approximation front is expanded as highlighted by the blue hypervolume.

Figure 4.24 details a typical run of the C-EHI algorithm when facing a too restricted budget to uncover the entire Pareto front of the MetaNACA for $d = 8$. During the first iterations, the center of the Pareto front is targeted. Once local convergence has been detected, the part of the Pareto front in which convergence can be accurately obtained within the remaining budget is forecasted, and then targeted. The approximation of $\mathcal{P}_\mathcal{Y}$ is enhanced in its central part. The targeting methodology gains in importance as the number of objectives increases because the relative number of Pareto optimal solutions grows and it becomes harder to approximate all of them. An illustrative C-EHI run with $m = 3$ objectives is shown in Figure 4.25.


C-EHI

Classical EHI

C-EHI

Classical EHI

Figure 4.24: Comparison of C-EHI (left) with the standard EHI (right). Top: approximation front after 20 iterations: C-EHI better converges to the center of the Pareto front to the detriment of the front ends. Bottom: approximation front after 40 iterations: after local convergence (at the 26th iteration here), a wider optimal improvement region (under the red square) is targeted for the 14 remaining iterations. Compared to the standard EHI, the Pareto front is sought in a smaller balanced part of the objective space, at the advantage of a better convergence.

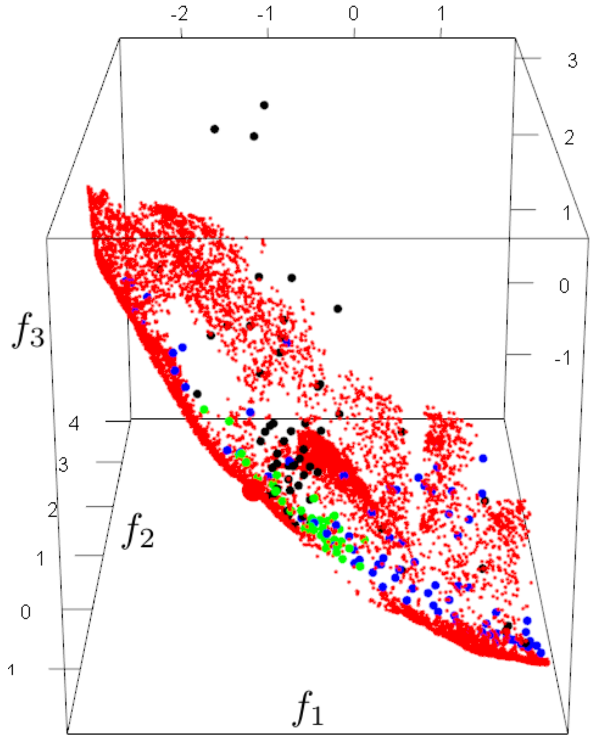


Figure 4.25: Typical C-EHI (green points) and EHI (blue points) runs on the MetaNACA problem with $m = 3$ objectives. Black dots are evaluations of the initial DoE. The true Pareto front (red) is attained at its center by C-EHI while it is approximated globally yet less accurately by EHI.

4.7.2 Test results

In this section, the capabilities of C-EHI to produce a well spread approximation of the Pareto front in the central part of $\mathcal{P}_\mathcal{Y}$ is compared with the state-of-the-art EHI. Experiments are completed on two popular multi-objective problems, and on the MetaNACA benchmark (Chapter 3).

4.7.2.1 Restricted performance metrics

Classical multi-objective indicators (Section 2.3.2) compare fronts which are aimed at finding the whole $\mathcal{P}_\mathcal{Y}$. This is not the case of C-EHI and R-EHI, which consider the unveiling of a smaller but more relevant part of $\mathcal{P}_\mathcal{Y}$ where to enhance convergence, and empirical Pareto fronts with similar shapes to the one shown in blue in Figure 4.20 will be measured as performing poorly as they do not cover the entire front. The metrics introduced in Chapter 2 have therefore to be adapted to assess C-EHI capabilities.

In order to focus on the central part of the Pareto front, the indicators and fronts are restricted to the regions of interest

$$\mathcal{I}_w := \{\mathbf{y} \in Y : \mathbf{y} \preceq \mathbf{R}^w\} \quad \text{where} \quad \mathbf{R}^w := (1-w)\mathbf{C} + w\mathbf{N}. \tag{4.10}$$

To focus on parts of $\mathcal{P}_\mathcal{Y}$ in the vicinity of $\mathbf{C}$, w's ranging between 0.05 and 0.3 will be used. Figures 4.26 and 4.27 show some $\mathcal{I}_w$.

Notice that these truncated indicators will also show if C-EHI was able to recover the real center of the Pareto front: if it was directed towards a wrong (in the sense not central) location of $\mathcal{P}_\mathcal{Y}$ during the first phase, the indicators will exhibit bad results.

If local convergence to $\mathcal{P}_\mathcal{Y}$ has been detected (Section 4.5), the algorithm determines an optimal improvement region $\mathcal{I}_{\mathbf{R}^*}$ (defined in Section 4.6) where new values are sought during the last iterations. Indicator values in this part of the Pareto front are also of interest as this area is targeted within the last iterations.

4.7.2.2 Experiments with analytical test functions

In this section, we investigate how C-EHI converges to the center of the Pareto front and compare it with two state-of-the-art algorithms: a Bayesian optimizer with the EHI infill criterion (Emmerich et al., 2011) and the Evolutionary Algorithm NSGA-II (Deb et al., 2002). As discussed in Section 4.2.1, EHI is defined up to a reference point which is instrumental in selecting the part of the objective space $\mathcal{I}_\mathbf{R}$ where $\mathcal{P}_\mathcal{Y}$ is sought. To target the entire $\mathcal{P}_\mathcal{Y}$ with EHI, $\mathbf{R}$ should be placed at the Nadir point of the true Pareto front. Since $\mathcal{P}_\mathcal{Y}$ is unknown, it is suggested (Feliot, 2017; Ishibuchi et al., 2018) to take a conservative empirical Nadir point, $\overline{\mathbf{N}} + r(\overline{\mathbf{N}} - \overline{\mathbf{I}})$ with $r = 0.1$, where $\overline{\mathbf{I}}$ and $\overline{\mathbf{N}}$ are the empirical Ideal and Nadir points. This is the default choice for setting $\mathbf{R}$ in `GPareto` (Binois and Picheny, 2015).

This EHI implementation depends on $\widehat{\mathcal{P}_\mathcal{Y}}$ through $\overline{\mathbf{I}}$ and $\overline{\mathbf{N}}$. We therefore consider three additional EHI variants. In the idealized EHI$_{\mathcal{P}_\mathcal{Y}}$, the reference point is $\mathbf{R} = \mathbf{N}$, the true Nadir point. In this variant, $\mathcal{I}_\mathbf{R} = \mathcal{I}_{\mathcal{P}_\mathcal{Y}}$: the considered improvement area is the right one. EHI$_{\mathcal{P}_\mathcal{Y}}$ corresponds to an utopian setting where it would be known in advance where to look for the Pareto front in the objective space. Its interest is that it provides an upper bound on the expected performance of EHI.

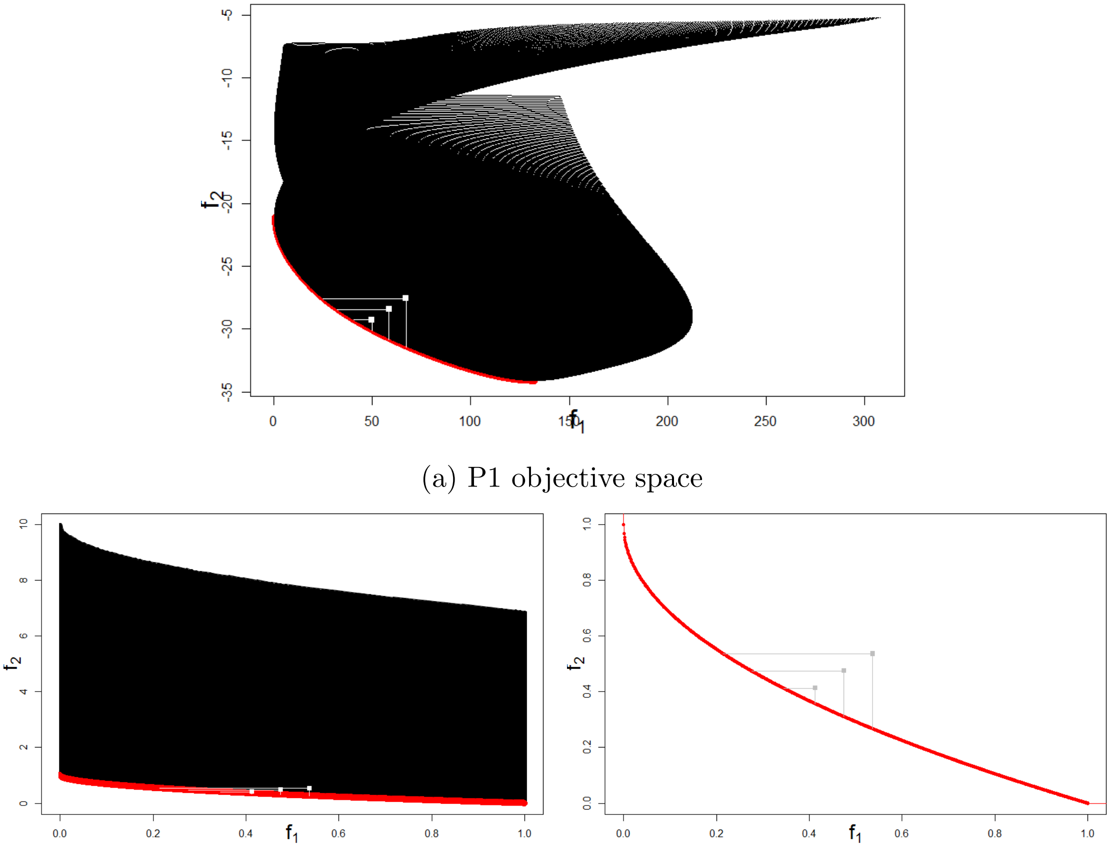
The third variant, EHI$_\mathrm{N}$, has $\mathbf{R}$ defined as the estimated Nadir point of the Pareto front, $\widehat{\mathbf{N}}$ using the techniques of Section 4.4.2.3. EHI$_\mathrm{N}$ is a new version of the EHI algorithm: instead of defining $\mathbf{R}$ relying on observed data such as the empirical front or extreme observations, $\mathbf{R}$ is set up according to the metamodels.

Last, we consider the EHI$_\mathrm{M}$ variant in which the reference point is $\mathbf{R} = \overline{\mathbf{M}}$ (maximal value observed). Contrarily to EHI$_\mathrm{N}$, the maximum is taken over all the points instead of over those in $\widehat{\mathcal{P}_\mathcal{Y}}$. Such a reference point will often have large components. If it covers all of the objective space, it may over-emphasize the extreme parts of the Pareto front.

The algorithms are benchmarked with two popular analytical test functions for multi-objective optimization. The first one is the P1 problem of Parr (2013), which has $d = 2$ dimensions and $m = 2$ objectives. It is initialized with a design of experiments of size $n = 8$ and run for $p = 12$ iterations. The second test problem is ZDT1 (Zitzler et al.,

2000) in $d = 4$ dimensions and $m = 2$ objectives, initialized with a design of experiments of size $n = 20$ and run for $p = 40$ additional iterations. In the case of ZDT1, a popular reference point for EHI being $(11, 11)^\top$, the $\text{EHI}_{(11,11)}$ experiment considers this setting.

Two comparison metrics are considered. The first one is the hypervolume indicator restricted to $\mathcal{I}_w$ for $w = 0.05, 0.15, 0.25$ to evaluate convergence and diversity in the central parts of the Pareto front. Figure 4.26 shows these improvement regions for both benchmark problems. I_H is normalized by the hypervolume of $\mathcal{P}_\mathcal{Y}$ in $\mathcal{I}_w$, such that the indicator is upper bounded by 1. The second performance metric is the attainment time which assesses the number of function evaluations (including the n initial designs) it takes to a method for entering the improvement region irrespectively of the final hypervolume covered.



(a) P1 objective space

(b) ZDT1 ($d = 4$) objective space

(c) Zoom on the ZDT1 ($d = 4$) Pareto front

Figure 4.26: Pareto fronts (red) and objective spaces (black) of the P1 problem (top) and of the ZDT1, $d = 4$, problem (bottom, zoom on $\mathcal{P}_\mathcal{Y}$ on the right) with the $\mathcal{I}_w$ areas to which the performance metrics are restricted. These correspond to a central part of the Pareto front.

Runs are repeated 10 times starting from different initial space-filling designs. The metrics means and standard deviations are reported in Tables 4.5 and 4.6. They are computed for C-EHI, the four EHI variants, and NSGA-II. The population size of NSGA-II is set to 12 and 20 for P1 and ZDT1, respectively. The performance of NSGA-II is recorded at the smallest number of generations such that the number of functions evaluations is larger or equal to that of the Bayesian algorithms. This number of generations is 2 and 3 for P1 and ZDT1 and the metrics are on the NSGA-II$_\flat$ row in Tables 4.5 and 4.6. For comparison purposes, NSGA-II runs are continued until 120 and 800 functions evaluations are reached for the P1 and ZDT1 functions. The final metrics are given in both tables on the NSGA-II$_+$ row.

	Hypervolume			Attainment time		
w	0.05	0.15	0.25	0.05	0.15	0.25
C-EHI	0.185 (0.233)	0.549 (0.263)	0.668 (0.185)	21.6 [7]	13.1 (2.7)	9.5 (1)
EHI	0.155 (0.218)	0.465 (0.179)	0.611 (0.114)	39.4 [4]	13.2 (2.6)	11.4 (2.6)
EHI$_{\mathcal{P}_\mathcal{Y}}$	0.269 (0.260)	0.446 (0.175)	0.636 (0.136)	30.0 [6]	14 (3.2)	11 (2.6)
EHI$_\mathrm{N}$	0.130 (0.158)	0.312 (0.223)	0.460 (0.192)	32.4 [5]	16.7 [9]	11.5 (3.5)
EHI$_\mathrm{M}$	0.012 (0.039)	0.202 (0.181)	0.389 (0.136)	180 [1]	22.7 [7]	12.6 (4.1)
NSGA-II$_\flat$	0	0.052 (0.110)	0.107 (0.183)	×	80 [2]	51.1 [3]
NSGA-II$_+$	0.188 (0.219)	0.576 (0.109)	0.705 (0.069)	169.6 [5]	50.4 (31.1)	41.3 (31.9)

Table 4.5: Hypervolume and attainment time averaged over 10 runs (standard deviation in brackets), for different central parts of the Pareto front on the P1 problem. When at least one run did not attain $\mathbf{R}^w$, red figures correspond to empirical runtimes with the number of successful runs in brackets. × indicates that no run was able to attain $\mathbf{R}^w$ in the given budget.

	Hypervolume			Attainment time		
w	0.05	0.15	0.25	0.05	0.15	0.25
C-EHI	0.703 (0.049)	0.895 (0.010)	0.936 (0.006)	26.8 (6.6)	23.4 (2.2)	23.4 (2.2)
EHI	0.065 (0.154)	0.097 (0.204)	0.101 (0.213)	145 [2]	145 [2]	145 [2]
EHI$_{\mathcal{P}_\mathcal{Y}}$	0.611 (0.066)	0.848 (0.029)	0.901 (0.023)	28.7 (2.8)	22.8 (2.3)	21.4 (0.5)
EHI$_\mathrm{N}$	0.362 (0.349)	0.650 (0.246)	0.740 (0.206)	48.1 [6]	22.2 (0.4)	22.2 (0.4)
EHI$_\mathrm{M}$	0.575 (0.107)	0.845 (0.038)	0.906 (0.022)	24.4 (5.6)	22.2 (0.6)	22.1 (0.3)
EHI$_{(11,11)}$	0.133 (0.13)	0.327 (0.251)	0.472 (0.218)	120 [2]	120 [2]	59.2 [5]
NSGA-II$_\flat$	0	0	0	×	×	×
NSGA-II$_+$	0.375 (0.161)	0.749 (0.075)	0.842 (0.052)	532.9 (143.4)	331.9 (121)	219.2 (101.5)

Table 4.6: Hypervolume and attainment time averaged over 10 runs (standard deviation in brackets), for different central parts of the Pareto front on the ZDT1 problem. When at least one run did not attain $\mathbf{R}^w$, red figures correspond to empirical runtimes with the number of successful runs in brackets. × indicates that no run was able to attain $\mathbf{R}^w$ in the given budget.

Before analyzing the results in more details, let us state the main conclusions of Tables

4.5 and 4.6. On both test problems, C-EHI consistently outperforms all other EHI variants in terms of hypervolume and time to reach the central parts of $\mathcal{P}_{\mathcal{Y}}$. The performances of the different optimizers depend on the test function and further explanations are given in the following. At the considered limited budget, the evolutionary algorithm NSGA-II gives a weaker approximation of the Pareto front central regions than the Bayesian methods, as measured by both the hypervolumes and the attainment times.

P1 problem

The statistics of the hypervolumes reported in Table 4.5 indicate that C-EHI better converges to the central part of the Pareto front than the other EHI algorithms. The helped $\text{EHI}_{\mathcal{P}_{\mathcal{Y}}}$ outperforms C-EHI only when $w = 0.05$. This is due to the fact that this benchmark contains a local Pareto front (which can be seen on Figure 4.26 for small f_1 values and $f_2 \approx -17$), which lightly deteriorates the Ideal and the Nadir point estimation, hence the estimation of the Center. The error in $\widehat{\mathbf{C}}$ leads to a slightly off-centered convergence which is highlighted by the fact that 3 C-EHI runs out of 10 did not attain this narrow part of $\mathcal{P}_{\mathcal{Y}}$. Some difficulties in estimating $\mathbf{N}$ through GPs simulations are visible in the moderate performance of EHI_{N} relatively to the standard EHI approach (where $\mathbf{R}$ is defined according to the empirical front). Yet, as stated in Proposition 4.4, the error in Nadir estimation barely affects C-EHI, but impacts EHI_{N} more significantly. Regarding EHI variants, EHI_{M} performs poorly when compared to the standard EHI and $\text{EHI}_{\mathcal{P}_{\mathcal{Y}}}$ because of the distant reference point which targets an unnecessarily large part of the objective space. At the same number of function evaluations (20), C-EHI clearly outperforms NSGA-II which needs approximately 6 times more function evaluations to achieve the same performance.

The attainment times recorded in Table 4.5 for the P1 problem confirm that the center-targeting C-EHI reaches the central regions faster than the other methods. The thinnest area of interest ($w = 0.05$) is attained more consistently (reached 7 times out of 10 against 6 times by $\text{EHI}_{\mathcal{P}_{\mathcal{Y}}}$, 5 times by EHI_{N}, 4 times by EHI and 1 time by EHI_{M}). Because of its distant $\mathbf{R}$, EHI_{M} is the Bayesian method which needs the most function evaluations to find $\mathcal{I}_w$. The evolutionary NSGA-II is not able to attain $\mathcal{I}_{0.05}$ within 24 function evaluations, only 2 runs out of 10 attain $\mathcal{I}_{0.15}$ and 3 out of 10 attain $\mathcal{I}_{0.25}$.

ZDT1 problem

As shown at the bottom of Figure 4.26, the ZDT1 problem has a wide f_2 range. In dimension $d = 4$, it is difficult to find f_2 values in $\mathcal{P}_{\mathcal{Y}}$'s range: only 0.8% of X leads to $f_2 \leq 1$. On the contrary, all f_1 values are in $\mathcal{P}_{\mathcal{Y}}$'s range. Therefore, the definition of the part of the objective space where to seek $\mathcal{P}_{\mathcal{Y}}$ through $\mathbf{R}$ is critical.

C-EHI correctly identifies the center of $\mathcal{P}_{\mathcal{Y}}$ and drives the optimization towards it, as evidenced by the larger hypervolumes of C-EHI in Table 4.6 for all w's. C-EHI has the best but one attainment time of $\mathcal{I}_{0.05}$ with 26.8 evaluations on the average. EHI_{M} solely attains $\mathcal{I}_{0.05}$ in fewer function evaluations. It is worth mentioning that only 5×10^{-6}%

of the design space has an image in $\mathcal{I}_{0.05}$, highlighting the performance of C-EHI (and EHI_M for the occasion). The number of function evaluations to reach $\mathcal{I}_{0.15}$ and $\mathcal{I}_{0.25}$ is slightly larger for C-EHI than for the other EHI's. This is due to the fact that the first mEI iterations of the C-EHI algorithm sometimes target parts of $\mathcal{P_Y}$ that are not exactly at the center, because of ZDT1's objective space shape. Nonetheless, C-EHI corrects this initial inaccuracy and, at the end of the second phase, a better convergence is achieved as confirmed by the hypervolume. Even though it is equipped with the correct $\mathbf{R}$, $\text{EHI}_{\mathcal{P_Y}}$ does not exhibit results as good as C-EHI, except the attainment time of the wider central parts ($\mathcal{I}_{0.15}$ and $\mathcal{I}_{0.25}$).

The EHI in which $\mathbf{R}$ is computed through the empirical Ideal and Nadir points performs poorly. Only two runs touch the central parts of $\mathcal{P_Y}$. Because the Pareto front of ZDT1 has a small f_2 range and a large f_1 range, the initial errors in $\mathbf{R}$ cut large f_1 values out of the improvement region. Graphically, the search seems directed towards the left-hand-side of the Pareto front. EHI_N is outperformed by C-EHI and $\text{EHI}_{\mathcal{P_Y}}$, but achieves a much better convergence than EHI. This shows the benefits of estimating the location of the Nadir point through GP simulations instead of picking the empirical Nadir for $\mathbf{R}$ in problems such as ZDT1, if the whole Pareto front is sought. Even though EHI_M does not work well on general functions because of a too large targeted part in the objective space $\mathcal{I}_\mathbf{R}$, it yields good results here both in terms of hypervolume and attainment time. Indeed, EHI_M avoids the pitfalls of ZDT1 that were just mentioned, i.e., it does not remove large f_1 values from the improvement region. At the same number of function evaluations (60, row NSGA-II_b), NSGA-II is never able to find any $\mathcal{I}_w$. Even when 800 designs (row NSGA-II_+) are evaluated, the hypervolume in these central areas is much smaller than that of C-EHI.

4.7.2.3 Experiments on the MetaNACA test bed

For the sake of brevity, only experiments on the instance with $d = 8$ parameters are reported here, with $m = 2, 3$ or 4 objectives, because the results on the MetaNACA with 3 or 22 dimensions led to the same conclusions. One run of the C-EHI algorithm on the MetaNACA in dimension $d = 22$, $m = 2$ objectives can be found at the end of this section, in Figure 4.29. The $\mathcal{I}_w$ areas to which the metrics are restricted are shown in Figure 4.27 for $m = 2$.

The Tables 4.7 and 4.8 below contain the hypervolume indicator and the IGD (Definition 2.13) to the true Pareto front for the 2, 3 and 4 objective MetaNACA test cases. They are computed in $\mathcal{I}_{0.1}$, $\mathcal{I}_{0.2}$ and $\mathcal{I}_{0.3}$, and averaged over 10 runs. Standard deviations are indicated in parentheses. The last column averages the indicator values restricted to $\mathcal{I}_{\mathbf{R}^*}$ (the optimal reference point of Equation 4.9) over the runs that reached the second phase. × indicates that no run has reached the second phase for the considered *budget*. Similarly to the attainment times in the previous section, red figures correspond to extrapolated indicators: when for at least one run, no solution was found in $\mathcal{I}_w$, the IGD is averaged over the runs which entered $\mathcal{I}_w$ and divided by the proportion of successful runs. Brackets indicate the number of successful runs. The indicator values of the C-

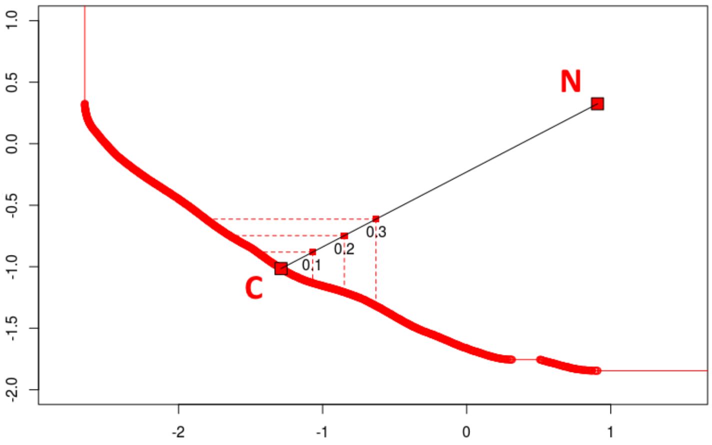


Figure 4.27: Central parts of the Pareto front of the MetaNACA ($d = 8$, $m = 2$) for which the indicators are computed. The $\mathcal{I}_{0.1}$ area only considers extremely close and central solutions.

EHI algorithm are compared to those obtained with the standard EHI implementation of the `R` package `GPareto` (right column) in which the default reference point is taken at $1.1\overline{\mathbf{N}} - 0.1\overline{\mathbf{I}}$. Dealing with parsimonious calls to the objective functions, four tight optimization budgets are considered: 40, 60, 80 and 100 calls to $\mathbf{f}(\cdot)$. The $n = 20$ first calls to $\mathbf{f}(\cdot)$ are devoted to the initialization of the GPs using an LHS space-filling design (Stein, 1987), and the experiments are repeated 10 times starting from different initial designs.

Figure 4.28 shows how the hypervolume indicator evolves with optimization iterations. The indicators are of course increasing with the iterations, and the C-EHI consistently outperforms the general EHI in finding points in the central part of the Pareto front for 2 and 3 objectives. For 4 objectives an important number of points obtained by both algorithms belongs to $\mathcal{I}_{0.2}$ and $\mathcal{I}_{0.3}$. While significantly more values (and Pareto-optimal values) are obtained by C-EHI in $\mathcal{I}_{0.2}$ and $\mathcal{I}_{0.3}$, EHI may episodically and non-significantly yield a larger hypervolume.

A few words of caution are needed to read the Tables 4.7 to 4.8. As the width of the Pareto front that is targeted in the second phase depends on the remaining budget, runs of the C-EHI algorithm with different *budget*s are not directly comparable. For instance, if convergence is detected after 35 iterations, the reference point that defines the targeted area for the last calculations $\mathbf{R}^*$ will be different if 5 or 45 iterations remain. The first case will concentrate on a very central part of the Pareto front, whereas the second will target a broader area. As a consequence, some numbers may express better performance in thinner portions of the Pareto front in spite of a smaller total budget, which is only due to the fact that they have explicitly targeted a smaller part of the solutions.

The average performance measures reported in Tables 4.7 to 4.8 confirm the behavior of the C-EHI algorithm already illustrated in Figure 4.24 for a typical run: mEI set to improve on the estimated center efficiently drives the algorithm towards the (unknown)

m	*budget*	$\mathbf{R}^{0.1}$		$\mathbf{R}^{0.2}$		$\mathbf{R}^{0.3}$		$\mathbf{R}^{*}$	
		C-EHI	EHI	C-EHI	EHI	C-EHI	EHI	C-EHI	EHI
2	40	0.275 (0.18)	0.025 (0.04)	0.498 (0.17)	0.227 (0.15)	0.581 (0.10)	0.386 (0.19)	0.664	0.253
	60	0.377 (0.19)	0.096 (0.12)	0.651 (0.11)	0.342 (0.14)	0.719 (0.09)	0.525 (0.12)	0.768 (0.13)	0.418 (0.24)
	80	0.548 (0.10)	0.118 (0.11)	0.759 (0.05)	0.398 (0.12)	0.821 (0.03)	0.572 (0.11)	0.881 (0.04)	0.606 (0.22)
	100	0.524 (0.14)	0.153 (0.16)	0.744 (0.08)	0.503 (0.13)	0.831 (0.05)	0.658 (0.08)	0.919 (0.02)	0.805 (0.08)
3	40	0.013 (0.02)	0 (0)	0.181 (0.09)	0.086 (0.05)	0.319 (0.05)	0.237 (0.07)	×	×
	60	0.058 (0.06)	0.010 (0.02)	0.267 (0.08)	0.136 (0.06)	0.394 (0.05)	0.305 (0.04)	0.286 (0.03)	0.021 (0.03)
	80	0.109 (0.08)	0.012 (0.02)	0.327 (0.14)	0.170 (0.10)	0.447 (0.17)	0.321 (0.13)	0.476 (0.08)	0.161 (0.11)
	100	0.160 (0.09)	0.016 (0.02)	0.412 (0.07)	0.218 (0.06)	0.546 (0.04)	0.391 (0.06)	0.584 (0.05)	0.224 (0.09)
4	40	0.113 (0.11)	0.075 (0.10)	0.291 (0.09)	0.240 (0.10)	0.374 (0.06)	0.378 (0.09)	×	×
	60	0.187 (0.15)	0.138 (0.09)	0.356 (0.08)	0.340 (0.09)	0.418 (0.05)	0.473 (0.07)	0.533	0.238
	80	0.312 (0.16)	0.198 (0.08)	0.470 (0.09)	0.413 (0.07)	0.516 (0.09)	0.533 (0.06)	0.617 (0.08)	0.338 (0.07)
	100	0.519 (0.08)	0.219 (0.07)	0.612 (0.11)	0.464 (0.07)	0.642 (0.12)	0.580 (0.06)	0.729 (0.05)	0.453 (0.04)

Table 4.7: Hypervolume indicator averaged over 10 runs for different central parts of the Pareto front, budgets and number of objectives. The true Pareto front has an hypervolume indicator of 1.

m	*budget*	$\mathbf{R}^{0.1}$		$\mathbf{R}^{0.2}$		$\mathbf{R}^{0.3}$		$\mathbf{R}^{*}$	
		C-EHI	EHI	C-EHI	EHI	C-EHI	EHI	C-EHI	EHI
2	40	0.130 [9]	0.391 [5]	0.176 (0.09)	0.246 [9]	0.228 (0.05)	0.293 (0.20)	0.069	0.175
	60	0.095 (0.05)	0.242 [7]	0.109 (0.05)	0.204 (0.08)	0.133 (0.06)	0.184 (0.06)	0.066 (0.02)	0.101 [9]
	80	0.059 (0.02)	0.203 [8]	0.058 (0.01)	0.171 (0.05)	0.067 (0.02)	0.161 (0.07)	0.050 (0.01)	0.149 (0.05)
	100	0.067 (0.02)	0.177 [8]	0.059 (0.02)	0.138 (0.05)	0.055 (0.02)	0.118 (0.03)	0.048 (0.02)	0.109 (0.03)
3	40	0.736 [5]	4.267 [1]	0.455 (0.13)	0.518 (0.13)	0.531 (0.12)	0.500 (0.10)	×	×
	60	0.390 [8]	0.961 [4]	0.388 (0.11)	0.460 (0.11)	0.471 (0.13)	0.439 (0.06)	0.196 (0.03)	0.287 [8]
	80	0.238 (0.10)	0.550 [5]	0.256 (0.12)	0.361 (0.17)	0.339 (0.14)	0.356 (0.14)	0.181 (0.05)	0.241 [9]
	100	0.226 (0.05)	0.510 [6]	0.250 (0.05)	0.349 (0.06)	0.335 (0.08)	0.351 (0.07)	0.183 (0.05)	0.349 (0.08)
4	40	0.345 [9]	0.624 [6]	0.381 (0.05)	0.447 (0.12)	0.626 (0.07)	0.571 (0.07)	×	×
	60	0.280 (0.13)	0.374 [8]	0.334 (0.04)	0.359 (0.06)	0.587 (0.07)	0.512 (0.07)	0.197	0.233
	80	0.210 (0.06)	0.282 (0.06)	0.285 (0.05)	0.298 (0.04)	0.523 (0.08)	0.460 (0.06)	0.212 (0.04)	0.262 (0.08)
	100	0.158 (0.02)	0.266 (0.06)	0.236 (0.05)	0.277 (0.03)	0.468 (0.08)	0.430 (0.05)	0.257 (0.04)	0.291 (0.08)

Table 4.8: Inverted Generational Distance averaged over 10 runs for different central parts of the Pareto front, budgets and number of objectives. Lower values are better.

central part of the real Pareto front. Table 4.7 summarizes test results expressed in terms of hypervolume improvements. In the most central part of the front ($w = 0.1$) C-EHI significantly surpasses the standard EHI. It is also remarkable that despite early GPs inaccuracies, the algorithm does not drift towards off-centered locations of the front. EHI outperforms C-EHI only with 4 objectives and $w = 0.3$, since in this case $\mathcal{I}_w$ is not a restrictive central part in such dimension.

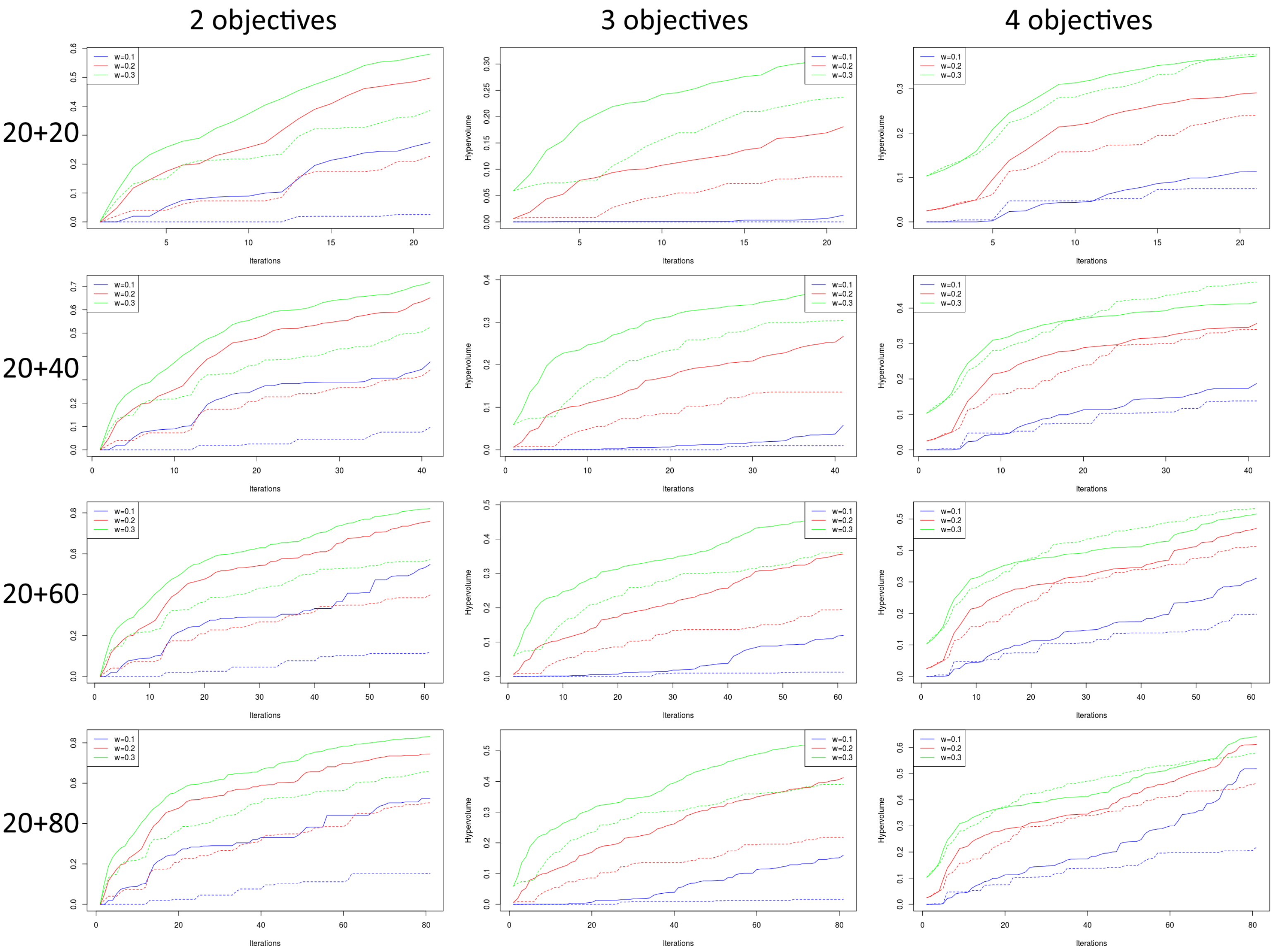


Figure 4.28: Mean hypervolume indicator for 2, 3 or 4 objectives (as columns) and *budget*s of 40, 60, 80, 100 (as rows). The blue, red and green colors correspond to the improvement regions $\mathcal{I}_{0.1}$, $\mathcal{I}_{0.2}$ and $\mathcal{I}_{0.3}$, respectively. Dashed lines correspond to the standard EHI, continuous lines to the C-EHI algorithm.

The IGD (Table 4.8) shows similar results. Notice that for at least one run, the classical EHI does not reach the $\mathcal{I}_{0.1}$ area in the two and three objective cases, even if 100 evaluations are allowed. In the 4 dimensional case, at least 80 iterations are needed. Again, the results show smaller distances between points in $\mathcal{P}_{\mathcal{Y}} \cap \mathcal{I}_w$ and $\widehat{\mathcal{P}_{\mathcal{Y}}}$ with C-EHI for 2 objectives, and when the restriction area is small. For 4 objectives and $w = 0.3$, EHI outperforms C-EHI, but in this case $\mathcal{I}_{0.3}$ is a quite large part of Y. Many solutions in $\mathcal{P}_{\mathcal{Y}} \cap \mathcal{I}_{0.3}$ are thus far away from the area where C-EHI converges.

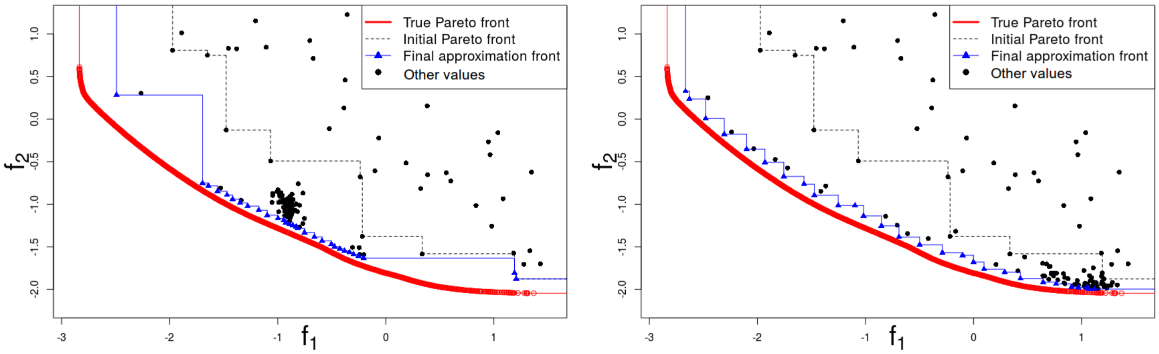


Figure 4.29: Comparison between C-EHI (left) and EHI (right) for one run of the MetaNACA problem in $d = 22$ dimensions. 150 calls to $\mathbf{f}(\cdot)$ were allowed and 50 of them were devoted to the initial DoE. Again, C-EHI improves the Pareto front at its center, EHI tries to uncover the whole front at the cost of a lower accuracy.

Other indicators such as attainment times or the ε-Indicator confirm the results reported above, but are not given here for reasons of conciseness.

The same conclusions are obtained with R-EHI which relies on the same mechanisms. No statistically significant results are shown here for the sake of brevity, but a typical run of R-EHI on the MetaNACA ($d = 8$, $m = 2$) where $\mathbf{R} = (-2.1, -0.2)^\top$ is shown in Figure 4.30.

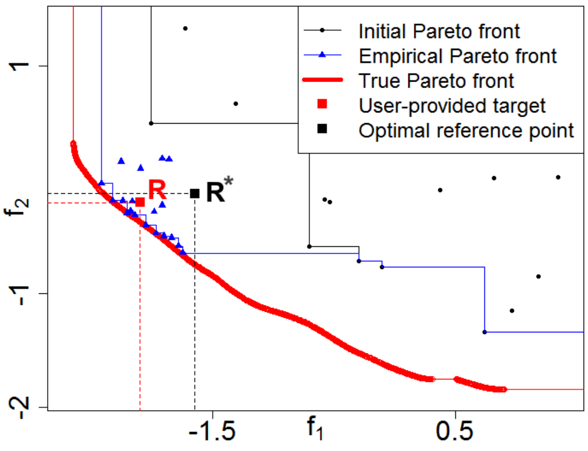


Figure 4.30: Typical run of the R-EHI algorithm with a budget of 20+20 function evaluations, when the target $\mathbf{R}$ (red square) is provided. Remark that after convergence detection, the eventually chosen $\mathbf{R}^*$ (black square, see Section 4.6) targets a wider part of $\mathcal{P}_\mathcal{Y}$ than the initially supplied $\mathbf{R}$ because accurate enough convergence has been forecasted for the remaining iterations inside $\mathcal{I}_{\mathbf{R}^*}$.

4.8 Conclusions

In this chapter, we have developed new concepts and have adapted existing Bayesian multi-objective optimization methods to enhance convergence to preferred solutions of a multi-objective optimization problem at severely restricted number of calls to the objective functions. A general definition of the Pareto front center, valid for non-convex, discontinuous, convoluted fronts has been given and some of its properties analyzed. In case no target has been expressed, the latter is implicitly preferred over other solutions. We have proposed the C-EHI optimization algorithm which first estimates the Pareto front center, then maximizes the mEI criterion and finally chooses a targeted central part of the Pareto front in accordance with the remaining budget. The R-EHI algorithm operates in the same logic, except that the part of the Pareto front to unveil first is user-dictated. Both algorithms aim at first converging towards a preferred part of the Pareto front before widening the approximation front taking the remaining resources into account. They have shown faster and better convergence to the critical part of the Pareto front than other state-of-the-art approaches.

Chapter 5

Extensions of the C-EHI/R-EHI algorithm

Contents

This chapter deals with extensions of the previously described C-EHI/R-EHI algorithm (Chapter 4). The first part is devoted to multi-point extensions such that the criteria return a batch of designs where to evaluate the objective functions at each iteration, instead of one single design. This is particularly attractive if the simulator can be run in parallel on q different computers or nodes of a cluster since the number of evaluated designs within the same wall-clock time can be multiplied by a factor q. In Section 5.1.1, we propose and study a multi-point extension to the mEI criterion, named q-mEI. We also explain why a tempting alternative criterion to q-mEI is inappropriate for optimization. Numerical tests comparing the sequential and batch versions of the mEI algorithm show a better convergence towards the preferred part of $\mathcal{P}_{\mathcal{Y}}$ at the same number of iterations. In Section 5.1.2, q-EHI, the multi-point extension to the EHI criterion used during the second phase of our algorithm is proposed and analyzed, as well as cheaper multi-point proxys to this figure of merit.

Section 5.2 deals with constraints the designs have to comply with. The steps of the C-EHI/R-EHI algorithm are adapted to account for the necessity of satisfying the constraints by modifying the Pareto-dominance relation. The incorporation of constraints within the mEI and EHI criteria is discussed, and the use of mEI in the case of severely constrained problems (i.e., where it is even hard to find one design which satisfies all constraints) analyzed.

Last, in Section 5.3, implementation details of C-EHI/R-EHI as well as possible ways to improve and extend the algorithm are proposed.

5.1 Batch criteria in Bayesian Optimization

In the context of costly objective functions, the temporal efficiency of Bayesian optimization algorithms can be improved by evaluating the functions in parallel on different computers (or on different cluster nodes). A batch version of these Bayesian algorithms directly stems from replacing the infill criteria with their multi-point pendants: if q points are produced by the maximization of the infill criterion, the $\mathbf{f}(\cdot)$'s can then be calculated in parallel. In some cases, there is a side benefit to the multi-point criterion in that it makes the algorithms more robust to inadequacies between the GPs and the true functions by spreading the points at each iteration while still complying with the infill criteria logic.

In a mono-objective setting, the multi-point Expected Improvement (q-EI) introduced in Schonlau (1997) searches an optimal batch of q points, instead of looking for only one. In Ginsbourger et al. (2010) it is defined as

$$\text{q-EI}(\{\mathbf{x}^{(t+1)},\dots,\mathbf{x}^{(t+q)}\}) = \mathbb{E}[\max_{i=1,\dots,q}(f_{min} - Y(\mathbf{x}^{(t+i)}))_+] = \mathbb{E}[(f_{min} - \min_{i=1,\dots,q} Y(\mathbf{x}^{(t+i)}))_+]. \tag{5.1}$$

$\{\mathbf{x}^{(t+1)*},\dots,\mathbf{x}^{(t+q)*}\}$ maximizing (5.1) are q promising points to evaluate simultaneously. It is clear from the q-EI criterion that the price to pay for multi-point infill criteria is an increase in the dimension of the inner optimization loop that creates the next iterates. In Algorithm 1, the next iterate $\mathbf{x}^{(t+1)}$ results from an optimization in d dimensions, while in a q-points algorithm there are $d \times q$ unknowns.

The multi-point Expected Improvement has received some attention recently, see for instance Frazier and Clark (2012); Ginsbourger et al. (2011); Ginsbourger and Le Riche (2010); Janusevskis et al. (2011, 2012), where the criterion is computed using Monte Carlo simulations. It has been calculated in closed form for $q = 2$ in Ginsbourger et al. (2010) and extended for any q in Chevalier and Ginsbourger (2013). An expression and a proxy for its gradient have then been calculated for efficiently maximizing it in X^q (Marmin et al., 2015, 2016).

To the best of our knowledge, there exists no Bayesian multi-point multi-objective infill criterion. In the same spirit, we wish to extend the multi-objective mEI and EHI criteria (Binois, 2015; Emmerich et al., 2006) employed in Chapter 4 so that they return q designs to evaluate in parallel. In Horn et al. (2015), several techniques have been proposed to obtain a batch of q different locations where the functions of multi-objective problems can be evaluated in parallel. Common practices consist in parallelizing certain steps of the algorithm. In ParEGO (Knowles, 2006) for instance, since randomly chosen coefficients define a mono-objective problem to be optimized, a straightforward step towards batch optimization is to consider q such problems simultaneously. Similarly, the decomposition framework of MOEA/D (Zhang and Li, 2007) was exploited for batch Bayesian multi-objective optimization (Zhang et al., 2009). Multi-point multi-objective infill criteria either rely on the simultaneous execution of multi-objective searches with q different goals (e.g., Deb and Sundar, 2006), on multi-objective infill criteria, e.g. a multi-objective maximization of the EI (Jeong and Obayashi, 2005; Ribaud, 2018), on the parallel evaluation of q points located on an estimation of the Pareto front (Namura et al., 2017a), or finally on q sequential steps of a multi-objective Kriging Believer strategy (Feliot, 2017; Ginsbourger et al., 2010). A problem involving q-EI's with two objectives is presented in the Chapter 3 of Ribaud (2018) but the formulation is likely to have the same flaws as the mq-EI below, i.e., each point can optimize only a criterion. In the current work, we investigate Bayesian multi-objective criteria whose maximization yields q points. The resulting strategy is therefore optimal with respect to the criterion.

5.1.1 Batch targeting in multi-objective optimization: the q-mEI criterion

5.1.1.1 A naive and a correct batch version of the mEI

mEI being a product of EI's, a first approach to extend the mEI criterion to a batch of q points is to use the product of single-objective q-EI's (called mq-EI for "multiplicative q-EI") using R_j instead of $\min\limits_{i=1,\dots,t} f_j(\mathbf{x}^{(i)})$ in (5.1):

$$\begin{aligned}\text{mq-EI}(\{\mathbf{x}^{(t+1)},\dots,\mathbf{x}^{(t+q)}\};\mathbf{R}) &= \prod_{j=1}^{m}\text{q-EI}_j(\{\mathbf{x}^{(t+1)},\dots,\mathbf{x}^{(t+q)}\};R_j)\\ &= \prod_{j=1}^{m}\mathbb{E}[\max_{i=1,\dots,q}(R_j - Y_j(\mathbf{x}^{(t+i)}))_+] = \mathbb{E}[\prod_{j=1}^{m}\max_{i=1,\dots,q}(R_j - Y_j(\mathbf{x}^{(t+i)}))_+] \qquad (5.2)\end{aligned}$$

because the $Y_j(\cdot)$'s are assumed independent. This criterion has however the drawback of not using a product of joint improvement in all objectives, as the max among the q points is taken independently for each objective j considered. This may lead to undesirable behaviors: the batch of q optimal points using this criterion may be composed of optimal points w.r.t. each individual EI_j. For example with $m = 2$ and $q = 2$, a batch $\{\mathbf{x}^{(1)*}, \mathbf{x}^{(2)*}\}$ with promising $Y_1(\mathbf{x}^{(1)*})$ and $Y_2(\mathbf{x}^{(2)*})$ may be optimal, without taking $Y_2(\mathbf{x}^{(1)*})$ and $Y_1(\mathbf{x}^{(2)*})$ into account. $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(2)*}$ may not even dominate $\mathbf{R}$ while scoring a high mq-EI. For these reasons, the mq-EI criterion breaks the coupling through $\mathbf{x}$ between the functions, allocating marginally each point to an objective. mq-EI does not tackle multi-objective problems.

Following the definition of q-EI (5.1), a proper multi-point extension of mEI (4.5) is

$$\text{q-mEI}(\{\mathbf{x}^{(t+1)},\ldots,\mathbf{x}^{(t+q)}\};\mathbf{R}) = \mathbb{E}\left[\max_{i=1,\ldots,q}\left(\prod_{j=1}^{m}(R_j - Y_j(\mathbf{x}^{(t+i)}))_+\right)\right]. \tag{5.3}$$

5.1.1.2 Properties of both criteria

We now give some properties and bounds for both criteria.

Proposition 5.1. *When evaluated twice at the same design, mq-EI and qm-EI reduce to mEI: mq-EI*$(\{\mathbf{x},\mathbf{x}\};\mathbf{R})$ = *q-mEI*$(\{\mathbf{x},\mathbf{x}\};\mathbf{R})$ = *mEI*$(\mathbf{x};\mathbf{R})$.

Proof:
mq-EI$(\{\mathbf{x},\mathbf{x}\};\mathbf{R}) = \prod_{j=1}^{m}$ q-EI$_j(\{\mathbf{x},\mathbf{x}\};R_j) = \prod_{j=1}^{m}EI_j(\mathbf{x};R_j)$ = mEI$(\mathbf{x};\mathbf{R})$.
q-mEI$(\{\mathbf{x},\mathbf{x}\};\mathbf{R}) = \mathbb{E}[(\prod_{j=1}^{m}(R_j - Y_j(\mathbf{x}))_+)]$ = mEI$(\mathbf{x};\mathbf{R})$. □

Proposition 5.2. *When* $\mathcal{P}_\mathcal{Y} \nprec \mathbf{R}$, *q-mEI calculated at two training points* $\mathbf{x}$ *and* $\mathbf{x}'$ *is null. q-mEI calculated at one training point* $\mathbf{x}$ *and one new point* $\mathbf{x}''$ *reduces to mEI at the latter: q-mEI*$(\{\mathbf{x},\mathbf{x}'\};\mathbf{R}) = 0$, *q-mEI*$(\{\mathbf{x},\mathbf{x}''\};\mathbf{R})$ = *mEI*$(\mathbf{x}'';\mathbf{R})$.

Proof:
As $\mathbf{x}$ and $\mathbf{x}'$ are training points, $\mathbf{Y}(\mathbf{x})$ and $\mathbf{Y}(\mathbf{x}')$ are no longer random variables, and the expectation vanishes. Since $\mathbf{R}$ is not dominated by the observed values $\mathbf{y} = \mathbf{Y}(\mathbf{x})$ and $\mathbf{y}' = \mathbf{Y}(\mathbf{x}')$, $\prod_{j=1}^{m}(R_j - Y_j(\mathbf{x}))_+ = \prod_{j=1}^{m}(R_j - y_j)_+ = 0$ and the same occurs with $\mathbf{y}'$. Finally, q-mEI$(\{\mathbf{x},\mathbf{x}'\};\mathbf{R}) = 0$.
In the case of one observed $\mathbf{x}$ and one unobserved $\mathbf{x}''$, $\prod_{j=1}^{m}(R_j - Y_j(\mathbf{x}''))_+ \geq \prod_{j=1}^{m}(R_j - Y_j(\mathbf{x}))_+ = 0$, and q-mEI$(\{\mathbf{x},\mathbf{x}''\};\mathbf{R}) = \mathbb{E}[\prod_{j=1}^{m}(R_j - Y_j(\mathbf{x}''))_+]$ = mEI$(\mathbf{x}'';\mathbf{R})$. □

Even though these properties seem obvious and mandatory for a multi-point infill criterion, they do not hold for mq-EI. To see this, let us consider a case with $m = 2$ objectives, $\mathbf{R}$ a non-dominated reference point, and $\mathbf{x}$ and $\mathbf{x}'$ two evaluated designs with responses $\mathbf{y} = \mathbf{f}(\mathbf{x}) = (y_1, y_2)^\top$, $\mathbf{y}' = \mathbf{f}(\mathbf{x}') = (y'_1, y'_2)^\top$, satisfying $y_1 < R_1 < y'_1$ and $y'_2 < R_2 < y_2$. By definition, mq-EI$(\{\mathbf{x},\mathbf{x}'\};\mathbf{R}) = \prod_{j=1}^{2}\mathbb{E}[\max((R_j - y_j)_+, (R_j - y'_j)_+)] = (R_1 - y_1)(R_2 - y'_2) > 0$. Furthermore, mq-EI$(\{\mathbf{x},\mathbf{x}''\};\mathbf{R}) = \prod_{j=1}^{2}\mathbb{E}[\max((R_j - y_j)_+, (R_j - Y_j(\mathbf{x}''))_+)]$ = EI$_2(\mathbf{x}'';R_2)\times\mathbb{E}[\max((R_1 - y_1), (R_1 - Y_1(\mathbf{x}''))_+)] >$ EI$_2(\mathbf{x}'';R_2)\times$EI$_1(\mathbf{x}'';R_1)$ = mEI$(\mathbf{x}'';\mathbf{R})$.

Some bounds can also be computed. We assume $q \geq m$ which will usually be verified. Let us denote $\mathbf{x}^{(j)*}$ the maximizers of EI$_j(\cdot;R_j)$ for $j = 1,\ldots,m$; $\mathbf{x}^{(m+1)*},\ldots,\mathbf{x}^{(q)*}$ any other points and $\mathbf{x}^*$ the maximizer of mEI$(\cdot,\mathbf{R})$. Then,

$$\max_{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(q)}} \text{mq-EI}(\{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(q)}\};\mathbf{R}) = \max_{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(q)}} \prod_{j=1}^{m} \text{q-EI}_j(\{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(q)}\};\mathbf{R}_j)$$
$$\geq \prod_{j=1}^{m} \text{q-EI}_j(\{\mathbf{x}^{(1)*},\dots,\mathbf{x}^{(m)*},\mathbf{x}^{(m+1)*},\dots,\mathbf{x}^{(q)*}\};\mathbf{R}_j) \geq \prod_{j=1}^{m} \text{EI}_j(\mathbf{x}^{(j)*};\mathbf{R}_j)$$

This inequality shows that mq-EI's maximum value is greater than the product of expected improvement maxima, which shows that this criterion does not minimize $f_1(\cdot),\dots,f_m(\cdot)$ jointly. The last term can be further lower bounded, $\prod_{j=1}^{m} \text{EI}_j(\mathbf{x}^{(j)*};\mathbf{R}) \geq \prod_{j=1}^{m} \text{EI}_j(\mathbf{x}^{*};\mathbf{R}) = \text{mEI}(\mathbf{x}^*;\mathbf{R})$.

For q-mEI, a trivial lower bound is the mEI maximum: $\max\limits_{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(q)}} \text{q-mEI}(\{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(q)}\};\mathbf{R}) \geq \max\limits_{\mathbf{x}} \text{mEI}(\mathbf{x};\mathbf{R}) = \text{mEI}(\mathbf{x}^*;\mathbf{R})$.

These lower bounds indicate that more improvement is expected within the q steps than during a single mEI step.

5.1.1.3 Kriging Believer strategy

The Kriging Believer strategy (Ginsbourger et al., 2010) was introduced in Section 4.6 for anticipating the behavior of the algorithm during the remaining iterations. Here it is employed to produce a batch of q points where to evaluate the simulator. At each iteration, the single-point mEI is maximized and its optimum $\mathbf{x}^{*KB}$ is virtually appended to the metamodel using the kriging predictions $\widehat{\mathbf{y}}(\mathbf{x}^{*KB})$ as an emulator for $\mathbf{f}(\cdot)$. The metamodel is updated (this procedure only changes the variances, $\mathbf{s}^2(\cdot)$, which vanish at $\mathbf{x}^{*KB}$ that will no longer be promoted) and a batch of designs to be evaluated by the true functions is generated by repeating this procedure q times successively. The Kriging Believer does not require Monte Carlo simulations being only made of analytical single-point mEI maximizations, for which even the gradient is available (see Section 4.3.2). The d dimensional space over which it is defined constitutes a supplementary advantage for its maximization and q-mEI-KB does not suffer as much as q-mEI from q's increase. It however heavily depends on the metamodel since each mEI maximizer is virtually incorporated together with its kriging prediction. This may be a drawback for functions that are weakly approximated by GPs and/or when the number of observations is too small to have an accurate surrogate. In the following, we will denote this criterion q-mEI-KB and compare it with the q-mEI and the sequential mEI.

In the Kriging Believer strategy, the virtual observations $\widehat{\mathbf{y}}(\mathbf{x}^{*KB})$ may dominate $\mathbf{R}$, which is recomputed (projection of the virtual empirical Pareto front $\widehat{\mathcal{P}_{\mathcal{Y}}}$ onto $\mathcal{L}$) at each iteration to stay non-dominated. While minor differences have been observed for small batches ($q = 2$ or $q = 4$), the update of $\mathbf{R}$ at each virtual step has shown to produce better results in case of larger batches ($q = 10$) than keeping the initial $\mathbf{R}$ during the q virtual steps. In the latter case, less diversity was observed as virtual solutions dominating $\mathbf{R}$ attracted the search.

5.1.1.4 Experiments with the batch targeting criteria

We now investigate the capabilities of the batch versions of mEI and compare them with the results obtained by the sequential mEI (Section 4.3.2). First, in Section 5.1.1.4, a comparison between q-mEI and mq-EI is made on the basis of two simple one-dimensional quadratic functions. This example illustrates why q-mEI is the correct multi-point extension of mEI. Then, the batch criterion q-mEI is compared with the sequential mEI for finding the Pareto front center using the MetaNACA test bed (Chapter 3). Finally, a larger comparison investigates the sequential and batch criteria on analytical functions and where an off-centered preference region is specified, as in Section 4.4.3.1.

Note that in the experiments, parallel executions of the algorithms are simulated on sequential computers. As usual in Bayesian optimization, we assume that the computation time is mainly taken by the calls to the objective functions and there are sufficient computing resources so that the speed-up is close to q. The term “wall-clock time” will therefore mean the number of calls to the objective functions divided by the batch size q.

The q-mEI and mq-EI criteria of formula (5.3) and (5.2) are calculated by Monte Carlo simulation with $N_{MC} = 10,000$ samples. To be more precise, q-mEI (5.3) at a candidate batch $\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}$ is computed by averaging over N_{MC} (joint) conditional GPs $\widetilde{Y}_j^{(k)}$, which leads to the estimator

$$\widehat{\text{q-mEI}}(\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}; \mathbf{R}) = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \left[\max_{i=1,\dots,q} \left(\prod_{j=1}^{m} (R_j - \widetilde{Y}_j^{(k)}(\mathbf{x}^{(t+i)}))_+ \right) \right]. \quad (5.4)$$

Because the optimization of the criteria is carried out in a $q \times d$ dimensional space and the gradients are not available, in the experiments, the number of iterates evaluated simultaneously is restricted to $q = 2$ and 4. Larger batches can be employed with the Kriging Believer criterion.

Comparison between mq-EI and q-mEI on quadratic functions

To compare q-mEI with mq-EI, we consider a simple example with $d = 1$, $q = 2$ and $m = 2$ quadratic objective functions:

$$\min_{x \in [0,1]} (f_1(x), f_2(x))$$

where $f_1(x) = 0.6x^2 - 0.24x + 0.1$ and $f_2(x) = x^2 - 1.8x + 1$, whose minima are respectively 0.2 and 0.9. The multi-objective optimality conditions (Miettinen, 1998) show that the Pareto set is $\mathcal{P}_\mathcal{X} = [0.2, 0.9]$ and the Pareto front $\mathcal{P}_\mathcal{Y} = \{\mathbf{y} = (f_1(x), f_2(x))^\top, x \in [0.2, 0.9]\}$. f_1 and f_2 are plotted in red in Figure 5.1, both in the design space $X = [0, 1]$ and in the objective space. Two independent GPs, $Y_1(\cdot)$ and $Y_2(\cdot)$, are fitted to $n = 3$ data points, $x^{(1)} = 0.05$, $x^{(2)} = 0.6$ and $x^{(3)} = 0.95$. Figure 5.1 also shows the kriging predictors $\widehat{f_1}(x)$ and $\widehat{f_2}(x)$, as well as the empirical Pareto front.

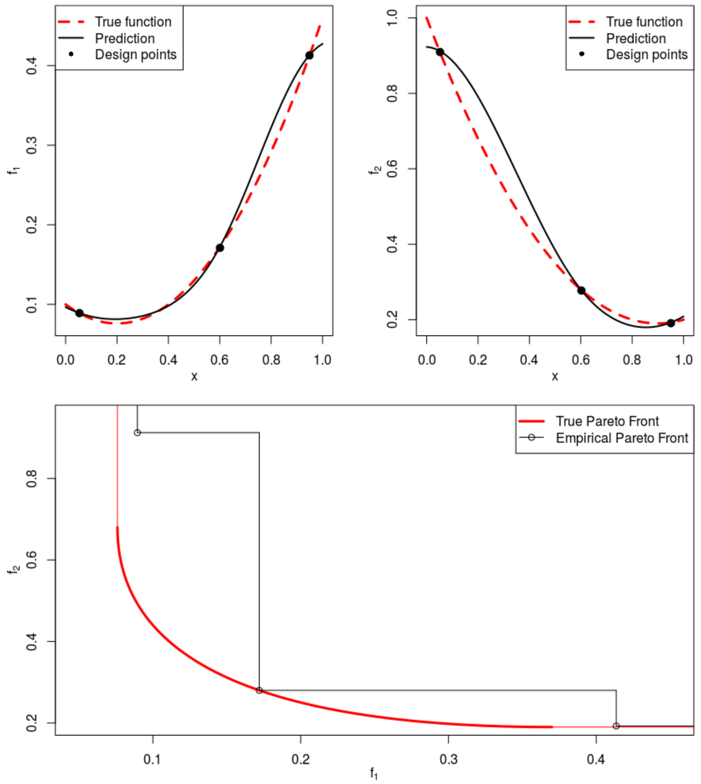


Figure 5.1: Top: Kriging predictors and true f_1 and f_2 functions. Bottom: true and empirical Pareto fronts.

Let us take the non-dominated reference point $\mathbf{R} = (0.15, 0.42)^\top$ that we will use both with mq-EI and q-mEI. With that reference point, shown in green in Figure 5.2, domination of $\mathbf{R}$ is achieved when $x \in [0.42, 0.55]$.

In a first experiment, we fix $x^{(n+1)}$ (but it is not a training point, its objective values are handled through the GPs) and search for the $x^{(n+2)}$ maximizing mq-EI($\{x^{(n+1)}, x^{(n+2)}\}; \mathbf{R}$) and q-mEI($\{x^{(n+1)}, x^{(n+2)}\}; \mathbf{R}$). Besides illustrating the difference between q-mEI and mq-EI, this experiment may serve as an introduction to the asynchronous versions of the batch criteria (Janusevskis et al., 2012), further discussed in Section 5.1.2, which are important in practical parallel implementations: as soon as one computing node becomes available, the q-points criteria are optimized with respect to 1 point while keeping the $q-1$ other points fixed at their currently running values. Two different settings are considered whose results are presented in Figures 5.2 and 5.3.

In the first setting, $x^{(n+1)} = 0.2$ is a bad choice as it corresponds to an extreme point of the Pareto set and its future response will not dominate $\mathbf{R}$, an information already seen on the GPs. q-mEI gives $x^{(n+2)} = 0.49$ which is very close to the (one-step) mEI maximizer, hence a relevant input as $\mathbf{f}(x^{(n+2)})$ will dominate $\mathbf{R}$. On the contrary, mq-EI separates the objectives. As $x^{(n+1)}$ is a good input for objective $f_1(\cdot)$, the criterion reaches its maximum when $x^{(n+2)} = 0.86$, which is a good input when considering $f_2(\cdot)$ alone. Figure 5.1 tells us that 0.86 is almost the minimizer of $\widehat{f_2}(x)$. However, the original goal of dominating $\mathbf{R}$ is not achieved.

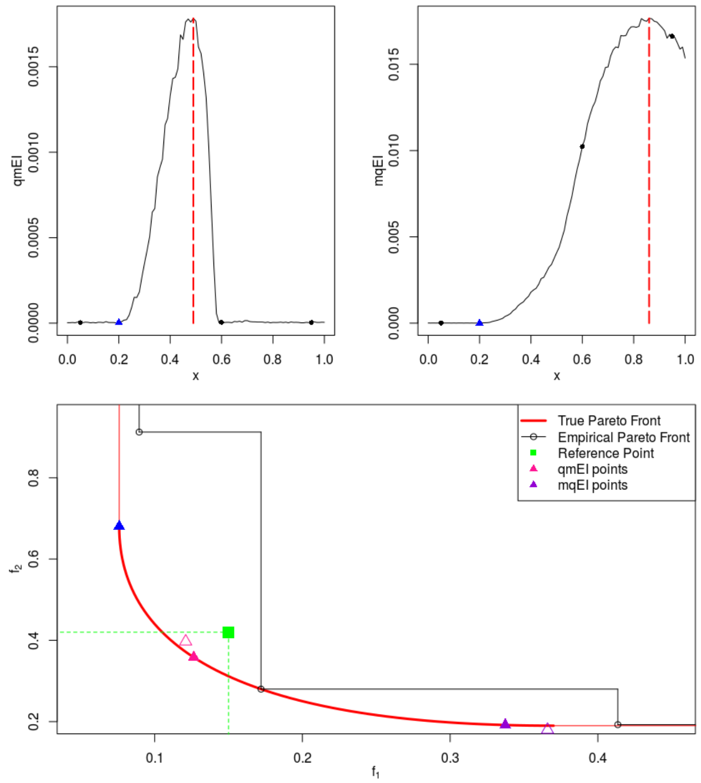


Figure 5.2: Setting 1: $x^{(n+1)} = 0.2$ (blue triangle). Top: q-mEI($\{x^{(n+1)}, x\}; \mathbf{R}$) (left) and mq-EI($\{x^{(n+1)}, x\}; \mathbf{R}$) (right) criteria for the second input in the design space. The maximum is achieved at different locations for both criteria. Also notice that for training points $x^{(i)}$ (black dots), mq-EI($\{x^{(n+1)}, x^{(i)}\}; \mathbf{R}$) $\neq$ mEI($x^{(n+1)}; \mathbf{R}$) ≈ 0, contrarily to q-mEI($x^{(n+1)}, x^{(i)}$). Bottom: corresponding values for $\mathbf{f}(x^{(n+2)})$. q-mEI provides an input whose image (pink) dominates $\mathbf{R}$. On the contrary, mq-EI's solution concentrates on the minimization of the second objective (purple). The transparent triangles correspond to the kriging predictions at $x^{(n+2)}$.

In the second setting, $x^{(n+1)} = 0.46$ is a good point as its image will dominate $\mathbf{R}$. q-mEI leads to $x^{(n+2)} = 0.53$ whose image also dominates $\mathbf{R}$. Notice that as 0.46 is chosen for $x^{(n+1)}$, the point that jointly maximizes q-mEI with that first point is slightly larger than 0.48 (the mEI maximizer), and provides more diversity in $\mathcal{I}_{\mathbf{R}}$. The second input for maximizing mq-EI is $x^{(n+2)} = 0.83$, an input that is good only to minimize $f_2(\cdot)$ (it is almost the same as in the previous case) but $\mathbf{f}(x^{(n+2)})$ does not dominate $\mathbf{R}$.

Now, we optimize directly mq-EI and q-mEI with respect to both inputs $x^{(n+1)}$ and $x^{(n+2)}$. The optimal batches are $\{0.43, 0.51\}$ for q-mEI and $\{0.26, 0.87\}$ for mq-EI. Figure 5.4 shows that these inputs lead to $\mathcal{I}_{\mathbf{R}}$ with q-mEI. On the contrary, the images of mq-

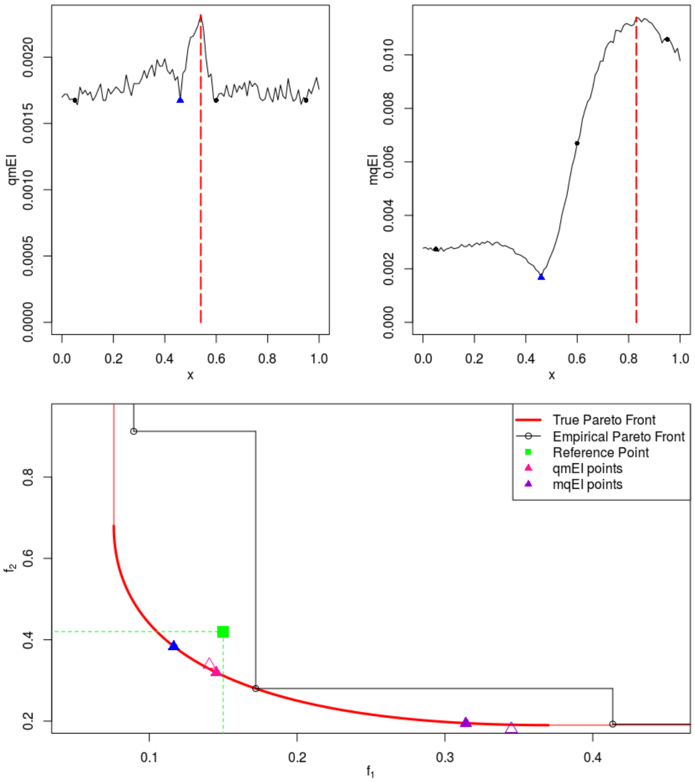


Figure 5.3: Setting 2: $x^{(n+1)} = 0.46$ (blue triangle). Top: q-mEI($\{x^{(n+1)}, x\}; \mathbf{R}$) (left) and mq-EI($\{x^{(n+1)}, x\}; \mathbf{R}$) (right) criteria for the second input in the design space whose maximum is again achieved at different locations. Bottom: corresponding values for $\mathbf{f}(x^{(n+2)})$. q-mEI provides an input whose image (pink) also dominates $\mathbf{R}$. On the contrary, mq-EI returns an input which concentrates on the minimization of the second objective (purple). The transparent triangles correspond to the kriging predictions at $x^{(n+2)}$.

EI's optimum are located at the boundaries of the Pareto front and none of them is in $\mathcal{I}_{\mathbf{R}}$. Figure 5.4 further indicates that q-mEI is high when both inputs are in the part of the design space that leads to domination of $\mathbf{R}$ (gray box) contrarily to mq-EI, which is high when each input leads to the improvement over one component of $\mathbf{R}$. Note that even though both criteria are symmetric with respect to their q inputs, the symmetry is slightly broken in the figure because of the Monte Carlo estimation.

Batch targeting of the Pareto front center

We now compare the multi-point criteria, q-mEI and q-mEI-KB, with the sequential mEI. As in Section 4.4.3.3, the tests are performed with the MetaNACA benchmark in $d = 8$ and $d = 22$ dimensions, and with $m = 2$ objectives. No user-defined reference point is provided so the center of the Pareto front is targeted.

The ability of mEI to quickly attain and converge towards central parts of the Pareto

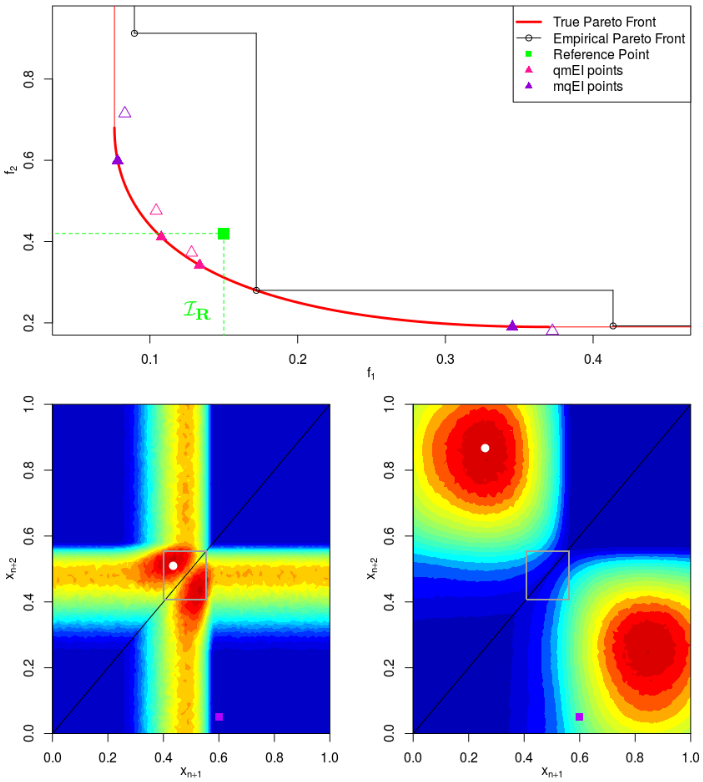


Figure 5.4: 2 points mq-EI and q-mEI. Top: values obtained in the objective space using both criteria. The images of $x^{(n+1)}$ and $x^{(n+2)}$ returned by q-mEI (pink) both dominate $\mathbf{R}$. None of those returned by mq-EI (purple) are in $\mathcal{I}_\mathbf{R}$, they rather improve over each function individually. Transparent triangles correspond to the kriging mean predictor. Bottom: criteria values for varying ($x^{(n+1)}$,$x^{(n+2)}$). For q-mEI (left), the best x's, in dark red, lead to domination of $\mathbf{R}$ (gray box). Conversely, for mq-EI (right), good x's improve upon $\mathbf{R}$'s components for each objective. The white dots correspond to both optima. The purple square is an example of a training point pair where q-mEI is null but mq-EI is not (this holds for all other training point pairs that are not shown).

front in comparison with the Bayesian EHI infill criterion and with NSGA-II was highlighted in Chapter 4. For the sake of clarity, only mEI's results are recalled here since it significantly outperformed EHI and NSGA-II. As in Table 4.4, the attainment time of a central part of $\mathcal{P}_\mathcal{Y}$ ($\mathcal{I}_{0.1}$, dominated by $\mathbf{R}_{0.1}$) and the hypervolume restricted to it are the comparison metrics. Since q-mEI's interest resides in the distributed computation of $\mathbf{f}(\cdot)$, an additional indicator is the number of calls to the infill criterion, denoted $\#crit$. Supposing q supplementary designs are evaluated after each call to the infill criterion, this metric enables the comparison of attainment times at approximately the same wall-clock time. mEI and q-mEI's performance are reported at different times of the optimization to compare the criteria at both the same number of function evaluations and the same number of iterations.

Before turning to statistically more significant comparisons where runs are repeated, Figure 5.5 allows a graphical comparison of the effects of the sequential and batch mEI criteria at constant wall-clock time or constant number of calls to the objective functions, on the MetaNACA in 8 dimensions. Under our assumptions of costly objective functions, 2-mEI with 2 × 10 iterations and mEI with 10 iterations roughly need the same wall-clock time. Similarly, 4-mEI with 4 × 5 iterations and mEI with fourth budget take the same time. On both rows of the figure, it is seen that at the same wall-clock time, q-mEI's approximations to the front center (left) are improved when compared to mEI's (right). For an equal number of added points, 2-mEI and mEI provide equivalent approximations to the center, and 4-mEI is slightly degraded (but the time is divided by 4). At the same number of evaluations, the small deterioration of q-mEI's results over those of mEI is explained as follows: when q increases, the batch versions of the criterion affect resources (i.e., choose the $\mathbf{x}$'s to be calculated and re-estimate the center) with increasingly incomplete information.

Tables 5.1 and 5.2 report the averages and standard deviations of the performance metrics over 10 independent runs for mEI, q-mEI and q-mEI-KB with $q = 2$ and 4 (and 10 in the KB version). Because of the $d \times q$ dimensional criterion input space, q-mEI is only considered with $q = 2$ for the MetaNACA 22. The number of considered function evaluations is reported in the row "Budget", where the first figure stands for the initial DoE ($n = 20$ for $d = 8$ and $n = 50$ for $d = 22$). mEI$_{\text{half}}$ and mEI$_{\text{fourth}}$ correspond to optimizations stopped at half or fourth the allowed iterations, $p = 20$ for $d = 8$ and $p = 50$ for $d = 22$. The t subscript indicates q-mEI is considered at the same wall clock time, i.e. after p iterations during which $p \times q$ supplementary design have been evaluated, and the blue color indicates that even less than p iterations were performed. In absence of the subscript, q-mEI runs with p additional designs are considered. As in Chapter 4, if at least one run does not enter in $\mathcal{I}_{0.1}$, an estimator of the empirical runtime is given in red together with the number of successful runs in brackets.

These empirical results indicate that at the same wall-clock time, q-mEI outperforms mEI in attainment time of $\mathbf{R}_{0.1}$: even though mEI attains this central target after less function evaluations, q-mEI is able to perform q calls to $\mathbf{f}(\cdot)$ during one iteration, which leads to a faster attainment of the center in terms of calls to the criterion with q-mEI than with mEI. Generally, at a fixed number of function evaluations, the hypervolume lightly decreases with the batch size, as q less metamodel updates and center-estimations are performed. However, at an equal number of iterations, $\mathcal{I}_{0.1}$ is attained faster in wall-clock time and the hypervolume becomes larger for increasing q's. The number of iterations after which the convergence criterion described in Section 4.5 is triggered (not reported here) also diminishes when using the multi-point criteria: the second phase of C-EHI would start earlier in wall-clock time, in a "true C-EHI" optimization[1].

The Kriging Believer strategy performs good on these test problems and outperforms the q-mEI criterion. Additionally, larger batches ($q = 10$) can be employed, and exhibit good performance after few iterations. For instance, 10mEI-KB largely outperforms

[1] For comparison purposes, we continue targeting the center even if local convergence is detected.

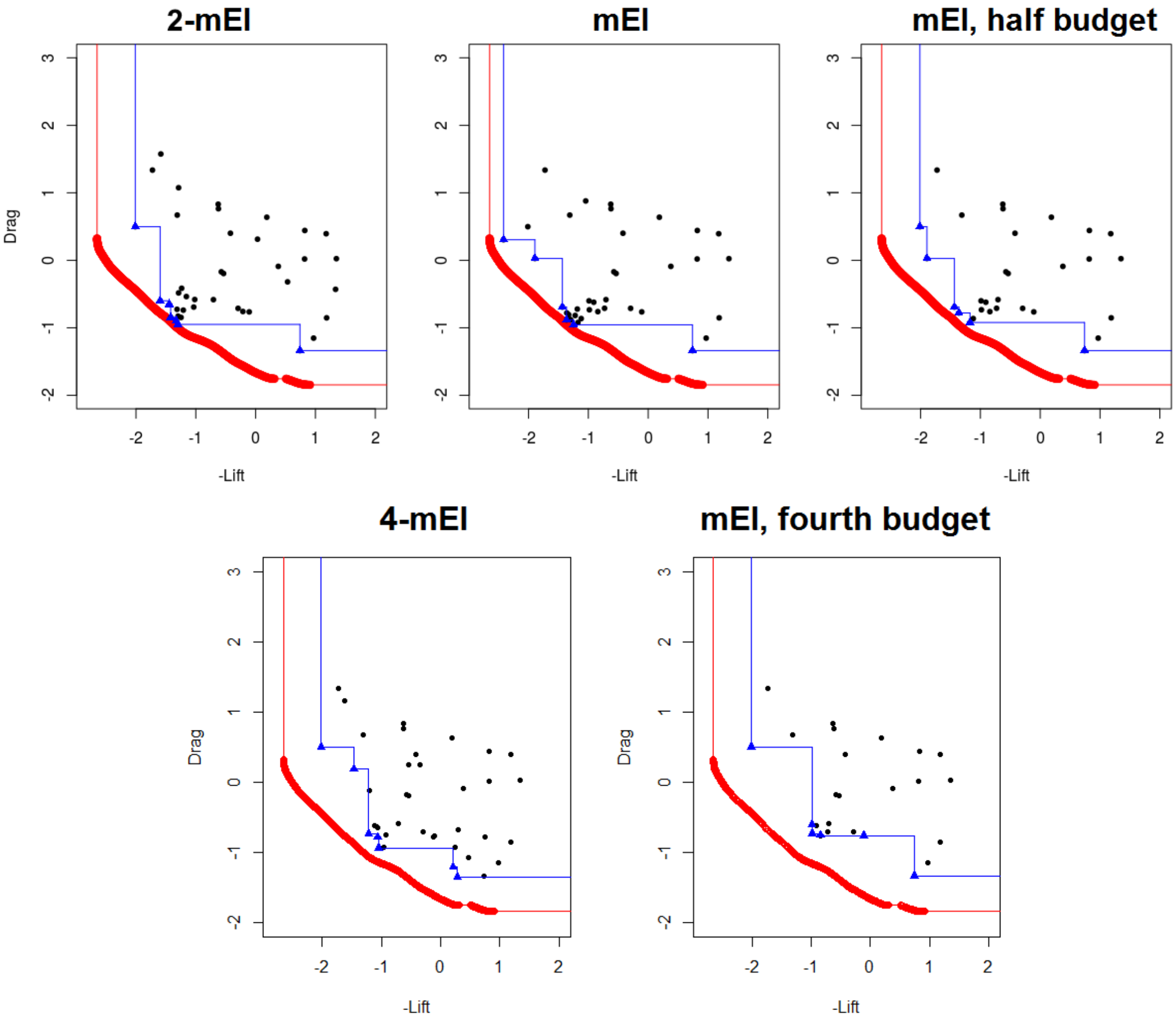


Figure 5.5: Top: example run using 2-mEI (left) with 2 × 10 additional designs, and mEI with 20 (center) and 10 (right) iterations. Bottom: example run using 4-mEI (left) with 4 × 5 additional designs, and mEI with 5 iterations (right). At a fixed wall-clock time, q-mEI converges more accurately to the center than mEI. At a fixed number of evaluations, the degradation is small.

other variants in terms of hypervolume after solely 8 supplementary iterations on the MetaNACA 8. These test functions may nevertheless overestimate the performance of KB strategies, because they stem from a Gaussian Process and may be learned easily.

Runs which do not attain $\mathcal{I}_{0.1}$ are nonetheless more frequent with multi-point criteria, especially on the MetaNACA 22. This is related to less metamodel updates and center estimations, which sometimes lead to the targeting of a slightly off-centered run, assessed as bad by the performance metrics. This bias will not be present in the following section, where the targeted area is defined through a user-defined $\mathbf{R}$.

	mEI			q-mEI			
Criterion	mEI	mEI$_{\text{half}}$	mEI$_{\text{fourth}}$	2mEI	2mEI$_t$	4mEI	4mEI$_t$
Budget	20+20	20+20/2	20+20/4	20+2×10	20+2×20	20+4×5	20+4×20
$\#\mathbf{f}(\cdot)$ to target	28.4 (5.4)	36.9 [7]	72.2 [3]	35.6 [8]	33.1 (5.7)	71.9 [4]	44.5 (17.3)
$\#crit$ to target	8.4 (5.4)	8.4 [7]	5.5 [3]	5.3 [8]	6.6 (2.9)	5.5 [4]	6.1 (4.3)
Hypervolume	0.256 (0.09)	0.134 (0.15)	0.077 (0.13)	0.170 (0.13)	0.280 (0.16)	0.056 (0.09)	0.296 (0.19)

	q-mEI-KB					
Criterion	2mEI-KB	2mEI-KB$_t$	4mEI-KB	4mEI-KB$_t$	10mEI-KB	10mEI-KB$_t$
Budget	20+2×10	20+2×20	20+4×5	20+4×20	20+10×2	20+10×8
$\#\mathbf{f}(\cdot)$ to target	34.2 [9]	33 (8.9)	57.6 [5]	38.5 (11.3)	49.2 [6]	37.2 (13.7)
$\#crit$ to target	6.0 [9]	6.5 (2.3)	4.4 [5]	4.6 (11.3)	2.8 [6]	2.5 (1.4)
Hypervolume	0.221 (0.14)	0.361 (0.12)	0.128 (0.16)	0.466 (0.25)	0.040 (0.08)	0.531 (0.17)

Table 5.1: MetaNACA 8 problem, indicators computed in $\mathcal{I}_{0.1}$ for mEI, q-mEI and q-mEI-KB at identical number of evaluations or wall-clock times. Averages (std. deviation) over 10 runs.

	mEI		q-mEI	
Criterion	mEI	mEI$_{\text{half}}$	2mEI	2mEI$_t$
Budget	50+50	50+50/2	50+2×25	50+2×50
$\#\mathbf{f}(\cdot)$ to target	56.3 (7.2)	56.3 (7.2)	71.3 [8]	71.3 [8]
$\#crit$ to target	6.3 (7.2)	6.3 (7.2)	4.7 [8]	4.7 [8]
Hypervolume	0.222 (0.12)	0.139 (0.10)	0.085 (0.09)	0.119 (0.10)

	q-mEI-KB					
Criterion	2mEI-KB	2mEI-KB$_t$	4mEI-KB	4mEI-KB$_t$	10mEI-KB	10mEI-KB$_t$
Budget	50+2×25	50+2×50	50+4×12	50+4×25	50+10×5	50+10×25
$\#\mathbf{f}(\cdot)$ to target	73.0 [8]	68.9 (23.0)	63.6 (12.7)	63.6 (12.7)	69.1 [9]	59.7 (7.2)
$\#crit$ to target	3.3 [8]	5.3 (5.8)	4 (3.1)	4 (3.1)	1.9 [9]	1.7 (0.7)
Hypervolume	0.121 (0.11)	0.260 (0.14)	0.215 (0.14)	0.398 (0.16)	0.100 (0.08)	0.440 (0.26)

Table 5.2: MetaNACA 22 problem, indicators computed in $\mathcal{I}_{0.1}$ for mEI, q-mEI and q-mEI-KB at identical number of evaluations or wall-clock times. Averages (std. deviation) over 10 runs.

Batch targeting of a user-defined region

Let us analyze the ability of q-mEI to attain a region of the Pareto front defined through a reference point $\mathbf{R}$. Following the experiments of Section 4.4.3.1, ZDT3 (in dimension $d = 4$) and P1 are employed, and $\mathbf{R}$ is taken as the Nadir point of the second sub-front, $\mathbf{R} = (0.258, 0.670)^\top$, and $\mathbf{R} = (10, -23)^\top$, respectively.

The sequential mEI is compared to q-mEI and to q-mEI-KB for batches of $q = 2$ and $q = 4$ designs. The initial DoE has size $n = 20$ for ZDT3 and $n = 8$ for P1. $p = 20$ or

$p = 12$ additional iterations are allowed. Remember that during these 20 (respectively 12) iterations, q-mEI and q-mEI-KB enables to evaluate $\mathbf{f}(\cdot)$ $20 \times q$ $(12 \times q)$ times, against 20 or 12 function evaluations for mEI.

As in the previous section, to compare the criteria at a fixed budget, the optimization is not stopped nor R-EHI's second phase starts once the local convergence criterion to the Pareto front is triggered, even though this situation frequently occurs. The same metrics (hypervolume, attainment times) are employed and shown in Tables 5.3 and 5.4. They are now restricted to $\mathbf{R}$ instead of the central $\mathbf{R}_{0.1}$. Additionally, the number of solutions that eventually dominate $\mathbf{R}$ is provided. This indicator does not express convergence to the Pareto front, but the capability to produce user-desired outputs.

		q-mEI				q-mEI-KB			
Criterion	mEI	2mEI	2mEI$_t$	4mEI	4mEI$_t$	2mEI-KB	2mEI-KB$_t$	4mEI-KB	4mEI-KB$_t$
Budget	20+20	20+2×10	20+2×20	20+4×5	20+4×20	20+2×10	20+2×20	20+4×5	20+4×20
#$\mathbf{f}(\cdot)$ to target	24.2 (2.6)	26.3 (4.3)	26.3 (4.3)	32.7 [9]	32.5 (6.6)	24.2 (2.6)	24.2 (2.6)	33.8 [8]	33.2 (15.8)
#*crit* to target	4.2 (2.6)	3.2 (2.2)	3.2 (2.2)	2.6 [9]	3.1 (1.7)	2.4 (1.3)	2.4 (1.3)	2.4 [8]	3.9 (4.0)
Hypervolume	0.634 (0.078)	0.548 (0.201)	0.621 (0.147)	0.424 (0.227)	0.622 (0.088)	0.513 (0.149)	0.513 (0.149)	0.445 (0.251)	0.518 (0.201)
Solutions $\preceq \mathbf{R}$	4.1 (1.8)	2.8 (1.0)	3.6 (0.8)	1.5 (1.0)	2.4 (1.0)	2.4 (0.7)	2.4 (0.7)	2.1 (0.9)	2.2 (0.8)

Table 5.3: Comparison of the different infill criteria on the ZDT3 function. The results are averaged over 10 runs, and the standard deviation is shown in brackets.

		q-mEI				q-mEI-KB			
Criterion	mEI	2mEI	2mEI$_t$	4mEI	4mEI$_t$	2mEI-KB	2mEI-KB$_t$	4mEI-KB	4mEI-KB$_t$
Budget	8+12	8+2×6	8+2×12	8+4×3	8+4×12	8+2×6	8+2×12	8+4×3	8+4×12
#$\mathbf{f}(\cdot)$ to target	12.6 (3.5)	12.7 (2.6)	12.7 (2.6)	15.1 (3.9)	15.1 (3.9)	12.6 [9]	13 (7.8)	15.5 [8]	14.3 (7.3)
#*crit* to target	4.6 (3.5)	2.4 (1.3)	2.4 (1.3)	1.8 (1.0)	1.8 (1.0)	3.0 [9]	3.4 (2.9)	2.5 [8]	2.4 (1.4)
Hypervolume	0.620 (0.165)	0.624 (0.063)	0.686 (0.042)	0.437 (0.207)	0.718 (0.054)	0.393 (0.214)	0.451 (0.167)	0.205 (0.190)	0.540 (0.179)
Solutions $\preceq \mathbf{R}$	6.5 (2.5)	5.6 (1)	9.7 (1.1)	2.6 (1.6)	11.4 (2.1)	1.8 (0.9)	2.3 (1.5)	1.2 (0.8)	4.1 (3.1)

Table 5.4: Comparison of the different infill criteria on the P1 function. The results are averaged over 10 runs, and the standard deviation is shown in brackets.

As observed when the center of $\mathcal{P}_{\mathcal{Y}}$ was targeted (Tables 5.1 and 5.2), the multi-point q-mEI and q-mEI-KB are able to attain $\mathcal{I}_{\mathbf{R}}$ faster than mEI in terms of wall-clock time. At the same number of function evaluations, the hypervolume lightly diminishes with q, but is larger when the results are compared at the same wall-clock time. Here, q-mEI performs better than q-mEI-KB. The P1 functions are less easily learned by a GP and the small number of observations degrades the performance of the Kriging Believer strategy. q-mEI-KB does not attain $\mathcal{I}_{\mathbf{R}}$ during the first p additional function evaluations in all runs, and the relatively large attainment time of 4mEI-KB$_t$ on ZDT3 is due to one run which took 19 iterations (93 function evaluations in total) before entering in $\mathcal{I}_{\mathbf{R}}$. The ZDT3 functions are easily learned by the GP but as the updated reference point $\widehat{\mathbf{R}}$ has quickly attained $\mathcal{P}_{\mathcal{Y}}$, q-mEI and q-mEI-KB become exploratory, hence the little increase in hypervolume between q-mEI and q-mEI$_t$ (and between q-mEI-KB and q-mEI-KB$_t$). This also the reason why fewer designs dominating $\mathbf{R}$ are found by q-mEI and q-mEI-KB; $\widehat{\mathbf{R}}$ attains $\mathcal{P}_{\mathcal{Y}}$ slower with mEI.

5.1.2 Towards a multi-point EHI: q-EHI and variants

5.1.2.1 The q-EHI criterion

A generic form of the q-EI (5.3) is $\mathbb{E}[\max\limits_{i=1,\dots,q}(I(\mathbf{x}^{(t+i)}))]$, where I is the improvement measure of the infill criterion, evaluated at one design $\mathbf{x} \in X$. In the case of the EI, I is the magnitude of progress measured by $I(\mathbf{z}) = (f_{\min} - Y(\mathbf{z}))_+$[2]. In the framework of EHI, the latter is the hypervolume increase: $I(\mathbf{z}) = I_H(\mathcal{P}_{\mathcal{Y}} \cup \{\mathbf{Y}(\mathbf{z})\}; \mathbf{R}) - I_H(\mathcal{P}_{\mathcal{Y}}; \mathbf{R})$ where $\mathcal{P}_{\mathcal{Y}}$ stands for the current approximation front, $\mathbf{Y}(\cdot)$ for the m metamodels, I_H for the hypervolume indicator (Zitzler, 1999) and $\mathbf{R}$ is the reference point. Following the EI-to-q-EI extension, a possible formulation of q-EHI is therefore

$$\text{q-EHI}_{\max}(\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}; \mathbf{R}) = \mathbb{E}[\max_{i=1,\dots,q} I_H(\mathcal{P}_{\mathcal{Y}} \cup \{\mathbf{Y}(\mathbf{x}^{(t+i)})\}; \mathbf{R})] - I_H(\mathcal{P}_{\mathcal{Y}}; \mathbf{R}). \quad (5.5)$$

However, the max operator may not be appropriate for multi-objective optimization based on the hypervolume measure. Indeed, it is a competition operator between the q points. While this can be understood in a mono-objective setting, where all designs share the same goal (improving over f_{min}), or in the m-EI sense (improving over $\mathbf{R}$), this is less meaningful with the hypervolume indicator. Since the improvement function aims at increasing the hypervolume the most after adding the batch of q points, the indicator will benefit from collaborative work. Instead of trying to individually improve the most the hypervolume indicator, the q designs should share tasks and focus on different parts where to improve $\mathcal{P}_{\mathcal{Y}}$ to *jointly* increase I_H the most. Instead of (5.5), the following multi-point EHI which looks for the hypervolume increase brought by $\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}$ can therefore be considered (Feliot, 2017):

$$\text{q-EHI}(\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}; \mathbf{R}) = \mathbb{E}[I_H(\mathcal{P}_{\mathcal{Y}} \cup \{\mathbf{Y}(\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)})\}; \mathbf{R})] - I_H(\mathcal{P}_{\mathcal{Y}}; \mathbf{R}). \quad (5.6)$$

5.1.2.2 Optimization of the criterion and hypervolume computation

Like the q-EI, the maximization of q-EHI is carried out in a space of dimension $d \times q$. In our implementation using the `GPareto` (Binois and Picheny, 2015) package, the analytical formula of the regular EHI is solely available for $m = 2$ or 3 objectives (Couckuyt et al., 2014; Emmerich et al., 2006, 2016; Yang et al., 2017) and we rely on Monte Carlo methods

[2] A threshold T can be used in lieu of $f_{\min}$.

when $m > 3$ (Binois and Picheny, 2015; Emmerich et al., 2006) because the expression of EHI for any m has been found very recently only (Yang et al., 2019a). An analytical formula for q-EHI is out of reach, as well as an expression for its gradient, which has been discovered recently for EHI in bi-objective problems (Yang et al., 2019b). Both criteria are estimated using Monte Carlo methods with N_{MC} samples: given a batch $\mathbb{X} = \{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}$, N_{MC} simulations of the joint $\mathbf{Y}(\mathbb{X})$ are carried out. The resulting N_{MC} simulated Pareto fronts are averaged to estimate (5.5) or (5.6). q-EHI's Monte Carlo estimation is

$$\begin{aligned}&\widehat{\text{q-EHI}}(\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}; \mathbf{R}) \\ &= \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \left[\Big(I_H(\mathcal{P}_{\mathcal{Y}} \cup \{\widetilde{\mathbf{Y}}^{(k)}(\mathbf{x}^{(t+1)}), \dots, \widetilde{\mathbf{Y}}^{(k)}(\mathbf{x}^{(t+q)})\}; \mathbf{R}) - I_H(\mathcal{P}_{\mathcal{Y}}; \mathbf{R}) \Big) \right] \qquad (5.7)\end{aligned}$$

where $\widetilde{\mathbf{Y}}^{(k)}(\mathbf{x})$ is the k-th simulated GP at $\mathbf{x}$.

The cost of q-EHI quickly becomes non-negligible because of the large number of hypervolume computations required (Beume et al., 2009; While et al., 2012). In our implementation, we use the genetic algorithm `genoud` (Mebane Jr et al., 2011) with a fixed population size and number of generations whose product is S. In reason of the curse of dimensionality, we make the population proportional to the criterion's dimension: $S \propto \tilde{d} := d \times q$. During the optimization, q-EHI or q-EHI$_{\text{max}}$ are therefore evaluated αdq times.

As (5.5) and (5.6) are evaluated by means of Monte Carlo simulations, N_{MC} hypervolumes are averaged during one q-EHI evaluation. For q-EHI$_{\text{max}}$, the maximum among q hypervolumes is averaged over N_{MC} simulations: qN_{MC} hypervolumes calculations are therefore required. When $N_{MC} = 10,000$, our settings lead to more than 15,000,000 hypervolume calculations for $q = 2$, and more than 30,000,000 when $q = 4$. This is a potential huge drawback, since the computation time of the hypervolume grows exponentially in the number of objectives. Here, we have considered low d, q and m's, but the number of required hypervolume calculations increases for more complex problems. Table 5.5 gives insights in the computation time of one hypervolume for varying numbers of non-dominated points p and objectives m in ms. Our implementation extends the current version of `GPareto` (Binois and Picheny, 2015), relying on the `emoa` package. Even though the computation of hypervolumes is an active field of research (Jaszkiewicz, 2018; Lacour et al., 2017; Russo and Francisco, 2014) and faster implementations than ours to compute hypervolumes exist, when the number of objectives and of non-dominated points grows, the computation and optimization of the criterion may become expensive. A smaller number of samples accelerates q-EHI's computation at the expense of a weaker estimation. Similarly to Janusevskis et al. (2011) in the mono-objective q-EI, it might be possible to derive bounds on q-EHI's precision to choose N_{MC} accordingly.

m/p	5	10	50	100
2	2.7×10^{-3}	4.8×10^{-3}	5.7×10^{-3}	7.2×10^{-3}
3	5.7×10^{-3}	6.8×10^{-3}	18×10^{-3}	24×10^{-3}
4	6.3×10^{-3}	9×10^{-3}	75×10^{-3}	177×10^{-3}
5	7×10^{-3}	11×10^{-3}	0.3	2.2
6	8.2×10^{-3}	25×10^{-3}	2.1	33.5

Table 5.5: Hypervolume computation time (ms) for different number of objectives (m, in rows) and non-dominated points (p, in columns).

5.1.2.3 Other EHI-based multi-point multi-objective infill criteria

In this section, two multi-point multi-objective infill criteria relying solely on the one step look ahead EHI are further introduced.

Asynchronous version

If the expensive functions $f_1(\cdot), \dots, f_m(\cdot)$ have varying runtimes depending on the evaluated design $\mathbf{x}$, it may be worth considering an asynchronous criterion, in order to run a new experiment as soon as a resource gets available. Given $q-1$ pending designs $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q-1)}$, an asynchronous q-EHI is

$$\text{q-EHI}_{\text{async}}(\mathbf{x}; \{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q-1)}\}; \mathbf{R}) = \mathbb{E}[I_H(\mathcal{P}_{\mathcal{Y}} \cup \{\mathbf{Y}(\mathbf{x}, \mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q-1)})\}; \mathbf{R})] - I_H(\mathcal{P}_{\mathcal{Y}}; \mathbf{R}) \tag{5.8}$$

More than permitting the evaluation of the $f_j(\cdot)$'s as soon as a resource gets available, (5.8) presents the additional advantage of being defined in a smaller d (instead of $d \times q$) dimensional space in which the maximization can be carried out more efficiently since $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q-1)}$ are fixed. The maximization of q-EHI$_{\text{async}}$ returns the design which is optimal given that $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q-1)}$ are currently being evaluated. In essence, it resembles to the Kriging Believer strategy defined in the next section, but requires GP simulations due to the correlation between $\mathbf{x}, \mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q-1)}$.

Kriging Believer Strategy

In Horn et al. (2015), a Kriging Believer (KB) strategy was proposed as multi-point multi-objective infill criteria. During one q-EHI-KB iteration, the (one-point) EHI maximizer is sought and the metamodel is virtually updated at this location using the kriging predictions as an emulator for $\mathbf{f}(\cdot)$. This sets the kriging variance of the latter to 0 so that it will no longer be promoted by the EHI. A batch of q designs to be evaluated by the true functions is generated by repeating this procedure q times successively. Compared with the other criteria, the Kriging Believer strategy has the advantage of maximizing the EHI criterion in dimension d. Moreover, it does not require Monte Carlo simulations[3] since the expectation of the GP is only considered at one $\mathbf{x}$ contrarily to the previous q-EHI variants, in which conditional GP simulations were required to take the correlation between the q designs into account. The drawback of the q-EHI-KB is that it heavily depends on the metamodel since each EHI maximizer is virtually incorporated together with its kriging prediction. This may be a drawback for functions that are weakly approximated by GPs and/or when the number of observations is too small to have an accurate surrogate.

5.1.2.4 Complexity comparison

Apart from the evaluation of $\mathbf{f}(\cdot)$, the main complexity of the algorithm resides in q-EHI's evaluation and maximization. The cost of the kriging model computation or update is upper bounded by the cost of inverting an $n \times n$ matrix. This cost is usually of a magnitude smaller than the computation and optimization of hypervolume-based infill criteria, especially for more than 2 objectives. The cost of simulating the GPs at q locations when Monte Carlo methods are needed to approximate the criterion requires one Cholesky decomposition of a matrix of small size $q \times q$, and one matrix product with matrices of respective sizes $q \times q$ and $q \times N_{MC}$. This is also negligible compared with the computation of the hypervolume of all N_{MC} simulated Pareto fronts to be averaged. In variants where the metamodels need to be updated and/or called within the maximization (as for q-EHI-KB or the asynchronous q-EHI) these computation times are also neglected.

Therefore, we compare the complexities of the multi-point EHI variants in terms of hypervolume computations. N_{MC} GPs are simulated when the criterion has to be estimated using Monte Carlo techniques, and d is the dimension of the problem (dimension of the $f_j(\cdot)$'s). We consider EHI to be analytical, even though it also requires Monte Carlo simulations when $m > 3$ in our implementations. $\tilde{d}$ is the dimension of the infill criterion, $\tilde{d} = d$ for EHI's maximization but $\tilde{d} = q \times d$ for q-EHI. Notice that there is a symmetry in the $q \times d$ dimensional q-EHI space since q-EHI$(\{\mathbf{x}, \mathbf{x}'\})$ = q-EHI$(\{\mathbf{x}', \mathbf{x}\})$ which might be exploited for enhancing the criterion's maximization.

As mentioned, the criterion is maximized with a genetic algorithm which evaluates the criterion $\alpha\tilde{d}$ times (Section 5.1.2.2).

- EHI's maximization requires $\alpha\tilde{d} = \alpha d$ calls to EHI. It is analytically known and does not require N_{MC} hypervolume computations.

- q-EHI's maximization requires $\alpha\tilde{d} = \alpha dq$ calls to q-EHI. The latter requires N_{MC} GP simulations and hypervolume computations. The total cost is therefore $\alpha dq N_{MC}$.

- q-EHI$_{\max}$'s maximization requires $\alpha\tilde{d} = \alpha dq$ calls to q-EHI$_{\max}$. The latter requires N_{MC} GP simulations. As the hypervolume contribution of each of the q points is

[3] For $m > 3$ the EHI computation in `GPareto` (Binois and Picheny, 2015) requires Monte Carlo methods (Emmerich et al., 2006). An analytical expression for EHI has nonetheless been found recently for any m in (Yang et al., 2019a).

considered, it requires qN_{MC} hypervolume computations per q-EHI$_{\text{max}}$ evaluation. Hence, the total cost is $\alpha\tilde{d}qN_{MC} = \alpha dq^2N_{MC}$.

- EHI$_{\text{async}}$'s maximization is equivalent to q successive q-EHI maximizations where $q-1$ variables are freezed, hence $\tilde{d} = d$. It however requires Monte Carlo estimations. The total cost is αdqN_{MC}.
- q-EHI-KB's maximization is equivalent to q successive EHI maximizations per iteration. It does not require simulations. The total cost is therefore $\alpha\tilde{d}q = \alpha dq$.

The following table summarizes the normalized number of hypervolume computations during one criterion maximization. EHI is of course the cheapest criterion but q-EHI-KB is only q times more expensive. q-EHI and q-EHI$_{\text{async}}$ are much more expensive because they rely on Monte Carlo simulations. q-EHI$_{\text{max}}$ is the most cumbersome criterion. When EHI needs to be estimated by simulation (i.e. when $m > 3$ in the current implementation of `GPareto` (Binois and Picheny, 2015), an extra N_{MC} term has to be added to EHI and to EHI-KB. Since $q \ll N_{MC}$, q-EHI, q-EHI$_{\text{max}}$ and q-EHI$_{\text{async}}$ are less dramatically expensive than EHI and q-EHI-KB in such cases. However, as a formula for EHI has recently been discovered for any m by Yang et al. (2019a), in the future, it may no longer be necessary to resort to Monte Carlo simulations for estimating the single-point criterion, for any m.

Criterion	Cost
EHI	1
q-EHI	qN_{MC}
q-EHI$_{\text{max}}$	q^2N_{MC}
q-EHI-KB	q
q-EHI$_{\text{async}}$	qN_{MC}

Table 5.6: Relative cost of the criteria maximization in terms of hypervolume computations.

5.1.2.5 Experiments

We now compare the performance of q-EHI$_{\text{max}}$ (5.5) and q-EHI (5.6), as well as the sequential EHI on two well-known test functions. The first one is the P1 problem (Parr, 2013) and the second one the ZDT1 problem (Zitzler et al., 2000). Both have $d = 2$ dimensions and $m = 2$ objectives which is ideal for optimizing the criterion and analysis of the results. As we are solely interested in the behavior of the multi-point algorithms and not in any artifact related to the reference point, we fix $\mathbf{R} = (132.7, -21.1)^\top$ and $\mathbf{R} = (1, 1)^\top$ for P1 and ZDT1 respectively, which is the Nadir point of each problem. Experiments are initialized with a space-filling DoE of $n = 8$ designs.

The third benchmark is the MetaNACA test bed (Chapter 3). We use the problem with $d = 8$ dimensions and $m = 2$ objectives, and an initial space-filling DoE of $n = 20$ observations. Even though they are the mean predictor of a Gaussian Process, the

MetaNACA's objective functions are slightly less regular than the ones considered in P1 and ZDT1.

The additional multi-point criteria introduced in Section 5.1.2.3 (namely q-EHI$_{\text{async}}$ and q-EHI-KB) are also benchmarked for further comparison. All experiments are run for 12 additional iterations and carried out for batches of size $q = 2$ or $q = 4$. The final approximation fronts obtained by the multi-point EHI's or by the sequential EHI are compared at the same number of function evaluations, or at the same wall-clock time (during which q more function evaluations can be carried out assuming that q-EHI's maximization time is negligible compared with $\mathbf{f}(\cdot)$'s evaluation). The chosen metric for analyzing the results is the hypervolume indicator (Zitzler, 1999) computed up to $\mathbf{R}$. Table 5.7 reports the mean of this metric obtained by the sequential EHI. Table 5.8 compares the performance of q-EHI and q-EHI$_{\text{max}}$ with $q = 2$ infills, and with $q = 4$ in Table 5.9. Tables 5.10 and 5.11 record the performance of EHI$_{\text{async}}$ and EHI-KB respectively, both for $q = 2$ and $q = 4$. To facilitate comparison at different number of function evaluations or at different wall-clock times, the hypervolume indicator is computed at different moments during the optimization. Figure 5.6 visually compares the hypervolume indicator for the different infill criteria, at the same wall clock time (left column) and with regard to the number of function evaluations (right column).

Criterion	EHI		
$\#\mathbf{f}(\cdot)$	12	6	3
P1	0.913 *(0.029)*	0.789 *(0.068)*	0.635 *(0.107)*
ZDT1	0.939 *(0.003)*	0.885 *(0.007)*	0.786 *(0.017)*
NACA	0.748 *(0.067)*	0.644 *(0.094)*	0.587 *(0.085)*

Table 5.7: Hypervolume indicator obtained by EHI for different computational budgets and benchmarks. Averages *(std. deviation)* over 10 runs.

Criterion	2-EHI		2-EHI$_{\text{max}}$	
$\#\mathbf{f}(\cdot)$	2×12	2×6	2×12	2×6
P1	0.963 *(0.007)*	0.898 *(0.036)*	0.949 *(0.008)*	0.882 *(0.026)*
ZDT1	0.970 *(0.001)*	0.939 *(0.002)*	0.961 *(0.003)*	0.926 *(0.006)*
NACA	0.831 *(0.046)*	0.718 *(0.093)*	0.807 *(0.046)*	0.725 *(0.077)*

Table 5.8: Hypervolume indicator obtained by q-EHI for different computational budgets and benchmarks with $q = 2$. Averages *(std. deviation)* over 10 runs.

These results indicate that at the same number of function evaluations ($p = 12$), EHI performs slightly better than all multi-point q-EHI variants. However when $\mathbf{f}(\cdot)$ is expensive to evaluate, which is a common assumption in Bayesian optimization, and is much more costly than maximizing the infill criterion[4], it is worth comparing the criteria

[4] This assumption clearly depends on the evaluation time of $\mathbf{f}(\cdot)$ since some criteria are no longer

Criterion	4-EHI		$4\text{-EHI}_{\text{max}}$	
$\#\mathbf{f}(\cdot)$	4×12	4×3	4×12	4×3
P1	0.984 *(0.002)*	0.856 *(0.054)*	0.964 *(0.005)*	0.817 *(0.063)*
ZDT1	0.980 *(0.001)*	0.934 *(0.004)*	0.960 *(0.004)*	0.883 *(0.012)*
NACA	0.866 *(0.026)*	0.705 *(0.065)*	0.845 *(0.018)*	0.712 *(0.053)*

Table 5.9: Hypervolume indicator obtained by q-EHI for different computational budgets and benchmarks with $q = 4$. Averages *(std. deviation)* over 10 runs.

Criterion	$2\text{-EHI}_{\text{async}}$		$4\text{-EHI}_{\text{async}}$	
$\#\mathbf{f}(\cdot)$	2×12	2×6	4×12	4×3
P1	0.963 *(0.009)*	0.893 *(0.048)*	0.983 *(0.002)*	0.852 *(0.061)*
ZDT1	0.970 *(0.001)*	0.940 *(0.003)*	0.983 *(0.001)*	0.933 *(0.003)*
NACA	0.841 *(0.038)*	0.744 *(0.064)*	0.887 *(0.025)*	0.714 *(0.049)*

Table 5.10: Hypervolume indicator obtained by $\text{q-EHI}_{\text{async}}$ for different computational budgets and benchmarks with $q = 2$ and $q = 4$. Averages *(std. deviation)* over 10 runs.

Criterion	2-EHI-KB		4-EHI-KB	
$\#\mathbf{f}(\cdot)$	2×12	2×6	4×12	4×3
P1	0.965 *(0.006)*	0.908 *(0.038)*	0.972 *(0.005)*	0.860 *(0.059)*
ZDT1	0.970 *(0.001)*	0.939 *(0.003)*	0.985 *(0)*	0.941 *(0.004)*
NACA	0.830 *(0.031)*	0.715 *(0.097)*	0.897 *(0.028)*	0.703 *(0.093)*

Table 5.11: Hypervolume indicator obtained by q-EHI-KB for different computational budgets and benchmarks with $q = 2$ and $q = 4$. Averages *(std. deviation)* over 10 runs.

at the same number of iterations. In this setting, q times more function evaluations can be obtained by the parallel criteria, and the hypervolume indicator reports a better uncovering of the Pareto front at the same wall clock time (i.e. the number of iterations). At the same number of function evaluations (2×6 or 4×3) the q-EHI variants with $q = 2$ outperform those where $q = 4$ which is explained by the higher number of metamodel updates (6 against 3). But at the same number of iterations (12), the variants which evaluate batches of $q = 4$ designs per iteration lead to a higher hypervolume than with $q = 2$, because twice more designs have been evaluated.

Even though being the natural extension of q-EI, the $\text{q-EHI}_{\text{max}}$ criterion behaves worse than q-EHI in all experiments and for all budgets (except in the NACA case in the earlier iterations, i.e., 2×6 or 4×3). This is explained by the fact that the q points are driven towards the same part of the Pareto front, since each design aims at individually leading to the largest hypervolume growth instead of collaborating to achieve a well-distributed

"easy to be maximized". Here, the additional cost of maximizing the criterion is neglected in regard of $\mathbf{f}(\cdot)$'s evaluations, and comparisons at "the same wall-clock time" correspond to the same number of iterations, during which q evaluations are carried out.

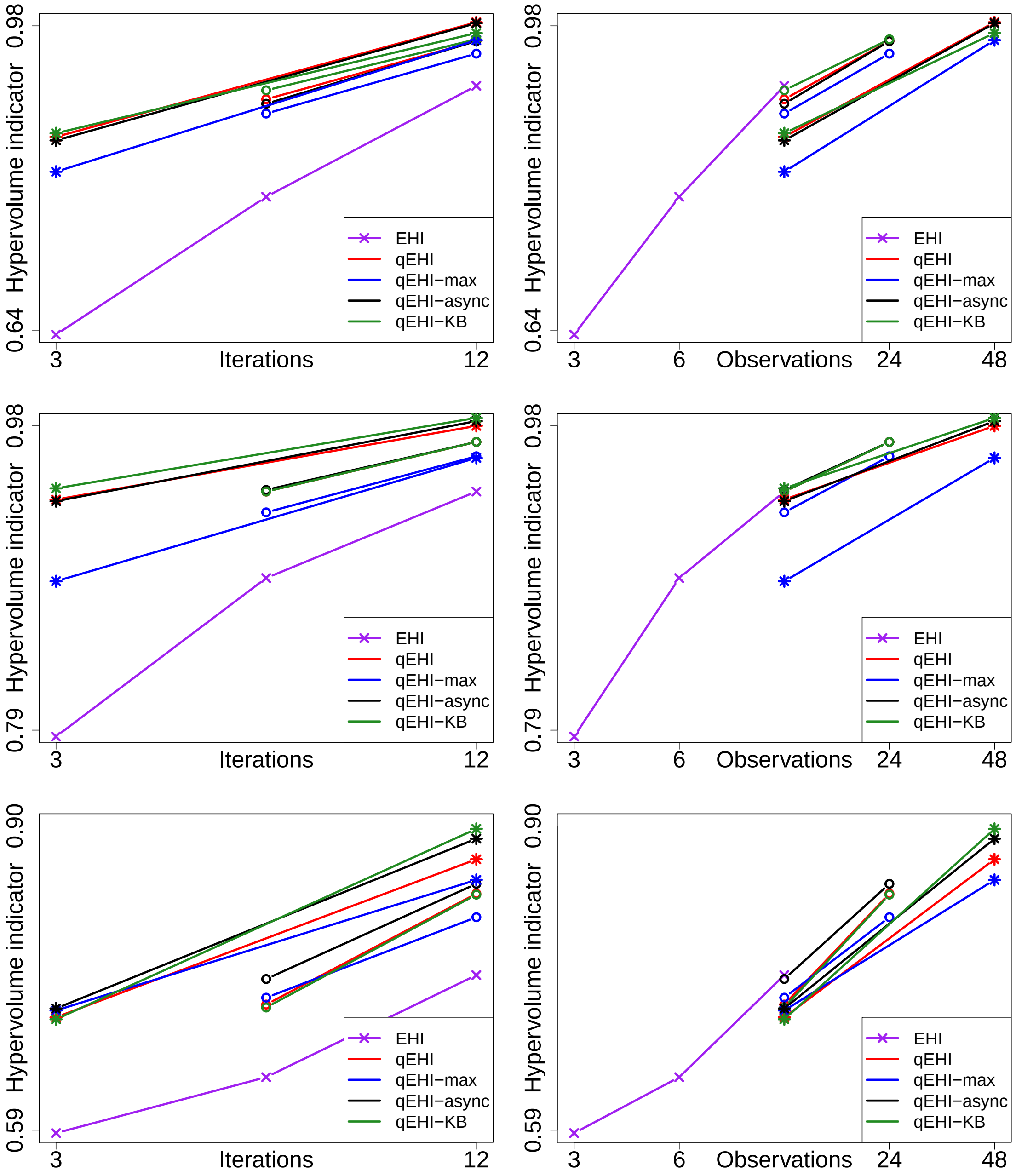


Figure 5.6: Hypervolume indicator comparison for the different infill criteria, with respect to the number of iterations (left column) or the number of function evaluations (right column). Curves with ∘ correspond to $q = 2$ and curves with * to $q = 4$. Top row: P1, middle row: ZDT1, bottom row: NACA.

and larger increase in hypervolume.

Besides from its practical interest, the asynchronous version of the q-EHI criterion, q-EHI$_{\text{async}}$, behaves even slightly better than q-EHI. This might be due to the easier optimization of the criterion, carried out in a q times lower dimensional space since only one design is sought given the $q-1$ pending evaluations.

Surprisingly, the Kriging Believer strategy in which EHI maximizers are virtually appended to the metamodel using the kriging mean predictor to obtain a q-points batch behaves as well or even better than the "truly multi-point" criteria, q-EHI and q-EHI$_{\text{async}}$. Maybe the three considered benchmarks are too regular and easy to be learned by a metamodel. Performance might be degraded in cases where believing in the kriging predictor is questionable. Nonetheless, this variant is the cheapest multi-point criterion, since it does not require the averaging of hypervolumes and its maximization is carried out in a d dimensional space, which makes it an attractive criterion.

5.1.3 Concluding remarks

In this section, we have considered the problem of finding a batch of q points where to evaluate the functions of a multi-objective problem in parallel. We have derived an optimal q points criterion for multi-objective optimization based on the mEI and on the hypervolume measure, called q-mEI and q-EHI, respectively. Assuming the functions to be optimized are expensive but can be computed in parallel, q-mEI and q-EHI attain faster the region of interest and achieve better convergence towards the Pareto front than their single-point counterparts, mEI and EHI, in wall-clock time.

Being computed by averaging the improvement of GP simulations, q-mEI and especially q-EHI's calculation and maximization may be expensive. Variants that come with an easier maximization and/or a reduced cost have also been investigated in this section. The first one is q-EHI$_{\text{async}}$, an asynchronous version of q-EHI, which looks for the optimal design where to start a new experiment while a batch of $q-1$ designs is currently evaluated. The original aim of this criterion resides in its practical interest: in case the objective functions $\mathbf{f}(\cdot)$ have heterogeneous running times depending on the design $\mathbf{x}$, it avoids waiting for the completion of all q simulations before launching a new batch; a new experiment is started as soon as a resource gets available instead. It is further computationally attractive: the maximization of this greedy acquisition function is more tractable by reason of the dimension reduction of the criterion input space (d instead of $q \times d$). In the reported experiments, q-EHI$_{\text{async}}$ has shown to perform comparably to q-EHI.

A second less expensive q points multi-objective criterion, hinging on a Kriging Believer strategy, q-mEI-KB or q-EHI-KB, respectively, has also been investigated. It consists in q combined maximizations of mEI (respectively EHI) and emulations using the kriging mean predictor to return a batch of q designs. It does not require expensive Monte Carlo methods and operates in a space of smaller dimension, d, hence it can be maximized faster. If available, the gradient of the infill criterion is also employed easing its maximization. Surprisingly, this cheaper alternative which blindly believes in the surrogate model has also shown comparable performance with q-EHI on the three benchmark problems studied in this section. In the case of mEI, its performance was comparable to that of q-mEI too.

The high-dimensional q-mEI and q-EHI maximization may be helped by combining the optimizer with the gradient of the infill criterion. q-mEI and q-EHI are not known in closed form and neither are their gradients. Reparameterization tricks (Kingma and Welling, 2014) nevertheless constitute a promising direction to obtain a proxy for q-mEI and q-EHI's gradient stemming from the Monte Carlo sample (Wang et al., 2016; Wilson et al., 2018; Wu and Frazier, 2016). This technique has led to a more efficient maximization of the q-EI criterion in Frazier and Clark (2012); Marmin et al. (2015), and may accelerate q-mEI and q-EHI's maximization. A better criterion maximizer might be found, which may further improve q-mEI and q-EHI's performance.

Reducing q-EHI's complexity remains a main challenge in future research since it would allow a fast maximization of the criterion and its use in moderately-expensive problems. Similarly to Yang et al. (2019c), searching the mEI maximizer in q regions of the objective space is a different way towards batch Bayesian multi-objective optimization, at a lower computational burden, further discussed in Section 5.3.3.

5.2 Constraints in Bayesian Multi-Objective Optimization

Building independent surrogate models $G_1(\cdot), \dots, G_{m_c}(\cdot)$ for the m_c constraints and incorporating this knowledge within the infill criterion is the most common way to handle constraints in Bayesian optimization, see Section 2.4, and straightforwardly extends to constrained multi-objective problems. In this section, we explain the modifications brought to the C-EHI/R-EHI algorithm such that each of its steps considers the fulfillment of constraints. We consider multi-objective problems with m_c constraints $(g_1(\cdot), \dots, g_{m_c}(\cdot))^\top =: \mathbf{g}(\cdot)$,

$$\min_{\substack{\mathbf{x}\in X \\ g_1(\mathbf{x})\leq 0 \\ \vdots \\ g_{m_c}(\mathbf{x})\leq 0}} \ (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \tag{5.9}$$

The "$\preceq$" ordering is extended to account for the feasibility of designs (Feliot, 2017; Feliot et al., 2017), called constrained Pareto dominance.

Definition 5.1. *(Constrained Pareto dominance). Denoting* $\mathbf{a} = [\mathbf{a^o}, \mathbf{a^c}] \in \mathbb{R}^{m+m_c}$ *the augmented vector (where* $\mathbf{a^o}$ *stand for the objective values and* $\mathbf{a^c}$ *for the constraint values)* $\mathbf{a} \preceq \mathbf{b}$ *in one of the following cases:*

- $\mathbf{a}$ *is feasible and* $\mathbf{b}$ *is not feasible (i.e.* $\mathbf{a^c} \preceq \mathbf{0}_{m_c}$[5] *and* $\mathbf{b^c} \npreceq \mathbf{0}_{m_c}$*);*
- $\mathbf{a}$ *and* $\mathbf{b}$ *are feasible and* $\mathbf{a^o} \preceq \mathbf{b^o}$*;*
- $\mathbf{a}$ *and* $\mathbf{b}$ *are not feasible but* $(\mathbf{a^c})_+ \preceq (\mathbf{b^c})_+$*.*

[5] By a minor abuse of notations, we write $\mathbf{a^c} \preceq \mathbf{0}_{m_c}$ even though no strict inequality $a_i^{\mathbf{c}} < 0$ is required here.

With this extended rule, feasible designs always dominate unfeasible ones regardless of the objective values, and the Pareto dominance in the space of constraints is employed for comparing non-feasible solutions, to quantify which one is the least worst. In this case, instead of the brute constraint value, the amount by which $g_j(\cdot)$ is violated (0 when the constraint is satisfied) is considered through the $(\cdot)_+$ operator to not compare the degree of constraint satisfaction. Remark that the number of violated constraints is not taken into account for domination in the constraint space. For instance, if a solution $\mathbf{a}$ violates $m_c - 1$ constraints and $\mathbf{b}$ does not satisfy the remaining one, $\mathbf{a}$ and $\mathbf{b}$ are incomparable even though one might wish $\mathbf{b} \preceq \mathbf{a}$. Figure 5.7 shows a front with constrained Pareto dominance. $\mathcal{P}_\mathcal{Y}$ (black stairs) is the set of non dominated points among the feasible designs (black dots). Even though some of them are non-dominated in the (f_1, f_2) space, perhaps dominating some solutions, non feasible solutions (gray dots) are filtered out for constrained Pareto dominance.

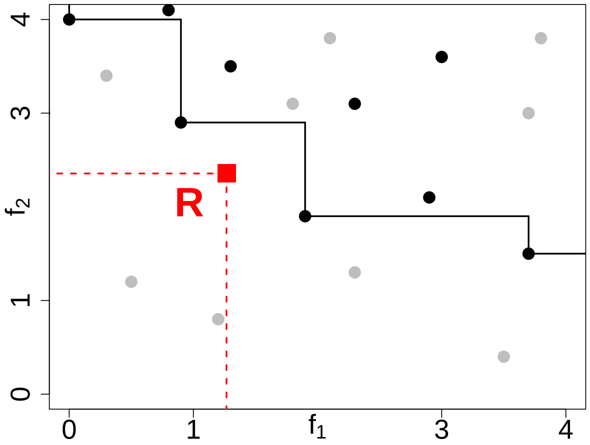


Figure 5.7: Illustration of constrained Pareto dominance: $\mathcal{P}_\mathcal{Y}$ is the non-dominated set among the feasible (black dots) designs. Non feasible designs are in gray.

5.2.1 C-EHI/R-EHI adjustments to cope with constraints

Assuming feasible designs exist[6], the Pareto set of (5.9) is $\mathcal{P}_\mathcal{X} = \{\mathbf{x} \in X : \mathbf{g}(\mathbf{x}) \preceq \mathbf{0}_{m_c}, \nexists \mathbf{x}' \in X, \mathbf{g}(\mathbf{x}') \preceq \mathbf{0}_{m_c}, \mathbf{f}(\mathbf{x}') \preceq \mathbf{f}(\mathbf{x})\}$ and its Pareto front is $\mathcal{P}_\mathcal{Y} = \mathbf{f}(\mathcal{P}_\mathcal{X})$. The empirical front $\widehat{\mathcal{P}_\mathcal{Y}} \subset \mathbb{R}^m$ is the non-dominated set among the observations $\mathbf{y}^{(1:t)^\mathbf{o}} = \{\mathbf{y}^{(1)^\mathbf{o}}, \dots, \mathbf{y}^{(t)^\mathbf{o}}\} \subset \mathbb{R}^m$ which comply with the constraints. In this part we assume at least one observed design is feasible, i.e. $\exists \mathbf{y} \in \mathbf{y}^{(1:t)^\mathbf{c}} : \mathbf{y}^\mathbf{c} \preceq \mathbf{0}_{m_c}$.

[6] Otherwise, the Pareto set/front would be the Pareto set/front in the space of constraints.

5.2.1.1 Updated target

Once a target $\mathbf{R}$ is provided, the R-EHI algorithm (Chapter 4) builds an updated reference $\widehat{\mathbf{R}}$ point on the $\mathbf{IRN}$ line. $\mathbf{I}$ and $\mathbf{N}$ are the Ideal and Nadir point of $\mathcal{P}_{\mathcal{Y}}$ which now restricts to feasible design. $\mathbf{I}$ and $\mathbf{N}$ are unknown and estimated by $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}}$ using samples of the Gaussian Processes (Section 4.4.2.3). The location in X where the GPs are simulated is critical and in Chapter 4, they were performed at $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+s)}$ with largest probabilities of contributing to one component of $\mathbf{I}$ or $\mathbf{N}$.

To account for feasibility, the density of this importance sampling procedure is multiplied by the probability of satisfying the constraints, $\text{PoF}(\mathbf{x})$ (2.12), which is a product of probabilities of improvement over 0 of the $G_j(\cdot)$'s. n_{sim} GPs are simulated at $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+s)}$ and the resulting fronts are filtered by constraint satisfaction. As in Section 4.4.2.3, $\widehat{\mathbf{I}}$ and $\widehat{\mathbf{N}}$ are the medians of the Ideal and Nadir points of the n_{sim} fronts, and finally, the updated reference point $\widehat{\mathbf{R}}$ is the closest point on $\widehat{\mathcal{L}}' = \widehat{\mathbf{I}}\mathbf{R}\widehat{\mathbf{N}}$ to $\widehat{\mathcal{P}_{\mathcal{Y}}}$ (or on $\widehat{\mathcal{L}} = \widehat{\mathbf{I}}\widehat{\mathbf{N}}$ if no preference is expressed).

5.2.1.2 Convergence detection

Similarly, the convergence detection relies on simulations of the $Y_1(\cdot), \dots, Y_m(\cdot)$ GPs. At this step, $\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+s)}$ are selected according to their probability of being non-dominated in the objective space, $\mathbb{P}(\widehat{\mathcal{P}_{\mathcal{Y}}} \not\preceq \mathbf{Y}(\mathbf{x}))$. The latter is multiplied by $\text{PoF}(\mathbf{x})$ to account for the satisfaction of constraints and the resulting simulated fronts are filtered by the extended Pareto rule. The uncertainty along $\widehat{\mathcal{L}}'$ or $\widehat{\mathcal{L}}$ is computed as in (4.7), after the simulated fronts have been filtered to remove any non-feasible simulated value.

5.2.1.3 Widening of the approximation front

Once convergence is detected, an optimal reference point $\mathbf{R}^* \in \widehat{\mathcal{L}}'$ is sought, by anticipating the behavior of the algorithm and more specifically of the EHI infill criterion during the remaining b iterations, via a Kriging Believer strategy (Section 4.6). The same logic is followed except that a modified EHI (detailed in Section 5.2.1.4), EHIPF, which accounts for the satisfaction of constraints is employed. The virtual fronts are filtered to remove any non feasible (according to the kriging prediction because of the KB strategy) design. The selection of $\mathbf{x}$'s where the final virtual GPs are simulated takes the satisfaction of constraints into account through $\mathbb{P}(\widehat{\mathcal{P}_{\mathcal{Y}}} \not\preceq \mathbf{Y}(\mathbf{x})) \times \text{PoF}(\mathbf{x})$. These simulated fronts are filtered by the extended Pareto rule too to quantify the uncertainty on the associated final virtual front.

5.2.1.4 Modified infill criteria

The most straightforward way to consider constraints in Bayesian optimization is to make the infill criterion account for the satisfaction of constraints. In mono-objective problems, the EI (or any other acquisition function) is multiplied by PoF (2.12, Schonlau, 1997). This is the adopted approach. We define the mEIPF and EHIPF,

$$\mathrm{mEIPF}(\mathbf{x};\mathbf{R}) = \mathrm{mEI}(\mathbf{x};\mathbf{R}) \times \mathrm{PoF}(\mathbf{x}) \tag{5.10}$$

$$\mathrm{EHIPF}(\mathbf{x};\mathbf{R}) = \mathrm{EHI}(\mathbf{x};\mathbf{R}) \times \mathrm{PoF}(\mathbf{x}) \tag{5.11}$$

where the reference point $\mathbf{R} \in \mathbb{R}^m$ is defined in the objective space, Y. Under the hypothesis of independence between the $Y_j(\cdot)$'s and the $G_j(\cdot)$'s, (5.10) and (5.11) are the expectation of constrained utility functions which equal 0 for non-feasible designs, namely $\prod_{j=1}^{m}(R_j - y_j)_+ \mathbb{1}_{g_1 \leq 0, \dots, g_{m_c} \leq 0}$ and $\underbrace{(I_H(\widehat{\mathcal{P}_\mathcal{Y}} \cup \{\mathbf{y}\};\mathbf{R}) - I_H(\widehat{\mathcal{P}_\mathcal{Y}};\mathbf{R}))}_{I(\mathbf{y};\mathbf{R})}\mathbb{1}_{g_1 \leq 0, \dots, g_{m_c} \leq 0}$.

An alternative is to maximize mEI and EHI with a constraint β_p on PoF (Sacher et al., 2018). mEIPF's gradient has closed form expression since both ∇mEI and ∇PoF have analytical gradients (see Equation 2.12, Equation 4.6, and Roustant et al., 2012), which is advantageous for the inner-loop optimization.

As shown in Figure 5.7, regarding only the objective space, $\mathbf{R}$ might be dominated by some non-feasible $\mathbf{y}$'s. mEIPF both promotes designs with large probability of being feasible which potentially dominate $\mathbf{R}$ and designs with large $\mathbf{R}$ domination and which might be feasible.

Modified batch infill criteria

In the context of parallel evaluations of $\mathbf{f}(\cdot)$ at a batch $\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}$, both q-mEI and q-EHI are estimated by means of Monte Carlo simulations. The Monte Carlo estimation of the constrained versions of the constrained q-mEI solely requires the indicator function of feasibility, as in the corresponding utility function:

$$\text{q-}\widehat{\mathrm{mEIP}}\mathrm{F}(\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\};\mathbf{R}) = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \left[\max_{i=1,\dots,q} \left(\prod_{j=1}^{m} (R_j - \widetilde{Y_j}^{(k)}(\mathbf{x}^{(t+i)}))_+ \prod_{l=1}^{m_c} \mathbb{1}_{\widetilde{G_l}^{(k)}(\mathbf{x}^{(t+i)}) \leq 0} \right) \right] \tag{5.12}$$

The constrained q-EHI rewards the hypervolume increase of feasible simulated values only,

$$\text{q-}\widehat{\mathrm{EHIP}}\mathrm{F}(\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\};\mathbf{R}) = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \left(I_H(\widehat{\mathcal{P}_\mathcal{Y}} \bigcup_{\substack{i=1 \\ \widetilde{\mathbf{G}}^{(k)}(\mathbf{x}^{(t+i)}) \preceq \mathbf{0}_{m_c}}}^{q} \{\widetilde{\mathbf{Y}}^{(k)}(\mathbf{x}^{(t+i)})\};\mathbf{R}) - I_H(\widehat{\mathcal{P}_\mathcal{Y}};\mathbf{R}) \right) \tag{5.13}$$

where $\widetilde{G_l}^{(k)}(\mathbf{x})$ is the k-th simulated constraint value at $\mathbf{x}$ of $G_l(\cdot)$ ($\widetilde{\mathbf{G}}^{(k)}(\mathbf{x}) = (\widetilde{G_1}^{(k)}(\mathbf{x}), \dots, \widetilde{G_{m_c}}^{(k)}(\mathbf{x}))^\top$) and $\widetilde{\mathbf{Y}}^{(k)}(\mathbf{x}) = (\widetilde{\mathbf{Y}_1}^{(k)}(\mathbf{x}), \dots, \widetilde{\mathbf{Y}_m}^{(k)}(\mathbf{x}))^\top$ the k-th simulated (m dimensional) objective value at $\mathbf{x}$ (joint simulations over $\{\mathbf{x}^{(t+1)}, \dots, \mathbf{x}^{(t+q)}\}$).

Constraints incorporation in mEI and EHI via the reference point

A way of handling constraints is to consider them inside a multi-objective problem (Knowles et al., 2001; Mezura-Montes and Coello, 2006; Saxena and Deb, 2007; Segura et al., 2016). The multi-objectivization of (5.9),

$$\min_{\mathbf{x}\in X}(f_1(\mathbf{x}),\dots,f_m(\mathbf{x}),g_1(\mathbf{x}),\dots,g_{m_c}(\mathbf{x})) \tag{5.14}$$

can therefore be considered. This problem can be of interest if one is not uniquely interested the satisfaction of $g_j(\cdot)\leq 0$, but also in an optimization of the $g_j(\cdot)$'s. The advantage of mEI and EHI for (5.14) resides in the targeting ability of the reference point. Here, in the "objectives" $m+1$ to $m+m_c$, a reference value is already known, $\mathbf{R^c}=\mathbf{0}_{m_c}$. One can therefore think of solving (5.14) using mEI or EHI with an augmented reference point $\mathbb{R}^{m+m_c}\ni\mathbf{R}^{aug}:=[\mathbf{R},\mathbf{0}_{m_c}]$, for targeting $\mathbf{R}$ in the objective space, and satisfying $\mathbf{g}(\cdot)\preceq\mathbf{0}_{m_c}$.

In the spirit of the R-EHI algorithm, instead of employing the potentially difficult $\mathbf{0}_{m_c}$, the reference point for $g_1(\cdot),\dots,g_{m_c}(\cdot)$ may be adapted, e.g. by using an $\mathbf{R^c}$ different from $\mathbf{0}_{m_c}$ inside $\mathbb{R}^{m+m_c}\ni\mathbf{R}^{aug}=[\mathbf{R},\mathbf{R^c}]$. $\mathbf{R^c}$ is computed through the current approximation front $\widehat{\mathcal{P}_\mathcal{Y}}$ (with constrained Pareto dominance) to accommodate observed feasible values, as will be shown in the next section (Figure 5.8). The main risk of (5.14) is to put too much emphasis on the constraints if $\mathbf{R^c}$ is not managed accordingly.

Substituting the $g_j(\cdot)$'s in (5.14) by their indicator function $\mathbb{1}_{g_j(\cdot)\leq 0}$ and solving the augmented problem using mEI or EHI together with $\mathbf{R}^{aug}=[\mathbf{R},\mathbf{0}_{m_c}]$ is conceptually equivalent to solving (5.9) via mEIPF$(\cdot;\mathbf{R})$ or EHIPF$(\cdot;\mathbf{R})$ and refers to the use of binary classifiers for handling constraints (Basudhar et al., 2012; Sacher et al., 2018).

5.2.1.5 Experiments: attaining the center of the Pareto front in constrained problems

In this part, the capability of the mEIPF criterion to attain the central part of constrained Pareto fronts are highlighted. mEIPF is compared with four other infill criteria: EHIPF, which does not target specifically the center of the Pareto front. mEIC, which considers Problem (5.14) and where the mEI criterion operates in an $m+m_c$-dimensional objective space. It accounts for the constraints by employing the $m+m_c$ metamodels with the reference point $\mathbf{R}^{aug}=[\mathbf{R},\mathbf{0}_{m_c}]$ augmented by zeros in the "objectives" that correspond to the $g_j(\cdot)$'s. Last, the performance of the standard mEI and EHI which do not take the constraints into account[7] is also investigated. Five popular multi-objective constrained problems are used for comparison: BNH (Deb, 2001), Two Bar, Welded Beam (Chafekar et al., 2003), Constr (Zitzler et al., 2000) and Water (Ray et al., 2001). The dimension, number of objectives and number of constraints of these problems are given in Table 5.12. The experiments start with a space-filling DoE of $n=3d$ observations and are run for $p=3d$ supplementary iterations.

[7] they however consider an improvement over the center of the constrained front, or the hypervolume improvement with respect to the constrained front, i.e. after having removed non feasible designs.

Problem	d	m	m_c
BNH	2	2	2
Two Bar	3	2	1
Welded Beam	4	2	4
Constr	2	2	2
Water	3	5	7

Table 5.12: Dimension, number of objectives and of constraints of the constrained multi-objective problems.

Three performance metrics are used for comparing the infill criteria. The first one is the restriction of the hypervolume indicator I_H to a central part of the Pareto front, $\mathcal{I}_{0.1}$, dominated by $\mathbf{R}_{0.1}$ (4.10), which indicates convergence to this part of $\mathcal{P}_\mathcal{Y}$. Only feasible designs are considered in $\mathcal{I}_{0.1}$. Depending on the objectives and on the constraints, $\mathcal{I}_{0.1}$ may be very hard to attain within the restricted budget of $p = 3d$ iterations. This indicator is normalized by the hypervolume of the true Pareto front restricted to $\mathcal{I}_{0.1}$. The second metric is the attainment time of $\mathcal{I}_{0.1}$, measured by the number of function evaluations (including the initial DoE) to dominate $\mathbf{R}_{0.1}$. It indicates how fast the algorithms attain this central part of $\mathcal{P}_\mathcal{Y}$. Last, the number of feasible designs found at the end of the search is investigated too.

The averaged metrics over 10 runs are reported in Table 5.13 with standard deviation in brackets. When at least one run did not attain $\mathcal{I}_{0.1}$, an estimator of the empirical runtime (Auger and Hansen, 2005) is given in red, the number of successful runs being reported in brackets. × indicates that no run was able to access $\mathcal{I}_{0.1}$. In each run, there was at least one feasible design in the initial DoE.

	Hypervolume in $\mathcal{I}_{0.1}$		$\mathcal{I}_{0.1}$ attainment time		Feasible designs	
	mEIPF	EHIPF	mEIPF	EHIPF	mEIPF	EHIPF
BNH	0.596 (0.050)	0.537 (0.090)	6.4 (0.5)	6.5 (0.7)	11.2 (0.6)	11.8 (0.4)
Two Bar	0.060 (0.092)	0.041 (0.060)	25.6 [4]	24.4 [4]	9.3 (1.9)	11.8 (2.5)
Welded Beam	0.026 (0.062)	0	87.5 [2]	×	7.8 (2.6)	10.8 (3.3)
Constr	0.293 (0.182)	0.366 (0.268)	8.9 [8]	10.2 [8]	7.0 (0.7)	8.7 (0.8)
Water	0.064 (0.107)	0.028 (0.087)	37.8 [3]	170.0 [1]	13.4 (1.1)	16.3 (0.7)

	Hypervolume in $\mathcal{I}_{0.1}$			$\mathcal{I}_{0.1}$ attainment time			Feasible designs		
	mEIC	mEI	EHI	mEIC	mEI	EHI	mEIC	mEI	EHI
BNH	0.583 (0.055)	0.603 (0.055)	0.535 (0.094)	6.4 (0.5)	6.4 (0.5)	6.5 (0.7)	11.6 (0.5)	11.5 (0.7)	11.8 (0.4)
Two Bar	0.112 (0.126)	0.034 (0.060)	0.034 (0.060)	18.3 [6]	30.0 [3]	30.0 [3]	9.1 (2.1)	7.9 (0.6)	14 (3)
Welded Beam	0	0	0	×	×	×	8.2 (2.7)	5.0 (1.2)	5.3 (1.5)
Constr	0.314 (0.129)	0.004 (0.011)	0.004 (0.011)	8.6 (1.3)	60.0 [1]	60.0 [1]	7.6 (0.8)	3.0 (0.7)	4.1 (1.2)
Water	0.086 (0.139)	0.031 (0.065)	0.011 (0.030)	38.9 [3]	60.0 [2]	70.0 [2]	14.4 (0.8)	13.7 (1.3)	15.7 (1.8)

Table 5.13: Performance metrics averaged over 10 runs (standard deviation in brackets) obtained by the five investigated infill criteria.

These results indicate that mEIPF is able to produce better convergence in $\mathcal{I}_{0.1}$ (even though the well spread approximation inside $\mathcal{I}_{0.1}$ is not the criterion's first aim) than EHIPF. Surprisingly, mEIC which handles the constraints through mEI's reference point performs well too, even though it also considers an improvement over the constraint threshold, $\mathbf{0}_{m_c}$. Its performance is comparable with that of mEIPF on these test functions.

mEIPF attains $\mathcal{I}_{0.1}$ faster and more consistently than EHIPF as measured by the attainment time. More solutions (not shown here) are also found inside $\mathcal{I}_{0.1}$ by mEIPF and mEIC than by EHIPF, mEI and EHI. It is worth noting that mEIPF and mEIC produce less feasible designs than EHIPF. Indeed, the EHI criterion values all designs that augment $\mathcal{P}_{\mathcal{Y}}$ whereas mEIPF and mEIC only promote designs which improve over $\widehat{\mathbf{C}}$ and $[\widehat{\mathbf{C}}, \mathbf{0}_{m_c}]$, respectively. Therefore, less candidates are relevant in mEI's logic than in EHI's one. The amount of such candidates that additionally comply with the constraints is further reduced which explains why mEIPF and mEIC find less feasible designs than EHIPF. Additionally, mEIC's satisfaction of constraints is treated by the more exploratory than PoF (Jones, 2001) EI which explains this difference. No significant difference is nonetheless observed here between mEIC and mEIPF for the number of found feasible designs.

When it is difficult to satisfy the constraints, mEI and EHI have the weakest performance. These criteria indeed consider solutions that improve over $\widehat{\mathbf{C}}$ or the hypervolume, respectively, as good, regardless of the satisfaction of the constraints. mEI nonetheless performs the best on the lightly constrained BNH problem and EHI behaves similarly to EHIPF on it. Both criteria find a large number of feasible solutions in this problem, even without considering the feasibility of designs. However, in more challenging problems, they face more difficulties to attain the central part of $\mathcal{P}_{\mathcal{Y}}$ and even to find feasible solutions. In Welded Beam for instance, 1.2 additional feasible designs have been found on average during the $p = 3d = 12$ additional iterations by EHI, and 0.9 by mEI.

5.2.2 mEI for severely constrained problems

mEI has proven to rapidly attain user desired objective values expressed through $\mathbf{R}$. In this part, the worth of mEI's targeting abilities are investigated for another type of problems: highly constrained (multi-objective) problems, where finding $\mathbf{x}$'s such that $\mathbf{g}(\mathbf{x}) \preceq \mathbf{0}_{m_c}$ is challenging. We consider the case where no feasible solution has been found, $\nexists \mathbf{y} \in \mathbf{y}^{(1:t)^{\mathbf{c}}} : \mathbf{y} \preceq \mathbf{0}_{m_c}$ (and $\widehat{\mathcal{P}_{\mathcal{Y}}} = \varnothing$). Constrained Pareto domination (Definition 5.1) boils down to Pareto domination in the constraint space[8] $G \subset \mathbb{R}^{m_c}$, regardless of the objective values, and the most urgent matter is to find a feasible designs, $\mathbf{y}^{\mathbf{c}} \preceq \mathbf{0}_{m_c}$: mEIPF and EHIPF are replaced by criteria aiming at finding such designs (remark that since $\widehat{\mathcal{P}_{\mathcal{Y}}} = \varnothing$, it is not possible to compute an updated goal $\widehat{\mathbf{R}}$, an estimated center $\widehat{\mathbf{C}}$, or a reference point in the objective space).

A first option to search for feasible designs is to evaluate PoF's maximizer (2.12). A second is the Expected Violation approach (Jiao et al., 2019, 2018) which assigns a

[8] negative constraint values are first replaced by 0.

constraint violation to each design, $v(\mathbf{x}^{(i)}) = \max_{j=1,\dots,m_c} (g_j(\mathbf{x}^{(i)}))_+$. For a feasible design, $v(\mathbf{x}) = 0$. As long as no feasible solution is found, in the spirit of the Expected Improvement,

$$\mathbf{x}^{(t+1)} = \arg\max_{\mathbf{x}\in X} \mathbb{E}[(v_{\min} - v(\mathbf{x}))_+], \tag{5.15}$$

the design which is expected to improve the most the smallest constraint violation $v_{\min} := \min_{i=1,\dots,t} v(\mathbf{x}^{(i)})$, is chosen. Note that $v(\cdot)$ has no longer Gaussian distribution, but (5.15) has semi-analytic expression. Another drawback is that the violation in the m_c constraints are aggregated by the max operator, and the constraint responsible for $v_{\min}$ may change over time.

In G, the target $\mathbf{0}_{m_c}$ is explicitly provided, and we therefore aim at using $\text{mEI}_{\mathbf{G}(\cdot)}(\cdot; \mathbf{0}_{m_c})$, where the $\mathbf{G}(\cdot)$ subscript indicates that the expected value is taken with regard to the $\mathbf{G}(\cdot)$ metamodels. In the update philosophy of $\mathbf{R}$-mEI where the reference point is adapted to the current state of the Pareto front, to avoid a drastically too severe target which may promote the most uncertain designs, $\mathbf{0}_{m_c}$ may benefit from an adaptation to $\widehat{\mathcal{P}_{\mathcal{Y}}^{\mathbf{c}}}$, the empirical Pareto front in G. Like in $\mathbf{R}$-mEI, a line $\mathcal{L}^{\mathbf{c}}$ joining two anchor points $\mathbf{I}^{\mathbf{c}}$ and $\mathbf{N}^{\mathbf{c}}$ is drawn, whose closest point to $\widehat{\mathcal{P}_{\mathcal{Y}}^{\mathbf{c}}}$, $\mathbf{R}^{\mathbf{c}}$, is used as $\text{mEI}_{\mathbf{G}(\cdot)}$'s reference point. $\mathbf{N}^{\mathbf{c}}$ does not appear to be critical to attain the feasible region and is chosen as the empirical Nadir point of $\widehat{\mathcal{P}_{\mathcal{Y}}^{\mathbf{c}}}$, $\mathbf{N}^{\mathbf{c}} = \overline{\mathbf{N}}^{\mathbf{c}}$. Two alternatives are relevant regarding $\mathbf{I}^{\mathbf{c}}$. One can choose the crude $\mathbf{0}_{m_c}$. Another possibility is to adapt to previous observations by using $\mathbf{I}^{\mathbf{c}} = \overline{\mathbf{I}^{\mathbf{c}}}$, the empirical Ideal point of $\widehat{\mathcal{P}_{\mathcal{Y}}^{\mathbf{c}}}$. Recall that by definition of constrained Pareto-dominance (Definition 5.1), all negative constraint values have been replaced by 0, which therefore is a lower bound for $\overline{\mathbf{I}^{\mathbf{c}}}$'s components: $\overline{I_j^{\mathbf{c}}} = \max(0, \min_{i=1,\dots,t} g_j(\mathbf{x}^{(i)}))$. Note that $\overline{\mathbf{I}^{\mathbf{c}}} = \mathbf{0}_{m_c}$ once marginally feasible designs have been observed in each constraint (i.e. $\min_{i=1,\dots,t} g_j(\mathbf{x}^{(i)}) \leq 0$, $j = 1, \dots, m$), a situation that mostly occurs. Using directly $\overline{\mathbf{I}^{\mathbf{c}}}$ or $\mathbf{0}_{m_c}$ for $\mathbf{R}^{\mathbf{c}}$ can lead to longer attainment times of $\mathcal{I}_{\mathbf{0}_{m_c}}$ as will be shown in the following experiments, and depending on the problem, it can be worth using the adapted $\mathbf{R}^{\mathbf{c}}$ which smoothly leads to $\mathbf{0}_{m_c}$.

Figure 5.8 illustrates these concepts. $\overline{\mathbf{I}^{\mathbf{c}}}$, $\mathbf{0}_{m_c}$ (red squares) and the adapted reference points $\mathbf{R}^{\mathbf{c}}$ they define are shown (blue squares) in the three different cases which may occur: either $\overline{\mathbf{I}^{\mathbf{c}}}$ equals $\mathbf{0}_{m_c}$ (left, feasible solutions have marginally been observed in each constraint, which is the most common situation). Or $\mathbf{0}_{m_c}$ dominates $\overline{\mathbf{I}^{\mathbf{c}}}$ (center, no feasible design has been found in any constraint). Or $\overline{\mathbf{I}^{\mathbf{c}}}$ and $\mathbf{0}_{m_c}$ differ in $0 < j < m_c$ constraints (right, feasible designs in $g_1(\cdot)$ have been found but not in $g_2(\cdot)$).

Experiments in severely constrained problems

Severely constrained multi-objective problems

In this part, three different benchmark problems are considered: OSY, SRN and TNK (Deb, 2001). The dimension, number of objectives and number of constraints of these

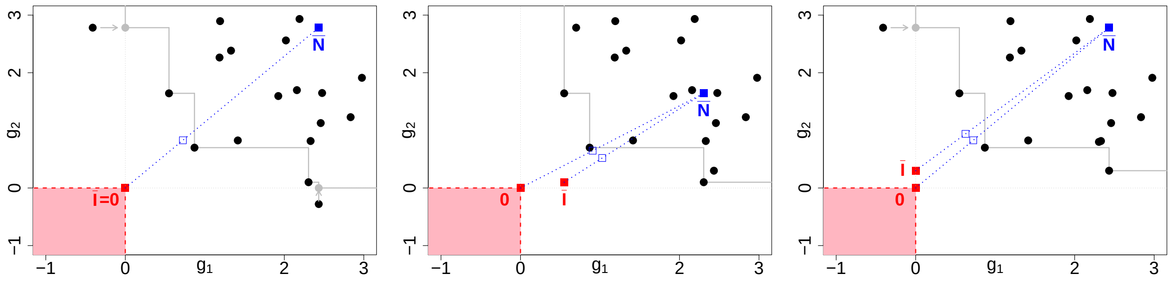


Figure 5.8: Severely constrained problem: in the space of constraints, no observation dominates $\mathbf{0}_{m_c}$. mEI's targeting ability can be employed for attaining the feasible light red part of G. $\overline{\mathbf{I^c}}$, $\mathbf{0}_{m_c}$, or their update (corresponding blue square on the line to $\overline{\mathbf{N^c}}$) can be used as $\mathbf{R^c}$, mEI's reference point in the constraint space. Grey dots are the projection of observed constraint values onto the positive orthant of $\mathbb{R}^{m_c}$.

problems are given in Table 5.14. The experiments start with a space-filling DoE of $n = 3d$ observations and are run for $p = 3d$ supplementary iterations.

Problem	d	m	m_c	# Feasible
OSY	6	2	6	4
SRN	2	2	2	6
TNK	2	2	2	2

Table 5.14: Dimension, number of objectives and of constraints, and runs with at least one feasible design in the initial DoE, for the constrained multi-objective problems.

The difference with the problems in Section 5.2.1.4 is that depending on the initial DoE, there may not exist any feasible $\mathbf{x} \in \mathbf{x}^{(1:n)}$. The number of runs (out of 10) with at least one feasible design is reported in the "# Feasible" column in Table 5.14. If no design in $\mathbf{x}^{(1:n)}$ is feasible, before using the constrained multi-objective mEIPF or EHIPF, a feasible design needs to be found via the use of a feasibility infill criterion. This is the aim of the following three criteria: mEI (with reference point set at $\mathbf{0}_{m_c}$ in this part), PoF, and EV. 6 different combinations of `Constrained multi-objective criterion + Feasibility criterion` can be defined and are investigated here: `EHIPF+mEI`, `EHIPF+PoF`, `EHIPF+EV`, `mEIPF+mEI`, `mEIPF+PoF` and `mEIPF+EV`. As in Section 5.2.1.4, the ability of these infill criteria to attain feasible central parts of the Pareto front are compared using the hypervolume indicator restricted to the central $\mathcal{I}_{0.1}$ (normalized by the hypervolume indicator of the true front in $\mathcal{I}_{0.1}$) and the attainment time of $\mathcal{I}_{0.1}$. The amount of feasible solutions is also compared, as well as the number of iterations required before finding a feasible design (column "Attainment time of $\mathbf{0}_{m_c}$"). This last indicator is set to $n = 3d$ if a feasible design is found in the initial DoE, and equals $t > n$ if $\mathbf{x}^{(t)}$ is the first feasible design. The attainment time of the feasible region only compares the ability of the `Feasibility criterion` to produce feasible designs, which is further investigated in the next experiments.

The indicators are averaged over 10 runs (standard deviation in brackets) and reported in Table 5.15. When at least one run did not attain $\mathcal{I}_{0.1}$, an estimator of the empirical runtime (Auger and Hansen, 2005) is given in red, the number of successful runs being reported in brackets. × indicates that no run was able to access $\mathcal{I}_{0.1}$.

OSY problem

	Hypervolume in $\mathcal{I}_{0.1}$	Attainment time of $\mathcal{I}_{0.1}$	Feasible designs	Attainment time of $\mathbf{0}_{m_c}$
EHIPF+mEI	0.428 (0.354)	35.3 [7]	10.6 (2.4)	18.6 (0.5)
EHIPF+PoF	0.178 (0.328)	91.1 [3]	11.7 (1.5)	18.6 (0.5)
EHIPF+EV	0.161 (0.303)	70 [4]	10.6 (2.4)	19 (1.1)
mEIPF+mEI	0.616 (0.130)	22.7 (1.2)	7.9 (1)	18.6 (0.5)
mEIPF+PoF	0.642 (0.118)	22.9 (1.8)	8 (1.3)	18.6 (0.5)
mEIPF+EV	0.589 (0.063)	22.5 (1.3)	7.2 (1.8)	19 (1.1)

SRN problem

	Hypervolume in $\mathcal{I}_{0.1}$	Attainment time of $\mathcal{I}_{0.1}$	Feasible designs	Attainment time of $\mathbf{0}_{m_c}$
EHIPF+mEI	0.191 (0.248)	23.8 [4]	3.7 (0.7)	7.4 (1.8)
EHIPF+PoF	0.118 (0.195)	32.3 [3]	3.6 (0.7)	6.5 (0.7)
EHIPF+EV	0.094 (0.186)	31.1 [3]	3.3 (0.9)	7.6 (2.1)
mEIPF+mEI	0.375 (0.171)	10.4 (1.3)	3.7 (1.1)	7.4 (1.8)
mEIPF+PoF	0.389 (0.169)	9.9 (0.9)	4 (0.8)	6.5 (0.7)
mEIPF+EV	0.398 (0.164)	10.5 (1.2)	3.7 (1.1)	7.6 (2.1)

TNK problem

	Hypervolume in $\mathcal{I}_{0.1}$	Attainment time of $\mathcal{I}_{0.1}$	Feasible designs	Attainment time of $\mathbf{0}_{m_c}$
EHIPF+mEI	0.016 (0.046)	55.5 [2]	2.5 (1)	7.7 (1.1)
EHIPF+PoF	0	×	2.4 (1)	7 (0.7)
EHIPF+EV	0	×	2.2 (1.1)	7.6 (1.5)
mEIPF+mEI	0.100 (0.109)	16.9 [6]	2.3 (0.8)	7.7 (1.1)
mEIPF+PoF	0.101 (0.140)	20.8 [5]	2 (0.8)	7 (0.7)
mEIPF+EV	0.098 (0.151)	25.6 [4]	1.7 (0.9)	7.6 (1.5)

Table 5.15: Performance metrics averaged over 10 runs (standard deviation in brackets) obtained by the three investigated infill criteria.

These results again indicate the advantage of using mEIPF for targeting the center of the Pareto front in a constrained problem over EHIPF. The central part of $\mathcal{P_Y}$ is attained much faster and consistently with this constrained multi-objective criterion. Again, the number of feasible designs obtained by using mEIPF is slightly smaller than the one returned by the EHIPF criterion. Concerning the Feasibility criterion used when no valid design is found in the initial DoE, PoF slightly outperforms mEI (in the space of constraints) and EV. The attainment times are nonetheless small for all criteria which quickly succeed in finding a feasible design. In these examples where it is not too complicated to find feasible designs, the lack of exploration of PoF (Jones, 2001) is not an issue, neither it is for mEI and EV which explore the design space a little more. In the case EHIPF is used as `Constrained multi-objective criterion`, the variants that have used mEI in the phase where no feasible design was found better attain $\mathcal{I}_{0.1}$ than `EHIPF+PoF`, even though the latter finds feasible designs slightly more rapidly. This phenomenon might be explained by the more exploratory behavior of mEI compared to PoF which is attracted by already visited points. While this may lead to a slightly longer time to find feasible designs, once some have been found, the metamodel in the `EHIPF+mEI` option may be more accurate than the one of `EHIPF+PoF`.

Finding feasible designs: the YUCCA problem

Finally, in this section, we investigate the performance of the mEI, PoF and EV criteria to find feasible designs in a highly constrained case, the YUCCA problem introduced in Feliot (2017). This problem has tunable dimension d and $m_c = 2d$ constraints $g_j(\mathbf{x})$. The latter are extremely antagonist and the proportion of the feasible space $\mathcal{F}_X \subset X : \mathbf{g}(\mathcal{F}_X) \preceq \mathbf{0}_{m_c}$ is only $(10^{-\kappa})^d$ where κ is a harshness parameter, that will be set to 1, 3, 5 in the following. The complexity of the YUCCA test suite grows with d as the design space is less densely populated. The κ parameter defines the size of the off-centered $\mathcal{F}_X$ hypercube. As in Feliot (2017), experiments are conducted with $d = 2, 5, 10, 20$ dimensions with initial designs of respectively $n = 10, 20, 30, 40$ observations. The algorithms are run for at most $p = 200$ iterations and are stopped once a feasible design is found (see Table 5.16).

d	n	p
2	10	200
5	20	200
10	30	200
20	40	200

Table 5.16: Investigated YUCCA problems.

For the different values of κ and the settings of Table 5.16, mEI, PoF and EV are compared in terms of attainment time of $\mathbf{0}_{m_c}$. The mEI `adapted` and PoF `adapted` variants rather use the adapted target $\mathbf{R}^c$ (see Figure 5.8) instead of $\mathbf{0}_{m_c}$, to smoothly direct the optimization towards the feasible region. In each run, all constraints are marginally satisfied (i.e. $\forall j = 1, \dots, 2d, \exists \mathbf{x} \in \mathbf{x}^{(1:n)} : g_j(\mathbf{x}) \leq 0$). Therefore, the leftmost situation in Figure 5.8 in which $\mathbf{0}_{m_c} = \overline{\mathbf{I}^\mathbf{c}}$ is encountered and both variants are the same.

Table 5.17 reports the number[9] of acquisition function queries required to attain $\mathcal{F}_X$. The results are averaged over 10 runs and the standard deviation is given in brackets. When at least one run did not enter $\mathcal{F}_X$ within the $p = 200$ iterations, an estimator of the empirical runtime (Auger and Hansen, 2005) is given in red, the number of successful

[9] Values smaller than 1 are due to the rare cases in which one feasible design was found in the initial DoE. This only happens two times in the easiest ($d = 2$, $\kappa = 1$) problem.

runs being reported in brackets. × means none of the 10 runs reached $\mathcal{F}_X$. The three tables correspond to $\kappa = 1, 3, 5$ and each row to a dimension d of the YUCCA problem (remember there are $2d$ constraints) given in Table 5.16.

$\kappa = 1$

d	mEI	mEI adapted	PoF	PoF adapted	EV
2	0.8 (0.4)	2.2 (1.3)	0.8 (0.4)	8.9 (13.8)	1.3 (0.8)
5	1 (0)	3 (1.2)	1 (0)	6.7 (6.9)	2 (0)
10	1.1 (0.3)	3 (0)	1 (0)	5.9 (0.6)	2 (0)
20	2.4 (1)	3.9 (1.5)	1.3 (0.5)	19.6 [5]	2.6 (0.5)

$\kappa = 3$

d	mEI	mEI adapted	PoF	PoF adapted	EV
2	2 (0)	12.1 (3.8)	2 (0)	×	2 (0)
5	4 (0.8)	6.7 (0.7)	3 (0)	×	3 (0)
10	11.4 (0.8)	11.4 (1.7)	4.1 (0.3)	×	3.7 (0.5)
20	130.7 (5.1)	106.2 (11.8)	7.6 (3.1)	×	6 (0)

$\kappa = 5$

d	mEI	mEI adapted	PoF	PoF adapted	EV
2	3 (0)	20.3 (7.4)	3 (0)	×	3 (0)
5	6.3 (0.8)	7 (0.8)	4.4 (0.5)	×	4.1 (0.3)
10	32.3 (3)	12.1 (1.2)	6.1 (0.6)	×	4.6 (0.5)
20	×	229.6 [5]	11.7 (2.3)	×	7.2 (0.4)

Table 5.17: Attainment time of $\mathcal{F}_X$ on the YUCCA problem with varying d and κ, for different infill criteria.

Overall, mEI is outperformed by PoF and EV. While it performs comparably well in problems where d (hence m_c) and κ are small, mEI faces difficulties as far as the problem becomes more challenging ($\kappa = 3, 5$) or when more constraints are considered ($d = 10, 20$). It is worth analyzing the effect of the adapted reference point. While in easy problems ($\kappa = 1$ and/or $d = 2, 5$) using an adapted, not as ambitious as $\mathbf{0}_{m_c}$ reference point slows the attainment of $\mathcal{F}_X$, in harder problems (smaller feasible region and/or more constraints) it avoids the exploratory behavior of mEI$_{\mathbf{G}(\cdot)}(\cdot; \mathbf{0}_{m_c})$ by considering stepwise improvements leading to $\mathbf{0}_{m_c}$. The weak performance of mEI adapted when $d = 2$ is due to the small DoE. In early iterations, mEI adapted targets a part of the constraint space which includes $\mathbf{0}_{m_c}$ but is slightly off-centered because of the YUCCA problem in which it is easy to satisfy the first even constraints ($g_2(\cdot)$, $g_4(\cdot), \dots$) but hard to satisfy the first odd constraints ($g_1(\cdot)$, $g_3(\cdot), \dots$). $\mathbf{R}^c$'s components are in general similar and > 0 in the first odd constraints, and close to 0 in the first even constraints. This bias is corrected as far as more observations get available and $\mathbf{R}^c$ becomes closer to $\mathbf{0}_{m_c}$. It remains that mEI is too ambitious for such a problem since the worth of a design is measured by the amount it may dominate $\mathbf{0}_{m_c}$, which is not a concern when only the attainment of $\mathcal{F}_X$ is aimed at. Its logic is not adapted to the small subset of designs that actually respect the constraints. mEI samples from a too wide range of designs and is not the best option for attaining a highly constrained $\mathcal{F}_X$.

By nature, PoF only looks for designs that outperform $\mathbf{0}_{m_c}$, whatever the magnitude. This is the reason why it surpasses mEI when the harshness and/or the number of constraints of the problem increase. Using the adapted reference point in lieu of $\mathbf{0}_{m_c}$ in PoF's formulation leads to the worst results. When used in a Probability of Improvement setting (2.7, Jones, 2001), this acquisition function is known for promoting designs that only slightly improve over the target. PoF `adapted` progresses towards $\mathbf{0}_{m_c}$ such slowly that it is not able to find any feasible design when $\kappa = 3, 5$. In the $\kappa = 1$ instance, it faces difficulties with $d = 2, 5$ because of the off-centered though adapted reference point used during the early iterations. In the $d = 20$ setting, its progress towards $\mathcal{F}_X$ heavily depends on the initial DoE and 5 runs out of 10 fail at reaching $\mathcal{F}_X$. Besides, the good performance of PoF suggests the use of the Probability of Improvement (2.7) in multi-objective problems with an ambitious but attainable target. A similar approach was recently proposed in Emmerich et al. (2020) where PI_ε, the Probability of Improvement over an optimistic front was employed.

Overall, EV is the acquisition function that performs the best on the YUCCA test problem. It is outperformed by PoF and even by mEI on the easiest instances ($\kappa = 1$ and/or $d = 2$), but attains $\mathcal{F}_X$ slightly faster than PoF when more constraints and/or smaller feasible design spaces are considered. Aggregating the m_c constraints into one function (the constraint violation) might seem hazardous in case of extremely antagonist functions as $g_1(\cdot)$ and $g_2(\cdot)$, because $v(\cdot)$ does not account for the constraint that is violated. However, the m_c YUCCA constraints being pairs of nonlinear box-constraints, the independence assumption PoF and mEI hinge on is clearly not verified. This may explain EV's good performance for $m_c = 20, 40$ constraints as their simultaneous satisfaction is circumvented by considering their maximal violation.

5.3 Further possible improvements

In this section, technical details regarding the C-EHI/R-EHI are discussed, as well as possibilities offered by the mEI criterion to cheaply approximate the EHI criterion or to conduct batch multi-objective optimization in a different manner.

5.3.1 On the choice of the updated reference point

In Chapter 4, the mEI criterion is defined together with a reference point that enables targeting a part of $\mathcal{P}_\mathcal{Y}$. Whether the reference point is user-provided or the center is targeted, a new goal on the Pareto front is defined (see Figure 4.6). This goal is the intersection with the $\mathcal{L}$ (or $\mathcal{L}'$) line for continuous fronts. In the case of a discontinuous or discrete front as the empirical front $\widehat{\mathcal{P}_\mathcal{Y}}$, the updated reference point $\widehat{\mathbf{R}}$ or the estimated center of the Pareto front, $\widehat{\mathbf{C}}$, accommodates $\widehat{\mathcal{P}_\mathcal{Y}}$ by being set at the projection of $\widehat{\mathcal{P}_\mathcal{Y}}$ on

$\widehat{\mathcal{L}}'$ (or on $\widehat{\mathcal{L}}$, respectively). It is then used as mEI's reference point, $\mathbf{R}$, whose maximizer with respect to $\mathbf{x}$ is the next evaluated design.

Even though other threshold values have been investigated (Jones, 2001), the spirit of the EGO algorithm (Jones et al., 1998) is to consider the improvement over $f_{\min}$, the currently best observed value. In a multi-objective setting, the equivalent to "$f_{\min}$" is "a non-dominated point". A plenty of reference points $\mathbf{R}$ extend in this sense. The hatched green parts of Figure 5.9 show the reference points which comply with this logic. Different choices of $\mathbf{R}$ are possible, even when restricting to a line. As for the EI threshold a (2.8), the more $\mathbf{R}$ is optimistic (i.e. small in all its components), the more the criterion becomes exploratory and may promote designs with large uncertainty; on the contrary, $\mathbf{R}$'s located in the vicinity of the Pareto front barrier (red line) might promote $\mathbf{x}$'s near to already observed designs that are close to dominating $\mathbf{R}$. More than the projection[10] on $\widehat{\mathcal{L}}$ (blue square), alternative definitions exist for the updated reference point. For instance, $\mathbf{R}$ could be set at the intersection between $\widehat{\mathcal{L}}$ and $\widehat{\mathcal{P}_\mathcal{Y}}$ (red square), or at the junction between the green dominating envelope and $\widehat{\mathcal{L}}$ (green square). It could also be thought as the middle between these points (brown square), or could even be the intersection with some simulated fronts. Copulas (Nelsen, 2007) are another way towards a continuous representation of $\mathcal{P}_\mathcal{Y}$ (Binois et al., 2015c) with which the intersection might be chosen.

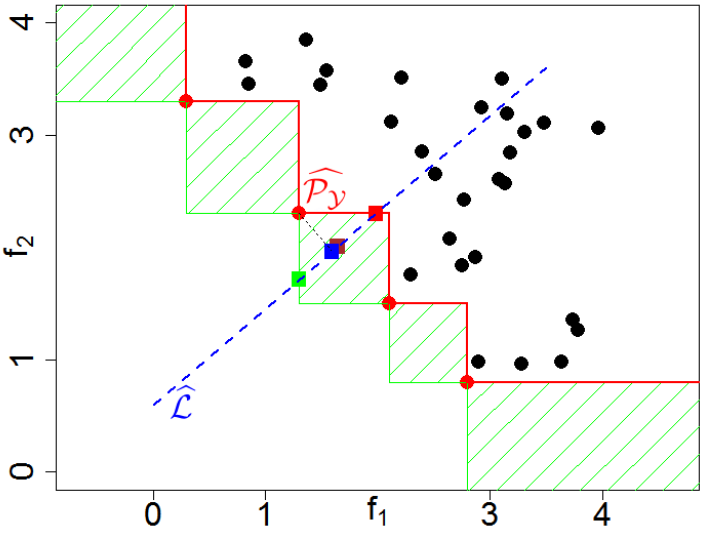


Figure 5.9: Empirical Pareto front ($\widehat{\mathcal{P}_\mathcal{Y}}$, red), estimated Ideal-Nadir line ($\widehat{\mathcal{L}}$, blue) and possible valid reference points on $\widehat{\mathcal{L}}$ (squares), belonging to the hatched green non-dominated zone.

Our investigations have shown that the way the reference point is produced to accommodate $\widehat{\mathcal{P}_\mathcal{Y}}$ is not critical in the optimization and is problem dependent. Further work may nonetheless address this question theoretically to derive the optimal management

[10] Which might lead to a point outside the green area, in which case it is projected onto the rectangle.

of $\mathbf{R}$. The intersection with $\widehat{\mathcal{P}_{\mathcal{Y}}}$ has the desirable property of valuing any $\mathbf{y} \in \widehat{\mathcal{L}}$ that increases the Pareto front. However in this setting, even though the improvement of any $\mathbf{y} \in \widehat{\mathcal{P}_{\mathcal{Y}}}$ is null, there exists one such $\mathbf{y}$ which (weakly) dominates $\mathbf{R}$ as $y_j < R_j$ in $m-1$ objectives and $y_{j_0} = R_{j_0}$ in the remaining objective j_0. The EHI-mEI equivalence (Proposition 4.1) holds but the search might be attracted towards this $\mathbf{y}$, because a little enhancement of y_{j_0} leads to a strictly positive improvement.

5.3.2 Anticipation of the attainable region

In Section 4.6, the computational cost of the determination of $\mathcal{I}_{\mathbf{R}^*}$, the area to target for the remaining iterations, can be further improved. Since EHI has a closed-form expression, its update can be accelerated using the kriging variance update formulae of Chevalier et al. (2014). This is computationally appealing if the maximization is carried out on a fixed discrete set of designs. Another possibility for accelerating the virtual iterations is to replace the potentially costly EHI by a cheaper and similar acquisition function such as SMS (Section 2.4, Ponweiser et al., 2008), or the Matrix-Based Expected Improvement (Zhan et al., 2017). A last alternative is to pre-compute the Pareto set of the kriging mean functions, $\mathcal{P}_{\mathcal{X}}(\widehat{\mathbf{y}}(\cdot)) \subset X$ using an EMOA, and to iteratively choose $\mathbf{x}^{*(i)} = \underset{\mathbf{x} \in \mathcal{P}_{\mathcal{X}}(\widehat{\mathbf{y}}(\cdot))}{\arg\max} \mathrm{EHI}(\mathbf{x}, \mathbf{R}^c)$.

Regarding the virtual steps, instead of a Kriging Believer strategy, the use of nested simulations of the GP to account for the metamodel's uncertainty combined with the reconditioning of previous GP simulations (Chevalier et al., 2015) was also considered. Several virtual optimizations were nonetheless necessary for each candidate reference point $\mathbf{R}^c$, which led to an even more expensive determination of $\mathbf{R}^*$.

Finally, a cheap way to choose $\mathbf{R}^*$ is to consider the proportion of the (estimated) Ideal-Nadir hyperbox in which convergence has occurred when the convergence criterion triggers, and to determine the $\mathbf{R}^*$ along the line for which a linear extrapolation in the b remaining iterations indicates convergence in the Ideal-$\mathbf{R}^*$ hyperbox at the end of the search. This criterion is computationally much more tractable, but is a much coarser approximation of the part of the objective space that can be unveiled in the remaining iterations. It does not take the logic of the infill criterion, nor the shape of the front into account.

5.3.3 Multiple targets

More generally, once $\mathbf{R}^*$ has been set up, the b remaining maximizations of $\mathrm{EHI}(\cdot; \mathbf{R}^*)$ within C-EHI and R-EHI may be cumbersome. Similarly to the Matrix-Based Expected Improvement, a computationally cheap proxy to EHI which is equivalent to the max of mEI's with the elements of $\widehat{\mathcal{P}_{\mathcal{Y}}}$ as reference points (Zhan et al., 2017), an mEI-like criterion can be devised to substitute EHI. Directly applying $\mathrm{mEI}(\cdot; \mathbf{R}^*)$ is not an option. Due to the targeting property and to the myopia of this criterion with respect to $\widehat{\mathcal{P}_{\mathcal{Y}}}$, a well spread approximation in $\mathcal{I}_{\mathbf{R}^*}$ will not be achieved. However an additive mEI criterion, or a maximal mEI criterion

$$\text{add-mEI}(\mathbf{x}; \{\mathbf{R}^1, \dots, \mathbf{R}^r\}) := \sum_{i=1}^{r} \text{mEI}(\mathbf{x}; \mathbf{R}^i),$$

$$\text{max-mEI}(\mathbf{x}; \{\mathbf{R}^1, \dots, \mathbf{R}^r\}) := \max_{i=1,\dots,r} \text{mEI}(\mathbf{x}; \mathbf{R}^i),$$

may have desirable properties if a suitable set of reference points $\{\mathbf{R}^1, \dots, \mathbf{R}^r\}$ is established. The $\mathbf{R}^i$'s have to be non-dominated, cover as much as possible $\mathcal{I}_{\mathbf{R}^*}$'s non-dominated subspace, and overlap as little as possible, as shown in the left part of Figure 5.10. Relevant points of $\widehat{\mathcal{P}_\mathcal{Y}}$ might be chosen as $\mathbf{R}^i$'s, or they could be the projection of $\widehat{\mathcal{P}_\mathcal{Y}}$ onto other lines than the Ideal-Nadir line.

As a box decomposition (Yang et al., 2019a) of EHI leads to a (potentially large) sum and subtraction of mEI's with particular reference points ($\sum_i \text{mEI}(\cdot, \mathbf{A}^i)$-$\sum_j \text{mEI}(\cdot, \mathbf{S}^j)$) as shown in the right part of Figure 5.10, restricting to few well-chosen terms $\mathbf{A}^i$ (and $\mathbf{S}^j$) will speed up the computation.

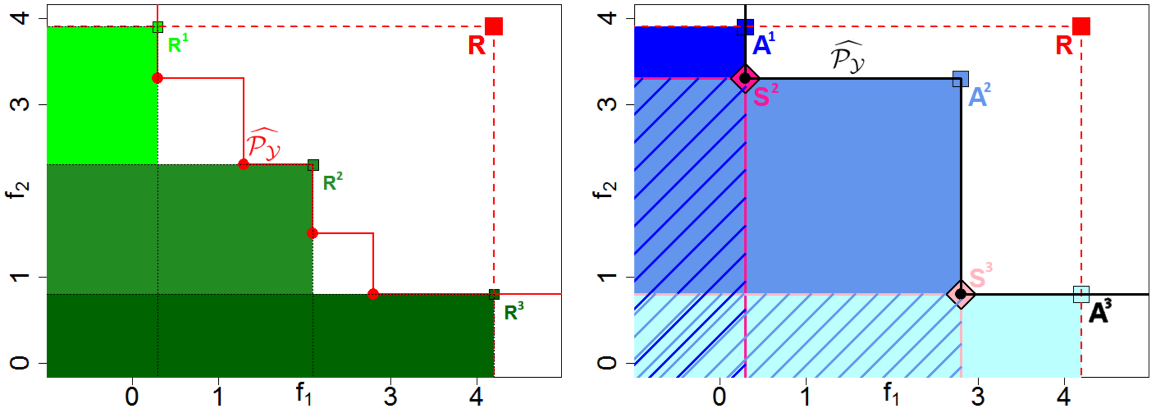


Figure 5.10: Left: example of three non-dominated reference point which cover $\mathcal{I}_\mathbf{R}$ and might be candidates in an additive or maximal mEI framework. Right: bi-objective example for the computation of EHI. The hypervolume improvement of $\mathbf{y} \in \mathbb{R}^m$ is the sum of product improvements with respect to $\mathbf{A}^1$, $\mathbf{A}^2$ and $\mathbf{A}^3$ (blue filled areas + hatched blue rectangles) minus the product improvements with respect to $\mathbf{S}^2$ and $\mathbf{S}^3$ (pink rectangles, to remove the overlap between the $\mathbf{A}^i$'s).

The use of multiple reference points opens a new possibility for conducting batch-optimization discussed in Section 5.1 too: instead of searching q designs that jointly improve the most over $\mathbf{R}$ in the mEI or EHI sense, q designs that individually improve over well-distributed $\mathbf{R}^1, \dots, \mathbf{R}^q$ can be searched. Splitting the objective space in areas where to conduct the optimization in parallel is a common practice in multi-objective optimization to obtain a batch of designs (Horn et al., 2015; Zhang and Li, 2007), and the targeting Truncated EHI criterion (Palar et al., 2018; Yang et al., 2016a,b) has recently been employed in this purpose by Yang et al. (2019c).

Chapter 6

From CAD to Eigenshapes for Optimization in Reduced Dimension

Contents

In this chapter, we consider the optimization of parametric shapes. In this framework, the minimization of an objective function $f(\mathbf{x})$ where $\mathbf{x}$ are CAD (Computer Aided Design) parameters, is aimed at. This task is difficult when $f(\cdot)$ is the output of an expensive-to-evaluate numerical simulator and the number of CAD parameters is large.

Most often, the set of all considered CAD shapes resides in a manifold of lower effective dimension in which it is preferable to build the surrogate model and perform the optimization. In this chapter, we uncover the manifold through a high-dimensional shape mapping and build a new coordinate system made of eigenshapes. The surrogate model is learned in the space of eigenshapes: a regularized likelihood maximization provides the most relevant dimensions for the output. The final surrogate model is detailed (anisotropic) with respect to the most sensitive eigenshapes and rough (isotropic) in the remaining dimensions. Last, the optimization is carried out with a focus on the critical dimensions, the remaining ones being coarsely optimized through a random embedding and the manifold being accounted for through a replication strategy. At low budgets, the methodology leads to a more accurate model and a faster optimization than the classical approach of directly working with the CAD parameters.

For the sake of clarity, contrarily to Chapters 3, 4, 5, a single-objective optimization problem is considered. As mentioned at the end of the chapter, an extension to multi-objective problems is nonetheless achieved through minor modifications of the methodology.

6.1 Introduction

The most frequent approach to shape optimization is to describe the shape by a vector of d CAD parameters, $\mathbf{x} \in X \subset \mathbb{R}^d$ and to search for the parameters that minimize an objective function, $\mathbf{x}^* = \arg\min_{\mathbf{x} \in X} f(\mathbf{x})$. In the CAD modeling process, the set of all possible shapes has been reduced to a space of parameterized shapes, $\boldsymbol{\Omega} := \{\Omega_{\mathbf{x}}, \mathbf{x} \in X\}$.

It is common for d to be large, $d \gtrsim 50$. Optimization in such a high-dimensional design space is difficult, especially when $f(\cdot)$ is the output of an expensive simulator that can only be run a restricted number of times (Shan and Wang, 2010). Surrogate-based approaches (Forrester and Keane, 2009; Sacks et al., 1989) relying on a metamodel (e.g., Gaussian Processes, Cressie, 1992; Rasmussen and Williams, 2006; Stein, 1999) used throughout the previous chapters have proven their effectiveness to tackle optimization problems in a few calls to $f(\cdot)$ by evaluating designs promoted by an acquisition function such as the Expected Improvement (Mockus, 1975), cf. Section 2.2. However, such techniques suffer from the curse of dimensionality (Bellman, 1961) when d is large. The budget is also typically too narrow to perform sensitivity analysis (Saltelli et al., 2004) and select variables prior to optimizing. A further issue is that the CAD parameters $\mathbf{x}$ commonly have heterogeneous impacts on the shapes $\Omega_{\mathbf{x}}$: many of them are intended to refine the shape locally whereas others have a global influence so that shapes of practical interest involve interactions between all the parameters.

Most often, the set of all CAD generated shapes, $\boldsymbol{\Omega}$, can be approximated in a δ-dimensional manifold, $\delta < d$. In Raghavan et al. (2013, 2014) this manifold is accessed through an auxiliary description of the shape, $\phi(\Omega)$, $\phi(\cdot)$ being either its characteristic function or the signed distance to its contour. The authors aim at minimizing an objective function using diffuse approximation and gradient-based techniques, while staying on the manifold of admissible shapes. Active Shape Models (Cootes et al., 1995) provide another way to handle shapes in which the contour is discretized (Stegmann and Gomez, 2002; Wang, 2012).

Building a surrogate model in reduced dimension can be performed in different ways. The simplest is to restrict the metamodel to the most influential variables. But typical evaluation budgets are too narrow to find these variables before the optimization. Moreover, correlations might exist among the original dimensions (here CAD parameters) so that a selection of few variables may not constitute a valid reduced order description and meta-variables may be more appropriate. In Wu et al. (2019), the high-dimensional input space is circumvented by decomposing the model into a series of low-dimensional models after an ANOVA procedure. In Bouhlel et al. (2016), a kriging model is built in the space of the first Partial Least Squares axes for emphasizing the most relevant directions. Related approaches for dimensionality reduction inside GPs consist in a projection of the input $\mathbf{x}$ on a lower dimensional hyperplane spanned by orthogonal vectors. These vectors are determined in different manners, e.g. by searching the active space in Constantine et al. (2014); Li et al. (2019), or during the hyperparameters estimation in Tripathy et al. (2016). A more detailed bibliography of dimension reduction in GPs is conducted in Section 6.3.

For optimization purposes, the modes of discretized shapes (Stegmann and Gomez, 2002) are integrated in a surrogate model in Li et al. (2018a). In Cinquegrana and Iuliano (2018), the optimization is carried out on the most relevant modes using evolutionary algorithms combined with an adaptive adjustment of the bounds of the design space, also employed in Shan and Wang (2004).

Following the same route, in Section 6.2, we retrieve a shape manifold with dimension $\delta < d$. Our approach is based on a Principal Component Analysis (PCA, Wall et al., 2003) of shapes described in an ad hoc manner in the same vein as Cinquegrana and Iuliano (2018); Li et al. (2018a) but it provides a new investigation of the best way to characterize shapes. Section 6.3 is devoted to the construction of a kriging surrogate model in reduced dimension. Contrarily to Li et al. (2018a, 2019), the least important dimensions are still accounted for. A regularized likelihood approach is employed for dimension selection, instead of the linear PLS method (Bouhlel et al., 2016). In Section 6.4, we employ the metamodel to perform global optimization (Jones et al., 1998) via the maximization of the Expected Improvement (Mockus, 1975). A reduction of the space dimension is achieved through a random embedding technique (Wang et al., 2013) and a pre-image problem is solved to keep the correspondence between the eigenshapes and the CAD parameters.

6.2 From CAD description to shape eigenbasis

CAD parameters are usually set up by engineers to automate shape generation. These parameters may be Bézier or Spline control points which locally readjust the shape. Other CAD parameters, such as the overall width or the length of a component, have a more global impact on the shape. While these parameters are intuitive to a designer, they are not chosen to achieve any specific mathematical property and in particular do not let themselves interpret to reduce dimensionality.

In order to define a better behaved description of the shapes that will help in reducing dimensionality, we exploit the fact that the time to generate a shape $\Omega_{\mathbf{x}}$ is negligible in comparison with the evaluation time of $f(\mathbf{x})$.

In the spirit of kernel methods (Schölkopf et al., 1997; Vapnik, 1995), we analyze the designs $\mathbf{x}$ in a high-dimensional feature space $\Phi \subset \mathbb{R}^D$, $D \gg d$ (potentially infinite dimensional) that is defined via a mapping $\phi(\mathbf{x})$, $\phi : X \to \Phi$. With an appropriate $\phi(\cdot)$, it is possible to distinguish a lower dimensional manifold embedded in Φ. As we deal with shapes, natural candidates for $\phi(\cdot)$ are shape representations.

This chapter is motivated by parametric shape optimization problems. However, the approaches developed for metamodeling and optimization are generic and extend to any situation where a pre-existing collection of designs $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ and a fast auxiliary mapping $\phi(\mathbf{x})$ exist. $\phi(\mathbf{x}) = \mathbf{x}$ is a possible case. If $\mathbf{x}$ are parameters that generate a signal, another example would be $\phi(\mathbf{x})$, the discretized times series.

6.2.1 Shape representations

In the literature, shapes have been described in different ways. First, the *characteristic function* of a shape $\Omega_{\mathbf{x}}$ (Raghavan et al., 2013) is

$$\chi_{\Omega_{\mathbf{x}}}(\mathbf{s}) = \begin{cases} 1 \text{ if } \mathbf{s} \in \Omega_{\mathbf{x}} \\ 0 \text{ if } \mathbf{s} \notin \Omega_{\mathbf{x}} \end{cases} \tag{6.1}$$

where $\mathbf{s} \in \mathbb{R}^2$ or $\mathbb{R}^3$ is the spatial coordinate. χ is computed at some relevant locations (e.g. on a grid) $\mathbb{S} = \{\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(D)}\}$ and is cast as a D-dimensional vector of of 0's or 1's depending on whether the $\mathbf{s}^{(i)}$'s are inside or outside the shape.

Second, the *signed distance to the contour* $\partial\Omega_{\mathbf{x}}$ (Raghavan et al., 2014) is

$$\mathbb{D}_{\Omega_{\mathbf{x}}}(\mathbf{s}) = \varepsilon(\mathbf{s}) \min_{\mathbf{y} \in \partial\Omega_{\mathbf{x}}} \|\mathbf{s} - \mathbf{y}\|_2, \text{ where } \varepsilon(\mathbf{s}) = \begin{cases} 1 \text{ if } \mathbf{s} \in \Omega_{\mathbf{x}} \\ -1 \text{ if } \mathbf{s} \notin \Omega_{\mathbf{x}} \end{cases} \tag{6.2}$$

and is also computed at some relevant locations (e.g. on a grid) $\mathbb{S}$, transformed into a vector with D components.

Finally, the Point Distribution Model (Cootes et al., 1995; Stegmann and Gomez, 2002) where $\partial\Omega_{\mathbf{x}}$ is discretized at D/k locations $\mathbf{s}^{(i)} \in \partial\Omega_{\mathbf{x}} \subset \mathbb{R}^k$ ($k = 2$ or 3), also leads to a D-dimensional representation of $\Omega_{\mathbf{x}}$ where $\mathcal{D}_{\Omega_{\mathbf{x}}} = (\mathbf{s}^{(1)^\top}, \dots, \mathbf{s}^{(D/k)^\top})^\top \in \mathbb{R}^D$. For different shapes Ω and Ω', $\mathbb{S}$ has to be the same for χ and $\mathbb{D}$, and the *discretizations* $\{\mathbf{s}^{(1)^\top}, \dots, \mathbf{s}^{(D/k)^\top}\}$ of Ω and Ω' need to be consistent for $\mathcal{D}$. Figure 6.1 illustrates these shape representations for two different designs. The first one consists of three circles parameterized by their centers and radii. The second design is a NACA airfoil with three parameters. These shapes are described by the mappings $\phi(\mathbf{x}) \in \mathbb{R}^D$ with $\phi(\mathbf{x}) =$

$\chi_{\Omega_{\mathbf{x}}}(\mathbb{S}), \mathbb{D}_{\Omega_{\mathbf{x}}}(\mathbb{S})$ and $\mathcal{D}_{\Omega_{\mathbf{x}}}$, respectively. Specifying another design with parameters $\mathbf{x}'$ generally leads to $\phi(\mathbf{x}) \neq \phi(\mathbf{x}')$.

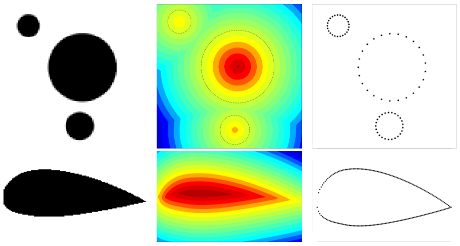


Figure 6.1: Shape representations for a design consisting of three circles (top) and for a NACA airfoil (bottom). The representations are the characteristic function (left), the signed distance to the contour (center), and the contour discretization(right).

6.2.2 PCA to retrieve the effective shape dimension

We map a large number (N) of plausible designs $\mathbf{x}^{(i)} \in X$ to $\Phi \subset \mathbb{R}^D$ and build the matrix $\mathbf{\Phi} \in \mathbb{R}^{N\times D}$ which contains the $\phi(\mathbf{x}^{(i)}) \in \mathbb{R}^D$ in rows and whose column-wise mean is $\overline{\boldsymbol{\phi}} \in \mathbb{R}^D$. In the absence of a set of relevant $\mathbf{x}^{(i)}$'s, these designs can be sampled from an a priori distribution, typically a uniform distribution. Next, we perform a Principal Component Analysis (PCA) on $\mathbf{\Phi}$: correlations are sought between the $\phi(\mathbf{x})_j$'s, $j = 1, \dots, D$. The eigenvectors of the empirical covariance matrix $\mathbf{C}_{\mathbf{\Phi}} := \frac{1}{N}(\mathbf{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top)^\top(\mathbf{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top)$, written $\mathbf{v}^j \in \mathbb{R}^D$, form an ordered orthonormal basis of Φ with decreasing importance as measured by the PCA eigenvalues λ_j, $j = 1, \dots, D$. They correspond to orthonormal directions in Φ that explain the most the dispersion of the high-dimensional representations of the shapes, $\phi(\mathbf{x}^{(i)})$. Any design $\mathbf{x}$ can now be expressed in the eigenbasis $\mathcal{V} := \{\mathbf{v}^1, \dots, \mathbf{v}^D\}$ since

$$\phi(\mathbf{x}) = \overline{\boldsymbol{\phi}} + \sum_{j=1}^{D} \alpha_j \mathbf{v}^j \tag{6.3}$$

where $(\alpha_1, \dots, \alpha_D)^\top =: \boldsymbol{\alpha} = \mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})$ are the coordinates in $\mathcal{V}$ (principal components), and $\mathbf{V} := (\mathbf{v}^1, \dots, \mathbf{v}^D) \in \mathbb{R}^{D\times D}$ is the matrix of eigenvectors (principal axes). α_j is the deviation from the mean shape $\overline{\boldsymbol{\phi}}$, in the direction of the eigenvector $\mathbf{v}^j$. The $\boldsymbol{\alpha}^{(i)}$'s form a manifold $\mathcal{A}_N := \{\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(N)}\}$ which approximates the true $\boldsymbol{\alpha}$ manifold, $\mathcal{A} := \{\boldsymbol{\alpha} \in \mathbb{R}^D : \exists \mathbf{x} \in X, \boldsymbol{\alpha} = \mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})\}$. Even though $\mathcal{A}_N \subset \mathbb{R}^D$, it is often a manifold of lower dimension, $\delta \ll D$, as we will soon see (Section 6.2.3).

Link with kernel PCA

N designs $\mathbf{x}^{(i)} \in \mathbb{R}^d$ have been mapped to a high-dimensional feature space $\Phi \subset \mathbb{R}^D$ in which PCA was carried out. This is precisely the task that is performed in Kernel PCA (Schölkopf et al., 1997), a nonlinear dimension reduction technique (contrarily to

PCA which seeks linear directions in $\mathbb{R}^d$). KPCA aims at finding a linear description of the data in a feature space Φ, by applying a PCA to nonlinearly mapped $\phi(\mathbf{x}^{(i)}) \in \Phi$. The difference with our approach is that the mapping $\phi(\cdot)$ as well as the feature space Φ are usually unknown in KPCA, since $\phi(\mathbf{x})$ may live in a very high-dimensional or even infinite dimensional space in which dot products cannot be computed efficiently. Instead, dot products are computed using designs in the original space X via a *kernel* which should not be mistaken with the kernel of GPs, $k_\phi : X \times X \to \mathbb{R}$, $k_\phi(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_\Phi$ (this is called the "kernel-trick", see Schölkopf et al., 1997; Vapnik, 1995). The eigencomponents of the points after mapping, $\alpha_j^{(i)} = \mathbf{v}^{j\top}(\phi(\mathbf{x}^{(i)}) - \overline{\boldsymbol{\phi}})$, can be recovered from the eigenanalysis of the $N \times N$ Gram matrix $\mathbf{K}_\phi$ with $K_{\phi_{ij}} = k_\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ (see Schölkopf et al., 1997; Wang, 2012, for algebraic details). Finding which original variables in $\mathbf{x}$ correspond to a given $\mathbf{v}^j$ is not straightforward and requires the resolution of a pre-image problem (Mika et al., 1999; Wang, 2012).

Having a shape-related and computable $\phi(\cdot)$ avoids these ruses and makes the principal axes $\mathbf{v}^j$ directly meaningful. It is further possible to give the expression of the equivalent kernel in our approach, in terms of the mapping $\phi(\cdot)$, from the polarization identity. By definition of the (centered) high dimensional mapping to Φ, $\mathbf{x} \mapsto \phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}$,

$$\begin{aligned}\|(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) - (\phi(\mathbf{x}') - \overline{\boldsymbol{\phi}})\|^2_{\mathbb{R}^D} &= \langle (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) - (\phi(\mathbf{x}') - \overline{\boldsymbol{\phi}}), (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) - (\phi(\mathbf{x}') - \overline{\boldsymbol{\phi}}) \rangle_{\mathbb{R}^D} \\ &= \|(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})\|^2_{\mathbb{R}^D} + \|(\phi(\mathbf{x}') - \overline{\boldsymbol{\phi}})\|^2_{\mathbb{R}^D} - 2\underbrace{\langle (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}), (\phi(\mathbf{x}') - \overline{\boldsymbol{\phi}}) \rangle_{\mathbb{R}^D}}_{k_\phi}\end{aligned}$$

hence,

$$k_\phi(\mathbf{x}, \mathbf{x}') = \frac{1}{2}(\|\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}\|^2_{\mathbb{R}^D} + \|\phi(\mathbf{x}') - \overline{\boldsymbol{\phi}}\|^2_{\mathbb{R}^D} - \|\phi(\mathbf{x}) - \phi(\mathbf{x}')\|^2_{\mathbb{R}^D}) \tag{6.4}$$

Logically, $k_\phi(\cdot, \cdot)$, a similarity measure between designs, is negatively proportional to the distance between the shape representations. Because of the size of the eigenanalyses to be performed, kernel PCA is advantageous over a mapping followed by a PCA when $D > N$, i.e. when the shapes have a very high resolution, and vice versa. In the current work where $\phi(\cdot)$ is known and D is smaller than 1000, we will follow the mapping plus PCA approach.

6.2.3 Experiments

In this section, all the parametric design problems used in the experiments throughout this chapter are introduced and discussed in terms of significant dimensions. Unless stated otherwise, the database $\boldsymbol{\Phi}$ is made of $N = 5000$ designs sampled uniformly in X. We start with 3 test cases of known intrinsic dimension, which will be complemented by 4 other test cases. The metamodeling and the optimization will be addressed later in Sections 6.3 and 6.4.

6.2.3.1 Retrieval of true dimensionality

In this part, we generate shapes of known low intrinsic dimension. In the Example 6.1 (cf. Figure 6.2), the shapes are circular holes of varying centers and radii, therefore described by 1, 2 or 3 parameters. In the Example 6.2 (cf. Figure 6.11), they are also circular holes but whose center positions and radii are described by sums[1] of parts of the 39 parameters. Last, in the Example 6.3 (cf. Figure 6.16), the shapes are made of three non overlapping circles with parameterized centers and radii. PCAs were then carried out on the $\mathbf{\Phi}$'s associated to the three mappings (characteristic function, signed contour distance and contour discretization). In each example, the 10 first PCA eigenvalues λ_j are reported. The $\boldsymbol{\alpha}$'s manifolds, $\mathcal{A}_N \subset \mathbb{R}^D$, are plotted in the first three dimensions as well as the first eigenvectors in the Φ space.

Example 6.1. *A hole in $\mathbb{R}^2$ parameterized by its radius ($d = 1$), its radius and the x-coordinate of its center ($d = 2$), or its radius and the x and y coordinates of its center ($d = 3$).*

[1] other algebraic operations such as multiplications have also led to the same conclusions.

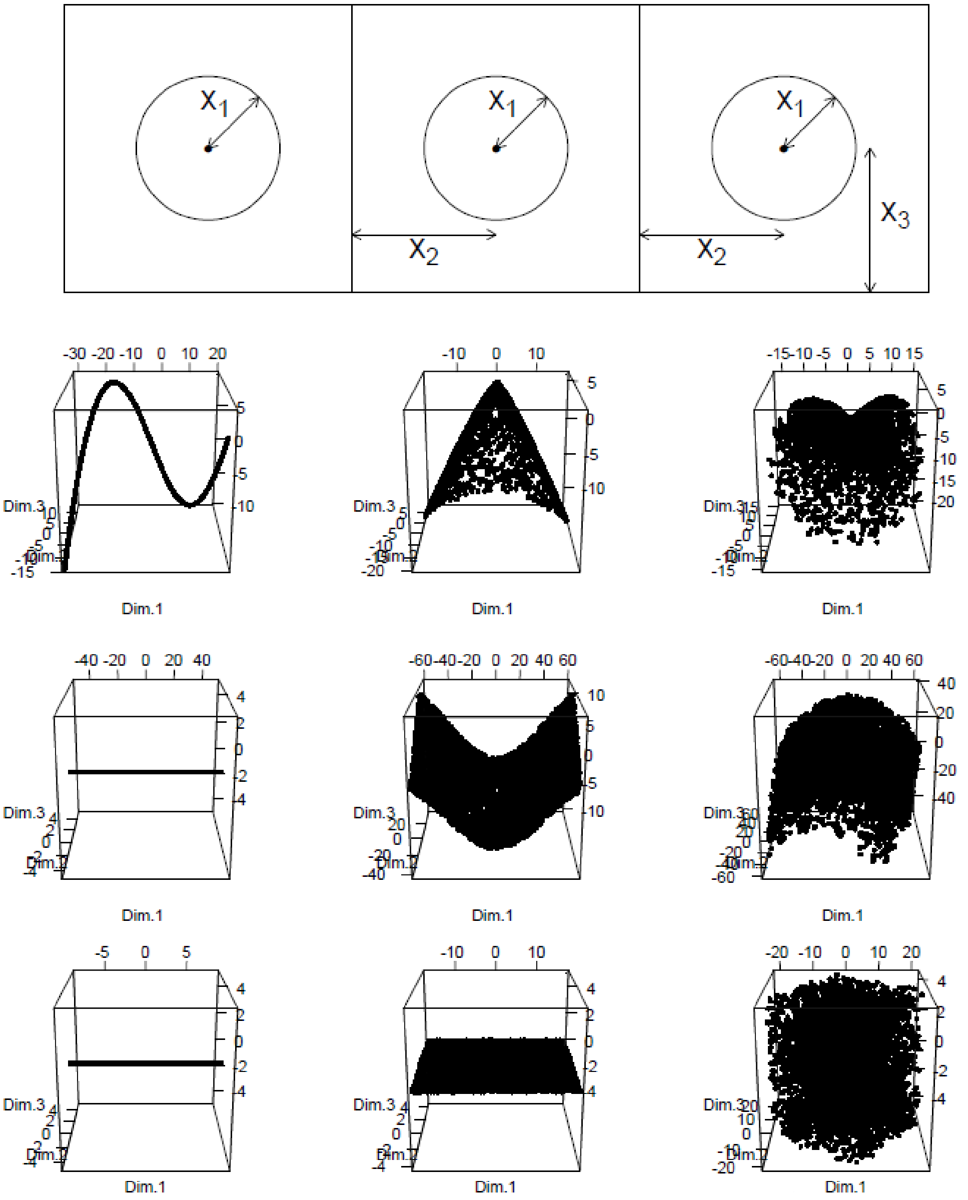


Figure 6.2: Example 6.1: three first eigencomponents of the $\boldsymbol{\alpha}^{(i)}$'s for three parametric test cases (columns) with low effective dimension equal to 1 (left), 2 (center) and 3 (right). The rows correspond to different ϕ's which are the characteristic function (top), the signed distance to the contour (middle) and the discretization of the contour (bottom).

	Characteristic function		Signed Distance		Discretization	
j	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage
1	324.63	63.09	840.14	100	25.20	100
2	75.98	77.86	0	100	0	100
3	32.69	84.21	0	100	0	100
4	18.20	87.75	0	100	0	100
5	11.48	89.98	0	100	0	100
6	8.12	91.56	0	100	0	100
7	5.92	92.71	0	100	0	100
8	4.45	93.57	0	100	0	100
9	3.50	94.25	0	100	0	100
10	2.79	94.80	0	100	0	100

Table 6.1: 10 first PCA eigenvalues for the different ϕ's, circle with $d = 1$ parameter.

	Characteristic function		Signed Distance		Discretization	
j	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage
1	60.90	26.50	1332.17	80.41	100.82	94.14
2	44.63	45.93	294.07	98.15	6.27	100
3	26.70	57.55	25.48	99.69	0	100
4	20.62	66.52	3.88	99.93	0	100
5	9.48	70.65	0.81	99.97	0	100
6	4.87	72.77	0.24	99.99	0	100
7	3.97	74.49	0.09	99.99	0	100
8	3.74	76.12	0.04	100	0	100
9	3.25	77.54	0.02	100	0	100
10	3.11	78.89	0.01	100	0	100

Table 6.2: 10 first PCA eigenvalues for the different ϕ's, circle with $d = 2$ parameters.

Figures 6.3-6.10 show the 9 first eigenvectors (if they have strictly positive eigenvalue) in the 3 cases of Example 6.1 with the three ϕ's.

	Characteristic function		Signed Distance		Discretization	
j	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage
1	26.48	10.12	1045.26	42.42	82.13	48.51
2	25.82	19.98	1037.44	84.53	80.82	96.26
3	20.58	27.84	300.14	96.71	6.34	100
4	19.38	35.24	33.83	98.08	0	100
5	15.65	41.22	18.49	98.83	0	100
6	11.36	45.56	14.40	99.42	0	100
7	11.20	49.84	3.78	99.57	0	100
8	11.05	54.06	3.64	99.72	0	100
9	7.52	56.93	1.58	99.78	0	100
10	7.21	59.69	1.55	99.84	0	100

Table 6.3: 10 first PCA eigenvalues for the different ϕ's, circle with $d = 3$ parameters.

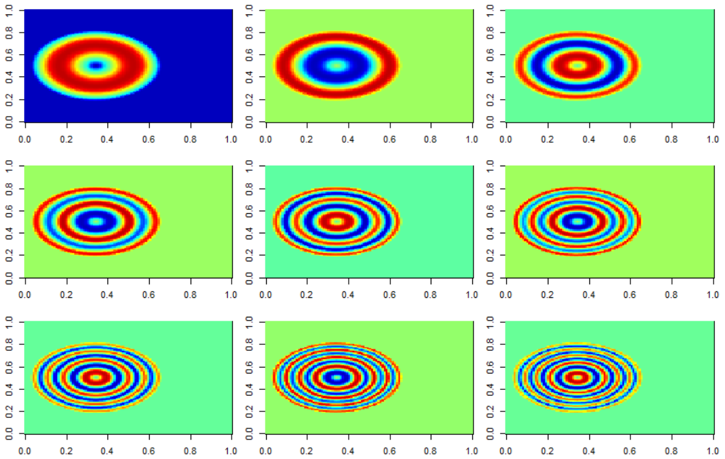


Figure 6.3: Example 6.1, circle with $d = 1$ parameter, 9 first eigenvectors (left to right and top to bottom) when ϕ = characteristic function.

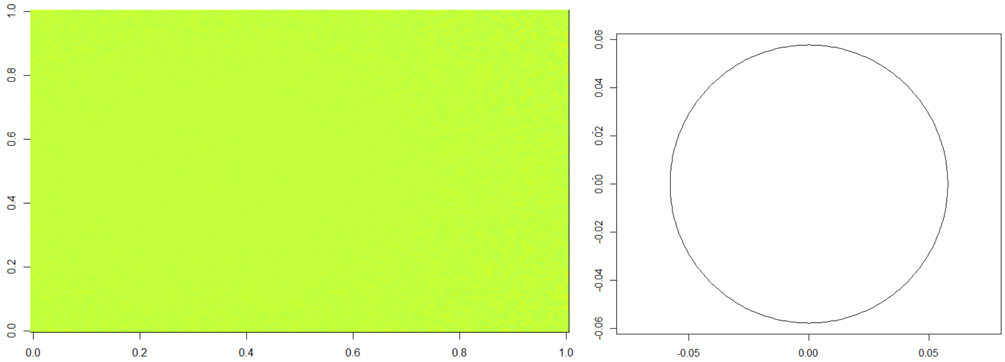


Figure 6.4: Example 6.1, circle with $d = 1$ parameter, first eigenvector when ϕ = signed distance (left) and when ϕ = contour discretization (right).

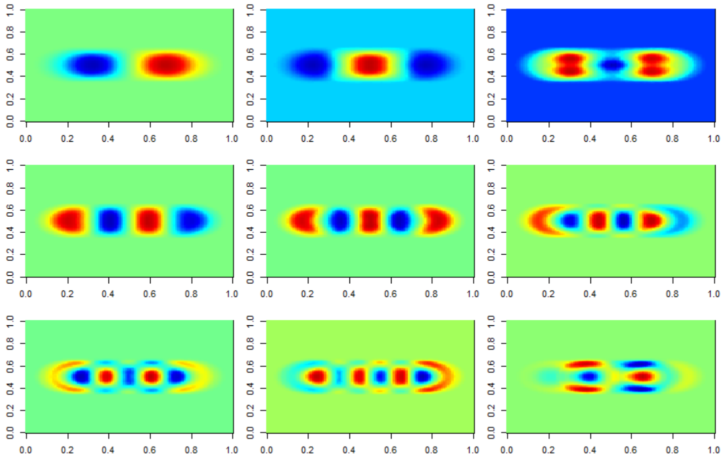


Figure 6.5: Example 6.1, circle with $d = 2$ parameters, 9 first eigenvectors (left to right and top to bottom) when ϕ = characteristic function.

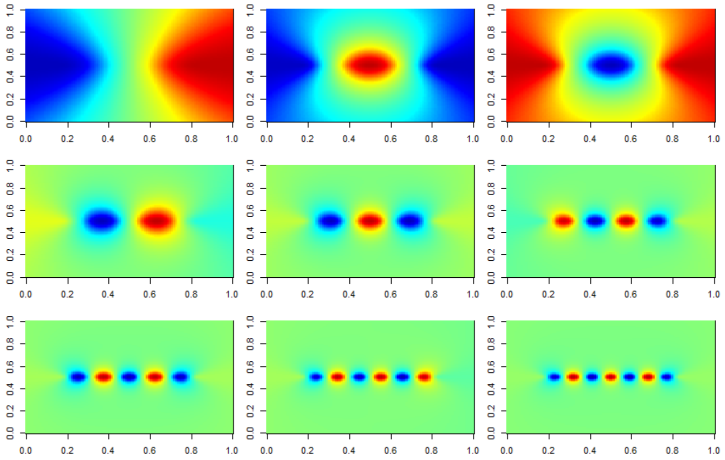


Figure 6.6: Example 6.1, circle with $d = 2$ parameters, 9 first eigenvectors (left to right and top to bottom) when ϕ = signed distance.

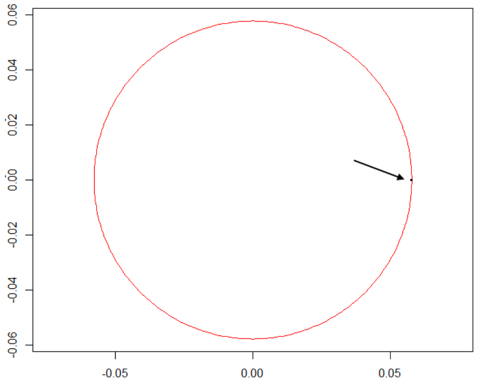


Figure 6.7: Example 6.1, circle with $d = 2$ parameters, 2 first eigenvectors (black and red) when $\phi =$ contour discretization.

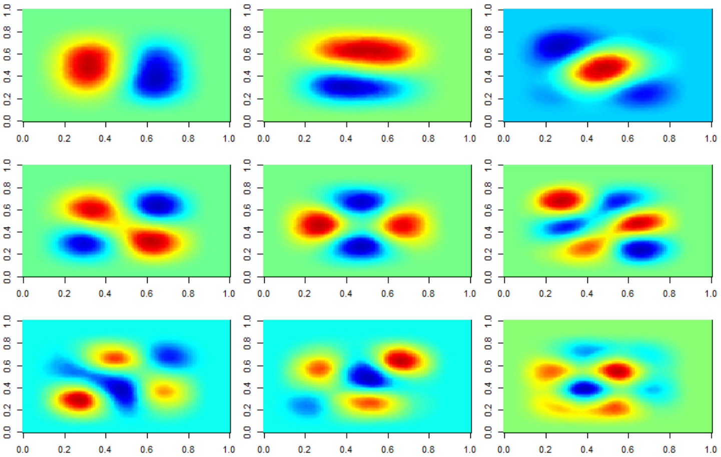


Figure 6.8: Example 6.1, circle with $d = 3$ parameters, 9 first eigenvectors (left to right and top to bottom) when $\phi =$ characteristic function.

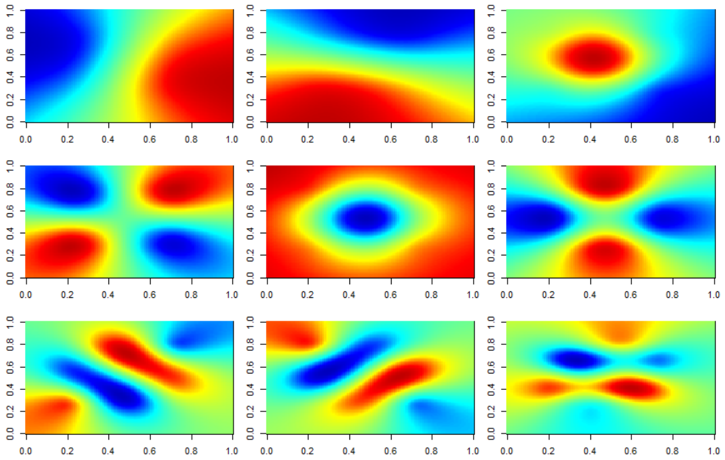


Figure 6.9: Example 6.1, circle with $d = 3$ parameters, 9 first eigenvectors (left to right and top to bottom) when $\phi =$ signed distance.

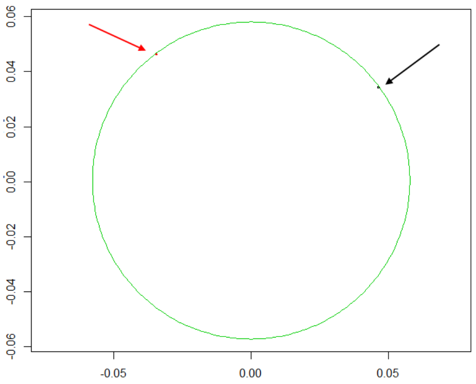


Figure 6.10: Example 6.1, circle with $d = 3$ parameters, 3 first eigenvectors (black, red, green) when $\phi =$ contour discretization.

A property of PCA is that a linear combination of the eigenvectors given in Equation (6.3) enables to retrieve any $\phi(\mathbf{x}^{(i)})$. Some of the eigenvectors are easy to interpret: in Figure 6.4 left (signed distance), the eigenvector is constant because the average shape is a map (an image) whose level lines are perfect circles so that adding a constant to it changes the radius of the null contour line; in Figure 6.7 where the mapping is a contour discretization, the first eigenvector (as well as the second in Figure 6.10) is a non-centered point that allows horizontal (and vertical) translations. The second (third in Figure 6.10) eigenvector is a circle which dilates or compresses the hole. As is seen in Tables 6.2 and 6.3, more eigenvectors are necessary for the characteristic function and for the signed distance than for the contour discretization. Contrarily to the characteristic function and the signed contour, when the mapping ϕ is the contour discretization, the first eigenvectors look like shapes on their own and therefore we will call them *eigenshapes*.

This does not mean however that all of them are valid shapes, as was seen in Figures 6.7 and 6.10 with the point vectors. In fact, most $\mathbf{v}^j$'s are "non-physical" in the sense that there may not exist one design $\mathbf{x}$ such that $\phi(\mathbf{x}) = \mathbf{v}^j$, see for instance Figure 6.26 where the eigenshapes do not correspond to a valid $\mathbf{x}$ from $\mathbf{v}^3$ on. In the case of the characteristic function, even though $\phi(\mathbf{x}) \in \{0,1\}^D$, the eigenvectors are real-valued (see Figure 6.3 for instance).

Example 6.2. *An over-parameterized hole in $\mathbb{R}^2$: the horizontal position of its center is $s := \sum_{j=1}^{13} x_j$, the vertical position of its center is $t := \sum_{j=14}^{26} x_j$ and its radius is $r := \sum_{j=27}^{39} x_j$, as shown in Figure 6.11. To increase the complexity of the problem, x_1, x_{14} and x_{27} are of a magnitude larger than the other x_j's: the circle mainly depends on these 3 parameters.*

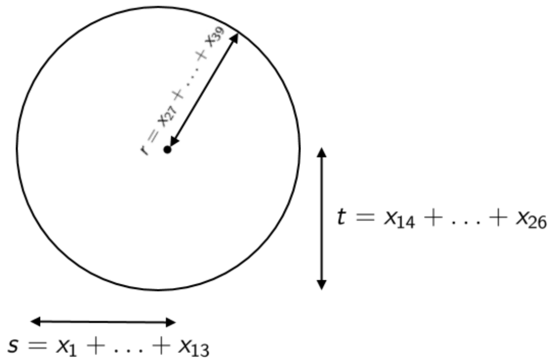

$r = x_{27} + \ldots + x_{39}$
$t = x_{14} + \ldots + x_{26}$
$s = x_1 + \ldots + x_{13}$

Figure 6.11: Second example: an over-parameterized circle.

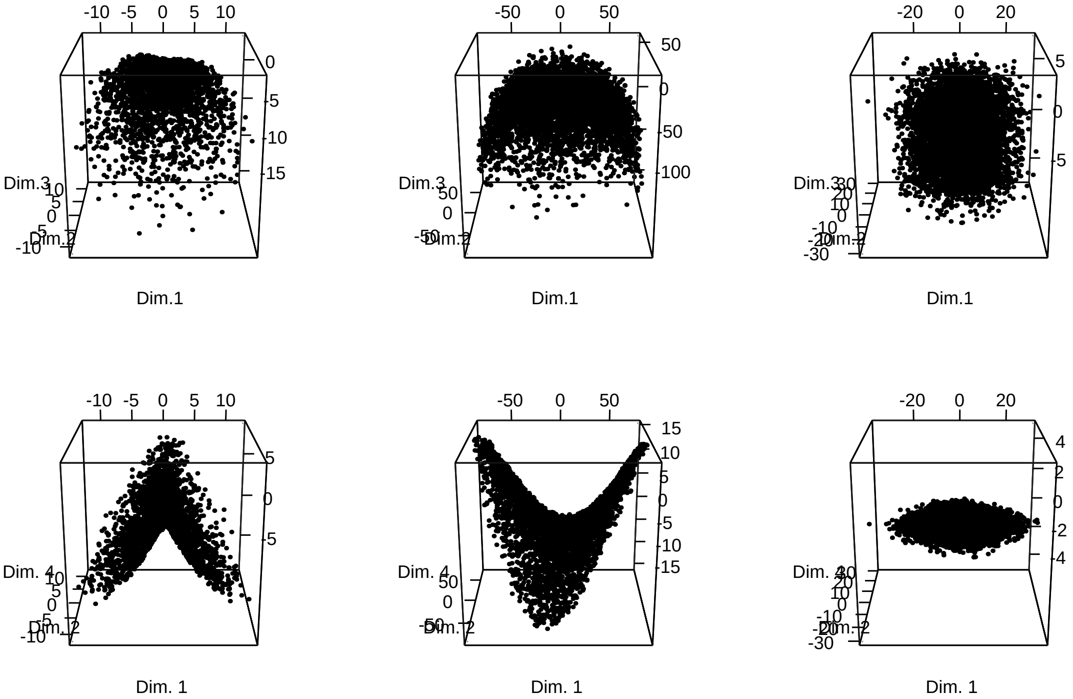


Figure 6.12: Four first eigencomponents of the $\boldsymbol{\alpha}^{(i)}$'s in the Example 6.2, for the three different shape representations ϕ. Left: characteristic function, middle: signed distance to the contour, right: discretization of the contour. The manifolds are shown in the $\{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3\}$ (top), and $\{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^4\}$ bases (bottom). As can be seen from the two-dimensional surface in the $\{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^4\}$ space when $\phi = \mathcal{D}$ (bottom right), the true dimension (3) is retrieved with the contour discretization. Note also that the associated manifold is convex.

The PCA eigenvalues for this example are given in Table 6.4 and are nearly the same as those in Table 6.3. Apart from the little modification in the uniform distribution for sampling the $\mathbf{x}^{(i)}$'s which might lead to a slightly different $\boldsymbol{\Phi}$, the over-parameterization is not a concern to retrieve the correct dimension. Figures 6.13-6.15 show the 9 first eigenvectors (if they have a strictly positive eigenvalue) for the three ϕ's.

	Characteristic function		Signed Distance		Discretization	
j	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage
1	9.24	9.48	1238.53	40.24	109.04	49.23
2	8.97	18.69	1210.72	79.57	104.69	96.50
3	8.76	27.68	516.05	96.33	7.75	100
4	5.95	33.79	39.70	97.62	0	100
5	5.28	39.21	24.47	98.42	0	100
6	3.93	43.25	21.83	99.13	0	100
7	3.59	46.93	6.10	99.33	0	100
8	3.36	50.38	6.03	99.52	0	100
9	2.90	53.35	3.27	99.63	0	100
10	2.80	56.23	3.12	99.73	0	100

Table 6.4: 10 first PCA eigenvalues for the different ϕ's, over-parameterized circle with $d = 39$ parameters, with real dimension $d = 3$.

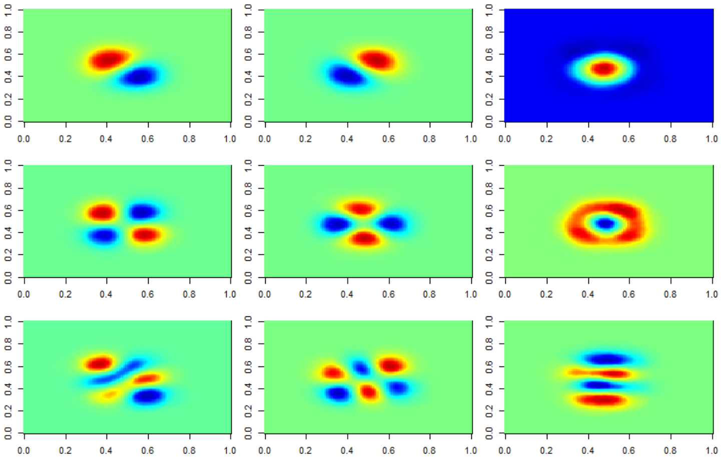


Figure 6.13: Example 6.2, over-parameterized circle with $d = 39$ parameters, 9 first eigenvectors (left to right and top to bottom) when ϕ = characteristic function.

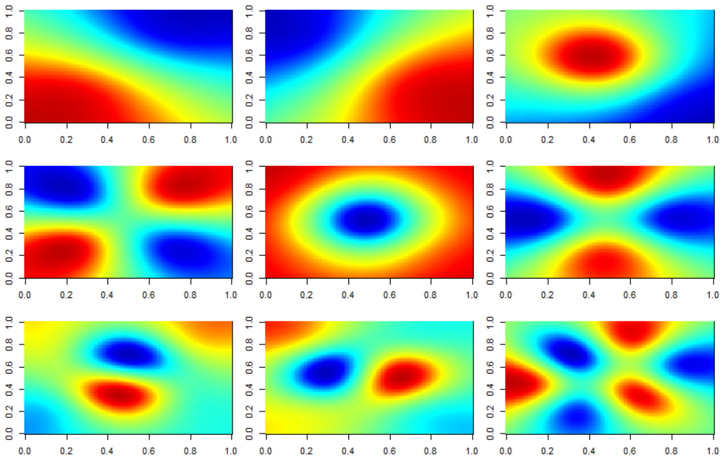


Figure 6.14: Example 6.2, over-parameterized circle with $d = 39$ parameters, 9 first eigenvectors (left to right and top to bottom) when $\phi =$ signed distance.

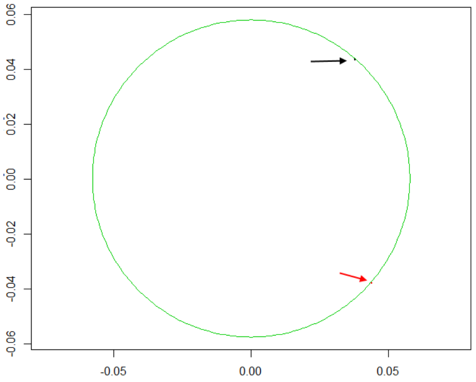


Figure 6.15: Example 6.2, over-parameterized circle with $d = 39$ parameters, 3 first eigenvectors when $\phi =$ contour discretization.

Example 6.3. *Three (non-overlapping) holes in $\mathbb{R}^2$, whose centers and radii are determined by x_1, x_2, x_3 (first circle), x_4, x_5, x_6 (second circle), and x_7, x_8, x_9 (third circle). This problem is more complex since it consists of three elements, and has $d = 9$ dimensions. For $\phi = \mathcal{D}$, the discretization vector $\phi(\mathbf{x}) \in \mathbb{R}^D$ is split into 3 parts of size $D/3$ which correspond to the discretization of each circle.*

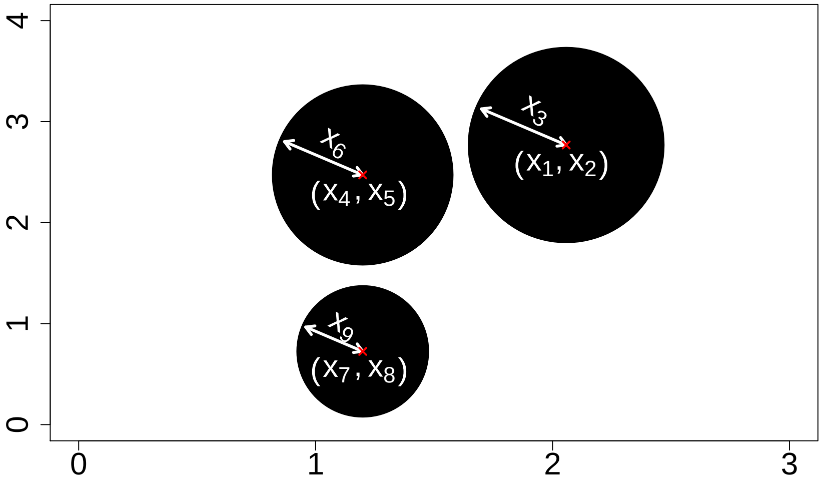


Figure 6.16: Third example: three circles with varying centers and radii.

	Characteristic function		Signed Distance		Discretization	
j	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage	Eigenvalue	Cumulative percentage
1	96.67	9.52	1785.93	31.51	154.26	19.06
2	81.57	17.56	1267.81	53.88	151.80	37.82
3	80.07	25.45	912.40	69.98	149.81	56.33
4	66.03	31.96	588.30	80.36	148.09	74.63
5	48.28	36.71	402.56	87.46	91.34	85.91
6	40.66	40.72	159.38	90.27	90.53	97.10
7	39.37	44.60	144.75	92.83	8.65	98.17
8	38.75	48.42	121.80	94.97	8.54	99.22
9	25.07	50.89	54.63	95.94	6.29	100
10	24.45	53.30	47.36	96.77	0	100

Table 6.5: 10 first PCA eigenvalues for the different ϕ's, three circles with $d = 9$ parameters.

The 9 first eigenvectors are illustrated for the three ϕ's in Figures 6.17 to 6.19.

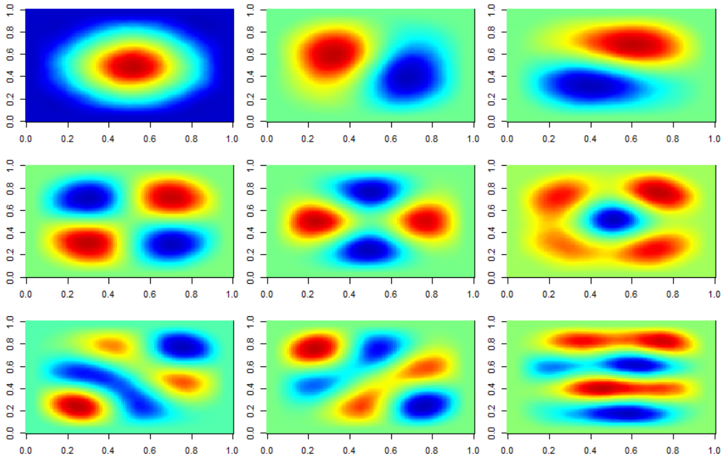


Figure 6.17: Example 6.3, three circles with $d = 9$ parameters, 9 first eigenvectors (left to right and top to bottom) when ϕ = characteristic function.

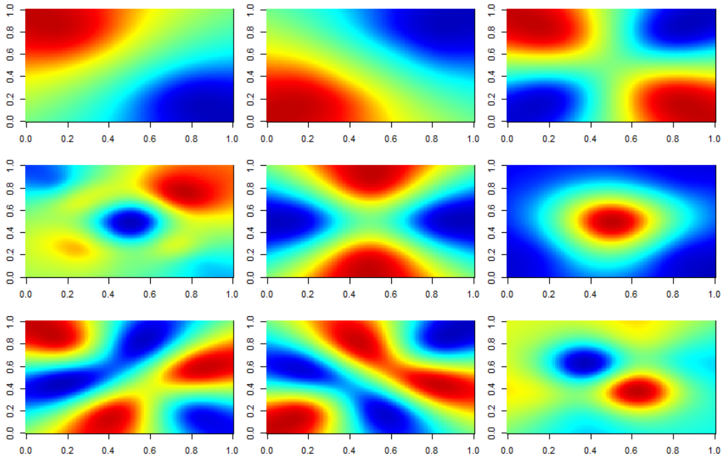


Figure 6.18: Example 6.3, three circles with $d = 9$ parameters, 9 first eigenvectors (left to right and top to bottom) when ϕ = signed distance.

In each example, for all $\phi(\cdot)$'s, any shape $\phi(\mathbf{x}^{(i)})$ can be reconstructed via Equation (6.3). $\boldsymbol{\alpha}^{(i)}$ is nonetheless D-dimensional hence no dimension reduction is obtained. We are therefore interested in low-rank approximations $\boldsymbol{\phi}_{1:\delta} := \overline{\boldsymbol{\phi}} + \sum_{j=1}^{\delta} \alpha_j \mathbf{v}^j$ which solely consider the δ first eigenvectors, while guaranteeing a sufficient precision. It is known (Jolliffe, 2011) that $\|\boldsymbol{\Phi} - \boldsymbol{\Phi}_{1:\delta}\|_F^2 = N \sum_{j=\delta+1}^{D} \lambda_j$ where $\boldsymbol{\Phi}_{1:\delta}$ is the reconstruction matrix using the δ first principal axes $\mathbf{v}^j$ only, and whose i-th row is $\overline{\boldsymbol{\phi}} + \sum_{j=1}^{\delta} \alpha_j^{(i)} \mathbf{v}^j$. $\boldsymbol{\Phi}_{1:\delta}$ is also known to be the closest (in terms of Frobenius norm) matrix to $\boldsymbol{\Phi}$ with rank lower or equal to δ. The λ_j's with $j > \delta$ inform us about the reconstruction loss. Hence, we look for a mapping $\phi(\cdot)$ for which the λ_j quickly go to zero. In Tables 6.1 to 6.5, the vanishing of

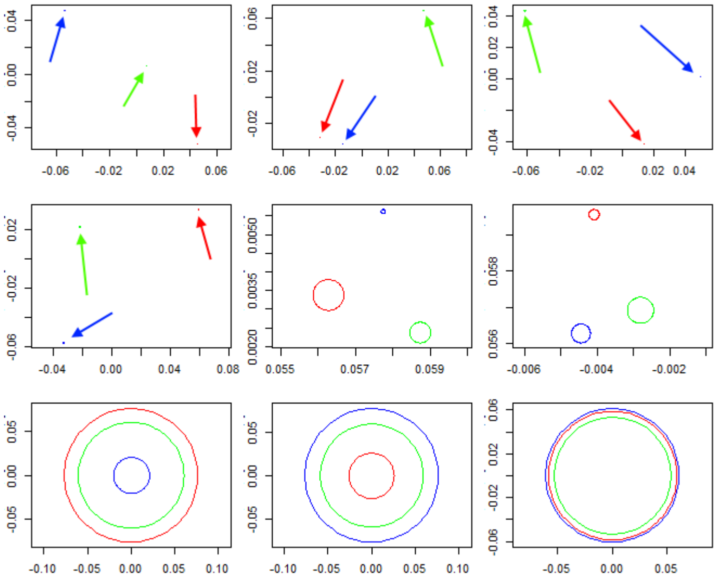


Figure 6.19: Example 6.3, three circles with $d = 9$ parameters, 9 first eigenvectors (from left to right, top to bottom) when $\phi =$ discretization. The blue part of each eigenvector acts on the first circle, the red part of each eigenvector modifies the second circle and the green part of each eigenvector applies on the third circle.

λ_j beyond the intrinsic dimension only happens when $\phi = \mathcal{D}$. With the other mappings, alternative techniques relying on local PCAs (Fukunaga and Olsen, 1971) on the $\boldsymbol{\alpha}^{(i)}$'s are required to estimate the dimensionality of manifolds such as the ones on the top row of Figure 6.2. The d first principal components, $\boldsymbol{\alpha}^{(i)}_{1:d}$ suffice to reconstruct $\phi(\mathbf{x}^{(i)})$ exactly using $\mathcal{D}$ as the $\phi(\cdot)$ mapping, while more than d components are required for $\phi(\mathbf{x}^{(i)})$ to be recovered using χ or $\mathbb{D}$. With $\mathcal{D}$, the eigenvectors $\mathbf{v}^j$ (Right plot of Figure 6.4, Figures 6.7, 6.10, 6.15 and 6.19) are physically meaningful: they can be interpreted as shape discretizations, which, being multiplied by coefficients α_j and added to the mean shape $\overline{\boldsymbol{\phi}}$, act on the hole's size (Eigenvector 1 in right plot of Figure 6.4, Eigenvector 2 in Figure 6.7, Eigenvector 3 in Figure 6.10, Eigenvector 3 in Figure 6.15, Eigenvectors 7-9 in Figure 6.19), or on the hole's position (Eigenvector 1 in Figure 6.7, Eigenvectors 1-2 in Figure 6.10, Eigenvectors 1-2 in Figure 6.15, Eigenvectors 1-6 in Figure 6.19). For example, very small eigenvectors such as the first one in Figure 6.7 displace the shape in the direction specified by the eigenvector's position. In Figure 6.19, the first eigenvectors tend to move each circle with respect to each other, while the sizes of the holes are affected by the last eigenvectors. Whereas the characteristic function χ and the signed distance $\mathbb{D}$ are images, the mapping $\mathcal{D}$ is a discretization of the final object we represent, a contour shape. Without formal proof, we think that this is related to the observed property that the d (the number of intrinsic dimensions) first eigencomponents $\boldsymbol{\alpha}^{(i)}_{1:d}$, $i = 1, \dots, N$ make a convex set as can be seen in Figures 6.2 and 6.12.

In a solid mechanics analogy, the $\overline{\boldsymbol{\phi}} + \sum_j \alpha_j \mathbf{v}^j$ reconstruction can be thought as a sum of pressure fields $\mathbf{v}^j$ applied on each node of the Point Distribution Model, and which deform the initial mean shape $\overline{\boldsymbol{\phi}}$ by a magnitude α_j to obtain $\boldsymbol{\phi}$. Such an interpretation cannot be conducted with the eigenvectors obtained via the χ or $\mathbb{D}$ mapping, shown in the other figures.

Because of its clear pre-eminence, in the following, we will only consider the $\boldsymbol{\alpha}$'s obtained using the contour discretization as ϕ mapping.

6.2.3.2 Hierarchic shape basis for the reduction of high-dimensional designs

Following these observations, we now deal with slightly more complex and realistic shapes $\Omega_{\mathbf{x}}$. Even though they are initially described with many parameters, they mainly depend on few intrinsic dimensions.

Example 6.4. *A rectangle ABCD with* $\mathbf{x} \in \mathbb{R}^{40}$ *whose parameters* x_1 *and* x_2 *are the location of A,* x_3 *and* x_4 *are the width and the height of ABCD, and* $\mathbf{x}_{5:13}$*,* $\mathbf{x}_{14:22}$*,* $\mathbf{x}_{23:31}$ *and* $\mathbf{x}_{32:40}$ *are small evenly distributed perturbations, on the AB, BC, CD and DA segments, respectively.*

$x_1, \dots, x_4$ are of a magnitude larger than the other parameters to ensure a close-to-rectangular shape, as shown in Figure 6.20.

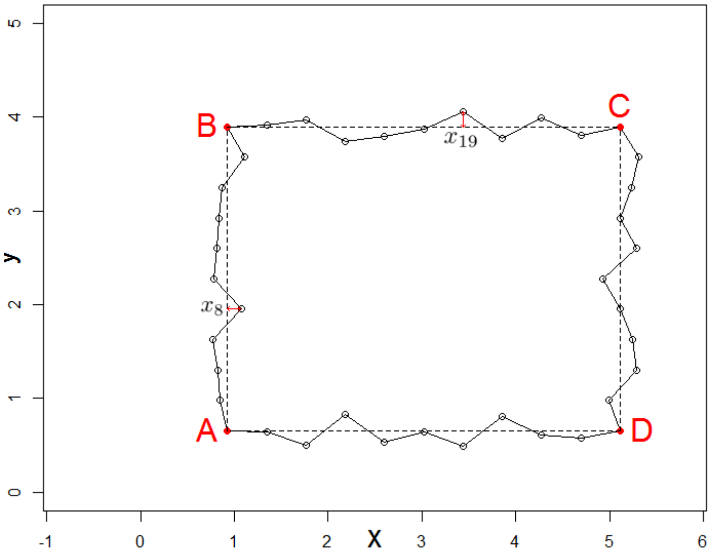


Figure 6.20: Example 6.4: a rectangle with varying position, size, and deformation of its sides.

In this example where 4 parameters (position and sizes) mainly explain the differences among shapes, we see that a reconstruction quality of 99.83% is attained with the 4 first eigenvectors $\mathbf{v}^j$.

Figure 6.21 details the eigenvectors. $\mathbf{v}^1$ and $\mathbf{v}^2$, the most influencing eigenshapes plotted in black and blue act as translations, while $\mathbf{v}^3$ and $\mathbf{v}^4$ (in red and green) correspond to

j	Eigenvalue	Cumulative percentage
1	867.65	48.73
2	866.90	97.42
3	21.46	98.62
4	21.43	99.83
5	0.13	99.83
6	0.13	99.84
7	0.13	99.85
8	0.13	99.86
9	0.12	99.86
10	0.12	99.87
⋮	⋮	⋮
39	0.04	99.99
40	0.04	100
41	0	100

Table 6.6: First PCA eigenvalues for ϕ = discretization, rectangles with $d = 40$ parameters (Example 6.4).

widening and heightening of the rectangle. The fluctuations along the segments appear from the 5th eigenshape on. Any shape is retrieved with the $d = 40$ first eigenshapes which corresponds to the total number of parameters.

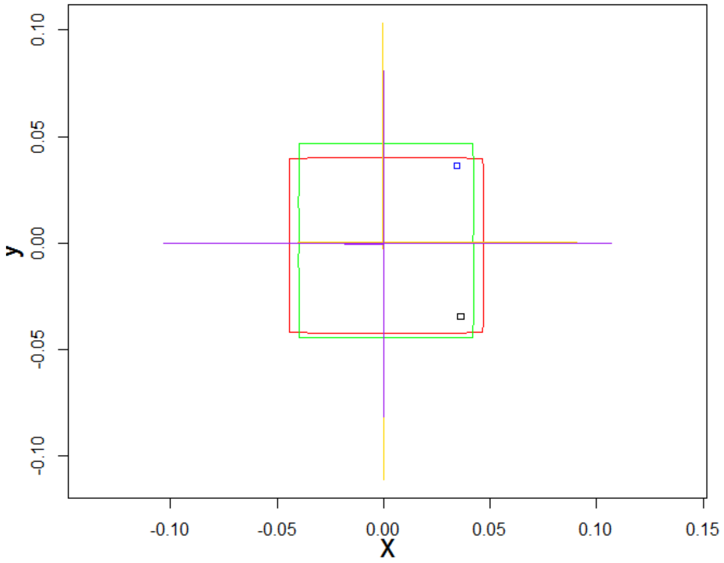


Figure 6.21: 6 first eigenshapes (in the order black, blue, red, green, yellow, purple) of the rectangles in Example 6.4.

Example 6.5. *A straight line joining two fixed points A and B, modified by smooth perturbations* $\mathbf{r} \in \mathbb{R}^{29}$*, evenly distributed along [AB] to approximate a smooth curve.*

The fifth example is inspired by the catenoid problem (Colding and Minicozzi, 2006). The perturbations $\mathbf{r}$ are generated by a Gaussian Process with squared exponential kernel and with length-scale 6 times smaller than [AB]. Therefore, in this example, the $N = 5000$ $\mathbf{r}^{(i)}$'s used for building $\boldsymbol{\Phi}$ are not uniformly distributed in X.

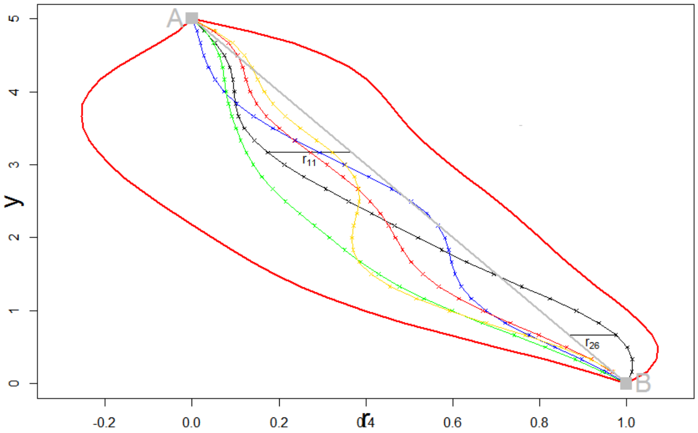


Figure 6.22: Example 6.5: a straight line joining two points, modified by the perturbations r_j to approximate a curve. Gray: the line joining A and B. Blue, red, yellow and green curve: examples of lines with regular r_j perturbations. Red envelope: boundaries for the r_j's.

j	Eigenvalue	Cumulative percentage
1	2.156	50.258
2	1.251	79.422
3	0.590	93.181
4	0.206	97.973
5	0.065	99.480
6	0.017	99.882
7	0.004	99.975
8	0.001	99.995
9	ε	99.999
10	ε	100
$\vdots$	$\vdots$	$\vdots$
28	ε	100
29	ε	100
30	0	100

Table 6.7: First PCA eigenvalues for ϕ = discretization, curve with $d = 29$ parameters. ε means the quantity is not exactly 0, but smaller than 10^{-3}, hence less than 0.04% of the first PCA eigenvalue.

Again, the initial dimension ($d = 29$) is recovered by looking at the strictly positive eigenvalues. Furthermore, the manifold is found to mainly lie in a lower dimensional space: $\mathcal{A}_N$ can approximated in $\delta = 7$ dimensions since $\sum_{j=1}^{\delta} \lambda_j / \sum_{j=1}^{D} \lambda_j = 99.975\%$.

Figure 6.23 shows the corresponding eigenshapes. The eigenshapes are similar to the ordered modes of the harmonic series with the associated eigenvalues ordered as the inverse of the frequencies.

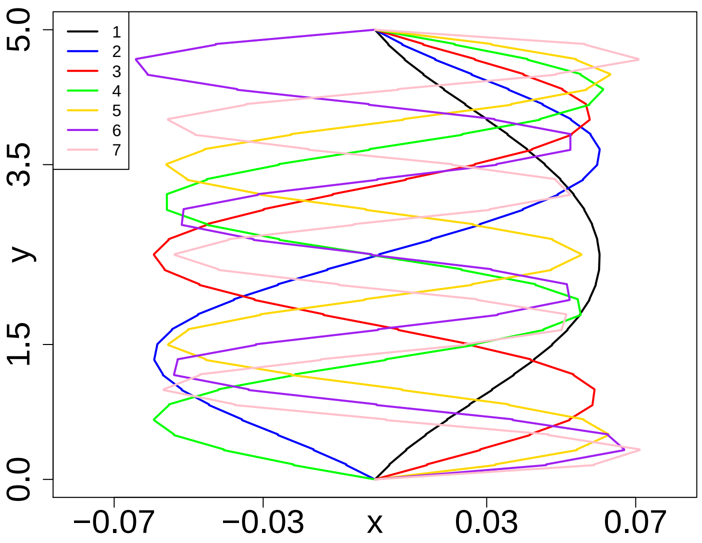


Figure 6.23: 7 first eigenshapes for the curves of Example 6.5.

Example 6.6. *A classical NACA airfoil parameterized by three parameters:* $\mathbf{x} = (M, P, T)^\top \in \mathbb{R}^3$ *(see Section 3.1 for a detailed description).*

j	Eigenvalue	Cumulative percentage
1	0.2819	54.619
2	0.2203	97.318
3	0.0129	99.814
4	0.0008	99.959
5	0.0001	99.983
6	ε	99.991
7	ε	99.996
8	ε	99.997
9	ε	99.999
10	ε	99.999

Table 6.8: First PCA eigenvalues of the NACA airfoil with $d = 3$ parameters (ϕ is the contour discretization). ε means the quantity is smaller than 10^{-4}, hence less than 0.04% of the first PCA eigenvalue.

In this example, a typical noise-truncation criterion such as discussed in Example 6.5 would retain 3 or 4 axes. In Example 6.6 too, the effective dimension can almost be retrieved from the λ's.

Figure 6.24 shows the 4 first eigenshapes (left) as well as the $\mathcal{A}_N$ manifold (right). The eigenvectors can be interpreted as a reformulation of the CAD parameters. The first eigenshape (blue) is a symmetric airfoil. Multiplying it by a coefficient (after adding it to the black mean shape) will increase or decrease the thickness of the airfoil, hence it plays a similar role to the T parameter. The second eigenshape is a cambered airfoil, whose role is similar to M (maximum camber). Last, the third airfoil, which has a much smaller eigenvalue λ_3, is very thin, positive in the first part of the airfoil, and negative in its second part. It balances the camber of the airfoil towards the leading edge or towards the rear and plays a role similar to P, the position of the maximum camber. $\mathbf{v}^3$'s effect is complemented by $\mathbf{v}^4$.

The analysis of $\mathcal{A}_N$ (Figure 6.24) is physically meaningful: even though $\mathbf{x}^{(i)}$ are sampled uniformly in X, $\mathcal{A}_N$ resembles a pyramid in the $(\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3)$ basis. Designs with minimal α_2 share the same α_3 value. Since negative α_2's correspond to wings with little camber, the position of this maximum camber has very little impact, hence the almost null α_3 value. By looking at $\mathcal{A}_N$, it is learned that the parameter P does not matter when M is small, which is intuitive but is not expressed by the (M, P, T) coordinates. Distances in $\mathcal{A}_N$ are therefore more representative of shape differences. An additional advantage of analyzing shapes is that correlations in the space of parameters (such as the one between M and P in this example) are discovered and removed, since $\mathcal{V}$ is an orthonormal basis. Here, orthogonality between eigenshapes is measured by the standard scalar product in $\mathbb{R}^D$. Depending on the application, there may exist natural definitions of the orthogonality between discretized shapes, which could be used by the PCA.

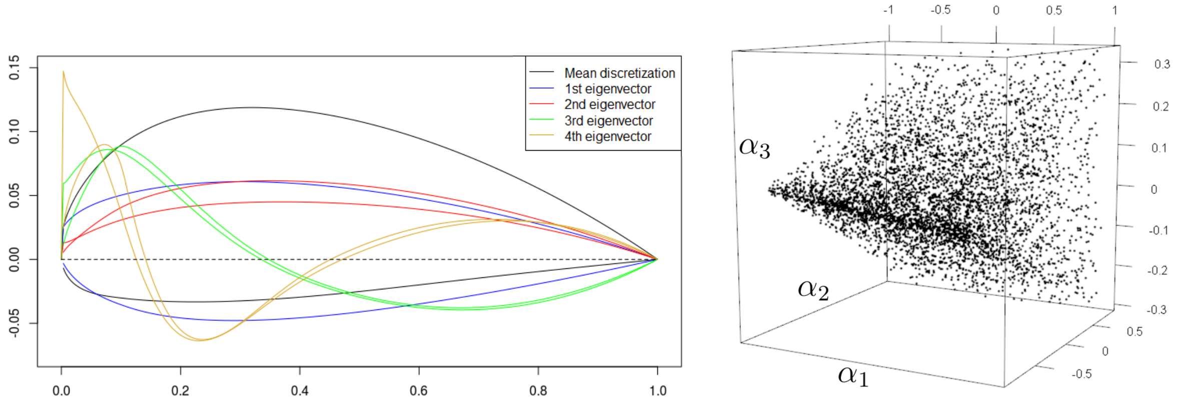


Figure 6.24: NACA airfoil with $d = 3$ parameters. Left: mean shape and 4 first eigenshapes (black, blue, red, green, yellow). Right: three first eigencomponents $(\alpha_1, \alpha_2, \alpha_3)$ of the $\mathcal{A}_N$ manifold.

Example 6.7. *The modified NACA airfoil with $d = 22$ parameters, $\mathbf{x} = (M, P, T, L_1, \ldots, L_{19})^\top \in \mathbb{R}^{22}$ (see Section 3.1.1 and Figure 3.2 for a detailed description).*

j	Eigenvalue	Cumulative percentage
1	0.2826	53.932
2	0.2205	96.021
3	0.0134	98.580
4	0.0011	98.798
5	0.0006	98.903
6	0.0005	99.006
7	0.0005	99.106
8	0.0005	99.202
9	0.0005	99.293
10	0.0004	99.377
$\vdots$	$\vdots$	$\vdots$
19	0.003	99.958
20	0.002	99.992
21	ε	99.995
22	ε	99.998
23	ε	99.999

Table 6.9: First PCA eigenvalues for $\phi =$ discretization, NACA with $d = 22$ parameters. ε means the quantity is not exactly 0, but smaller than 10^{-4}, hence less than 0.04% of the first PCA eigenvalue.

Here, as in the Example 6.6, the noise-truncation criteria will retain between 6 and 20 dimensions, depending on the reconstruction quality required. Indeed, when looking at specimen of NACA 22 airfoils as the one in the upper left part of Figure 6.25, less than 22 dimensions are expected to be necessary to retrieve an approximation of sufficient quality.

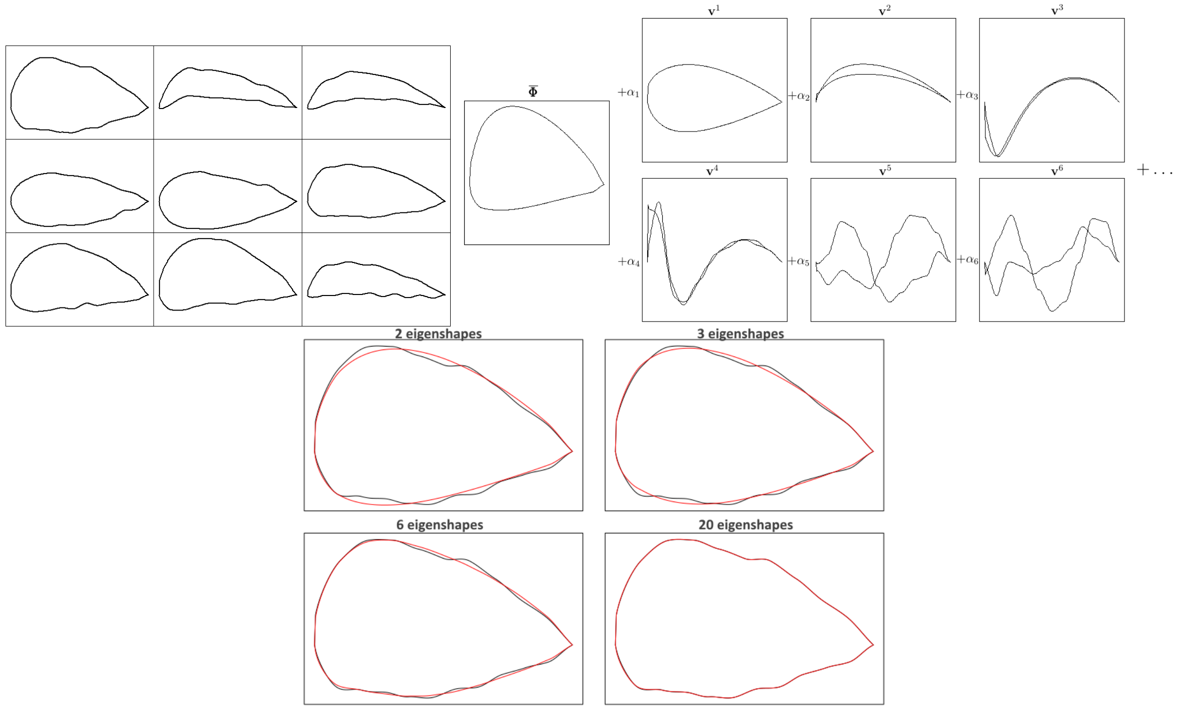


Figure 6.25: Left: examples of NACA 22 airfoils. Even though the true dimension is 22, less dimensions may suffice to approximate the shapes well enough. Right: reconstruction scheme of any NACA 22 shape: a weighted deviation from the mean shape $\overline{\boldsymbol{\phi}}$ in the direction of the eigenshapes. Bottom: example of shape reconstruction (red) using 2, 3, 6 or 20 eigenshapes. The more $\mathbf{v}^j$'s, the better the reconstruction but the larger the dimension of $\boldsymbol{\alpha}$.

The analysis of eigenshapes, shown in Figure 6.26, is similar to the one of Example 6.6. Small details that act on the airfoil such as the bumps only appear from the 4th eigenshape on. Not taking them into account leads to a weaker reconstruction, as shown in the bottom part of Figure 6.25.

According to these experiments, the eigenvectors $\mathbf{v}^j$, $j \in \{d+1, \ldots, D\}$, can already be discarded without even considering the values of the associated objective functions since the d first shape modes explain the whole variability of the discretized shapes. In practice, to filter numerical noise and to remove non-informative modes in shapes that are truly over-parameterized, we only consider the d' first eigenshapes, $d' := \min(d, \tilde{d})$ where $\tilde{d}$ corresponds to the smallest number of axes that explain more than a given level of diversity in Φ (e.g. 99.9, 99.95 or 99.99%), measured by $100 \times \sum_{j=1}^{\tilde{d}} \lambda_j / \sum_{j=1}^{D} \lambda_j$. Another alternative is to define $\tilde{d}$ according to the dimensions for which λ_j/λ_1 is smaller than a prescribed threshold (e.g. 1/1000). Even though the notation D is kept, the eigenvectors $\mathbf{v}^j$ and the principal components α_j, are considered to be null $\forall j > d'$ so that in fact $D = d'$ in the following.

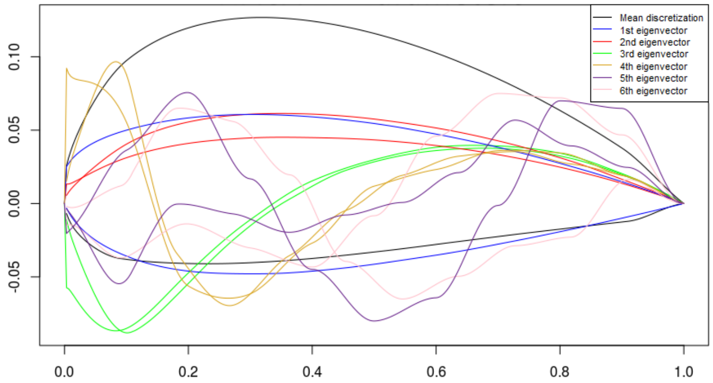


Figure 6.26: Mean shape (black) and 6 first eigenshapes (blue, red, green, yellow, purple, pink) for the NACA with 22 parameters. The three first eigenvectors are similar to those observed on Figure 6.24 for the original NACA 3. Fluctuations along the eigenshapes are found from the 4th eigenshape on. They allow to reconstruct the local refinements (bumps) of the airfoils.

6.3 GP models for reduced eigenspaces

Building a surrogate model in the space of principal components has already been investigated in the context of reduced order models (Berkooz et al., 1993). In most applications, the dimension reduction is carried out in the output space, which has large dimension when it corresponds to values on a finite element mesh. The response is approximated by a linear combination of a small number of modes, and the metamodel is a function of the modes coefficients. The construction of surrogates with inherent dimensionality reduction has also been considered. In the active subspace method (Constantine et al., 2014), the dimension reduction comes from a linear combination of the inputs which is carried out by projecting $\mathbf{x}$ onto the hyperplane spanned by the directions of largest $\nabla f(\mathbf{x})$ variation. The reduced-dimension GP is then $Y(\mathbf{W}^\top \mathbf{x})$ with $\mathbf{W} \in \mathbb{R}^{d\times\delta}$ containing these directions in columns. In Palar and Shimoyama (2018), cross-validation is employed for choosing the number of such axes. An application to airfoils is given in Li et al. (2019) where the authors take the directions of largest drag and lift gradients as columns of $\mathbf{W}$, even though this basis is no longer orthogonal. Another related technique with a $Y(\mathbf{W}^\top \mathbf{x})$ GP which does not require the knowledge of $\nabla f(\mathbf{x})$ is the Kriging and Partial Least Squares (KPLS) method (Bouhlel et al., 2016), where $\mathbf{x}$ is projected onto the hyperplane spanned by the first δ axes of a PLS regression (Frank and Friedman, 1993). The dimension reduction is output-driven but $\mathbf{W}$ is no longer orthogonal, and information may be lost when $n < d'$ because any shape (of effective dimension d') cannot be exactly reconstructed (Equation 6.3) with these n vectors. Coordinates in the PLS space are therefore incomplete and metamodeling loses precision when n is too small. In the same spirit, a double maximum-likelihood procedure is developed in Tripathy et al. (2016) to build an output-related and orthogonal matrix $\mathbf{W}$ for the construction of a Gaussian Process with built-in dimensionality reduction. Rotating the design space through hyperparameters determined by maximum likelihood is also performed in Namura et al. (2017b).

6.3.1 Unsupervised dimension reduction

Instead of the space of CAD parameters $\mathbf{x}$, we reduce the dimension of the input space by building the surrogate with information from the space of shape representations, Φ, as in Li et al. (2018a). To circumvent the high dimensionality of $\Phi \subset \mathbb{R}^D$, a linear dimension reduction of $\phi(\mathbf{x})$ is achieved by building the model in the space spanned by $\mathbf{W}^\top \phi(\mathbf{x})$. A natural candidate for $\mathbf{W}$ is a restriction to few columns (eigenshapes) of the matrix $\mathbf{V}$. Notice that contrarily to the other dimension reduction techniques which operate a linear dimension reduction of $\mathbf{x}$, this approach is nonlinear in $\mathbf{x}$ since it operates linearly on the nonlinear transformation $\phi(\mathbf{x})$. Also, it operates on a better suited representation of the designs, their shapes, instead of their parameters.

A first idea to reduce the dimension of the problem is to conserve the δ first eigenvectors $\mathbf{v}^j$ according to some reconstruction quality criterion measured by the eigenvalues. Given a threshold T (e.g., 0.95 or 0.99), only the first δ modes such that $\frac{\sum_{j=1}^{\delta} \lambda_j}{\sum_{j=1}^{D} \lambda_j} > T$ are retained in $\mathbf{V}_{1:\delta} \in \mathbb{R}^{D \times \delta}$ because they contribute for $100 \times T\%$ of the variance in Φ. The surrogate model is implemented in the space of the δ first principal components as

$$Y(\boldsymbol{\alpha}_{1:\delta}) = Y(\mathbf{V}_{1:\delta}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})). \tag{6.5}$$

Using a stationary kernel for the $Y(\boldsymbol{\alpha}_{1:\delta})$ GP, i.e. $k(\boldsymbol{\alpha}_{1:\delta}, \boldsymbol{\alpha}'_{1:\delta}) = \tilde{k}(\|\boldsymbol{\alpha}_{1:\delta} - \boldsymbol{\alpha}'_{1:\delta}\|_{\mathbb{R}^\delta})$, the correlation between designs is $k(\boldsymbol{\alpha}_{1:\delta}, \boldsymbol{\alpha}'_{1:\delta}) = \tilde{k}(\|\mathbf{V}_{1:\delta}^\top(\phi(\mathbf{x}) - \phi(\mathbf{x}'))\|_{\mathbb{R}^\delta}) = \tilde{k}(r)$ with $r^2 = (\phi(\mathbf{x}) - \phi(\mathbf{x}'))^\top \mathbf{M}(\phi(\mathbf{x}) - \phi(\mathbf{x}'))$ where $\mathbf{M} = \mathbf{V}_{1:\delta}\mathbf{V}_{1:\delta}^\top$ is a $D \times D$ matrix with low rank (δ). Hence, this model implements a Gaussian Process in the Φ space with an integrated linear dimensionality reduction step (Rasmussen and Williams, 2006). Note that the kernel is non-stationary in the original X space.

The approaches of Bouhlel et al. (2016); Constantine et al. (2014); Tripathy et al. (2016) mainly differ from that proposed in Equation (6.5) in the construction of the reduced basis: in Equation (6.5), dimension reduction is carried out without the need to call the expensive $f(\mathbf{x})$ (or its gradient): the directions of largest variation of an easy to compute mapping $\phi(\cdot)$ are used instead. This also prevents from a spurious or incomplete projection when n is smaller than D and avoids recomputing the basis at each iteration.

This is nonetheless a limitation since the $Y(\boldsymbol{\alpha}_{1:\delta})$ approach relies only on considerations about the shape geometry. The output y is not taken into account for the dimension reduction even though some $\mathbf{v}^j$, $j \in \{1, \dots, \delta\}$ may influence y or not. Two shapes which differ in the α_j components with $j \leq \delta$ may behave similarly in terms of output y, so that further dimension reduction is possible. Vice versa, eigencomponents that have a small geometrical effect and were neglected may be reintroduced because they matter for y.

As an illustration consider the red and black shapes of Figure 6.27. Both are associated to parameters $\mathbf{x}$ and $\mathbf{x}'$ and their discretizations $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ are quite different.

Depending on the objective function, $f(\mathbf{x})$ and $f(\mathbf{x}')$ might differ widely. However, when considering the $\overline{\boldsymbol{\phi}} + \sum_{j=1}^{\delta} \alpha_j \mathbf{v}^j$ reconstruction with $\delta = 3$, they look very similar because $\boldsymbol{\alpha}_{1:3} \approx \boldsymbol{\alpha}'_{1:3}$. Even though $\mathcal{V}_{1:3} := \{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3\}$ is a tempting basis because it explains 98.5% of the discretizations variance, it is not a good choice if $f(\mathbf{x})$ and $f(\mathbf{x}')$ are different: because of continuity assumptions a surrogate model would typically suffer from inputs $\boldsymbol{\alpha} \approx \boldsymbol{\alpha}'$ with $y \neq y'$.

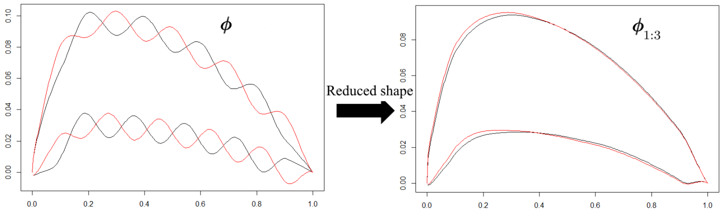


Figure 6.27: Example of two different shapes (black and red) whose reconstruction in the space of the three first eigenshapes is very similar.

For this reason, instead of building the surrogate in the space spanned by the most relevant shape modes, we would prefer to build it in the $\mathcal{V}_a \subset \mathcal{V}$ basis of the most output-influencing eigenshapes $\boldsymbol{\alpha}^a$. Additionally, since the remaining "inactive" components $\boldsymbol{\alpha}^{\overline{a}}$ refine the shape and might explain small fluctuations of y, instead of omitting them (which is equivalent to stating $\boldsymbol{\alpha}^{\overline{a}} = \mathbf{0}$), we would like to keep them in the surrogate model while prioritizing $\boldsymbol{\alpha}^a$: a GP $Y^a(\mathbf{W}_a\phi(\mathbf{x})) + Y^{\overline{a}}(\mathbf{W}_a\phi(\mathbf{x}))$ is detailed in Sec. 6.3.2.2.

6.3.2 Supervised dimension reduction

6.3.2.1 Selection of active eigenshapes

To select the eigencomponents that impact y the most, the penalized log-likelihood (Yi et al., 2011) of a regular, anisotropic GP in the high dimensional space of $\boldsymbol{\alpha}$'s is considered,

$$\max_{\vartheta} pl_\lambda(\boldsymbol{\alpha}^{(1:t)}, y^{(1:t)}; \vartheta) \quad \text{where} \quad pl_\lambda(\boldsymbol{\alpha}^{(1:t)}, y^{(1:t)}; \vartheta) := \widehat{l}(\boldsymbol{\alpha}^{(1:t)}, y^{(1:t)}; \vartheta) - \lambda \|\boldsymbol{\theta}^{-1}\|_1 \quad (6.6)$$

The ϑ are the GP's hyperparameters made of the length-scales θ_j, a constant mean term β, and the variance of the GP σ^2. $\boldsymbol{\alpha}^{(1:t)}$ are the eigencomponents of the evaluated designs $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}$, and $y^{(1:t)}$ the associated outputs, $y^{(1:t)} = (y^{(1)}, \ldots, y^{(t)})^\top = (f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(t)}))^\top$. The mean and the variance terms can be solved for analytically by setting the derivative of the log-likelihood equal to 0 (cf. Section 2.1.3) and are substituted in (6.6) by $\widehat{\beta}$ and $\widehat{\sigma}^2$ which yields the (concentrated) penalized log-likelihood

$$pl_\lambda(\boldsymbol{\alpha}^{(1:t)}, y^{(1:t)}; \vartheta) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log(|\mathbf{K}_\vartheta|) - \frac{1}{2}(y^{(1:t)} - \mathbf{1}\widehat{\beta})^\top \mathbf{K}_\vartheta^{-1}(y^{(1:t)} - \mathbf{1}\widehat{\beta}) - \lambda\|\boldsymbol{\theta}^{-1}\|_1 \quad (6.7)$$

where $\mathbf{K}_\vartheta$ is the covariance matrix with entries $K_{\vartheta_{ij}} = \widehat{\sigma^2} k_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, with determinant $|\mathbf{K}_\vartheta|$. The penalization is applied to $\boldsymbol{\theta}^{-1} := (1/\theta_1, \ldots, 1/\theta_D)^\top$, the vector containing the inverse length-scales of the GP. It is indeed clear (Ben Salem et al., 2018) that if $\theta_j \to +\infty$, the direction $\mathbf{v}^j$ has no influence on y as all the points are perfectly correlated together, making the GP flat in this dimension. The L^1 penalty term applied to the θ_j's performs variable selection: this Lasso-like procedure promotes zeros in the vector of inverse length-scales, hence sets many θ_j's to $+\infty$. Few directions with small θ_j are selected. Even if the maximization of pl_λ is carried out in a D-dimensional space, the problem is tractable since the gradients of pl_λ are analytically known, and because the L^1 penalty convexifies the problem. We solve it using standard gradient-based techniques such as BFGS (Liu and Nocedal, 1989) with multistart.

Numerical experiments not reported here for reasons of brevity have shown that most local optima to this problem solely differ in θ_j's that are already too large to be relevant and consistently yield the same set of active variables $\boldsymbol{\alpha}^a$. Notice that in Yi et al. (2011), a similar approach is undertaken but the penalization was applied on the reciprocal variables $\mathbf{w} = (w_1, \dots, w_D)^\top$ with $w_j = 1/\theta_j$. In our work, the inverse length-scales are penalized, the gradient of the penalty is proportional to $1/\theta_j^2$. This might help the optimizer since directions with θ_j's that are not large yet are given more emphasis. In comparison, the $\mathbf{w}$ penalty function's gradient is isotropic. Since we can restrict the number of variables to $d' \ll D$ with no loss of information (cf. discussion at the end of Section 6.2.3), the dimension of Problem (6.6) is substantially reduced which leads to a more efficient resolution. Because the α_j's have zero mean and variance λ_j, they have magnitudes that decrease with j. When $m < n$, $1/\theta_n$ is typically larger than $1/\theta_m$, meaning that the optimizer is better rewarded by diminishing $1/\theta_n$ than $1/\theta_m$. Starting from reasonable θ_j values[2] the first θ_j's are therefore less likely to be increased in comparison with the last ones, i.e. they are less likely to be found inactive. This can be seen as a bias which can be removed by scaling all α_j's to the same interval. However, we do not normalize the $\boldsymbol{\alpha}$ variables for two reasons. First, since the α_j's correspond to reconstruction coefficients associated to normalized eigenshapes ($\|\mathbf{v}^j\|_{\mathbb{R}^D} = 1$), they share the same physical dimension and can be interpreted in the same manner. Second, this bias is equivalent to assuming that the most significant shape variations are responsible for the largest output variations, which is a reasonable prior. In experiments that are not reported here for the sake of brevity, we have noticed that a BFGS algorithm optimizing Problem (6.6) got trapped by weak local optima more frequently when the α_j's were normalized.

Definition 6.1 (Selection of active dimensions)**.** *Let a GP be indexed by $\alpha_1, \dots, \alpha_D \in [\boldsymbol{\alpha}^{\min}, \boldsymbol{\alpha}^{\max}] \subset \mathbb{R}^D$ and $\{\boldsymbol{\alpha}^{(1:t)}, y^{(1:t)})\}$ be the data to model. The length-scales $\boldsymbol{\theta}$ of the GP are set by maximizing the L^1 penalized concentrated log-likelihood of Equation (6.7). A dimension j is declared* active *if*

$$\frac{\theta_j}{range(\alpha_j)} \leq 10 \times \min_{i=1,\dots,D} \frac{\theta_i}{range(\alpha_i)}.$$

The δ such active dimensions are denoted $\boldsymbol{\alpha}^a = (\alpha_{a_1}, \dots, \alpha_{a_\delta}) \in \mathbb{R}^\delta$.

[2] Typically of the order of range(α_j).

Since the α_j's have different (decreasing) ranges, the length-scales have to be normalized by the range of $\boldsymbol{\alpha}_j^{(1:t)}$ to be meaningful during this θ_j comparison. Our implementation extends the likelihood maximization of the `kergp` package (Deville et al., 2015) to include the penalization term. After a dimensional analysis of pl_λ, we have chosen to take $\lambda = \frac{t}{D}$ to balance both terms. Other techniques such as cross-validation or the use of different λ's for obtaining a pre-defined number of active components can also be considered.

On the NACA 22 benchmark with few observations of $f(\cdot)$ (DoE of $n = 15$ observations here), Figure 6.28 gives the only few active components that are selected by the penalized maximum likelihood procedure. The three first principal axes, $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$ are retained when considering the drag (top). Indeed, these are the eigenshapes that globally impact the shape the most and change its drag. When the output y is the lift (bottom), only the second principal axis is selected. This eigenshape modifies the camber of the shape, which is known to highly impact the lift. The other eigenvectors are detected to be less critical for y's variations. When n grows, more eigenshapes get selected because they also slightly impact the output. For instance when $n = 50$, some eigenshapes that contain bumps (the 4th, the 5th, the 8th, etc.) are selected for modeling the lift. They also contribute to changing the camber of the airfoil, hence its lift.

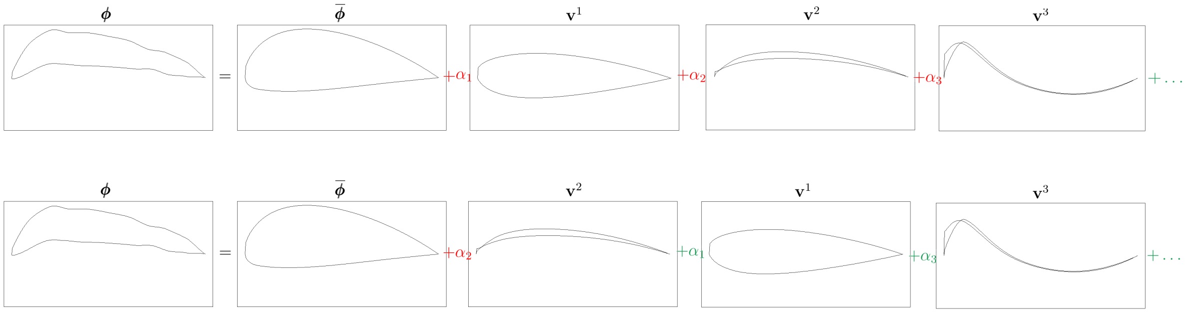


Figure 6.28: Variable selection on the NACA 22 benchmark by penalized maximum likelihood. For the drag (top), the three first eigenshapes that act on the shape, hence on its drag, are selected (red coefficients). For the lift, only the second eigencomponent ($\mathbf{v}^2$) is selected (bottom). Indeed $\mathbf{v}^2$ modifies the camber of the airfoil, hence it plays a major role on the lift. The other eigenbasis vectors (green coefficients) are estimated to be less influential on y.

6.3.2.2 Additive GP between active and inactive eigenshapes

Completely omitting the non-active dimensions, $\boldsymbol{\alpha}^{\overline{a}} \in \mathbb{R}^{D-\delta}$, and building the surrogate model $Y(\cdot)$ in the sole $\boldsymbol{\alpha}^a$ space may amount to erasing some geometric patterns of the shapes which contribute to small variations of y. For this reason, an additive GP (Durrande et al., 2012; Duvenaud et al., 2011) with zonal anisotropy (Allard et al., 2016) between the active eigenshapes and the residual ones is considered:

$$Y(\boldsymbol{\alpha}) = \beta + Y^a(\boldsymbol{\alpha}^a) + Y^{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}}). \tag{6.8}$$

$Y^a(\boldsymbol{\alpha}^a)$ is the anisotropic main-effect GP which works in the reduced space of active variables. It requires the estimation of $\delta + 1$ hyperparameters (the length-scales θ_j and a GP variance σ_a^2) and aims at capturing most of y's variation, related to $\boldsymbol{\alpha}^a$'s effect. $Y^{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}})$ is a GP over the large space of inactive components. It is a GP which just takes residual effects into account. To keep $Y^{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}})$ tractable, it is considered isotropic, i.e., it only has 2 hyperparameters, a unique length-scale $\theta_{\overline{a}}$ and a variance $\sigma_{\overline{a}}^2$. In the end, even though $Y(\boldsymbol{\alpha})$ operates with $\boldsymbol{\alpha}$'s $\in \mathbb{R}^D$ and there are fewer observations than dimensions[3], $n \ll D$, it remains tractable since only a total of $\delta + 3 \ll n$ hyperparameters have to be learned, which guarantees the identifiability, i.e. the unicity of the hyperparameters solution even when the number of observations is small. Although the α_j's have different ranges, they are homogeneous in that they all multiply normalized eigenshapes. Thus, the distances inside the shape manifold, $\mathcal{A}$, should be relevant and an isotropic model is a possible assumption, which again, tends to emphasize eigenshapes that appear the most within the designs. This additive model can be interpreted as a GP in the $\boldsymbol{\alpha}^a$ space, with an inhomogeneous noise fitted by the $Y^{\overline{a}}(\cdot)$ GP (Durrande, 2011). It aims at modeling a function that varies primarily along the active dimensions, and fluctuates only marginally along the inactive ones, as illustrated in Figure 6.29.

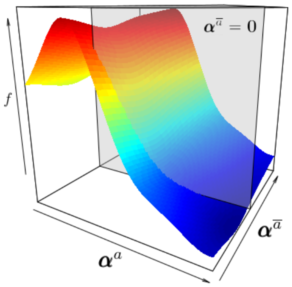


Figure 6.29: Example of a function that primarily varies along the $\boldsymbol{\alpha}^a$ direction, and secondarily along $\boldsymbol{\alpha}^{\overline{a}}$. If $\boldsymbol{\alpha}^{\overline{a}}$ is omitted, one implicitly considers the restriction of $f(\cdot)$ to the gray plane where $\boldsymbol{\alpha}^{\overline{a}} = \mathbf{0}$.

Denoting $k_a(\cdot,\cdot)$ and $k_{\overline{a}}(\cdot,\cdot)$ the kernels of the GPs, the hyperparameters $\vartheta_a = (\theta_{a_1}, \dots, \theta_{a_\delta}, \sigma_a^2)$ and $\vartheta_{\overline{a}} = (\theta_{\overline{a}}, \sigma_{\overline{a}}^2)$ are estimated by maximizing the log-likelihood of (6.8) given the observed data $y^{(1:t)}$,

$$\widehat{l}_Y(\boldsymbol{\alpha}^{(1:t)}, y^{(1:t)}; \vartheta_a, \vartheta_{\overline{a}}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log(|\mathbf{K}|) - \frac{1}{2}(y^{(1:t)} - \mathbf{1}\widehat{\beta})^\top \mathbf{K}^{-1}(y^{(1:t)} - \mathbf{1}\widehat{\beta}),$$

using the `kergp` package (Deville et al., 2015). $\mathbf{K} = \mathbf{K}_a + \mathbf{K}_{\overline{a}}$, with $K_{aij} = \sigma_a^2 k_a(\boldsymbol{\alpha}^{a(i)}, \boldsymbol{\alpha}^{a(j)})$, and $K_{\overline{a}ij} = \sigma_{\overline{a}}^2 k_{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}(i)}, \boldsymbol{\alpha}^{\overline{a}(j)})$, and $\widehat{\beta}$ is the Generalized Least Squares estimate, see Section 2.1.3. The correlation between $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ being

[3] Even if pruning the α_j components for $j > d'$ (see comments at the end of Section 6.2.3), $n < d'$ may hold.

$k(\boldsymbol{\alpha},\boldsymbol{\alpha}') = \sigma_a^2 k_a(\boldsymbol{\alpha}^a,\boldsymbol{\alpha}^{a\prime}) + \sigma_{\overline{a}}^2 k_{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}},\boldsymbol{\alpha}^{\overline{a}\prime})$, the kriging predictor and variance (Equations 2.2 and 2.3) of this additive GP are

$$\begin{aligned} \widehat{y}(\boldsymbol{\alpha}) &= \mathbf{1}_n\widehat{\beta} + k(\boldsymbol{\alpha},\boldsymbol{\alpha}^{(1:t)})^\top \mathbf{K}^{-1}(y^{(1:t)} - \mathbf{1}_n\widehat{\beta}) \\ s^2(\boldsymbol{\alpha}) &= \sigma_a^2 + \sigma_{\overline{a}}^2 - k(\boldsymbol{\alpha},\boldsymbol{\alpha}^{(1:t)})^\top \mathbf{K}^{-1} k(\boldsymbol{\alpha},\boldsymbol{\alpha}^{(1:t)}) + \frac{(1 - \mathbf{1}_t^\top \mathbf{K}^{-1} k(\boldsymbol{\alpha},\boldsymbol{\alpha}^{(1:t)}))^2}{\mathbf{1}_t^\top \mathbf{K}^{-1}\mathbf{1}_t} \end{aligned} \tag{6.9}$$

6.3.3 Experiments: metamodeling in the eigenshape basis

We now study the performance of the variable selection and of the additive GP described in the previous section. The different versions of GPs that are compared are the following:

- `GP(`X`)` is the GP in the original space of parameters X;
- `GP(`$\boldsymbol{\alpha}_{\_\_}$`)` indicates the GP is built in the space of __ (to be specified) principal components;
- `GP(`$\boldsymbol{\alpha}^a$`)` means the GP works with the active $\boldsymbol{\alpha}$'s only;
- `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)` refers to the additive GP (Section 6.3.2.2).

We equip the example designs 6.2, 6.4, 6.5 and 6.7 (Section 6.2.3) with objective functions $f(\mathbf{x})$ that are to be modeled by the fitted GPs. For each function, the predictive capability of different models is compared on a distinct test set using the R2 coefficient of determination. Later, in Section 6.4.3.2, the objective functions will be optimized.

- Example 6.2: $f_2(\mathbf{x}) = r - \pi r^2 - \|(x,y)^\top - (3,2)^\top\|_2$, where x, y and r correspond to the position of the center and the radius of the over-parameterized circle (and accessible through $\mathbf{x}$), respectively.
- Example 6.4: $f_4(\mathbf{x}) = \|\Omega_{\mathbf{t}} - \Omega_{\tilde{\mathbf{x}}}\|_2^2$ where $\tilde{\mathbf{x}} := \mathbf{x} - (x_1 + 2.5, x_2 + 2.5, 0, \dots, 0)^\top$ corresponds to the centered design, and $\Omega_{\mathbf{t}}$, $\Omega_{\tilde{\mathbf{x}}}$ are the nodal coordinates of the shapes, see Figure 6.20. The goal is to retrieve a target shape $\mathbf{t} = (t_1, \dots, t_{40})^\top$ whose lower left point (A) is set at $t_1 = t_2 = 2.5$ with the flexible rectangle defined by $\mathbf{x}$. The A point of any shape $\mathbf{x}$ is first moved towards (2.5, 2.5) too, and f_4 measures the discrepancy. Here, the target $\mathbf{t}$ is the rectangular heart shown in Figure 6.30.
- Example 6.5: $f_5(r) = 2\pi\int_{y_A}^{y_B} r(y)\sqrt{1 + r'(y)^2}dy$: inspired by the catenoid problem (Colding and Minicozzi, 2006), we aim at finding a regular curve joining two points A $= (0, y_A)$ and B $= (1, y_B)$, with the smallest axisymmetric surface. The curve $r(y)$ is the straight line between A and B, modified by $\mathbf{r} = (r_1, \dots, r_{29})^\top$, see Figure 6.22.

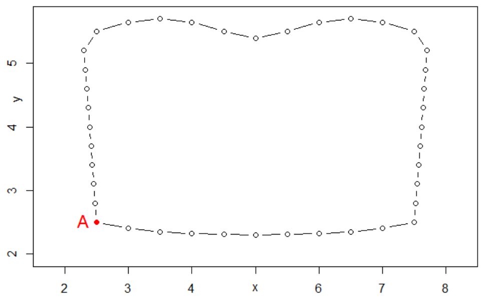


Figure 6.30: Rectangular heart target shape of Example 6.4.

- Example 6.7: the objective functions are the lift coefficient and the drag coefficient of the airfoil, f_{7L}, f_{7D}. The latter are computed using a commercial Computational Fluid Dynamics (CFD) computer code. Remark that in this section, we do not use the MetaNACA of Chapter 3, but rather true CFD computations (in particular some evaluations that where used for building the MetaNACA).

Over-parameterized circle (Example 6.2)

For the over-parameterized circle, the objective function is $f_2(\mathbf{x}) = r - \pi r^2 - \|(x, y)^\top - (3, 2)^\top\|_2$, where x, y and r correspond to the position of the center and the radius of the circle (accessible through $\mathbf{x}$), respectively. f_2 explicitly depends on the parameters that truly define the circle. Three models are compared

- A model using the CAD parameters $\mathbf{x} \in \mathbb{R}^{39}$;
- A model using the 3 first eigencomponents, $(\alpha_1, \alpha_2, \alpha_3)$;
- A model built over the *true* circle parameters (x, y, r).

Table 6.10 gives the average R2 over 10 runs with different space-filling DoEs of size $n = 20, 50, 100, 200$. Since $d = 39 > 20$, no GP was fitted in the CAD parameter space when $n = 20$.

n	GP(X)	GP($\boldsymbol{\alpha}_{1:3}$)	GP(True)
20	-	0.99741	0.99701
50	0.78193	0.99954	0.99951
100	0.86254	0.99984	0.99985
200	0.93383	0.99992	0.99997

Table 6.10: Average R2 over 10 runs for the prediction of f_2. GP(X) is the GP in the 39-dimensional CAD parameter space, GP($\boldsymbol{\alpha}_{1:3}$) corresponds to a GP fitted to the 3 first principal components $\alpha_1, \alpha_2, \alpha_3$, and GP(True) to the GP with the space of minimal circle coordinates.

f_2 is easily learned by the surrogate model as shown by large R2 values. Obviously, the quality of prediction increases with n and the eigenshape GP ($\texttt{GP}(\boldsymbol{\alpha}_{1:3})$) built in a 3-dimensional space outperforms the GP in the CAD parameters space ($\texttt{GP}(X)$, $d = 39$). Yet, the $\texttt{GP}(\boldsymbol{\alpha}_{1:3})$ performs as well (and even better for small n's) as $\texttt{GP(True)}$.

Heart target (Example 6.4)

We turn to the metamodeling of f_4. It is a 40-dimensional function, $f_4(\mathbf{x}) = \|\Omega_{\mathbf{t}} - \Omega_{\tilde{\mathbf{x}}}\|_2^2$ that explicitly depends on the CAD parameters. Unlike the previous test problem, the shapes do not have superfluous parameters since all x_j's are necessary to retrieve $\mathbf{t}$.

7 different models detailed through Sections 6.3.1 and 6.3.2 are investigated. $\texttt{GP}(X)$, the standard GP carried out in the space of CAD parameters. $\texttt{GP}(\boldsymbol{\alpha}_{1:40})$, the metamodel built in the space of 40 first principal components. Indeed, Table 6.6 informed us that any shape is retrieved via its 40 first eigenshape coefficients. To build surrogates in reduced dimension, considering the cumulative eigenvalue sum in Table 6.6, $\texttt{GP}(\boldsymbol{\alpha}_{1:2})$, $\texttt{GP}(\boldsymbol{\alpha}_{1:4})$ and $\texttt{GP}(\boldsymbol{\alpha}_{1:16})$ are models that consider the 2, 4 and 16 first principal components only. Finally, $\texttt{GP}(\boldsymbol{\alpha}^a)$ and $\texttt{AddGP}(\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}})$ are also compared.

Table 6.11 reports the average R2 indicator over 10 runs starting with space-filling DoEs of size $n = 20, 50, 100, 200$. Figure 6.31 shows a boxplot of the results (for the sake of clarity, only runs with R2 $\geq$ 0.8 are shown). The input dimension for $\texttt{GP}(X)$ and for $\texttt{GP}(\boldsymbol{\alpha}_{1:40})$ is too large for coping with $n = 20$ observations. $\texttt{GP}(\boldsymbol{\alpha}_{1:40})$ is given beside $\texttt{GP}(X)$ because both GPs have the same input space dimension.

n	$\texttt{GP}(X)$	$\texttt{GP}(\boldsymbol{\alpha}_{1:40})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:2})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:4})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:16})$	$\texttt{GP}(\boldsymbol{\alpha}^a)$	$\texttt{AddGP}(\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}})$
20	-	-	-0.063	0.979	0.844	0.935	0.967
50	0.455	0.542	-0.009	0.984	0.968	0.983	0.991
100	0.662	0.868	0	0.986	0.986	0.986	0.997
200	0.873	0.988	0	0.987	0.991	0.987	0.999

Table 6.11: Average R2 over 10 runs when metamodeling f_4.

The benefits of the additive GP appear to be threefold. First, it ensures sparsity by selecting a small number of eigenshapes for the anisotropic part of the kernel. A high-dimensional input space hinders the predictive capabilities when n is small, as confirmed by the weak performance of $\texttt{GP}(X)$, $\texttt{GP}(\boldsymbol{\alpha}_{1:40})$ and even $\texttt{GP}(\boldsymbol{\alpha}_{1:16})$ for $n = 20$. When n increases, higher-dimensional models become more accurate. For $n = 100$ and $n = 200$, the model with 16 principal components outperforms the one with 4 principal components, even though the latter was more precise with $n = 20$ or $n = 50$ observations. In the case $n = 200$, even $\texttt{GP}(\boldsymbol{\alpha}_{1:40})$ outperforms the 4 dimensional one ($\texttt{GP}(\boldsymbol{\alpha}_{1:4})$). This is due to the fact that more principal components mean a more realistic shape, hence less "input space errors". When few observations are available, these models suffer from the curse of dimensionality, but become accurate as soon as their design space gets infilled enough. With more observations, $\texttt{GP}(\boldsymbol{\alpha}_{1:40})$ may become the best model.

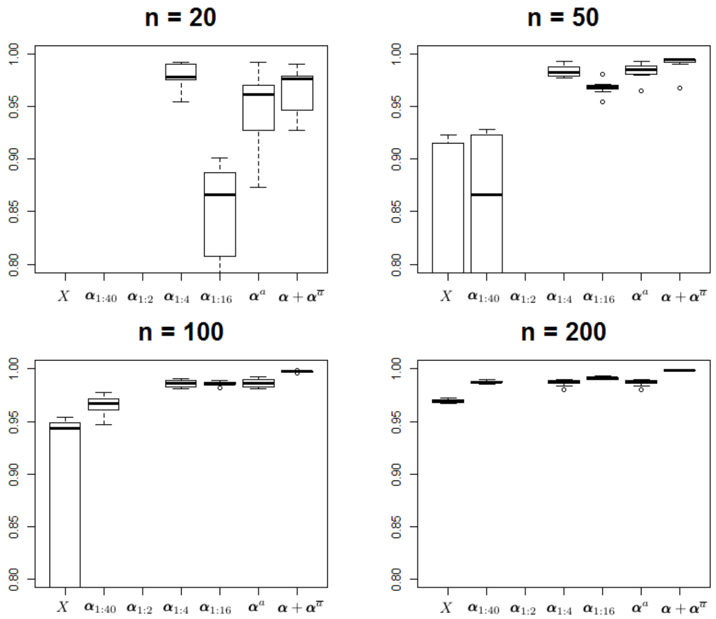


Figure 6.31: Boxplots of R2 coefficient for the different models, rectangle test case (Example 6.4).

Besides the dimension reduction, the selection of eigenshapes that truly influence the output is also critical. According to Table 6.6, a tempting decision to reduce the dimension would be to retain the two first principal components, i.e. `GP(`$\boldsymbol{\alpha}_{1:2}$`)`. But since the 2 first eigenshapes act on the shape's position (see Figure 6.21) to which f_4 is insensitive, this is a weak option, as pointed out by the R2 scores which are close to 0 for this model. Here, the selected variables are usually the 3rd and the 4th eigenshape which act on the size of the rectangle, hence are of first order importance for f_4. In about 30% of the runs, they are accompanied by the first and the second one, and more rarely by other eigenshapes.

Third, the `AddGP(`$\boldsymbol{\alpha}^a+\boldsymbol{\alpha}^{\overline{a}}$`)` outperforms `GP(`$\boldsymbol{\alpha}^a$`)`. Indeed, the less important eigenshapes (from a geometric point of view) $\mathbf{v}^5, \dots, \mathbf{v}^{40}$ locally modify the rectangle, and allow the final small improvements in f_4. This highlights the benefits of taking the remaining eigenshapes which act as local shape refinements into account.

Last, even though their input spaces have the same dimension, `GP(`$\boldsymbol{\alpha}_{1:40}$`)` consistently outperforms `GP(`X`)`. This confirms our comments about the NACA manifold of Figure 6.24: the eigenshapes are a better representation than the CAD parameters for statistical prediction.

Catenoid shape (Example 6.5)

In relation with the catenoid, we introduce the objective function $f_5(r) = 2\pi \int_{y_A}^{y_B} r(y)\sqrt{1+r'(y)^2}dy$. f_5 is an integral related to the surface of the axisymmetric surface given by the rotation of a curve $r(y)$. In our example, $r(y)$ is the line between two points A and B modified by regularly spaced deviations $\mathbf{r} = (r_1, \dots, r_{29})^\top$. Only $\mathbf{r}$'s generated by a GP that lead to a curve inside a prescribed envelope (see Figure 6.22) are kept in the same spirit as Li et al. (2019) where a smoothing operator is applied to consider realistic airfoils. With this, it is expected that less than 29 dimensions suffice to accurately describe all designs. This is confirmed by the eigenvalues in Table 6.7 and the true dimensionality detected to be 7.

In this experiment, we compare the predictive capabilities of six models. The first one is the classical $\texttt{GP(}X\texttt{)}$. The objective function explicitly depends on $\mathbf{r}$ but its high-dimensionality may be a drawback for metamodeling. Even though less dimensions are necessary and many eigenshapes correspond to noise, a GP fitted to all $d = 29$ eigenshapes, $\texttt{GP(}\boldsymbol{\alpha}_{1:29}\texttt{)}$, is considered. Along with it, $\texttt{GP(}\boldsymbol{\alpha}_{1:4}\texttt{)}$ and $\texttt{GP(}\boldsymbol{\alpha}_{1:7}\texttt{)}$ are considered. The former is an unsupervised dimension reduction, considering the λ_j's, while the latter is the full dimensional eigenshape GP, since the eigenshapes 8 to 29 are non-informative. Finally, the GPs with variable selection $\texttt{GP(}\boldsymbol{\alpha}^a\texttt{)}$ and $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)}$, are also compared.

Table 6.12 reports the average R2 indicator over 10 runs starting with space-filling DoEs of size $n = 20, 50, 100, 200$. Figure 6.32 shows a boxplot of the results (for the sake of clarity, only runs with R2 $\geq$ 0.95 are shown). The input dimension for $\texttt{GP(}X\texttt{)}$ and for $\texttt{GP(}\boldsymbol{\alpha}_{1:29}\texttt{)}$ is too large for coping with $n = 20$ observations. $\texttt{GP(}\boldsymbol{\alpha}_{1:29}\texttt{)}$ is given beside $\texttt{GP(}X\texttt{)}$ because these GPs have the same input space dimension.

n	$\texttt{GP(}X\texttt{)}$	$\texttt{GP(}\boldsymbol{\alpha}_{1:29}\texttt{)}$	$\texttt{GP(}\boldsymbol{\alpha}_{1:4}\texttt{)}$	$\texttt{GP(}\boldsymbol{\alpha}_{1:7}\texttt{)}$	$\texttt{GP(}\boldsymbol{\alpha}^a\texttt{)}$	$\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)}$
20	-	-	0.966	0.958	0.914	0.992
50	0.976	0.925	0.954	0.987	0.938	0.997
100	0.992	0.968	0.958	0.997	0.957	0.999
200	0.997	0.981	0.952	0.998	0.951	0.999

Table 6.12: Average R2 over 10 runs for the metamodeling of f_5.

These results indicate a better performance of $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)}$ which benefits from the prioritization of the most influential eigenshapes in the additive model and, at the same time, accounts for all the 7 eigenshapes. Modeling in the space of the full $\boldsymbol{\alpha}$'s ($\texttt{GP(}\boldsymbol{\alpha}_{1:7}\texttt{)}$) performs fairly well too because the low true dimensionality (7). Despite its lower dimensionality, $\texttt{GP(}\boldsymbol{\alpha}_{1:4}\texttt{)}$ does not work well. This is because the refinements induced by $\mathbf{v}^5$, $\mathbf{v}^6$ and $\mathbf{v}^7$ are disregarded while acting on f_5. This explanation also stands for the moderate performance of $\texttt{GP(}\boldsymbol{\alpha}^a\texttt{)}$ in which mainly the 4 first principal components are selected. Including the remaining components in a coarse GP as is done inside $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)}$ increases the performance.

Even though there are $d = 29$ CAD parameters, $\texttt{GP(}X\texttt{)}$ exhibits correct performances: since only smooth curves are considered, they are favorable to GP modeling and the curse of dimensionality is damped. In this example, considering all 29 eigenshapes ($\texttt{GP(}\boldsymbol{\alpha}_{1:29}\texttt{)}$), even though it was assumed that solely 7 were necessary, leads to the worst results, since the non-informative eigenshapes augment the dimension without bringing additional information.

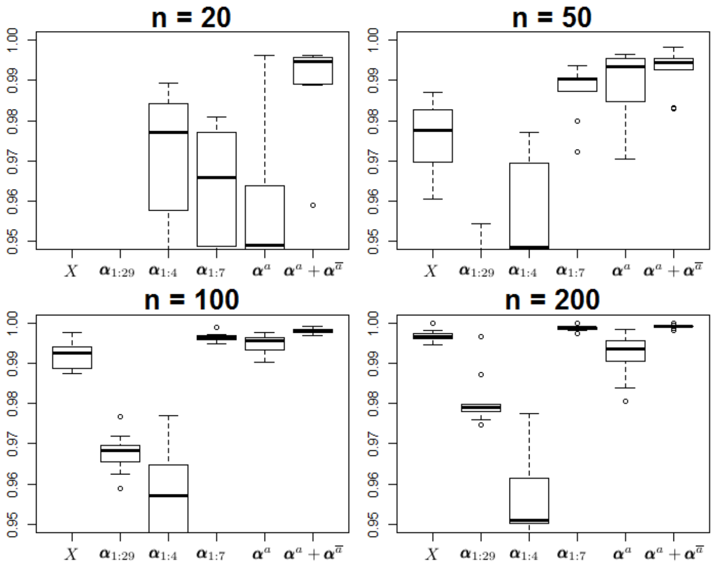


Figure 6.32: Boxplots of R2 coefficient for the different models, catenoid test case (Example 6.5).

NACA 22 airfoil (Example 6.7)

The last example brings us closer to real world engineering problems. The objective functions associated to the NACA airfoil with 22 parameters (Example 6.7), f_{7L} and f_{7D} are the lift and the drag coefficient of this airfoil. f_{7L}, f_{7D} depend implicitly and nonlinearly on $\mathbf{x}$ through $\Omega_{\mathbf{x}}$.

Table 6.9 shows that only the 20 first eigenvectors are informative. Seven metamodeling strategies are compared: GP(X); GP($\boldsymbol{\alpha}_{1:20}$), the surrogate in the space of all 20 meaningful eigenshapes; GP($\boldsymbol{\alpha}_{1:2}$), GP($\boldsymbol{\alpha}_{1:3}$), GP($\boldsymbol{\alpha}_{1:6}$) where fewer eigenshapes are considered; GP($\boldsymbol{\alpha}^a$); and AddGP($\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\bar{a}}$). GP($\boldsymbol{\alpha}_{1:20}$) is given beside GP($X$) because these GPs have almost the same input space dimension.

Table 6.13 reports the average R2 indicator over 10 runs starting with space-filling DoEs of $n = 20, 50, 100, 200$ observations. Figure 6.33 shows a boxplot of the results (for the sake of clarity, only runs with R2 ≥ 0.8 for f_{7L} and ≥ 0.6 for f_{7D} are shown). The input dimension for the GP(X) ($d = 22$) and for GP($\boldsymbol{\alpha}_{1:20}$) is too large for coping with $n = 20$ observations.

In this example too, AddGP($\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\bar{a}}$) exhibits the best predictive capabilities. Even though they are coarsely taken into account, the non active eigenshapes which mostly represent bumps, are included in the surrogate model. For the lift, GP($\boldsymbol{\alpha}^a$) performs quite well too since the f_{7L} relevant dimensions have been selected. The variable selection method provides contrasted results between f_{7L} and f_{7D}. For the lift, the first eigenshape is not always selected. The second and the third one, as well as some higher order eigenshapes get selected, which confirms the effect of the bumps on the lift (see Figures 6.25 and 6.26). For the drag (f_{7D}) however, only the 2 or 3 first eigenshapes are usually selected.

f_{7L}

n	$\texttt{GP}(X)$	$\texttt{GP}(\boldsymbol{\alpha}_{1:20})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:2})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:3})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:6})$	$\texttt{GP}(\boldsymbol{\alpha}^a)$	$\texttt{AddGP}(\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\bar{a}})$
20	-	-	0.857	0.907	0.930	0.935	0.957
50	0.956	0.973	0.714	0.935	0.950	0.970	0.984
100	0.975	0.989	0.708	0.938	0.962	0.981	0.992
200	0.987	0.995	0.515	0.954	0.968	0.993	0.996

f_{7D}

n	$\texttt{GP}(X)$	$\texttt{GP}(\boldsymbol{\alpha}_{1:20})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:2})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:3})$	$\texttt{GP}(\boldsymbol{\alpha}_{1:6})$	$\texttt{GP}(\boldsymbol{\alpha}^a)$	$\texttt{AddGP}(\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\bar{a}})$
20	-	-	0.443	0.806	0.720	0.800	0.796
50	0.771	0.847	0.259	0.866	0.882	0.878	0.896
100	0.861	0.921	0.192	0.915	0.928	0.925	0.945
200	0.915	0.958	-0.008	0.920	0.950	0.946	0.969

Table 6.13: Average R2 over 10 runs for the metamodeling of f_{7L} (top) and f_{7D} (bottom).

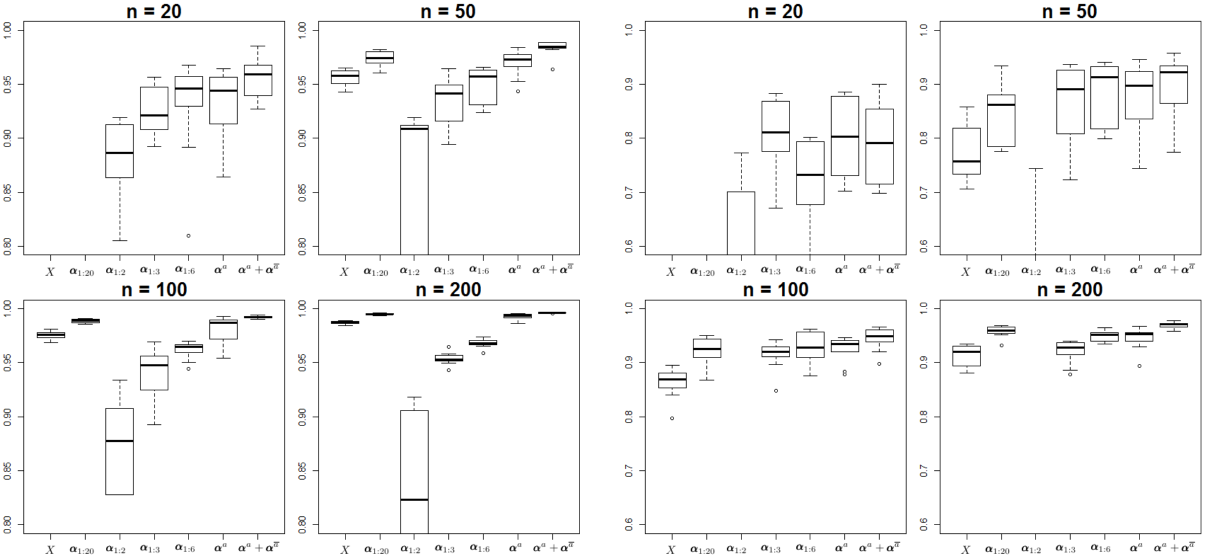


Figure 6.33: Boxplots of R2 coefficient for the different models, NACA 22 airfoil example. Left: Lift, f_{7L}. Right: Drag, f_{7D}.

We have also noticed that the number of selected components tends to grow with n. This is a desirable property since with larger samples, an accurate surrogate can be built in a higher dimensional space. As already remarked in the previous examples (e.g. Table 6.11), it is seen here in Figure 6.33 that models with more eigenshapes become more accurate when the number of observations grows. For f_{7D} (bottom table) for example, when n is small, $\texttt{GP}(\boldsymbol{\alpha}_{1:3})$ is better than $\texttt{GP}(\boldsymbol{\alpha}_{1:6})$ and $\texttt{GP}(\boldsymbol{\alpha}_{1:20})$, but this changes as n grows, $\texttt{GP}(\boldsymbol{\alpha}_{1:6})$ and $\texttt{GP}(\boldsymbol{\alpha}_{1:20})$ becoming in turn the best eigenshape truncation-based model. For f_{7L} (top table), in spite of the dimension reduction, very poor results are achieved when retaining only 2 or 3 components, even with small n's. When considering only the two first eigenshapes ($\texttt{GP}(\boldsymbol{\alpha}_{1:2})$), the R2 is weak as the third eigenshape significantly modifies the camber. For this GP, the performance decreases with n because of situations like the one shown in Figure 6.27 where shapes that falsely look similar when considering $\boldsymbol{\alpha}_{1:2}$ only actually differ in lift. Such situations are more likely to occur during the training of the GP as n grows, which degrades performance. The example of f_{7L} is informative in the sense that $\texttt{GP(}\boldsymbol{\alpha}_{1:20}\texttt{)}$ always outperforms $\texttt{GP(}\boldsymbol{\alpha}_{1:6}\texttt{)}$ which outperforms $\texttt{GP(}\boldsymbol{\alpha}_{1:3}\texttt{)}$, for any n (including very little n's), despite the higher dimension. By ignoring second order eigenshapes, $\texttt{GP(}\boldsymbol{\alpha}_{1:3}\texttt{)}$ and $\texttt{GP(}\boldsymbol{\alpha}_{1:6}\texttt{)}$ provide less reconstruction details. These details are nonetheless important since they change the camber of the airfoil and this is why $\texttt{GP(}\boldsymbol{\alpha}_{1:20}\texttt{)}$, a more precise reconstruction, performs better. Indeed, the remaining $\boldsymbol{\alpha}$'s mainly reconstruct the bumps of this airfoil as can be seen in Figure 6.26, which does influence the lift.

This is also the reason why $\texttt{GP(}X\texttt{)}$ is better at predicting lift than $\texttt{GP(}\boldsymbol{\alpha}_{1:3}\texttt{)}$ and $\texttt{GP(}\boldsymbol{\alpha}_{1:6}\texttt{)}$, which could seem counter-intuitive at first glance since the dimension is reduced.

Last, let us point out than even though the dimension is almost the same, $\texttt{GP(}\boldsymbol{\alpha}_{1:20}\texttt{)}$ consistently outperforms $\texttt{GP(}X\texttt{)}$ for both the lift and the drag: it confirms that the eigenshape basis $\mathcal{V}$ is more relevant than the CAD parameters basis for GP surrogate modeling.

GP in reduced dimension: summary of results

These four examples have proven the worth of the additive GPs: they are the models that perform the best because of the selection and prioritization of active variables. Models in reduced dimension that exclusively rely on the active eigenshapes provide accurate predictions too, but are slightly outperformed as they disregard smaller effects. GPs built in the space of all (informative) eigenshapes always outperform the ones built in the space of CAD parameters, even when both models have the same dimension. Among the GPs over the reduced space of δ first principal axes, further removing dimensions generally produces better predictions when the number of data points n is small. As n increases, more eigenshapes lead to better metamodeling. Models where dimensions have been chosen only from a geometric criterion (the PCA) have a prediction quality that depends on the output: if the first modes do not impact y, as the 2 first eigenshapes of the rectangle problem, predictions are poor. Ignoring reconstruction details that affect the output as second-order eigenshapes in f_{7L} also degrades the performance, highlighting the importance of finding the active variables that affect the output.

6.4 Optimization in reduced dimension

We now turn to the problem of finding the shape that minimizes an expensive objective function $f(\cdot)$. To this aim, we employ the previous additive GP, which works in the space of eigencomponents $\boldsymbol{\alpha}$, in an Efficient Global Optimization procedure (Jones et al., 1998): at each iteration, a new shape is determined given the previous t observations $\{(\boldsymbol{\alpha}^{(1)}, y^{(1)}), \dots, (\boldsymbol{\alpha}^{(t)}, y^{(t)})\}$ by maximizing the Expected Improvement (EI, Mockus, 1975, see Equation 2.8) as calculated with the GP $Y(\boldsymbol{\alpha})$:

$$\boldsymbol{\alpha}^{(t+1)^*} = \arg\max_{\boldsymbol{\alpha}\in\mathbb{R}^D} \mathrm{EI}(\boldsymbol{\alpha}; Y(\boldsymbol{\alpha})) \tag{6.10}$$

6.4.1 Alternative Expected Improvement maximizations

Maximization in the entire $\boldsymbol{\alpha}$ space

The most straightforward way to maximize the EI is to consider its maximization in $\mathbb{R}^D$ as in Equation (6.10). However, this optimization is typically difficult as the EI is a multi-modal and high (D) dimensional function[4].

Maximization in the $\boldsymbol{\alpha}^a$ space

We can however take advantage of the dimension reduction beyond the construction of $Y(\cdot)$: $\boldsymbol{\alpha}^a \in \mathbb{R}^\delta$ are the variables that affect y the most and should be prioritized for the optimization of $f(\cdot)$. A second option is therefore to maximize the EI solely with respect to $Y^a(\boldsymbol{\alpha}^a)$ in dimension δ. This option is nonetheless incomplete as the full GP $Y(\cdot)$ requires the knowledge of $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^a, \boldsymbol{\alpha}^{\overline{a}}]$.

A first simple idea to augment $\boldsymbol{\alpha}^a$ is to set $\boldsymbol{\alpha}^{\overline{a}}$ equal to its mean, $\mathbf{0}$. The inactive part of the covariance matrix $\mathbf{K}_{\overline{a}}$ would be filled with the same scalar and the full covariance matrix $\mathbf{K} = \mathbf{K}_a + \mathbf{K}_{\overline{a}}$ would have a degraded conditioning. A second simple idea is to sample $\boldsymbol{\alpha}^{\overline{a}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\lambda}_{\overline{a}})$. However, $\boldsymbol{\alpha}^{\overline{a}}$ act as local refinements to the shape that contribute a little to y, and should also be optimized. In Li et al. (2019), the authors observed that despite the gain in accuracy of surrogate models in a reduced basis (directions of largest variation of the gradient of the lift and drag in their application), a restriction to too few directions led to poorer optimizations since small effects could not be accounted for.

Optimization in $\boldsymbol{\alpha}^a$ space complemented with a random embedding in $\boldsymbol{\alpha}^{\overline{a}}$

This leads to the third proposed EI maximization: a maximization of the EI with respect to $\boldsymbol{\alpha}^a$ and the use of a random embedding (Wang et al., 2013) to coarsely optimize the components $\boldsymbol{\alpha}^{\overline{a}}$: $\mathrm{EI}([\boldsymbol{\alpha}^a, \overline{\alpha}\overline{\mathbf{a}}])$ is maximized, where $\overline{\alpha} \in \mathbb{R}$ is the coordinate along a random line in the $\boldsymbol{\alpha}^{\overline{a}}$ space, $\overline{\mathbf{a}} = (\overline{\mathbf{a}}_1, \dots, \overline{\mathbf{a}}_{D-\delta})^\top$. Since $\boldsymbol{\alpha}^{\overline{a}}$ have been classified as inactive, it is not necessary to make a large effort for their optimization. This approach can be viewed as an extension of REMBO (Wang et al., 2013). In REMBO, a lower dimensional vector $\mathbf{y} \in \mathbb{R}^\delta$ is embedded in X through a linear random embedding, $\mathbf{y} \mapsto \mathbf{A}_R\mathbf{y}$, where $\mathbf{A}_R \in \mathbb{R}^{D\times\delta}$ is a random matrix. Instead of choosing a completely random and linear embedding with user-chosen (investigated in Binois et al., 2017) effective dimension δ, our embedding is nonlinear (effect of the mapping $\phi(\cdot)$), supervised and semi-random (choice of the active/inactive directions). The dimension is no longer arbitrarily chosen

[4] As explained at the end of Section 6.2, we can restrict all calculations to $\boldsymbol{\alpha}$'s d' first coordinates. Even though $d' \ll D$, it has approximately the same dimension as d, hence the optimization is still carried out in a high dimensional space.

since it is determined by the number of selected active components (Section 6.3.2.1), and the random part of the embedding is only associated to the inactive parts of $\boldsymbol{\alpha}$: denoting $\boldsymbol{\alpha}^a = (\alpha_{a_1}, \dots, \alpha_{a_\delta})^\top$ the selected components (that are not necessarily the δ first axes) and $\boldsymbol{\alpha}^a = (\alpha_{\overline{a}_1}, \dots, \alpha_{\overline{a}_{D-\delta}})^\top$ the inactive ones, our embedding matrix $\mathbf{A}_{emb} \in \mathbb{R}^{D\times(\delta+1)}$ transforms $[\boldsymbol{\alpha}^a, \overline{\alpha}]$ into the $\boldsymbol{\alpha}$ space to which the $\mathbf{x}$'s are nonlinearly mapped. The δ first columns of $\mathbf{A}_{emb}$, $\mathbf{A}_{emb}^{(i)}$, $i = 1, \dots, \delta$, correspond to $\boldsymbol{\alpha}^a$ and contain the δ first vectors of the canonical basis of $\mathbb{R}^D$, $\mathbf{e}_D^{(i)}$, i.e. $\mathbf{A}_{emb}^{(i)} = \delta_{a_i i}$, where δ_{ij} stands for the Kronecker symbol here, $\delta_{ij} = 1$ if $i = j$, 0 else. The $\delta + 1$-th column of $\mathbf{A}_{emb}$ contains $\overline{\mathbf{a}}$ in the rows which correspond to $\boldsymbol{\alpha}^{\overline{a}}$, $\mathbf{A}_{emb\ \overline{a}_i}^{(\delta+1)} = \overline{\mathbf{a}}_i$, $i = 1, \dots, D - \delta$. Rows corresponding to active $\boldsymbol{\alpha}$'s equal 0.

Assuming ϕ^{-1} exists, the proposed approach is the embedding of a lower dimensional design $[\boldsymbol{\alpha}^a, \overline{\alpha}]$ whose dimension $\delta + 1$ has carefully be chosen, in X, via the nonlinear and problem-related mapping $[\boldsymbol{\alpha}^a, \overline{\alpha}] \mapsto \phi^{-1}(\mathbf{V}\mathbf{A}_{emb}[\boldsymbol{\alpha}^a, \overline{\alpha}] + \overline{\boldsymbol{\phi}})$. The approach can alternatively be considered as an affine mapping of $[\boldsymbol{\alpha}^a, \overline{\alpha}]$ to the complete space spanned by the eigenshapes $\mathcal{V}$,

$$[\boldsymbol{\alpha}^a, \overline{\alpha}] \mapsto \mathbf{V}_{emb}[\boldsymbol{\alpha}^a, \overline{\alpha}] + \overline{\boldsymbol{\phi}} \quad \text{with} \quad \mathbf{V}_{emb} := \mathbf{V}\mathbf{A}_{emb} \tag{6.11}$$

The shapes generated by the map of Equation (6.11) are embedded in the space of all discretized shapes. The columns of $\mathbf{V}_{emb} \in \mathbb{R}^{D\times(\delta+1)}$ associated to active components are the corresponding eigenshapes, while its last column is sum of the remaining eigenshapes, weighted by random coefficients, namely $\mathbf{V}\overline{\mathbf{a}}$, hence a supervised and semi-random embedding. Another difference to Wang et al. (2013) is that only the EI maximization is carried out in the REMBO framework; the surrogate model is not built in terms of $[\boldsymbol{\alpha}^a, \overline{\alpha}]$ but rather with the full $\boldsymbol{\alpha}$'s via the additive GP (Section 6.3.2.2).

In this variant, the EI maximization is carried out in a much more tractable $\delta + 1$ -dimensional space and still has analytical gradients (see next section). From its optimum $\boldsymbol{\alpha}^* = [\boldsymbol{\alpha}^{a*}, \overline{\alpha}^*] \in \mathbb{R}^{\delta+1}$ arises a D-dimensional vector, $\boldsymbol{\alpha}^{(t+1)^*} = \mathbf{A}_{emb}[\boldsymbol{\alpha}^{a*}, \overline{\alpha}^*]$ to be evaluated by the true function (this is the pre-image problem discussed in Section 6.4.2).

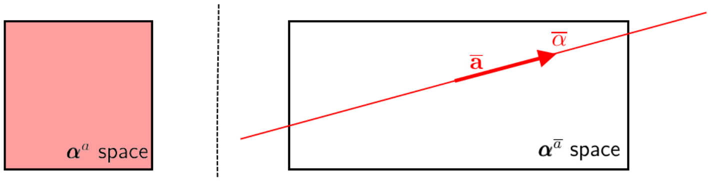

$\boldsymbol{\alpha}^a$ space
$\overline{\mathbf{a}}$
$\overline{\alpha}$
$\boldsymbol{\alpha}^{\overline{a}}$ space

Figure 6.34: EI maximization in $\boldsymbol{\alpha}^a$ complemented by the maximization along $\overline{\mathbf{a}}$, a random line in the $\boldsymbol{\alpha}^{\overline{a}}$ space.

EI gradient in $\boldsymbol{\alpha}$ space

The Expected Improvement is differentiable and its derivative is known in closed-form (Roustant et al., 2012). In the case of the additive GP (6.8), the mean and variance $\widehat{y}(\boldsymbol{\alpha})$

and $s^2(\boldsymbol{\alpha})$ are given by (6.9). Using the notations of Section 6.3.2.2 and exploiting the symmetry of $\mathbf{K}$, few calculations lead to

$$\begin{aligned}\nabla\widehat{y}(\boldsymbol{\alpha}) &= \nabla k(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{(1:t)})^\top \mathbf{K}^{-1}(y^{(1:t)} - \mathbf{1}_n\widehat{\beta})\\ \nabla s(\boldsymbol{\alpha}) &= -\frac{\nabla k(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{(1:t)})^\top \mathbf{K}^{-1} k(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{(1:t)})}{s(\boldsymbol{\alpha})}\end{aligned} \tag{6.12}$$

where $\nabla k(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{(1:t)}) = \sigma_a^2 \nabla k_a(\boldsymbol{\alpha}^a, \boldsymbol{\alpha}^{a^{(1:t)}}) + \sigma_{\overline{a}}^2 \nabla k_{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}}, \boldsymbol{\alpha}^{\overline{a}^{(1:t)}})$, which are plugged in (6.12) and in ∇EI's expression (2.9) to obtain $\nabla\text{EI}(\boldsymbol{\alpha}; Y(\boldsymbol{\alpha}))$. In the alternatives proposed before, given an $\boldsymbol{\alpha} \in \mathbb{R}^D$, the gradient of the EI can be computed efficiently, accelerating its maximization which is carried out by the genetic algorithm using derivatives `genoud` (Mebane Jr et al., 2011). In the random embedding of $\overline{\alpha}$ case, the EI of $[\boldsymbol{\alpha}^a, \overline{\alpha}] \in \mathbb{R}^{\delta+1}$ is given by $\text{EI}(\mathbf{A}_{emb}[\boldsymbol{\alpha}^a, \overline{\alpha}]; Y(\boldsymbol{\alpha}))$, and its gradient by $\mathbf{A}_{emb}^\top \nabla\text{EI}(\mathbf{A}_{emb}[\boldsymbol{\alpha}^a, \overline{\alpha}]; Y(\boldsymbol{\alpha}))$.

Setting bounds on $\boldsymbol{\alpha}$ for the EI maximization

As seen in the examples of Section 6.2.3, neither the manifold of $\boldsymbol{\alpha}$'s, nor its restriction to $\boldsymbol{\alpha}^a$ need to be hyper-rectangular domains, which is a common assumption made by most optimizers such as `genoud` (Mebane Jr et al., 2011), the algorithm used in our implementation. Two strategies were imagined to control the space in which the EI is maximized (6.10): the first one is to restrict the EI maximization to $\mathcal{A}$ by setting it to zero for $\boldsymbol{\alpha}$'s that are outside of the manifold. The benefit of this approach is that only realistic $\boldsymbol{\alpha}$'s are proposed. But it might suffer from an incomplete description of the entire manifold of $\boldsymbol{\alpha}$'s, $\mathcal{A}$, which is approximated by $\mathcal{A}_N$. Additionally, given $\mathcal{A}_N$, the statement "being inside/outside the manifold" has to be clarified. We rely on a nearest neighbor strategy in which the 95th quantile of the distances to the nearest neighbor within $\mathcal{A}_N$, $d_{0.95}$, is computed and used as a membership threshold: a new $\boldsymbol{\alpha}$ is considered to belong to $\mathcal{A}$ if and only if the distance to its nearest neighbor within $\mathcal{A}_N$ is smaller than $d_{0.95}$. In the light of these limitations, a second strategy, in which the EI is maximized in $\mathcal{A}_N$'s covering hyper-rectangle, is also investigated. The variant of EI maximization with embedding (random line in $\boldsymbol{\alpha}^{\overline{a}}$), introduces an $\overline{\alpha}$ coordinate which has to be bounded too. The $\overline{\alpha}_{\min}$ and $\overline{\alpha}_{\max}$ boundaries are computed as the smallest and largest projection of $\mathcal{A}_N$ on $\overline{\mathbf{a}}$. But depending on $\mathcal{A}_N$ and on $\overline{\mathbf{a}}$, this may lead to a too large domain since the embedded $\overline{\alpha}\overline{\mathbf{a}}$ might stay outside the $\boldsymbol{\alpha}^{\overline{a}}$ covering hyper-rectangle. In the spirit of Binois et al. (2015b), to avoid this phenomenon, the largest $\overline{\alpha}_{\min}$ and the smallest $\overline{\alpha}_{\max}$ such that $\overline{\alpha}\overline{\mathbf{a}}$ belongs to the covering hyper-rectangle $\forall \overline{\alpha} \in [\overline{\alpha}_{\min}, \overline{\alpha}_{\max}]$, are chosen.

EI maximization via the CAD parameters

A last option consists in carrying the maximization in the X space through the mapping $\phi(\cdot)$ by $\max_{\mathbf{x}\in X} \text{EI}(\mathbf{x}; Y(\underbrace{\mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})}_{\boldsymbol{\alpha}})) = \text{EI}(\mathbf{x}; Y(\boldsymbol{\alpha}(\mathbf{x})))$. This avoids both the aforementioned optimization domain handling and the pre-image search described in the following section. However, this optimization might be less efficient since it is a maximization in

$d > \delta$ dimensions, and since $\nabla\phi(\mathbf{x})$ is unknown, the EI loses the closed-form expression of its gradient.

6.4.2 From the eigencomponents to the original parameters: the pre-image problem

The (often expensive) numerical simulator underlying the objective function can only take the original (e.g. CAD) parameters as inputs. When the EI maximization is carried out in the eigencomponents space, the $\boldsymbol{\alpha}$'s need to be translated into $\mathbf{x}$'s. To this aim, the *pre-image* problem consists in finding the CAD parameter vector $\mathbf{x}$ whose description in the shape representation space Φ equals $\mathbf{V}\boldsymbol{\alpha}^{(t+1)^*} + \overline{\boldsymbol{\phi}}$. Because there are more $\boldsymbol{\alpha}$'s than $\mathbf{x}$'s, $D \gg d$, a strict equality may not hold and the pre-image problem is relaxed into:

$$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}\in X} \|(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) - \mathbf{V}\boldsymbol{\alpha}^{(t+1)^*}\|^2_{\mathbb{R}^D}. \tag{6.13}$$

To complete an iteration, the pre-image problem (6.13) is solved and its solution $\mathbf{x}^{(t+1)}$, the parametric shape that resembles $\boldsymbol{\alpha}^{(t+1)^*}$ the most, is evaluated by the simulator, which returns $y^{(t+1)} = f(\mathbf{x}^{(t+1)})$. Solving the pre-image problem does not involve calls to the simulator so that it is relatively not costly. The surrogate model is then updated with $y^{(t+1)}$ and $\boldsymbol{\alpha}^{(t+1)} := \mathbf{V}^\top(\phi(\mathbf{x}^{(t+1)}) - \overline{\boldsymbol{\phi}})$, the $\mathbf{x}^{(t+1)}$ description in the $\mathcal{V}$ basis.

Depending on the $\boldsymbol{\alpha}^{(t+1)^*}$ yielded by the EI maximization (remember it may not stay on the manifold $\mathcal{A}$), $\boldsymbol{\phi}^{(t+1)^*} := \mathbf{V}\boldsymbol{\alpha}^{(t+1)^*} + \overline{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}^{(t+1)} := \mathbf{V}\boldsymbol{\alpha}^{(t+1)} + \overline{\boldsymbol{\phi}}$, the shape representation of the $\boldsymbol{\alpha}$ promoted by the EI and the shape representation of $\mathbf{x}^{(t+1)}$, respectively, may substantially differ. While it is mandatory to update the GP (6.8) with the pair $(\boldsymbol{\alpha}^{(t+1)}, y^{(t+1)})$, it may at first seem unclear what should be done with $\boldsymbol{\alpha}^{(t+1)^*}$. When $\boldsymbol{\alpha}^{(t+1)^*}$ does not belong to $\mathcal{A}$ and does not have a pre-image, it might seem straightforward to ignore it. However, if $\boldsymbol{\alpha}^{(t+1)^*}$ was yielded by the EI, it is very likely to be promoted in the following iterations, since its uncertainty, $s^2(\boldsymbol{\alpha}^{(t+1)^*})$, has not vanished. Therefore, if $\boldsymbol{\phi}^{(t+1)^*}$ and $\boldsymbol{\phi}^{(t+1)}$ are substantially different, the virtual pair $(\boldsymbol{\alpha}^{(t+1)^*}, y^{(t+1)})$ is included in the GP (6.8) too in a strategy called *replication*. We define replication in general terms.

Definition 6.2 (Replication). *In Bayesian optimization, when the GP is built over coordinates $\boldsymbol{\alpha}$ that are a mapping*[5] *of the original coordinates $\mathbf{x}$, $\boldsymbol{\alpha} = T(\mathbf{x})$, at the end of each iteration a pre-image problem such as (6.13) must be solved to translate the new acquisition criterion maximizer $\boldsymbol{\alpha}^{(t+1)^*}$ into the next point to evaluate $\mathbf{x}^{(t+1)}$ and the associated iterate $\boldsymbol{\alpha}^{(t+1)} = T(\mathbf{x}^{(t+1)})$. The* replication *strategy consists in updating the GP with both $\big(\boldsymbol{\alpha}^{(t+1)}, f(\mathbf{x}^{(t+1)})\big)$ and $\big(\boldsymbol{\alpha}^{(t+1)^*}, f(\mathbf{x}^{(t+1)})\big)$ provided $\boldsymbol{\alpha}^{(t+1)^*}$ and $\boldsymbol{\alpha}^{(t+1)}$ are sufficiently different.*

[5] In this article, the mapping $T(\cdot)$ is the composition of $\phi(\cdot)$ with the projection onto a subspace of $(\mathbf{v}^1, \dots, \mathbf{v}^D)$.

Here, the difference between $\boldsymbol{\alpha}^{(t+1)^*}$ and $\boldsymbol{\alpha}^{(t+1)}$ is calculated as the distance between the associated shapes $\boldsymbol{\phi}^{(t+1)^*}$ and $\boldsymbol{\phi}^{(t+1)}$. Since the database $\boldsymbol{\Phi}$ contains the shape representation of N distinct designs, $d_0 := \min_{\substack{i,j=1,\dots,N \\ i\neq j}} \|\boldsymbol{\Phi}_i - \boldsymbol{\Phi}_j\|_{\mathbb{R}^D}$, the minimal distance between two different designs in $\boldsymbol{\Phi}$ is used as a threshold beyond which $\boldsymbol{\phi}^{(t+1)}$ and $\boldsymbol{\phi}^{(t+1)^*}$ are considered to be different. The replication strategy is further motivated by the fact that since $\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}\in X} \|(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) - \mathbf{V}\boldsymbol{\alpha}^{(t+1)^*}\|^2_{\mathbb{R}^D} = \arg\min_{\mathbf{x}\in X} \|\underbrace{\mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})}_{\boldsymbol{\alpha}(\mathbf{x})} - \boldsymbol{\alpha}^{(t+1)^*}\|^2_{\mathbb{R}^D}$, where the last equality expresses just a change of basis since $\mathbf{V}$ is orthogonal, $\boldsymbol{\alpha}^{(t+1)}$ is an orthogonal projection[6] of $\boldsymbol{\alpha}^{(t+1)^*}$ on $\mathcal{A}$, see Figure 6.35.

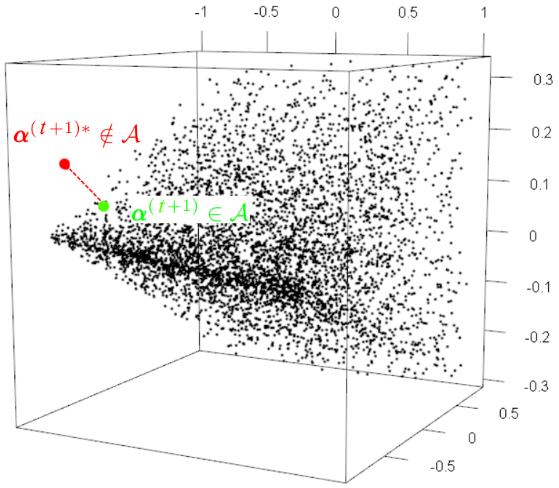


Figure 6.35: When $\boldsymbol{\alpha}^{(t+1)^*} \notin \mathcal{A}$, the solution of the pre-image problem (in the $\boldsymbol{\alpha}$ space), $\boldsymbol{\alpha}^{(t+1)}$, is its projection on $\mathcal{A}$.

This is somehow similar to Raghavan et al. (2013, 2014) where the authors project non realistic shapes on a smooth surface built via diffuse approximation or a local polynomial fitting, using the points of $\mathcal{A}_N$, to retrieve a realistic design. In our approach, unrealistic shape representations are directly projected onto $\mathcal{A}$ through the resolution of (6.13). Incorporating the non physical $(\boldsymbol{\alpha}^{(t+1)^*}, y^{(t+1)})$ in the surrogate model can be viewed as an extension of the surrogate model outside its domain (Shahriari et al., 2016) (outside the manifold $\mathcal{A}$ in our case) by constant prolongation.

6.4.3 Experiments

Many algorithms result from the combination of versions of the GP metamodel and the EI maximization. They are related to the space in which these operations are performed (the initial X or the eigencomponents $\mathcal{A}$ with the retained number of dimensions), the classical or additive GP, and the use of embedding or not. Before further explaining and

[6] Since we do not know the convexity of $\mathcal{A}$, the projection might not be unique.

testing them, we introduce a shorthand notation. The algorithms names are made of two parts separated by a dash, `GP version-EI version`. The GP part may either be an anisotropic GP with Matérn kernel, in which case it is noted `GP`, or an additive GP made of an anisotropic plus an isotropic kernel noted `AddGP`. The spaces on which they operate are specified in parentheses. For example, `GP(`$\boldsymbol{\alpha}_{1:3}$`)` is an anisotropic GP in the space spanned by $(\alpha_1, \alpha_2, \alpha_3)$, `AddGP(`$X_{1:3} + X_{4:40}$`)` is an additive GP where the kernel is the sum of an anisotropic kernel in (x_1, x_2, x_3) and an isotropic kernel in $(x_4, \dots, x_{40})$. The space over which the EI maximization is carried out is specified in the same way. Unspecified dimensions in the EI have their value set to the middle of their defining interval, e.g., `GP(`X`)-EI(`$X_{1:2}$`)` means that the EI maximization is done on the 2 first components of $\mathbf{x}$, the other ones being fixed to 0 if the interval is centered. The EI descriptor can also be a keyword characterizing the EI alternative employed (see Section 6.4.1). For example, `AddGP(`$\boldsymbol{\alpha}_{1:2} + \boldsymbol{\alpha}_{3:20}$`)-EI embed` means that the EI is maximized in a 3 dimensional space made of α_1, α_2 and the embedding $\overline{\alpha}$.

6.4.3.1 Optimization of a function with low effective dimension

A set of experiments is now carried out that aims at comparing the three optimization alternatives involving GPs which have been introduced in Section 6.4.1 when a subset of active variables has been identified: EI maximization in the space of active variables, in the space of active variables with an embedding in the inactive space, and in the entire space. In order to test the EI maximization separately from the space reduction method (the mapping, PCA and regularized likelihood), we start by assuming that the effective variables are known. Complete experiments will be given later.

We minimize a function depending on a small number of parameters, the following modified version of the Griewank function (Molga and Smutnicki, 2005),

$$f_{\text{MG}}(\mathbf{x}) = f_{\text{Griewank}}(\mathbf{x}) + f_{\text{Sph}}(\mathbf{x}),\ \mathbf{x} \in [-600, 600]^d \tag{6.14}$$

where $f_{\text{Griewank}}(\mathbf{x})$ is the classical Griewank function in dimension 2,

$$f_{\text{Griewank}}(\mathbf{x}) = \frac{1}{4000}\sum_{j=1}^{2} x_j^2 - \prod_{j=1}^{2}\cos(\frac{x_j}{\sqrt{j}}) + 1$$

defined in $[-600, 600]^2$ and whose optimum, located in $(0,0)^\top$, is 0. To create a high-dimensional function where only few variables act on the output, the f_{Sph} function is added to f_{Griewank}, where f_{Sph} is a sphere centered in $\mathbf{c}$, with smaller magnitude than f_{Griewank}, and which only depends on the variables $x_3, \dots, x_{10}$:

$$f_{\text{Sph}}(\mathbf{x}) = \frac{1}{400,000}\sum_{j=3}^{10}(x_j - c_{j-2})^2.$$

$f_{\text{Sph}}(\mathbf{x})$ is the squared Euclidean distance between $(x_3, \dots, x_{10})^\top$ and $\mathbf{c}$ which is set to $\mathbf{c} = (-140, -100, -60, -20, 20, 60, 100, 140)^\top$ in our experiments. Completely ignoring $(x_3, \dots, x_{10})^\top$ therefore does not lead to the optimum of f_{MG}. We define f_{MG} in

$[-600, 600]^d$, $d \geq 10$: the variables $x_{11}, \dots, x_d$ do not have any influence on f_{MG} but augment the dimension. In the following experiments, we take $d = 40$.

The additive GP described in Section 6.3.2.2 operates between the active space composed of x_1 and x_2, and the inactive space of $X_{3:d}$. With the additive GP, three ways to optimize the EI are investigated: AddGP($X_{1:2} + X_{3:40}$)-EI($X_{1:2}$) where the EI is optimized along the active space only and $x_3, \dots, x_d$ are set to the middle of their intervals ($\mathbf{0}$), AddGP($X_{1:2} + X_{3:40}$)-EI embed where the EI is optimized in the active space completed by the embedding in the inactive space, and AddGP($X_{1:2}+X_{3:40}$)-EI(X) where the EI is optimized in the entire X. These Bayesian optimization algorithms with additive GPs are compared to three classical optimizers: one based on the GP built in the entire space (GP(X)-EI(X)), another based on the building of the GP in the $X_{1:2} := (x_1, x_2)$ space (GP($X_{1:2}$)-EI($X_{1:2}$)), and one working in the $X_{1:10} := (x_1, \dots, x_{10})$ space (GP($X_{1:10}$)-EI($X_{1:10}$)).

We start the experiments with an initial DoE of $n = 20$ points, which is space-filling in X (or in $X_{1:2}$ or $X_{1:10}$ for the variants where the metamodel is built in these spaces). We then try to find the minimum of f_{MG}, $\mathbf{x}^* := (0, 0, \mathbf{c}, *)$ in the limit of $p = 80$ iterations[7]. For the instance where the metamodel is built in $X \subset \mathbb{R}^{40}$, we cannot start with an initial DoE of $n = 20$ points, and the experiments are initialized with $n = 50$ designs, only $p = 50$ iterations being allowed. The EI being maximized by the genetic algorithm genoud (Mebane Jr et al., 2011), we use the same population and number of generations in each variant for fair comparison.

The lowest objective function values obtained by the algorithms are reported in Table 6.14. They are averaged over 10 runs with different initial designs, and standard deviations are given in brackets. The left-hand side columns correspond to standard GPs carried out in different spaces, and the right-hand side columns correspond to runs using the additive GP of Section 6.3.2.2 together with different EI maximization strategies.

Metamodel	Standard GP GP($X_{1:2}$)-	GP($X_{1:10}$)-	GP(X)-	Additive GP AddGP($X_{1:2} + X_{3:40}$)-		
EI maximization	EI($X_{1:2}$)	EI($X_{1:10}$)	EI(X)	EI($X_{1:2}$)	EI embed	EI(X)
Optimum (sd)	0.776 (0.221)	1.127 (0.214)	0.669 (0.280)	0.545 (0.210)	0.481 (0.185)	0.986 (0.366)

Table 6.14: Objective function values obtained within 100 (20+80 or 50+50 for the third column) evaluations of the 40-dimensional f_{MG}, with different metamodels and varying EI maximization strategies.

The results in Table 6.14 show that the methods using the additive GP usually outperform those where the GP is built in a more or less truncated X space. The results of GP($X_{1:10}$)-EI($X_{1:10}$) are surprisingly bad. Additional experiments have shown that they seem to be linked with a too small initial DoE. Notice that with another version of f_{MG} (where $\mathbf{c}$ is closer to the boundaries of $X_{3:10}$, not reported here), GP($X_{1:10}$)-EI($X_{1:10}$) outperforms GP($X_{1:2}$)-EI($X_{1:2}$) and the classical GP(X)-EI(X), which is normal since

[7] that is to say EI maximizations, whose optima are evaluated by f_{MG}.

in this situation $X_{3:10}$ become active. However, the AddGP-EI embed and AddGP-EI(X) versions with the additive GP remain better.

The maximization of the EI for the additive GP between the active and inactive components performs the best when the maximization strategy combines the advantage of a low-dimensional active space with a rough maximization in the larger inactive subspace, the AddGP-EI embed strategy. It is also worth mentioning that it is the variant with lowest standard deviation. AddGP-EI(X), searching in a 40 dimensional space, is not able to attain the optimum as well. Even though it is carried out in a very small dimension, AddGP-EI($X_{1:2}$) is also slightly outperformed by AddGP-EI embed, because it cannot optimize the $\mathbf{x}^{\overline{a}}$'s. In this instance of f_{MG} where $\mathbf{c}$ is relatively close to $\mathbf{0}$, AddGP-EI($X_{1:2}$) does not suffer to much from disregarding $\mathbf{x}^{\overline{a}}$'s. However, in the additional experiment where $\mathbf{c}$ is close to the boundaries of $X_{3:10}$, AddGP-EI($X_{1:2}$) exhibits poor results, while AddGP-EI embed still performs well. In this case AddGP-EI(X) performs slightly better than AddGP-EI embed, because it benefits from the maximization over the complete X while the restriction on $\overline{x}$ hinders AddGP-EI embed to get as close to the solution, but AddGP-EI embed still performs reasonably well and has a smaller standard deviation than AddGP-EI(X). For all these reasons, the additive GP with random embedding (AddGP-EI embed) strategy is assessed as the safest one.

6.4.3.2 Experiments with shape optimization

We now turn to the shape optimization of the designs introduced in Section 6.2.3 whose objective functions were defined in Section 6.3.3. We compare the standard approach where the designs are optimized in the CAD parameters space with the methodologies where the surrogate model is built in the eigenshape basis (all variants described in Section 6.4.1). For fair comparison, the same computational effort is put on the internal EI maximization.

Catenoid shape

We want to find a curve $r(y)$ which minimizes the associated axisymmetric surface as expressed by the integral making $f_5(\mathbf{x})$ in the catenoid problem (Example 6.5).

The different versions of Bayesian optimizers that are now tested are the following:

- the standard GP(X)-EI(X) where both the GP and the EI work with the original x's, i.e. CAD parameters;

- GP($\boldsymbol{\alpha}_{\_\_}$)-EI($\boldsymbol{\alpha}_{\_\_}$) indicates the GP is built in the space of $\_\_$ (to be specified) principal components over which the EI is maximized; $\_\_$ are taken equal to 1:4 and 1:7 because, as seen in Table 6.7, 4 and 7 eigencomponents account for 98% and all of the shape variance, respectively.

- GP($\boldsymbol{\alpha}_{\_\_}$)-EI(X) indicates the GP is built in the space of $\_\_$ principal components but the EI is maximized in the X space;

- `AddGP(`$\boldsymbol{\alpha}^a$`+`$\boldsymbol{\alpha}^{\overline{a}}$`)` refers to the additive GP, for which three EI maximizations have been described (Section 6.4.1): `EI embed` where $\boldsymbol{\alpha}^a$ and an embedding in the $\boldsymbol{\alpha}^{\overline{a}}$ space is maximized, `EI(`$\boldsymbol{\alpha}^a$`)` where only the actives $\boldsymbol{\alpha}$'s are maximized (the remaining ones being set to their mean value in $\mathcal{A}_N$, $\mathbf{0}$), and `EI(`$\boldsymbol{\alpha}$`)` where all $\boldsymbol{\alpha}$'s are maximized;

- `GP(`$\boldsymbol{\alpha}^a$`)-EI(`$\boldsymbol{\alpha}^a$`)` means the GP is built over the space of active $\boldsymbol{\alpha}$'s, over which the EI maximization is carried out.

Regarding the EI maximization in $\mathcal{A}$, `on manifold` states that the search is restricted to $\boldsymbol{\alpha}$'s close to $\mathcal{A}_N$. If not, the maximization is carried out in $\mathcal{A}_N$'s covering hyper-rectangle, and `with replication` indicates that both $\boldsymbol{\alpha}^{(t+1)}$ and $\boldsymbol{\alpha}^{(t+1)^*} \notin \mathcal{A}$ are used for the metamodel update, while `no replication` indicates that only the $\boldsymbol{\alpha}^{(t+1)}$'s are considered by the surrogate.

The best objective function values obtained by the algorithms are reported in Table 6.15. They are averaged over 10 runs with different initial DoEs, and standard deviations are given in brackets. The algorithms start with a space-filling DoE of 20 individuals and are run for 60 additional iterations. In the case of the CAD parameters, since $d = 29 > 20$, the initial DoE contains 40 designs and the algorithm is run for 40 iterations. The number of function evaluations to reach certain levels is also reported, to compare the ability of the algorithms to quickly attain near-optimal values. When at least one run has not reached the target, a rough estimator of the empirical runtime (Auger and Hansen, 2005), $\overline{T_s}/p_s$, is provided in red, the number of runs achieving the target value being reported in brackets. $\overline{T_s}$ and p_s correspond to the average number of function evaluations of runs that reach the target and the proportion of runs attaining it.

Method	Best value	Time to 27	Time to 30	Time to 35
`GP(`X`)-EI(`X`)`	31.83 (2.10)	×	570.0 [1]	68.5 (9.9)
`GP(`$\boldsymbol{\alpha}_{1:7}$`)-EI(`$\boldsymbol{\alpha}_{1:7}$`) on manifold`	26.93 (0.18)	86.9 [7]	40.2 (10.5)	40.2 (10.5)
`GP(`$\boldsymbol{\alpha}_{1:7}$`)-EI(`$\boldsymbol{\alpha}_{1:7}$`) with replication`	26.16 (0.10)	30.5 (2.8)	24.3 (0.8)	23.4 (0.5)
`GP(`$\boldsymbol{\alpha}_{1:7}$`)-EI(`$\boldsymbol{\alpha}_{1:7}$`) no replication`	27.62 (0.72)	147.5 [2]	25.4 (2.5)	23.5 (0.5)
`GP(`$\boldsymbol{\alpha}_{1:7}$`)-EI(`$X$`)`	40.57 (11.61)	370.0 [1]	163.3 [3]	120.0 [4]
`AddGP(`$\boldsymbol{\alpha}^a$`+`$\boldsymbol{\alpha}^{\overline{a}}$`)-EI embed on manifold`	50.67 (0.05)	×	×	×
`AddGP(`$\boldsymbol{\alpha}^a$`+`$\boldsymbol{\alpha}^{\overline{a}}$`)-EI embed no replication`	27.58 (0.53)	172.5 [2]	23.6 (1.4)	22.3 (0.7)
`AddGP(`$\boldsymbol{\alpha}^a$`+`$\boldsymbol{\alpha}^{\overline{a}}$`)-EI embed with replication`	26.19 (0.16)	28.4 (4.1)	24.2 (3.1)	22.8 (1.9)
`GP(`$\boldsymbol{\alpha}_{1:4}$`)-EI(`$\boldsymbol{\alpha}_{1:4}$`) with replication`	27.12 (0.13)	550.0 [1]	27.0 (3.9)	25.4 (3.8)

Table 6.15: Best objective function values found and number of iterations required to attain a fixed target (average over 10 runs, standard deviations in brackets) for different metamodels and optimization strategies, on the catenoid problem (Example 6.5). Red figures correspond the empirical runtime, with the number of runs which attained the target in brackets, and × signifies that no run was able to attain it within the limited budget.

Comparing the results in Table 6.15 of the algorithms that stay on the manifold with the others indicates that restricting the search of EI maximizers to the vicinity of $\mathcal{A}_N$ worsens the convergence. Indeed, promising $\boldsymbol{\alpha}$'s are difficult to attain or are even falsely considered as outside $\mathcal{A}$. This observation gets even worse with the additive GP: staying in the neighborhood of $\mathcal{A}_N$ has even stronger consequences because of the restriction to the random line $\bar{\mathbf{a}}$. The EI should therefore be optimized in the covering hyper-rectangle of $\mathcal{A}_N$.

For tackling the issue of EI maximizers $\boldsymbol{\alpha}^{(t+1)*} \notin \mathcal{A}$, the replication strategy exhibits better performance than the strategy where only the projection, $\boldsymbol{\alpha}^{(t+1)}$, is used for updating the GP. Figure 6.36 shows the typical effect of the replication strategy. On the left, the inner EI maximization is carried out in the covering hyper-rectangle of $\mathcal{A}_N$ but only the $\boldsymbol{\alpha} \in \mathcal{A}$ obtained through the pre-image problem solving are used to construct the surrogate model. On the right, all EI maximizers have been used for the GP, including $\boldsymbol{\alpha} \notin \mathcal{A}$. Without replication, since the variance of the GP at previous EI maximizers has not vanished, the EI continues promoting the same $\boldsymbol{\alpha}$'s, which have approximately the same pre-image. The same part of the $\boldsymbol{\alpha}$ space is sampled, which not only leads to a premature convergence (the best observed value has already been attained after 6 iterations), but also increases the risk of getting a singular covariance matrix. With the replication, the GP variance vanishes for all EI maximizers, even those outside $\mathcal{A}$, removing any further EI from these $\boldsymbol{\alpha}$'s. The $\boldsymbol{\alpha}$ space is better explored with benefits on the objective value (26.26 against 27.13 here).

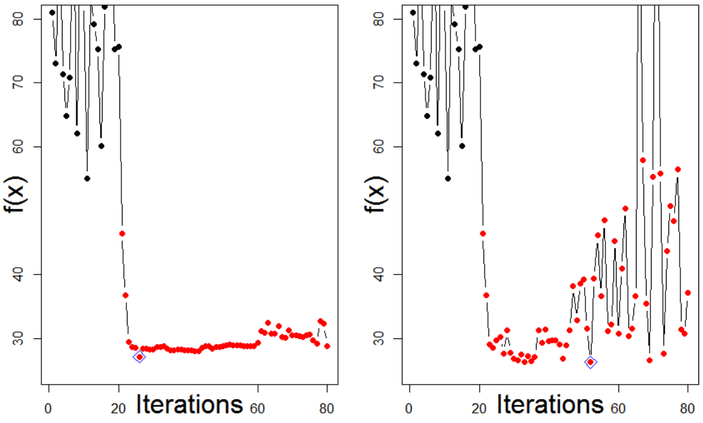


Figure 6.36: Optimization with EI maximization in the covering hyper-rectangle of $\mathcal{A}_N$ without (left) or with (right) replication strategy.

The EI strategy which consists in maximizing via the X space of CAD parameters avoids the $\boldsymbol{\alpha}$ manifold issues. However, it does not perform well, because of the higher dimensional space where the criterion is maximized. An additional drawback for efficient maximization is that ∇EI is not known analytically in this case.

In this catenoid example, the additive GP and the GP in the space of (all) 7 principal components achieve comparable results, both in terms of best value, and of function evaluations to attain the targets. Indeed, the true dimension (7) is relatively low, and we have noticed that the 5, 6 or even 7 first eigenshapes often got classified as active for the additive GP.

Heart rectangle

We now consider Example 6.4 and the minimization of $f_4(\mathbf{x})$ that expresses the distance from a shape to a rectangle deformed as an heart.

As before, different metamodeling and EI maximization options are benchmarked. They include: the standard approach of doing the process in the space of CAD parameters (in dimension $d = 40$); the optimization in the space of 2, 4, 16 or 40 first principal components, where 100% of the shapes variability is recovered with 40 eigencomponents as seen in Table 6.6. Supervised eigenshape selection methods (Section 6.3.2) are also used: the GP built over $\boldsymbol{\alpha}^a$ only, and the additive model over $\boldsymbol{\alpha}^a$ and $\boldsymbol{\alpha}^{\overline{a}}$. For the latter, the 4 EI maximization options of Section 6.4.1 are compared. In light of the above optimization results on the catenoid, the three EI maximization strategies are carried out in the covering hyper-rectangle of $\mathcal{A}_N$ (as opposed to restricted to the neighborhood of $\mathcal{A}_N$), and EI maximizers which do not belong to $\mathcal{A}$ are nonetheless used for the GP update. Henceforth, the `with replication` strategy becomes the new default in all algorithms carrying out EI maximizations in $\boldsymbol{\alpha}$'s and it will no longer be specified in the algorithms names.

The statistics on the solutions proposed by the algorithms are reported in Table 6.16. They consist in the best objective function values averaged over 10 runs with different initial designs, with standard deviations given in brackets. The average and standard deviation of the number of function evaluations to reach certain levels is also given, to compare the ability of the algorithms to quickly attain near-optimal values. When at least one run failed in attaining the target, it is replaced by a rough estimator of the empirical runtime. The algorithms start with a space-filling DoE of 20 individuals and are run for 80 supplementary iterations. In the case of the CAD parameters GP(X)-EI(X) and of GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$), since $d = 40 > 20$, the initial DoE contains 50 designs and the algorithm is run for 50 iterations.

In this test case, as shown in Figure 6.21, the 2 first eigenshapes modify the shape's position, to which f_4 is insensitive. Poor results are therefore obtained by GP($\boldsymbol{\alpha}_{1:2}$)-EI($\boldsymbol{\alpha}_{1:2}$) even though $\mathbf{v}^1$ and $\mathbf{v}^2$ account for 80% of shape reconstruction, highlighting the benefits of the determination of active eigenshapes. In a first order approximation, $\mathbf{v}^3$ and $\mathbf{v}^4$ are the most influential eigenshapes with regard to f_4, which measures the nodal difference between $\Omega_{\mathbf{x}}$ and the target $\Omega_{\mathbf{t}}$. GP($\boldsymbol{\alpha}_{1:4}$)-EI($\boldsymbol{\alpha}_{1:4}$) exhibits very good results, as well as GP($\boldsymbol{\alpha}^a$)-EI($\boldsymbol{\alpha}^a$), which mainly selects $\mathbf{v}^3$ and $\mathbf{v}^4$ ($\mathbf{v}^1$, $\mathbf{v}^2$ and other eigenshapes are sometimes selected too). Even though the shape reconstruction is enhanced, GP($\boldsymbol{\alpha}_{1:16}$)-EI($\boldsymbol{\alpha}_{1:16}$) and GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$) have poor results because of the increase in dimension which is not accompanied by additional information, as already pointed out during the comparison of the predictive capability of these GPs for small budgets, see Table 6.11. GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$) performed better than GP(X)-EI(X) in Table 6.11,

Method	Best value	Time to 0.5	Time to 1	Time to 3
GP(X)-EI(X)	1.18 (0.45)	×	166.9 [4]	42.1 (26.5)
GP($\boldsymbol{\alpha}_{1:2}$)-EI($\boldsymbol{\alpha}_{1:2}$)	9.21 (0.80)	×	×	×
GP($\boldsymbol{\alpha}_{1:4}$)-EI($\boldsymbol{\alpha}_{1:4}$)	0.33 (0.07)	48.8 (21.8)	21.8 (2.2)	21.0 (0.0)
GP($\boldsymbol{\alpha}_{1:16}$)-EI($\boldsymbol{\alpha}_{1:16}$)	0.59 (0.15)	197.8 [3]	50 (15.4)	35.0 (9.7)
GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$)	2.95 (0.97)	×	×	194.4 [5]
GP($\boldsymbol{\alpha}^a$)-EI($\boldsymbol{\alpha}^a$)	0.32 (0.09)	33.7 (9.4)	24.5 (3.7)	21.8 (1.3)
AddGP($\boldsymbol{\alpha}^a$+$\boldsymbol{\alpha}^{\overline{a}}$)-EI($X$)	0.54 (0.19)	199.4 [4]	40.2 (12.3)	30.2 (10.5)
AddGP($\boldsymbol{\alpha}^a$+$\boldsymbol{\alpha}^{\overline{a}}$)-EI embed	0.37 (0.08)	49.0 (21.4)	26.1 (5.6)	22.2 (1.9)
AddGP($\boldsymbol{\alpha}^a$+$\boldsymbol{\alpha}^{\overline{a}}$)-EI($\boldsymbol{\alpha}^a$)	0.37 (0.09)	33.3 (14.6)	22.7 (2.6)	21.4 (0.7)
AddGP($\boldsymbol{\alpha}^a$+$\boldsymbol{\alpha}^{\overline{a}}$)-EI($\boldsymbol{\alpha}$)	0.60 (0.26)	106.7 [6]	41.2 [9]	21.5 (0.5)

Table 6.16: Minimum objective function values found and number of function evaluations required to attain a fixed target (average over 10 runs, standard deviations in brackets) for different metamodels and optimization strategies, rectangular heart problem (Example 6.4). The red figures correspond the empirical runtime, with the number of runs which attained the target in brackets, and × signifies that no run was able to attain it within the limited budget. All algorithms performing an EI search in $\boldsymbol{\alpha}$'s do it `with replication`, the henceforth default.

yet its optimization performance is decreased. This is certainly due to the initial DoE: both DoEs are space-filling in their respective input space (X or the hyper-rectangle of $\boldsymbol{\alpha} \in \mathcal{A}$ containing $\mathcal{A}_N$). However, there is a significant difference between the minima in these DoEs: the average minimum over the 10 runs was 2.57 for GP(X)-EI(X) (hence better than the eventual average best value for GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$)), and 9.22 for GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$). While GPs built over the entire $\boldsymbol{\alpha}$ space (e.g. the additive one) suffer from the same drawback, the selection of variables identifies the dimensions to focus on to rapidly decrease the objective function. This remark applies only to the rectangular heart test case and one may wonder what level of generality it contains. Contrarily to the previous example where building the GP in the space of all (informative) eigenshapes led to the best results, this strategy (GP($\boldsymbol{\alpha}_{1:40}$)-EI($\boldsymbol{\alpha}_{1:40}$)) performs weakly here because of the higher dimension.

The variants of the additive GP perform well too but they are slightly outperformed by GP($\boldsymbol{\alpha}_{1:4}$)-EI($\boldsymbol{\alpha}_{1:4}$). As the objective function mainly depends on $\mathbf{v}^3$ and $\mathbf{v}^4$, always classified as active, strategies that do not put too much emphasis or that neglect $\boldsymbol{\alpha}^{\overline{a}}$ (namely, AddGP($\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$)-EI embed and AddGP($\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$)-EI($\boldsymbol{\alpha}^a$)) perform the best. This explains the good performance of GP($\boldsymbol{\alpha}_{1:4}$)-EI($\boldsymbol{\alpha}_{1:4}$), which disregards $\alpha_5, \dots, \alpha_{40}$. The maximization of the EI with respect to the full $\boldsymbol{\alpha}$ is hindered by the high dimension. Again, the performance decreases when the EI is maximized via the X space. AddGP($\boldsymbol{\alpha}^a +$ $\boldsymbol{\alpha}^{\overline{a}}$)-EI embed and GP($\boldsymbol{\alpha}_{1:4}$)-EI($\boldsymbol{\alpha}_{1:4}$) need more iterations to attain good values (smaller than 0.5) than GP($\boldsymbol{\alpha}^a$)-EI($\boldsymbol{\alpha}^a$) and AddGP($\boldsymbol{\alpha}^a$+$\boldsymbol{\alpha}^{\overline{a}}$)-EI($\boldsymbol{\alpha}^a$) which are early starters. This might be due to the additional though less critical components ($\overline{\alpha}$ or α_1, α_2, respectively) considered by these methods.

NACA 22 optimization

In this last test case, we compare two of the aforementioned algorithms by optimizing the lift coefficient and the drag coefficient of a NACA 22 airfoil (f_{7L} and f_{7D}). The simulation is made with a computational fluids dynamic code that solves the Reynolds Averaged Navier-Stokes (RANS) equations with $k - \varepsilon$ turbulence model, see Chapter 3. Since a single call to the simulator (one calculation of f_7) takes about 20 minutes on a standard personal computer[8], only two runs are compared for each objective. The first algorithm is the classical Bayesian optimizer where the GP is built in CAD parameter space, $\texttt{GP(}X\texttt{)-EI(}X\texttt{)}$. In the second algorithm, $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)-EI embed}$, the GP is built in the $\mathcal{V}$ basis of eigenshapes, while prioritizing the active dimensions, $\boldsymbol{\alpha}^a$, via the additive GP and the EI random embedding method with the replication option, see Section 6.4.2. The optimization in the eigenshape basis starts with a DoE of $n = 10$ designs and is run for $p = 90$ additional iterations while, because there are 22 x_i's, the optimization in the CAD parameters space starts using $n = 50$ designs and is run for $p = 50$ iterations.

Figure 6.37 shows the optimization runs of both algorithms for the minimization of the NACA 22's drag (top) and lift (bottom), and Figure 6.38 the resulting airfoils.

In this application, the main advantage of the $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)-EI embed}$ (Figure 6.37, top left and bottom left) over the standard Bayesian optimizer (top center and bottom center) is that it enables an early search for low drag, respectively high lift airfoils, at a time when the standard approach is still computing its initial DoE. Indeed, the classical method needs much more function evaluations for building the initial surrogate model (black dots) because the inputs live in a space of higher dimension. The approach introduced in this chapter would further gain in relevance in problems with more than $d = 22$ CAD parameters, where it would almost be impossible to build a large enough initial design of experiments (whose size is typically of the order of 10× dimension, Loeppky et al., 2009).

It is observed in Figure 6.38 that smoother airfoils are obtained with $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)-EI embed}$ (right column), because it uses a shape coordinate system instead of treating the L_i's (i.e., x_i's with local influences on the airfoil, see Figure 3.2) separately, as is done by $\texttt{GP(}X\texttt{)-EI(}X\texttt{)}$ (left column). When the optimization aims at minimizing the drag, the $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)-EI embed}$ airfoil (top right) is smoother than the $\texttt{GP(}X\texttt{)-EI(}X\texttt{)}$ one (top left). And when the objective is to maximize the lift, the camber of the $\texttt{AddGP(}\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}\texttt{)-EI embed}$ airfoil (bottom right) is increased in comparison with the design yielded by $\texttt{GP(}X\texttt{)-EI(}X\texttt{)}$ (bottom left).

Multi-objective extension

The method extends to the multi-objective setting considered in Chapters 3, 4 and 5. As all m (independent) GPs are employed for the maximization of the multi-objective infill

[8] Contrarily to Chapters 3, 4, 5, the real simulator is employed in this section instead of the MetaNACA.

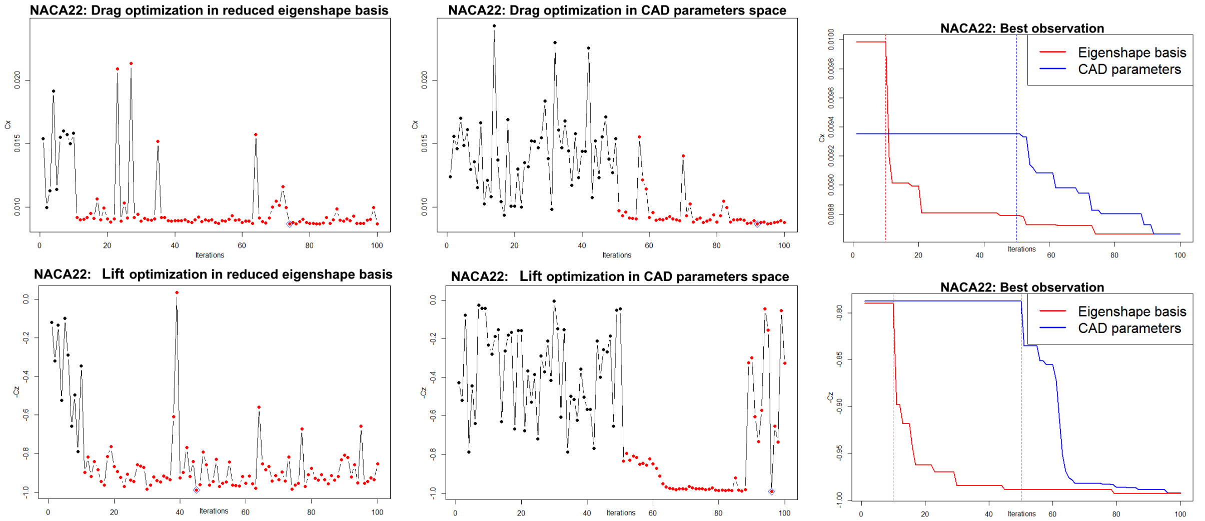
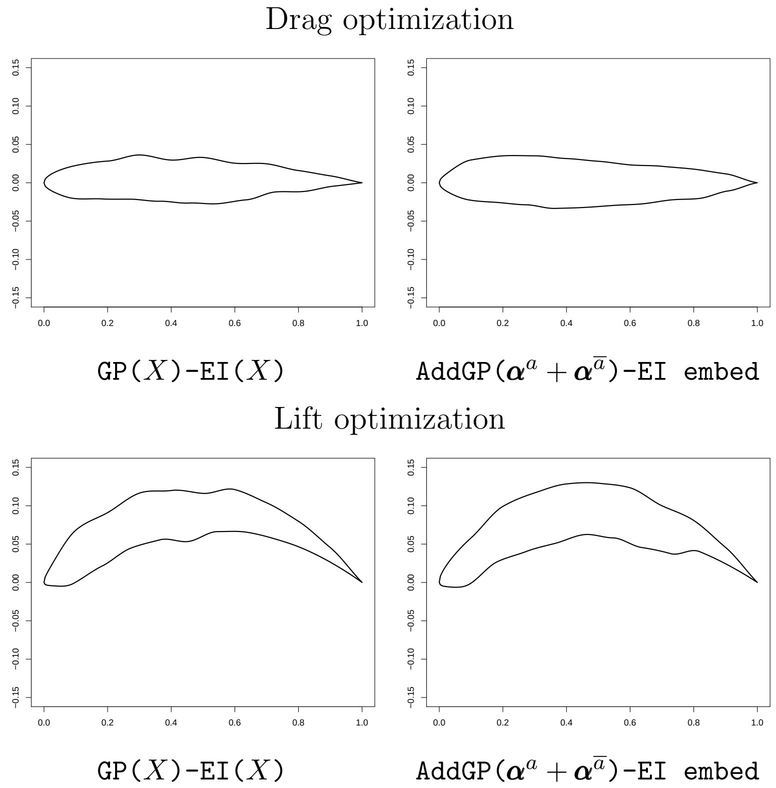


Figure 6.37: Top row: drag optimization of the NACA 22 airfoil in the reduced eigenbasis with `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)-EI embed` (left) or carried out in the CAD parameters space with `GP(`X`)-EI(`X`)` (center). Low drag airfoils are found with `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)-EI embed` while the classical method still evaluates the airfoils of the initial design of experiments (right). Bottom row: lift optimization of the NACA 22 airfoil in the reduced eigenbasis with `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)-EI embed` (left) or carried out in the CAD parameters space with `GP(`X`)-EI(`X`)` (center). High lift airfoils are found while the classical method still evaluates the airfoils of the initial design of experiments (right), i.e., lower objective functions are obtained faster.

criterion (e.g. mEI or EHI), they need to share the same basis. This is a further reason why output-driven dimension reduction techniques such as the Active Subspace Method (Constantine et al., 2014), PLS (Bouhlel et al., 2016; Frank and Friedman, 1993) or SIR (Li, 1991) may not be adapted since they would yield a different basis per objective function. The only way to maximize the infill criterion with metamodels operating on different bases would be to carry out the maximization through the X space by querying the j-th surrogate with $\mathbf{W}_j^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})$, where each projection matrix $\mathbf{W}_j$ depends on the considered objective. But this option has turned out to be the weakest one among the EI maximizations in Section 6.4.3.2 (`GP(`$\boldsymbol{\alpha}$`_)-EI(`X`)` option).

Even though the additive GPs $Y_j(\boldsymbol{\alpha}) = Y_j^a(\boldsymbol{\alpha}^a) + Y_j^{\overline{a}}(\boldsymbol{\alpha}^{\overline{a}})$ operate with $\boldsymbol{\alpha} \in \mathbb{R}^D$, two options of the EI maximization handle an input in lower dimension, $\boldsymbol{\alpha}^a$ or $[\boldsymbol{\alpha}^a, \overline{\alpha}]$. The active components are nonetheless not the same in each objective. Similarly to the multi-objective extension to REMBO (Qian and Yu, 2017), to avoid the maximization of the infill criterion in the large-dimensional space (D) of $\boldsymbol{\alpha}$'s, the latter can prioritize the dimensions that are critical in at least one objective, $\boldsymbol{\alpha}^a = \bigcup_{j=1}^m \boldsymbol{\alpha}^{a_j}$ where $\boldsymbol{\alpha}^{a_j}$ stands for the active dimensions in objective j, as was done in the `EI(`$\boldsymbol{\alpha}^a$`)` or in the `EI embed` strategy.

Another option is to build each of the active GPs $Y_j^a(\cdot)$ over the space of dimensions


Drag optimization

`GP(`X`)-EI(`X`)` `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)-EI embed`

Lift optimization

`GP(`X`)-EI(`X`)` `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)-EI embed`

Figure 6.38: Airfoils found by the compared optimization algorithms. Top: drag minimization, bottom: lift maximization. Left: optimization with the `GP(`X`)-EI(`X`)` algorithm, right: optimization with the `AddGP(`$\boldsymbol{\alpha}^a + \boldsymbol{\alpha}^{\overline{a}}$`)-EI embed` algorithm.

that are active for at least one objective ($\bigcup_{j=1}^{m} \boldsymbol{\alpha}^{a_j}$), and to build the $Y_j^{\overline{a}}(\cdot)$ GPs over the remaining dimensions. The advantage of this approach is that each dimension active in at least one objective is modeled finely by all GPs. The maximization of the infill criterion should benefit from this enhanced precision. The increase in dimension of the $Y_j^a(\cdot)$ GPs may nonetheless result in a weaker accuracy.

6.5 Multi-element shapes

In Example 6.3, any shape is made of $e = 3$ non-overlapping circles. $\mathbf{x} \in \mathbb{R}^9$ are the parameters (position of the center and radius) of these circles. $\mathbf{x}_{1:3}$ correspond to the first circle, $\mathbf{x}_{4:6}$ to the second, and $\mathbf{x}_{7:9}$ to the last circle. Such shapes made of multiple elements raise additional questions regarding their discretization and metamodeling in the $\boldsymbol{\alpha}$ space. Since two designs $\mathbf{x} \neq \mathbf{x}'$ may lead to exactly the same shape hence to the same output, dimension reduction and metamodeling can be enhanced.

6.5.1 Contour discretization

If the design parameters $\mathbf{x}$ describe each of the e elements in the sense that $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_e]$ where $\mathbf{x}_i$ are the parameters of the i-th element, $i = 1, \dots, e$, the first option $\mathcal{D}_1$ is to discretize each element in D/e coordinates and to stack these discretizations in a vector $\boldsymbol{\phi}$ of size D, in the order given by $\mathbf{x}$ (this was the approach followed in 6.3). This is sketched in the left part of Figure 6.39. The drawback is the sensitivity of the mapping to permutations in $\mathbf{x}$: $\boldsymbol{\phi}$ associated to $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_e]$ and $\boldsymbol{\phi}'$ associated to $\mathbf{x}' = [\mathbf{x}_2, \mathbf{x}_1, \dots, \mathbf{x}_e]$ are different even though the shapes are the same. This issue is illustrated in Figure 6.39. The designs $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)^\top$ and $\mathbf{x}' = (x_1, x_2, x_3, x_7, x_8, x_9, x_4, x_5, x_6)^\top$, which correspond to the same shapes $\Omega_\mathbf{x}$ and $\Omega_{\mathbf{x}'}$ are considered. Under $\mathcal{D}_1$, their shape representations $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ are different, as well as their coordinates in the $\mathcal{V}$ basis since $\boldsymbol{\alpha} = \mathbf{V}^\top(\boldsymbol{\phi} - \overline{\boldsymbol{\phi}})$ and $\boldsymbol{\alpha}' = \mathbf{V}^\top(\boldsymbol{\phi}' - \overline{\boldsymbol{\phi}})$ are different.

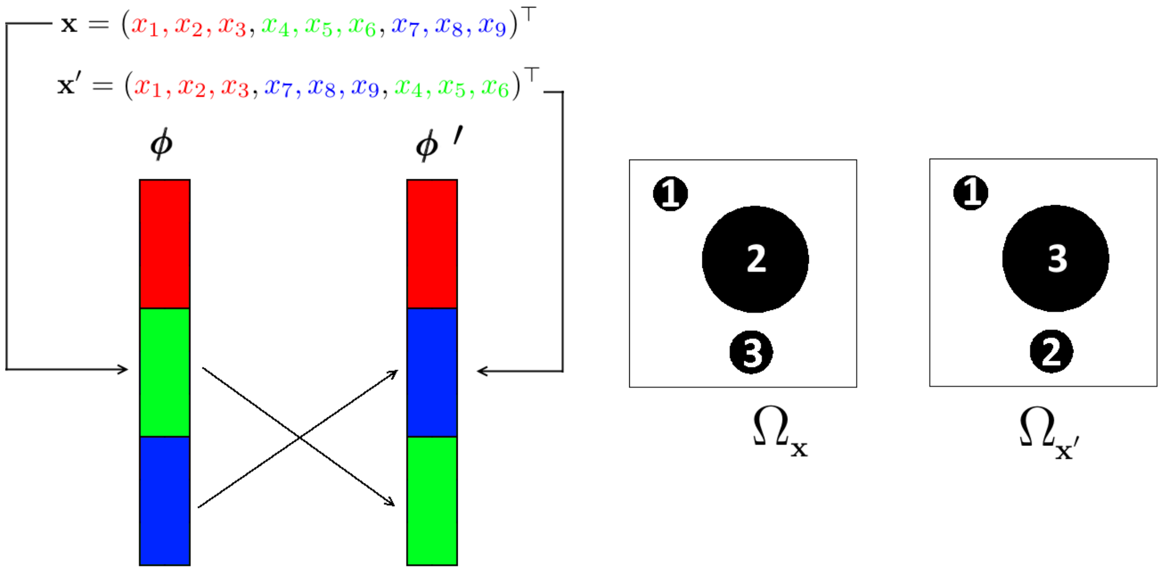

$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)^\top$
$\mathbf{x}' = (x_1, x_2, x_3, x_7, x_8, x_9, x_4, x_5, x_6)^\top$
$\boldsymbol{\phi}$
$\boldsymbol{\phi}'$
1
2
3
1
3
2
$\Omega_\mathbf{x}$
$\Omega_{\mathbf{x}'}$

Figure 6.39: Example of permutation of $\mathbf{x}$ under which the shapes $\Omega_\mathbf{x}$ and $\Omega_{\mathbf{x}'}$ are invariant. However, the shape representations $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ yielded by $\mathcal{D}_1$ are different.

A second option, $\mathcal{D}_2$, is to discretize each element and to stack all nodal coordinates (regardless of the element to which they belong) in their order of appearance when scanning the physical space ($\mathbb{R}^2$ or $\mathbb{R}^3$) in a given order. In this case, the $\boldsymbol{\phi}$'s in Figure 6.39 are the same. The drawback is that contiguous elements in the resulting $\boldsymbol{\phi}$ may no longer belong to the same element.

The last option, $\mathcal{D}_3$, is to discretize each element and to stack it entirely in the order the element appears when scanning the physical space in a given order. In this variant, two designs with swapped elements $\mathbf{x}$ and $\mathbf{x}'$ lead to the same $\boldsymbol{\phi}$. By scanning the physical space from left to right and from top to bottom, the circle 1 appears before the circle 2 and the circle 3 in $\Omega_\mathbf{x}$. $\boldsymbol{\phi}$ is the concatenated discretization of circle 1, circle 2 and circle 3 (see Figure 6.39). In $\Omega_{\mathbf{x}'}$, the circle 1 appears before the circle 3 and the circle 2, hence $\boldsymbol{\phi}'$ is the concatenation of the discretized circle 1, circle 3 and circle 2. With $\mathcal{D}_3$, $\Omega_{\mathbf{x}} = \Omega_{\mathbf{x}'} \Rightarrow \boldsymbol{\phi} = \boldsymbol{\phi}'$, even though $\mathbf{x} \neq \mathbf{x}'$. The drawback of this approach is that a small variation in $\mathbf{x}$ may change the order of appearance of the elements hence the way the discretized elements are stacked in $\boldsymbol{\phi}$. Large discontinuities may therefore exist in this variant of the $\phi(\cdot)$ mapping.

6.5.2 PCA and eigenshapes

Depending on the discretization variant, the database $\boldsymbol{\Phi}$ significantly differs as $\phi(\mathbf{x})$ is different whether $\phi = \mathcal{D}_1$, $\mathcal{D}_2$ or $\mathcal{D}_3$. The left plot of Figure 6.40 shows the cumulative sum of the PCA eigenvalues for the 3 circles example. $\mathcal{D}_3$ is the variant where the first λ_j's are the largest. Exactly 100% of reconstruction in Φ is attained when the $d = 9$ first eigenshapes are considered, as is also the case for $\mathcal{D}_1$. However, the true dimension is not retrieved with $\mathcal{D}_2$ (green curve). The correlation between discretizations in $\boldsymbol{\Phi}$ is degraded because the order of appearance of nodal coordinates in each $\boldsymbol{\phi}$ highly depends on the design, and $\boldsymbol{\Phi}$ contains vectors that are not the contiguous discretization of the different elements. The mean shape $\overline{\boldsymbol{\phi}}$ and the eigenshapes $\mathbf{v}^j$ are spurious discretizations (they are no longer circles) that do not help reducing the dimension in Φ.

The mean shapes obtained through $\mathcal{D}_1$ and $\mathcal{D}_3$ are shown in the middle and in the right plot of Figure 6.40. For $\mathcal{D}_1$, since the $\mathbf{x}^{(i)}$'s have been sampled uniformly, $\overline{\boldsymbol{\phi}}$ is a shape of three centered circles. For $\mathcal{D}_3$, as the shapes have been discretized in a prescribed order of appearance (left to right here), the first element of the mean shape (blue) is the left-most circle, the second circle (red) is in the center, and the third one (green) is on the right. Differences are perceived among the resulting eigenshapes as shown in Figure 6.41 (recall that eigenvectors that appear as points displace the shape in the direction specified by the eigenvector's position). Since an ordering of the circles is induced by $\mathcal{D}_3$, the eigenshapes move or grow in the x-axis direction or in the y-axis direction, but never on both (the centers of the eigenvectors have always x-coordinate 0 or y-coordinate 0) contrarily to the eigenvectors of $\mathcal{D}_1$.

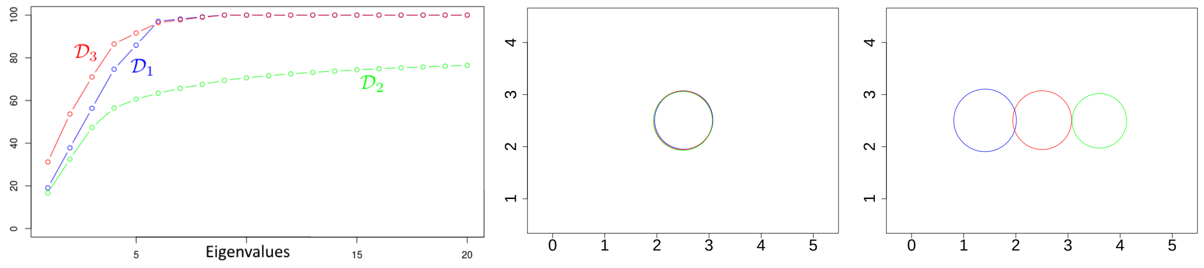


Figure 6.40: Eigendecomposition for different multi-element mappings, 3 circles test case. Left: PCA eigenvalues cumulative sum (in %). Middle: mean shape $\overline{\boldsymbol{\phi}}$ when $\phi = \mathcal{D}_1$. Right: mean shape $\overline{\boldsymbol{\phi}}$ when $\phi = \mathcal{D}_3$.

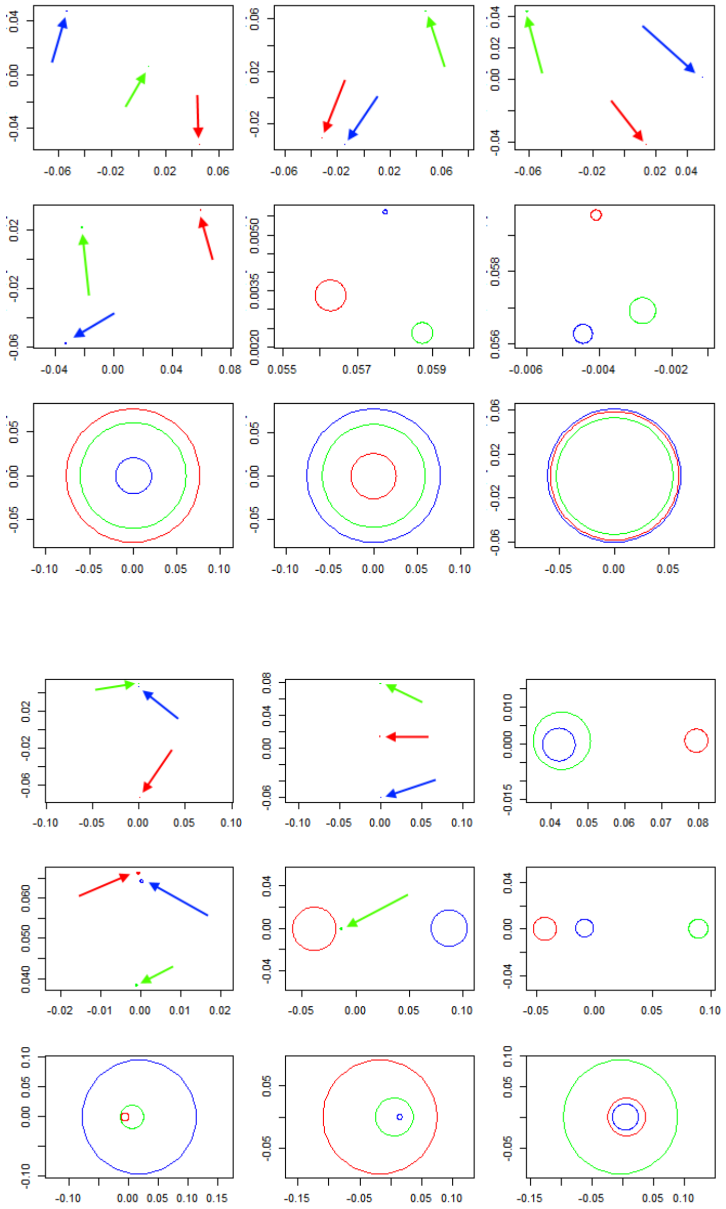


Figure 6.41: Example 6.3, three circles with $d = 9$ parameters, 9 first eigenvectors (from left to right, top to bottom) when $\phi = \mathcal{D}_1$ (9 top plots) or $\phi = \mathcal{D}_3$ (9 bottom plots). The blue part of each eigenvector acts on the first circle, the red part acts on the second circle and the green part on the third circle.

Element-wise PCA

Previously, PCA was carried out on $\mathbf{\Phi}$ by searching the eigenvectors of the covariance matrix $\mathbf{C}_{\mathbf{\Phi}} = \frac{1}{N}(\mathbf{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top)^\top(\mathbf{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top)$. Since each row of $\mathbf{\Phi}$ is composed of the discretization of three distinct circles in $\mathcal{D}_1$ (and in $\mathcal{D}_3$ also), it might be of interest to enforce a null correlation among nodes belonging to different circles by setting $\mathbf{C}_{\mathbf{\Phi}ij} = 0$ when the discretizations indices i and j do not correspond to the same element, as schematically shown in the upper plot of Figure 6.42. The $d = 9$ first eigenvectors obtained after applying an eigendecomposition of this modified $\mathbf{C}_{\mathbf{\Phi}}$ explain 100% of shape discretization variance (measured by the eigenvalues), i.e. the effective dimension is still retrieved. As correlations between different elements have been erased, the eigenvectors $\mathbf{v}^j$ act on each element individually: the part of $\mathbf{v}^j$ associated to two elements out of three is null as shown in Figure 6.42, meaning that the reconstruction of $e - 1$ elements is insensitive to $\mathbf{v}^j$. This is equivalent to applying e independent PCAs on the columns of $\mathbf{\Phi}$ associated to each element, and to augment each eigenvector by $\mathbf{0}$ at discretization indices corresponding to other elements.

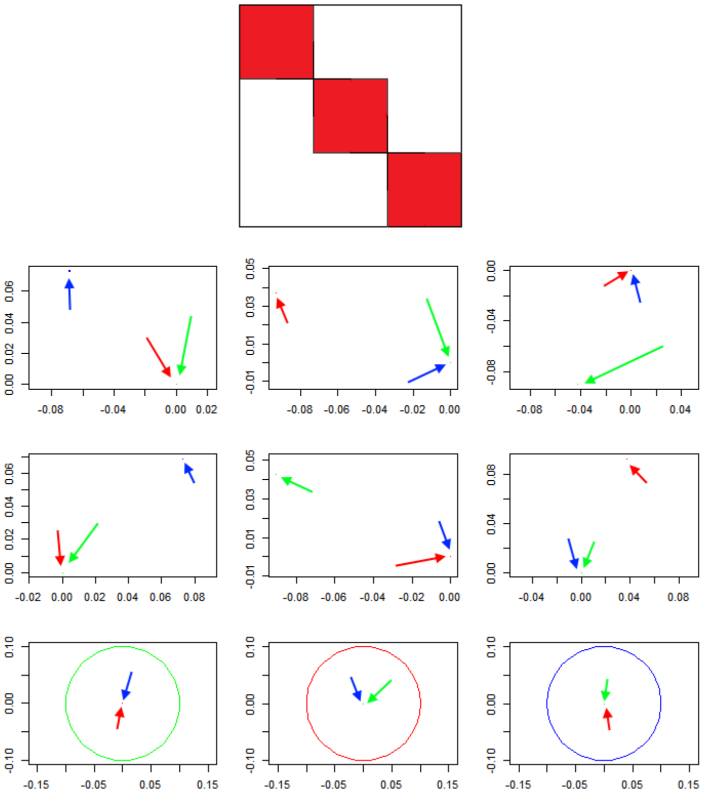


Figure 6.42: Top: shape of the correlation matrix $\mathbf{C}_{\mathbf{\Phi}}$ when enforcing a null correlation (white) between nodes of distinct elements. Bottom: 9 eigenvectors of $\mathbf{C}_{\mathbf{\Phi}}$. Since correlations between different circles have been removed, each $\mathbf{v}^j$ operates on one element.

This approach which yields eigenvectors associated to one unique element may be attractive if the underlying problem depends on each element. However, the issue of permutation sensitivity of $\mathcal{D}_1$ is not resolved. This subject is further addressed in the next sections. Finally, in the case of non-overlapping circles, the position of one circle informs about locations where the other circles are not. A null correlation between elements might therefore be too high an assumption. Additionally, the elements in $\mathbf{x}$ are not always randomly ordered and may reflect choices of the designer.

6.5.3 Symmetries and kriging in $\mathcal{V}$

In this part, we consider only $\mathcal{D}_1$ as ϕ mapping. Even though promising eigenshapes have been found by $\mathcal{D}_3$, it seems easier to circumvent $\mathcal{D}_1$'s drawback (permutations) than $\mathcal{D}_3$'s (lack of continuity).

Two options are considered in this part for tackling the issue of swapped elements in designs having the same shape. First, it is proposed to modify the mean shape and the eigenvectors such that the permutations $\tau \in \boldsymbol{\tau} := \{\tau : X \to X, \Omega_{\tau\circ\mathbf{x}} = \Omega_{\mathbf{x}}\}$ are (different) permutations in the $\boldsymbol{\alpha}$ space too. In this case, permutation invariant kriging (Ginsbourger et al., 2012) with respect to τ's pendants in the $\boldsymbol{\alpha}$ space can be performed in the $\mathcal{V}$ basis.

The second option is to perform invariant kriging in the space of $\boldsymbol{\alpha}$'s without modifying the $\mathbf{v}^j$'s. This is done by propagating the permutations $\sigma \in \boldsymbol{\sigma} := \{\sigma : \Phi \to \Phi : \exists \tau \in \boldsymbol{\tau}, \sigma \circ \phi(\mathbf{x}) = \phi(\tau \circ \mathbf{x})\}$ under which discretizations $\boldsymbol{\phi}$ and $\sigma \circ \boldsymbol{\phi}$ correspond to the same shape into the space of eigenshapes.

6.5.3.1 $\boldsymbol{\alpha}$ invariance by eigenshape modification

In the case of the circle with 9 parameters, the permutation τ that changes $(x_1, \dots, x_9)^\top$ into $(x_4, \dots, x_9, x_1, x_2, x_3)^\top$ is one element of $\boldsymbol{\tau}$ for which $\Omega_{\mathbf{x}} = \Omega_{\tau\circ\mathbf{x}}$. Nevertheless, $\phi(\mathbf{x}) \neq \phi(\tau \circ \mathbf{x})$ when $\phi = \mathcal{D}_1$, because the elements of $\mathbf{x}$ and of $\tau \circ \mathbf{x}$ are discretized in the order specified by $\mathbf{x}$. τ's pendant in the space of shape discretizations is σ which verifies $\sigma \circ \phi(\mathbf{x}) = \phi(\tau \circ \mathbf{x})$. As $\boldsymbol{\alpha} = \mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})$ and $\boldsymbol{\alpha}' = \mathbf{V}^\top(\phi(\tau \circ \mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top(\sigma \circ \phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})$, to obtain $\boldsymbol{\alpha} = \boldsymbol{\alpha}'$, both $\overline{\boldsymbol{\phi}}$ and the $\mathbf{v}^j$'s need to be invariant to all $\sigma \in \boldsymbol{\sigma}$. Indeed, let $\sigma \circ \overline{\boldsymbol{\phi}} = \overline{\boldsymbol{\phi}}$ and $\sigma \circ \mathbf{v}^j = \mathbf{v}^j$, $j = 1, \dots, D$. Then

$$\boldsymbol{\alpha}' = \mathbf{V}^\top(\sigma \circ \phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top(\sigma \circ \phi(\mathbf{x}) - \sigma \circ \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top(\sigma \circ (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}))$$

Therefore, the j-th component of $\boldsymbol{\alpha}'$ is

$$\alpha'_j = \sum_{k=1}^{D} v^j_k(\sigma \circ (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}))_k = \sum_{k=1}^{D} \sigma \circ (v^j_k(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})_k) = \sum_{k=1}^{D} v^j_k(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})_k = \alpha_j$$

because the sum is invariant to the permutation of a vector.

The following modification of $\overline{\boldsymbol{\phi}}$ and of the $\mathbf{v}^j$'s obtained through PCA

$$\overline{\boldsymbol{\phi}} \leftarrow \Bigg[\underbrace{\frac{1}{e}\sum_{i=1}^{e}\overline{\boldsymbol{\phi}}_i, \dots, \frac{1}{e}\sum_{i=1}^{e}\overline{\boldsymbol{\phi}}_i}_{e \text{ times}}\Bigg] \quad \text{and} \quad \mathbf{v}^j \leftarrow \Bigg[\underbrace{\frac{1}{e}\sum_{i=1}^{e}\mathbf{v}^j_i, \dots, \frac{1}{e}\sum_{i=1}^{e}\mathbf{v}^j_i}_{e \text{ times}}\Bigg], \; j = 1, \dots, D$$

where $\overline{\boldsymbol{\phi}}_i$ and $\mathbf{v}^j_i \in \mathbb{R}^{D/e}$ are the mean discretization of the i-th element, and the discretization of the i-th element in the j-th eigenvector respectively, makes them invariant to any $\sigma \in \boldsymbol{\sigma}$ and leads to $\boldsymbol{\alpha} = \boldsymbol{\alpha}'$ where $\boldsymbol{\alpha}'$ are the coefficients in the $\mathcal{V}$ basis of $\tau \circ \mathbf{x}$. The drawback of this approach is that the rank of the modified $\mathbf{V}$ matrix is divided by e, and since $\overline{\boldsymbol{\phi}}$ and $\mathbf{v}^j$'s reconstruct each of the e elements identically (see Equation 6.3), only designs with element-wise identical discretization are recovered. Finally, more designs than just those corresponding to a τ permutation[9] of $\mathbf{x}$ have the same $\boldsymbol{\alpha}$.

6.5.3.2 Retrieving σ in the space of $\boldsymbol{\alpha}$'s

Instead of obtaining $\boldsymbol{\alpha} = \boldsymbol{\alpha}'$ when $\mathbf{x}' = \tau \circ \mathbf{x}$, here we aim at retrieving $\sigma \circ \boldsymbol{\alpha} = \boldsymbol{\alpha}'$ by modifying $\overline{\boldsymbol{\phi}}$ and the $\mathbf{v}^j$'s. To be σ invariant, first, the $\overline{\boldsymbol{\phi}} \leftarrow \left[\underbrace{\frac{1}{e}\sum_{i=1}^{e}\overline{\boldsymbol{\phi}}_i, \dots, \frac{1}{e}\sum_{i=1}^{e}\overline{\boldsymbol{\phi}}_i}_{e \text{ times}}\right]$ modification is considered. Next, the eigendecomposition of $\mathbf{C}_{\boldsymbol{\Phi}} = \frac{1}{N}(\boldsymbol{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top)^\top(\boldsymbol{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top)$ is completed. As $\overline{\boldsymbol{\phi}}$ has been modified, it is not the standard way of doing PCA since the $\boldsymbol{\Phi} - \mathbf{1}_N\overline{\boldsymbol{\phi}}^\top$ matrix is not centered. The issue in the previous section was that the $\mathbf{v}^j$'s reconstructed all elements in the same manner. To enforce the eigenvectors to be specific to each element, let $\mathbf{C}_{\boldsymbol{\Phi}ij} = 0$ if the discretization indices i and j do not belong to the same element, as in Section 6.5.2. Under the assumption that each element has the same importance in the eigendecomposition, the submatrices of $\mathbf{V}$, $\mathbf{v}^{(1:e)}$, $\mathbf{v}^{(e+1:2e)}, \dots$ denoted as $\mathbf{V}^{(j)} := (\mathbf{v}^{(j-1)e+1}, \mathbf{v}^{(j-1)e+2}, \dots, \mathbf{v}^{(j-1)e+e} = \mathbf{v}^{je}) \in \mathbb{R}^{D\times e}$, $j = 1, \dots, D/e$, are block diagonal with D/e non-zero entries per column. Among a $\mathbf{V}^{(j)}$, the non-zero entries, which apply to the discretization of one among the e elements, are averaged to yield the vector $\widetilde{\mathbf{v}}^j \in \mathbb{R}^{D/e}$, and the non-zero entry of each vector inside $\mathbf{V}^{(j)}$ is replaced by $\widetilde{\mathbf{v}}^j$. This makes the eigenvectors specific to one element. At the same time, if a permutation σ is applied to $\boldsymbol{\phi}$, the multiplication between the modified $\mathbf{V}^{(j)}$ matrix and $\sigma \circ \boldsymbol{\phi}$ conserves the permutation because the non zero entries of $\mathbf{V}^{(j)}$ are the same. The original $\mathbf{V}^{(j)}$ submatrix as well as its modification are illustrated in Figure 6.43.

Finally, the redefined eigenvector matrix is $\mathbf{V} = (\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(D/e)}) = (\mathbf{v}^1, \dots, \mathbf{v}^D)$. When a permutation τ is applied to $\mathbf{x}$, its counterpart σ is retrieved in $\boldsymbol{\alpha}$. Indeed, with the modified $\mathbf{V}$ and $\overline{\boldsymbol{\phi}}$, let

$$\boldsymbol{\alpha} = \mathbf{V}^\top(\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) \quad \text{and} \quad \boldsymbol{\alpha}' = \mathbf{V}^\top(\phi(\tau \circ \mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top(\sigma \circ \phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top(\sigma \circ (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}))$$

$\boldsymbol{\alpha}'$'s j-th component is $\alpha'_j = \sum_{i=1}^{D} v_i^j(\sigma \circ (\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}))_i$. Since there exist $e-1$ vectors in $\mathbf{V}$ that are the same as $\mathbf{v}^j$ modulo a σ permutation (these vectors are the $e-1$ ones that belong to the same $\mathbf{V}^{()}$ matrix as $\mathbf{v}^j$), and as only D/e components of $\mathbf{v}^j$ are non zero (see Figure 6.43), there exists a vector of $\mathbf{V}$, $\mathbf{v}^{\tilde{j}}$, such that $\sum_{i=1}^{D} v_i^j(\sigma \circ (\phi(\mathbf{x}) -$

[9] For instance, discretizations with identical $\sum_{k=1}^{e} \boldsymbol{\phi}_{i+\frac{D}{e}(k-1)} =: \widetilde{\boldsymbol{\phi}}_i$'s, $\forall i = 1, \dots, D/e$, that is to say discretizations with same sum (through the e elements) of nodal coordinates $i = 1, \dots, D/e$, have the same $\boldsymbol{\alpha}$'s.

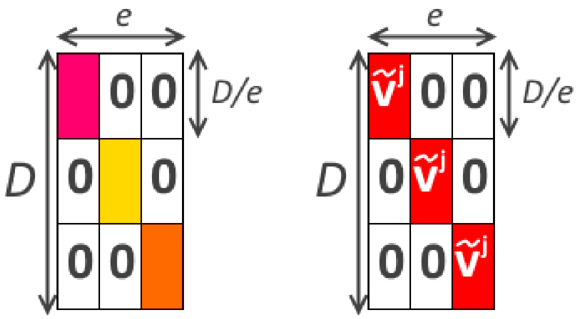


Figure 6.43: Modification of the $\mathbf{V}^{(j)}$ submatrix in an example with $e = 3$ elements. Left: the original $\mathbf{V}^{(j)}$ matrix obtained by eigendecomposition of the sparse $\mathbf{C}_{\boldsymbol{\Phi}}$ matrix. Right: modified $\mathbf{V}^{(j)}$ matrix. The average of the pink, yellow and orange vector in the left matrix is $\tilde{\mathbf{v}}^j$, which is employed in $\mathbf{V}^{(j)}$ such that these e vectors apply the same transformation to their corresponding element.

$\bar{\boldsymbol{\phi}}))_i = \sum_{i=1}^D v_i^{\tilde{j}}(\phi(\mathbf{x}) - \bar{\boldsymbol{\phi}})_i = \alpha_{\tilde{j}}$. Therefore, $\alpha'_j = \alpha_{\tilde{j}}$. More precisely, $\mathbf{v}^{\tilde{j}}$ is the inverse permutation σ^{-1} of $\mathbf{v}^j$. Indeed,

$$\sum_{i=1}^{D} v_i^j(\sigma \circ (\phi(\mathbf{x}) - \bar{\boldsymbol{\phi}}))_i = \sum_{i=1}^{D} (\mathbf{v}^j)_i(\sigma \circ (\phi(\mathbf{x}) - \bar{\boldsymbol{\phi}}))_i$$

$$= \sum_{i=1}^{D} \Big(\sigma^{-1} \circ (\mathbf{v}^j \odot (\sigma \circ (\phi(\mathbf{x}) - \bar{\boldsymbol{\phi}})))\Big)_i = \sum_{i=1}^{D} (\underbrace{\sigma^{-1} \circ \mathbf{v}^j}_{\mathbf{v}^{\tilde{j}}})_i(\phi(\mathbf{x}) - \bar{\boldsymbol{\phi}})_i$$

Therefore $\boldsymbol{\alpha}' = \sigma^{-1} \circ \boldsymbol{\alpha}$, hence $\sigma \circ \boldsymbol{\alpha}' = \boldsymbol{\alpha}$: with these adapted $\bar{\boldsymbol{\phi}}$ and $\mathbf{V}$, τ's pendant is retrieved in the $\boldsymbol{\alpha}$ space, as shown in Figure 6.44.

The evidenced σ permutation between $\boldsymbol{\alpha}$'s coming from $\mathbf{x}$'s that differ by the related permutation τ can be exploited inside permutation-invariant kriging (Ginsbourger et al., 2012). A GP employing the covariance kernel $k_{perm}(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\alpha}^{(j)}) := \frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma \in \boldsymbol{\sigma}} k(\boldsymbol{\alpha}^{(i)}, \sigma \circ \boldsymbol{\alpha}^{(j)})$ is invariant to any permutation of $\boldsymbol{\sigma}$, i.e. $Y(\boldsymbol{\alpha}) = Y(\sigma \circ \boldsymbol{\alpha})$. An example of invariant kriging with respect to the first bisector ($(x_1, x_2) \mapsto (x_2, x_1)$) is shown in the left part of Figure 6.45. Only designs with $x_1 > x_2$ have been observed but the invariant kernel imposes the predictor to be symmetric with respect to the purple line. If training points (x_1, x_2) are associated to a response y, the prediction at (x_2, x_1) is y too and the uncertainty is null there. Predictions are improved in comparison with a GP with a non-symmetric kernel (right plot).

One drawback of this option is that the columns of $\mathbf{V}$ no longer form an orthonormal basis. Another possible issue is that $\tilde{\mathbf{v}}^j$ is the average over the non-zero entries of the e eigenvectors in $\mathbf{V}^{(j)}$. This makes sense in our problem because the vectors that belong to the same $\mathbf{V}^{(j)}$ apply more or less the same transformation to each circle since the designs have been sampled uniformly, and the same kind of patterns are retrieved for the circles

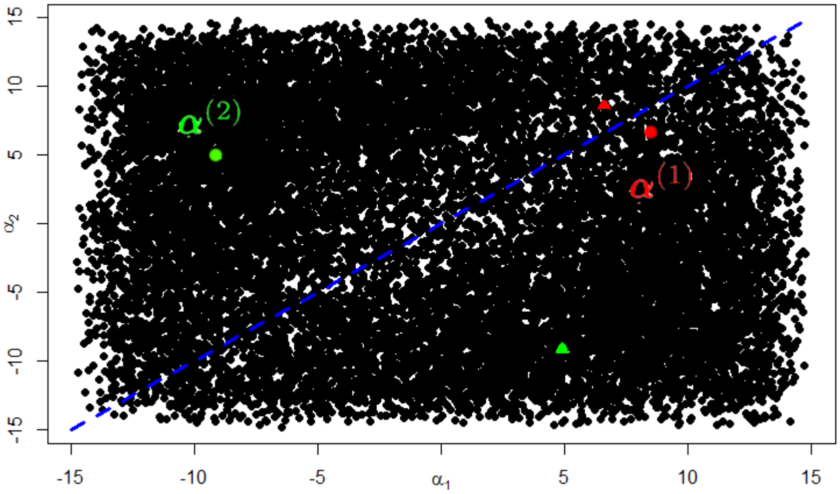


Figure 6.44: Coordinates in the eigenvector basis (in the (α_1, α_2) plan) of the N designs. Two designs $\boldsymbol{\alpha}^{(1)}$ (red circle) and $\boldsymbol{\alpha}^{(2)}$ (green circle) are shown as well as the coordinates in the $\boldsymbol{\alpha}$ space of the permuted designs (when τ interchanges (x_1, x_2, x_3) and (x_4, x_5, x_6), triangles), $\boldsymbol{\alpha}^{(i)\prime} = \mathbf{V}^\top(\phi(\tau \circ \mathbf{x}^{(i)}) - \overline{\boldsymbol{\phi}})$. The symmetry along the line $\alpha_1 = \alpha_2$ is clearly visible.

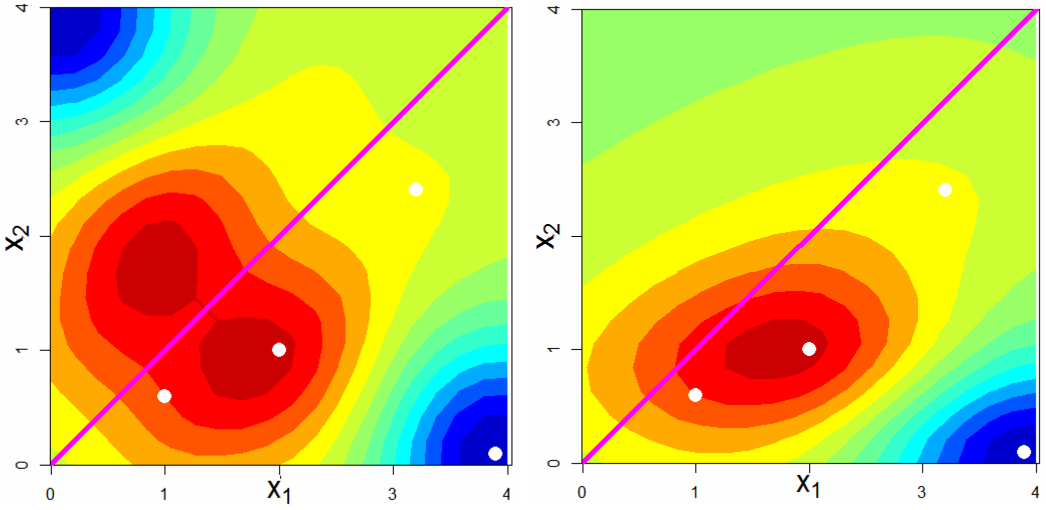


Figure 6.45: Left: example of kriging with invariance with respect to the first bisector (purple line). Right: GP with standard kernel. Predictions are enhanced by providing the symmetry information.

within $\mathbf{\Phi}$ (for instance, the rows of Figure 6.42 correspond to $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)}$ and are similar). However, having quite different $\mathbf{v}$'s inside the same $\mathbf{V}^{(j)}$ would be problematic.

A more robust implementation of permutation-insensitive kriging should not modify the eigenshapes and retrieve a permutation in the rotated space spanned by the $\mathbf{v}^j$'s. This is the subject of the next part.

6.5.3.3 Invariant kriging in the eigenbasis

The permutations σ in the Φ space can be expressed as matrices $\mathbf{A}_\sigma \in \mathbb{R}^{D\times D}$ which contain ones in off-centered diagonals. The permutations $\mathbf{A}_\sigma\boldsymbol{\phi}$ of $\boldsymbol{\phi}$ that lead to the same shape are not exhibited directly in the $\boldsymbol{\alpha}$ space. Figure 6.46 shows the coordinates in the (α_1, α_2) plane of two $\boldsymbol{\alpha} = \mathbf{V}^\top(\boldsymbol{\phi}-\overline{\boldsymbol{\phi}})$ (red and green circles) together with the coordinates of the 5 discretizations having the same shape, $\mathbf{V}^\top(\mathbf{A}_\sigma\boldsymbol{\phi} - \overline{\boldsymbol{\phi}})$, $\sigma \in \boldsymbol{\sigma}$ (red and green triangles). No symmetry can be directly retrieved at first glance, even if the relative position of these points appears to exhibit some regularity.

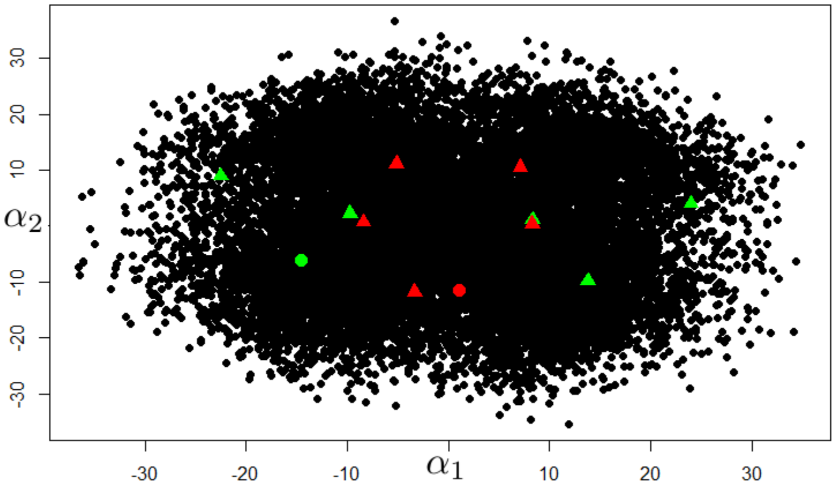


Figure 6.46: Two $\boldsymbol{\alpha}$'s (red and green circle) and their discretizations in the $\mathcal{V}$ basis which correspond to an identical shape (triangle). Symmetries cannot be retrieved directly in $\mathcal{V}$ (test case with three circles).

However, by multiplying the $\boldsymbol{\alpha}$ coordinates of a permuted discretization $\mathbf{A}_\sigma\phi(\mathbf{x})$ by the matrix $\mathbf{V}_\sigma := \mathbf{V}^\top\mathbf{A}_\sigma^{-1}\mathbf{V}$, one gets

$$\mathbf{V}_\sigma\mathbf{V}^\top(\mathbf{A}_\sigma\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top\mathbf{A}_\sigma^{-1}\mathbf{V}\mathbf{V}^\top(\mathbf{A}_\sigma\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top\phi(\mathbf{x}) - \mathbf{V}^\top\mathbf{A}_\sigma^{-1}\overline{\boldsymbol{\phi}}$$

i.e. the original $\boldsymbol{\alpha}$ is retrieved as far as $\overline{\boldsymbol{\phi}}$ is σ invariant, $A_\sigma\overline{\boldsymbol{\phi}} = \overline{\boldsymbol{\phi}}\ \forall\sigma \in \boldsymbol{\sigma}$. This assumption is almost true when using the discretization $\mathcal{D}_1$ (see central plot of Figure 6.40, the 3 circles are nearly the same, and interchanging the indices of the circle's discretizations leads to almost the same $\overline{\boldsymbol{\phi}}$), and can be enforced by setting $\overline{\boldsymbol{\phi}} \leftarrow \left[\underbrace{\frac{1}{e}\sum_{i=1}^{e}\overline{\boldsymbol{\phi}}_i, \dots, \frac{1}{e}\sum_{i=1}^{e}\overline{\boldsymbol{\phi}}_i}_{e \text{ times}}\right]$.

There is a $\mathbf{V}^\top\mathbf{A}_\sigma^{-1}\mathbf{V}$ shape-invariance in the $\boldsymbol{\alpha}$ space. More generally, considering a multi-element shape described by $\mathbf{A}_{\sigma_i}\phi(\mathbf{x})$ with respect to an initial ordering of the elements, applying any of the valid permutation to its $\boldsymbol{\alpha}$ yields another permuted $\boldsymbol{\alpha}$,

$$\mathbf{V}_{\sigma_j}\boldsymbol{\alpha} = \mathbf{V}_{\sigma_j}(\mathbf{V}^\top(\mathbf{A}_{\sigma_i}\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}})) = \mathbf{V}^\top(\mathbf{A}_{\sigma_j}^{-1}\mathbf{A}_{\sigma_i}\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}) = \mathbf{V}^\top(\mathbf{A}_{\sigma_k}\phi(\mathbf{x}) - \overline{\boldsymbol{\phi}}).$$

It is therefore possible to perform $\mathbf{V}_\sigma$-invariant kriging by employing the kernel $k_{\text{inv}}(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}) := \frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma\in\boldsymbol{\sigma}} k(\boldsymbol{\alpha}^{(1)}, \mathbf{V}_\sigma\boldsymbol{\alpha}^{(2)})$ where $k(\cdot,\cdot)$ is a usual covariance function, e.g. a Matérn, Squared Exponential, Exponential, ... kernel, see Example 2.1.

Remark that invariant kernels are not necessarily stationary: in a two-dimensional problem, let $\bar{k}(\mathbf{x}, \mathbf{x}') := \frac{1}{2}\left(k(\mathbf{x}, \mathbf{x}') + k(\mathbf{x}, \sigma_{21}\circ\mathbf{x}')\right)$ be a first bisector invariant kernel where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$, and σ_{21} is the first bisector permutation, $\sigma_{21} : (x_1, x_2) \mapsto (x_2, x_1)$.

One can easily verify that $\bar{k}(\cdot,\cdot)$ implements the symmetry invariance about the first bisector as $\bar{k}((0,2)^\top, (0,2)^\top) = \bar{k}((0,2)^\top, (2,0)^\top)$. However, $\bar{k}((0,2)^\top, (2,0)^\top) \neq \bar{k}((0,1)^\top, (1,0)^\top)$: the correlation depends on the distance to the $x_1 = x_2$ line, which is undesirable as both points are the same under the permutation invariance assumption. The following kernel (Rasmussen and Williams, 2006) is a stationary version of $\bar{k}(\cdot,\cdot)$: $\tilde{k}(\mathbf{x}, \mathbf{x}') = \frac{\bar{k}(\mathbf{x},\mathbf{x}')}{\sqrt{\bar{k}(\mathbf{x},\mathbf{x})}\sqrt{\bar{k}(\mathbf{x}',\mathbf{x}')}}$. It exhibits symmetry along the first bisector while being stationary, i.e. it solely depends on the distance between (the symmetrized) $\mathbf{x}$ and $\mathbf{x}'$.

In the same vein, the final symmetric kernel in the $\boldsymbol{\alpha}$ space is

$$\tilde{k}_{\text{inv}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \frac{\frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma\in\boldsymbol{\sigma}} k(\boldsymbol{\alpha}, \mathbf{V}_\sigma\boldsymbol{\alpha}')}{\sqrt{\frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma\in\boldsymbol{\sigma}} k(\boldsymbol{\alpha}, \mathbf{V}_\sigma\boldsymbol{\alpha})}\sqrt{\frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma\in\boldsymbol{\sigma}} k(\boldsymbol{\alpha}', \mathbf{V}_\sigma\boldsymbol{\alpha}')}} \tag{6.15}$$

6.5.3.4 Experiments: invariant kriging in the eigenbasis

For metamodeling purposes, we define the following objective function for the designs of Example 6.3:

$$f(\mathbf{x}) = \pi x_3^2 + \pi x_6^2 + \pi x_9^2 - d(\mathbf{x})$$

where $d(\mathbf{x}) = \min_{\tau\in\boldsymbol{\tau}} \|(1,1)^\top - ((\tau\circ\mathbf{x})_1, (\tau\circ\mathbf{x})_2)^\top\|_2 + \|(4,2)^\top - ((\tau\circ\mathbf{x})_4, (\tau\circ\mathbf{x})_5)^\top\|_2 + \|(5,5)^\top - ((\tau\circ\mathbf{x})_7, (\tau\circ\mathbf{x})_8)^\top\|_2$ is the summed distance between the points (1,1), (4,2), (5,5) and the centers of the three circles, regardless of the order in which they are defined in $\mathbf{x}$ (min over all permutations).

The prediction capabilities of 4 GPs built over a DoE of $n = 25$ designs is investigated. 4 metamodeling versions are compared. In the first one, `GP(`X`)`, the metamodel is built over the space of CAD parameters. In `GP(`$\boldsymbol{\alpha}$`)`, the surrogate operates on the $\boldsymbol{\alpha}$ coordinates of those designs. `AugmGP(`$\boldsymbol{\alpha}$`)` considers not only the $n = 25$ $\boldsymbol{\alpha}$'s but the $\boldsymbol{\alpha}$ coordinates of all $\tau\circ\mathbf{x}$, $\tau \in \boldsymbol{\tau}$, permuted designs. There are $n \times e!$ such designs, increasing the size of the DoE to 150. Such an option is cumbersome as far as the shape contains more elements, or when more observations are available. Finally, `SymGP(`$\boldsymbol{\alpha}$`)` is the GP with invariant kernel $\tilde{k}_{\text{inv}}(\cdot,\cdot)$ (6.15) built using the DoE of $n = 25$ $\boldsymbol{\alpha}$'s.

Table 6.17 gives the accuracy of these models as measured by the R2 coefficient computed over a test set of 1000 designs. The average over 20 runs starting from different DoEs as well as the standard deviation are reported.

As remarked in the examples of Section 6.3.3, the metamodel in the $\mathcal{V}$ basis outperforms `GP(`X`)`. Logically, `AugmGP(`$\boldsymbol{\alpha}$`)` performs better than `GP(`$\boldsymbol{\alpha}$`)` because it contains more

Metamodel	R2
GP(X)	0.356 (0.115)
GP($\boldsymbol{\alpha}$)	0.553 (0.095)
AugmGP($\boldsymbol{\alpha}$)	0.709 (0.091)
SymGP($\boldsymbol{\alpha}$)	0.765 (0.059)

Table 6.17: Three circles example. R2 on a test set for the 4 GPs, averaged over 20 runs. Standard deviations are given in brackets.

designs. SymGP($\boldsymbol{\alpha}$) has the best accuracy. It outperforms GP($\boldsymbol{\alpha}$) and even AugmGP($\boldsymbol{\alpha}$) whose DoE has been enriched by the permuted designs (with 6 times more observations). The standard deviation of the R2 is also smaller with this GP. Last, the kriging variance of this GP is also reduced in comparison with GP($\boldsymbol{\alpha}$) which is an additional advantage when turning to optimization with the EI as in Section 6.4.

6.6 Conclusions

In this chapter a new methodology to apply Bayesian optimization techniques to parametric shapes and other problems where a pre-existing set of relevant points and a fast auxiliary mapping exist has been proposed. Instead of working directly with the CAD parameters, which are too numerous for an efficient optimization and may not be the best representation of the underlying shape, we unveil the lower dimensional manifold of shapes through the auxiliary mapping and PCA. The dimensions of this manifold that contribute the most to the variation of the output are identified through an L^1 penalized likelihood and then used for building an additive Gaussian Process with a zonal anisotropy on the selected variables and isotropy on the other variables. This GP is then utilized for Bayesian optimization.

The construction of the reduced space of variables opens the way to several strategies for the maximization of the acquisition criterion, in particular the restriction or not to the manifold and the replication. The different variants for the construction of the surrogate model and for the EI maximization have been compared on 7 examples, 6 of them being analytical and easily reproducible, the last one being a realistic airfoil design.

Even though specific variants are more or less adapted to features of specific test problems, the supervised dimension reduction approach and the construction of an additive GP between active and inactive components have given the most reliable results.

Regarding the EI maximization our experiments highlight the efficiency of the random embedding in the space of inactive variables in addition to the detailed optimization of the active variables. It is a trade-off between optimizing the active variables only, and optimizing all variables. Benefits have been observed for not restricting this inner maximization to the current approximation of $\mathcal{A}$ as well as for the virtual replication of points outside $\mathcal{A}$ when $\boldsymbol{\alpha} \notin \mathcal{A}$ is promoted by the EI.

An extension of the approach in the case of shapes made of multiple elements has also been proposed. It exploits the presence of symmetries in Φ by invariant kriging to the propagated symmetries in $\mathcal{V}$.

Instead of improving the modeling and optimization of variables that are considered as active when analyzing the whole design space, a possible extension could be to find and to prioritize the eigenshapes which impact the output in (Pareto) optimal regions of X (Spagnol et al., 2019).

Chapter 7

Conclusions

Contents

7.1 Summary of contributions

In this thesis, motivated by engineering applications, we have focused on the multi-objective optimization of expensive black-box functions via Bayesian algorithms (Emmerich et al., 2006; Jones et al., 1998).

First, in Chapter 3, a multi-objective benchmark problem called MetaNACA has been created. It has a variable number of CAD parameters ($d = 3, 8, 22$) and a flexible number of physical objectives ($m = 2, 3, 4$). It was built applying surrogate modeling techniques to real-world aerodynamic data. Since this test case belongs to the class of problems we aim at solving, it was extensively yet not uniquely used to benchmark existing algorithms as well as for comparing different methods developed throughout this thesis.

In Chapter 4, a new Bayesian multi-objective optimization algorithm was developed. It was designed for coping with expensive problems in which optimizations are budgeted. Under the restriction of few function evaluations, it was evidenced that even though they approach the Pareto front quicker than other algorithms, Bayesian methods are not always able to produce a high-quality approximation of the entire front. The issue of a non-converged Pareto front gets worse when augmenting the number of objectives because of the growth of the Pareto set size. Contrarily to classical Bayesian approaches which attempt to uncover the whole Pareto front regardless of the optimization resources, the C-EHI and R-EHI developed in Chapter 4 address the multi-objective optimization in steps. First, a part of the front is assessed as a priority area. The initial objective of the search is to converge towards the Pareto front in this region. The preferred part of the objective space is asked to the decision maker in R-EHI while it is implicitly defined as the center of the Pareto in the absence of preferences in C-EHI. The center of the Pareto front is a contribution of this thesis. It is defined in Section 4.4.2, together with properties and estimation methods. It corresponds to an equilibrium among the objectives and is therefore a particularly appealing solution to a multi-objective problem. In both cases, a simple modification to the EHI acquisition function (Emmerich et al., 2006) enables the targeting of the preferred part of the front. The mEI criterion, a simplified EHI, was proposed to guide both R-EHI and C-EHI algorithms in their first phase. A local convergence criterion to the preferred part of the front was devised to trigger C-EHI/R-EHI's second phase. It relies on a progress uncertainty measured through Gaussian Process simulations. In this second step, the target becomes a wider part of the Pareto front which can be accurately unveiled during the remaining function evaluations. The width of the attempted enlargement is determined by anticipating the behavior of the algorithm for the remaining iterations and by forecasting the uncertainty that would remain at the end of the search.

The C-EHI/R-EHI algorithms are extended to enhance their efficiency as well as their applicability to other problems. Batch criteria (Schonlau, 1997) are proposed and studied in Chapter 5 for both phases of the algorithm. This permits the evaluations of the objective functions in parallel if several computers or nodes of a cluster are available. At the same wall-clock time (which usually determines the allowed budget), a larger number of designs are evaluated. C-EHI and R-EHI's mechanisms are also adapted in Chapter 5 to comply with constraints (Feliot, 2017; Schonlau, 1997).

Motivated by high-dimensional parametric shape optimization problems, Chapter 6 proposes a way to address the curse of dimensionality (Bellman, 1961) which comes from the large number of variables. Instead of considering the CAD parameters which are numerous and may not express the object induced by the parameters (here a shape) in a suitable manner, designs are considered in a shape basis. The latter is constructed through the Principal Component Analysis of a database of discretized shapes. This basis comes with appealing properties: the axes have decreasing (geometric) importance and describe the shapes globally, as opposed to the CAD parameters of heterogeneous nature and which mostly correspond to local refinements. Beyond metamodeling in the basis of the first (hence most important) eigenvectors, a regularized likelihood indicates the directions that impact the objective function the most. An additive GP (Durrande et al., 2012) is then built. It is anisotropic over the reduced space of active components and coarser, isotropic, over the space of inactive ones. In this way, the GP also accounts for the non-selected shape vectors since they contribute a little to the output's variation too. By prioritizing the few more important parameters while considering all vectors of the shape basis, the obtained metamodel better deals with the curse of dimensionality and shows an increased accuracy. The input space is less uncertain too, which makes the subsequent Bayesian optimization more efficient. The optimization is conducted in reduced dimension by prioritizing the active components while the remaining eigenshapes are coarsely optimized through an embedding strategy (Binois et al., 2015b; Wang et al., 2013).

7.2 Possible improvements and perspectives

7.2.1 Objective space dimension reduction

This thesis deals with problems having potentially more than 2 or 3 objectives. In Chapter 3, it was shown that the quality of solutions returned by a multi-objective optimization on four objectives restricted to two objectives was degraded in comparison with the bi-objective optimization. This opens the question of the necessity of all objectives (Brockhoff and Zitzler, 2006a,b; López Jaimes et al., 2008). Before running an m-objective optimization, it may be worth checking whether some objectives can be neglected without a drastic change of the Pareto front structure (Corne and Knowles, 2007). In applications, some objectives may correspond to the same physical phenomenon observed at different operating conditions, as the example of the MetaNACA, whose lift and drag are computed at an angle of attack $\alpha_I = 0°$ or $\alpha_I = 8°$. The independence between such objectives is questionable and a correlation structure between some objectives could be exploited via co-kriging for instance (Cressie, 1992; Forrester and Keane, 2009; Fricker et al., 2013; Shah and Ghahramani, 2016; Svenson, 2011). Dimension reduction carried out via PCA in the input space in Chapter 6 could also be a way to aggregate some functions linearly (Deb and Saxena, 2006). The quadratic functions described in the Appendix D may help comparing alternatives as two spheres with similar centers are correlated objectives.

7.2.2 Output-driven shape basis construction

In Chapter 6, we have built the $\mathcal{V}$ basis through the non-supervised learning of a shape database. The axes of $\mathcal{V}$ that contribute the most to the output's fluctuation have then been detected through a regularized likelihood maximization and have been emphasized within the additive GP. In the spirit of Partial Least Squares (PLS, Frank and Friedman, 1993) or Sliced Inverse Regression (SIR, Li, 1991), a promising alternative would be the construction an orthonormal basis common to the m objective functions determined according to the output's fluctuation, in spite of the little observations ($n \ll D$). As seen through the examples of Section 6.3.3, the first modes are not necessarily the most relevant ones in the PCA approach, whereas they would in such methods.

In the same vein, the more flexible Student-t Processes (Shah et al., 2014) or the recent Deep Gaussian Processes (Bui et al., 2016; Damianou and Lawrence, 2013) may constitute a promising direction to evidence the link between the $\mathbf{x}$'s (or the $\phi(\mathbf{x})$'s) and the output.

Appendices

A MetaNACA benchmark experiments

In this section, different experiments carried out on the MetaNACA problems described in Chapter 3 are detailed. They aim at visualizing the effects of the input space and objective space dimension ($d = 3, 8, 22$ and $m = 2, 3, 4$), differences in the $budget = n + p$ allocation (initial DoE/infill criterion), and between the acquisition functions (EHI, EMI, SMS and SUR, see Section 2.4). For the sake of brevity, we only consider the (ratio of) hypervolume indicator (Definition 2.11) with reference point taken as the Nadir of the true Pareto front to compare the experiments because other indicators as the ε-indicator, the IGD, or attainment times led to similar conclusions drawn in Chapter 3.

The experiments are budgeted, i.e., a computational *budget* is defined and allocated between the initial DoE (n) and function evaluations driven by the infill criterion (p). $budget = 60$ in the $d = 3$ instance, 100 in the $d = 8$ problems and 200 for the MetaNACA with $d = 22$ parameters. Table A.1 gives the investigated sizes of initial DoE for each dimension, the number of calls to the acquisition function being $p = budget - n$ (including $p = 0$, i.e. all observations stem from a space filling DoE).

d	*budget*	n
3	60	10 30 50 60
8	100	20 40 60 80 100
22	200	50 100 150 200

Table A.1: *budget* distribution in the MetaNACA experiments.

A.1 Distribution of the computational budget

Here, we aim at comparing the optimizations regarding the *budget* allocation. We take the same infill criterion (EHI) and compare the hypervolume indicator at the end of the optimization, i.e. the MetaNACA has been called *budget* times with different $n + p$ repartitions. The runs are repeated ten times and presented in boxplots, for $m = 2, 4$, and $d = 3, 8, 22$ in Figure A.1. The size of the initial DoE (n) is reported in the x-axis.

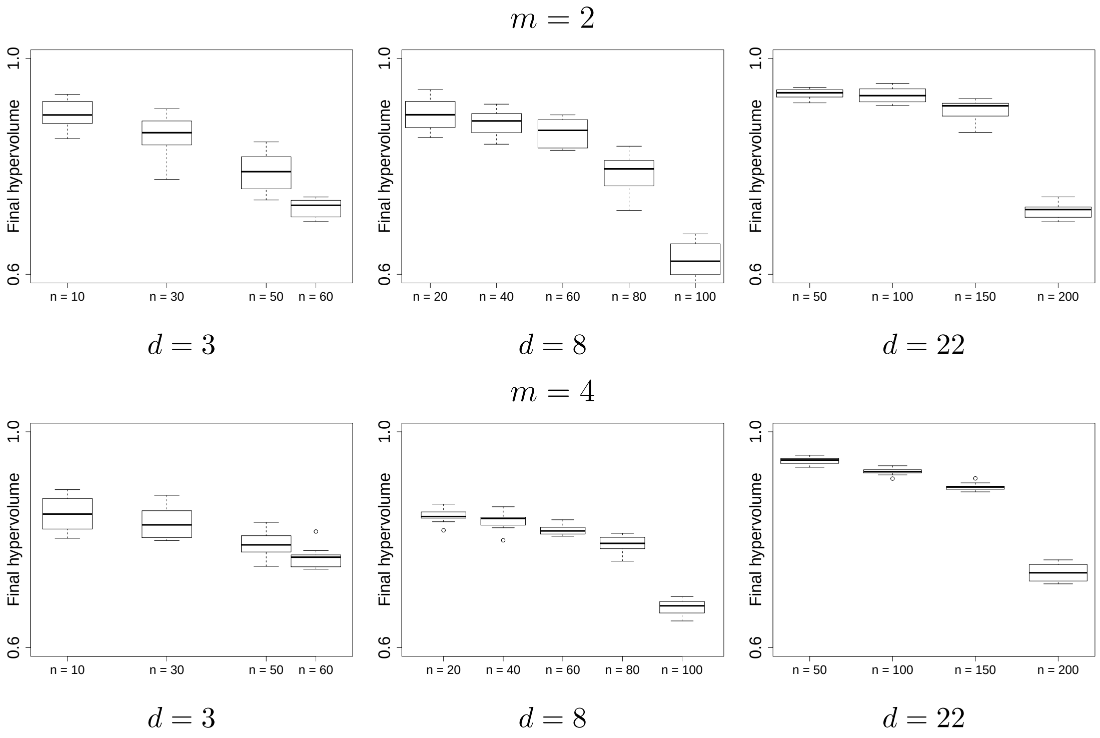


Figure A.1: Comparison of $budget = n + p$ allocations on different MetaNACA instances ($d = 3, 8, 22$, $m = 2, 4$).

For analyzing the convergence at different moments of the optimization in function of the *budget* repartition, the evolution of the hypervolume indicator, averaged over the 10 runs, is shown in Figure A.2 for different DoE sizes (see Table A.1), for the $m = 2, 4$, and $d = 3, 8, 22$ MetaNACA problems.

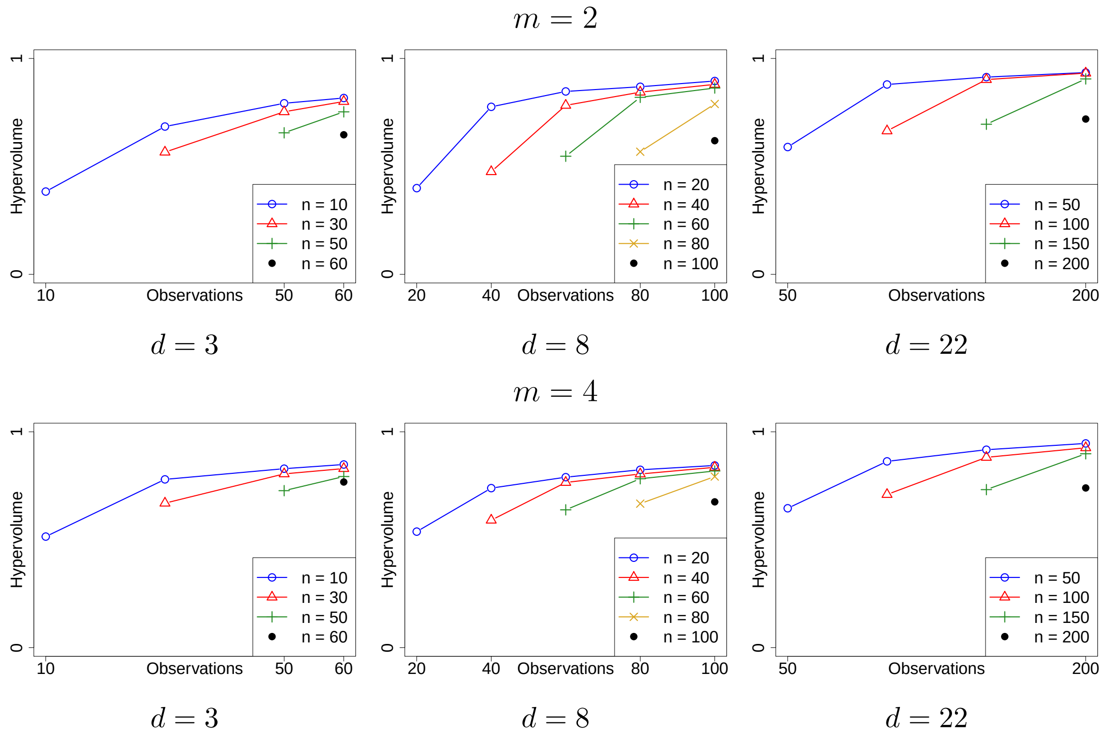


Figure A.2: Evolution of the hypervolume indicator against the number of observations, for varying sizes of DoE (n), on different MetaNACA instances ($d = 3, 8, 22$, $m = 2, 4$).

These results indicate that whatever the dimension or the number of objectives, on the MetaNACA test problems, it is worth using an as small as possible[1] initial DoE to enable the evaluation of a large number of designs chosen by the infill criterion.

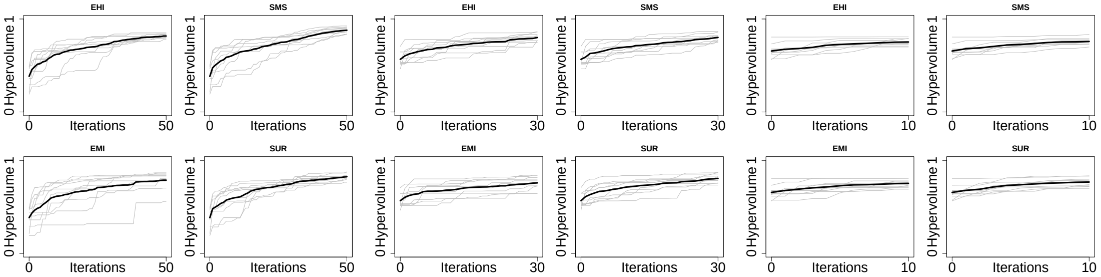
A.2 Evolution of the hypervolume indicator in the given budget

In this section, we compare the increase of the hypervolume indicator for all MetaNACA problems ($d = 3, 8, 22$, $m = 2, 3, 4$) and all infill criteria (EHI, EMI, SMS), during the search. Starting with n observations, the average hypervolume indicator (black curve) over the 10 runs (grey curves) is shown at any iteration $t \in [0, p]$ of the Bayesian optimization in Figures A.3 to A.5. As *budget* is fixed, p is not the same depending on n and the x-axes do not have the same range. Following the conclusion of Section A.1, the mean hypervolume indicator obtained by the different infill criteria in the setting with smallest initial DoE size (see Table A.1) are further compared at different times during the search. The evolution of the hypervolume is compared for different (d, m) problems separately.

[1] Here, we have considered $n \approx 2d + 4$ as smallest initial DoE.

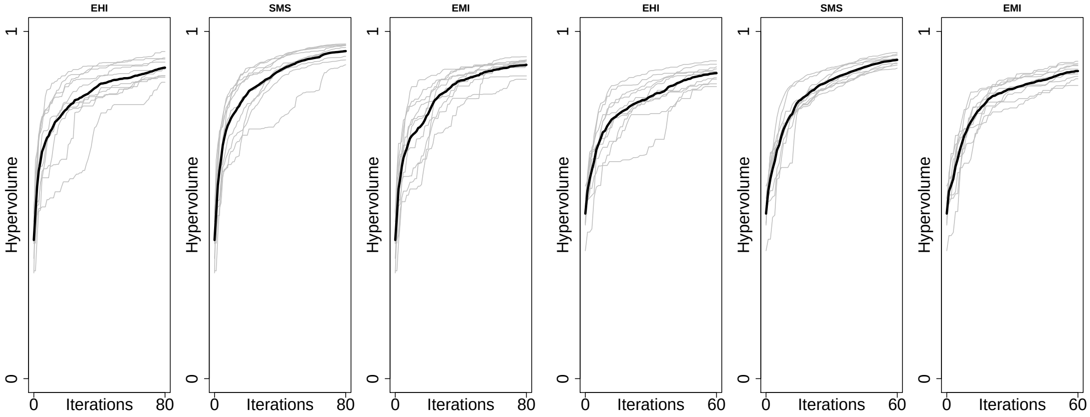
$m = 2$ **objectives**

$d = 3$

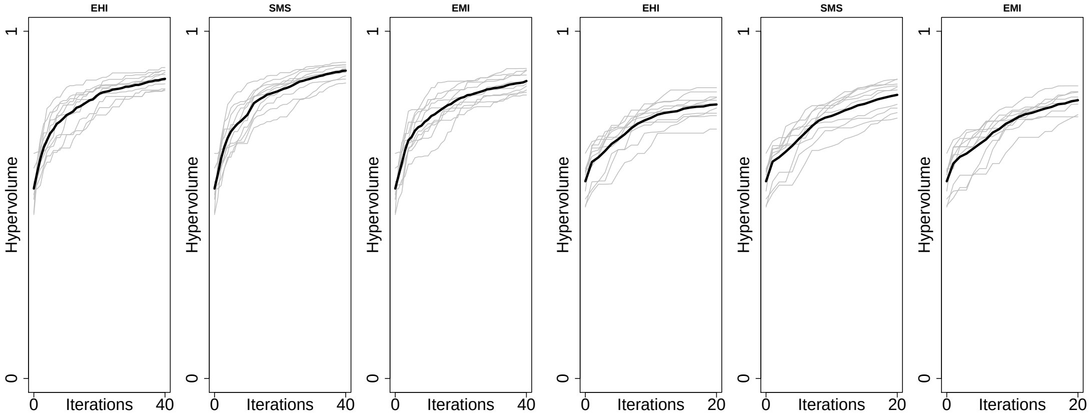


$n = 10$ $n = 30$ $n = 50$

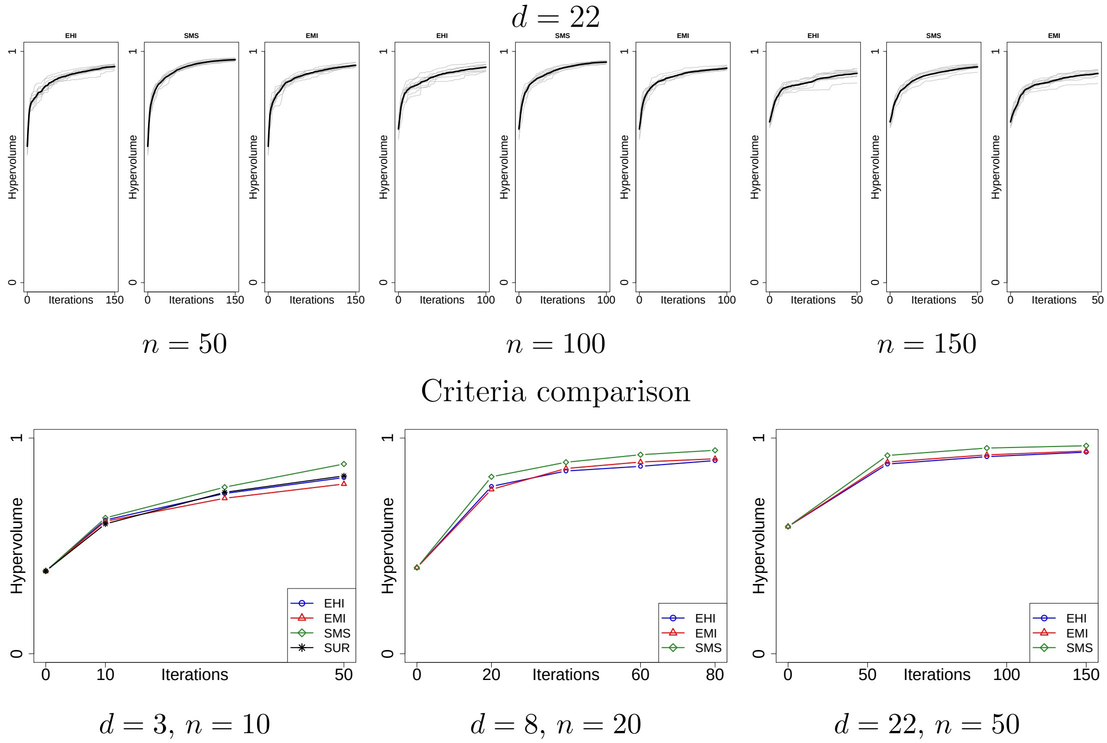
$d = 8$



$n = 20$ $n = 40$



$n = 60$ $n = 80$


$d = 22$

$n = 50$ $n = 100$ $n = 150$

Criteria comparison

$d = 3,\ n = 10$ $d = 8,\ n = 20$ $d = 22,\ n = 50$

Figure A.3: Evolution of the hypervolume indicator in the remaining budget, for different sizes of initial DoE (n), dimensions (d) and infill criteria.

$m = 3$ objectives

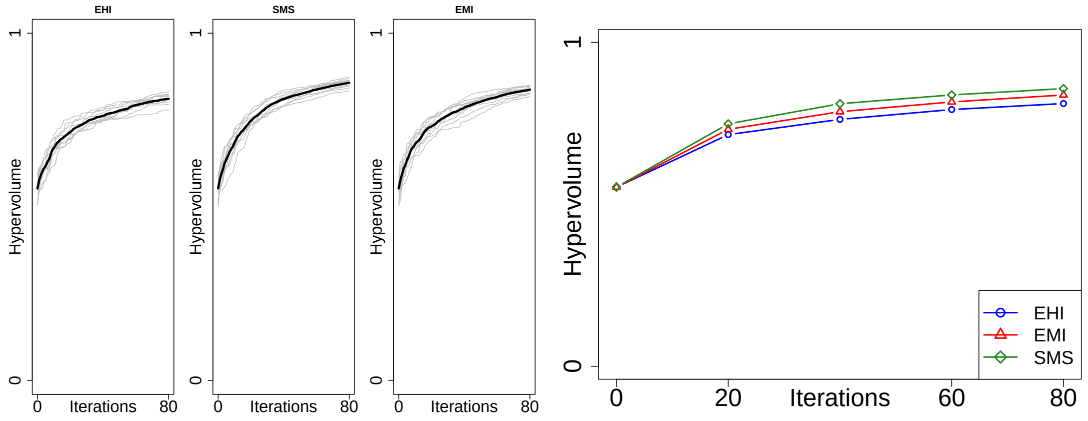
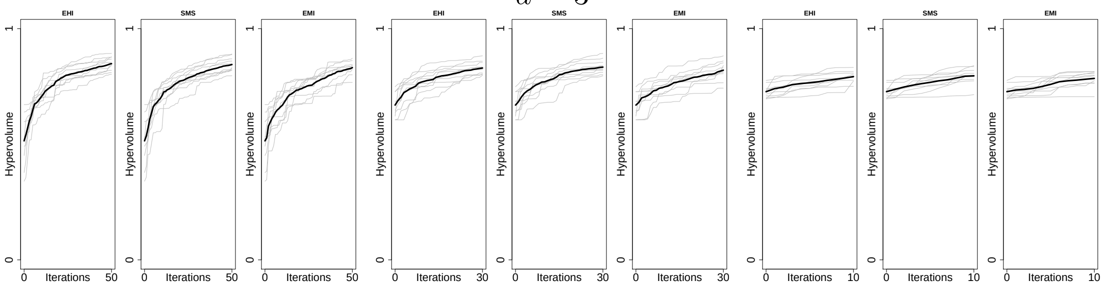


Figure A.4: Evolution of the hypervolume indicator in the remaining budget for the different infill criteria, for $d = 8$ and $n = 20$ ($budget = 100$).

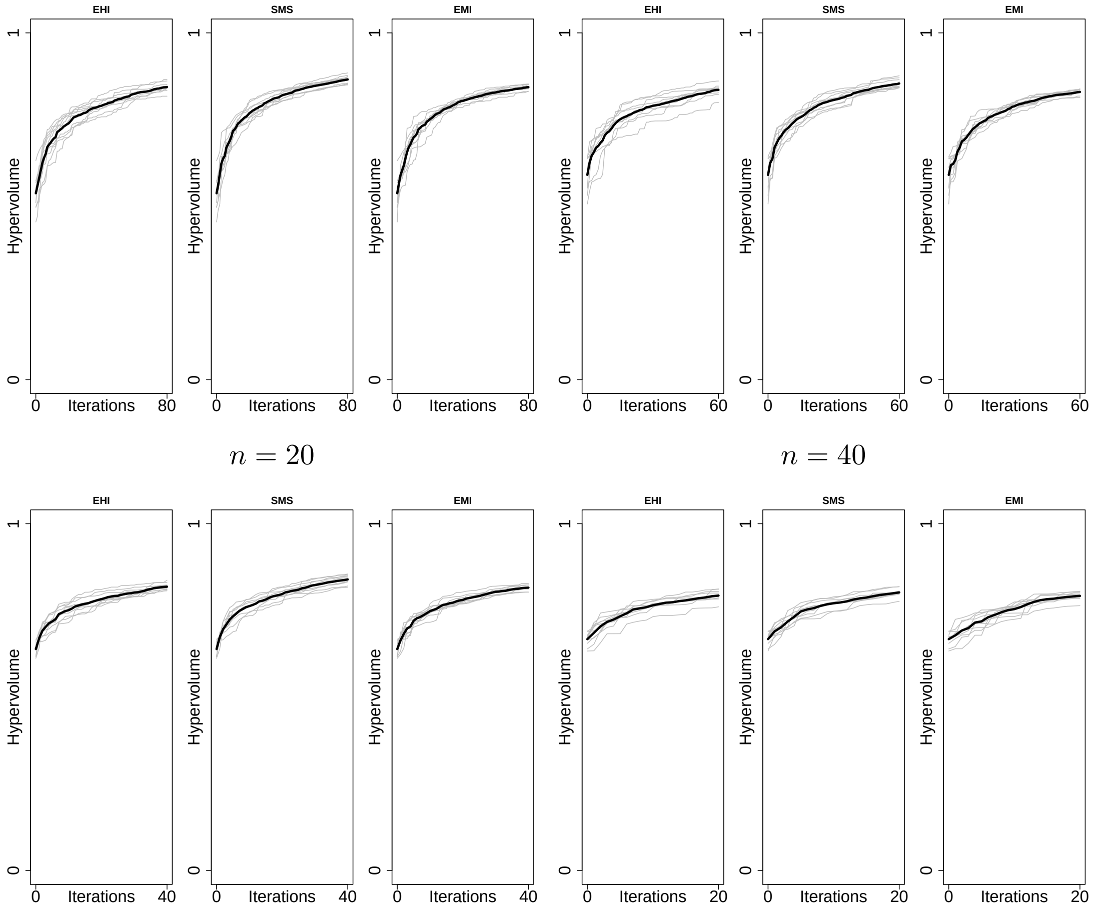
$m = 4$ objectives

$d = 3$



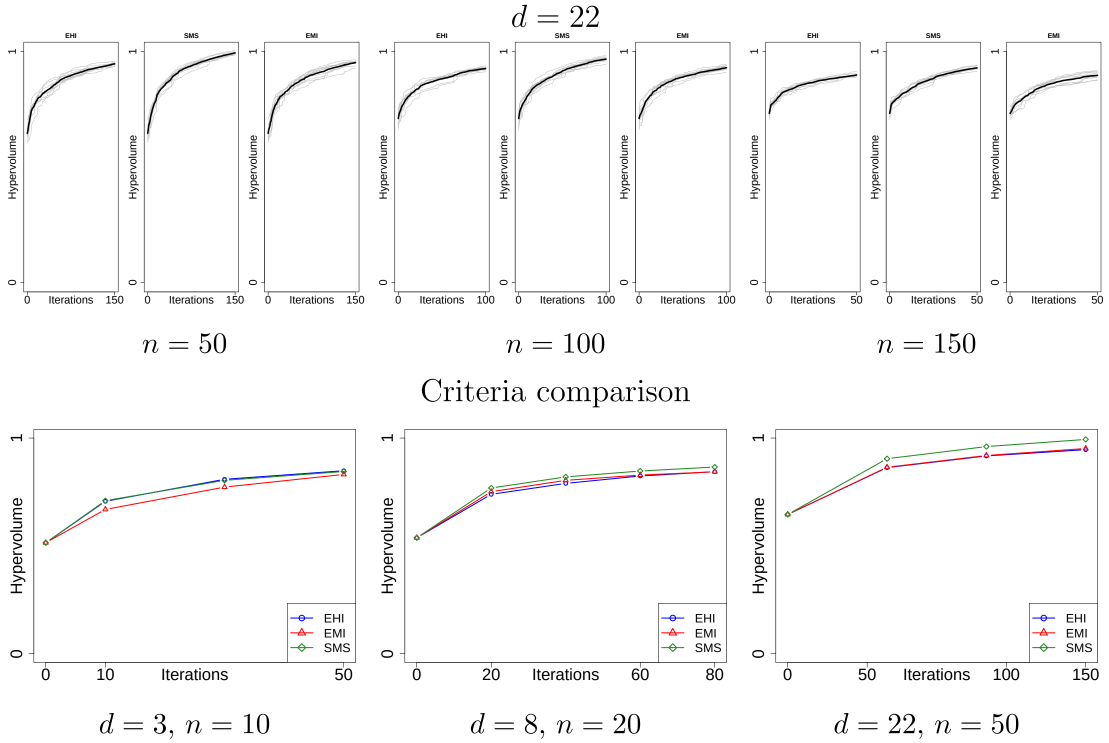
$n = 10$ $n = 30$ $n = 50$

$d = 8$



$n = 20$ $n = 40$

$n = 60$ $n = 80$



$d = 22$

$n = 50$ $n = 100$ $n = 150$

Criteria comparison

$d = 3,\ n = 10$ $d = 8,\ n = 20$ $d = 22,\ n = 50$

Figure A.5: Evolution of the hypervolume indicator in the remaining budget, for different sizes of initial DoE (n), dimensions (d) and infill criteria.

These figures show the same type of convergence is achieved, regardless of the dimension of the input space, the number of objectives, the size of the initial DoE or the criterion. A large increase in the hypervolume indicator is achieved from the first iterations on. The convergence then slows down. Yet, the curves do not seem to stagnate as the slope is not null at completion of the budget, meaning that the Pareto front could not be entirely unveiled. This appears to be even more true in the case $m = 4$, because of the larger size of the Pareto front.

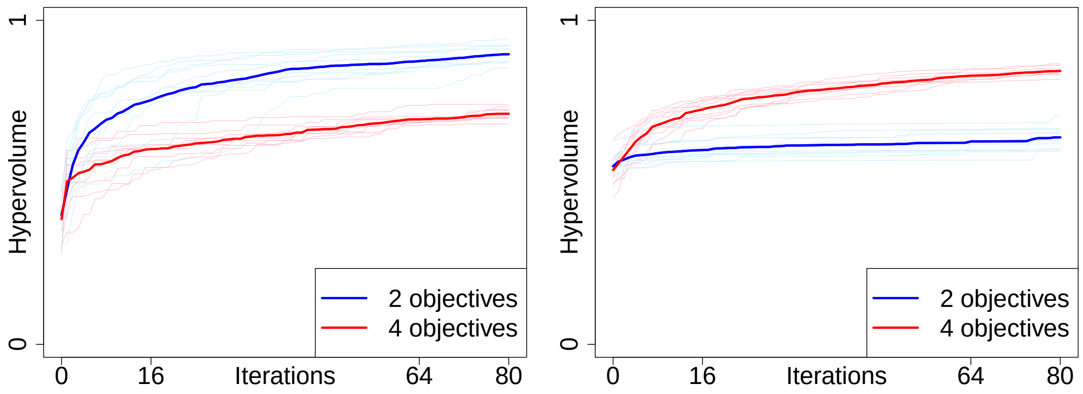
A.3 Optimizing too much or too less objectives

Since the 4 objectives of the MetaNACA are correlated (lift and drag at different angles of attack), an interesting question is the necessity of all objectives: could good results be achieved by optimizations focusing on two objectives only? Conversely, is there a price to pay by running an optimization on more objectives than are truly relevant?

To answer this question, we compare the outcomes of two optimizations. The first one is the optimization of the lift and the drag of the MetaNACA at $\alpha_I = 0°$ only, i.e. $m = 2$. The second one is the simultaneous optimization of all $m = 4$ objectives. In all cases, the MetaNACA in dimension 8 with the EHI infill criterion is employed, starting from a DoE with $n = 20$ observations.

Two situations are compared. The convergence of both optimizations restricted to the two first objective functions (the values observed by the optimizations with $m = 4$ objectives in the last two objectives are ignored) is shown in Figure A.6a. The hypervolume increase of optimizations with 4 objectives (red curve) rapidly slows down because critical parts in the 4 objective dimensional space were not necessarily associated with a large hypervolume contribution in the two first objectives. There is a price to pay by considering too much objectives.

In Figure A.6b, the opposite situation is considered. The goal is to optimize all 4 objectives, and standard optimizations considering all objectives are compared with optimizations only concentrated on the two first objectives, whose values in $f_3(\cdot)$ and in $f_4(\cdot)$ are computed for the occasion. It is seen that the latter (blue curve) almost does not increase the hypervolume indicator in comparison with the red curve. Indeed, designs chosen to achieve an improvement in the bi-objectives have been evaluated. These do not bring as much improvement to the 4-objectives problem as optimizations intended for this purpose. Despite the correlations in the objectives, an incomplete problem formulation leads to a weaker progress in the original problem.



(a) $m = 2$ objectives optimization.

(b) $m = 4$ objectives optimizations.

Figure A.6: Evolution of the hypervolume indicator in an m-objective problem using an m or an m'-objective infill criterion.

B Proofs of propositions related to the center

Proofs relative to the sensitivity of the center of the Pareto front to affine scalings of the objective space, or to the stability of **C** (Section 4.4.2.2) are provided in this section.

B.1 Center invariance to linear scaling, intersection case

Proof of Proposition 4.2.

Proof. First, it is clear that if $\mathcal{P_Y}$ intersects $\mathcal{L}$, the intersection is unique. Indeed, as in non degenerated cases $\mathbf{I} \prec \mathbf{N}$, $t\mathbf{I} + (1-t)\mathbf{N} \prec t'\mathbf{I} + (1-t')\mathbf{N} \Leftrightarrow t > t'$. Two points on $\mathcal{L}$ are different as long as $t \neq t'$. $\mathcal{P_Y}$ being only composed of non-dominated points it is impossible to find two different points $t\mathbf{I} + (1-t)\mathbf{N}$ and $t'\mathbf{I} + (1-t')\mathbf{N}$ that belong simultaneously to $\mathcal{P_Y}$. Obviously, as it lies on $\mathcal{L}$, $\nexists \mathbf{y} \in \mathcal{P_Y}$ that is closer to it.

Let $\mathbf{C}$ be this intersection. S being a linear scaling, it can be expressed in the form $S(\mathbf{y}) = \mathbf{Ay} + \mathbf{b}$ with $\mathbf{A}$ an $m \times m$ diagonal matrix with entries $a_i > 0$, and $\mathbf{b} \in \mathbb{R}^m$. Applying this scaling to the objective space modifies $\mathbf{C}$ to $\mathbf{C}' = \mathbf{AC}+\mathbf{b}$, $\mathbf{I}$ to $\mathbf{I}' = \mathbf{AI}+\mathbf{b}$ and $\mathbf{N}$ to $\mathbf{N}' = \mathbf{AN} + \mathbf{b}$. Because the scaling preserves orderings of the objectives, $\mathbf{C}'$ remains non-dominated, and $\mathbf{I}'$ and $\mathbf{N}'$ remain the Ideal point and the Nadir point of $\mathcal{P_Y}$ in the scaled objective space. As $\mathbf{C}$ belongs to $\mathcal{L}$ it writes $t\mathbf{I} + (1-t)\mathbf{N}$ for one $t \in [0,1]$, and therefore

$$\begin{aligned}\mathbf{C}' &= \mathbf{A}(t\mathbf{I} + (1-t)\mathbf{N}) + \mathbf{b}\\ &= t\mathbf{AI} + (1-t)\mathbf{AN} + \mathbf{b}\\ &= t(\mathbf{AI} + \mathbf{b}) + (1-t)(\mathbf{AN} + \mathbf{b})\\ &= t\mathbf{I}' + (1-t)\mathbf{N}'\end{aligned}$$

$\mathbf{C}'$ is thus the unique point belonging to both the Pareto front and to the Ideal-Nadir line in the transformed objective space: it is the center in the scaled objective space. □

B.2 Center invariance to linear scaling, 2D case

Proof of Proposition 4.3.

Proof. Let A be the area of the $\mathbf{IyN}$ triangle and A' be the area of $\mathbf{Iy'N}$. Applying a linear scaling $S(\mathbf{y}) = \mathbf{Ay} + \mathbf{b}$ with $\mathbf{A} = \begin{pmatrix}\alpha & 0\\ 0 & \beta\end{pmatrix}$, $\alpha, \beta > 0$ to Y will modify the areas A and A' by the same factor $\alpha\beta$. Thus, $\|S(\mathbf{y}) - \Pi_{S(\mathcal{L})}(S(\mathbf{y}))\| \leq \|S(\mathbf{y}') - \Pi_{S(\mathcal{L})}(S(\mathbf{y}'))\|$ still holds: in the transformed subspace, $\mathbf{y}$ remains closer to $\mathcal{L}$ than $\mathbf{y}'$. □

B.3 Example with $m > 2$ where the center is modified by a linear scaling of the objectives

Let us consider the case of a Pareto front composed of the five following non-dominated points (in rows) in a three-dimensional objective space: $\mathbf{P} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 0.5 & 0.5 & 0.6\\ 0.5 & 0.55 & 0.5\end{bmatrix}$, $\mathcal{P_Y} = \{\mathbf{P}^{(1)}, \dots, \mathbf{P}^{(5)}\}$. The Ideal point is $\mathbf{I} = (0,0,0)^\top$ and the Nadir point $\mathbf{N} = (1,1,1)^\top$. The squared Euclidean distance to $\mathcal{L}$ of these 5 points equals respectively 2/3, 2/3, 2/3, 0.02/3 and 0.005/3, hence $\mathbf{P}^{(5)} = (0.5, 0.55, 0.5)^\top$ is the closest point to $\mathcal{L}$. Let us now apply a linear scaling $S(\mathbf{y}) = \mathbf{A}\mathbf{y}$ with $\mathbf{A} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. In the modified objective space, we now have $\widetilde{\mathbf{P}} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \\ 1.5 & 1.5 & 0.6 \\ 1.5 & 1.65 & 0.5 \end{bmatrix}$, $\widetilde{\mathbf{I}} = (0,0,0)^\top$ and $\widetilde{\mathbf{N}} = (3,3,1)^\top$. The squared distances to $\widetilde{\mathcal{L}}$ after scaling are now respectively 1710/361, 1710/361, 342/361, 3.42/361, 4.275/361. After scaling, the fourth point becomes the closest to the line. As the projection of the latter on $\widetilde{\mathcal{L}}$ is different from the projection of the fifth point, the center of the Pareto front will change after this scaling.

B.4 Stability with respect to I or N's variation

Before proving Proposition 4.4, Lemma B.1 gives a condition on the normal vector to the Pareto front that will be needed to prove the proposition.

Lemma B.1. *Let $\mathbf{y}^* \in \mathbb{R}^m$ be a Pareto optimal solution, and the Pareto front be continuous and differentiable at $\mathbf{y}^*$ with $\mathbf{d} \in \mathbb{R}^m$ the normal vector to the Pareto front at $\mathbf{y}^*$. Then all components of $\mathbf{d}$ have the same sign.*

Proof. Because of the differentiability assumption at $\mathbf{y}^*$ and the definition of Pareto dominance, $\mathbf{d}$ cannot have null components. Suppose that some components in $\mathbf{d}$ have opposite signs, $\mathbf{d}^+$ corresponding to positive ones and $\mathbf{d}^-$ to negatives ones, $\mathbf{d} = [\mathbf{d}^+, \mathbf{d}^-]$. Let ε^+ and ε^- be two small positive scalars such that $\frac{\varepsilon^+}{\varepsilon^-} = \frac{\sum_{i:d_i<0} {d_i}^2}{\sum_{i:d_i>0} {d_i}^2}$. Then, $\mathbf{f} = \mathbf{y}^* + \begin{pmatrix} -\varepsilon^+\mathbf{d}^+ \\ \varepsilon^-\mathbf{d}^- \end{pmatrix}$ dominates $\mathbf{y}^*$ and belongs to the local first order approximation to $\mathcal{P}_\mathcal{Y}$ since $\mathbf{d}^\top(\mathbf{f} - \mathbf{C}) = 0$, which is a contradiction as $\mathbf{y}^*$ is Pareto optimal. □

We can now prove Proposition 4.4.

Proof. If $\mathcal{P}_\mathcal{Y}$ is locally continuous and $m-1$ dimensional, $\mathbf{C}$ is the intersection between $\mathcal{L}$ and $\mathcal{P}_\mathcal{Y}$. For simplicity, the Pareto front is scaled between 0 and 1, that is, $\mathbf{I} = \mathbf{0}_m$ and $\mathbf{N} = \mathbf{1}_m$. Proposition 4.2 ensures that the center is not modified by such a scaling. The tangent hyperplane to $\mathcal{P}_\mathcal{Y}$ at $\mathbf{C}$ writes $\mathbf{d}^\top\mathbf{f} + e = 0$ where $\mathbf{d} \in \mathbb{R}^m$, the normal vector to the tangent hyperplane, and $e \in \mathbb{R}$ depend on $\mathcal{P}_\mathcal{Y}$ and are supposed to be known. Lemma B.1 ensures that d_i, $i = 1, \dots, m$ have the same sign, that we choose positive. $\mathbf{C}$ satisfies both $\mathbf{d}^\top\mathbf{C} = -e$ and $\mathbf{C} = (1-\alpha_C)\mathbf{I} + \alpha_C\mathbf{N} = \alpha_C\mathbf{1}_m$ for some $\alpha_C \in]0,1[$. Hence,

$$\mathbf{C} = \frac{-e}{\mathbf{d}^\top\mathbf{N}}\mathbf{N},\ C_i = \frac{-e}{\mathbf{d}^\top\mathbf{N}}N_i$$

$\forall j = 1, \dots, m, j \neq i$,

$$\frac{\partial C_i}{\partial N_j} = \frac{eN_i d_j}{(\mathbf{d}^\top\mathbf{N})^2} = \frac{-d_j}{\sum_k d_k N_k}C_i = \frac{-d_j}{\sum_k d_k}C_i$$

For $i = j$,

$$\frac{\partial C_i}{\partial N_i} = \frac{-e\mathbf{d}^\top \mathbf{N} + eN_i d_i}{(\mathbf{d}^\top \mathbf{N})^2} = \frac{C_i}{N_i} - \frac{C_i}{\sum_k d_k N_k} = C_i \left(1 - \frac{d_i}{\sum_k d_k}\right)$$

$C_i = \alpha_C \in]0,1[\ \forall i = 1,\dots,m$ and as the d_i's share the same sign, $|d_i| \leq |\sum_k d_k|$. Therefore, $|\frac{\partial C_i}{\partial N_i}| < 1$ and $|\frac{\partial C_i}{\partial N_j}| < 1$. Consider now that $\mathbf{N}$ is modified into $\mathbf{N} + \Delta\mathbf{N}$, which changes the center to $\mathbf{C} + \Delta\mathbf{C}$. One has $\Delta\mathbf{C} = \nabla\mathbf{C} \cdot \Delta\mathbf{N}$ where $\nabla\mathbf{C}$ is the $m \times m$ matrix with entries $\frac{\partial C_i}{\partial N_j}$. Rearranging the terms of the derivatives into matrix form yields

$$\nabla\mathbf{C} = \alpha_C \left[I_m - \underbrace{\frac{1}{\sum_k d_k} \begin{pmatrix} d_1 & d_2 & \cdots & d_m \\ \vdots & \vdots & \vdots & \vdots \\ d_1 & d_2 & \cdots & d_m \end{pmatrix}}_{D} \right]$$

where I_m stands for the identity matrix here. D is a rank 1 matrix with positive entries whose rows sum to 1, and has eigenvalues 0 and 1 with respective multiplicity $m-1$ and 1. Consequently, $\nabla\mathbf{C}$'s largest eigenvalue is $\alpha_C \in]0,1[$. Finally, $\|\Delta\mathbf{C}\|_2 \leq \|\nabla\mathbf{C}\|_2 \|\Delta\mathbf{N}\|_2 \leq \|\Delta\mathbf{N}\|_2$. By symmetry, the proposition extends to the sensitivity of the center to the Ideal point, $|\frac{\partial C_i}{\partial I_j}| < 1, \ i,j = 1,\dots,m$ and $\|\Delta\mathbf{C}\|_2 < \|\Delta\mathbf{I}\|_2$. □

C Nadir point estimation using Gaussian Processes

In the field of EMOA's, estimation procedures for extreme points, thus components of $\mathbf{N}$, have been proposed (Bechikh et al., 2010; Deb et al., 2010). In the Gaussian Process framework, we look for $\mathbf{x}$'s that are likely to be extreme design points (Definition 2.10). Estimating the Nadir point through surrogates is a difficult task. When $m > 2$, the Nadir components come from extreme points that are not necessarily optimal in a single objective (cf. Definition 2.8). A straightforward estimation of the Nadir involves the knowledge of the whole Pareto front, as each component j of the Nadir point is dependent on the j-th objective function, but also on all other functions through the component-wise non-domination property of $\mathbf{N}$. However, the C-EHI algorithm only targets central solutions. With this algorithm, the GPs may not be accurate at non central locations of $\mathcal{P_Y}$. Using simulated values of the GPs instead of the kriging prediction should nonetheless reduce the impact of a potential inaccuracy as the latter is implicitly considered. Applying a step of mono-objective $f_j(\cdot)$ minimization (e.g. using EGO) might diminish this difficulty (at least for the $\mathbf{I}$ estimation), at the expense of m costly evaluations of the computer code.

We now explain the proposed estimation approach. Extreme points $\boldsymbol{\nu}^j \in Y$, $j = 1,\dots,m$, are responsible for the j-th component of $\mathbf{N}$, $\nu^j_j = N_j$ (see Definition 2.10). They are both large in the j-th objective (largest y_j value inside $\mathcal{P_Y}$) and not dominated

(ND). To simulate possible values of extreme points, we are thus interested in $\mathbf{x}$'s with a high probability $\mathbb{P}(Y_j(\mathbf{x}) > a_j, \mathbf{Y}(\mathbf{x}) \text{ ND})$, for $j = 1, \dots, m$. A typical choice for a_j is the j-th component of the Nadir of the current Pareto front approximation, $\overline{N}_j = \overline{\nu}^j_j$. Non-domination refers to the current Pareto front approximation $\widehat{\mathcal{P}_\mathcal{Y}}$. These events are not independent since $\mathbf{Y}(\mathbf{x})$ contains $Y_j(\mathbf{x})$. However, by conditioning on $\{Y_j(\mathbf{x}) > \overline{\nu}^j_j\}$, $\mathbb{P}(Y_j(\mathbf{x}) > \overline{\nu}^j_j, \mathbf{Y}(\mathbf{x}) \text{ ND}) = \mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND} | Y_j(\mathbf{x}) > \overline{\nu}^j_j) \times \mathbb{P}(Y_j(\mathbf{x}) > \overline{\nu}^j_j)$. The first part can be further simplified: to be non-dominated by $\widehat{\mathcal{P}_\mathcal{Y}}$, a vector $\mathbf{z} \in \mathbb{R}^m$ with $z_j > \max\limits_{\mathbf{y} \in \widehat{\mathcal{P}_\mathcal{Y}}} y_j$ has to be non-dominated by $\widehat{\mathcal{P}_\mathcal{Y}}$ with regard to objectives $1, \dots, j-1, j+1, \dots, m$. Hence, $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND} | Y_j(\mathbf{x}) > \overline{\nu}^j_j) = \mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND}_{\setminus\{j\}})$ where $\text{ND}_{\setminus\{j\}}$ stands for non-domination omitting the objective j. Finally, the most promising candidates for generating extreme points of the Pareto front are those with large probability $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND}_{\setminus\{j\}}) \times \mathbb{P}(Y_j(\mathbf{x}) > \overline{\nu}^j_j)$.

Besides these candidates, a second scenario will lead to new extreme points. If $\mathbb{R}^m \ni \mathbf{z} \preceq \overline{\boldsymbol{\nu}}^j$ is obtained through simulations, $\overline{\boldsymbol{\nu}}^j$ will no longer belong to the simulated Pareto front. Consequently, the j-th component of the Nadir point of the simulated front will also be modified in that case. When $m = 2$, the new $\overline{\nu}^j_j$ will be z_j, but this does not necessarily hold in higher dimensions.

In short, two events will lead to new extreme points: dominating the j-th current extreme point, $\{\mathbf{Y}(\mathbf{x}) \preceq \overline{\boldsymbol{\nu}}^j\}$, or being both larger than it in the j-th objective and ND with respect to the approximation front in the remaining objectives, $\{Y_j(\mathbf{x}) > \overline{\nu}^j_j, \mathbf{Y}(\mathbf{x}) \text{ ND}_{\setminus\{j\}}\}$. The areas corresponding to these events are sketched with a 2D example in Figure C.7. Being disjoint, the probability of the union of these events equals the sum. In the end, for estimating the j extreme points and by extension $\mathbf{N}$, the most promising candidates are those maximizing

$$\mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND}_{\setminus\{j\}}) \times \mathbb{P}(Y_j(\mathbf{x}) > \overline{\nu}^j_j) + \mathbb{P}(\mathbf{Y}(\mathbf{x}) \preceq \overline{\boldsymbol{\nu}}^j), \tag{1}$$

for $j = 1, \dots, m$. $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND}_{\setminus\{j\}})$ is the probability of being non-dominated with respect to an $m-1$ dimensional front (which is smaller than the restriction of $\widehat{\mathcal{P}_\mathcal{Y}}$ to $\{1, \dots, m\} \setminus \{j\}$) and is the more computationally demanding term for a given $\mathbf{x}$. The other terms are univariate and product of univariate Gaussian CDF's, respectively.

In the particular case of two objectives, the union of these events reduces to dominating $\overline{\boldsymbol{\nu}^j}$ in all objectives but j, that is to say, in the other objective $\bar{j}$. This is equivalent to looking for candidates with lower $f_{\bar{j}}(\cdot)$, which has already been investigated when looking for candidates for estimating $I_{\bar{j}}$. Unfortunately, in a general m-dimensional case no simplification occurs. The set of candidates that are likely to dominate $\overline{\boldsymbol{\nu}}^j$ in all objectives but j is included but not equal to the set of candidates likely to maximize (1), whose probabilities are respectively $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \preceq_{\setminus\{j\}} \overline{\boldsymbol{\nu}^j})$ and $\mathbb{P}(\mathbf{Y}(\mathbf{x}) \text{ ND}_{\setminus\{j\}}) \times \mathbb{P}(Y_j(\mathbf{x}) > \overline{\nu}^j_j) + \mathbb{P}(\mathbf{Y}(\mathbf{x}) \preceq \overline{\boldsymbol{\nu}}^j)$, as the latter encompasses more cases for producing new extreme points when $m > 2$. It is indeed possible to construct $\mathbf{z} \in \mathbb{R}^m$ such that $z_j > \overline{\nu}^j_j$, $\mathbf{z}$ $\text{ND}_{\setminus\{j\}}$ and $\mathbf{z} \not\preceq_{\setminus\{j\}} \overline{\boldsymbol{\nu}}^j$. Such a $\mathbf{z}$ will become the j-th extreme point without dominating the previous j-th extreme point in objectives $\{1, \dots, m\} \setminus \{j\}$.

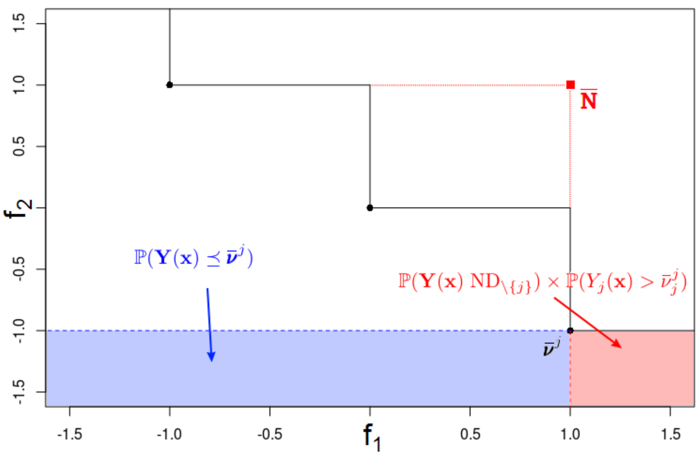


Figure C.7: Areas leading to a new first component ($j = 1$) of $\mathbf{N}$. A point in the red zone (larger than the first extreme point, $\bar{\boldsymbol{\nu}}^1$, in the first objective and non-dominated) or in the blue zone (dominating $\bar{\boldsymbol{\nu}}^1$) becomes the new (first) extreme point, and therefore induces a modification of $\mathbf{N}$.

D Quadratic objective functions

Despite their simplicity, quadratic functions are a powerful tool to study the behavior of (multi-objective) optimizers (Brockhoff et al., 2015; Igel et al., 2007; Toure et al., 2019). Here, we consider the m objective functions defined by

$$f_j(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{c}^j\|_2^2 \tag{2}$$

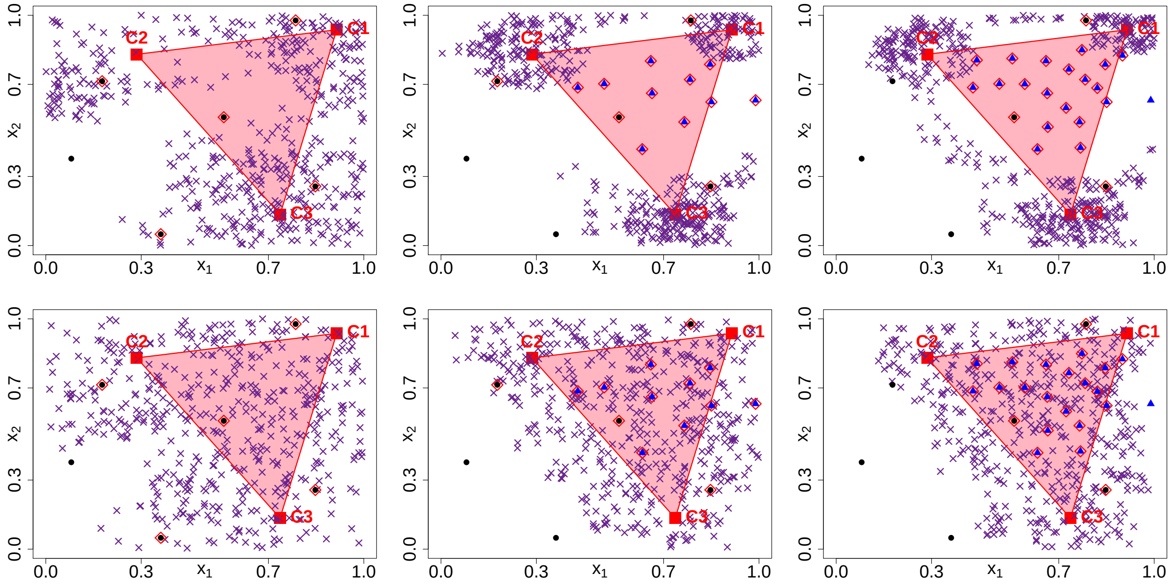
where $\mathbf{x} \in X = [0,1]^d$ and $\mathbf{c}^j \in X$ is the center of the j-th sphere. The multi-objective problem

$$\min_{\mathbf{x} \in X} \left(f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\right) \tag{3}$$

is considered. The necessary Pareto-optimal conditions ($\mathbf{x}^*$ Pareto optimal $\Rightarrow \exists \lambda_1, .., \lambda_m \geq 0 : \sum_{j=1}^m \lambda_j \nabla f_j(\mathbf{x}^*) = \mathbf{0}_m$, Miettinen, 1998; Toure et al., 2019) state that the Pareto set of (3) is the convex hull of $\{\mathbf{c}^1, \dots, \mathbf{c}^m\}$. Therefore, when $m > d$ and the centers are not aligned, the ratio of Pareto optimal solutions equals the volume of this hull.

In this section we illustrate the selection of designs in X where to perform the conditional GP simulations described in Chapter 4. We both resort to simulated fronts to estimate the Ideal and Nadir point of the Pareto front (used for estimating the $\mathcal{L}$ or $\mathcal{L}'$ line, on which $\mathbf{R}$ is driven towards the Pareto front, Section 4.4.2.3) and to check local convergence to the Pareto front (Section 4.5). In the first case, $\mathbf{x}$'s are selected proportionally to their probability of being an extreme design (i.e. an $\mathbf{x}$ whose image is non-dominated and has one coordinate of the Nadir point) or of being a minimum design (an $\mathbf{x}$ whose image is the minimum in one objective function), while in the second case they are chosen according to their non-domination probability as in Binois (2015).

In the optimization problem (3) with $m = 3$ objectives defined through three non-aligned circles and $d = 2$ dimensions, Figure D.8 shows which points in X are selected for both tasks (purple crosses), at different times of the optimization ($t = 0$, 10 or 20 iterations). Conditional GPs are simulated at these case-oriented locations of X.



$p = 0$ $p = 10$ $p = 20$

Figure D.8: $\mathbf{x}$'s where conditional GP simulations are performed (purple crosses) for estimating $\mathbf{I}$ and $\mathbf{N}$ (top row) and for being non-dominated (bottom row). Left to right: after 0, 10, or 20 additional function evaluations. The centers of the $f_j(\cdot)$'s are represented by red squares, black dots are the initial DoE points, and the blue triangles correspond to sequential infills. Non-dominated solutions are surrounded by a red diamond.

Remark D.1. *The necessity of all objectives, discussed in Section A.3 and in perspectives for future research (Section 7.2) might be investigated through these quadratic functions. For instance, in a two-dimensional input space, let $m > 3$ and $\mathbf{c}^1$, $\mathbf{c}^2$, $\mathbf{c}^3$ be three non-aligned points. Let the remaining centers $\mathbf{c}^j$, $j = 4, \ldots, m$ belong to the convex hull of $\{\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3\}$ as in Figure D.9. The $f_j(\cdot)$'s, $j = 4, \ldots, m$ do not modify the Pareto set nor the Pareto dominance relation in the objective space Y. However, they increase the dimension (in terms of objectives) of the problem. They can be disregarded and a criterion measuring the necessity of objective functions inside a multi-objective problem should indicate that they are redundant.*

Likewise, the correlation between objectives can be expressed by the proximity between $\mathbf{c}^j$'s, such as $\mathbf{c}^3$ and $\mathbf{c}^6$ in Figure D.9.

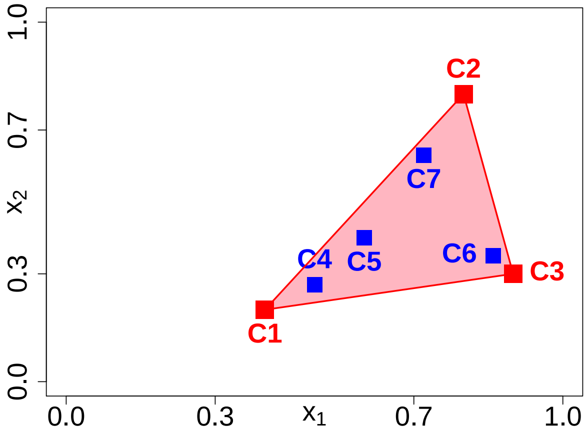


Figure D.9: Quadratic objective functions whose centers $\mathbf{c}$ are shown in the $X = [0, 1]^2$ space.

References

(2011). *2017-2025 Model Year Light-Duty Vehicle GHG Emissions and CAFE Standards: Supplemental.*

(2016). *Commission Regulation (EU) 2016/646 of 20 April 2016 amending Regulation (EC) No 692/2008 as regards emissions from light passenger and commercial vehicles (Euro 6).*

Allaire, G. (2005). *Analyse numérique et optimisation: une introduction à la modélisation mathématique et à la simulation numérique.* Editions Ecole Polytechnique.

Allard, D., Senoussi, R., and Porcu, E. (2016). Anisotropy models for spatial data. *Mathematical Geosciences*, 48(3):305–328.

Anderson Jr, J. D. (1984). *Fundamentals of aerodynamics.* Tata McGraw-Hill Education.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009a). Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 555–562. ACM.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009b). Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 563–570. ACM.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009c). Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102. ACM.

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

Auger, A. and Hansen, N. (2005). Performance evaluation of an advanced local search evolutionary algorithm. In *2005 IEEE congress on evolutionary computation*, volume 2, pages 1777–1784. IEEE.

Bachoc, F., Helbert, C., and Picheny, V. (2019). Gaussian process optimization with simulation failures.

Bachoc, F., Lagnoux, A., and Nguyen, T. M. N. (2017). Cross-validation estimation of covariance parameters under fixed-domain asymptotics. *Journal of Multivariate Analysis*, 160:42–67.

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

Basudhar, A., Dribusch, C., Lacaze, S., and Missoum, S. (2012). Constrained efficient global optimization with support vector machines. *Structural and Multidisciplinary Optimization*, 46(2):201–221.

Bautista, D. C. T. (2009). *A sequential design for approximating the Pareto front using the expected Pareto improvement function*. PhD thesis, The Ohio State University.

Bechikh, S., Kessentini, M., Said, L. B., and Ghédira, K. (2015). Preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art. In *Advances in Computers*, volume 98, pages 141–207. Elsevier.

Bechikh, S., Said, L. B., and Ghedira, K. (2010). Estimating Nadir point in multi-objective optimization using mobile reference points. In *Evolutionary computation (CEC), 2010 IEEE congress on*, pages 1–9. IEEE.

Bect, J., Bachoc, F., and Ginsbourger, D. (2016). A supermartingale approach to Gaussian process based sequential design of experiments. *arXiv preprint arXiv:1608.01118*.

Bect, J., Li, L., and Vazquez, E. (2017). Bayesian subset simulation. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):762–786.

Bellman, R. E. (1961). *Adaptive control processes: a guided tour*. Princeton university press.

Ben Salem, M., Bachoc, F., Roustant, O., Gamboa, F., and Tomaso, L. (2018). Sequential dimension reduction for learning features of expensive black-box functions.

Benassi, R., Bect, J., and Vazquez, E. (2011). Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion. In *International Conference on Learning and Intelligent Optimization*, pages 176–190. Springer.

Berkooz, G., Holmes, P., and Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575.

Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L., and Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082.

Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669.

Binois, M. (2015). *Uncertainty quantification on Pareto fronts and high-dimensional strategies in Bayesian optimization, with applications in multi-objective automotive design*. PhD thesis, École Nationale Supérieure des Mines de Saint-Etienne.

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

Binois, M., Ginsbourger, D., and Roustant, O. (2015b). A warped kernel improving robustness in Bayesian optimization via random embeddings. In *International Conference on Learning and Intelligent Optimization*, pages 281–286. Springer.

Binois, M., Ginsbourger, D., and Roustant, O. (2017). On the choice of the low-dimensional domain for global optimization via random embeddings. *arXiv preprint arXiv:1704.05318*.

Binois, M. and Picheny, V. (2015). GPareto: An R package for Gaussian-process based multi-objective optimization and analysis.

Binois, M., Picheny, V., Taillandier, P., and Habbal, A. (2019). The Kalai-Smorodinski solution for many-objective Bayesian optimization. *arXiv preprint arXiv:1902.06565*.

Binois, M., Rullière, D., and Roustant, O. (2015c). On the estimation of pareto fronts from the point of view of copula theory. *Information Sciences*, 324:270–285.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J. (2016). Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53(5):935–952.

Bowman, V. J. (1976). On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple criteria decision making*, pages 76–86. Springer.

Branke, J., Deb, K., Dierolf, H., and Osswald, M. (2004a). Finding knees in multi-objective optimization. In *International conference on parallel problem solving from nature*, pages 722–731. Springer.

Branke, J., Schmeck, H., Deb, K., et al. (2004b). Parallelizing multi-objective evolutionary algorithms: Cone separation. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, volume 2, pages 1952–1957. IEEE.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. J. (1984). Classification and regression trees.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Brockhoff, D., Bader, J., Thiele, L., and Zitzler, E. (2013). Directed multiobjective optimization based on the weighted hypervolume indicator. *Journal of Multi-Criteria Decision Analysis*, 20(5-6):291–317.

Brockhoff, D., Tran, T.-D., and Hansen, N. (2015). Benchmarking numerical multiobjective optimizers revisited. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 639–646. ACM.

Brockhoff, D. and Zitzler, E. (2006a). Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature-PPSN IX*, pages 533–542. Springer.

Brockhoff, D. and Zitzler, E. (2006b). Dimensionality reduction in multiobjective optimization: The minimum objective subset problem. In *Operations Research Proceedings*, pages 423–429. Springer.

Broomhead, D. S. and Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom).

Buchanan, J. and Gardiner, L. (2003). A comparison of two reference point methods in multiple objective mathematical programming. *European Journal of Operational Research*, 149(1):17–34.

Bui, T., Hernández-Lobato, D., Hernandez-Lobato, J., Li, Y., and Turner, R. (2016). Deep Gaussian Processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481.

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

Chan, T. M. (2013). Klee's measure problem made easy. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 410–419. IEEE.

Chevalier, C., Emery, X., and Ginsbourger, D. (2015). Fast update of conditional simulation ensembles. *Mathematical Geosciences*, 47(7):771–789.

Chevalier, C. and Ginsbourger, D. (2013). Fast computation of the multi-points expected improvement with applications in batch selection. In *International Conference on Learning and Intelligent Optimization*, pages 59–69. Springer.

Chevalier, C., Ginsbourger, D., and Emery, X. (2014). Corrected kriging update formulae for batch-sequential data assimilation. In *Mathematics of Planet Earth*, pages 119–122. Springer.

Cinquegrana, D. and Iuliano, E. (2018). Investigation of adaptive design variables bounds in dimensionality reduction for aerodynamic shape optimization. *Computers & Fluids*, 174:89–109.

Coello, C. A. C. and Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.

Coello Coello, C. A. (2016). Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 563–587. ACM.

Colding, T. H. and Minicozzi, W. P. (2006). Shapes of embedded minimal surfaces. *Proceedings of the National Academy of Sciences*, 103(30):11106–11111.

Collette, Y. and Siarry, P. (2002). *Optimisation multiobjectif*. Editions Eyrolles.

Constantine, P., Dow, E., and Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59.

Corne, D. W. and Knowles, J. D. (2007). Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 773–780. ACM.

Couckuyt, I., Deschrijver, D., and Dhaene, T. (2014). Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 60(3):575–594.

Cressie, N. (1988). Spatial prediction and ordinary kriging. *Mathematical geology*, 20(4):405–421.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova*, 4(5):613–617.

Damianou, A. and Lawrence, N. (2013). Deep Gaussian Processes. In *Artificial Intelligence and Statistics*, pages 207–215.

Das, I. and Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657.

Davins-Valldaura, J., Moussaoui, S., Pita-Gil, G., and Plestan, F. (2017). ParEGO extensions for multi-objective optimization of expensive evaluation functions. *Journal of Global Optimization*, 67(1-2):79–96.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.

Deb, K., Miettinen, K., and Chaudhuri, S. (2010). Toward an estimation of Nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, 14(6):821–841.

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

Deb, K. and Saxena, D. (2006). Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, pages 3352–3360.

Deb, K. and Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 635–642. ACM.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer.

Deville, Y., Ginsbourger, D., Durrande, N., and Roustant, O. (2015). Package 'kergp'.

Draper, N. R. and Smith, H. (1998). *Applied regression analysis*, volume 326. John Wiley & Sons.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

Durrande, N. (2011). *Étude de classes de noyaux adaptées à la simplification et à l'interprétation des modèles d'approximation. Une approche fonctionnelle et probabiliste.* PhD thesis, École Nationale Supérieure des Mines de Saint-Étienne.

Durrande, N., Ginsbourger, D., and Roustant, O. (2012). Additive covariance kernels for high-dimensional Gaussian process modeling. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 481–499.

Duvenaud, D., Nickisch, H., and Rasmussen, C. E. (2011). Additive Gaussian processes. In *Advances in neural information processing systems*, pages 226–234.

Eaton, M. L. (1983). Multivariate statistics: a vector space approach. *John Wiley & Sons, Inc.*

Eggensperger, K., Hutter, F., Hoos, H., and Leyton-Brown, K. (2015). Efficient benchmarking of hyperparameter optimizers via surrogates. In *Twenty-Ninth AAAI Conference on Artificial Intelligence.*

Eiben, A. E. and Smith, J. E. (2003). *Introduction to evolutionary computing, second edition*, volume 53. Springer.

Emmerich, M., Beume, N., and Naujoks, B. (2005). An EMO algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer.

Emmerich, M., Deutz, A. H., and Klinkenberg, J. W. (2011). Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2147–2154. IEEE.

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

Emmerich, M. and Klinkenberg, J.-w. (2008). The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Technical Report, Leiden University*, 34:7–3.

Emmerich, M., Yang, K., and Deutz, A. (2020). Infill criteria for multiobjective Bayesian optimization. In *High-Performance Simulation-Based Optimization*, pages 3–16. Springer.

Emmerich, M., Yang, K., Deutz, A., Wang, H., and Fonseca, C. M. (2016). A multicriteria generalization of Bayesian global optimization. In *Advances in Stochastic and Deterministic Global Optimization*, pages 229–242. Springer.

Fang, K.-T., Li, R., and Sudjianto, A. (2005). *Design and modeling for computer experiments.* Chapman and Hall/CRC.

Feliot, P. (2017). *Une approche Bayesienne pour l'optimisation multi-objectif sous contraintes*. PhD thesis, Universite Paris-Saclay.

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

Feliot, P., Bect, J., and Vazquez, E. (2018). User preferences in Bayesian multi-objective optimization: the expected weighted hypervolume improvement criterion. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 533–544. Springer.

Fonseca, C. M. and Fleming, P. J. (1995). Multiobjective optimization and multiple constraint handling with evolutionary algorithms 1: A unified formulation.

Forrester, A. I. and Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1-3):50–79.

Frank, I. E. and Friedman, J. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135.

Frazier, P. I. and Clark, S. C. (2012). Parallel global optimization using an improved multi-points expected improvement criterion. In *INFORMS Optimization Society Conference*, volume 26.

Fricker, T. E., Oakley, J. E., and Urban, N. M. (2013). Multivariate Gaussian process emulators with nonseparable covariance structures. *Technometrics*, 55(1):47–56.

Fukunaga, K. and Olsen, D. R. (1971). An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 100(2):176–183.

Gal, T., Stewart, T., and Hanne, T. (1999). *Multicriteria decision making: advances in MCDM models, algorithms, theory, and applications*, volume 21. Springer Science & Business Media.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., and Cunningham, J. P. (2014). Bayesian optimization with inequality constraints. In *ICML*, pages 937–945.

Ginsbourger, D., Bay, X., Roustant, O., and Carraro, L. (2012). Argumentwise invariant kernels for the approximation of invariant functions. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 501–527.

Ginsbourger, D., Janusevskis, J., and Le Riche, R. (2011). Dealing with asynchronicity in parallel Gaussian process based global optimization. In *4th International Conference of the ERCIM WG on computing and statistics (ERCIM'11)*.

Ginsbourger, D. and Le Riche, R. (2010). Towards Gaussian process-based optimization with finite time horizon. In *mODa 9–Advances in Model-Oriented Design and Analysis*, pages 89–96. Springer.

Ginsbourger, D., Le Riche, R., and Carraro, L. (2010). Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pages 131–162. Springer.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1st edition.

Gramacy, R. B. and Lee, H. K. (2009). Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145.

Gretton, A. (2013). Introduction to RKHS, and some simple kernel algorithms. *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 16.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90.

Hartikainen, M., Miettinen, K., and Wiecek, M. M. (2011a). Constructing a Pareto front approximation for decision making. *Mathematical Methods of Operations Research*, 73(2):209–234.

Hartikainen, M., Miettinen, K., and Wiecek, M. M. (2011b). Decision making on Pareto front approximations with inherent nondominance. In *New State of MCDM in the 21st Century*, pages 35–45. Springer.

Hartikainen, M., Miettinen, K., and Wiecek, M. M. (2012). PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Computational optimization and applications*, 52(3):845–867.

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2).

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926.

Ho, Y.-C. and Pepyne, D. L. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of optimization theory and applications*, 115(3):549–570.

Horn, D., Wagner, T., Biermann, D., Weihs, C., and Bischl, B. (2015). Model-based multi-objective optimization: taxonomy, multi-point proposal, toolbox and benchmark. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 64–78. Springer.

Hussein, R. and Deb, K. (2016). A generative kriging surrogate model for constrained and unconstrained multi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 573–580. ACM.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28.

Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., and Nojima, Y. (2010). Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In *International Conference on Parallel Problem Solving from Nature*, pages 91–100. Springer.

Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2018). How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation*, 26(3):411–440.

Jacobs, E. N., Ward, K. E., and Pinkerton, R. M. (1933). The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel.

Janusevskis, J., Le Riche, R., and Ginsbourger, D. (2011). Parallel expected improvements for global optimization: summary, bounds and speed-up. Technical report, Institut Fayol, École des Mines de Saint-Étienne.

Janusevskis, J., Le Riche, R., Ginsbourger, D., and Girdziusas, R. (2012). Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In *Learning and Intelligent Optimization*, pages 413–418. Springer.

Jaszkiewicz, A. (2018). Improved quick hypervolume algorithm. *Computers & Operations Research*, 90:72–83.

Jeong, S. and Obayashi, S. (2005). Efficient Global Optimization (EGO) for multi-objective problem and data mining. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2138–2145. IEEE.

Jiao, R., Zeng, S., Li, C., Jiang, Y., and Jin, Y. (2019). A complete expected improvement criterion for Gaussian process assisted highly constrained expensive optimization. *Information Sciences*, 471:80–96.

Jiao, R., Zeng, S., Li, C., Jiang, Y., and Wang, J. (2018). Expected improvement of constraint violation for expensive constrained optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1039–1046. ACM.

Johnson, C. R. and Horn, R. A. (1985). *Matrix analysis*. Cambridge university press Cambridge.

Jolliffe, I. (2011). *Principal component analysis*. Springer.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383.

Jones, D. R. (2008). Large-scale multi-disciplinary mass optimization in the auto industry. In *MOPTA 2008 Conference*.

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

Kalai, E. and Smorodinsky, M. (1975). Other solutions to Nash's bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 513–518.

Keane, A. J. (2006). Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891.

Kim, I. Y. and de Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, 29(2):149–158.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

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

Knowles, J. and Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 711–716. IEEE.

Knowles, J. and Corne, D. (2003). Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116.

Knowles, J. D., Watson, R. A., and Corne, D. W. (2001). Reducing local optima in single-objective problems by multi-objectivization. In *International conference on evolutionary multi-criterion optimization*, pages 269–283. Springer.

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139.

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

Lacour, R., Klamroth, K., and Fonseca, C. M. (2017). A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79:347–360.

Launder, B. E. and Spalding, D. B. (1983). The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Elsevier.

Li, J., Bouhlel, M. A., and Martins, J. (2018a). A data-based approach for fast airfoil analysis and optimization. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1383.

Li, J., Cai, J., and Qu, K. (2019). Surrogate-based aerodynamic shape optimization with the active subspace method. *Structural and Multidisciplinary Optimization*, 59(2):403–419.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327.

Li, Z., Wang, X., Ruan, S., Li, Z., Shen, C., and Zeng, Y. (2018b). A modified hypervolume based expected improvement for multi-objective efficient global optimization method. *Structural and Multidisciplinary Optimization*, 58(5):1961–1979.

Liu, D. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

Liu, W., Zhang, Q., Tsang, E., Liu, C., and Virginas, B. (2007). On the performance of metamodel assisted MOEA/D. In *International Symposium on Intelligence Computation and Applications*, pages 547–557. Springer.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4):366–376.

López Jaimes, A., Coello Coello, C. A., and Chakraborty, D. (2008). Objective reduction using a feature selection technique. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 673–680. ACM.

Loshchilov, I., Schoenauer, M., and Sebag, M. (2010). Dominance-based Pareto-surrogate for multi-objective optimization. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 230–239. Springer.

Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395.

Marmin, S., Chevalier, C., and Ginsbourger, D. (2015). Differentiating the multipoint expected improvement for optimal batch design. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 37–48. Springer.

Marmin, S., Chevalier, C., and Ginsbourger, D. (2016). Efficient batch-sequential Bayesian optimization with moments of truncated Gaussian vectors. *arXiv preprint arXiv:1609.02700*.

Matheron, G. (1962). *Traité de géostatistique appliquée. 1 (1962)*, volume 1. Editions Technip.

Matheron, G. (1969). *Le krigeage universel*, volume 1. École nationale supérieure des mines de Paris Paris.

McCullagh, P. and Nelder, J. (1989). Generalized linear models.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Mebane Jr, W. R., Sekhon, J. S., et al. (2011). Genetic optimization using derivatives: the rgenoud package for R. *Journal of Statistical Software*, 42(11):1–26.

Mezura-Montes, E. and Coello, C. A. C. (2006). A survey of constraint-handling techniques based on evolutionary multiobjective optimization. In *Workshop paper at PPSN*.

Michalewicz, Z. (2013). *Genetic algorithms + data structures = evolution programs*. Springer Science & Business Media.

Miettinen, K. (1998). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., and Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542.

Mockus, J. (1975). On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer.

Molchanov, I. (2005). *Theory of Random Sets*. Probability and Its Applications. Springer London.

Molga, M. and Smutnicki, C. (2005). Test functions for optimization needs. *Test functions for optimization needs*, 101.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381–402.

Namura, N., Shimoyama, K., and Obayashi, S. (2017a). Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(6):898–913.

Namura, N., Shimoyama, K., and Obayashi, S. (2017b). Kriging surrogate model with coordinate transformation based on likelihood and gradient. *Journal of Global Optimization*, 68(4):827–849.

Nelsen, R. B. (2007). *An introduction to copulas*. Springer Science & Business Media.

Nocedal, J. and Wright, S. (2006). *Numerical optimization.* Springer Science & Business Media.

Palar, P. S. and Shimoyama, K. (2018). On the accuracy of kriging model in active subspaces. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0913.

Palar, P. S., Yang, K., Shimoyama, K., Emmerich, M., and Bäck, T. (2018). Multi-objective aerodynamic design with user preference using truncated expected hypervolume improvement. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1333–1340. ACM.

Parr, J. M. (2013). *Improvement criteria for constraint handling and multiobjective optimization.* PhD thesis, University of Southampton.

Parr, J. M., Forrester, A. I., Keane, A. J., and Holden, C. M. (2012a). Enhancing infill sampling criteria for surrogate-based constrained optimization. *Journal of Computational Methods in Sciences and Engineering*, 12(1-2):25–45.

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

Phan, D. H. and Suzuki, J. (2013). R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1836–1845. IEEE.

Picheny, V. (2014). A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial Intelligence and Statistics*, pages 787–795.

Picheny, V. (2015). Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280.

Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., and Kim, N.-H. (2010). Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7):071008.

Picheny, V., Gramacy, R. B., Wild, S., and Le Digabel, S. (2016). Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In *Advances in neural information processing systems*, pages 1435–1443.

Ponweiser, W., Wagner, T., Biermann, D., and Vincze, M. (2008). Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection. In *International Conference on Parallel Problem Solving from Nature*, pages 784–794. Springer.

Pronzato, L. (2017). Minimax and maximin space-filling designs: some properties and methods for construction.

Qian, H. and Yu, Y. (2017). Solving high-dimensional multi-objective optimization problems with low effective dimensions. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Rachmawati, L. and Srinivasan, D. (2006). Preference incorporation in multi-objective evolutionary algorithms: A survey. In *2006 IEEE International Conference on Evolutionary Computation*, pages 962–968. IEEE.

Raghavan, B., Breitkopf, P., Tourbier, Y., and Villon, P. (2013). Towards a space reduction approach for efficient structural shape optimization. *Structural and Multidisciplinary Optimization*, 48(5):987–1000.

Raghavan, B., Le Quilliec, G., Breitkopf, P., Rassineux, A., Roelandt, J.-M., and Villon, P. (2014). Numerical assessment of springback for the deep drawing process by level set interpolation using shape manifolds. *International journal of material forming*, 7(4):487–501.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

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

Ribaud, M. (2018). *Krigeage pour la conception de turbomachines: grande dimension et optimisation robuste*. PhD thesis, Université de Lyon.

Roustant, O., Ginsbourger, D., and Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization.

Rullière, D., Durrande, N., Bachoc, F., and Chevalier, C. (2018). Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing*, 28(4):849–867.

Russo, L. M. and Francisco, A. P. (2014). Quick hypervolume. *IEEE Transactions on Evolutionary Computation*, 18(4):481–502.

Sacher, M., Duvigneau, R., Le Maitre, O., Durand, M., Berrini, E., Hauville, F., and Astolfi, J.-A. (2018). A classification approach to efficient global optimization in presence of non-computable domains. *Structural and Multidisciplinary Optimization*, 58(4):1537–1557.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, pages 409–423.

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M. (2004). Sensitivity analysis in practice: a guide to assessing scientific models. *Chichester, England*.

Sawaragi, Y., Nakayama, H., and Tanino, T. (1985). *Theory of multiobjective optimization*, volume 176. Elsevier.

Saxena, D. K. and Deb, K. (2007). Trading on infeasibility by exploiting constraint's criticality through multi-objectivization: A system design perspective. In *2007 IEEE Congress on Evolutionary Computation*, pages 919–926. IEEE.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schonlau, M. (1997). *Computer experiments and global optimization*. PhD thesis, University of Waterloo.

Segura, C., Coello, C. A. C., Miranda, G., and León, C. (2016). Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization. *Annals of Operations Research*, 240(1):217–250.

Shah, A. and Ghahramani, Z. (2016). Pareto frontier learning with expensive correlated objectives. In *International Conference on Machine Learning*, pages 1919–1927.

Shah, A., Wilson, A., and Ghahramani, Z. (2014). Student-t processes as alternatives to Gaussian processes. In *Artificial intelligence and statistics*, pages 877–885.

Shahriari, B., Bouchard-Côté, A., and Freitas, N. (2016). Unbounded Bayesian optimization via regularization. In *Artificial Intelligence and Statistics*, pages 1168–1176.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Shan, S. and Wang, G. G. (2004). Space exploration and global optimization for computationally intensive design problems: a rough set based approach. *Structural and Multidisciplinary Optimization*, 28(6):427–441.

Shan, S. and Wang, G. G. (2010). Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241.

Shimoyama, K., Sato, K., Jeong, S., and Obayashi, S. (2012). Comparison of the criteria for updating kriging response surface models in multi-objective optimization. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.

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

Singh, H. K., Bhattacharjee, K. S., and Ray, T. (2016). A projection-based approach for constructing piecewise linear Pareto front approximations. *Journal of Mechanical Design*, 138(9):091404.

Singh, P., Couckuyt, I., Ferranti, F., and Dhaene, T. (2014). A constrained multi-objective surrogate-based optimization algorithm. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3080–3087. IEEE.

Smith, G. D. (1985). *Numerical solution of partial differential equations: finite difference methods*. Oxford university press.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. (2015). Scalable Bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180.

Sobol', I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802.

Spagnol, A., Le Riche, R., and Da Veiga, S. (2019). Global sensitivity analysis for optimization with variable selection. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2):417–443.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Stegmann, M. B. and Gomez, D. D. (2002). A brief introduction to statistical shape analysis. *Informatics and mathematical modelling, Technical University of Denmark, DTU*, 15(11).

Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151.

Stein, M. (1999). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Steuer, R. E. and Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical programming*, 26(3):326–344.

Stork, J., Friese, M., Zaefferer, M., Bartz-Beielstein, T., Fischbach, A., Breiderhoff, B., Naujoks, B., and Tusar, T. (2020). Open issues in surrogate-assisted optimization. In *High-Performance Simulation-Based Optimization*, pages 225–244. Springer.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979.

Svenson, J. (2011). *Computer experiments: Multiobjective optimization and sensitivity analysis*. PhD thesis, The Ohio State University.

Svenson, J. and Santner, T. J. (2010). Multiobjective optimization of expensive black-box functions via expected maximin improvement. *The Ohio State University, Columbus, Ohio*, 32.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574.

Toure, C., Auger, A., Brockhoff, D., and Hansen, N. (2019). On bi-objective convex-quadratic problems. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 3–14. Springer.

Tripathy, R., Bilionis, I., and Gonzalez, M. (2016). Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation. *Journal of Computational Physics*, 321:191–223.

Van Laarhoven, P. J. and Aarts, E. H. (1987). Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, V. and Chervonenkis, A. (1974). Theory of pattern recognition.

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

Wagner, T., Emmerich, M., Deutz, A., and Ponweiser, W. (2010). On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 718–727. Springer.

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer.

Wang, J., Clark, S. C., Liu, E., and Frazier, P. I. (2016). Parallel Bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*.

Wang, Q. (2012). Kernel principal component analysis and its applications in face recognition and active shape models. *arXiv preprint arXiv:1207.3538*.

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and De Freitas, N. (2013). Bayesian optimization in high dimensions via random embeddings. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

While, L., Bradstreet, L., and Barone, L. (2012). A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95.

Wierzbicki, A. (1980). The use of reference objectives in multiobjective optimization. In *Multiple criteria decision making theory and application*, pages 468–486. Springer.

Wierzbicki, A. (1999). Reference point approaches. published in multicriteria decision making: Advances in MCDM models, algorithms, theory, and applications. T. Gal, TJ Stewart and T. Hanne.

Wilson, J., Hutter, F., and Deisenroth, M. (2018). Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 9884–9895.

Wolpert, D. H., Macready, W. G., et al. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Wu, J. and Frazier, P. (2016). The parallel knowledge gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 3126–3134.

Wu, X., Peng, X., Chen, W., and Zhang, W. (2019). A developed surrogate-based optimization framework combining HDMR-based modeling technique and TLBO algorithm for high-dimensional engineering problems. *Structural and Multidisciplinary Optimization*, pages 1–18.

Yang, K., Deutz, A., Yang, Z., Back, T., and Emmerich, M. (2016a). Truncated expected hypervolume improvement: Exact computation and application. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 4350–4357. IEEE.

Yang, K., Emmerich, M., Deutz, A., and Bäck, T. (2019a). Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*.

Yang, K., Emmerich, M., Deutz, A., and Bäck, T. (2019b). Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and evolutionary computation*, 44:945–956.

Yang, K., Emmerich, M., Deutz, A., and Fonseca, C. M. (2017). Computing 3-D expected hypervolume improvement and related integrals in asymptotically optimal time. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 685–700. Springer.

Yang, K., Gaida, D., Bäck, T., and Emmerich, M. (2015). Expected hypervolume improvement algorithm for PID controller tuning and the multiobjective dynamical control of a biogas plant. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 1934–1942. IEEE.

Yang, K., Li, L., Deutz, A., Back, T., and Emmerich, M. (2016b). Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on*, pages 276–281. IEEE.

Yang, K., Palar, P. S., Emmerich, M., Shimoyama, K., and Bäck, T. (2019c). A multi-point mechanism of expected hypervolume improvement for parallel multi-objective Bayesian global optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 656–663. ACM.

Yi, G., Shi, J., and Choi, T. (2011). Penalized Gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics*, 67(4):1285–1294.

Zeleny, M. (1976). The theory of the displaced ideal. In *Multiple criteria decision making Kyoto 1975*, pages 153–206. Springer.

Zhan, D., Cheng, Y., and Liu, J. (2017). Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(6):956–975.

Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.

Zhang, Q., Liu, W., Tsang, E., and Virginas, B. (2009). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474.

Zhao, G., Arroyave, R., and Qian, X. (2018). Fast exact computation of expected hypervolume improvement. *arXiv preprint arXiv:1812.07692*.

Zitzler, E. (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications.

Zitzler, E., Brockhoff, D., and Thiele, L. (2007). The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103.

Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms-a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301. Springer.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca Grunert, V. (2002). Performance assessment of multiobjective optimizers: An analysis and review. *TIK-Report*, 139.

Zuhal, L. R., Palar, P. S., and Shimoyama, K. (2019). A comparative study of multi-objective expected improvement for aerodynamic design. *Aerospace Science and Technology*, 91:548–560.

Zurada, J. M. (1992). *Introduction to artificial neural systems*, volume 8. West publishing company St. Paul.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT: 2019LYSEM026

David GAUDRIE

High-Dimensional Bayesian Multi-Objective Optimization

Speciality: Applied Mathematics

Keywords: Gaussian Processes, Bayesian Optimization, Multi-Objective Optimization, Dimension Reduction.

Abstract:

This thesis focuses on the simultaneous optimization of expensive-to-evaluate functions that depend on a high number of parameters. This situation is frequently encountered in fields such as design engineering through numerical simulation. Bayesian optimization relying on surrogate models (Gaussian Processes) is particularly adapted to this context. The first part of this thesis is devoted to the development of new surrogate-assisted multi-objective optimization methods. To improve the attainment of Pareto optimal solutions, an infill criterion is tailored to direct the search towards a user-desired region of the objective space or, in its absence, towards the Pareto front center introduced in our work. Besides targeting a well-chosen part of the Pareto front, the method also considers the optimization budget in order to provide an as wide as possible range of optimal solutions in the limit of the available resources. Next, inspired by shape optimization problems, an optimization method with dimension reduction is proposed to tackle the curse of dimensionality. The approach hinges on the construction of hierarchized problem-related auxiliary variables that can describe all candidates globally, through a principal component analysis of potential solutions. Few of these variables suffice to approach any solution, and the most influential ones are selected and prioritized inside an additive Gaussian Process. This variable categorization is then further exploited in the Bayesian optimization algorithm which operates in reduced dimension.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2019LYSEM026

David GAUDRIE

Optimisation bayésienne multi-objectif en haute dimension

Spécialité : Mathématiques Appliquées

Mots clefs : Processus gaussiens, Optimisation bayésienne, Optimisation multi-objectif, Réduction de dimension.

Résumé :

Dans cette thèse, nous nous intéressons à l'optimisation simultanée de fonctions coûteuses à évaluer et dépendant d'un grand nombre de paramètres. Cette situation est rencontrée dans de nombreux domaines tels que la conception de systèmes en ingénierie au moyen de simulations numériques. L'optimisation bayésienne, reposant sur des méta-modèles (processus gaussiens) est particulièrement adaptée à ce contexte. La première partie de cette thèse est consacrée au développement de nouvelles méthodes d'optimisation multi-objectif assistées par méta-modèles. Afin d'améliorer le temps d'atteinte de solutions Pareto optimales, un critère d'acquisition est adapté pour diriger l'algorithme vers une région de l'espace des objectifs plébiscitée par l'utilisateur ou, en son absence, le centre du front de Pareto introduit dans nos travaux. Outre le ciblage, la méthode prend en compte le budget d'optimisation, afin de restituer un éventail de solutions optimales aussi large que possible, dans la limite des ressources disponibles. Dans un second temps, inspirée par l'optimisation de forme, une approche d'optimisation avec réduction de dimension est proposée pour contrer le fléau de la dimension. Elle repose sur la construction, par analyse en composantes principales de solutions candidates, de variables auxiliaires adaptées au problème, hiérarchisées et plus à même de décrire les candidats globalement. Peu d'entre elles suffisent à approcher les solutions, et les plus influentes sont sélectionnées et priorisées au sein d'un processus gaussien additif. Cette structuration des variables est ensuite exploitée dans l'algorithme d'optimisation bayésienne qui opère en dimension réduite.