OPTIMISATION BAYÉSIENNE MULTI-OBJECTIF APPLICATION A LA PLANIFICATION INNOVANTE DES RÉSEAUX DE DISTRIBUTION EN PRÉSENCE D'ÉNERGIES RENOUVELABLES

Rapport de stage de fin d'étude – Master 2 MVA

Christophe Vuong

Christophe Vuong
Elève en double-diplôme master à Télécom Paris
École Normale Supérieure Paris-Saclay (France)
christophe.vuong@telecom-paris.fr

Encadré par
Bruno Tebbal Barracosa, EDF Lab Paris-Saclay, Palaiseau (France)
Héloïse Baraffe EDF Lab Paris-Saclay, Palaiseau (France)
Julien Bect, L2S, CentraleSupélec (France)
Emmanuel Vazquez, L2S, CentraleSupélec (France)

Mai – Octobre 2020

Table des matières

1 Introduction

Ce stage de recherche est effectué dans le cadre du Master 2 Mathématiques Vision et Apprentissage de l'ENS Paris-Saclay, conjointement au sein de l'équipe "Fonctionnement des systèmes électriques et raccordement" (R12) du département R&D "Economie, Fonctionnement et Etude des Systèmes Energétiques (EFESE) d'EDF Lab Paris-Saclay, et du laboratoire des Signaux et Systèmes (L2S) de CentraleSupélec à Gif-Sur-Yvette. EDF Lab Saclay est un des sites de recherche du premier fournisseur d'électricité en France. Le département EFESE travaille entre autres sur le projet RAPSODI qui porte notamment sur la planification des réseaux de distribution électrique, en anticipant l'intégration des énergies renouvelables, source d'incertitudes. Mon stage s'intègre dans ce projet de recherche, à dominante électrotechnique. Ce projet dure de 2020 à 2022. Le début de stage s'est déroulé en télétravail à temps plein. Je suis revenu sur site à partir de juillet à hauteur d'une fois par semaine en raison notamment, du temps passé dans les transports pour venir au lieu de stage.

Le travail effectué porte sur l'optimisation des stratégies de planification des réseaux de distribution d'électricité. Le stage s'insère dans le travail de Bruno Tebbal Barracosa, actuellement en thèse CIFRE au sein de l'équipe R12. Ses encadrants de thèse Emmanuel Vazquez et Julien Bect ont aussi supervisé mon travail. La thèse formalise l'optimisation comme un problème d'optimisation multi-objectif avec des données de sortie bruitées. C'est un problème peu abordé dans la littérature, mais il y a beaucoup de méthodes pour l'optimisation mono-objectif bruité, et l'optimisation multi-objectif avec des données non bruitées. L'idée est de joindre les deux mondes pour proposer une résolution du problème. Les techniques d'optimisation bayésienne sont les solutions privilégiées. Les enjeux du stage sont de comprendre des algorithmes d'optimisation bayésienne, d'implémenter certains d'entre eux, à savoir PESMO et MESMO et de les adapter au contexte très bruité. Ma contribution principale a été d'étudier l'algorithme d'optimisation bayésienne multi-objectif à l'état de l'art PESMO (Hernández-Lobato *et al.*, 2016), et en particulier l'algorithme d'approximation bayésienne sous-jacent, *Expectation Propagation* avec une attention portée sur les paramètres pouvant intervenir dans celui-ci.

2 Remerciements

Le télétravail a pu être déroutant initialement, mais j'ai tout de même passé cinq mois enrichissants. Tout d'abord, j'exprime toute ma gratitude à mes tuteurs de stage d'EDF Bruno Tebbal Barracosa et Héloïse Baraffe pour leur attention à mon égard même lors de périodes charnières pour la thèse. Le suivi pendant le télétravail a été important pour me débloquer, et me transmettre les enjeux de la planification. Nous avons privilégié l'outil Microsoft Teams pour les points de stage et autres conversations. Par ailleurs ce travail aurait moins de valeur ajoutée si les résultats n'étaient comparés à ceux de méthodes éprouvées. En ce sens, les travaux de Bruno Tebbal Barracosa sur PALS notamment, ont été précieux tant dans la phase d'implémentation des méthodes d'optimisation, que dans la phase d'évaluation de leurs performances.

Ensuite, je remercie chaleureusement Julien Bect pour sa disponibilité à tout instant, sa compétence et maîtrise des algorithmes. Il m'a donné de nombreuses suggestions concernant les tests d'implémentation à réaliser. Je remercie aussi Emmanuel Vazquez pour les mises au point théoriques et les conseils pour la présentation. L'article sur l'algorithme PESMO n'était pas simple à décrypter, et les mises à point avec eux ont permis de comprendre mieux la démarche de l'auteur de celui-ci. Ils ont mis à disposition la bibliothèque STK (Bect *et al.*, 2020). Elle m'a fait gagner un temps précieux, fournissant tous les outils nécessaires pour faire du krigeage, de l'optimisation multi-objectif et la visualisation des résultats.

3 Présentation générale

3.1 Contexte du stage

Traditionnellement, pour résoudre des contraintes de tension ou de courant sur le réseau électrique, notamment dans le cas de nouveaux arrivants sur le réseau, le gestionnaire de réseau de distribution a recours à du renforcement qui consiste essentiellement à rajouter des transformateurs, ou augmenter la section des câbles pour tolérer une plus grande puissance. A cela s'ajoute la production d'énergies renouvelables (ENR) qui connaît des pics importants sur une partie de l'année seulement. Les solutions citées ci-dessus concernent le cas de production stable sur une année. D'autres leviers d'action existent comme l'écrêtement de la production des producteurs d'électricité, mais ce sont des solutions d'étude. L'étude d'autres leviers d'action vise à permettre l'intégration des ENR à moyen/long terme (Dutrieux, 2015, Chapitre 1). Un des enjeux est de pouvoir planifier efficacement la distribution d'énergie. Une première approche est de procéder par recherche exhaustive dans l'espace des paramètres de stratégie pour arriver à une utilisation optimale de ces leviers d'action. Or, cette approche est trop lente pour pouvoir réitérer plusieurs fois les expériences d'optimisation dans un temps limité. L'utilisation de méthodes découlant sur des algorithmes plus puissants s'impose. Comme la planification se fait sur une plage de temps de 5 à 20 ans à partir de maintenant, elle est simulée. Nous considérons les sorties d'un simulateur que l'on cherche à optimiser. Une évaluation du simulateur, appelé PARADIS (Dutrieux, 2015), renvoie des indicateurs de performance technico-économique d'une stratégie de planification paramétrée pour un réseau de distribution et un scénario pluriannuel d'arrivée d'ENR (Figure 1).

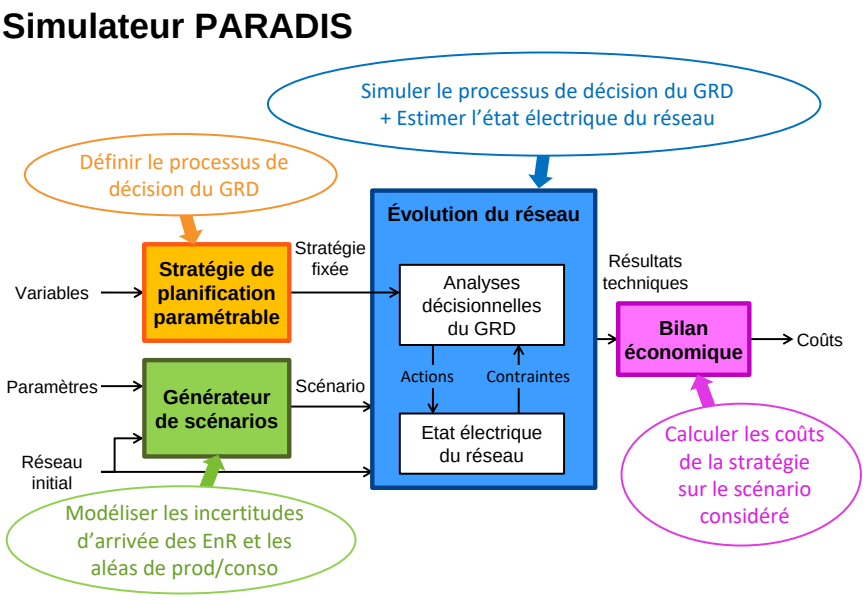


FIGURE 1 – Fonctionnement du simulateur PARADIS (Planification Avancée des Réseaux Actifs de DIStribution)

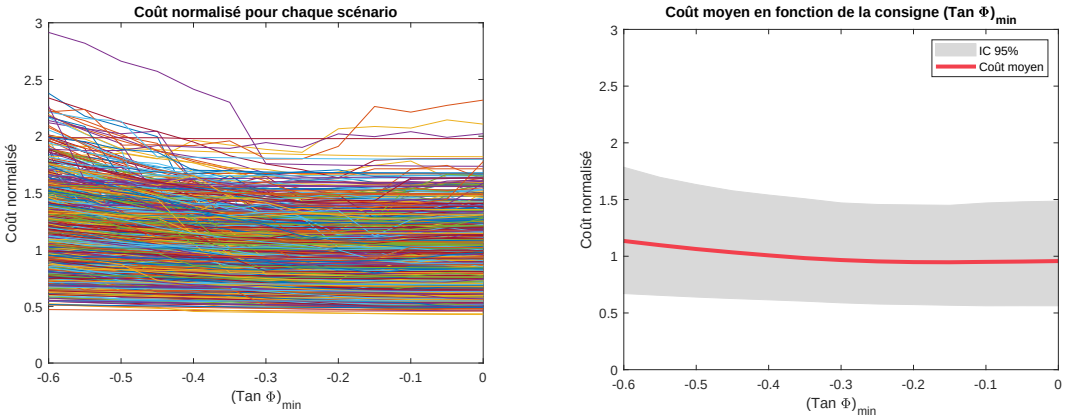
Les décisions sont prises par le gestionnaire de réseau pour raccorder de nouveaux producteurs et résoudre des contraintes sur le réseau. Le processus de décision (ou stratégie de planification) du gestionnaire de réseau décrit notamment les leviers d'action à disposition du gestionnaire pour résoudre les contraintes. Celui-ci se fait en utilisant un "arbre de décision" dont les branches sont définies par des règles d'arbitrage ou lois en vigueur. Dans la suite, on appelle l'espace d'entrée $\mathbb{X}$, l'espace des paramètres de stratégies. Il est de cardinal fini N. Dans notre cadre, c'est un sous-espace de $\mathbb{R}^D$. Tout au long du stage, le cas d'étude à traiter correspond à $D = 2$. Un paramètre de stratégie x décrit :

- la consigne tangente phi $x_1 = (\tan \Phi)_{min}$;
- le taux d'effacement $x_2 = \tau_{eff,max}$.

L'autre entrée du simulateur concerne les scénarios. Un scénario définit deux choses :

- la liste des caractéristiques des nouveaux producteurs à raccorder (puissance nominale, type de production entre éolien et photovoltaïque, localisation géographique et année d'arrivée) ;
- les profils de production et de consommation annuels au pas 10 minutes.

Bien qu'on peut décider sur quel scénario on lance la simulation, les scénarios d'insertion des ENR sont multiples. En effet, les études déjà menées donnent une idée des gros producteurs à raccorder dans les 2 premières années, mais pas au-delà. Comme on ne peux prévoir quel scénario va se produire, on ne sait pas exactement quelles contraintes risquent d'apparaître et donc quels coûts leur résolution va engendrer. La plupart des énergies renouvelables donnent une production très variable et peu prévisible sur une année. En utilisant un générateur aléatoire de scénarios, il est possible de prendre en compte l'incertitude sur les indicateurs de performance. Le simulateur PARADIS peut être considéré comme stochastique si à chaque nouvelle évaluation, un scénario est en effet généré aléatoirement. La figure 2 montre des valeurs d'indicateurs de coûts normalisés par souci de confidentialité vis-à-vis des données fournies par EDF Lab. Il est à noter l'ampleur de l'aléa (Figure 2).



(a) (b)

FIGURE 2 – Coût normalisé pour 1091 scénarios issu du simulateur PARADIS

Les observations des fonctions objectif dépendent des scénarios, mais **chacune des fonctions objectif dépend uniquement des paramètres de stratégie x. Ce sont des espérances des sorties aléatoires de PARADIS.** Sur la figure 2, on a affiché par rapport les coûts pour chaque scénario, et le coût moyen par rapport à une coordonnée de x, la consigne tangente Φ. L'optimisation vise aussi à prendre en compte tous les scénarios envisageables, et pas seulement le scénario du pire des cas (Dutrieux, 2015, Chapitre 6). Si on travaille sur le type de sortie indiqué dans la figure 1, les fonctions à optimiser sont des coûts moyens (Figure 2b). Les observations ou évaluations des indicateurs de sortie de PARADIS (Figure 2a) sont vues comme des évaluations bruitées de fonctions objectif dans le cadre du problème d'optimisation.

De plus, une contrainte majeure est le temps de calcul d'une évaluation. Il est de l'ordre de 30 secondes, ce qui limite le budget pour l'optimisation, appelé N_{budget}. En effet, **les algorithmes utilisés sont amenés à procéder à l'optimisation pour divers réseaux de distribution d'électricité.** Le coût de l'évaluation des fonctions objectif a pour conséquence que les méthodes classiques d'optimisation globale par exploration aléatoire (Monte-Carlo, recuit simulé), qui requièrent des milliers d'évaluations, sont inutilisables.

3.2 Objectifs visés

Une première solution d'optimisation du problème a été de traiter le problème mono-objectif de la minimisation du coût moyen vis-à-vis d'une stratégie. Or, l'équipe qui travaille sur PARADIS a jugé que d'autres critères peuvent venir compléter l'analyse de performance d'une stratégie. Dans le cadre de la thèse de mon tuteur de stage, quelques indicateurs statistiques plus pertinents sont proposés, comme le taux annuel maximal de consommateurs raccordés en basse tension et sujets à des contraintes de tension, dit "taux de clients mal alimentés". Le but est alors de minimiser le coût moyen d'une stratégie paramétrable sur un réseau donné,

compte tenu des incertitudes liées à la production d'énergies renouvelables, tout en assurant un taux de clients mal alimentés faible comme critère de qualité. Pour ce problème, le nombre de fonctions objectif r est pris égal à 2 par défaut. Nous voulons résoudre le problème suivant :

$$\min_{x \in \mathbb{X}} f_1(x), \ldots, f_r(x), \tag{3.1}$$

qui est un problème multi-objectif. Les modélisations des fonctions objectif et des paramètres d'entrée ont été transposées sous forme de code MATLAB par mes tuteurs de stage, en amont du stage. Les fonctions objectif sont notées f_k. L'ordre de ces fonctions objectifs n'intervient pas dans l'optimisation, mais nous fixons dès lors, la définition des fonctions objectif dans les scripts de code comme indiqué dans la figure 3. **Par souci de confidentialité, j'ai travaillé sur des données du simulateur récoltées et utilisées dans les travaux de mon tuteur.**

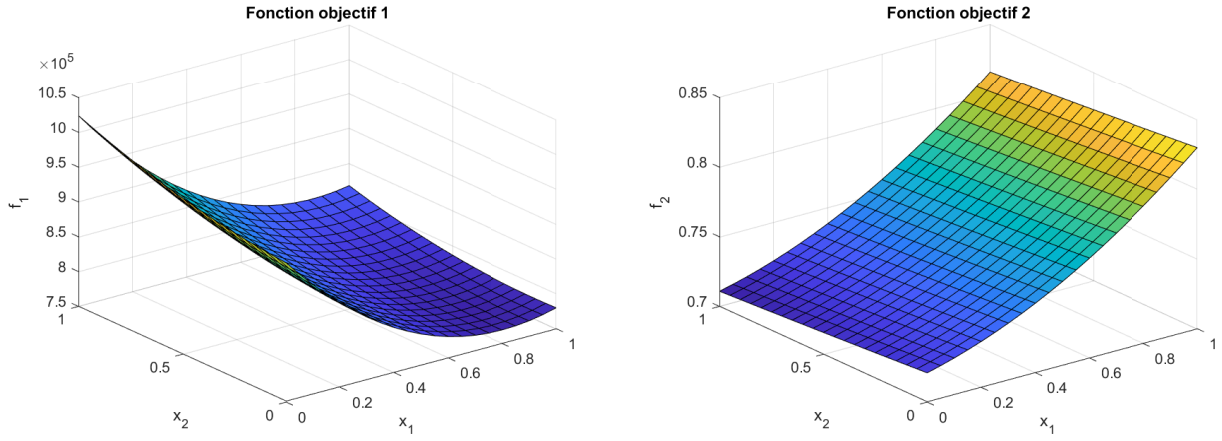


FIGURE 3 – Fonctions objectif

Dans le cas idéal, on espère pouvoir trouver une solution optimale en résolvant chacun des problèmes mono-objectifs $\min_{x \in \mathbb{X}} f_k(x)$ pour $k = 1, \ldots, r$ individuellement. Or, les deux critères de performance sont conflictuels : une stratégie moins coûteuse peut en effet conduire à un taux de clients mal alimentés plus élevé. Actuellement il est difficile d'estimer avec exactitude le coût associé au taux de clients mal alimentés. C'est pourquoi cet indicateur doit être dissocié de celui du coût dans le problème. Dans le cas contraire, des stratégies d'apparence moins chère seraient privilégiées alors qu'elles dégraderaient la qualité d'approvisionnement en électricité sur les réseaux basse tension. La prise en compte de ces deux objectifs conflictuels conduit donc à une multitude de solutions. On peut privilégier l'aspect couverture des clients au coût moyen ou l'inverse. Nous nous intéressons donc à plusieurs solutions de paramètres de stratégie qui soient "plutôt bonnes" pour les deux objectifs, s'apparentant à des compromis. La difficulté est donc double :

- mener une optimisation multi-objectif qui donnent un large panel de compromis entre les r fonctions objectif ;
- faire de l'optimisation à partir d'un nombre restreint d'observations stochastiques.

Une première approche de résolution du problème multi-objectif est de procéder par scalarisation du problème en minimisant une combinaison linéaire des fonctions objectif. On se ramène alors à un problème mono-objectif, même si en pratique, ces méthodes n'explorent qu'une fraction des solutions de compromis. Dans la suite, nous introduisons la formalisme de l'optimisation multi-objectif. Au début du stage, des articles de l'état de l'art ont été mis à disposition (Hernández-Lobato *et al.*, 2016).

> Nous noterons f l'ensemble des r fonctions objectifs $\{f_1, \ldots, f_r\}$.

3.3 Notions de Pareto-optimalité

Dans un contexte de minimisation sur un espace $\mathbb{X}$, la notion de minimiseur global x^* est remplacée par celle d'ensemble de Pareto $\mathcal{X}^*$ qui correspond à une collection de **solutions Pareto-optimales**. Ce sont des

points de l'espace d'entrée pour lesquels aucune valeur de fonctions objectif f_k ne peut être améliorée sans en détériorer une autre.

Définition 3.1. Domination

$$f(x) \prec f(x^{'}) \iff \forall k \in [\![1,r]\!] \quad f_k(x) \leq f_k(x^{'}) \text{ et } \exists j \in [\![1,r]\!] \quad f_j(x) < f_j(x^{'}). \tag{3.2}$$

On dit que $f(x)$ Pareto-domine $f(x^{'})$. A f fixé, on écrira que x domine $x^{'}$.

La définition d'un sens faible pour la domination se restreint à la première contrainte.

Définition 3.2. Non-domination

Soit $x \in \mathbb{X}$, le point x est dit non-dominé si $\nexists x^{'} \in \mathbb{X}$ tel que $f(x^{'}) \prec f(x)$.

Définition 3.3. Ensemble de Pareto

L'ensemble de Pareto correspond à l'ensemble des points non dominés de $\mathbb{X}$, i.e. : $\{x^* \in \mathbb{X} : \nexists x^{'} \in \mathbb{X} \quad f(x^{'}) \prec f(x^*)\}$. Une autre définition est :

$$\mathcal{X}^* := \{x^* \in \mathbb{X} : \forall x \in \mathbb{X} \backslash \{x^*\}, \exists k \in [\![1,r]\!] \quad f_k(x^*) < f_k(x) \text{ ou } \forall j \in [\![1,r]\!] \quad f_j(x^*) \leq f_j(x)\}. \tag{3.3}$$

On peut alternativement considérer l'ensemble des points non dominés au sens strict :

$$\mathcal{X}^* := \{\, x^* \in \mathbb{X} \,:\, \forall x \in \mathbb{X} \setminus \{x^*\},\, \exists k \in [\![1,r]\!] \quad f_k(x^*) < f_k(x) \,\}. \tag{3.4}$$

Définition 3.4. Front de Pareto

Le front de Pareto $\mathcal{Y}^*$ est l'image par les fonctions objectifs de l'ensemble de Pareto.

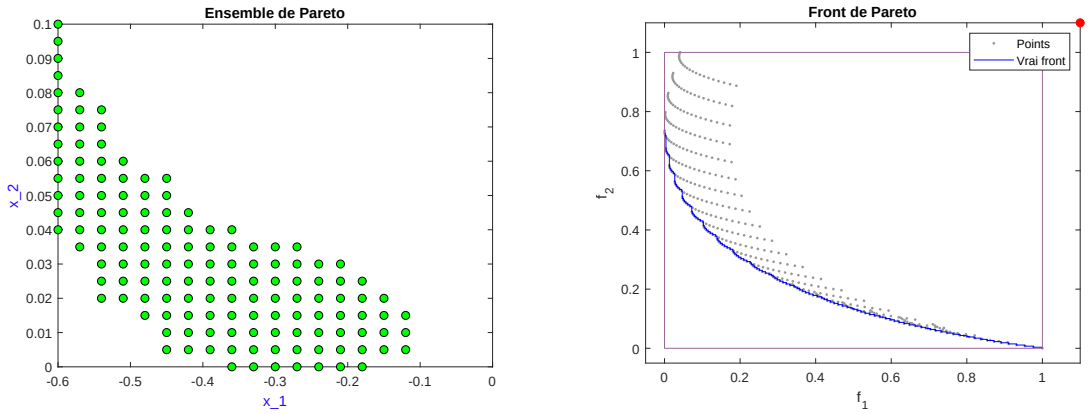
Remarque 3.1. Les points de $\mathcal{X}^*$ ne se dominent pas les uns les autres.

Remarque 3.2. L'ensemble des points non dominés au sens strict est un sous-ensemble de l'ensemble de Pareto qui capture l'essentiel de $\mathcal{X}^*$ dans le problème que l'on considère. Les points de $\mathcal{X}^*$ hors de ce sous-ensemble sont des points de bords.

Comme notre espace d'entrée $\mathbb{X}$ est fini, l'ensemble de Pareto $\mathcal{X}^*$ est fini. Il est non vide car les fonctions objectif (Figure 3) sont continues bornées. Pour notre problème bi-objectif, nous disposons de données MATLAB qui forme le problème dit de "base".

Remarque 3.3. Procéder à une normalisation des fonctions objectifs ne modifie pas l'ensemble de Pareto. Le facteur de normalisation peut être différent pour chaque fonction objectif.

En pratique, nous normalisons de sorte que les valeurs des fonctions objectif se trouvent dans l'intervalle $[0, 1]$. L'ensemble et front de Pareto qui en découlent sont :



(a) (b)

FIGURE 4 – Ensemble / front de Pareto

Le minimum par objectifs est le point $I(0,0)$. Il n'est pas atteignable. Le front de Pareto est tel qu'aucun point (Figure 4b) de l'espace fonctionnel atteignable ne puisse être en-dessous. Les points de l'ensemble de Pareto représentent donc les solutions de compromis optimal. Les techniques d'optimisation étudiées utilisent comme entrée des paramètres de stratégie $x \in \mathbb{X}$. Elles traitent pas les coordonnées des vecteurs x séparément.

> Dans la suite, on indexe l'espace d'entrée par $[\![1, N]\!]$. On note les points de l'espace d'entrée $(x_i)_{i \in [\![1,N]\!]}$. On notera n le nombre de points évalués à l'itération courante des algorithmes d'optimisation.

> **Remarque 3.4.** Si on avait accès à toutes les évaluations non bruitées des fonctions objectif sur $\mathbb{X}$, le problème multi-objectif se résoudrait en testant pour chaque $x \in \mathbb{X}$ la règle de domination. C'est ainsi que l'on obtient les ensemble et front de Pareto (Figure 4) permettant de mesurer les performances des algorithmes d'optimisation.

Les fonctions objectif ne peuvent qu'être approchées. On peut alors obtenir l'ensemble de Pareto associé à l'ensemble des r approximations de fonctions. Il existe plusieurs méthodes d'approximations de f à partir d'observations non bruitées $(x^{(1:n)}, f^{(1:n)})$:

- l'interpolation par morceaux ;
- l'interpolation barycentrique ;
- la régression polynomiale ;
- l'approximation par splines ;
- le krigeage.

Or, les méthodes d'interpolation sont simples à mettre en oeuvre mais ne sont pas les plus efficaces pour obtenir une approximation très régulière dans le voisinage des points d'évaluation dans l'espace d'entrée. La méthode de régression polynomiale dépend comme les méthodes d'interpolation de la configuration des points d'évaluation de l'espace d'entrée. Certaines configurations peuvent entraîner une erreur d'approximation qui augmente indéfiniment avec le nombre d'évaluations n, problème que l'on appelle phénomène de Runge. Les deux dernières méthodes permettent, en théorie, d'obtenir une bonne approximation suffisamment lisse et plate adaptée à notre problème (Figure 2b) même en cas d'observations très dispersées (Dutrieux, 2015). Le krigeage peut être vu comme une régression régularisée, équivalente à une approximation par splines en matière d'estimation (Vazquez, 2005). Ce sont toutes des méthodes qui fournissent un méta-modèle, une approximation peu coûteuse sur tout l'espace d'entrée $\mathbb{X}$. Dans le cadre du stage, le méta-modèle sur lequel se base l'optimisation est construit par krigeage (Jones *et al.*, 1998). Les autres méthodes d'approximation donnent des résultats intéressants d'approximation, mais c'est le krigeage qui a été retenu pour la thèse et le stage. Le méta-modèle probabiliste issu du krigeage permet d'obtenir les estimations locales ainsi que les variances d'estimation associées. Il est donc possible de calculer des erreurs d'approximation et d'analyser la fiabilité globale du méta-modèle.

4 Démarche de travail

4.1 Méta-modèle de krigeage et bruit

Chacune des fonctions objectif f_k est modélisée par $F_k(x) = m_k(x) + \xi_k(x)$ pour $k \in [\![1, r]\!]$ avec :

- m_k : la structure déterministe de l'espérance de F_k ;
- ξ_k : un processus aléatoire, souvent supposé stationnaire, d'espérance nulle, et de fonction de covariance dont les paramètres sont à estimer.

Nous adoptons des notations différentes des séries temporelles en notant les processus $(F_k(x))_{x\in\mathbb{X}}$ pour $k \in [\![1, r]\!]$ et une trajectoire de processus pour une particule ω de l'univers donnée, $f : x \longmapsto F(x)(\omega)$.

> Comme le domaine de définition des fonctions est un espace de cardinal N, l'ensemble f désigne le vecteur déplié de toutes les valeurs prises par les fonctions objectif $[f_1(x_1) \dots f_1(x_N) \dots\dots f_r(x_1) \dots f_r(x_N)]^\top$. Dans la suite nous considérons les processus aléatoires dépliés F et ξ et le vecteur $\mu = [\mu_1 \dots \mu_r]^\top$.

Il existe trois formes de krigeage selon la forme de l'espérance μ :

- le krigeage simple, quand μ est une constante connue ;
- le krigeage ordinaire, quand μ est une constante inconnue ;
- le krigeage universel, quand μ est une combinaison linéaire de fonctions quelconques.

Dans la suite, on s'intéressera à du krigeage ordinaire, notamment dans les implémentations d'algorithmes. Concrètement, cela consiste, à partir du modèle F, à chercher une estimation de f en x sous la forme d'une combinaison linéaire de n points déjà observés :

$$\hat{F}(x) = \sum_{i=1}^{n} \lambda_i(x) f(x^{(i)}) = \lambda(x)^\top f^{(1:n)}. \tag{4.1}$$

$\hat{F}(x)$ est la prédiction par krigeage de $F(x)$ définie comme la prédiction linéaire non biaisée qui minimise la variance de prédiction $\Delta F(x) = \mathbb{E}[(\hat{F}(x) - F(x))^2]$. La méthode de krigeage est très flexible sur le cadre probabiliste, et permet notamment de prendre en compte le bruit des évaluations dans le méta-modèle. Dans la figure 5, l'histogramme rend compte de la répartition des évaluations du coût de scénario normalisé pour un paramètre de stratégie donné x. Des lois plus complexes comme des lois de Student ou des lois normales tronquées peuvent être considérées pour la modélisation de cette distribution, mais nous prenons le modèle le plus courant pour modéliser le bruit.

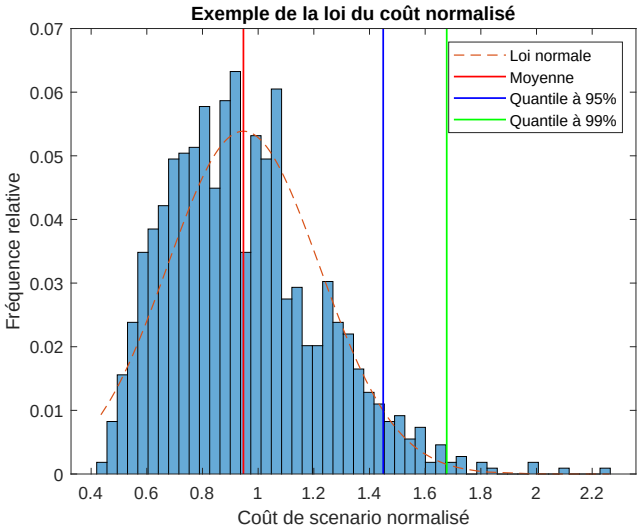


FIGURE 5 – Histogramme du coût normalisé par scénario

La loi normale correspondante à cette répartition est $\mathcal{N}(0.9471, 0.2733^2)$. La moyenne correspond à la valeur de la fonction objectif au point donné x. Il en découle une modélisation des observation bruitées :

$$Y(x) = \underbrace{F(x)}_{\text{moyenne}} + \underbrace{B(x)}_{\text{bruit additif}} . \tag{4.2}$$

Pour $k \in [\![1, r]\!]$. les processus ξ_k sont indexés sur $\mathbb{X}$ de fonction de covariance $\gamma : \mathbb{X} \times \mathbb{X} \longmapsto \mathbb{R}$ qui a les propriétés de symétrie hermitienne et de positivité. Les processus sont pris stationnaires. Différentes fonctions de covariance invariantes par translation peuvent être être considérées suivant l'information *a priori* sur les fonctions objectif. Le point de vue du krigeage permet de considérer un noyau comme une covariance et les méthodes d'estimation statistique deviennent alors applicables pour les paramètres de covariance. Pour le stage, nous nous utilisons la classe des noyaux de Matérn qui suppose une certaine régularité des fonctions. La fonction de Matérn peut s'écrire :

$$K(h) = \frac{\check{\sigma}^2}{2^{\nu-1}\Gamma(\nu)} \left(\frac{2\nu^{1/2}}{\rho} \right) \mathcal{K}_\nu \left(\frac{2\nu^{1/2} h}{\rho} \right),$$

où $\mathcal{K}_\nu$ est la fonction de Bessel modifiée de seconde espèce. Les paramètres de *portée de la corrélation entre les données* ρ et $\check{\sigma}^2$ sont à estimer par maximum de vraisemblance (voir annexe A.3). Des problèmes numériques peuvent apparaître pour des valeurs élevées, donc nous considérons typiquement $\nu < 5$. En pratique, nous prenons $\nu = 5/2$. Dans la suite, nous ne nous intéresserons pas à l'estimation des paramètres qui se fait en marge de l'optimisation. Nous notons donc K le noyau en omettant la dépendance aux paramètres de covariance. Ainsi :

$$Cov(F(x_s), F(x_t)) = K(\|x_s - x_t\|). \tag{4.3}$$

A partir de (4.3), on en déduit la covariance du processus Y :

$$Cov(Y(x_s), Y(x_t)) = K(\|x_s - x_t\|) + Cov(B(x_s), F(x_t)) + Cov(B(x_t), F(x_s)) + Cov(B(x_t), B(x_s)).$$

Une hypothèse classique est de considérer le bruit additif comme indépendant du processus aléatoire F, donc on se ramène à :

$$Cov(Y(x_s), Y(x_t)) = K(\|x_s - x_t\|) + Cov(B(x_t), B(x_s)). \tag{4.4}$$

> Par souci de simplicité, on suppose que pour tout k, B_k est un bruit blanc gaussien.

Dans la réalité, la matrice de covariance $Cov(B)$ est loin d'être diagonale. Pour les 13 paramètres x_t qui permettent d'obtenir les précédentes figures, on peut calculer un estimateur de covariance basé sur les 1091 coûts de scénarios. Nous avons une différence relative en norme infinie et 2 de l'ordre de 0,91 entre la covariance estimée et la covariance en égalant à 0 les termes diagonaux pour ces 13 paramètres. La fonction de covariance associée au processus Y vérifie donc :

$$Cov(Y(x_s), Y(x_t)) = K(\|x_s - x_t\|) + \sigma^2(x_s)\mathbb{1}_{\{s=t\}}. \tag{4.5}$$

> On suppose que le bruit est homoscédastique, i.e. $Y_k \sim \mathcal{GP}(\mu_k, K + \sigma_k^2 I)$ pour $k \in [\![1, r]\!]$.

Contrairement à la précédente, cette hypothèse semble vérifiée par les données (Figure 6).

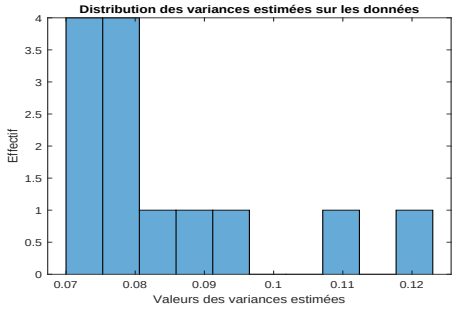


FIGURE 6 – Histogramme des valeurs des variances de bruit pour 13 paramètres de stratégie

La variance de bruit σ_k^2 pour tout $k \in [\![1, r]\!]$ est un paramètre de covariance de plus à estimer. Il est à noter que l'amplitude du bruit est très grande, de l'ordre du tiers de la moyenne, ce qui est à prendre en compte dans l'implémentation d'une méthode d'optimisation.

4.2 Optimisation bayésienne dans un contexte multi-objectif très bruité

Il s'agit d'évaluer des "bons points" de l'espace d'entrée qui fournit de l'information sur les solutions que l'on cherche. C'est sur ce principe que repose les algorithmes d'optimisation bayésienne. A chaque itération d'un algorithme d'optimisation bayésienne, on sélectionne un point à évaluer de façon à enrichir le méta-modèle pour l'optimisation (Figure 7). A chaque nouveau point évalué, on a une nouvelle loi *a posteriori* conditionnée par les évaluations de ces points. La moyenne prédictive μ de la loi *a posteriori* du processus gaussien s'interprète comme une fonction de régression des données sélectionnées $(x^{(i)}, f(x^{(i)}))_{i \in [\![1,n]\!]}$ dans le cadre non bruité. Il est à noter que dans le cas bruité, la moyenne prédictive ne prend pas forcément les mêmes valeurs que les évaluations des points sélectionnés. La sélection des points $(x^{(i)})_{i \in [\![1,n]\!]}$ repose sur la maximisation d'un critère d'échantillonnage, i.e. :

$$x^{(n+1)} = \underset{x \in \mathbb{X}}{\operatorname{argmax}}\, J(x, \mathcal{D}_n), \tag{4.6}$$

où J est un critère d'échantillonnage et $\mathcal{D}_n = \cup_{i=1}^n \{x^{(i)}, Y(x^{(i)})\}$. Le meilleur prédicteur linéaire non biaisé $\hat{F}$ est le prédicteur bayésien (voir annexe A.2) ou moyenne prédictive. L'ajustement du méta-modèle de krigeage F aux données $\mathcal{D}_n$ permet de calculer la fonction d'acquisition.

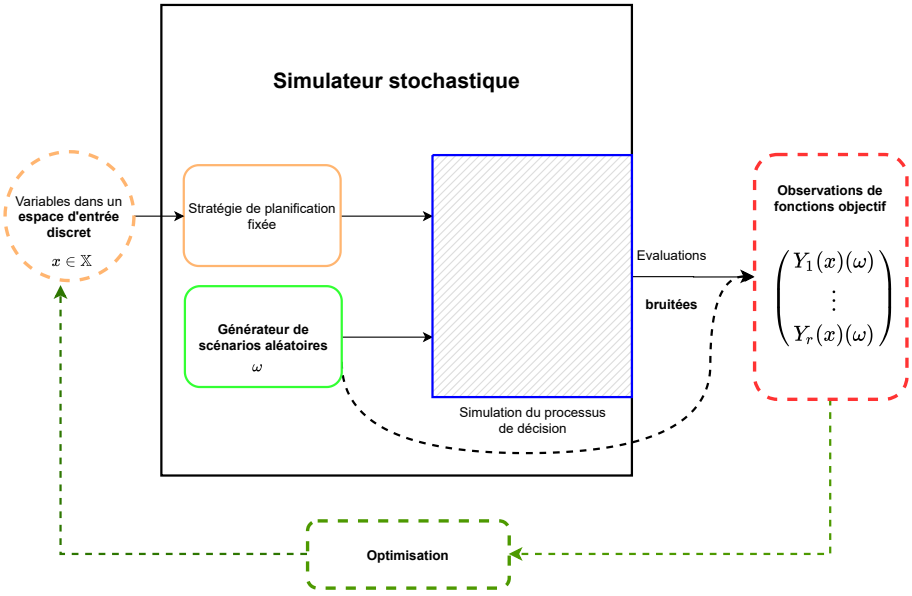


FIGURE 7 – Optimisation à partir d'un simulateur stochastique

Le cadre multi-objectif nécessite en plus, de faire une hypothèse supplémentaire sur les processus gaussiens F_k qui compose le processus déplié F.

> Par souci de simplicité, nous supposons que les observations des fonctions objectif sont *a priori* indépendantes : $p(y|\mathcal{D}_n, x) = \prod_{k=1}^r p(y_k|\mathcal{D}_n, x)$. Les processus F_k sont en effet supposés mutuellement indépendants.

Cette hypothèse intervient lors du calcul de la fonction d'acquisition $J_n = J(\cdot|\mathcal{D}_n)$. Les paramètres associés à une classe de noyau dépendent souvent du problème, et sont généralement estimés au fil des itérations de l'optimisation. A l'initialisation de l'algorithme d'optimisation bayésienne, cela se fait en évaluant n_0 points initiaux. Cette étape sert notamment à la normalisation des fonctions objectif qui s'avère primordiale en pratique, et l'estimation du niveau de bruit. De façon générale, un algorithme d'optimisation sélectionne itérativement des points d'évaluation comme présenté dans la figure 8 :

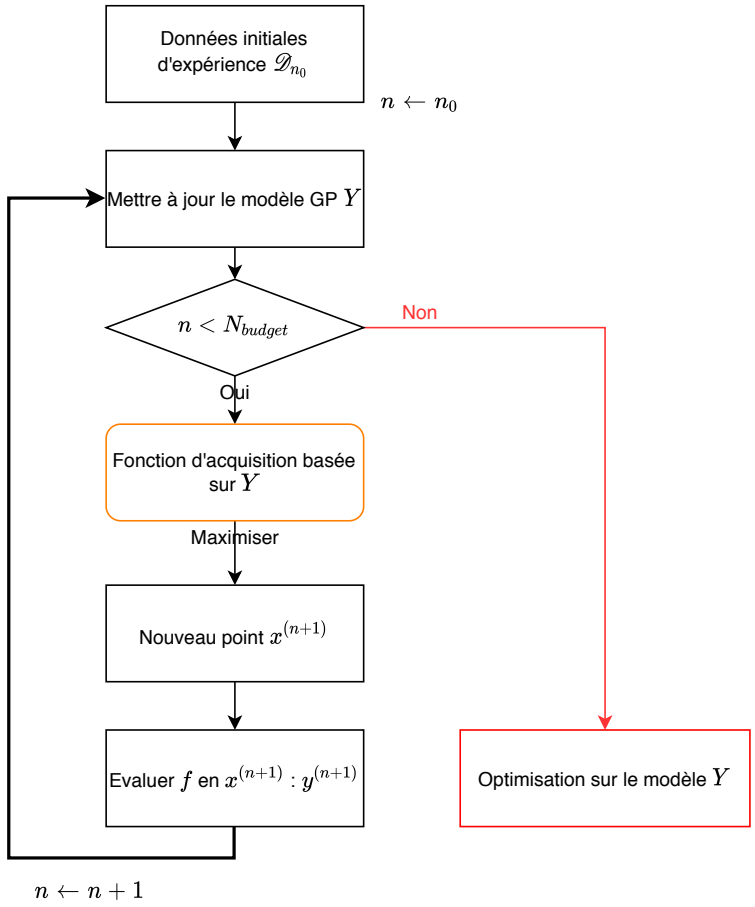
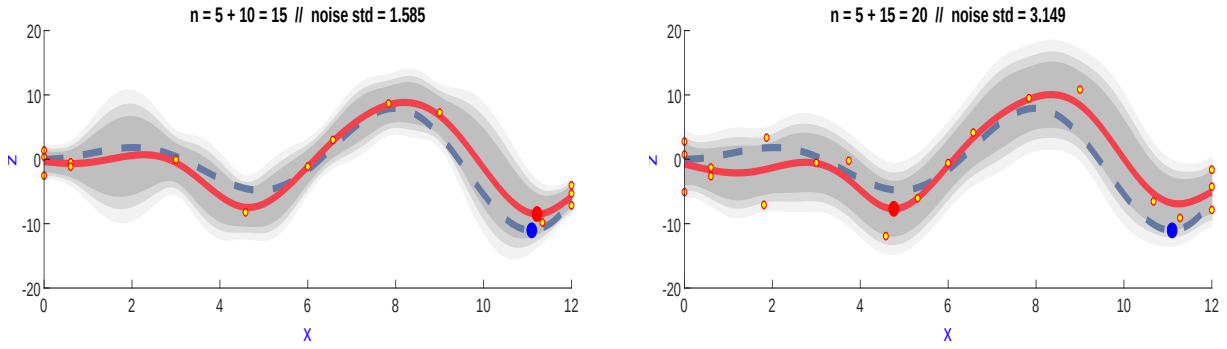


FIGURE 8 – Une itération d'optimisation bayésienne

Une fois que les paramètres de covariance sont estimés, la performance de l'algorithme d'optimisation dépend du critère d'échantillonnage, tant en terme de temps de calcul qu'en terme d'apport d'information sur l'ensemble de Pareto. Comme pour de l'optimisation globale mono-objectif, même avec un très bon critère d'échantillonnage, les algorithmes d'optimisation bayésienne multi-objectif ne renvoient pas exactement l'ensemble de Pareto. Il s'agit donc de trouver la meilleure approximation possible de l'ensemble de Pareto. En effet, il est possible de définir un classement des solutions du problème en fonction des rangs de classification de Pareto. On pose les ensembles $\mathcal{X}_0^*, \mathcal{X}_1^*, \dots$ tels que : $\mathcal{X}_0^* = \mathcal{X}^*$, et pour tout $k \geq 0$, $\mathcal{X}_{k+1}^*$ est l'ensemble de Pareto sur $\mathbb{X} \setminus \left(\sqcup_{j=1}^{k} \mathcal{X}_j^*\right)$. $\mathcal{X}_k^*$ est l'ensemble de Pareto de rang k. Pour $k \geq 1$, ce n'est plus l'ensemble de Pareto, mais pour k proche de 1, les points de $\mathcal{X}_k^*$ sont de bonnes solutions du problème. Visuellement, ils font partie du nuage de points les plus proches du front de Pareto au sens de la distance euclidienne sur $\mathbb{R}^D$. Il vaut mieux fournir une liste exhaustive de "bonnes" solutions dans notre contexte. De cette notion de rang, il en découle une métrique qui mesure le rang maximal parmi les solutions du front estimé pour le méta-modèle. Or, en raison du bruit très important, les rangs de classification peuvent être sous-estimés ou sur-estimés. Cela se traduit aussi par une difficulté à juger si un front estimé est meilleur qu'un autre pour approximer le vrai front de Pareto. Néanmoins, si le bruit n'est pas trop important, cette métrique donne une bonne idée de l'"épaisseur" du front estimé. En pratique, elle n'est pas utilisée, car elle n'est pas assez fine pour décrire la diversité des formes géométriques de front de Pareto à épaisseur égale. Nous considérons donc des métriques à valeurs dans $\mathbb{R}$ dans la suite. Les premières conclusions des expériences réalisées dans le cadre de la thèse de mon maître de stage montrent que le niveau de bruit d'origine du problème est trop élevé pour pouvoir comparer des algorithmes d'optimisation bayésienne en termes de valeurs de métriques.

Prenons l'exemple jouet d'un problème mono-objectif pour illustrer. La figure 9 montre des itérations d'optimisation bayésienne dans un contexte très bruité pour la fonctions $x \longmapsto x\sin(x)$. Les points ont été sélectionnés avec un des critères d'échantillonnage décrits dans la suite du rapport.

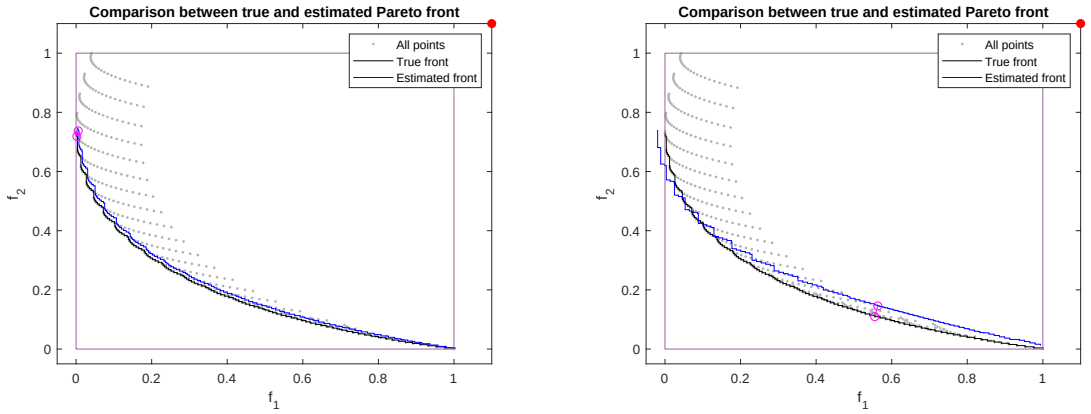


(a) Ecart-type de bruit $\sigma = 2$ (b) Ecart-type de bruit $\sigma = 4$

FIGURE 9 – Comparaison de position de minimum avec le critère PES mono

Les 2 expériences ont été réalisées avec la même graine de générateur aléatoire, en ajoutant un bruit blanc gaussien. La légende pour chacune des figures montre l'estimation de niveau de bruit, et l'itération courante de l'algorithme. En doublant l'écart-type du bruit gaussien, le problème devient plus dur à résoudre à un budget fixé. Avec un écart-type de bruit modéré (Figure 9a), on obtient une bonne estimation du minimiseur (en bleu). L'incertitude du méta-modèle de krigeage en zone grisée reste certes importante, car peu de points ont été évalués. L'estimation du minimum n'est en revanche pas très bonne. Dans le cas de la figure 9b, même après 20 points sélectionnés, l'estimation du minimiseur (en rouge) est mauvaise, car la valeur de la moyenne prédictive du minimiseur est bien plus élevée que la vraie valeur de la fonction (en pointillé) en ce point.

Pour réduire l'impact du bruit sur le résultat de l'optimisation, il faut consommer plus de budget. D'une part, réévaluer plusieurs fois les points $x^{(1:n)}$ permet de réduire l'amplitude du bruit, et ainsi de faire en sorte que les prédictions par krigeage restent proches de la vraie fonction aux points d'évaluation. En effet, si on évalue un point x par lot de n_{batch}, comme les variables aléatoires de bruit sont supposées $i.i.d$, les variances résultantes du bruit valent $\frac{1}{n_{batch}}\sigma_k^2$ pour $k \in [\![1, r]\!]$. C'est aussi une quantité n_{batch} du budget qui est consommée à chaque itération, ramenant le nombre maximal d'itérations pour un algorithme d'optimisation à $\lfloor \frac{N_{budget} - n_0 \times n_{batch}}{n_{batch}} \rfloor$. D'autre part, pour bien optimiser, il est fondamental d'explorer suffisamment l'espace $\mathbb{X}$, le risque étant d'explorer trop loin des zones d'intérêts où se trouvent les points non dominés. Il y a donc un compromis à trouver pour une bonne utilisation du budget d'évaluations.



(a) Contexte non bruité (b) Contexte très bruité

FIGURE 10 – Ensemble / front de Pareto

La figure 10b montre un exemple où le front estimé 1 est meilleur sur la partie supérieure de l'espace fonctionnel, et moins bon sur la partie inférieure par rapport au vrai front. Ce n'est presque jamais le cas après quelques itérations d'un algorithme d'optimisation dans le cadre non bruité (Figure 10a). Il n'est pas évident dans le cas de la figure 10b de juger à partir de la visualisation du front si les solutions trouvées sur la partie

supérieure du front estimé sont Pareto-optimales. C'est pourquoi des métriques remaniées à partir des métriques existantes dans la littérature ont été retenues pour comparer les implémentations des algorithmes d'optimisation bayésienne (cf. section 9.1).

4.3 Notations

Dans cette sous-section, je répertorie les notations qui reviennent à plusieurs reprises dans ce rapport de stage. Dans la suite, à certains passages, nous noterons des lois de probabilité comme des densités par rapport à la mesure de Lebesgue. En effet, toutes les lois de probabilité considérées sont absolument continues. De plus, comme l'espace $\mathbb{X}$ d'indexation est fini, les ensembles de fonctions objectif sont notés comme des vecteurs que l'on dira dépliés. Les figures concernant l'optimisation sont légendées en anglais, car pouvant être amenées à être réutilisées dans d'autres documents.

Notations :

$\mathbb{X}$	l'espace d'entrée fini de cardinal N
$\mathbb{Y}$	l'espace de sortie (fonctionnel)
f	le vecteur déplié des r fonctions objectif $\{f_1, \dots f_r\}$
F	le processus déplié des r processus gaussiens $F_k \sim \mathcal{GP}(0, K) \quad \forall k \in [\![1, r]\!]$
f	le vecteur déplié de fonctions objectif ou de trajectoires de F
y	le vecteur des observations bruitées sous processus gaussien
$x^{(i)}$	un point sélectionné par l'optimisation bayésienne
$f^{(i)}$	$f(x^{(i)})$
$y^{(i)}$	$y(x^{(i)})$
$x^{(1:n)}$	l'ensemble des points sélectionnés
$y^{(1:n)}$	le vecteur des n observations sélectionnées $(y^{(1)}, \dots, y^{(n)})$
$\mathcal{D}_n$	$\big((x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\big)$ les points et leurs évaluations à l'itération n
Y	le processus aléatoire déplié des observations
$\mathcal{X}^*$	l'ensemble de Pareto déterministe associé à f
x^*	un point de l'ensemble de Pareto (point non dominé)
$\mathcal{Y}^*$	le front de Pareto associé à f
$\mathcal{X}^{all}$	l'ensemble aléatoire de Pareto sous processus gaussiens
$\mathcal{X}^{sub}$	ensemble aléatoire de points non dominés en tant que sous-ensemble de $\mathcal{X}^{all}$
$\mathcal{Y}^{all}$	l'ensemble aléatoire de front de Pareto sous processus gaussiens
$\mathcal{Y}^{sub}$	ensemble aléatoire du front de points non dominés
J_n	la fonction d'acquisition sachant $\mathcal{D}_n$
$\mathcal{I}_n$	le gain d'information (information mutuelle) sachant $\mathcal{D}_n$
H_n	l'entropie sachant $\mathcal{D}_n$
$\mathcal{H}_n$	l'entropie différentielle sachant $\mathcal{D}_n$
$p(\cdot \mid \dots)$	la densité de $\cdot$ sachant $\dots$ par rapport à une mesure de référence à préciser

5 Critère d'échantillonnage de réduction d'entropie

Nous nous intéressons à des critères d'échantillonnage se basant la théorie de l'information, en particulier ceux reposant sur la réduction d'entropie (ou d'incertitude) sur les solutions Pareto-optimales.

5.1 Distribution de probabilité d'ensemble de Pareto

Plutôt que de s'intéresser à des estimations de l'ensemble de Pareto, nous proposons d'adapter une méthode de probabilité des minimiseurs (Villemonteix, 2008, Chapitre 1) au cadre multi-objectif. A chaque trajectoire de F correspond une réalisation d'ensemble de Pareto. Formellement, on définit un ensemble aléatoire à valeurs dans $\mathcal{P}(\mathbb{X})$, l'ensemble des parties finies de $\mathbb{X}$.

Définition 5.1. Ensemble aléatoire de Pareto

$$\mathcal{X}^{all} = \{x^* \in \mathbb{X} \,:\, \nexists x^{'} \in \mathbb{X} \setminus \{x^*\} \quad F(x^{'}) \prec F(x^*)\} \tag{5.1}$$

$\mathcal{X}^{all}$ est presque sûrement non vide, car les composantes du processus déplié F ont des trajectoires continues bornées presque sûrement. Comme $\mathbb{X}$ est un espace fini, la loi de $\mathcal{X}^{all}$ peut être décrite par :

$$\mathbb{P}(\mathcal{X}^{all} = \mathcal{X} \mid F = f) = \mathbb{1}(\mathcal{X} = \mathcal{X}^*). \tag{5.2}$$

Les points dominés correspondent alors à des réalisations dans cet ensemble aléatoire.

Définition 5.2. Ensemble aléatoire de points non dominés

Appelons $\mathcal{P}(\mathcal{X}^{all})$ l'ensemble des parties finies de $\mathcal{X}^{all}$. L'ensemble aléatoire $\mathcal{X}^{sub}$ est tiré uniformément dans $\mathcal{P}(\mathcal{X}^{all})$. $\mathcal{X}^{sub}$ est à valeurs dans $\mathcal{P}(\mathbb{X})$.

Pour déterminer l'entropie de cet ensemble aléatoire, il s'agit d'étudier la densité de probabilité $p_{\mathcal{X}^{sub}}(\mathcal{X}|\mathcal{D}_n)$ de l'ensemble (aléatoire) contenant des vecteurs aléatoires X^* de points non dominés. Cependant, l'existence de cette densité n'est pas simple à établir, et nous ne disposons pas d'une expression analytique dans un espace d'entrée euclidien. Comme $\mathbb{X}$ est fini, la mesure de référence est la mesure de comptage sur $\mathcal{P}(\mathbb{X})$. Il est à noter que pour l'ensemble aléatoire, il n'y a pas de choix canoniques pour une mesure de référence. Dans ce cas, on peut exprimer la probabilité d'être ensemble de points non dominés.

Soit $\mathcal{X} \in \mathcal{P}(\mathbb{X})$. On rappelle que $\mathcal{X}^*$ l'ensemble de Pareto associé à f, d'où :

$$\begin{aligned}\mathbb{P}(\mathcal{X}^{sub} = \mathcal{X} \mid F = f) &= \prod_{x^* \in \mathcal{X}} \mathbb{1}(x^* \in \mathcal{X}^*) \,/\, |\mathcal{P}(\mathcal{X}^*)| \\ &= \prod_{x^* \in \mathcal{X}} \prod_{x' \in \mathbb{X}} \left(1 - \prod_{k=1}^{r} \mathbb{1}_{\{f_k(x^*) - f_k(x') \geq 0\}}\right) \,/\, |\mathcal{P}(\mathcal{X}^*)| \quad \text{non-domination au sens strict.}\end{aligned} \tag{5.3}$$

Le produit a bien un nombre fini de facteurs alors que si l'espace $\mathbb{X}$ était non dénombrable, il faudrait définir un "produit infini" (Hennig et Schuler, 2012, Appendix A).

Remarque 5.1. La notion d'ensemble aléatoire de points non dominés a été peu décrite dans la littérature, contrairement à son équivalent en optimisation globale mono-objectif (Villemonteix, 2008; Hennig et Schuler, 2012). L'article *Predictive Entropy Search for Multi-Objective Bayesian Optimization* (Hernández-Lobato *et al.*, 2016) présente cette notion de façon informelle afin de proposer un calcul de fonction d'acquisition par réduction d'entropie.

Dans des articles récents au sujet de l'optimisation bayésienne multi-objectif (Belakaria *et al.*, 2019), le critère d'échantillonnage proposé se base sur l'ensemble aléatoire de front de Pareto au lieu de l'ensemble de

Pareto. On peut définir l'ensemble (aléatoire) $\mathcal{Y}^{all}$ comme le front de Pareto associé aux trajectoires f de F. Cela mène à une approche de réduction d'entropie dans l'espace de sortie tandis que la méthode qui est décrite par la suite concerne l'espace d'entrée.

5.2 Entropie conditionnelle des solutions Pareto-optimales

En analogie aux travaux de Villemonteix (2008), quantifions la connaissance accumulée sur les points non dominés par l'entropie de $p_{\mathcal{X}^{sub}}(\cdot|\mathcal{D}_n)$ après n itérations du processus d'optimisation, c'est-à-dire définir l'entropie de l'ensemble aléatoire de points non dominés :

$$H_n(\mathcal{X}^{sub}) = H(\mathcal{X}^{sub}|\mathcal{D}_n) = -\sum_{\mathcal{X}\in\mathcal{P}(\mathbb{X})} p_{\mathcal{X}^{sub}}(\mathcal{X}|\mathcal{D}_n)\log\left(p_{\mathcal{X}^{sub}}(\mathcal{X}|\mathcal{D}_n)\right). \tag{5.4}$$

A partir de cette définition de l'entropie des points non dominés, l'idée est d'échantillonner là où il y a le plus de gain d'information dans l'espace d'entrée sur la loi de $\mathcal{X}^{sub}$. Pour cela, on se base sur les travaux de Hennig et Schuler (2012) en cherchant à maximiser l'information de gain de $\mathcal{X}^{sub}$ sur $Y(x)$ une observation (ou information mutuelle de $\mathcal{X}^{sub}$ et $Y(x)$).

Définition 5.3. Réduction d'entropie dans l'espace d'entrée

$$\begin{aligned}\mathcal{I}_n(\mathcal{X}^{sub}; Y(x)) &= H_n(\mathcal{X}^{sub}) - H_n(\mathcal{X}^{sub}|Y(x)) \\ &= H_n(\mathcal{X}^{sub}) - \mathbb{E}_{Y\sim\mathbb{P}_{Y|\mathcal{D}_n}}\left[H_n(\mathcal{X}^{sub}|Y(x))\right]\end{aligned} \tag{5.5}$$

$$x^{(n+1)} = \underset{x\in\mathbb{X}}{\operatorname{argmax}}\, \mathcal{I}_n(\mathcal{X}^{sub}; Y(x)) \tag{5.6}$$

De façon analogue, la réduction d'entropie peut être adaptée pour quantifier l'information sur des points du front de Pareto au lieu de l'ensemble de Pareto.

Définition 5.4. Réduction d'entropie dans l'espace de sortie

$$\begin{aligned}\mathcal{I}_n(\mathcal{Y}^{sub}; Y(x)) &= H_n(\mathcal{Y}^{sub}) - H_n(\mathcal{Y}^{sub}|Y(x)) \\ &= H_n(\mathcal{Y}^{sub}) - \mathbb{E}_{Y\sim\mathbb{P}_{Y|\mathcal{D}_n}}\left[H_n(\mathcal{Y}^{sub}|Y(x))\right]\end{aligned} \tag{5.7}$$

Comme l'intérêt porte sur l'ensemble de Pareto, nous délaissons dans un premier temps la réduction d'entropie dans l'espace de sortie. Or, la loi de l'ensemble aléatoire de Pareto est inconnue. De plus, la fonction d'acquisition donnée ici ne peut être calculée exactement. En revanche, on connaît la loi *a posteriori* de Y. Une approche est d'utiliser la symétrie de l'information mutuelle (Houlsby *et al.*, 2012; Hernández-Lobato *et al.*, 2014). Ici, il est à noter que les ensembles aléatoires ont une densité par rapport à la mesure de comptage évoquée précédemment tandis que le vecteur aléatoire déplié Y a une densité par rapport à la mesure de Lebesgue. Cela n'est pas un problème dans l'expression de l'information mutuelle, la seule condition pour pouvoir écrire l'information mutuelle étant que la loi jointe soit absolument continue par rapport au produit des lois marginales. Néanmoins, ce renversement d'écriture amène à considérer la loi d'un vecteur aléatoire absolument continue par rapport à la mesure de Lebesgue. Le calcul de l'information mutuelle fait donc intervenir la notion d'entropie différentielle.

6 Predictive Entropy Search for Multi-Objective problem

6.1 Redéfinition de la fonction d'acquisition

Le renversement d'écriture est exploitée dans l'article *Predictive Entropy Search for Multi objective Bayesian Optimization* (Hernández-Lobato *et al.*, 2016) ainsi que la thèse de Shah (2020) sur des méthodes d'optimisation bayésienne. Nous notons $\mathcal{H}_n$ l'entropie différentielle sachant $\mathcal{D}_n$ définie comme :

$$\mathcal{H}_n(X) := -\int p_{X|\mathcal{D}_n}(x)\log(p_{X|\mathcal{D}_n}(x))dx. \tag{6.1}$$

Définition 6.1. Fonction d'acquisition de PESMO

$$\begin{aligned} J_n(x) &= \mathcal{I}_n(Y(x); \mathcal{X}^{sub}) \\ &= \underbrace{\mathcal{H}_n[Y(x)]}_{\text{0) Entropie de la loi conditionnelle à } \mathcal{D}_n} - \underbrace{\mathbb{E}_{\mathcal{X}\sim\mathbb{P}_{\mathcal{X}^{sub}|\mathcal{D}_n}}\left[\mathcal{H}_n\left[Y(x)|\mathcal{X}\right]\right]}_{\text{1) Générer}\mathcal{X}^{(s)} \quad \text{2) Approximer } \mathcal{H}_n[Y(x)|\mathcal{X}]} \end{aligned} \tag{6.2}$$

Cette approche a été étudiée plus tôt pour de l'optimisation globale pour des problèmes mono-objectif (Hernández-Lobato *et al.*, 2014). Les auteurs se focalisent sur la mise en oeuvre et l'implémentation des approximations pour le calcul de la fonction d'acquisition. L'avantage de cette écriture de l'information mutuelle tient dans le fait que les approximations pour le calcul de (6.2) peuvent être réalisées sans échantillonner selon la loi des points non dominés. En effet, la définition (5.3) de $\mathcal{X}^{sub}$ donne une décomposition en facteurs de la probabilité des solutions Pareto-optimales. Celle-ci peut être exploitée par la technique d'approximation bayésienne *Expectation Propagation* (section 7). Ce type d'approximation offre selon les auteurs une meilleure garantie par rapport aux approximations nécessaires pour le calcul de (5.5). Cela a été montré en comparant les performances dans le cadre de l'optimisation globale mono-objectif des techniques d'*Entropy Search* (Dutrieux *et al.*, 2015; Hennig et Schuler, 2012) et de celles de *Predictive Entropy Search* (Hernández-Lobato *et al.*, 2014).

La fonction d'acquisition de PESMO est définie essentiellement par le méta-modèle de krigeage. Le terme 0) s'obtient aisément comme l'entropie d'une loi normale *a posteriori* par krigeage. La difficulté d'implémentation de PESMO vient du second terme de (6.2). En effet, il pose deux difficultés :

- l'approximation de l'intégrale sous la loi $\mathbb{P}_{\mathcal{X}^{sub}|\mathcal{D}_n}$;
- la loi du processus conditionné à la réalisation d'ensemble de points dominés $\mathcal{X}$.

6.2 Génération d'échantillons d'ensembles de points non dominés

Pour calculer (6.2) et en particulier l'intégrale, on procède au calcul d'une approximation par une méthode de Monte-Carlo avec échantillonnage selon la loi $\mathbb{P}_{\mathcal{X}^{sub}|\mathcal{D}_n}$:

$$\mathbb{E}_{\mathcal{X}\sim\mathbb{P}_{\mathcal{X}^{sub}|\mathcal{D}_n}}\left[\mathcal{H}_n\left[Y(x)|\mathcal{X}\right]\right] \approx \frac{1}{S}\sum_{s=1}^{S}\mathcal{H}_n\left[Y(x)|\mathcal{X}^{(s)}\right]. \tag{6.3}$$

Dans un premier temps, il s'agit de générer $\mathcal{X}^{(s)}$ des ensembles de points non dominés. L'algorithme PESMO présenté dans l'article propose de générer des ensembles de points non dominés à la place d'ensembles de Pareto associés aux méta-modèles. C'est en fait une considération pratique dans le cadre de l'optimisation sur un espace d'entrée non fini. Dans cette situation, des ensembles de points non dominés à cardinal fixe sont générés par des algorithmes d'optimisation multi-objectif comme l'algorithme génétique NSGA-II. Dans notre cas, à une trajectoire donnée du méta-modèle, on peut trouver l'ensemble de Pareto associé à sa moyenne de krigeage de façon exacte. Ainsi, nous proposons pour cette étape de générer des ensembles de Pareto $\mathcal{X}^{(s)}$, associées respectivement à trajectoires $(f^{(s)})_{s\in[\![1,S]\!]}$ indépendantes identiquement distribuées selon $\mathbb{P}_{F|\mathcal{D}_n}$. Si l'espace d'entrée est non fini, la méthode de *Random Kernel Features* pour la génération d'approximation de trajectoires a un intérêt (voir annexe C). Néanmoins, dans un contexte très bruité, quelques premiers essais ont montré que cette méthode n'était pas très précise en terme de prédiction de moyenne, même s'il faudrait tester en pratique l'impact sur les performances des algorithmes d'optimisation implémentés. Pour générer selon la loi *a posteriori* $\mathbb{P}_{F|\mathcal{D}_n}$, il faut tout d'abord estimer les paramètres de covariance, ce qui peut se faire par maximum de vraisemblance restreint à partir des données (voir annexe A.3). Chaque trajectoire est générée à partir de paramètres de covariance fixes. On se différencie donc de l'approche de l'article, qui prône une méthode entièrement bayésienne où on échantillonne S paramètres de covariance $\alpha^{(s)}$ selon une loi *a posteriori* de façon à générer les trajectoires $f^{(s)}$. Les variables latentes associées aux paramètres de covariance sont marginalisées, donc l'approximation de l'intégrale requiert un nombre important d'échantillons. Pour la méthode de génération d'échantillons à paramètre de covariance fixe, il n'y a pas de recommandation particulière sur le nombre de trajectoires (et d'ensembles de Pareto) à générer.

Remarque 6.1. Il est parfois nécessaire d'ajouter un paramètre de régularisation à la matrice de covariance K_n de la loi *a posteriori* à une itération donnée n pour pouvoir générer les trajectoires selon la loi *a posteriori*.

Il faut par la suite, déterminer à $\mathcal{X}^{(s)}$ fixé, la loi $\mathbb{P}_{Y(x)|\mathcal{D}_n,\mathcal{X}^{(s)}}$, de manière à pouvoir calculer l'entropie de 2).

6.3 Loi conditionnelle aux points non dominés

Nous supposons que l'ensemble aléatoire des points non dominés est indépendant de $\mathcal{D}_n$ sachant f. La densité pour la loi conditionnelle par rapport à la mesure de Lebesgue est donnée par :

$$\begin{aligned}
p(y(x)|\mathcal{D}_n,\mathcal{X}) &= \int p(y(x)|f(x))\underbrace{p(f(x)|\mathcal{D}_n,\mathcal{X})}_{\text{à calculer}}df(x) \quad \text{par la formule de Bayes avec conditionnement} \\
&= p(y|f(x))\int \underbrace{p((\tilde{f},f(x))|\mathcal{D}_n,\mathcal{X})}_{\text{loi conditionnelle}}df \quad \text{avec } \tilde{f} = \left[f(x^{'})\right]_{x^{'}\in\mathbb{X}\setminus\{x\}} \\
&= p(y(x)|f(x))\int p(f|\mathcal{D}_n,\mathcal{X})df \quad \text{car } f=[\tilde{f},f(x)] \\
&\propto \int p(y(x)|f(x))p(f,\mathcal{X}|\mathcal{D}_n)df \quad \text{loi jointe continue-discrète par la formule de Bayes} \\
&= p(y|f(x))\int \mathbb{P}(\mathcal{X}^{sub}=\mathcal{X}|\mathcal{D}_n,f)p(f|\mathcal{D}_n)df \quad \text{par la formule de Bayes sachant } \mathcal{D}_n \\
&= p(y(x)|f(x))\int \underbrace{\mathbb{P}(\mathcal{X}^{sub}=\mathcal{X}|f)}_{\text{produit de facteurs}}\underbrace{p(f|\mathcal{D}_n)}_{\text{krigeage}}df \text{ en supposant } \mathcal{X}^{sub}\perp\!\!\!\perp \mathcal{D}_n \mid f.
\end{aligned} \tag{6.4}$$

Par commodité, on indice x^* à partir de 1 sans tenir compte de l'indexation sur $\mathbb{X}$. Soit M la taille de l'échantillon d'ensemble de points non dominés, les facteurs expriment des contraintes sur la trajectoire :

$$\boxed{\psi_{i,j}(f) = 1 - \prod_{k=1}^{r}\mathbb{1}_{\{f_k(x_i^*)-f_k(x_j^{'})\geq 0\}} \text{ pour } (i,j)\in[\![1,M]\!]\times[\![1,N]\!]} \tag{6.5}$$

On a par (5.3) la loi de l'ensemble aléatoire de points non dominés qui suit :

$$\mathbb{P}(\mathcal{X}^{sub}=\mathcal{X}\mid F=f) = \frac{1}{|\mathcal{P}(\mathcal{X}^*)|}\prod_{(i,j)\in[\![1,M]\!]\times[\![1,N]\!]}\psi_{i,j}(f). \tag{6.6}$$

Le loi *a posteriori* conditionnelle aux points non dominés (6.4) s'écrit alors :

$$p(f|\mathcal{D}_n,\mathcal{X}) \propto p(f|\mathcal{D}_n)\prod_{(i,j)\in[\![1,M]\!]\times[\![1,N]\!]}\psi_{i,j}(f). \tag{6.7}$$

Il y a deux difficultés qui apparaissent avec la loi de $\mathbb{P}_{F|\mathcal{D}_n,\mathcal{X}}$:

- elle dépend de plusieurs processus gaussiens ;
- c'est une loi sachant des contraintes d'inégalité sur des couples de variables aléatoires de ces processus.

Comme les processus gaussiens sont supposés mutuellement indépendants, il est naturel d'approximer la densité par rapport à la mesure de Lebesgue par un produit de r distributions :

$$p(f|\mathcal{D}_n,\mathcal{X}) \approx q(f) = \prod_{k=1}^{r} q_k(f_k). \tag{6.8}$$

Remarque 6.2. La décomposition en facteurs (6.6) ne se prête pas à la séparabilité en k groupes de facteurs du produit. L'hypothèse d'indépendance est encore une fois supposée par souci de simplicité.

La loi du processus gaussien déplié sous contraintes d'inégalités est approximée, via la technique d'*Expectation Propagation*, par une loi normale multivariée, dont on sait calculer l'entropie différentielle. On pourra considérer d'abord l'exemple intermédiaire qui suit.

7 *Expectation Propagation* pour processus gaussien contraint

On considère dans un premier temps l'exemple suivant. On observe un seul processus F, sur une grille de points en 1D $\mathbb{X}$ de taille N. En fait dans les applications que nous allons faire, nous pouvons considérer le processus gaussien comme un vecteur gaussien. Dans cet exemple, nous choisissons des points de contraintes $x_1 < x_2 < \ldots < x_{l+1}$. L'*Expectation Propagation* vise alors à approximer la loi du processus F sachant $F(x_1) < F(x_2) < \ldots < F(x_{l+1})$ que l'on notera P. La loi *a priori* correspond ici à la loi du processus gaussien que l'on note p. Il en suit cette écriture de la loi *a posteriori* conditionnelle aux contraintes :

$$p(f|f(x_1) < f(x_2) < \ldots < f(x_{l+1})) \propto p(f) \prod_{i=1}^{l} \mathbb{1}_{\{f(x_i)<f(x_{i+1})\}}. \tag{7.1}$$

C'est une loi normale tronquée multivariée. Pour cette décomposition, on appelle facteurs :

$$\phi_i(f) = \mathbb{1}_{\{f(x_i)-f(x_{i+1})<0\}}. \tag{7.2}$$

L'*Expectation Propagation* procède de façon itérative à mettre à jour des approximations de facteurs $\tilde{\phi}_i(f)$. L'approximation de la loi *a posteriori* est notée q avec :

$$q(f) = p(f) \prod_{i=1}^{l} \underbrace{\tilde{\phi}_i(f)}_{\text{approximation de facteur}}, \tag{7.3}$$

où $\tilde{\phi}_i$ est une densité dans une famille de lois. L'ordre des facteurs n'est pas unique, et peut avoir un impact sur les performances de l'*Expectation Propagation*. Dans le cadre du stage, on étudie essentiellement ces propriétés pour l'implémentation de PESMO. L'ordre d'indexation des facteurs est donc fixé par la définition du problème. La famille de lois, que l'on note $\mathcal{F}$ doit vérifier que le produit de densités est bien une densité. Les familles exponentielles vérifient cette propriété. Partant de cette famille de lois, le but de l'*Expectation Propagation* est de donner une heuristique pour trouver la loi correspondant à $\operatorname{argmin}_{Q\in\mathcal{F}} D_{KL}(P \,\|\, Q)$.

7.1 Famille exponentielle

Définition 7.1. Famille exponentielle
Un ensemble $\mathcal{F}$ de lois de densités

$$\begin{aligned} p(x|\Theta) &= \exp(\langle\theta, T(x)\rangle_{\mathcal{F}} - \Omega(\theta)), \theta \in \Theta \\ \Omega(\theta) &= \log \int \exp(\langle\theta, T(x)\rangle_{\mathcal{F}}) d\lambda(x), \end{aligned} \tag{7.4}$$

par rapport à la mesure de Lebesgue λ est appelé famille exponentielle. Ici, θ est appelé paramètre naturel. Θ est l'espace des paramètres naturels, T est la statistique exhaustive, et $\Omega(\theta)$ est la log-fonction de partition. On exprime le produit scalaire $\langle\cdot|\cdot\rangle_{\mathcal{F}}$ à définir en fonction de l'espace des paramètres naturels. Les paramètres de moments sont $\eta = \mathbb{E}_\theta[T(X)]$ où l'espérance désigne le couple de l'espérance des vecteur et matrice aléatoire.

Cette reparamétrisation, issue de Seeger (2008) permet d'avoir la propriété suivante :

Remarque 7.1. Un produit de densité dans $\mathcal{F}$ est une densité non normalisée dans $\mathcal{F}$:

$$\prod_{i=1}^{l} p(x|\theta_i) = p(x|\sum_{i=1}^{l}\theta_i), \tag{7.5}$$

du moment que $\sum_{i=1}^{l}\theta_i \in \Theta$.

Définition 7.2. Paramètres naturels et moments pour gaussiennes

Une loi normale multivariée $\mathcal{N}(\mu,\Sigma)$ de $\mathbb{R}^N$ est une loi de la famille exponentielle $\mathcal{F}$ de paramètres naturels :

$$\theta = (\Sigma^{-1}\mu, \Sigma^{-1}) = (m,\Lambda) \in \Theta, \quad \Theta = R^N \times \mathbb{S}_N^{++}. \tag{7.6}$$

La statistique exhaustive pour $\mathcal{F}$ est : $T(x) = (x, -1/2xx^\top)$ et $\eta = (\mu, -1/2(\Sigma + \mu\mu^\top))$. Le produit scalaire désigne l'application $\langle\cdot|\cdot\rangle_{\mathcal{F}} : (\theta_1,\theta_2) \longmapsto (m_1^\top m_2, \Lambda_1 \odot \Lambda_2)$, le produit de Hadamard entre deux matrices semi-définies positives. Comme une loi marginale d'une gaussienne est une gaussienne, on peut étendre $\mathcal{F}$ aux lois multivariées pour toute dimension $d \leq N$.

Définition 7.3. Famille exponentielle tiltée

Si $\mathcal{F}$ est une famille exponentielle de paramètres naturels θ, et g est une fonction positive telle que le logarithme de la constante de normalisation $\Omega_g(\theta) = \log \int g(x) exp(\theta^\top T(x))dx + \Omega(\theta)$ est finie pour tout $\theta \in \Theta$, alors la famille exponentielle tiltée $\mathcal{F}_g$ induite par g de $\mathcal{F}$ contient les densités :

$$p_g(x|\theta) = \exp(\langle\theta|T(x)\rangle_{\mathcal{F}} - \Omega_g(\theta)) \text{ pour } \theta \in \Theta, \tag{7.7}$$

par rapport à la mesure modifiée $d\lambda_g = g(x)d\lambda(x)$.

Cette famille est une famille exponentielle "décalée" de $\mathcal{F}$, car pour un paramètre naturel θ, les moments associés η et η_g sont distincts. Dans l'exemple, on considère les familles exponentielles tiltées $\mathcal{F}_{\phi_i}$ pour $i = 1,\dots,l$ qui sont des sous-familles des lois normales tronquées multivariées. Comme elles se ramènent à considérer des lois normales tronquées univariées, les constantes de normalisation de ces lois ont une expression analytique. Soit $\check{Q} \in \mathcal{F}$, on note $\check{Q} = \mathcal{N}(\check{\mu},\check{\Sigma})$ et $\check{\theta}$ les paramètres naturels de Q. Notons $Z_i(\check{\mu},\check{\Sigma}) = \int g(x) exp(\check{\theta}^\top T(x))dx$. Soit $\check{P}^{+i}$ de densité $p_{\phi_i}(\cdot|\check{\theta})$ dans $\mathcal{F}_{\phi_i}$ pour $i \in [\![1,l]\!]$, leurs paramètres de moments d'ordre 1 et 2 associés à $\check{\theta}$ se calculent en fonction des dérivées d'ordre 1 et 2 de la log-fonction de partition à l'aide des identités exponentielles :

$$\boxed{\mathbb{E}_{X\sim\check{P}^{+i}}[X] = \frac{\partial \log \Omega_{\phi_i}}{\partial m} = \frac{\partial \log \Omega(\check{\theta})}{\partial m} + \frac{\partial \check{\mu}}{\partial m}\Big|_{m=\check{\Sigma}^{-1}\check{\mu}} \frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}} = \check{\mu} + \check{\Sigma}\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}}, \tag{7.8}$$

$$\begin{aligned} Cov_{X\sim\check{P}^{+i}}(X) &= \mathbb{E}_{X\sim\check{P}^{+i}}\left[XX^\top\right] - \mathbb{E}_{X\sim\check{P}^{+i}}[X]\,\mathbb{E}_{X\sim\check{P}^{+i}}[X]^\top \\ &= -2\frac{\partial \log Z_i}{\partial \Lambda} - \left(\check{\mu} + \check{\Sigma}\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}\right)\left(\check{\mu} + \check{\Sigma}\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}\right)^\top = -2\frac{\partial \log \Omega(\check{\theta})}{\partial \Lambda} - 2\frac{\partial \log Z_i}{\partial \Lambda} \\ &= \check{\Sigma} + \check{\mu}\check{\mu}^\top + 2\Sigma\frac{\partial \log Z_i}{\partial \Sigma}\Big|_{\Sigma=\check{\Sigma}}\Sigma - \check{\mu}\check{\mu}^\top - \check{\Sigma}\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}^\top\check{\Sigma} - 2\check{\mu}\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}^\top\check{\Sigma} \end{aligned}$$

$$\boxed{Cov_{X\sim\check{P}^{+i}}(X) = \check{\Sigma} - \check{\Sigma}\left(\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}} \frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\check{\mu}}^{\top} \Sigma - 2\frac{\partial \log Z_i}{\partial \Sigma}\Big|_{\Sigma=\check{\Sigma}}\right)\check{\Sigma}}. \tag{7.9}$$

A partir de la définition de $\mathcal{F}$, on peut définir une famille de lois à laquelle les approximations de facteurs appartiennent.

Définition 7.4. Famille exponentielle non normalisée

Définissons $\mathcal{F}^U$ la famille exponentielle non normalisée associée à $\mathcal{F}$ dont les éléments ont pour densité :

$$p^U(x|\theta) = \exp(\langle\theta|T(x)\rangle_{\mathcal{F}}),\ \theta = \theta_1 - \theta_2 \text{ avec } (\theta_1, \theta_2) \in \Theta^2, \tag{7.10}$$

par rapport à la mesure de Lebesgue.

Les lois de cette famille n'ont pas de densité de probabilité en général. En effet, $\mathbb{S}_N^{++}(\mathbb{R})$ n'est pas stable pas soustraction. Pour qu'une densité soit définie, il faut que $\Lambda \in \mathbb{S}_N^{++}(\mathbb{R})$, et donc $\Sigma \in \mathbb{S}_N^{++}(\mathbb{R})$.

Soient θ les paramètres naturels associés à la loi Q, θ_0 ceux associés à l'*a priori* gaussien et $(\tilde{\theta}_i)_{i\in[\![1,n]\!]}$ ceux associés aux approximations de facteurs, l'invariant de boucle (7.3) se traduit comme une relation entre ces paramètres naturels :

$$\theta = \theta_0 + \sum_{i=1}^{l} \tilde{\theta}_i. \tag{7.11}$$

7.2 Principe d'*Expectation Propagation* pour des probabilités gaussiennes

Même si les approximations de facteurs sont dans $\mathcal{F}^U$, tant que la somme est bien dans Θ, la densité associée au paramètre θ est aussi dans $\mathcal{F}$. Q est l'approximation gaussienne que l'on peut renormaliser, de densité q :

$$q(f) \propto \exp\left(-\frac{1}{2}(f-\mu)^{\top}\Sigma^{-1}(f-\mu)\right). \tag{7.12}$$

On l'appelle loi globale, car elle incorpore tous les facteurs dans l'expression de sa densité. Le but de la méthode est de mettre à jour ces paramètres naturels, et donc la loi Q. On initialise les approximations de facteurs de manière à ce qu'à l'itération 0, q soit la densité q_0. Par (7.11), il suffit d'initialiser $\tilde{\theta}_i = (0,0)$, ce qui revient à donner une moyenne indéterminée et une variance infinie pour chacune des approximations de facteurs. Il est à noter que les approximations de facteurs vérifient dès l'initialisation $\tilde{\phi}_i \in \mathcal{F}^U$ pour tout i. Le principe d'*Expectation Propagation* est le suivant :

Algorithme 1 : Algorithme EP

Données : Loi initiale $Q = \mathcal{N}(\mu_0, \Sigma_0)$
Résultat : Approximation gaussienne $\mathcal{N}(\mu, \Sigma)$

```
t ← 0
tant que non convergence faire
    /* Itération d'Expectation Propagation */
    pour i = 1, ..., l faire
        Mettre à jour l'approximation de facteur φ̃_i à partir de Q
        Mettre à jour l'approximation globale Q par l'invariant de boucle (7.3)
    fin pour
    t ← t + 1
fin tq
```

Les étapes de mise à jour d'une approximation de facteur impliquent de déterminer des lois intermédiaires que l'on nomme loi cavité, loi tiltée (Figure 11).

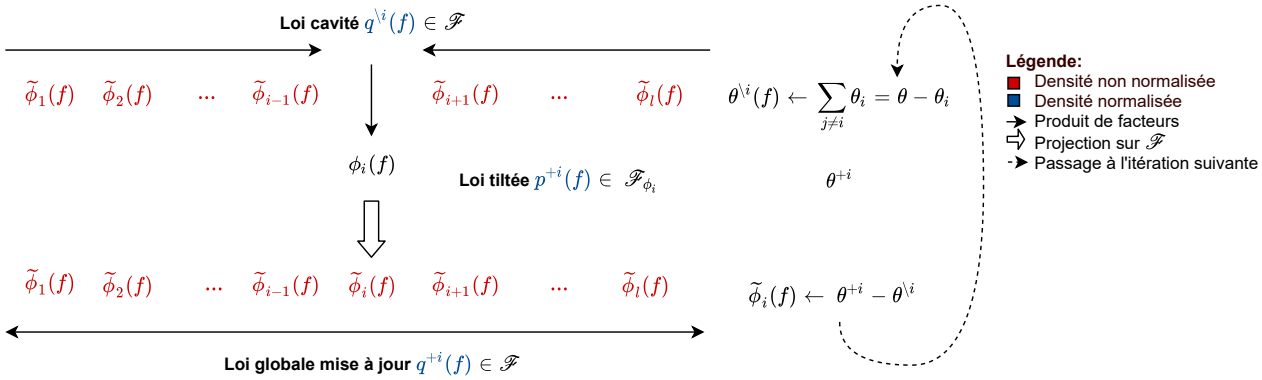


FIGURE 11 – Principe d'une mise à jour de facteur $\tilde{\phi}_i$

Algorithme 2 : Une mise à jour d' approximation de facteur dans *Expectation Propagation*

Données : Approximations de facteurs $\tilde{\phi}_i$, loi globale Q
Résultat : Nouvelle approximation de facteur $\tilde{\phi}_i$

1 Enlever l'approximation de facteur de $\tilde{\phi}_i(f)$ du produit $q(f)$, puis renormaliser si possible,

$$q^{\setminus i}(f) \propto q(f)/\tilde{\phi}_i(f).$$

2 Déterminer les paramètres de moments de la loi tiltée P^{+i} de densité

$$p^{+i}(f) = q^{\setminus i}(f)\phi_i(f).$$

3 **si** $|Cov_{X \sim P^{+i}}(X)| > 0$ **alors**
4 Déterminer la loi projetée $Q^{+i} \in \mathcal{F}$ ayant les mêmes premier et second moments, (μ^{+i}, Σ^{+i}) que P^{+i}.
5 Mettre à jour l'approximation de facteur

$$\tilde{\phi}_i(f) = q^{+i}(f)/q^{\setminus i}(f).$$

6 **fin si**

Les densités de la loi cavité $Q^{\setminus i}$ et de la loi globale Q sont des produits de gaussiennes, donc elles appartiennent à $\mathcal{F}$ à une renormalisation près. Dans la forme standard d'EP, rien n'indique que la loi $Q^{\setminus i}$ appartienne à $\mathcal{F}$. Si ce n'est pas le cas, l'algorithme ne peut en principe pas mettre à jour l'approximation de facteur. En pratique, on procède à la mise à jour d'approximation de facteur tant que les paramètres de la loi cavité ne mènent pas à des problèmes numériques dans les étapes suivantes de l'algorithme 2. L'étape-clé est la projection de la loi $\hat{P}^{+i}$ sur $\mathcal{F}$. Celle-ci consiste à trouver une loi Q^{+i} qui vérifie :

$$Q^{+i} = \underset{Q \in \mathcal{F}}{\operatorname{argmin}} D_{KL}(P^{+i} \,\|\, Q). \tag{7.13}$$

Comme indiqué dans l'annexe B, cela revient à faire correspondre les moments de la statistique exhaustive de $\hat{P}_i$ et Q, i.e. les moments de Q^{+i} d'ordre 1 et 2 sont égaux à ceux de $\hat{P}_i$. La fonction de partition vaut :

$$Z_i = \int \mathbb{1}_{\{f(x_i) - f(x_{i+1}) < 0\}} q^{\setminus i}(f) df. \tag{7.14}$$

Elle s'exprime comme une intégrale selon 2 variables d'intégration. Notons $\bar{Q}^{\setminus i}$, la loi obtenue en marginalisant toutes les variables aléatoires du vecteur gaussien F n'intervenant pas dans le facteur ϕ_i, de densité $\bar{q}^{\setminus i}$. Sa densité par rapport à la mesure de Lebesgue est donnée par :

$$\bar{q}^{\setminus i}([f(x_i) \quad f(x_{i+1})]^\top) = \int q^{\setminus i}(f) df(x_1) \dots df(x_{i-1}) df(x_{i+2}) \dots df(x_N). \tag{7.15}$$

Elle désigne la loi $\mathcal{N}(\bar{\mu}^{\setminus i}, \bar{\Sigma}^{\setminus i})$ dont les paramètres sont des sous-matrices de la loi cavité en dimension pleine. La constante de normalisation Z_i s'exprime en fonction de ceux-ci :

$$\begin{aligned} Z_i &= \int \mathbb{1}_{\{f(x_i)-f(x_{i+1})<0\}} \bar{q}^{\setminus i}([f(x_i) \quad f(x_{i+1})]^\top) df(x_i) df(x_{i+1}) \\ &= \int\int \mathbb{1}_{\{f(x_i)-f(x_{i+1})<0\}} \mathcal{N}\left([f(x_i) \quad f(x_{i+1})]^\top \; ; \; \bar{\mu}^{\setminus i}, \bar{\Sigma}^{\setminus i}\right) df(x_i) df(x_{i+1}). \end{aligned} \tag{7.16}$$

On suppose que : $\bar{\Sigma}^{\setminus i}_{1,1} - \bar{\Sigma}^{\setminus i}_{1,2} \neq \bar{\Sigma}^{\setminus i}_{2,2} - \bar{\Sigma}^{\setminus i}_{2,1}$. On effectue le changement de variable suivant :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -1 \\ a & b \end{bmatrix}}_{A} \times \begin{bmatrix} f(x_i) \\ f(x_{i+1}) \end{bmatrix}, \tag{7.17}$$

avec $a \neq b$ tel que : $A\bar{\Sigma}^{\setminus i}A^\top$ soit diagonale, ce qui équivaut à :

$$a(\bar{\Sigma}^{\setminus i}_{1,1} - \bar{\Sigma}^{\setminus i}_{1,2}) - b(\bar{\Sigma}^{\setminus i}_{2,2} - \bar{\Sigma}^{\setminus i}_{2,1}) = 0. \tag{7.18}$$

Ainsi, le vecteur gaussien

$$\begin{bmatrix} U \\ V \end{bmatrix} = A \begin{bmatrix} F(x_i) \\ F(x_{i+1}) \end{bmatrix} \tag{7.19}$$

vérifie $U \perp\!\!\!\perp V$. U suit une loi normale tronquée univariée dont on sait exprimer la constante de normalisation :

$$Z_i = \Phi\left(\frac{0 - \mathbb{E}[U]}{\sqrt{Var[U]}}\right) = \Phi\left(-\frac{c^\top \bar{\mu}^{\setminus i}}{\sqrt{c^\top \bar{\Sigma}^{\setminus i} c}}\right), \tag{7.20}$$

avec $c = [1 \quad -1]^\top$. Les moments de la loi tiltée marginale $\hat{P}^{+i}$ se calculent à l'aide des formules (7.8) et (7.9) :

$$\begin{aligned} \hat{\mu}^{+i} &= \bar{\mu}^{\setminus i} + \bar{\Sigma}^{\setminus i} \frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\bar{\mu}^{\setminus i}} \\ \hat{\Sigma}^{+i} &= \bar{\Sigma}^{\setminus i} - \bar{\Sigma}^{\setminus i}\left(\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\bar{\mu}_{\setminus i}} \frac{\partial \log Z_i}{\partial \mu}\Big|^\top_{\mu=\bar{\mu}^{\setminus i}} - 2\frac{\partial \log Z_i}{\partial \Sigma}\Big|_{\Sigma=\bar{\Sigma}^{\setminus i}}\right)\bar{\Sigma}^{\setminus i}. \end{aligned} \tag{7.21}$$

En notant $\alpha_i = c^\top \bar{\Sigma}_{\setminus i} c$ et $\beta_i = -\frac{\bar{\mu}_1^{\setminus i} - \bar{\mu}_2^{\setminus i}}{\sqrt{\alpha_i}}$, on obtient :

$$\frac{\partial \log Z_i}{\partial \mu}\Big|_{\mu=\mu^{\setminus i}} = -\frac{\phi(\beta_i)c}{Z_i\sqrt{\alpha_i}} \quad \frac{\partial \log Z_i}{\partial \Sigma}\Big|_{\Sigma=\Sigma^{\setminus i}} = -\frac{\phi(\beta_i)\beta_i c}{Z_i\sqrt{\alpha_i}}, \tag{7.22}$$

d'où :

$$\begin{aligned} \hat{\mu}^{+i} &= \bar{\mu}_{\setminus i} - \frac{\phi(\beta_i)}{Z_i\sqrt{\alpha_i}}\Sigma_{\setminus i}c, \\ \hat{\Sigma}^{+i} &= \bar{\Sigma}_{\setminus i} - \left(\frac{\phi(\beta_i)}{Z_i\alpha_i}\right)\left(\frac{\phi(\beta_i)}{Z_i} + \beta_i\right)\left(\Sigma_{\setminus i}c\right)\left(\Sigma_{\setminus i}c\right)^\top. \end{aligned} \tag{7.23}$$

On se ramène à la loi tiltée P^{+i} en démarginalisant la loi $\hat{P}^{+i}$. Il est à noter que la projection de la loi tiltée sur $\mathcal{F}$ n'est pas tout le temps possible, vu que la matrice de covariance pour P^{+i} n'est pas nécessairement définie positive. Dans le cas contraire, la mise à jour d'approximation de facteur est simplement rejetée. Sinon, la moyenne et la covariance de la loi tiltée permettent de mettre à jour la loi globale Q par $Q^{new} = Q^{+i}$, et de déterminer la nouvelle approximation de facteur $\tilde{\phi}_i$ en passant par leurs paramètres naturels.

7.3 *Expectation Propagation* en dimension réduite et parallélisation

Comme on peut marginaliser la loi cavité, et par conséquent la loi tiltée, il existe une simplification de la procédure d'origine d'*Expectation Propagation*. En effet, les facteurs ne dépendant que d'un vecteur en dimension 2, les approximations de facteurs peuvent être considérées marginalisées, autrement dit réduites à des lois bivariées de $\mathcal{F}^U$. La loi cavité marginale $\bar{Q}^{\setminus i}$ se déduit de la loi globale marginale $\bar{Q}_i$ qui correspond à $\mathcal{N}(\bar{\mu}^i, \bar{\Sigma}^i)$. Or, l'approximation finale est une loi globale sur l'espace d'entrée. Donc la mise à jour de cette loi globale nécessite de démarginaliser les approximations de facteurs obtenues à l'issue de leur mise à jour. La mise à jour d'une approximation de facteur se déroule de la même manière que dans l'algorithme 2, en considérant les lois cavité et projetée sur $\mathcal{F}$ marginales.

Les notations pour les lois considérées sont rappelées ci-dessous. Dans la version en dimension réduite, les lois intermédiaires interviennent sous leur forme marginale.

Nom de loi	Paramètres de loi	Nom de loi marginale	Paramètres de loi en dimension 2
Loi globale Q	$\mathcal{N}(\mu, \Sigma)$, $\theta = (m, \Lambda)$	$\bar{Q}^i$	$\mathcal{N}(\bar{\mu}^i, \bar{\Sigma}^i)$, $\bar{\theta}^i = (\bar{m}_i, \bar{\Lambda}_i)$
Approximation de facteur i	X	$\tilde{\phi}_i$	$\tilde{\theta}_i = (\tilde{m}_i, \tilde{\Lambda}_i)$
Loi cavité $Q^{\setminus i}$	X	$\bar{Q}_{\setminus i}$	$\mathcal{N}(\bar{\mu}^{\setminus i}, \bar{\Sigma}^{\setminus i})$, $\bar{\theta}^{\setminus i} = (\bar{m}^{\setminus i}, \bar{\Lambda}^{\setminus i})$
Loi tiltée	X	$\hat{P}^{+i}$	$\mathcal{N}(\hat{\mu}^{+i}, \hat{\Sigma}^{+i})$
Projection sur $\mathcal{F}$ de la loi tiltée Q^{+i}	$\mathcal{N}(\mu^{+i} \quad \Sigma^{+i})$, $\theta^{+i} = (m^{+i}, \Lambda^{+i})$	$\bar{Q}^{+i}$	$\mathcal{N}(\bar{\mu}^{+i}, \bar{\Sigma}^{+i})$, $\bar{\theta}^{+i} = (\bar{m}^{+i}, \bar{\Lambda}^{+i})$

La loi cavité marginale s'obtient à partir la loi globale marginale. La marginalisation de $Q = \mathcal{N}(\mu, \Sigma)$ donne :

$$\begin{aligned} \bar{\mu}^i &= C_i^\top \mu = [\mu_i \quad \mu_{i+1}], \\ \bar{\Sigma}^i &= C_i^\top \Sigma C_i = \Sigma_{\{i,i+1\}}, \end{aligned} \tag{7.24}$$

où C_i est la matrice de taille $N \times 2$ dont la première colonne i-ième ligne et la deuxième colonne $i+1$-ième ligne valent 1, et les autres éléments de la matrice 0. Les lois cavité et lois tiltées marginales sont définies comme précédemment. Après une mise à jour dans l'espace de dimension 2, les approximations de facteurs sont démarginalisées. Comme l'algorithme est itératif sur les facteurs, il n'y a qu'une seule démarginalisation à faire qui concerne l'approximation de facteur mise à jour. Le procédé de démarginalisation est expliqué dans l'algorithme 3.

Algorithme 3 : Démarginalisation

Données : Approximations de facteurs en dimension 2 mises à jour de paramètres naturels $(\tilde{m}_i)_{i \in [\![1,n]\!]}$
Résultat : Loi globale Q mise à jour

```
/* Démarginalisation des approximations de facteurs                    */
```
1 $\tilde{m}_i \leftarrow C_i \tilde{m}_i$ pour $i = 1, \dots n$
2 $\tilde{\Lambda}_i \leftarrow C_i \tilde{\Lambda}_i C_i^\top$ pour $i = 1, \dots, n$

```
/* \bar{\theta}_{\setminus 0} = [\bar{m}_{\setminus 0}  \bar{\Lambda}_{\setminus 0}]^\top   */
```
3 $\bar{\theta}_{\setminus 0} \leftarrow \sum_{i=1}^n \tilde{\theta}_i$ // transformation en dimension pleine
4 $\theta \leftarrow \theta_{(0)} + \bar{\theta}^{\setminus 0}$ // Mise à jour de la loi globale

La loi *a priori* n'est pas concernée par la marginalisation, donc elle ne l'est pas non plus par la démarginalisation.

Remarque 7.2. Cette simplification en dimension réduite est équivalente à l'algorithme standard d'*Expectation Propagation*. En effet, les opérations effectuées d'EP n'affectent que les moments des lois marginales. Cela permet de réduire le coût de stockage des paramètres naturels des approximations de facteurs.

En revanche, cette façon de procéder à un coût supplémentaire pour le passage de la loi globale mise à jour à l'itération t, à la loi globale marginale à l'itération $t+1$. A l'issue de la démarginalisation de l'itération t on a les paramètres naturels de la loi globale mise à jour. Or, la marginalisation de la loi globale se fait sur les paramètres de moyenne et de covariance. Contrairement à l'algorithme standard, le passage d'une itération à l'autre n'est pas direct et le coût de ce passage provient de l'inversion de la matrice de précision Λ pour

obtenir Σ de complexité $\mathcal{O}\left(N^3\right)$. Pour rendre moins coûteux l'EP, on peut choisir de mettre à jour la loi globale qu'une fois par itération au lieu de l fois par itération comme décrit dans l'algorithme 4. A une itération t de l'algorithme EP, l'invariant de boucle devient :

$$\theta^{(t)} = \theta_0 + \sum_{i=1}^{l} Dem\left(\tilde{\theta}_i^{(t)}\right), \tag{7.25}$$

où *Dem* est la fonction de démarginalisation d'approximation de facteur.

Algorithme 4 : Algorithme EP (parallélisable)

Données : Loi initiale $Q = \mathcal{N}(\mu_0, \Sigma_0)$
Résultat : Approximation gaussienne $Q = \mathcal{N}(\mu, \Sigma)$

$t \leftarrow 0$
tant que *non convergence* **faire**
  /* Itération d'*Expectation Propagation* */
  Marginaliser $Q^{(t)}$ pour avoir $\bar{Q}^{(t)}$
  pour $i = 1, \ldots, n$ **faire**
    Mettre à jour l'approximation de facteur $\tilde{\phi}_i^{(t)}$ à partir de $\bar{Q}^{(t)}$ pour obtenir $\tilde{\phi}_i^{(t+1)}$
  fin pour
  Démarginaliser pour obtenir $Q^{(t+1)}$
  $t \leftarrow t + 1$
fin tq

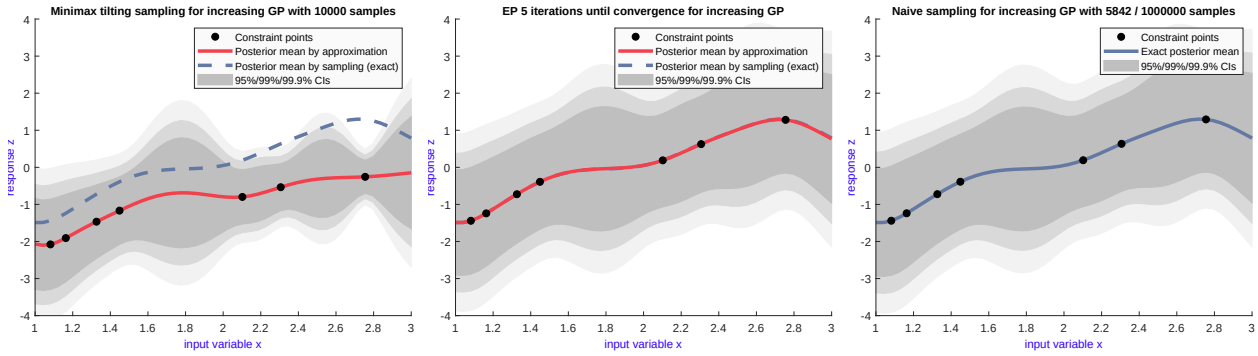
7.4 Tests de EP

Cette façon d'estimer un processus gaussien contraint a été peu ou pas abordée (Lopez lopera, 2019). En général, les articles sur le sujet s'intéressent aussi à des méthodes de génération de tels processus. Nous proposons de tester si l'approximation par *Expectation Propagation* est satisfaisante, c'est-à-dire si la moyenne et matrice de covariance du processus gaussien contraint sont bien estimés. Le processus gaussien *a priori* est défini avec une moyenne nulle et une covariance de Matérn isotropique. L'espace d'indexation du processus est constitué de 50 points régulièrement espacé dans un intervalle donné qui n'a pas d'importance, car le processus est stationnaire. On définit une distance statistique entre deux lois qui est fonction de ces deux premiers moments :

$$d(P, Q) = \max\left(\|\mathbb{E}_{X\sim P}[X] - \mathbb{E}_{X\sim Q}[X]\|_\infty,\ \|Cov_{X\sim P}(X) - Cov_{X\sim Q}(X)\|_\infty\right). \tag{7.26}$$

Dans toute la suite, on supposera sauf mention, que l'algorithme s'arrête dès que : $d(Q^{(t+1)}, Q^{(t)}) < \epsilon_0$, pour un ϵ_0 donné, et adapté au problème. C'est ce critère de convergence qui est utilisé dans l'implémentation de *PESMO*. Il y a d'autres manières de définir le critère de convergence d'*Expectation Propagation* (Seeger, 2008; Cunningham *et al.*, 2011; Vehtari *et al.*, 2014).

L'expérience est réalisée pour différentes répartitions de points de contraintes. Un autre type de méthode est généralement utilisé pour un tel problème, recourant à de l'échantillonnage. Plus particulièrement, la méthode MCMC de slice sampling général (Robert et Casella, 2013, Chapitre 8.2) est utilisé dans le cas où la *loi a posteriori* admet une décomposition en produit. Nous utiliserons néanmoins des méthodes donnant des échantillons *i.i.d*, celle d'échantillonnage naïf qui a un gros taux de rejet, et celle d'échantillonnage par *Minimax Tilting* (Botev, 2016), pour estimer les moments. Cette dernière est présentée comme meilleure que la méthode par échantillonnage par MCMC ou la méthode de Genz qui sont couramment utilisées pour ce type de problèmes. Une méthode MCMC basée sur l'échantillonnage de Gibbs n'a pas du tout donné des résultats concluants sur ces expériences. Pour les résultats, nous avons affiché les variances et moyennes prédictives de l'approximation gaussienne de la loi tronquée multivariée pour chaque méthode. En effet, comme nous le verrons par la suite, il n'est pas utile d'estimer tous les éléments de la matrice de covariance donnée par *Expectation Propagation* pour *PESMO*. Pour l'échantillonnage naïf selon la vraie loi, il faut tirer un très grand nombre de trajectoires pour obtenir suffisamment d'échantillons vérifiant les contraintes. On suppose qu'avec plus de 5000 échantillons retenus pour l'estimation, la méthode est exacte. La méthode d'échantillonnage par *Minimax Tilting* rejette la moitié des échantillons générés. Nous obtenons ainsi les résultats suivants :



(a) Par Minimax Tilting (b) Par EP (c) Par échantillonnage naïf

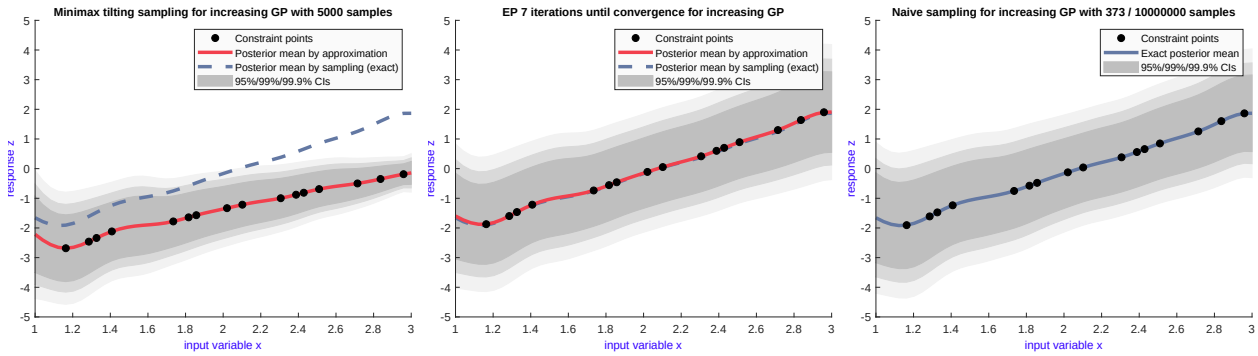
FIGURE 12 – Estimation de la moyenne et variance du processus gaussien contraint pour 7 points de contraintes

Les durées ci-dessous sont données pour montrer les ordres de grandeur de temps de calul de chacune des méthodes avec le nombre d'échantillons générés pour les méthodes d'échantillonnage. L'estimation par EP est bien plus rapide que l'estimation par les autres méthodes.

EP	*Minimax Tilting* - 100	Minimax Tilting - 5000	Echantillonnage naïf - 587	Echantillonnage naïf - 5842
0.16	0.39	0.53	0.20	0.90

TABLE 1 – Durée d'estimation en secondes pour 7 points de contraintes

Il s'avère qu'en générant 5×10^5 trajectoires de manière à garder plus de 3000 échantillons, la distance entre l'estimateur de moyenne par échantillonnage naïf et celle par *Expectation Propagation* est plus grande qu'avec le nombre de trajectoires indiquées sur la figure 12c. Il est à noter, que contrairement aux méthodes d'échantillonnage, les estimateurs donnés par *Expectation Propagation* sont déterministes, ne dépendant que du paramètre de la loi normale multivariée non contrainte et de l'expression des facteurs. Plus le nombre de points de contraintes est grand, et moins il y aura d'échantillons retenus vérifiant toutes les contraintes.



(a) Par Minimax Tilting (b) Par EP (c) Par échantillonnage naïf

FIGURE 13 – Estimation de la moyenne et variance du processus gaussien contraint pour 16 points de contraintes

EP	*Minimax Tilting* - 5000	Echantillonnage naïf - 373
0.26	2.8	72

TABLE 2 – Durée d'estimation en secondes pour 16 points de contraintes

Les méthodes d'échantillonnage sont imprécises ou trop coûteuses.

De plus, pour plusieurs contraintes d'inégalités, et répartitions de points de contraintes, l'*Expectation Propagation* donne un résultat satisfaisant dès la 1ière ou la 2ième itération, même si l'algorithme n'a pas encore convergé. La moyenne prédite par EP se rapproche de la vraie moyenne. La variance prédictive semble, elle aussi similaire d'itération en itération. Il y a donc intérêt à lancer l'algorithme avec un nombre fixe d'itérations, par exemple 5. En revanche, pour inférer la loi *a posteriori* d'un processus gaussien avec une position de minimum donnée, l'algorithme oscille d'une itération à l'autre :

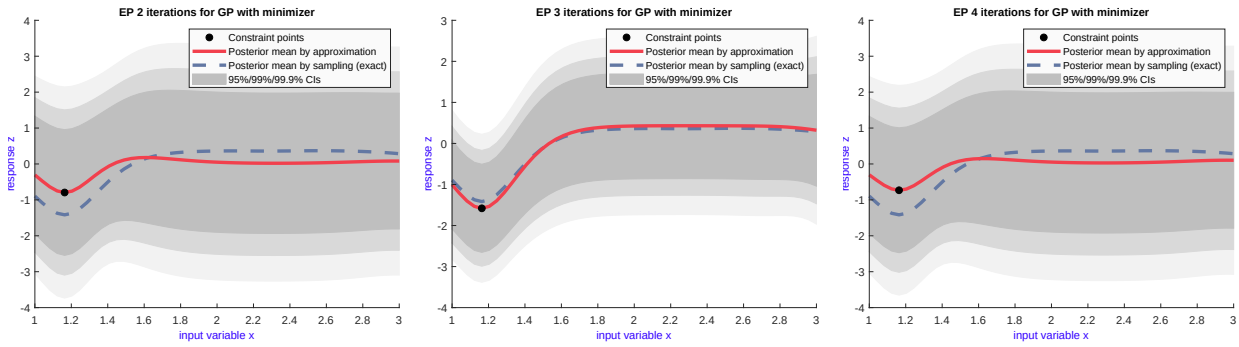


FIGURE 14 – Oscillations lors d'EP

Ce phénomène se produit lorsqu'il y a de la redondance dans les contraintes (Cunningham *et al.*, 2011). La solution à cette instabilité est d'amortir les mises à jour d'approximation de facteur pour tout $i \in [\![1, l]\!]$, c'est-à-dire :

$$\tilde{\theta}_i^{(t+1)} = \delta \tilde{\theta}_i^{new} + (1-\delta)\tilde{\theta}_i^{(t)}. \tag{7.27}$$

Cela revient à avoir $\tilde{\phi}_i^{(t+1)} = \tilde{\phi}_i^{new} \tilde{\phi}_i^{(t)\ 1-\delta}$, et aussi une mise à jour amortie de la loi globale :

$$\theta^{(t+1)} = \delta \theta^{new} + (1-\delta)\theta^{(t)}. \tag{7.28}$$

Pour l'expérience ci-dessus, avec un paramètre $\delta = 0.8$, on parvient à atteindre la convergence. Plus on diminue δ, plus on favorise la convergence. Les articles qui traitent de EP dans d'autres contextes proposent d'amortir dès le début de l'algorithme les mises à jour, et de faire décroître lentement le paramètre δ (Vehtari *et al.*, 2014; Cunningham *et al.*, 2011). L'oscillation sans échec de mise à jour vient d'une déviation de la loi mise à jour, étant donné que l'on implémente une version parallélisable d'*Expectation Propagation* au lieu de l'algorithme 1. Dans toute la suite, nous initialisons le paramètre δ à une valeur inférieure à 1.

Une itération d'*Expectation Propagation* peut échouer à l'étape-clé de projeter la loi tiltée (marginale ou non) sur $\mathcal{F}$ (Vehtari *et al.*, 2014, Appendix A). Il faut s'assurer dans l'implémentation de ne mettre à jour que les approximations de facteurs $\tilde{\phi}_i$ pour lesquels les paramètres naturels de la loi tiltée vérifient $\hat{\Lambda}^{+i} \in \mathbb{S}_2^{++}$. Dans les exemples jouets présentés, ce n'est jamais le cas, donc toutes les mises à jour d'approximations de facteurs peuvent se faire. Lorsque les valeurs des paramètres de covariance (régularité et de portée et écart-type de bruit) sont fixées à des valeurs plus élevées que celles utilisées par défaut, ces problèmes peuvent apparaître.

L'application d'*Expectation Propagation* ne se limite pas à ces problèmes. La loi initiale peut être une loi conditionnelle à des évaluations de fonctions comme dans la suite. En outre, en procédant à une réduction à une dimension comme le fait l'article de Cunningham *et al.* (2011), nous pouvons introduire des contraintes sur le signe du processus en certains points donnés.

Pour visualiser les résultats, on a utilisé un script MATLAB basé sur `stk_plot1d` de la bibliothèque `STK` (Bect *et al.*, 2020). Ces tests permettent de vérifier si l'*Expectation Propagation* est la méthode d'approximation adaptée pour calculer le terme 2).

8 Résultats de l'approximation de la fonction d'acquisition

8.1 *Expectation Propagation* pour PESMO

On cherche à approximer les facteurs par des lois normales bivariées $\tilde{\psi}_{i,j}$ non normalisées. Pour le traitement des facteurs, on considère donc la réduction à un espace de dimension $2 \times r$. Soit la loi marginale $\dagger Q_{i,j}$ de densité

$$\dagger q_{i,j}(f) = Z_{i,j}^{-1} \left(\int p(f|\mathcal{D}_n) df_{\setminus i,j} \right) \prod_{(i,j) \in [\![1,M]\!] \times [\![1,N]\!]} \tilde{\psi}_{i,j}(f), \tag{8.1}$$

on peut se ramener à considérer un sous-facteur par processus gaussien F_k pour $k \in [\![1, r]\!]$ à partir d'un facteur. En utilisant l'indépendance mutuelle des processus, on a la séparabilité suivante :

$$\tilde{\psi}_{i,j}(f) = \prod_{k=1}^{r} \tilde{\phi}_{i,j,k}(f_k), \tag{8.2}$$

avec les paramètres naturels des sous-approximations de facteurs notés $\tilde{m}^*_{i,j,k}$, $\tilde{v}^*_{i,j,k}$, $\tilde{m}_{i,j,k}$, $\tilde{v}_{i,j,k}, \tilde{w}_{i,j,k}$:

$$\boxed{\tilde{\phi}_{i,j}(f_k) = \exp\{-0.5 \left(f_k(x_i^*)^2 \tilde{v}^*_{i,j,k} + f_k(x_j^{'})^2 \tilde{v}_{i,j,k} + f_k(x_i^*) f_k(x_j^{'}) \tilde{w}_{i,j,k} \right) + \tilde{m}_{i,j,k} f_k(x_j^{'}) + \tilde{m}^*_{i,j,k} f_k(x_i^*)\}}.$$

La mise à jour des paramètres se fait aussi sous forme séparée. En effet, on a aussi la forme séparée pour la loi gaussienne $\dagger Q_{i,j}(f)$:

$$\begin{aligned} \dagger q_{i,j}(f) &= \prod_{k=1}^{r} q_{i,j,k}(f_k) \\ \text{avec } q_{i,j}(f_k) &= Z_{i,j}^{-r} \left(\int p(f|\mathcal{D}_n) df_{\setminus i,j} \right) \prod_{(i,j) \in [\![1,M]\!] \times [\![1,N]\!]} \tilde{\phi}_{i,j,k}(f_k). \end{aligned} \tag{8.3}$$

Une mise à jour des facteurs entraîne des mises à jour découplées des sous-approximations de facteurs. L'entropie de r variables aléatoires indépendantes est la somme des entropies de ces variables aléatoires. Comme dans l'exemple, l'*Expectation Propagation* s'applique en réduction de dimension. Les facteurs par marginalisation sont des lois bivariées, ce qui revient à :

$\tilde{\phi}_{i,j,k}(f_k)$ sous-approximations de facteurs (lois bivariées)

avec les moments qui s'expriment en fonction des paramètres naturels :

$$\begin{aligned} \tilde{\Sigma}_{i,j,k} &= \begin{bmatrix} \tilde{v}^*_{i,j,k} & \tilde{w}_{i,j,k} \\ \tilde{w}_{i,j,k} & \tilde{v}_{i,j,k} \end{bmatrix}^{-1}, \\ \tilde{\mu}_{i,j,k} &= [\tilde{\mu}^*_{i,j,k}, \tilde{\mu}^{'}_{i,j,k}]^\top = \tilde{\Sigma}_{i,j,k} [\tilde{m}^*_{i,j,k}, \tilde{m}^{'}_{i,j,k}]^\top. \end{aligned} \tag{8.4}$$

Le déroulé de l'algorithme d'*Expectation Propagation* est plus compliqué que précédemment.

Les lois cavité, tiltée et globale sont exprimées sous leur forme marginale.

Tout d'abord, on retire une approximation de facteur de $\dagger Q$. La loi cavité a pour densité :

$$\dagger q^{\setminus i,j} \propto \frac{\dagger q_{i,j}}{\tilde{\psi}_{i,j}}.$$

Cela revient à retirer pour chaque $k \in [\![1, r]\!]$ les sous-approximations de facteurs de cette loi par (8.3) :

$$q_k^{\setminus i,j} \propto \frac{q_{i,j}}{\tilde{\phi}_{i,j,k}}.$$

Ensuite, un facteur est ajouté à $\dagger q_{\setminus i,j}$ pour obtenir la loi tiltée :

$$\begin{aligned}\hat{p}^{i,j} &\propto \psi_{i,j} \dagger q^{\setminus i,j} \\ &= Z_{i,j}^{-1} \psi_{i,j} \prod_{k=1}^{r} q_k^{\setminus i,j}.\end{aligned}$$

En notant en exposant selon que les éléments de la moyenne et covariance de la loi cavité correspondent à l'indice du point non dominé x_i^* ou à celui du point qui est dominé $x_j^{'}$, on obtient :

$$\begin{aligned}Z_{i,j} &= \int \prod_{k=1}^{r} q^{\setminus i,j}(f_k) df - \int \prod_{k=1}^{r} \mathbb{1}_{\{f_k(x_i^*) - f_k(x_j') \geq 0\}} q^{\setminus i,j}(f_k) df \\ &= \prod_{k=1}^{r} \int q^{\setminus i,j}(f_k) df_k - \int \prod_{k=1}^{r} \mathbb{1}_{\{f_k(x_i^*) - f_k(x_j') \geq 0\}} q^{\setminus i,j}(f_k) df_k \\ &= 1 - \prod_{k=1}^{r} \underbrace{\int \mathbb{1}_{\{\tilde{f}_k(x_i^*) - \tilde{f}_k(x_j') \geq 0\}} q^{\setminus i,j}(f_k) df_k}_{\text{calcul analogue à l'exemple d'EP}} \\ &= 1 - \prod_{k=1}^{r} \Phi\left(\frac{\mu_k^{\setminus i,j*} - \mu_k^{\setminus i,j'}}{\sqrt{\Sigma_k^{\setminus i,j**} + \Sigma_k^{\setminus i,j''} - 2\Sigma_k^{\setminus i,j*'}}} \right) = 1 - \prod_{k=1}^{r} \Phi\left(\frac{\beta_{i,j,k}}{\sqrt{\alpha_{i,j,k}}} \right).\end{aligned} \tag{8.5}$$

On peut ainsi obtenir, si elle est bien définie, la projection sur $\mathcal{F}$ de $\hat{P}^{i,j}$ par *Moment Matching* comme dans la section précédente. La loi projetée admet une densité séparable pour la mise à jour des sous-approximations de facteurs. Il en découle les formules pour les paramètres ces lois projetées bivariées pour chaque k :

$$\begin{aligned}\hat{\mu}_k^{i,j} &= \mu_k^{\setminus i,j} + \Sigma_k^{\setminus i,j} \frac{\partial \log Z_{i,j}}{\partial \mu}\Big|_{\mu=\mu_k^{\setminus i,j}} \\ \hat{\mu}_k^{i,j} &= \mu_k^{\setminus i,j} - \frac{\phi(\beta_{i,j,k})}{\sqrt{\alpha_{i,j,k} Z_{i,j}}} \Sigma_k^{\setminus i,j} c, \\ \hat{\Sigma}_k^{i,j} &= \Sigma_k^{\setminus i,j} - \Sigma^{\setminus i,j} \left(\frac{\partial \log Z_{i,j}}{\partial \mu}\Big|_{\mu=\mu_k^{\setminus i,j}} \frac{\partial \log Z_{i,j}}{\partial \mu}\Big|_{\mu=\mu_k^{\setminus i,j}}^{\top} - 2 \frac{\partial \log Z_{i,j}}{\partial \Sigma}\Big|_{\Sigma=\Sigma_k^{\setminus i,j}} \right) \Sigma_k^{\setminus i,j} \\ \hat{\Sigma}_k^{i,j} &= \Sigma_k^{\setminus i,j} - \left(\frac{\phi(\beta_{i,j,k})}{\alpha_{i,j,k} Z_{i,j}} \right) \left(\frac{\phi(\beta_{i,j,k})}{Z_{i,j}} - \beta_{i,j,k} \right) (\Sigma_k^{\setminus i,j} c)(\Sigma_k^{\setminus i,j} c)^{\top},\end{aligned} \tag{8.6}$$

où $c = [-1\ 1]^{\top}$. La décomposition des approximations facteurs en sous-approximations de facteurs entraîne la séparabilité de la loi tiltée en des lois tiltées par processus.

Remarque 8.1. Il est possible, dans le déroulement d'EP, d'ignorer les facteurs qui n'apporte aucune information, c'est-à-dire les $\psi_{i,i}$ constantes qui valent 1.

Par transformation des paramètres naturels en dimension 2 dans l'espace de dimension N, on obtient les paramètres naturels de Q_k pour $k = 1, \ldots, r$, ce qui donne la loi des processus sachant les contraintes. Les paramètres naturels m_k et Λ_k s'expriment en fonction de $\tilde{m}_{i,j,k}^*$, $\tilde{v}_{i,j,k}^*$, $\tilde{m}_{i,j,k}$, $\tilde{v}_{i,j,k}$, $\tilde{w}_{i,j,k}$ et le paramètre naturel $\theta_{k,0}$ de la loi à l'initialisation d'*Expectation Propagation* en procédant à la démarginalisation (voir section 7). Il n'y a finalement besoin que de calculer les matrices de précision Λ_k pour $k \in [\![1, r]\!]$:

$$\Lambda_k = \Lambda_{k,0} + \sum_{i,j \in [\![1,M]\!] \times [\![1,N]\!]} C_{i,j} \begin{bmatrix} \tilde{v}_{i,j,k}^* & \tilde{w}_{i,j,k} \\ \tilde{w}_{i,j,k} & \tilde{v}_{i,j,k} \end{bmatrix} C_{i,j}^{\top}, \tag{8.7}$$

où $C_{i,j}$ correspond à la matrice de taille $2 \times N$ dont la i-ème colonne de la première ligne vaut 1, ainsi que la j-ième colonne de la 2ième ligne, et dont les autres éléments valent 0. Nous appellerons l'algorithme de PESMO décrit jusqu'ici comme PESMO1.

Algorithme 5 : Fonction d'acquisition de PESMO1 $J_n(\cdot)$ à l'itération n

Données : $\mathcal{D}_n$
Résultat : $\alpha_n(\cdot)$, $\mathcal{X}_*$
Mettre à jour les paramètres de covariance du méta-modèle ν_k, σ_k^2 pour $k = 1, \dots r$
/* Calcul de $\mathcal{H}_n(Y(x))$ */
$J_n(x) \leftarrow 0.5 \sum_{k=1}^{r} (\log[2e(v_{n,k}(x) + \sigma_k^2)] + \log(2\pi e))$ // $v_{n,k}$ variance prédictive sachant $\mathcal{D}_n$
/* Calcul de $\mathbb{E}_Y[\mathcal{H}_n(Y(x)|\mathcal{X}^*)]$ */
Générer des ensembles de trajectoires $f^{(1)}, \dots, f^{(S)}$ selon la loi *a posteriori* des processus gaussiens
pour $s = 1 : S$ **faire**
  Déterminer $\mathcal{X}^{(s)}$ par résolution du problème multi-objectif associé à $f^{(1)}, \dots, f^{(s)}$
  Calculer Σ_k sachant $\mathcal{X}^{(s)}$ par *Expectation propagation* pour $k = 1, \dots, r$
  $v_{n,k}(\cdot|\mathcal{X}^{(s)}) \leftarrow diag(\Sigma_k)$ pour $k = 1, \dots, r$
fin pour
$J_n(x) \leftarrow J_n(x) - \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{r} 0.5(\log[(v_{n,k}(\cdot|\mathcal{X}^{(s)}) + \sigma_k^2)] + \log(2\pi e))$

Une deuxième version de PESMO utilisant l'approximation de l'espace d'entrée expliquée dans l'article (Hernández-Lobato *et al.*, 2016) a été implémentée pour comparer à PESMO1. A une itération n et pour un ensemble de Pareto $\mathcal{X}^{(s)}$ de taille M, elle consiste à appliquer *Expectation Propagation* avec seulement les facteurs concernant que les points $\{x^{(1)}, \dots x^{(n)}\} \cup \mathcal{X}^{(s)}$. L'algorithme d'approximation bayésienne met à jour la loi marginale des processus gaussiens en ces points. Puis, il s'agit de réappliquer pour tout $x \in \mathbb{X} \setminus \mathcal{X}^{(s)}$ *Expectation Propagation* à 1 itération à la loi marginale des processus en $\mathcal{X}^{(s)} \cup x$, ce qui revient à appliquer *Expectation Propagation* sur tout l'espace d'entrée après avoir mis à jour la loi marginale de Q_k concernant les points de $\mathcal{X}^{(s)}$, pour obtenir la loi Q_k^{EP}. L'avantage de procéder ainsi est que l'on peut utiliser la formule d'inversion par bloc pour déterminer les variances prédictives $v_{n,k}(x|\mathcal{X}^{((s))}) = (\Lambda_k)^{EP\,-1}_{M+1,M+1}$ à partir de l'inverse de $(\Lambda_k^{EP})_{\{M,M\}}$. En effet, on peut ignorer les autres éléments de la matrice de précision, car pour l'application d'*Expectation Propagation*, les facteurs ne concernent que les points de $\mathcal{X}^{(s)}$ et x.

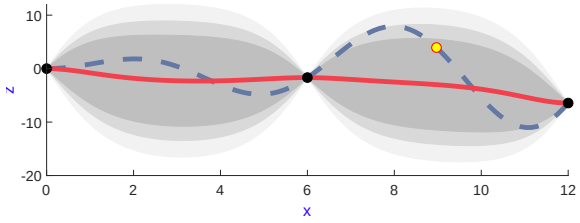
8.2 Tests de PES mono et PESMO

Le langage de programmation est MATLAB. La boîte à outils STK contient toutes les fonctions nécessaires pour construire un méta-modèle de krigeage, mais aussi déterminer des ensembles de Pareto d'une fonction définie sur une grille de points. Nous avons donné une technique d'approximation bayésienne pour calculer la fonction d'acquisition. Il s'agit d'abord de comparer le critère d'échantillonnage avec un calcul exact mais coûteux de l'entropie. Nous procédons avec l'échantillonnage naïf évoqué dans la section 7. Le calcul de l'entropie de ces trajectoires se fait grâce à l'estimateur non paramétrique de l'entropie différentielle par la méthode des k-plus proches voisins ou estimateur KL (Kraskov *et al.*, 2004) :

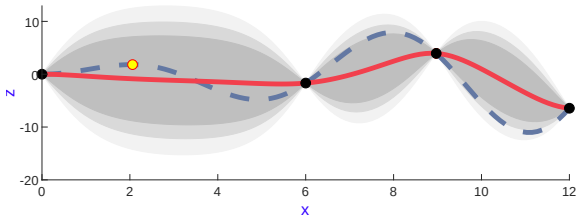
$$\hat{H}_n(Y|\mathcal{X}) = \psi(N_{traj}) - \psi(k) + \log(c_d) + \frac{d}{N_{traj}} \sum_{i=1}^{N_{traj}} \log(\epsilon(i)), \tag{8.8}$$

où $\epsilon(i)$ est la distance entre la i-ème trajectoire et son plus proche k-voisin parmi les N_{traj} trajectoires, les trajectoires s'exprimant comme un vecteur à N coordonnées pour chaque point de l'espace d'entrée, et d correspondant donc à N. On prend $c_d = 2^d$ pour la distance définie comme la norme infinie, $c_d = \pi^{d/2}/\Gamma(1+d/2)$ où Γ est la fonction Gamma pour la norme 2. Il y a d'autres estimateurs basés sur les plus proches voisins décrits dans des articles plus récents. Pour nos expériences, on se restreint à l'estimateur classique.

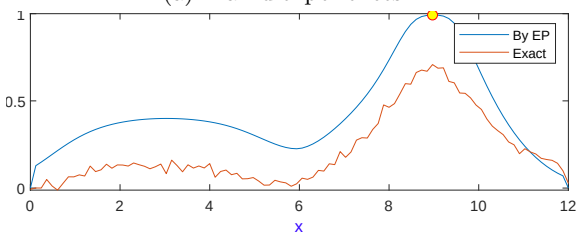
L'algorithme d'optimisation multi-objectif peut être appliqué à un problème mono-objectif. La notion d'ensemble de Pareto est remplacée par celle de position de minimum. Dans ce cas le facteur peut se mettre sous la forme plus simple $\psi_i(f) = \mathbb{1}_{\{f(x^*)<f(x_i)\}}$ pour $i \in [\![1, N]\!]$.



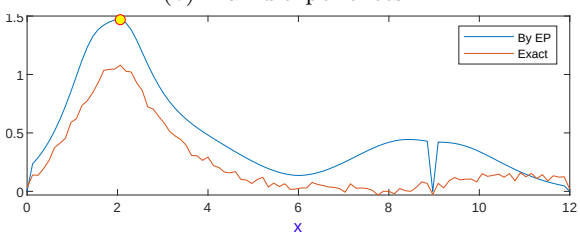
(a) Plan d'expériences 1



(b) Plan d'expériences 2



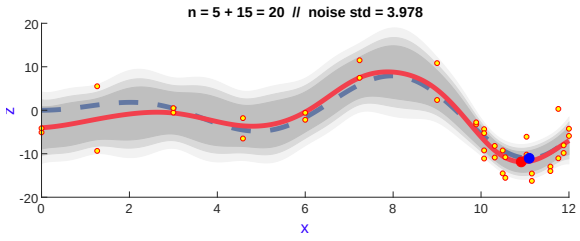
(c) Fonction d'acquisition lors d'une itération de PES



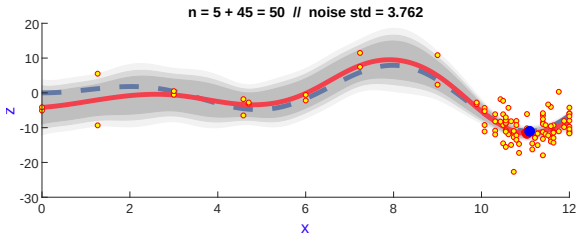
(d) Fonction d'acquisition lors d'une itération de PES

FIGURE 15 – Comparaison du calcul entre l'approximation de la fonction d'acquisition et l'estimation par méthode exacte Monte-Carlo

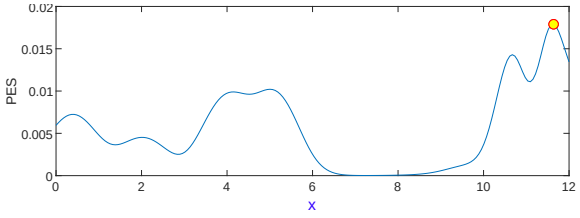
La méthode, dite exacte est réalisée avec au moins 100 trajectoires conditionnées à la position de minimum. On remarque que plus on a sélectionné de points, et plus la génération de trajectoires conditionnées à ces points est longue. Par ailleurs, avec une grosse variance de bruit par rapport aux valeurs des fonctions objectif, l'estimateur d'entropie s'avère mauvais, les valeurs d'information mutuelle obtenues étant négatives. Dans un contexte très bruité, la fonction d'acquisition de PES mono a une multitude de modes :



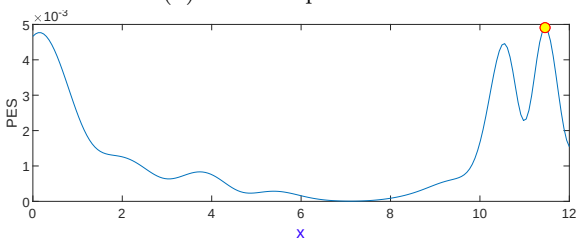
(a) Plan d'expériences 1



(b) Plan d'expériences 2



(c) Fonction d'acquisition lors d'une itération de PES



(d) Fonction d'acquisition lors d'une itération de PES

FIGURE 16 – Valeurs de la fonction d'acquisition à différentes itérations de l'algorithme

Ce critère d'échantillonnage se différencie des autres, notamment *AKG*, car la fonction d'acquisition calculée ne consiste pas en des pics. Très souvent, il y a deux modes assez rapprochés autour de la position de minimum. Quelques expériences sur des problèmes mono-objectifs dans un contexte bruité ont eté menés, avec à première vue des bonnes performances en comparaison avec le critère d'échantillonnage d'*Expected Improvement* et *AKG*. Il s'avère que l'algorithme explore bien l'espace d'entrée, au moins sur des exemples simples comme ci-dessus. C'est aussi le cas pour les problèmes bi-objectifs étudiés dans le stage. Ces observations sur l'exploration de l'algorithme nous conforte dans l'idée que le critère d'échantillonnage est adapté au cas bruité.

Remarque 8.2. C'est une version différente de l'algorithme PES Hernández-Lobato *et al.* (2014) qui considère les facteurs d'une décomposition de la loi conditionnelle aux minimiseurs comme dépendant des gradients et hessiennes de processus gaussien.

Pour un problème multi-objectif, la méthode naïve consiste à tirer des trajectoires et ne garder que celles qui ont ces solutions Pareto-optimales. Pour cette méthode, l'implémentation actuelle présente des bugs de conditionnement de matrice au moment de réaliser une mise à jour d'EP, surtout à l'initialisation quand peu de points ont été évalués. La méthode d'échantillonnage naïve nécessite de générer plusieurs centaines de milliers de trajectoires afin d'obtenir un nombre suffisant de trajectoires qui vérifient la condition d'avoir les solutions Pareto-optimales fixées par PESMO. Dans mes expériences, l'estimateur d'entropie semble surestimer de beaucoup la fonction d'acquisition. Le protocole de vérification est qualitativement meilleur que celui présenté dans l'article *Predictive Entropy Search for Multi-Objective Bayesian Optimization*, car il propose de générer des trajectoires exactes au lieu d'approximations de trajectoires par *Random Kernel Features*. Néanmoins, on n'a pas pu vérifier si la fonction d'acquisition est bien approximée pour des problèmes bi-objectifs.

Or, pour PESMO appliqué à certains problèmes, les étapes de l'EP coincent car, pour une partie des facteurs $\tilde{\phi}_i$, leur densité $q^{\setminus i}$ n'est pas normalisable. Pour résoudre cela, plusieurs stratégies peuvent être mises en place. Une d'entre elles, proposée dans Vehtari *et al.* (2014) est de faire décroître rapidement δ si l'algorithme bloque au niveau du calcul des paramètres de la loi cavité. Les paramètres s'obtiennent alors grâce à la loi globale dont la mise à jour a été fortement amortie. Cette technique n'a pas prouvé ses effets dans nos applications. De même, procéder en enlevant partiellement l'approximation de facteur de la loi globale pour réduire la variance des lois cavités (Seeger, 2008, Section 2.3) n'a pas réglé les problèmes de mises à jour.

Finalement, il a fallu traiter à part, les approximations de facteurs concernées par les loi cavités non normalisables en ne les mettant pas à jour ou en forçant leurs paramètres naturels à une valeur fixe, 0 pour imposer une variance infinie. Procéder à la première ou la seconde stratégie ne change pas significativement les résultats. Le but de ces stratégies est de débloquer une itération d'*Expectation Propagation* en espérant que les matrices de covariances des lois cavités soient définies positives et permettent la normalisation dans les itérations suivantes. Nous nous assurons aussi que le pourcentage de mise à jour au cours d'une itération ne soit pas trop bas. Après plusieurs expériences sur les problèmes d'optimisation en local, le pourcentage tombe très rarement en-dessous de 40%, et est régulièrement au-dessus de 70%. Il est à noter, qu'ignorer les mises à jour d'approximation, revient à négliger certains facteurs durant les itérations d'EP, ce qui est acceptable pour l'implémentation de cet algorithme. Finalement, nous nous focalisons plutôt sur les performances de PESMO.

9 Application finale

Des algorithmes d'optimisation bayésienne pour le problème multi-objectif ont déjà été implémentés et adaptés au contexte très bruité. Il s'agit de comparer les performances des implémentations mis à disposition en amont du stage. Pour cela, nous nous basons sur les métriques de performance suivantes.

9.1 Métriques

Les métriques présentées ci-dessous sont utilisées dans le cadre de la thèse de mon maître de stage. Elles sont inspirées des métriques classiques dans la littérature de l'optimisation multi-objectif.

Définition 9.1. Hypervolume de front de Pareto
Soit la région définie entre le front de Pareto $\mathcal{Y}^*$ et un point de référence R comme étant les points qui dominent R, et qui sont Pareto-dominés par un élément de $\mathcal{Y}^*$, $D(\mathcal{Y}^*) = \bigcup_{y^* \in \mathcal{Y}^*} \{y \in \mathbb{Y} : y^* \prec y \prec R\}$. Le volume de cette région est appelé hypervolume de de front de Pareto.

Dans le cas très bruité, la différence d'hypervolume est définie comme suit :

$$V_d(\mathcal{Y}^*, \mathcal{Y}) = V((D(\mathcal{Y}) - D(\mathcal{Y}^*) \cup (D(\mathcal{Y}^*) - D(\mathcal{Y}))). \tag{9.1}$$

Définition 9.2. taux de mauvaise classification de points non dominés

Un point de l'espace d'entrée peut être classé comme non dominé (appartenant à $\mathcal{X}^*$) ou dominé. Le taux de mauvaise classification de points non dominés est défini comme le pourcentage de points dont la prédiction de classification diffère de la vraie classe parmi celle des points dominés et celle des points non dominés :

$$M(\mathcal{X}^*, \mathcal{X}) = \frac{1}{N} \sum_{x \in \mathbb{X}} \left(\mathbb{1}_{\{x \in \mathcal{X}^*\}} - \mathbb{1}_{\{x \in \mathcal{X}\}} \right)^2 . \tag{9.2}$$

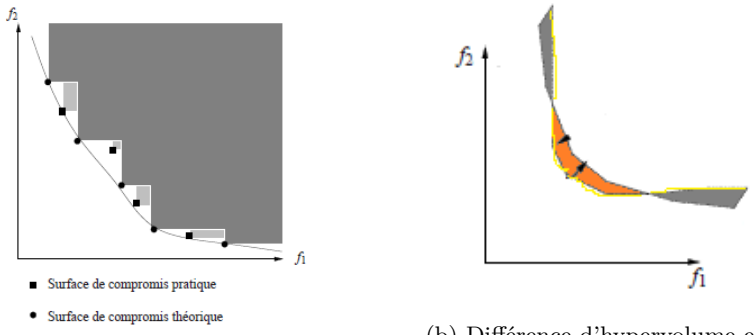
Définition 9.3. Distance d'Hausdorff

Soient deux ensembles finis A et B, la distance d'Hausdorff est donnée par :

$$d_H(A, B) = \max\{\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(a, b)\}. \tag{9.3}$$

La distance d'Hausdorff est petite si pour un point donné dans l'ensemble A, il existe un point de l'autre ensemble B qui est proche.

Pour définir les métriques de distance d'Hausdorff entre fronts ou entre ensemble de points, d correspond à la distance euclidienne sur l'espace de sortie ou d'entrée respectivement. Ci-dessous, on donne une visualisation des métriques calculées (Figure 17b).



(a) Hypervolume de front (cas non bruité)

(b) Différence d'hypervolume entre le front de Pareto estimé et le vrai front de Pareto (jaune)

FIGURE 17 – Métriques d'hypervolume et distance d'Hausdorff de front de Pareto

Dans le cas non bruité, la différence d'hypervolume correspondrait à la somme des aires des zones grisées signées selon que le front estimé est au-dessus du vrai front ou non dans la figure. 17a. Cette métrique est réduite pour les points de la partie supérieure pour le front estimé (Figure 10b), alors qu'elle ne le devrait pas. En même temps, on ne sait pas à quel point la prédiction de front pour cette partie doit être pénalisante du fait du bruit. Nous nous concentrons donc à évaluer la métrique de différence d'hypervolume, quand le front estimé est au-dessus du vrai front de Pareto. En pratique, cela donne une bonne indication de la performance pour le front de Pareto. Nous avons aussi représenté la distance d'Haussdorff en terme de front, par les flèches. Les métriques dans l'espace d'entrée ne diffèrent pas de leur définition dans le cadre non bruité.

Le but est de minimiser au fil des itérations des algorithmes les métriques. Par leur définition, les métriques basées sur la distance d' Hausdorff sont locales, car elles pénalisent des propositions d'ensemble de Pareto ou de front de Pareto qui ont au moins un point très éloigné du vrai ensemble de Pareto ou front de Pareto. Pour l'application à la planification, on peut se permettre de fournir un nombre minime de jeux de paramètres de mauvaises stratégies tant que le taux de mauvaise classification de points non dominés est bien bas. Les métriques globales sont celles qui nous intéressent, en particulier le taux de mauvaise classification de points non dominés.

9.2 Performances des algorithmes

Les problèmes utilisés pour tester la performance des algorithmes consistent en des problèmes dont les solutions Pareto-optimales ont été préalablement déterminées, ce qui permet ainsi de calculer les 4 métriques. Pour rappel, nous simulons sous MATLAB le bruit par un bruit gaussien. Les problèmes-test ont tous en commun les caractéristiques suivantes :

- $\mathbb{X}$ une grille de 21×21 ;
- 2 fonctions objectif ;
- bruit additif gaussien.

L'allocation de budget est de 50 200 évaluations, avec $n_{batch} = 200$, $n_0 = 20$ sélectionnés en lot de 10. Le nombre de lot d'évaluations n'intervient pas dans le processus d'optimisation en lui-même. Il est défini pour réduire suffisamment la variance de bruit des évaluations. Le niveau de bruit fixé pour chacun des problèmes a été adapté selon la performance au niveau des métriques d'une méthode de base de brute force qui résout le problème multi-objectif à partir de toutes les évaluations bruitées. Par exemple, pour une fonction objectif f_k de maximum 10^5, la variance de bruit σ_k^2 est de l'ordre de 10^8, d'où la nécessité de procéder à des évaluations en lot pour l'optimisation. Le niveau de bruit, même réduit par les évaluations en lot, reste considérablement plus élevé comparé à celui dans les problèmes traités dans la littérature. Les problèmes-test sont :

Problèmes	Fonctions	Formules
base_in01	Coût de la stratégie Taux CMA	Fonctions objectif 1 et 2
test_a1	Branin-Hoo modifiée Exponential_stk	$f_1(x) = \left(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$ $f_2(x) = \exp(1.8(x_1 + x_2)/5) + 3x_1/5 + 6/5 \times x_2^2 + 3\sin(4\pi x_1/5)$
test_a2	Branin-Hoo modifiée Rosenbrock	$f_1(x) = \left(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$ $f_2(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
test_a3	Exponential_stk Branin-Hoo modifiée	$f_1(x) = \exp(1.8(x_1 + x_2)/5) + 3x_1/5 + 6/5 \times x_2^2 + 3\sin(4\pi x_1/5)$ $f_2(x) = \left(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$
test_b1 test_b2 test_b3 test_b4 test_b5	Polynômes	Pour plusieurs coefficients

TABLE 3 – Problèmes-tests

Les problèmes avec des polynômes sont considérés comme les plus simples à résoudre. Ils permettent d'identifier si l'algorithme d'optimisation sélectionne les bons points à évaluer. Si les courbes en moyenne des métriques en fonction des itérations pour un algorithme donné restent bien en-dessous de la courbe associée à la méthode de sélection aléatoire des points (voir annexe D). C'est bien le cas pour toutes les méthodes testées. Les tests a correspondent à des problèmes plus difficiles indépendamment du niveau de bruit.

Nous avons essayé une première approche qui consiste à implémenter MESMO ou *Max-Value Entropy Search for Multi-Objective Bayesian Optimization* Belakaria *et al.* (2019) dont le critère d'échantillonnage se base sur de la réduction d'entropie dans l'espace de sortie. Il diffère de celui de PESMO, dans le sens où il approxime la loi

$$\mathbb{P}_{Y(x)|\mathcal{D}_n,\mathcal{Y}^{sub}} \text{ par } \mathbb{P}_{Y(x)|\mathcal{D}_n,MV} \text{ où } MV \text{désigne la contrainte } \forall j, F_j(x) \geq \min_{z \in \mathcal{Y}^{sub}} z_j. \tag{9.4}$$

C'est une approximation simplificatrice par dimension d'un autre ordre que celle faite par PESMO nous permettant de ramener le calcul de l'entropie des distribution finies dimensionnelles du processus gaussien contraint à un calcul d'entropie d'une loi normale tronquée univariée. Or, la fonction d'acquisition telle que présentée dans l'article ne dépend pas du facteur σ_k^2 associé au bruit gaussien. Des premiers tests en local sur les problèmes montrent que l'algorithme n'explore pas du tout l'espace d'entrée sur quelques uns des problèmes-tests, sélectionnant toujours le même point pendant la plupart des itérations.

Les expériences sont tournées avec 500 initialisations différentes sur les serveurs de CentraleSupélec en parallèle via l'utilisation de la boîte à outils MATLAB Parallel Computing Toolbox. On teste PESMO1 et PESMO2 avec plusieurs valeurs de S : 1, 5, 10. Au-delà, la génération d'ensembles de Pareto est trop coûteuse. La comparaison est réalisée avec PALS. C'est un algorithme dérivé de PAL ou *Pareto Active Learning* (Zuluaga *et al.*, 2013) un autre algorithme d'optimisation bayésienne dont la fonction d'acquisition est basée sur la théorie de l'information. PAL procède à chaque itération, à une classification des points de l'espace d'entrée en 3 catégories : Pareto-optimaux, non Pareto-optimaux ou incertain. Chacun des points incertains de l'espace d'entrée a une région d'incertitude rectangulaire qui permet la classification. Le point sélectionné est celui qui maximise la diagonale de la région d'incertitude, encourageant donc l'exploration. Cet algorithme a des garanties théoriques contrairement à PESMO, concernant la croissance du gain d'information à mesure que l'on évalue les observations. PAL a été adapté et paramétré pour optimiser les métriques. En effet, dans un contexte très bruité, PAL n'atteint pas les performances attendues. Il a été autorisé notamment de sélectionner plusieurs fois le même point, comme pour PESMO.

De manière générale, PALS performe mieux, même si les valeurs de métriques sont quasiment les mêmes pour avec la meilleure version de PESMO sur de nombreux problèmes-test. Le seul problème pour lequel les méthodes de PESMO sont bien meilleures est le test_a1. En effet, le taux de mauvaise classification de points non dominés est bien décroissant au fil des itérations, contrairement à la méthode Random et à la méthode PALS. Les performances des méthodes PESMO semblent bien plus variables pour les problèmes a, même en moyennant sur 500 initialisations, ce qui est probablement dû à quelques problèmes numériques lors d'inversions de matrices de covariance obtenues par krigeage notamment.

Problèmes	PALS	PESMO1-1	PESMO1-10	PESMO1-5	PESMO2-1	Random
base_in01	0.0063122	**0.0059412**	0.0081489	0.0080037	0.0081364	0.0077406
test_a1	**0.0095462**	0.0099941	0.011701	0.011248	0.010513	0.010049
test_a2	0.0091311	0.010533	**0.009014**	0.0094858	0.01197	0.010551
test_a3	**0.01132**	0.013313	0.016259	0.012383	0.012591	0.012119
test_b1	**0.0069385**	0.008495	0.0085442	0.008163	0.010696	0.011016
test_b2	**0.004688**	0.0071714	0.0079597	0.0071746	0.013622	0.014113
test_b3	**0.0039821**	0.0057623	0.0062418	0.0058286	0.0096994	0.0094398
test_b4	**0.0062032**	0.0065989	0.0065458	0.0069061	0.0081641	0.0086191
test_b5	**0.0056154**	0.0062011	0.0066356	0.0064757	0.010056	0.010748

TABLE 4 – Différence de l'hypervolume à la dernière itération en moyenne

Problèmes	PALS	PESMO1-1	PESMO1-10	PESMO1-5	PESMO2-1	Random
base_in01	**0.056036**	0.059805	0.05805	0.065007	0.082512	0.072399
test_a1	0.010503	**0.0045442**	0.0056054	0.0061179	0.0047347	0.012349
test_a2	**0.032549**	0.035238	0.035166	0.038744	0.039841	0.035796
test_a3	**0.019342**	0.025469	0.022268	0.02098	0.026612	0.021206
test_b1	0.032544	0.032254	0.031347	**0.027986**	0.036322	0.03858
test_b2	**0.003873**	0.004771	0.0050113	0.0061451	0.0066122	0.0069524
test_b3	0.025574	0.025782	0.027592	0.02502	**0.022494**	0.026249
test_b4	**0.038086**	0.040594	0.040268	0.04039	0.040444	0.043918
test_b5	**0.0095692**	0.0097642	0.011057	0.0097551	0.012508	0.013932

TABLE 5 – taux de mauvaise classification de points non dominés à la dernière itération en moyenne

PALS	PESMO2-1	PESMO1-1	PESMO1-5
$\mathbf{0.235037 \pm 0.031604}$	0.512335 ± 0.146350	0.735311 ± 0.288385	1.764110 ± 0.442559

TABLE 6 – Durée d'une itération en moyenne sur le problème de base bruité sur 30 initialisations différentes

Les durées d'itération sont données ici qu'à titre d'indication. Il est à noter que PALS est au moins deux fois plus rapide que les différentes versions implémentées de PESMO. La variabilité de la durée d'itération pour les versions de PESMO est principalement due au fait que les ensembles de Pareto générés à chaque itération de l'algorithme sont potentiellement de tailles différentes. Même si PESMO2 est plus rapide comme attendu, les résultats sur les métriques sont meilleurs pour PESMO1. Comme PESMO2 est une approximation de ce qui est fait dans PESMO1, ce n'est pas surprenant. En revanche, les valeurs de métriques pour PESMO2 sont parfois plus élevées que celles obtenues avec la méthode Random. Cela est peut-être dû au fait que PESMO2 tel qu'il est implémenté n'est pas robuste au nombre de trajectoires tirées S comme l'indique Belakaria *et al.* (2019). Nous n'avons pu essayer que pour $S = 1$. Les calculs sur les serveurs ont planté pour des valeurs plus élevées. Le bug a été résolu en incluant tous les facteurs, même ceux constants égaux à 1 pour l'inférence par EP, mais nous n'avons pas eu le temps de refaire tourner les scripts. De manière générale, PESMO1 avec $S = 1$ affiche les meilleures performances sur toutes les métriques. Tous les résultats sur les 9 problèmes pour chaque itération se trouvent dans l'annexe D.

10 Perspectives

Ce stage a été l'occasion de mettre en pratique les éléments théoriques vus en master sur les processus gaussiens, et de se familiariser avec des problématiques qui sont posées dans un problème industriel ici, l'exploitation de sorties d'un simulateur stochastique. Le stage a abouti à une implémentation de l'algorithme d'optimisation bayésienne adaptée aux problèmes d'optimisation multi-objectif considérés dans la thèse de mon maître de stage. Le principe de PESMO est repris avec moins d'approximations au détriment d'un temps de calcul plus long. De toute façon, les résultats dans l'article original *Predictive Entropy Search for Multi-Objective Bayesian Optimization* montrent aussi que le temps de calcul est plus long que pour d'autres méthodes tel que PAL, mais équivalent à celui de SUR. Comme le contexte n'est pas le même que celui de l'article original, nous n'avons pas pu comparer les deux implémentations pendant le stage. Un prolongement du travail effectué serait de parvenir à faire tourner l'implémentation de l'article disponible dans le dépôt Git Spearmint dans la branche PESM (`https://github.com/HIPS/Spearmint/tree/PESM`), puis de comparer cette implémentation à celle

décrite dans le rapport. Le travail du stage s'inscrit dans la comparaison de plusieurs algorithmes d'optimisation bayésienne multi-objectif que sont PAL, PESMO et SUR adaptés à un contexte très bruité.

Un algorithme dont le calcul de la fonction d'acquisition est plus long peut être utilisé sur le problème avec des évaluations directement issues du simulateur s'il permet d'obtenir des valeurs de métriques bien plus basses sur les problèmes test. Les expériences montrent que ce n'est pas vraiment le cas pour l'implémentation de PESMO. PALS semble pour l'instant la solution privilégiée, même si d'autres tests sur les algorithmes sont encore à faire.

Par ailleurs, il est envisagé de considérer dans la suite de la thèse un plus grand nombre de paramètres de stratégie, et plus de fonctions objectif. Ce changement de cadre peut aussi affecter la performance des algorithmes sur le problème de base qui changerait de dimension dans l'espace d'entrée et l'espace de sortie.

Une autre piste concernant la modélisation du problème de planification est de considérer des méta-modèles probabilistes qui sont moins classiques comme des processus stochastiques établis à partir de lois de Student. En effet, ces familles de processus stochastiques sont choisies en fonction de la loi qui décrit la distribution du bruit. En l'occurrence, la distribution a des queues lourdes. De plus, les fonctions objectif ne sont pas inférieures à 0, ce qui est permis pour les méta-modèles de krigeage associées aux fonctions. Une loi normale tronquée est adaptée pour modéliser les coûts et taux évalués en sortie du simulateur, mais nous rentrons dans des méta-modèles plus complexes. La familles des lois de Student n'est pas une famille exponentielle, ce qui réduit les chances de pouvoir appliquer le principe de PESMO utilisant *Expectation Propagation* avec des méta-modèles plus complexes. La résolution du problème de planification efficace par des algorithmes d'optimisation donne lieu à de nombreuses pistes de résolution. Celle pour laquelle nous avons opté, repose sur de nombreuses hypothèses, et le travail de la thèse de Bruno Tebbal Barracosa dans sa dernière année de thèse est d'étudier des pistes qui supposent moins de simplifications.

Annexes

A Détermination de la loi *a posteriori* d'une fonction objectif

Cette section est implémentée dans `STK`. Une version de la toolbox est disponible sur `https://github.com/stk-kriging/stk`. L'outil est implémenté en MATLAB et se charge des calculs avec processus gaussiens dans les algorithmes d'optimisation bayésienne.

A.1 Prédicteur bayésien

On suppose que le processus gaussien F dans l'espace fonctionnel a pour fonction de covariance un noyau K. Après avoir évalué n points, $K_n = \left(K(\|x^{(i)} - x^{(j)}\|) \right)_{i,j \in [\![1,n]\!]} = k(x^{(i)}, x^{(j)})$. Quand on ajoute à $F^{(1:n)}$ l'évaluation d'un candidat x, on a :

$$\tilde{F} = \begin{pmatrix} F(x^{(1)}) \\ \vdots \\ F(x^{(n)}) \\ F(x) \end{pmatrix} \sim \mathcal{N}(0, K) \quad \text{avec :} \quad K = \left[\begin{array}{c|c} K_n & k_n(x) \\ \hline k_n(x)^\top & k(x,x) \end{array}\right], \quad k_n(\cdot) = \begin{pmatrix} k(x^{(1)}, \cdot) \\ \vdots \\ k(x^{(n)}, \cdot) \end{pmatrix}.$$

On peut aussi se baser sur des observations bruitées $(y(x^{(i)}))_{i=1}^n$ de f que l'on modélise comme F avec bruit blanc avec variance σ^2. On suppose que le bruit est i.i.d., donc :

$$Y_n = \begin{pmatrix} Y(x^{(1)}) \\ \vdots \\ Y(x^{(n)}) \end{pmatrix} \sim \mathcal{N}(0, K_n + \sigma^2 I_n), \tag{A.1}$$

$$\tilde{Y} = \begin{pmatrix} Y(x^{(1)}) \\ \vdots \\ Y(x^{(n)}) \\ f(x) \end{pmatrix} \sim \mathcal{N}(0, K) \qquad \text{avec :} \qquad K = \left[\begin{array}{c|c} K_n + \sigma^2 I_n & k_n(x) \\ \hline k_n(x)^\top & k(x,x) \end{array}\right].$$

Par le lemme d'inversion de matrice,

$$\left[\begin{array}{c|c} K_n & k_n(x) \\ \hline k_n(x)^\top & k(x,x) \end{array}\right]^{-1} = \left[\begin{array}{c|c} \tilde{K} & \tilde{k} \\ \hline \tilde{k}^\top & r \end{array}\right]$$

avec :

$$\begin{cases} \tilde{K} &= K_n^{-1} + K_n^{-1} k_n(x)\big(k(x,x) - k_n(x)^\top K_n^{-1} k_n(x)\big)^{-1} k_n^\top K_n^{-1} \\ \tilde{k} &= -K_n^{-1} k_n(x)\big(k(x,x) - k_n(x)^\top K_n^{-1} k_n(x)\big)^{-1} \\ \tilde{r} &= \frac{1}{k(x,x) - k_n^\top K_n^{-1} k_n}. \end{cases}$$

D'où :

$$\begin{aligned} p(y(x)|\mathcal{D}_n) &\propto p(\tilde{y}) \\ &\propto e^{-\tilde{y}^\top K^{-1} \tilde{y}} \\ &\propto e^{-(\tilde{k}^\top y^{(1:n)})^\top y(x) - \frac{\tilde{r} y(x)^2}{2}}. \end{aligned}$$

La moyenne et la variance de la loi postérieure $\mathbb{P}_{F(x)|\mathcal{D}_n}$ correspondent à :

$$\mu_n(x) = -\tilde{k}^\top v_n(x)\, y^{(1:n)} \quad \text{avec : } v_n(x) = k(x,x) - k_n(x)^\top K_n^{-1}\, k_n(x),$$

soit :

$$\mu_n(x) = (K_n^{-1} k_n(x))^\top\, y^{(1:n)} \text{ et } v_n(x) = k(x,x) - k_n(x)^\top K_n^{-1}\, k_n(x).$$

Pour des observations bruitées, on a les moyenne et variance prédictives :

$$\mu_n(x) = (K_n^{-1} k_n(x))^\top\, y^{(1:n)} \text{ et } v_n(x) = k_n(x)^\top K_n^{-1}\, \tilde{y}_n \quad \text{avec : } v_n(x) = k(x,x) - k_n(x)^\top (K_n + \sigma^2 I_n)^{-1}\, k_n(x).$$

A.2 Prédicteur par krigeage

Nous présentons ici une autre approche pour la prédiction par processus gaussien qui est celle de la bibliothèque de code que j'ai utilisée durant le stage. La technique de krigeage fournit le meilleur prédicteur linéaire **non-biaisé** (Jones *et al.*, 1998), c'est-à-dire : $\mathbb{E}[\hat{F}(x)] = \mathbb{E}[F(x)] = m(x)$, avec :

$$\hat{F}(x) = \lambda(x)^\top f^{(1:n)}, \tag{A.2}$$

où $f^{(1:n)}$ est le vecteur des valeurs prises par la fonction aléatoire F en n points de l'espace. On peut alors estimer l'erreur quadratique de l'estimateur :

$$\begin{aligned}
\Delta(x) &= \mathbb{E}[(\hat{F}(x) - F(x))^2] \\
&= \mathbb{E}[\hat{f}(x)^2] - 2\mathbb{E}[F(x)\lambda(x)^\top F_n] + \mathbb{E}[F(x)^2] \\
&= Var(\hat{F}(x)) + m(x)^2 - 2\lambda(x)^\top[\mathrm{Cov}(F(x_1), F(x)), \dots, \mathrm{Cov}(F(x_n), F(x))]^\top + 2\lambda(x)^\top\mathbb{E}[F(x)]\mathbb{E}[F_n] \\
&+ Var(F(x)) + m(x)^2 \\
&= Var(\hat{F}(x)) + m(x)^2 - 2\lambda(x)^\top[\mathrm{Cov}(F(x_1), F(x)), \dots, \mathrm{Cov}(F(x_n), F(x))]^\top + 2\mathbb{E}[F(x)]\mathbb{E}[\lambda(x)^\top F_n] \\
&+ Var(F(x)) + m(x)^2 \\
&= Var(\hat{F}(x)) + m(x)^2 - 2\lambda(x)^\top[\mathrm{Cov}(F(x_1), F(x)), \dots, \mathrm{Cov}(F(x_n), F(x))]^\top + 2m(x)^2 \\
&+ Var(F(x)) + m(x)^2 \\
&= \lambda(x)^\top K_n\lambda(x) - 2\lambda(x)^\top k_n(x) + k(x,x).
\end{aligned}$$

On suppose que la moyenne sur F et donc son estimateur est une combinaison linéaire de fonction simples ξ, d'où la contrainte tout le long : $\lambda(x)^\top\mathbb{E}[F_n] = \beta^\top\Xi(x)$ ou encore $\beta^\top\Xi_n\lambda(x) = \beta^\top\Xi(x)$. On cherche à minimiser l'erreur de l'estimateur par rapport aux coefficients $\lambda(x)$, ce qui donne le problème suivant :

$$\begin{aligned}
&\min \Delta(x) - k(x,x) \\
&s.t.\ (\Xi_n)^\top\lambda(x) = \Xi(x),
\end{aligned}$$

pour tout β doit être vrai. On pose le lagrangien avec un multiplicateur de Lagrange $2\nu(x)$. On minimise selon le vecteur de poids λ. on ignore $k(x,x)$ On a donc :

$$\mathcal{L}(\lambda, \nu) = \lambda(x)^\top K_n\lambda(x) - 2\lambda(x)^\top q(x) + 2\nu(x)((\Xi_n)^\top\lambda(x) - \Xi(x)).$$

L'annulation du gradient donne : $2K_n\lambda(x) - 2q(x) + 2\nu(x)\Xi_n = 0$. Avec la contrainte pour laquelle la solution doit vérifier, on obtient alors le système :

$$\begin{bmatrix} K_n & \Xi_n \\ \Xi_n^\top & 0 \end{bmatrix} \begin{bmatrix} \lambda(x) \\ \nu(x) \end{bmatrix} = \begin{bmatrix} k_n(x) \\ \Xi(x) \end{bmatrix}. \tag{A.3}$$

D'où :

$$\begin{aligned}
\Delta(x) &= k(x,x) + \lambda(x)^\top[K_n\lambda(x) - 2k_n(x)] \\
&= k(x,x) + \lambda(x)[K_n\lambda(x) - k_n(x) - k_n(x)] \\
&= k(x,x) - \lambda(x)^\top k_n(x) - \lambda(x)^\top\Xi_n \\
&= k(x,x) - \lambda(x)^\top k_n(x) - \Xi(x)\nu(x).
\end{aligned}$$

On remarque que quand $m(x) = 0$, la tendance de krigeage est donc nulle, ce qui ramène les Ξ à des fonctions nulles quelque soit β. Dans ce cas, (A.3) devient : $K_n\lambda(x) = k_n(x)$, soit : $\lambda(x) = K_n^{-1}k_n(x)$. Les poids attribués correspondent exactement aux poids attribués pour le prédicteur bayésien de l'annexe A.1. Ce prédicteurs est donc optimal par rapport au critère de krigeage.

En revanche, dans le cas plus général où $m(x)$ est inconnue, dans un cadre bayésien on ajoute une incertitude sur cette moyenne, ce qui reviendrait à placer une variance infinie sur le vecteur de coefficients β dans la moyenne $m(x)$ (Dempfle, 1977). Pour des observations bruitées, comme pour la loi *a posteriori* dans le cadre bayésien, on ajoute un paramètre de variance de bruit σ^2 à la matrice de covariance K_n. Cela a aussi pour effet de régulariser la matrice de covariance, dans le cas où elle est proche d'être singulière.

A.3 Estimation des paramètres de covariance par maximum de vraisemblance

A chaque itération n, on procède à une estimation des paramètres par maximum de vraisemblance par rapport aux paramètres de covariance que l'on dénote ici α. Considérons la loi jointe du vecteur gaussien des

observations $f^{(1:n)}$ et supposons la moyenne $m(x)$ de F connue. L'estimateur $\hat{\alpha}$ par maximum de vraisemblance du vecteur α des paramètres de la covariance s'obtient alors en maximisant la log-vraisemblance :

$$\ell(\bar{f}^{(1:n)}, \alpha) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log\det K(\alpha) - \frac{1}{2}(\bar{f}^{(1:n)})^\top K(\alpha)^{-1}\bar{f}^{(1:n)}, \tag{A.4}$$

où $\bar{f}^{(1:n)}$ sont centrées. En revanche, quand la moyenne (tendance a priori) n'est pas connue, ce qui est le cas dans la plupart des exemples notamment le nôtre, on a recours au maximum de vraisemblance restreint. Supposons que la moyenne de F s'écrive $m(x) = \beta^\top p(x)$. Si le vecteur β n'est pas connu a priori, il est néanmoins possible d'estimer $\hat{\alpha}$ en considérant la vraisemblance d'un vecteur de contrastes, c'est-à-dire d'une combinaison linéaire des observations dont la distribution jointe ne dépend pas de β (Stein, 2012).

B Moment Matching

Dans cette section, nous justifions l'usage de *Moment Matching* pour minimiser la divergence de Kullback-Leibler entre les distributions. Dans l'article de Hernández-Lobato *et al.* (2016), les lois tiltées sont impropres. On utilise alors la KL-divergence pour des lois non normalisées P et Q. Elle se définit comme suit :

$$D_{KL}(P \,\|\, Q) = \int p(u)\log\frac{p(u)}{q(u)}du + \int q(u)du - \int p(u)du. \tag{B.1}$$

Or, projeter sur l'espace $\mathcal{F}$ revient à faire correspondre les moments d'ordres 1 et 2. Notons qu'ici $q(u) \in \mathcal{F}^U$, et prend la forme :

$$q(u) = Z_q \exp\{\langle\theta, \phi(u)\rangle_{\mathcal{F}} - \Phi(\theta)\}, \tag{B.2}$$

où Z_q est une constante quelconque, pas nécessairement de normalisation. Donc :

$$\begin{aligned} D_{KL}(P \,\|\, Q) &= \int p(u)(\Phi(\theta) - \langle\theta|\phi(u)\rangle_{\mathcal{F}} - \log Z_q)du + Z_q - \int p(u)du - \mathcal{H}[p] \\ &= Z_p\Phi(\theta) - \langle\theta|\int \phi(u)p(u)du\rangle_{\mathcal{F}} - Z_p\log Z_q + (Z_q - Z_p) - \mathcal{H}[p], \end{aligned}$$

avec Z_p la constante de normalisation de la densité p. La dérivée partielle de cette expression par rapport à Z_q vaut :

$$\frac{\partial}{\partial Z_q}D_{KL}(P \,\|\, Q) = -\frac{Z_p}{Z_q} + 1. \tag{B.3}$$

La dérivée partielle seconde est positive, donc le minimiseur global est $Z_q^* = Z_p$. Les moments d'ordre 0 correspondent. Pour calculer une dérivée partielle par par rapport à θ du produit $\langle\theta|\rangle_{\mathcal{F}}$, on calcule la dérivée par rapport à m de $\langle m|\cdot\rangle_{R^N}$, et celle par rapport à Λ du produit de Hadamard $\Lambda \odot \cdot$, ce qui donne :

$$\begin{aligned} \frac{\partial}{\partial\theta}D_{KL}(P \,\|\, Q) = Z_p\nabla_\theta\Phi(\theta) - \int p(u)\phi(u)du = 0 \\ \implies \nabla_\theta\Phi(\theta) = \int \frac{p(u)}{Z_p}\phi(u)du. \end{aligned}$$

Ainsi, les paramètres de moments η de p et q sont donc égaux.

C Random Kernel Features

Cette technique permet de générer des pseudo-trajectoires de processus gaussiens de façon peu coûteuse. On utilise le théorème de Bochner qui nous permet d'exprimer le noyau en fonction de sa transformée de

Fourier-Stieltjes. Soit p la densité spectrale normalisée du noyau k et α la constante de normalisation, on a :

$$\begin{aligned} k(x, x^{'}) &= \alpha \mathbb{E}_{W\sim p}\left[e^{-iW^\top(x-x^{'})}\right] \\ &= 2\mathbb{E}_{(W,B)\sim p_{W,B}}\left[cos(W^\top x + B)cos(W^\top x^{'} + B)\right] \\ &\approx \frac{2\alpha}{m}\sum_{k=1}^{m} cos(w_k^\top x + b_k)cos(w_k^\top x^{'} + b_k) \\ &\approx \phi(x)^\top \phi(x^{'}), \end{aligned} \tag{C.1}$$

où $B \sim \mathcal{U}([0, 2\pi])$ indépendant de W, et m le nombre d'échantillons pour l'approximation par Monte-Carlo. $\phi(x) = \sqrt{\frac{2\alpha}{m}}\cos(wx + b)$, avec w et b les matrices dont les lignes sont $w_k^\top$ et $b_k^\top$ pour $k = 1, \dots m$.

Dans le problème d'optimisation, on se sert de la connaissance des points $\mathcal{D}_n$ pour estimer la fonction. Ainsi, on modélise $f \in \texttt{Span}(k_{x_1}, \dots, k_{x_n})$. On pose : $F(x) = \phi(x)^\top\theta$ où $\theta \sim \mathcal{N}(0, I_m)$, et $Y(x) = F(x) + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, \sigma^2)$ ce qui donne la prior suivant pour $F(x) \sim \mathcal{N}(0, \phi(x)\phi(x)^\top)$, et donc $Y(x) \sim \mathcal{N}(0, \phi(x)\phi(x)^\top + \sigma^2)$. Par A et l'approximation du noyau par les features, on a la distribution prédictive sur F :

$$\begin{aligned} \mu_n(x) &= \phi(x)^\top\Phi_n(\Phi_n\Phi_n^\top + \sigma^2 I_n)^{-1}y^{(1:n)}, \\ v_n(x) &= \phi(x)^\top\phi(x) - \phi(x)\Phi^\top(\Phi_n\Phi_n^\top + \sigma^2 I_n)^{-1}\Phi_n\phi(x). \end{aligned}$$

Par une application du lemme d'inversion de matrice, on a :

$$\begin{aligned} \mu_n(x) &= \phi(x)^\top(\Phi_n^\top\Phi_n + \sigma^2 I_m)^{-1}\Phi_n y_n, \\ v_n(x) &= \phi(x)^\top(\Phi_n^\top\Phi_n + \sigma^2 I_m)^{-1}\phi(x)\sigma^2. \end{aligned}$$

Donc

$$\theta|\mathcal{D}_n \sim \mathcal{N}(A^{-1}\Phi_n^\top y_n, \sigma^2 A^{-1}),$$

où $A = \Phi_n^\top\Phi_n + \sigma^2 I_m$.

Algorithme 6 : Random kernel features (parallélisable)

Données : Candidat x, $\mathcal{D}_n$
Résultat : $f^{(1)}, \dots, f^{(S)}$

```
/* mesure spectrale / uniforme                                          */
pour i=1,..., S faire
    Générer w^(i), b^(i) selon la loi jointe p_{W,B} // DSP Student multivariée d'ordre 3,5 pour Matern 3/2,5/2
    Φ^(i) = sqrt(2α/m) cos(w^(i) x + b^(i))
    Générer θ^(i) selon N(A^{-1}(Φ_n^(i))^⊤ y^(1:n), σ^2 A^{-1})
    f^(i) ← (Φ^(i))^⊤ θ^(i)
fin pour
```

Remarque C.1. Cette technique a l'avantage d'avoir le facteur dépendant des moyennes et covariance de krigeage θ qui ne fait pas intervenir x. Il suffit de le générer une fois pour toute. La génération de $\phi(x)$ ne nécessite pas contrairement à la génération classique de trajectoires de conditionner par rapport à $\mathcal{D}_n$. On a donc à moindre coût une pseudo-trajectoire sur un espace continu avec la méthode.

D Résultats sur tous les problèmes tests

- Résultats sur base_in01

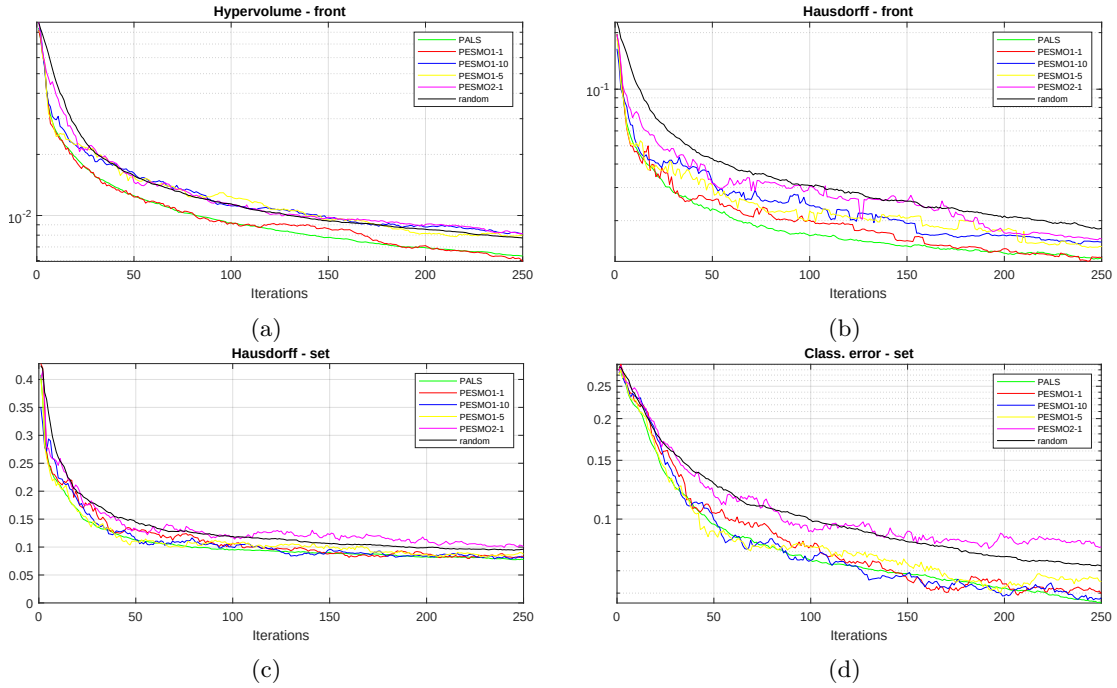


FIGURE 18 – Comparaison des algorithmes pour les métriques moyennés sur 500 initialisations

- Résultats sur test_a1

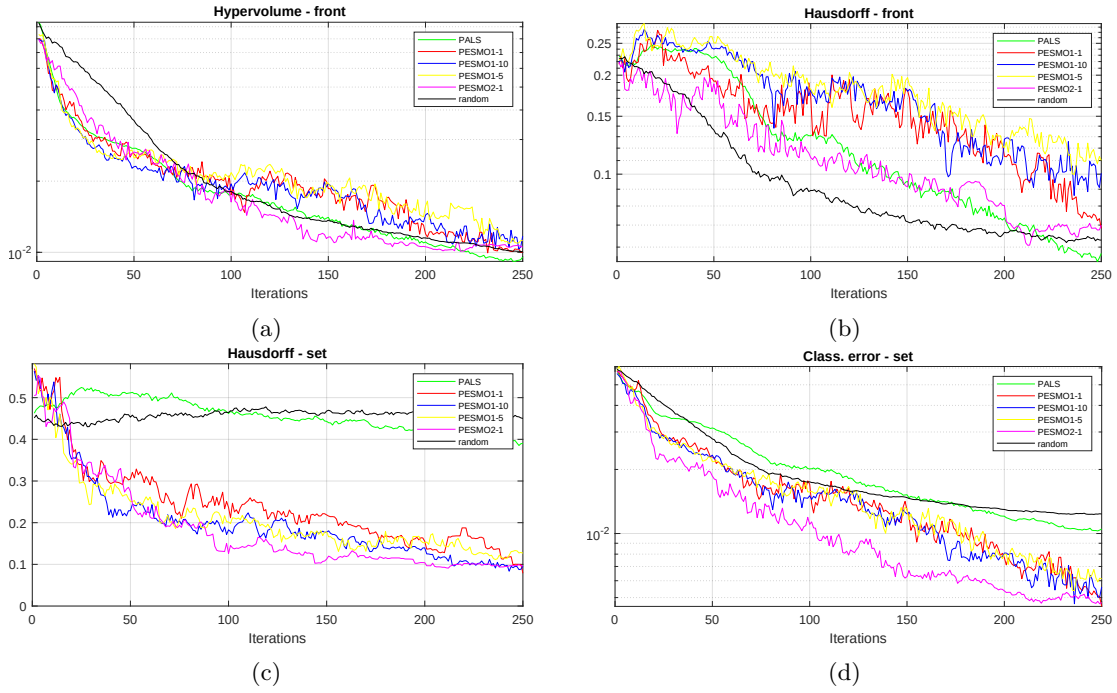
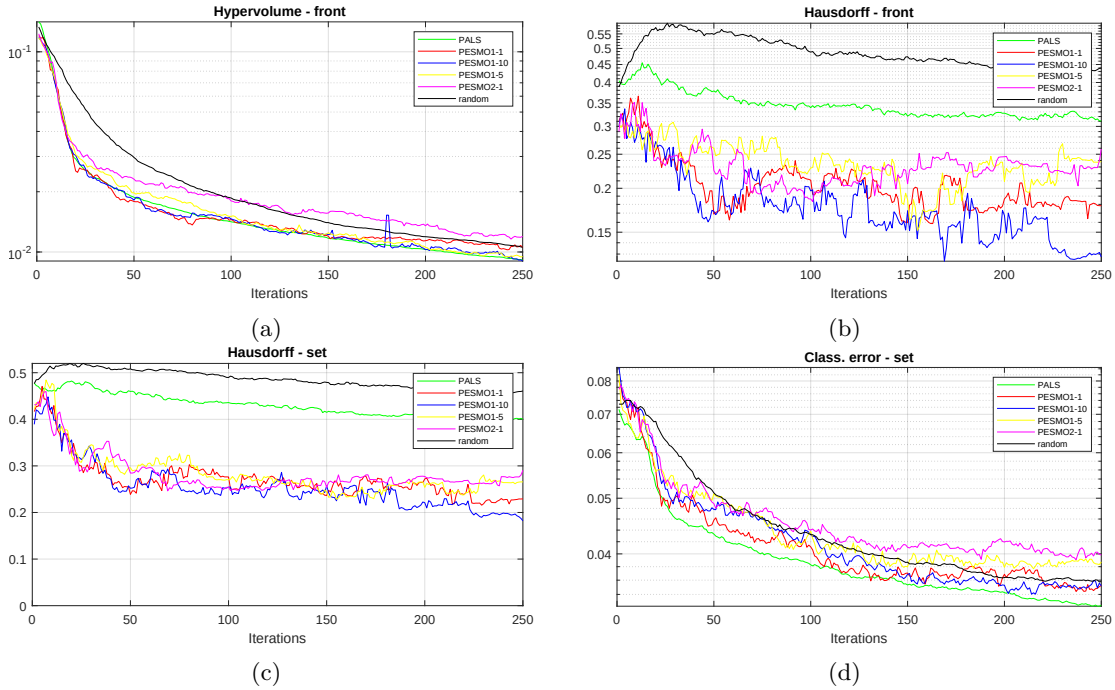


FIGURE 19 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

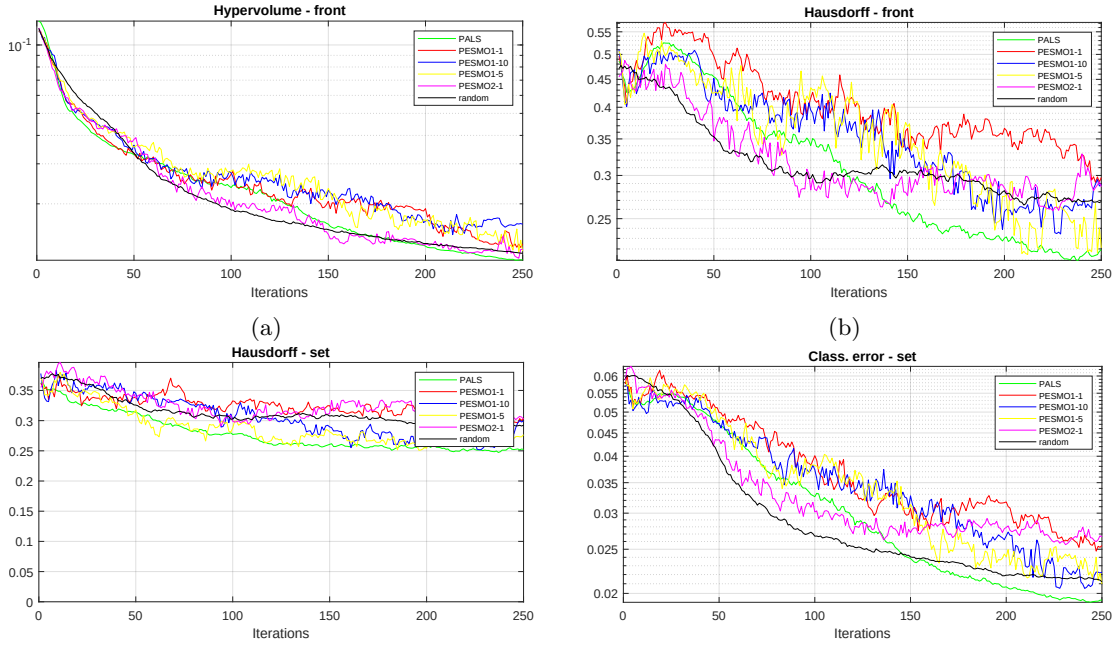
- Résultats sur test_a2



(a) (b) (c) (d)

FIGURE 20 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

- Résultats sur test_a3



(a) (b) (c) (d)

FIGURE 21 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

- Résultats sur test_b1

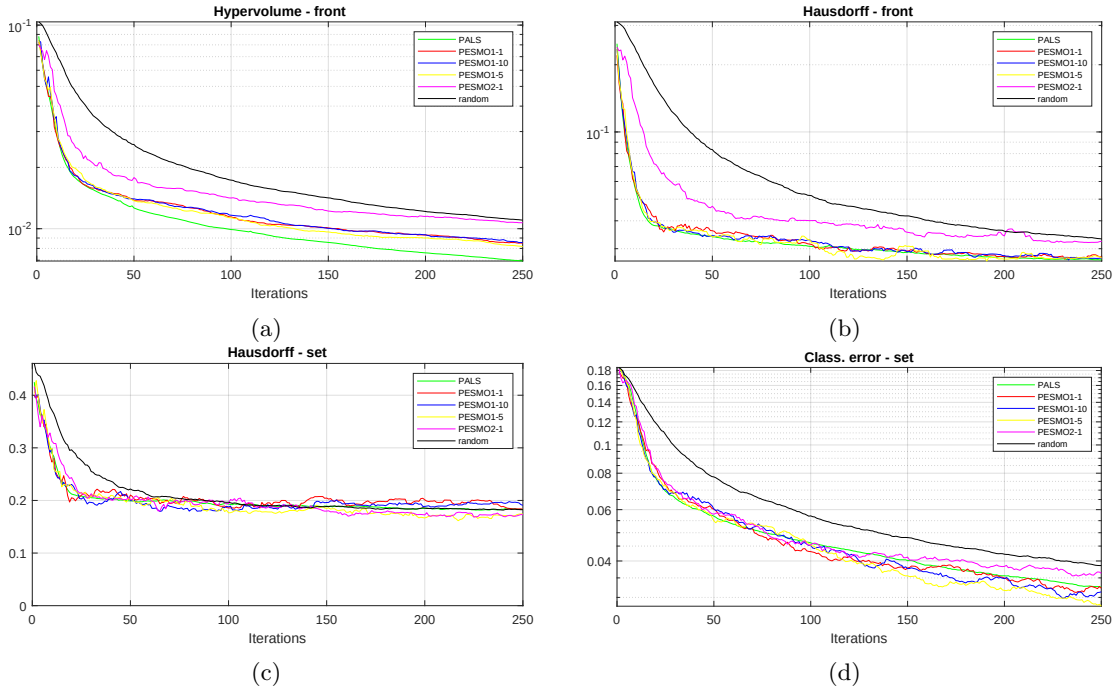


FIGURE 22 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

- Résultats sur test_b2

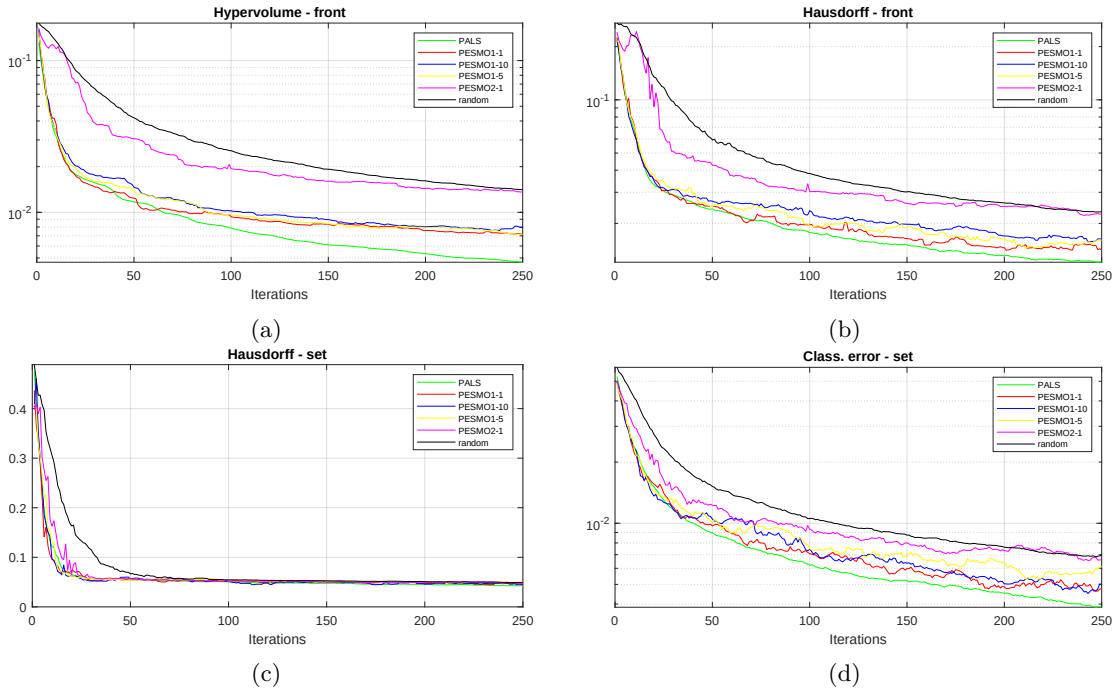


FIGURE 23 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

- Résultats sur test_b3

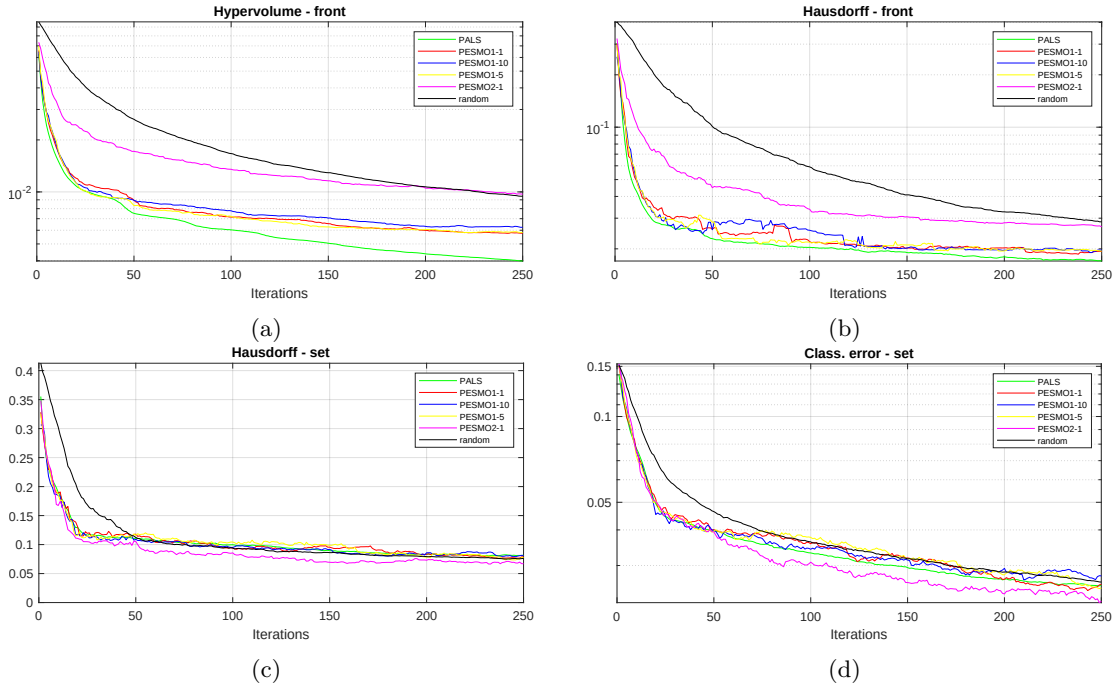


FIGURE 24 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

- Résultats sur test_b4

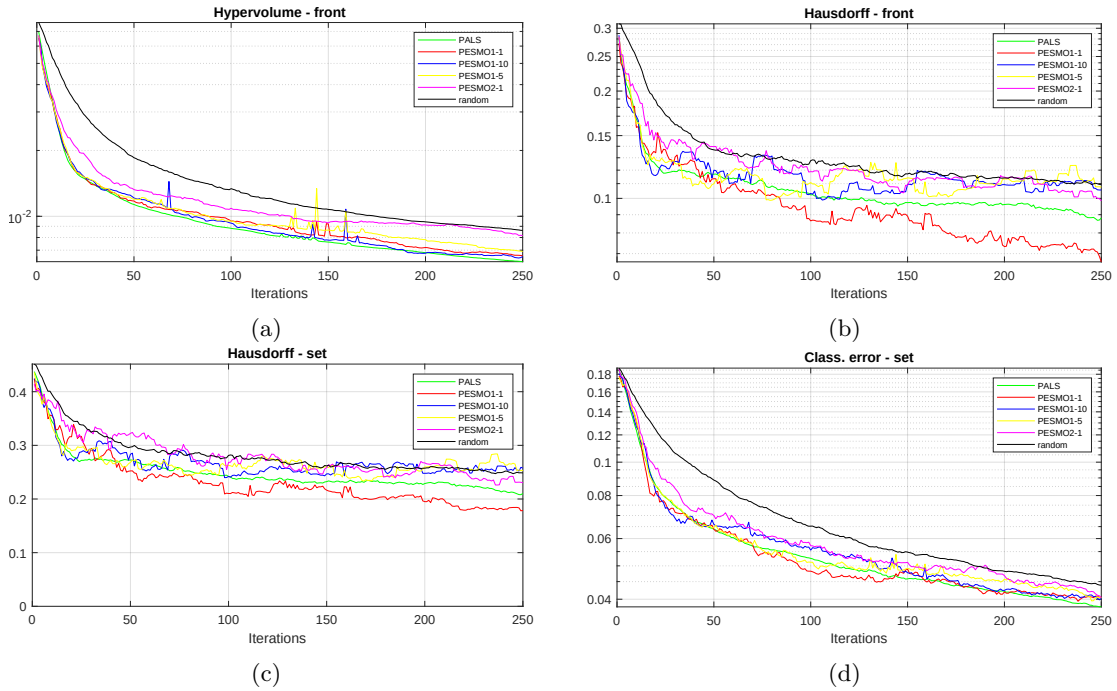


FIGURE 25 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

- Résultats sur test_b5

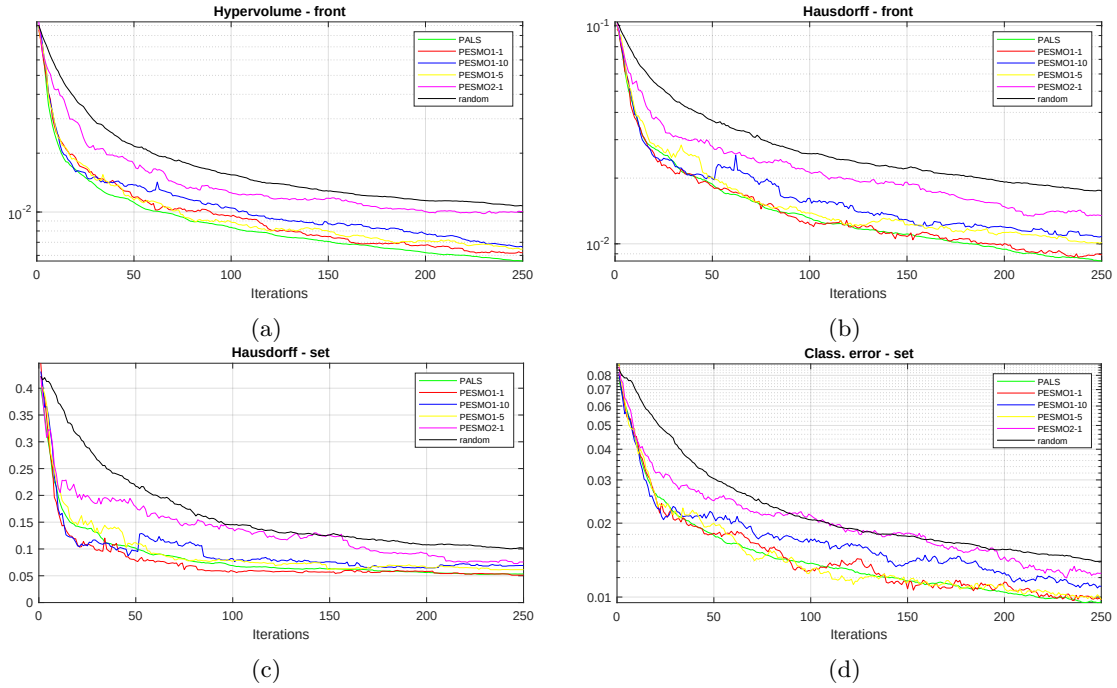


FIGURE 26 – Comparaison de plusieurs algorithmes pour les métriques moyennées sur 500 initialisations

Bibliographie

D. HERNÁNDEZ-LOBATO, J. M. HERNANDEZ-LOBATO, A. SHAH et R. ADAMS : Predictive entropy search for multi-objective bayesian optimization. *In International Conference on Machine Learning*, pages 1492–1501, 2016.

J. BECT, E. VAZQUEZ *et al.* : STK : a Small (Matlab/Octave) Toolbox for Kriging. Release 2.7.0, 2020. URL http://kriging.sourceforge.net.

H. DUTRIEUX : *Methods for the multi-year planning of distribution networks. Application to the techno-economic analysis of the solutions for integrating intermittent renewable energy sources.* Theses, Ecole Centrale de Lille, novembre 2015. URL https://tel.archives-ouvertes.fr/tel-01491053.

E. VAZQUEZ : *Modélisation comportementale de systèmes non-linéaires multivariables par méthodes à noyaux et applications.* Theses, Université Paris Sud - Paris XI, mai 2005. URL https://tel.archives-ouvertes.fr/tel-00010199.

D. JONES, M. SCHONLAU et W. WELCH : Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 12 1998.

J. VILLEMONTEIX : *Optimisation de fonctions coûteuses Modèles gaussiens pour une utilisation efficace du budget d'évaluations : théorie et pratique industrielle.* Theses, Université Paris Sud - Paris XI, décembre 2008. URL https://tel.archives-ouvertes.fr/tel-00351406.

P. HENNIG et C. J. SCHULER : Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 13(1):1809–1837, 2012.

S. BELAKARIA, A. DESHWAL et J. R. DOPPA : Max-value entropy search for multi-objective bayesian optimization. *In Advances in Neural Information Processing Systems*, pages 7825–7835, 2019.

F. Houlsby, N.and Huszar, Z. Ghahramani et J. M. Hernández-lobato : Collaborative gaussian processes for preference learning. *In* F. Pereira, C. J. C. Burges, L. Bottou et K. Q. Weinberger, éditeurs : *Advances in Neural Information Processing Systems 25*, pages 2096–2104. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4700-collaborative-gaussian-processes-for-preference-learning.pdf.

J. M. Hernández-Lobato, M. W. Hoffman et Z. Ghahramani : Predictive entropy search for efficient global optimization of black-box functions, 2014.

A. Shah : Predictive entropy search for multi-objective optimisation. *In Bayesian single- and multi-objective optimisation with nonparametric priors*, 2020. URL https://doi.org/10.17863/CAM.42311.

H. Dutrieux, I. Aleksovska, J. Bect, E. Vazquez, D. Gauthier et B. Francois : The Informational Approach to Global Optimization in presence of very noisy evaluation results. Application to the optimization of renewable energy integration strategies. *In 47èmes Journées de Statistique de la SFdS (JdS 2015)*, Lille, France, juin 2015. URL https://hal-supelec.archives-ouvertes.fr/hal-01163565.

M. Seeger : Expectation propagation for exponential families, 2008.

A. F. Lopez lopera : *Gaussian Process Modelling under Inequality Constraints*. Theses, Université de Lyon, septembre 2019. URL https://tel.archives-ouvertes.fr/tel-02863891.

J. P. Cunningham, P. Hennig et S. Lacoste-Julien : Gaussian probabilities and expectation propagation. *arXiv preprint arXiv :1111.6832*, 2011.

A. Vehtari, A. Gelman, T. Sivula, P. Jylänki, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich et C. Robert : Expectation propagation as a way of life : A framework for bayesian inference on partitioned data, 2014.

C. Robert et G. Casella : *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Z. I. Botev : The normal law under linear restrictions : simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 79(1):125–148, Feb 2016. ISSN 1369-7412. URL http://dx.doi.org/10.1111/rssb.12162.

A. Kraskov, H. Stögbauer et P. Grassberger : Estimating mutual information. *Physical Review E*, 69(6), Jun 2004. ISSN 1550-2376. URL http://dx.doi.org/10.1103/PhysRevE.69.066138.

M. Zuluaga, G. Sergent, A. Krause et M. Püschel : Active learning for multi-objective optimization. *In* Sanjoy Dasgupta et David McAllester, éditeurs : *Proceedings of the 30th International Conference on Machine Learning*, numéro 1 *in* Proceedings of Machine Learning Research, pages 462–470, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/zuluaga13.html.

L. Dempfle : Relation entre BLUP (Best Linear Unbiased Prediction) et estimateurs bayésiens (1). *Annales de génétique et de sélection animale*, 9(1):27–32, 1977. URL https://hal.archives-ouvertes.fr/hal-00892799.

M. L. Stein : *Interpolation of spatial data : some theory for kriging*. Springer Science & Business Media, 2012.