



HAL
open science

Optimisation multi-objectifs à base de métamodèle pour les procédés de mise en forme

Mohsen Ejday

► **To cite this version:**

Mohsen Ejday. Optimisation multi-objectifs à base de métamodèle pour les procédés de mise en forme. Matériaux. École Nationale Supérieure des Mines de Paris, 2011. Français. NNT : 2011ENMP0009 . pastel-00583805

HAL Id: pastel-00583805

<https://pastel.hal.science/pastel-00583805>

Submitted on 6 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale n°364 : Sciences Fondamentales et Appliquées

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris **Spécialité " Mécanique Numérique "**

présentée et soutenue publiquement par

Mohsen EJDAY

le 17 mars 2011

Optimisation Multi-Objectifs à base de Métamodèle **pour les Procédés de Mise en Forme**

Directeur de thèse : **Lionel FOURMENT**

Jury

M. Rodolphe LE RICHE , Docteur HDR, École des Mines de Saint-Étienne	Rapporteur
M. Piotr BREITKOPF , Docteur HDR, Université de Technologie de Compiègne	Rapporteur
M. Jean-Antoine DESIDERI , Professeur, INRIA Sophia	Examineur
M. Ying Qiao GUO , Professeur, Université de Reims Champagne-Ardenne	Examineur
M. Jean-Loup CHENOT , Professeur, CEMEF, Mines ParisTech	Examineur
M. Stéphane MARIE , Docteur, Projet LOGIC, Transvalor	Examineur

**T
H
È
S
E**

Avant-propos

Les travaux de recherche présentés dans ce mémoire ont été réalisés à l'École Nationale Supérieure des Mines de Paris (ENSMP), au Centre de Mise en Forme des Matériaux (CEMEF), au sein de l'équipe de Calcul Intensif en Mise en Forme (CIM).

Cette thèse est menée dans le cadre du projet LOGIC (Logiciel d'Optimisation de Gammes Intégré et Convivial) en partenariat avec Transvalor, Cetim, Atelier de la Haute Garonne, Atelier des Janves, Bollhof Otal, Industeel, Forges de Courcelles, Creas Ascometal, Gevelot Extrusion, Setforge et Arcelormittal

Remerciements

Mes sincères remerciements vont aux personnes qui ont contribué au bon déroulement et à l'aboutissement de cette thèse.

En premier lieu, je tiens à exprimer ma profonde reconnaissance à Monsieur Lionel Fourment, HDR à l'École Nationale Supérieure des Mines de Paris (ENSMP) au Centre de Mise en Forme des Matériaux (CEMEF), qui m'a donné la chance de réaliser ces travaux. Je tiens à lui remercier pour avoir accepté de diriger cette thèse avec une formidable compétence. Grâce à ses conseils et à son intérêt, il m'a été possible de mener cette thèse à son terme.

Je tiens aussi à dire merci à Monsieur Jean-Loup Chenot, Professeur à ENSMP au CEMEF, qui a accepté de remplacer Monsieur Lionel Fourment pour les corrections de ma thèse malgré les nombreuses activités dont il a la charge. Je remercie Monsieur Stéphane Marie de la société Transvalor et Monsieur Youssef Mesri, Docteur à l'IFP pour leurs aides. Je remercie aussi tous nos partenaires industriels.

Remerciements et profonde gratitude vont également aux membres de la commission d'examen. J'exprime ainsi toute ma reconnaissance à Monsieur Rodolphe Le Riche, HDR à l'École des Mines de Saint-Etienne, qui a accepté d'être rapporteur de ce mémoire. J'adresse aussi mes sincères remerciements à Monsieur Piotr Breitkopf, HDR à l'Université Technologique de Compiègne, qui a également accepté d'être rapporteur de ce travail. Mes remerciements à Monsieur Jean-Antoine Désidéri, Professeur à l'INRIA à Sophia-Antipolis, à Monsieur Ying Qiao Guo, Professeur à l'Université de Reims Champagne-Ardenne, à Monsieur Jean-Loup Chenot, Professeur à ENSMP au CEMEF et à Monsieur Stéphane Marie, Docteur à la société Transvalor qui ont accepté d'examiner mon travail.

C'est avec chaleur et sincérité que je salue les membres du laboratoire. J'exprime aussi mes remerciements pour mon collègue de bureau, Mohamad Ramadan.

Il me serait impossible, enfin, de ne pas dire un grand MERCI à ma femme HAJER et à ma petite fille YAKINE, ainsi qu'à ma famille, particulièrement mon cher père, pour le soutien et leurs encouragements qu'ils n'ont pas cessé, tout au long de cette période qui a été parfois très difficile. Je suis sûr que sans leur présence, je ne serais pas arrivé à finir cette thèse.

Enfin, je remercie Monsieur Frej Chaouachi, Maître de conférences à l'École Supérieure de Sciences et Techniques de Tunis.

Résumé

Dans le domaine de mise en forme, la réduction des coûts et l'amélioration des produits sont des défis permanents à relever. Pour ce faire, le procédé de mise en forme doit être optimisé. Optimiser le procédé revient alors à résoudre un problème d'optimisation. Généralement ce problème est un problème d'optimisation multi-objectifs très coûteux en terme de temps de calcul, où on cherche à minimiser plusieurs fonctions coût en présence d'un certain nombre de contraintes. Pour résoudre ce type de problème, on a développé un algorithme robuste, efficace et fiable. Cet algorithme, consiste à coupler un algorithme évolutionnaire avec un métamodèle, c'est-à-dire des approximations des résultats des simulations coûteuses.

Dans ce mémoire, on a commencé par la présentation du problème d'optimisation multi-objectifs, des algorithmes d'optimisation (algorithmes évolutionnaires) et des métamodèles les plus utilisés (Chapitre **I**), où on a choisi l'algorithme génétique élitiste de tri non dominé (NSGA-II) qui sera couplé avec le métamodèle basé sur la méthode de différence finies sans maillage. Ce métamodèle, qui est développé au cours de cette thèse, constitue le contenu de Chapitre **II**. Dans le Chapitre **III**, on étudie différentes manières de couplage de notre métamodèle avec NSGA-II. Ces couplages, C-Constant, C-Actualisé C-Évolutif-H1 et C-Évolutif-MC, diffèrent par le choix des points maîtres au cours des itérations, le type d'erreur utilisée et l'évolution du métamodèle. Ils sont validés sur plusieurs problèmes tests d'optimisation tels que des problèmes mono-objectif, multi-objectifs, sans contraintes et sous contraintes. Le meilleur couplage, C-Évolutif-MC, est utilisé pour la résolution des problèmes de mise en forme mono et multi-objectifs (Chapitre **IV**). Les résultats obtenus montrent l'efficacité de notre méthode.

Abstract

Cost saving and product improvement have always been important goals in the metal forming industry. To achieve them, metal forming processes need to be optimized, which consequently requires solving optimization problems. Usually these problems have multiple objectives; they aim at minimizing several objective functions in the presence of several explicit and implicit constraints. Each function evaluation is quite time consuming. To solve this type of problems, a robust and efficient algorithm is developed. It consists in coupling an evolutionary algorithm (a multi-objective genetic algorithm) with a metamodel (an approximation of the problem functions). Then, the number of expensive function evaluations can be significantly reduced by partly replacing exact evaluations with fast approximates of the objective functions.

In this report, we first present the multi-objective optimization problem, multi-objective optimization (evolutionary) algorithms and the most commonly used metamodels ([Chapter I](#)). We select the elitist non-dominated sorting genetic algorithm (NSGA-II) to be coupled with the metamodel based on the meshless finite difference method. This metamodel, which is presented and enhanced within this thesis, is presented in [Chapter II](#). In [Chapter III](#), different coupling strategies between NSGA-II and the metamodel are investigated, C-Constant, C-Actualise, C-Updated-H1 and C-Updated-MC. The different studied methods differ in the choice of master points (the exact evaluations of the objective functions), the evolution of the metamodel, and the utilized approximation error. They are studied and compared on several analytical functions, mono-objective functions, multi-objective functions and constrained functions. The best method, C-Updated-MC, is then used to optimize metal forming processes in [Chapter IV](#). The obtained results show the efficiency of our method.

Table des notations

Notations	Nom	Première apparition
\tilde{f}	Fonction coût approchée	I.4
$\Delta\tilde{f}$	Erreur d'approximation locale	I.4
$\mathbf{b} = \nabla f$	Gradient de la fonction f	II.2.1.1
$\mathbf{c} = \nabla^2 f$	Hessienne de la fonction f	II.2.2.1
$\bar{\gamma}$	Moyenne de la métrique de convergence	III.3.2.2
$\bar{\Delta}$	Moyenne de la métrique de diversité	III.3.2.2
Δf	Erreur exacte	II.5.5.1
$f_{\text{mérite}}$	Fonction mérite	I.4
E	Erreur de prédiction	I.6.2.2.2
$e(h)$	Erreur de discrétisation	II.5.5.1
f	Fonction coût exacte	I.2.2
F	Espace fonctionnel	I.2.2
g	Fonction de lissage	II.3.3
M	Nombre de fonctions coût	I.2.2
\mathbf{M}_t	Métamodèle, M-MDFSM à la génération t	III.2.1
n	Nombre de paramètres	I.2.2
N	Nombre de points approximés	II.5.5.1
N_u	Nombre maximal de générations où l'enrichissement est autorisé	III.2.2
P_t	Population parent à la génération t	I.5
Q_t	Population enfant à la génération t	I.5
R^2	Coefficient de détermination	I.6.2.2.2

R^2_A	Coefficient de détermination ajusté	I.6.2.2.2
S	Espace de décision	I.2.2
t	Numéro de la génération courante	I.5
$var(\varepsilon)=\sigma^2$	Variance de l'erreur	I.6.2
X	Vecteur de paramètres	I.2.2
\mathbf{X}	Vecteur des inconnues	II.2.1.1
\mathbf{A}	Matrice symétrique	II.2.1.1
\mathbf{B}	Vecteur second membre	II.2.1.1
E	Erreur moyenne quadratique	II.2
$E(\varepsilon)$	Moyenne de l'erreur	I.6.2.2.1
β	Coefficient de régression	I.6.2
ε	Terme d'erreur	I.6.2
η	Fonction de pondération	II.3.2
λ	Nombre initial de points maîtres	II.2
λ_{\max}	Nombre maximal de points maîtres	III.2.1
θ	Nombre de nouveaux points maîtres utilisés pour l'enrichissement du métamodèle M-MDFSM	III.2.2
ω	Fonction poids	II.2.1.1

Table des abréviations

Abréviations	Nom	Première apparition
AEMO	Algorithme Évolutionnaire Multi-Objectifs	I.3.3
AE	Algorithme Évolutionnaire	I.3
AG	Algorithme Génétique	I.3.2.1
ANN	Réseaux de neurones artificiels (Artificial Neural Networks)	I.1
ANOVA	Analyse de la variance	I.6.2.2.2
BFGS	Algorithme de quasi-Newton (Méthode de Broyden–Fletcher–Goldfarb–Shanno)	I.1
C-Actualisé	Couplage de NSGA-II avec le métamodèle évolutif, M-MDFSM. M-MDFSM est enrichi par l’ajout de nouveaux points maîtres durant un nombre fixe de générations. Ces nouveaux points maîtres sont les premiers individus de la population enfant courante.	III.2.2
C-Constant	Couplage de NSGA-II avec le métamodèle fixe, M-MDFSM	III.2.1
C-Évolutif-FM	Couplage de NSGA-II avec le métamodèle évolutif, M-MDFSM. M-MDFSM est enrichi par l’ajout de nouveaux points maîtres durant un nombre fixe de générations. Ces nouveaux points maîtres sont les meilleurs individus de la population parent classés dans l’ordre décroissant de leurs performances en utilisant une fonction mérite comme critère de choix	III.2.3
C-Évolutif-H1	C’est un C-Évolutif-FM qui utilise l’erreur H1 pour calculer la fonction mérite	III.3
C-Évolutif-MC	C’est un C-Évolutif-FM qui utilise l’erreur MC pour calculer la fonction mérite	III.3
Erreur EAP	Erreur a priori résultant d’un développement en séries de Taylor à l’ordre deux au voisinage d’un point inconnu	II.5.2
Erreur H1	Erreur d’approximation locale liée à un développement en séries de Taylor à l’ordre un au voisinage d’un point inconnu. Elle utilise la hessienne issue d’un développement en séries de Taylor à l’ordre deux	II.5.4
Erreur MC	Erreur d’approximation locale basée sur la méthode de moindres carrés. Elle utilise un développement en séries de	II.5.3

	Taylor à l'ordre deux au voisinage d'un point maître	
Erreur MC1	Erreur d'approximation locale basée sur la méthode de moindres carrés. Elle utilise un développement en séries de Taylor à l'ordre deux au voisinage d'un point inconnu	II.5.1
FBR	Fonctions à Bases Radiales	I.1
FOP	Front Optimal de Pareto	I.2.2
M-AEMO	Couplage de l'algorithme génétique élitiste de tri non dominé (NSGA-II) avec le métamodèle régularisé (M-MDFSM)	III.1
MDFSM	Méthode de Différences Finies Sans Maillage	II.1
MEF	Méthode des Éléments Finis	Introduction Générale
MLS	Moindres Carrés Mobiles	II.1
M-MDFSM	Métamodèle régularisé qui combine les avantages des deux modèles M-MDFSM-1 et M-MDFSM-2	II.2
M-MDFSM-1	Métamodèle linéaire basé sur un développement en séries de Taylor à l'ordre un	II.2
M-MDFSM-2	Métamodèle quadratique basé sur un développement en séries de Taylor à l'ordre deux	II.2
MSCE	Moyenne de la Somme des Carrés de l'Erreur	I.6.2.1
NSGA-II	Algorithme génétique élitiste de tri non dominé	Introduction Générale
POMO	Problème d'Optimisation Multi-objectifs	I.2.3
RMSE _{CV}	Racine carrée de la moyenne de l'erreur quadratique calculée par la technique de la validation croisée (Cross validation Root Mean Squared Error)	I.6.2.2.2
MSR	Méthodologie des Surfaces de Réponse	I.1
SCE	Somme des Carrés de l'Erreur	I.6.2.2.2
SCR	Somme des Carrés résultants de la Régression	I.6.2.2.2
SCT	Somme des Carrés Totaux	I.6.2.2.2
SE	Stratégie d'Évolution	I.3.2.2

SE-META	Stratégie d'Évolution à base de Métamodèle du krigeage	I.1
SOP	Ensemble Optimal de Pareto	I.2.2
SQP	Programmation quadratique séquentielle (Sequential Quadratic Programming)	I.1

Table des matières

Introduction Générale	14
Chapitre I Optimisation Multi-objectifs, Algorithmes Évolutionnaires et Métamodèles	16
I.1 Introduction	16
I.2 Optimisation multi-objectifs	19
I.2.1 Introduction	19
I.2.2 Définition	19
I.2.3 Buts de l'optimisation multi-objectifs	22
I.3 Algorithmes Évolutionnaires (AEs)	22
I.3.1 Principe de fonctionnement d'un algorithme évolutionnaire	23
I.3.2 Algorithme génétique et stratégie d'évolution	24
I.3.2.1 Algorithme génétique	24
I.3.2.1.1 Représentation	25
I.3.2.1.2 Sélection	25
I.3.2.1.3 Croisement	25
I.3.2.1.4 Mutation	26
I.3.2.1.5 Remplacement	26
I.3.2.2 Stratégie d'évolution	26
I.3.2.2.1 Sélection	26
I.3.2.2.2 Croisement	27
I.3.2.2.3 Mutation	27
I.3.3 Les AEs mono-objectif et les AEs multi-objectifs	27
I.4 SE-META	28
I.5 Algorithme Génétique Élitiste de Tri Non- Dominé (NSGA-II)	29
I.6 Métamodèles	30
I.6.1 Définition	30
I.6.2 La Méthodologie des Surfaces de Réponse (MSR)	30
I.6.2.1 Construction de la Méthodologie des Surfaces de Réponse	31
I.6.2.2 Validation de la Méthodologie des Surfaces de Réponse	33
I.6.2.2.1 Validation des hypothèses proposées pour la construction de la MSR	33
I.6.2.2.2 Validation de la précision de la MSR	34
I.6.3 Le Krigeage	36
I.6.3.1 Construction du krigeage	36
I.6.3.2 Validation du Krigeage	40
I.6.3.2.1 Validation des hypothèses	40
I.6.3.2.2 Validation de la précision du krigeage	40
I.6.4 Les Fonctions à Bases Radiales	40

I.7	Conclusion	41
Chapitre II	Métamodèle à base de Différences Finies Sans Maillage	42
II.1	Introduction	42
II.2	Construction du métamodèle	42
II.2.1	Métamodèle linéaire : M-MDFSM-1	43
II.2.1.1	Formulation du métamodèle linéaire	43
II.2.1.2	Performance du métamodèle linéaire	45
II.2.2	Métamodèle quadratique : M-MDFSM-2	48
II.2.2.1	Formulation du modèle quadratique	48
II.2.2.2	Performance du métamodèle quadratique	50
II.2.3	Approximation mixte : M-MDFSM	53
II.2.3.1	Formulation du métamodèle régularisé	54
II.2.3.2	Performance du métamodèle régularisé	55
II.3	Paramétrage du métamodèle M-MDFSM	56
II.3.1	Critère de singularité	56
II.3.2	Ordre de la fonction de pondération η	56
II.3.3	Valeur de la fonction de lissage g	58
II.4	Validation du métamodèle M-MDFSM	59
II.5	Étude de l'erreur du métamodèle M-MDFSM	61
II.5.1	Erreur MC1	62
II.5.2	Erreur EAP	63
II.5.3	Erreur MC	63
II.5.4	Erreur H1	65
II.5.5	Analyse de l'erreur	65
II.5.5.1	Convergence asymptotique	65
II.5.5.2	Application de critère de convergence asymptotique	66
II.6	Étude de bruit	68
II.6.1	Introduction	68
II.6.2	Métamodèle avec bruit	68
II.6.3	Application	71
II.7	Conclusion	74
Chapitre III	Couplage du M-MDFSM avec NSGA-II	75
III.1	Introduction	75
III.2	Couplage de NSGA-II avec le métamodèle M-MDFSM	75
III.2.1	Couplage constant : C-Constant	75
III.2.2	Couplage Actualisé aléatoirement: C-Actualisé	76
III.2.3	Couplage évolutif avec fonction mérite: C-Évolutif-FM	77
III.3	Problèmes tests analytiques	79

III.3.1	Problèmes tests d'optimisation mono-objectif	79
III.3.1.1	Fonction de Camel-Back	79
III.3.1.2	Fonction de Griewank	80
III.3.1.3	Fonction de Rastrigin	81
III.3.1.4	Fonction de Rosenbrock	81
III.3.2	Problèmes tests d'optimisation multi-objectifs	82
III.3.2.1	Construction d'un problème test	83
III.3.2.1.1	L'approche des fonctions mono-objectif multiple	83
III.3.2.1.2	L'approche ascendante	83
III.3.2.1.3	L'approche de surface avec contrainte	83
III.3.2.2	Critère de comparaison des algorithmes	84
III.3.2.3	Problèmes tests multi-objectifs	85
III.3.2.3.1	Min-Ex	85
III.3.2.3.2	Max-Ex	87
III.3.2.3.3	SCH1	89
III.3.2.3.4	SCH2	90
III.3.2.3.5	Const-Min-Ex	92
III.3.2.3.6	POL	96
III.3.2.3.7	VNT	97
III.3.2.3.8	TNK	98
III.3.2.3.9	KUR	99
III.4	Conclusion	101
Chapitre IV Optimisation des procédés de mise en forme		102
IV.1	Introduction	102
IV.2	Paramétrage de l'algorithme C-Évolutif-MC	102
IV.3	Paramétrage des procédés de mise en forme	103
IV.3.1	Procédé de remplissage d'une bielle	103
IV.3.2	Procédé de remplissage d'une couronne	104
IV.3.3	Procédé de tréfilage	104
IV.4	Optimisation du procédé de remplissage d'une bielle	105
IV.5	Optimisation du procédé de remplissage d'une couronne	110
IV.6	Optimisation du procédé de tréfilage	113
IV.6.1	Description	113
IV.6.2	Choix des paramètres d'optimisation	114
IV.6.3	Choix des fonctions coût	115
IV.6.4	Étude de l'optimisation du procédé de tréfilage	115
IV.6.4.1	Optimisation mono-objectif du procédé de tréfilage	116
IV.6.4.1.1	Tréfilage à deux paramètres	116

IV.6.4.1.2	Tréfilage à trois paramètres	118
IV.6.4.1.3	Tréfilage à quatre paramètres	120
IV.6.4.2	Optimisation multi-objectifs du procédé de tréfilage	121
IV.6.4.2.1	Tréfilage à deux paramètres	122
IV.6.4.2.2	Tréfilage à quatre paramètres	123
IV.7	Conclusion	125
	Conclusion générale	126
	Bibliographie	128

INTRODUCTION GENERALE

Les procédés de mise en forme (forgeage, emboutissage, tréfilage, fonderie, filage, laminage,...) connaissent une utilisation massive dans l'industrie. Ils sont fréquemment utilisés dans différents domaines, comme l'aéronautique et l'automobile. Les pièces obtenues par ces procédés doivent satisfaire à des propriétés géométriques, métallurgiques et à des caractéristiques de qualité. Pour répondre à ces exigences, plusieurs essais expérimentaux sont souvent nécessaires avant d'aboutir à une pièce satisfaisante. Toutefois, le coût très élevé d'un essai limite le nombre d'expériences, de sorte qu'avoir une pièce respectant les caractéristiques souhaitées et avec un coût acceptable devient une tâche difficile.

La réduction des coûts et l'amélioration des produits sont des défis permanents à relever. Pour ce faire, le procédé de mise en forme doit être optimisé. Au cours de ces dernières décennies, des logiciels de simulation basés sur des méthodes numériques – en particulier la Méthode des Éléments Finis (MEF) - ont largement contribué à la conception des procédés. Cette simulation numérique est un outil performant pour l'analyse, la compréhension et l'optimisation du procédé. Elle réduit le coût de ce dernier tout en évitant les expériences coûteuses. Elle offre la possibilité de suivre l'écoulement de la matière durant tout le processus. De plus, elle permet d'accéder à des informations difficiles à mesurer expérimentalement ; ce qui résulte en une amélioration de la compréhension du procédé et permet de l'analyser avant sa mise en œuvre effective.

Plus récemment, la possibilité du couplage d'un logiciel de simulation numérique avec des algorithmes d'optimisation, offre des opportunités très prometteuses pour la conception optimale des procédés. Optimiser le procédé revient alors à résoudre un problème d'optimisation. On rencontre deux types de problèmes. Le premier est celui mono-objectif, où l'on cherche une seule solution optimale. Le deuxième est un problème d'optimisation multi-objectifs, dans lequel on recherche un ensemble de solutions optimales, connu sous le nom d'ensemble optimal de Pareto.

Pour la résolution d'un problème d'optimisation, on souhaite utiliser des algorithmes robustes et efficaces, comme les algorithmes évolutionnaires multi-objectifs. Ces derniers permettent d'obtenir des optima globaux et ils sont capables de trouver un ensemble de solutions en une seule itération d'optimisation [Zitzler et al, 2000]. Le nombre important d'évaluation des fonctions qu'ils nécessitent est leur inconvénient majeur [Le Riche et al, 2007].

Dans la pratique, l'optimisation est parfois difficile à mettre en œuvre compte tenu du coût des calculs. Par exemple, pour les problèmes de mise en forme, une seule simulation (une seule évaluation de la fonction coût) peut prendre quelques heures ou plusieurs jours, même sur des machines de calcul parallèle. Cette charge de calcul est encore bien plus élevée pour les problèmes d'optimisation multi-objectifs, qui nécessitent davantage d'évaluations. Ce coût de calcul très élevé rend alors l'optimisation impossible.

Durant ces dernières décades, les méthodes numériques ne cessent d'être améliorées afin d'appliquer l'optimisation dans le secteur industriel et de répondre aux exigences demandées. Ces dernières conduisent à la recherche d'algorithmes robustes, efficaces et fiables pour optimiser des problèmes de plus en plus coûteux en terme de temps de calcul. Une famille de ces algorithmes, la plus répandue et celle qui offre les meilleurs résultats, consiste à coupler

des algorithmes évolutionnaires avec des métamodèles, c'est-à-dire des approximations des résultats des simulations coûteuses. Cette direction constitue le sujet de ce travail de recherche : l'optimisation multi-objectifs à base de métamodèle pour les procédés de mise en forme.

Ce mémoire de thèse est organisé de la manière suivante. Le premier chapitre (**Chapitre I**) est consacré à la présentation de l'optimisation multi-objectifs, des algorithmes d'optimisation (algorithmes évolutionnaires), en particulier les deux algorithmes SE-META et l'algorithme génétique élitiste de tri non dominé (NSGA-II), et des métamodèles les plus utilisés, tels que le krigeage, la Méthodologie des Surfaces de Réponse et les Fonctions à Bases Radiales. Un métamodèle basé sur la méthode de différences finies sans maillage, développé au cours de cette thèse, est ensuite détaillé au **Chapitre II**. Dans le **Chapitre III**, nous étudions les différentes manières d'effectuer le couplage de notre métamodèle avec l'algorithme NSGA-II développé par Kalyanmoy Deb et ses étudiants [Deb et al, 2000a] [Deb et al, 2002a] [Deb et al, 2000b]. Chaque approche est ensuite évaluée sur plusieurs fonctions tests analytiques (problèmes tests d'optimisation), constituant le cœur et la partie la plus importante de ce chapitre. Enfin, l'algorithme le plus performant est utilisé pour la résolution des cas industriels (**Chapitre IV**).

CHAPITRE I

OPTIMISATION MULTI-OBJECTIFS, ALGORITHMES ÉVOLUTIONNAIRES ET METAMODELES

I.1 INTRODUCTION

L'amélioration des produits et la réduction des coûts ont toujours été des objectifs importants dans le domaine de mise en forme. Ces dernières années, le couplage, entre des logiciels basés sur la méthode des éléments finis (MEF) et les techniques d'optimisation, a montré qu'il était possible d'apporter de nouvelles contributions à la conception des procédés [Bonte et al, 2008]. Le concept d'optimisation comprend deux phases (Figure 1). La première est la modélisation à partir de laquelle on définit la fonction coût (ou les fonctions coût), puis on sélectionne les variables principales d'optimisation et on détermine les contraintes. La deuxième phase est la résolution où on cherche à optimiser le problème en faisant appel à un algorithme d'optimisation [Bonte et al, 2010]

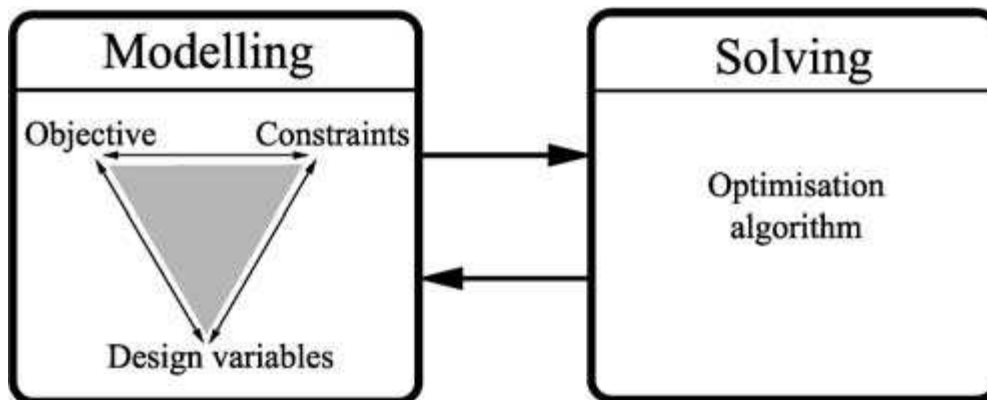


Figure 1 : concept de l'optimisation : modélisation et résolution [Bonte et al, 2007].

Durant ces dernières années, de nombreuses études ont été menées sur l'optimisation en mise en forme. Elles ont été concentrées sur la recherche d'algorithmes améliorés et adaptés à ce domaine d'application particulier. Selon le mode de couplage de l'algorithme d'optimisation avec la méthode des éléments finis (Figure 2), on distingue différents types d'approches:

Les algorithmes de gradient : Une façon d'optimiser le procédé de mise en forme est d'utiliser un algorithme classique, c'est-à-dire un algorithme d'optimisation mono-objectif basé sur le calcul du gradient de la fonction coût. Cet algorithme est couplé directement avec la méthode des éléments finis (Figure 2, a). Chaque itération de l'algorithme nécessite au moins une exécution du code éléments finis, et davantage si le gradient est calculé par différences finies.

Ces algorithmes sont bien connus et largement utilisés. En effet, leur convergence vers un optimum nécessite généralement un nombre relativement faible d'itérations. Ils sont largement utilisés pour l'optimisation mono-objectif. Par exemple, Jensen et al. [Jensen et al, 1998] ont utilisé l'algorithme de la plus forte pente pour optimiser l'usure de l'outil en emboutissage. Lin et al. [Lin et al, 2003] ont appliqué l'algorithme SQP (Programmation quadratique séquentielle) (en anglais Sequential Quadratic Programming) pour améliorer la durée de vie de l'outil en extrusion à chaud. Fourment et al. [Fourment et al, 1996] se sont servis de l'algorithme de quasi-Newton BFGS pour l'optimisation de forme en forgeage. Cao et al. [Cao et al, 2001] ont appliqué la méthode de recherche linéaire pour minimiser le nombre de passes durant le procédé d'emboutissage. Naceur et al. [Naceur et al, 2001] ont utilisé les deux méthodes SQP et BFGS pour optimiser la forme de l'outil sur le procédé de formage des tôles. Kim et al. [Kim et al, 2001] utilisent l'algorithme SQP pour optimiser la forme de l'outil sur le procédé d'emboutissage des tôles.

Les algorithmes de gradient ont quelques limitations. À cause de leur nature séquentielle, il n'est pas possible de mener en parallèle les simulations coûteuses des procédés de mise en forme. De plus, ces algorithmes nécessitent de connaître le gradient de la fonction coût. Or ce dernier n'est pas toujours disponible et ne peut pas toujours être calculé par différences finies. Enfin, ils ne garantissent pas la convergence vers un optimum global et peuvent être piégés dans un optimum local. Une autre difficulté tient à leur performance : un algorithme peut s'avérer très efficace sur un problème et peu efficace sur un autre [Kleinermann and Ponthot, 2003]

Les algorithmes évolutionnaires : Le deuxième groupe d'algorithmes couplés d'une manière directe avec des logiciels utilisant la méthode des éléments finis est celui des algorithmes génétiques et des stratégies d'évolution. Ces algorithmes évolutionnaires sont utilisés en raison de leur capacité à converger vers un optimum global, à permettre d'optimiser plusieurs fonctions coût en même temps (optimisation multi-objectifs) et à se prêter facilement au calcul parallèle. En revanche, le nombre important d'évaluations des fonctions qu'ils nécessitent en général apparaît comme leur inconvénient majeur [Emmerich et al, 2002]. Plusieurs publications sont l'objet du couplage entre les algorithmes évolutionnaires et un logiciel de la méthode des éléments finis pour optimiser des problèmes mono-objectif en mise en forme, voir par exemple Poursina et al. [Poursina et al, 2006], Kadkhodaei et al. [Kadkhodaei et al, 2007], Castro et al. [Castro et al, 2004], Schenk et Hillmann [Schenk and Hillmann, 2004] et Ledoux et al. [Ledoux et al, 2010].

Les algorithmes d'optimisation à base de métamodèle : La troisième approche pour l'optimisation des problèmes coûteux en terme d'évaluations des fonctions coût consiste à remplacer le couplage direct des algorithmes d'optimisation avec le logiciel de la méthode des éléments finis par un couplage indirect, en ajoutant un métamodèle entre les deux (Figure 2, b). Un métamodèle est un modèle approché construit à partir d'un ensemble de points « maîtres » (points auxquels la fonction coût est calculée de manière exacte par une simulation de la méthode des éléments finis) et permettant de remplacer un grand nombre d'évaluations exactes par leurs approximations. Il existe plusieurs types de métamodèle, parmi lesquels on cite la Méthodologie des Surfaces de Réponse (MSR) (en anglais Response Surface Methodology) (voir à titre d'exemple, Tunali et Batmaz [Tunali and Batmaz, 2000], Batmaz et Tunali [Batmaz and Tunali, 2003], Myers et al. [Myers et al, 2009], Jourdan [Jourdan, 2009] et Jin [Jin, 2005]), le Krigeage (voir à titre d'exemple, Jones [Jones, 2001],

Jones et al. [Jones et al, 1998], Ahmed et Qin [Ahmed and Qin, 2009] et Abderrahmane et Regis [Abderrahmane and Regis, 2009]), les réseaux de neurones artificiels (ANN, Artificial Neural Networks) (voir à titre d'exemple, Fonseca et al. [Fonseca et al, 2003] et Chambers et Mount-Campbell [Chambers and Mount-Campbell, 2002]), et les Fonctions à Bases Radiales (FBR) (voir à titre d'exemple, Rippa [Rippa, 1999], Li et al. [Li et al, 2008], Sobester et al. [Sobester et al, 2005] et Abderrahmane et Regis [Abderrahmane and Regis, 2009]).

L'utilisation d'un métamodèle permet de surmonter le problème du coût du temps de calcul lié au couplage direct, de sorte que les algorithmes d'optimisation à base de métamodèle sont très largement utilisés. La littérature en propose différentes approches pour les problèmes de mise en forme. De nombreuses publications concernent les problèmes d'optimisation mono-objectif suivant différentes techniques, tels que des algorithmes itératifs à base de métamodèle construit par approximations successives (voir à titre d'exemple, Bonte [Bonte, 2007], Bonte et al. [Bonte et al, 2007] [Bonte et al, 2008] [Bonte et al, 2010], Fourment [Fourment, 2007], Ben Ayed et al. [Ben Ayed et al, 2005], Oudjene et al. [Oudjene et al, 2009], Naceur et al. [Naceur et al, 2008] [Naceur et al, 2006], Breitkopf et al. [Breitkopf et al, 2005], Raza et Kim [Raza and Kim, 2008] et Do [Do, 2006]). Une autre approche consiste à coupler les algorithmes évolutionnaires avec des métamodèles, comme dans Wu et Hsu [Wu and Hsu, 2002], Abedrabbo et al. [Abedrabbo et al, 2004], Büche et al. [Büche et al, 2005], Naceur et al. [Naceur et al, 2004], Do [Do, 2006], Le Riche et al. [Le Riche et al, 2009] ou encore dans Emmerich et al. [Emmerich et al, 2002]

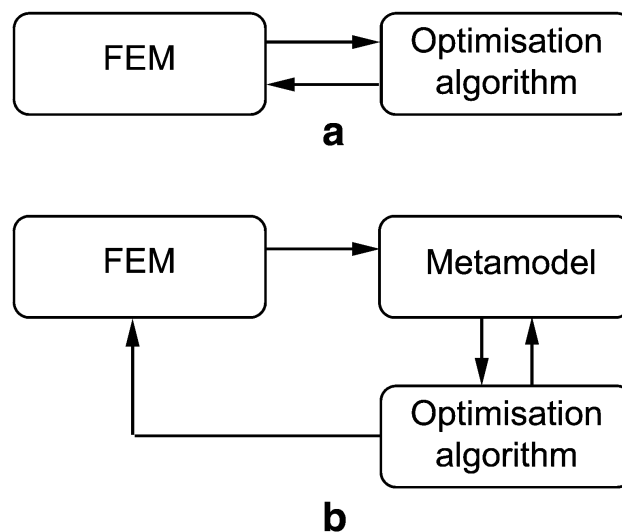


Figure 2 : (a) Optimisation directe et (b) optimisation approximée à base de métamodèle [Bonte et al, 2008].

Les problèmes industriels d'optimisation ont souvent plusieurs objectifs antagonistes qui doivent être améliorés en même temps, ce qui nécessite de se tourner vers une optimisation de type multi-objectifs. Elle permet de surmonter les inconvénients des approches reposant sur plusieurs optimisations mono-objectif successives; car l'optimisation vis-à-vis d'un objectif particulier permet un gain sur ce dernier souvent au prix d'une perte sur les autres objectifs. Prenons le cas d'école où on cherche à minimiser le coût et maximiser la qualité. Si on traite ce problème sous la forme de deux problèmes d'optimisation mono-objectif indépendants, le premier consiste à minimiser le coût et le deuxième à maximiser la qualité. Chaque problème permet d'optimiser sa fonction coût mais avec une perte au niveau de la fonction coût de l'autre problème, car les deux fonctions coût sont en conflit (la maximisation de la qualité est

en général accompagnée d'une augmentation du coût). L'optimisation multi-objectifs permet d'optimiser au mieux les différents objectifs en proposant toutes les solutions de compromis. Chacune est optimale vis-à-vis d'un certain nombre d'objectifs, pour une valeur donnée des autres objectifs. Durant ces dernières années, de nombreuses études ont été menées sur l'optimisation multi-objectifs, comme pour exemple celles de Wei et Yuying [Wei and Yuying, 2008], Ghanmi et al. [Ghanmi et al, 2005], Queipo et al. [Queipo et al, 2005], Emmerich et al. [Emmerich et al, 2006], Emmerich et Naujoks [Emmerich and Naujoks, 2004], Emmerich et al. [Emmerich et al, 2005], Beume et al. [Beume et al, 2007], Naujoks et al. [Naujoks et al, 2005], Goal et al. [Goal et al, 2007] [Goal et al, 2004] et Ingarao et al. [Ingarao et al, 2009].

Le reste de ce chapitre est structuré de la manière suivante : On commence par une présentation de l'optimisation multi-objectifs (partie I.2). Ensuite, une description des algorithmes évolutionnaires constitue la troisième partie, où les algorithmes génétiques et les stratégies d'évolution sont présentés. La présentation de la stratégie d'évolution à base de métamodèle, SE-META, fait l'objet de la quatrième partie. La cinquième partie est consacrée à la description de l'Algorithme Génétique élitiste de Tri Non- Dominé (NSGA-II). Dans la sixième partie, on étudie les trois métamodèles les plus utilisés, tels que la Méthodologie des Surfaces de Réponse, le krigeage et les Fonctions à Bases Radiales.

I.2 OPTIMISATION MULTI-OBJECTIFS

I.2.1 Introduction

L'optimisation multi-objectifs peut être définie comme la recherche de la meilleure ou des meilleures solutions possibles d'un problème donné. Souvent, les problèmes d'optimisation sont des problèmes multi-objectifs. Si les objectifs sont antagonistes, alors on n'a pas une seule solution optimale mais un ensemble de solutions de compromis.

I.2.2 Définition

Un problème d'optimisation multi-objectifs consiste à rechercher les meilleures solutions qui minimisent un nombre M de fonctions, appelées fonctions coût, $f_m(X)$, $m = 1, \dots, M$, par rapport à un vecteur X , qui est le vecteur des n variables de contrôle (ou paramètres) : $X = [x_1 \ x_2 \ \dots \ x_n]^T$, en satisfaisant un certain nombre de contraintes, explicites comme les contraintes de borne ($x_i^I \leq x_i \leq x_i^S$, $i = 1, \dots, n$) ou implicites, d'égalité ($h_k(X) = 0$, $k = 1, \dots, K$) ou d'inégalité ($g_j(X) \geq 0$, $j = 1, \dots, J$). On se limite aux problèmes de minimisation, puisque la maximisation d'une fonction $f(X)$ peut facilement être transformée en un problème de minimisation :

$$\max (f(X)) = -\min (-f(X)) \quad (\text{I.1})$$

Un problème d'optimisation peut s'écrire sous la forme générale suivante :

$$\begin{array}{l} \text{Minimiser} \quad f_m(X), \quad m=1, \dots, M; \\ \text{satisfaisant} \quad \begin{cases} g_j(x) \geq 0, \quad j=1, \dots, J; \\ h_k(x) = 0, \quad k=1, \dots, K; \\ x_i^I \leq x_i \leq x_i^S, \quad i=1, \dots, n; \end{cases} \end{array} \quad (\text{I.2})$$

Quand le problème d'optimisation contient une seule fonction coût ($M=1$), il est appelé mono-objectif. Ce problème, utilise un seul espace de recherche, connu aussi sous le nom d'*espace de décision* S . Il s'agit de l'ensemble des combinaisons possibles des paramètres qui satisfont toutes les contraintes. En revanche, si le problème d'optimisation contient plusieurs fonctions coût ($M > 1$), et si les fonctions coût sont antagonistes, il est appelé multi-objectifs [Deb, 2004]. Un problème d'optimisation multi-objectifs porte sur deux espaces, que sont l'espace de décision S et l'espace fonctionnel (l'espace des fonctions coût) F . Pour le résoudre (Eq (I.2)), un critère de comparaison des solutions est nécessaire. Ce critère est celui de *dominance* [Deb, 2001], illustré sur la Figure.3.

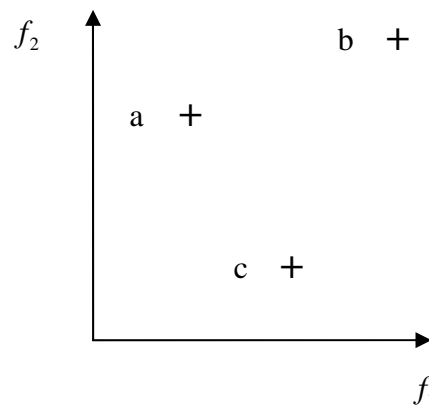


Figure.3– Illustration du critère de dominance pour un problème de minimisation à deux fonctions coût :
 la solution (a) domine la solution (b) car elle est meilleure pour les deux fonctions coût.
 La solution (a) est indifférente vis-à-vis de la solution (c) car chacune est meilleure pour une des deux fonctions coût [Büche, 2003].

Définition 1 Une solution X^1 est dite dominer une solution X^2 ($X^1 \succ X^2$), si et seulement si X^1 est meilleure ou égal à X^2 en toutes les fonctions coût et que X^1 est strictement meilleure à X^2 pour au moins en une fonction coût. Celui-ci peut être expliqué de la manière suivante :

$$X^1 \succ X^2 \text{ ssi } \begin{cases} \forall m \in \{1, 2, \dots, M\}: f_m(X^1) \leq f_m(X^2) \\ \exists m \in \{1, 2, \dots, M\}: f_m(X^1) < f_m(X^2) \end{cases} \quad (\text{I.3})$$

Définition 2 Les solutions X^1 et X^2 sont dites indifférentes si et seulement si aucune d'elles ne domine l'autre. Une solution X est dite non dominée s'il n'existe aucune solution $X^* \in S$ qui la domine.

Si on a des contraintes d'optimisation, on applique le critère de dominance avec contraintes.

Critère de dominance avec contraintes :

En présence des contraintes, on calcule les deux critères :

- Critère de violation de la contrainte j pour la solution X^i : $w_j(X^i)$
- Critère de violation des contraintes pour la solution X^i : $\Omega(X^i)$

Pour les J contraintes, on calcule pour chaque solution X^i :

$$w_j(X^i) = \begin{cases} |g_j(X^i)|, & \text{si } g_j(X^i) < 0 \\ 0, & \text{si non} \end{cases}$$

En suite, on calcule le critère de violation total :

$$\Omega(X^i) = \sum_{j=1}^J w_j(X^i)$$

Pour comparer deux solutions X^1 et X^2 , on suit la procédure suivante :

- 1- Si $\Omega(X^1) = \Omega(X^2)$, on utilise le **Critère de dominance** (Définition 1).
- 2- Si $\Omega(X^1) = 0$ et $\Omega(X^2) > 0$ (X^1 respecte les contraintes et X^2 viole les contraintes), on dit que la solution X^1 domine la solution X^2 .
- 3- Si $\Omega(X^1) > 0$ et $\Omega(X^2) > 0$ (X^1 et X^2 violent les contraintes) : la solution ayant Ω inférieure domine l'autre

Si l'on n'a pas de préférence parmi les fonctions coût, seul le critère de dominance (avec contraintes ou sans contraintes) permet de déterminer les solutions les plus performantes [Fonseca and. Fleming, 1995], et les solutions d'un problème d'optimisation multi-objectifs sont les solutions non dominées, qui forment l'ensemble optimal de Pareto (SOP). Suivant les liens existants entre les différentes fonctions, cet ensemble peut contenir une ou plusieurs solutions.

Définition 3 Deux fonctions coût sont dites corrélées, si leur ensemble optimal de Pareto ne contient qu'une seule solution. Elles sont antagonistes, si l'ensemble optimal de Pareto contient plus d'une solution.

La relation de corrélation entre les fonctions coût est illustrée sur la Figure 4. Pour des fonctions corrélées, optimiser une fonction permet d'optimiser les autres. Le problème est

similaire à un problème d'optimisation simple. En cas de fonctions antagonistes, il n'existe pas de solution unique, mais un ensemble de solutions optimales qui sont toutes d'importances comparables. Chacune représente un compromis entre les différentes fonctions : un gain sur un objectif se traduit par un sacrifice sur un autre objectif.

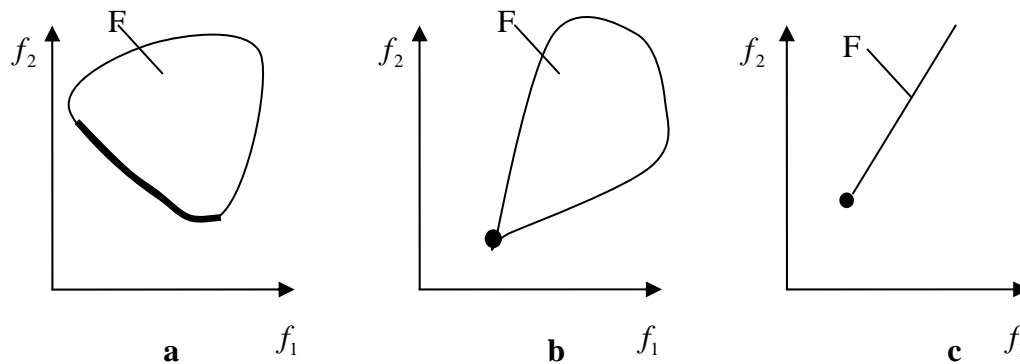


Figure 4– Illustration de la relation entre les fonctions coût d'un problème d'optimisation avec deux objectifs. L'ensemble optimal de Pareto est marqué en gras sur la frontière de l'ensemble des solutions possibles, F . (a) Les deux objectifs sont antagonistes et en (b) (c) ils sont corrélés.

Dans le cas où le nombre de fonctions coût est supérieur à deux ou trois, la visualisation du front optimal de Pareto (FOP) (une représentation graphique de SOP), et donc le choix d'une solution parmi toutes celles proposées, n'est pas aisée. De nombreux auteurs, dont Kalyanmoy Deb du Kânpur Genetic Algorithms Laboratory [Deb, 2001], proposent des procédures de choix d'une solution au sein de l'ensemble optimal de Pareto en exprimant des critères de priorité entre les fonctions.

I.2.3 Buts de l'optimisation multi-objectifs

D'une manière générale, la résolution d'un Problème d'optimisation multi-objectifs (POMO) consiste à atteindre deux objectifs : 1) trouver un ensemble de solutions qui converge vers le front optimal de Pareto en utilisant le critère de dominance, et 2) que cet ensemble de points soit suffisamment riche ou bien distribué, en se basant sur le critère de distance d'encombrement, pour représenter au mieux ce front optimal de Pareto.

I.3 ALGORITHMES ÉVOLUTIONNAIRES (AES)

Les algorithmes évolutionnaires (AEs) sont des méthodes stochastiques qui simulent le processus de l'évolution naturelle dans la résolution des problèmes d'optimisation [Zitzler et al, 2003]. Ils reposent sur l'analogie avec l'un des principes Darwiniens les plus connus : la survie de l'individu le mieux adapté [Lepadatu, 2006]. Ils sont basés sur la notion de « population d'individus », dans laquelle chaque individu représente une solution potentielle de l'espace de recherche. Ces algorithmes permettent de surmonter certains problèmes rencontrés avec les méthodes classiques. Ce sont des méthodes d'optimisation globales qui assurent la convergence vers les optima globaux malgré la présence d'optima locaux, et indépendamment de la répartition de la population initiale. Ils peuvent également

être utilisés pour des problèmes discrets et discontinus. En outre, en tant qu'algorithmes à base de population, leur parallélisation est aisée : il suffit de distribuer l'évaluation des fonctions coût sur autant de processeurs qu'il y a d'individus dans la population. Ils sont fréquemment utilisés à cause de leur robustesse et de leur souplesse, qui leur permettent d'aborder les problèmes les plus raides.

Historiquement, les algorithmes évolutionnaires sont utilisés depuis les années soixante. Ils se divisent en quatre catégories principales, voir la **Figure 5**.

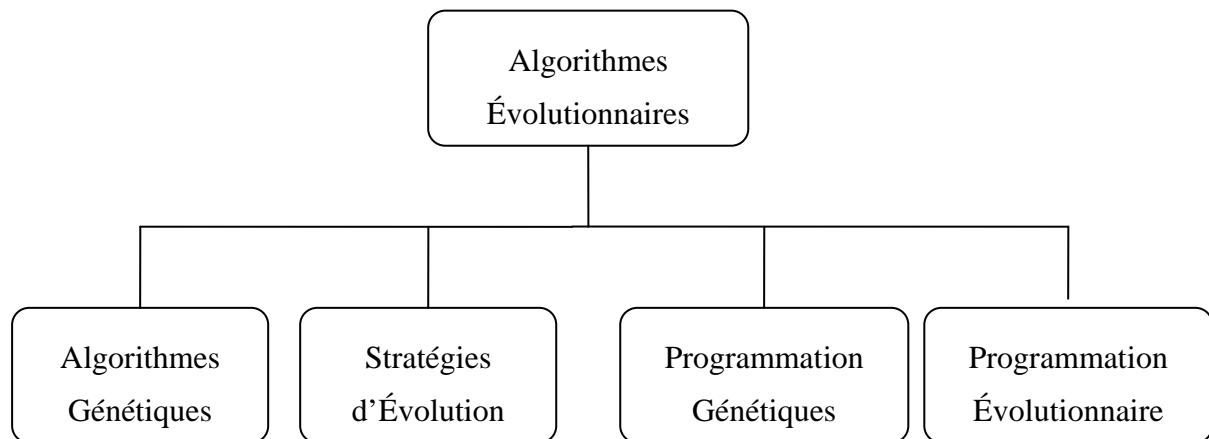


Figure 5– Principales catégories d'algorithmes évolutionnaires [Do, 2006]

Dans le reste de cette section, on présente la description du principe de fonctionnement d'un algorithme évolutionnaire, puis des algorithmes génétiques et des stratégies d'évolution, avant de discuter de la différence entre les AEs mono-objectif et multi-objectifs.

I.3.1 Principe de fonctionnement d'un algorithme évolutionnaire

Comme présenté par l'organigramme de la **Figure 6**, le principe de fonctionnement d'un AE est extrêmement simple. On part d'un ensemble initial d'individus (chacun correspondant à une valeur du jeu de paramètres), nommé « population parent initiale » et générée le plus souvent d'une manière aléatoire dans l'espace de recherche. Ensuite, on évalue de manière exacte la performance de chaque individu en calculant la valeur de la fonction coût correspondant à son jeu de paramètres. L'application des trois opérateurs génétiques de « sélection, croisement et mutation » permet de créer un nouvel ensemble d'individus appelé « population enfant ». Cette population est évaluée à son tour pour indiquer la performance de chacun de ses individus. Cette connaissance permet de décider lesquels des individus enfants méritent de remplacer certains parents. La nouvelle population obtenue, appelée « population parent », constitue la population parent de la nouvelle génération. Si les critères d'arrêt sont vérifiés, on considère la ou les solutions obtenues comme satisfaisantes, autrement, on recommence le cycle jusqu'à satisfaction de ces critères.

Le critère d'arrêt d'un AE peut être la convergence de l'ensemble des individus de la génération courante vers un même extremum, dans le cas d'un problème mono-objectif, ou alors vers un même ensemble optimal de Pareto, dans le cas multi-objectifs. Le plus souvent, l'algorithme est arrêté au bout d'un nombre d'itérations (générations) fixé a priori.

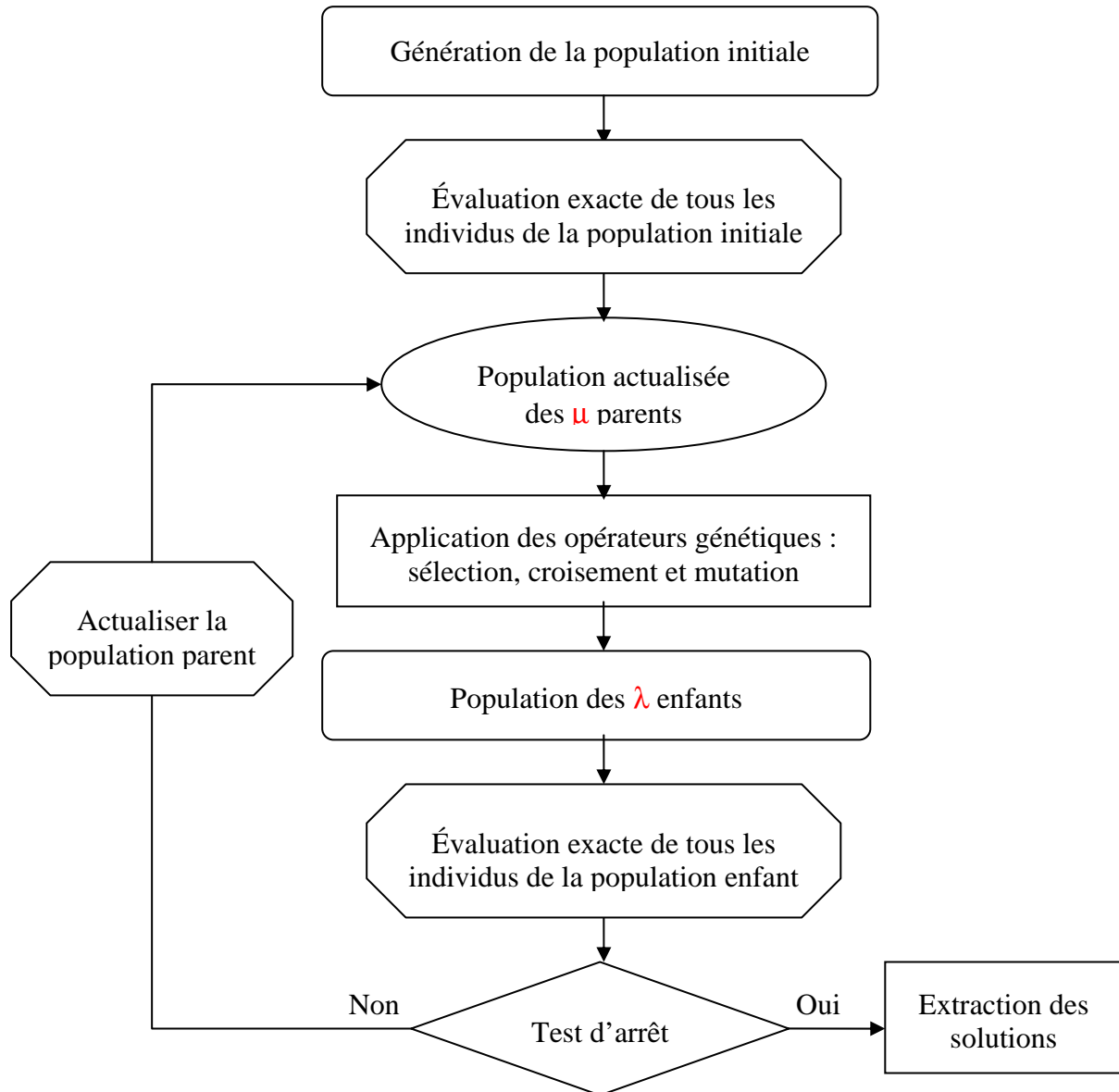


Figure 6– Organigramme de fonctionnement d’un algorithme évolutionnaire.

I.3.2 Algorithme génétique et stratégie d’évolution

I.3.2.1 Algorithme génétique

Les algorithmes génétiques (AGs) ont été mis au point par John Holland de l’université du Michigan aux États-Unis dans les années 60 [Holland, 1962]. Ils ont ensuite été raffinés par De Jong [De Jong, 1975] et popularisés par Goldberg [Goldberg, 1989] et Goldberg et Holland [Goldberg and Holland, 1988]. Suivant le type de codage utilisé, c’est-à-dire suivant la représentation des paramètres utilisée à la place des paramètres eux-mêmes, on distingue deux types d’algorithmes génétiques : les AGs avec un codage binaire et ceux avec un codage réel.

I.3.2.1.1 Représentation

Représentation binaire :

Le codage binaire consiste à représenter chaque vecteur des paramètres X par une chaîne de bits dont chaque instance prend la valeur 0 ou 1. Cette chaîne peut être une concaténation des différents paramètres à optimiser, chaque paramètre étant transformé en sa valeur binaire avec un nombre de bits fixé a priori. La **Figure 7** présente un exemple du codage binaire d'une solution avec trois paramètres, soit $X = (10, 6, 3)$, dans laquelle on utilise 5 bits pour coder chacun des trois paramètres.

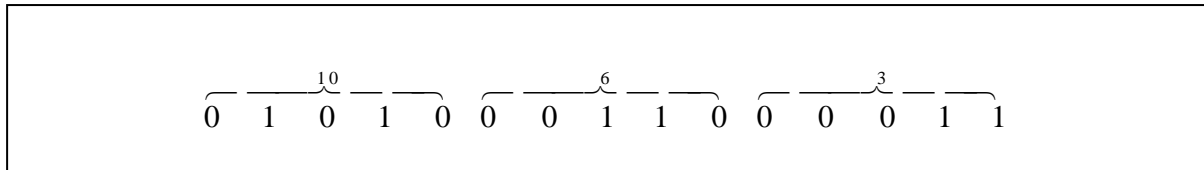


Figure 7- Exemple de codage binaire d'une solution potentielle avec trois paramètres

Ce codage s'adapte bien aux problèmes où les paramètres ont une valeur binaire, comme les problèmes booléens ou discrets, ou une valeur entière. Pour les problèmes d'optimisation ayant un espace de recherche continu, la précision de la solution dépend du nombre de bits affecté au codage (et fixé a priori).

Représentation réelle :

Le codage réel utilise directement la valeur réelle des paramètres. Selon Kalyanmoy Deb [Deb, 2001], dans le cas des problèmes d'optimisation avec des variables réelles, la résolution est plus efficace en utilisant des AGs avec un codage réel que binaire.

I.3.2.1.2 Sélection

La sélection est un opérateur essentiel pour améliorer la qualité d'une population. Son objectif est de retenir et multiplier les meilleurs individus qui contribueront à l'opération de croisement, et d'éliminer les plus mauvais, en accord avec la théorie de Darwin. Indépendamment du codage utilisé, la littérature propose différents types de sélection, comme celle par tournoi, proportionnelle (roulette) ou par rang. Pour plus de détails, le lecteur peut consulter Deb [Deb, 2001] et Do [Do, 2006].

I.3.2.1.3 Croisement

Le croisement est l'opérateur principal des algorithmes génétiques (AGs). Son rôle consiste à choisir aléatoirement deux individus parents parmi ceux sélectionnés pour les combiner et créer deux nouveaux individus enfants. Il joue un rôle important sur la convergence de l'AG, en lui permettant de concentrer une partie de la population autour des meilleurs individus. À chaque type de codage correspondent différents types de croisement. Pour les AGs avec un codage binaire, on utilise le croisement à un point (chaque bit du « génome » caractérisant le nouvel individu est celui de l'un de ses « parents » choisi aléatoirement) [Holland, 1962], à multipoints [De Jong and Spears, 1992] [Spears and De Jong, 1991] et le croisement uniforme [Syswerda, 1989]. Selon Deb [Deb, 2001], les AGs avec un codage réel peuvent utiliser la méthode de croisement linéaire (la valeur de l'individu enfant est une combinaison linéaire de celle de ses parents), naïf, mélangé ou encore binaire simulé.

I.3.2.1.4 Mutation

Les AGs utilisent l'opérateur de mutation pour conserver la diversité de la population et évite en particulier à l'AG de converger vers un optimum local [Gildemyn, 2008]. Pour les AGs avec un codage binaire, elle consiste à changer la valeur d'un ou plusieurs bits de l'individu de la population parent d'une valeur 1 à la valeur 0, ou réciproquement, avec une probabilité de mutation p_m fixée. Concernant les AGs avec un codage réel, il existe différentes méthodes de mutation telles que la mutation aléatoire, la mutation non uniforme, la mutation distribuée normalement et la mutation polynomiale [Deb, 2001].

I.3.2.1.5 Remplacement

Le remplacement consiste à déterminer quels sont les individus de la génération courante, constituée de parents et d'enfants, qui seront les parents de la génération suivante. Plusieurs stratégies de remplacement peuvent être utilisées :

Remplacement générationnel

La nouvelle population est composée uniquement des enfants de la génération précédente. L'inconvénient majeur de cette approche est la disparition des meilleurs individus, auxquels on ne permet pas de survivre, et donc la possibilité de perdre de bonnes solutions.

Remplacement continu

Cette approche consiste à remplacer les parents les moins performants par des enfants choisis aléatoirement.

Remplacement élitiste

L'opérateur d'élitisme [Deb, 2001] permet de conserver les tout meilleurs individus de la population courante. Il est essentiel pour l'optimisation multi-objectifs [Zitzler et al, 2000]. L'élitisme peut être utilisé de manière locale, lorsque deux enfants viennent d'être créés : ils sont comparés à leurs parents. Parmi les quatre individus parents et enfants, seuls les deux meilleurs sont conservés ; les parents sont en concurrence directe avec leurs enfants. Il peut également être utilisé d'une manière globale : une fois la population enfant créée, elle est combinée avec la population parent, et seuls les meilleurs individus sont retenus pour former la nouvelle population. Ainsi, une bonne solution trouvée très tôt dans l'optimisation n'est pas perdue même si une autre meilleure solution est trouvée. Donc l'élitisme accroît la probabilité de créer des meilleurs enfants et d'assurer une convergence rapide vers la solution optimale.

I.3.2.2 Stratégie d'évolution

La méthode d'optimisation à base de stratégies d'évolution (SE) fut initialement proposée par Ingo Rechenberg, en 1965, [Rechenberg, 1965] à l'Université Technique de Berlin, en Allemagne. Il l'a ensuite développée en collaboration avec Hans-Paul Schwefel. Cette approche s'appuie sur une représentation en nombres réels. Elle utilise un opérateur de mutation gaussienne. Pour plus de détails, le lecteur peut consulter [Beyer and Schwefel, 2002]. En tant qu'AE, une stratégie d'évolution est déterminée par les trois opérateurs évolutionnaires : sélection, croisement (ou recombinaison) et mutation. Elle commence par la génération d'une population parent initiale, de taille μ . À la génération t , la population enfant de taille λ est obtenue en appliquant les opérateurs génétiques.

I.3.2.2.1 Sélection

La sélection augmente la performance d'une population en retenant les meilleurs individus afin de guider la recherche vers des régions prometteuses. Elle repose sur un processus déterministe qui transmet les meilleurs μ individus de la génération courante à la nouvelle génération. Selon la prise en compte de la population parent ou non, on distingue deux types de SE :

$(\mu + \lambda)$ -SE : On sélectionne les meilleurs μ individus parmi la population des parent et des enfants de la population courante pour créer la population parent de la nouvelle génération. Ce type de sélection est élitiste. Il garantit la survie des meilleurs individus et l'amélioration de la population.

(μ, λ) -SE : La sélection des individus de la nouvelle population parent se limite à la population enfant. Beyer et Schwefel [Beyer and Schwefel, 2002] considèrent que l'utilisation d'une population enfant de taille $\lambda > \mu$ est nécessaire pour la convergence de cette méthode vers la solution optimale.

I.3.2.2.2 Croisement

Le croisement est facultatif pour les SEs. Il combine les informations des individus d'une population pour créer des autres individus. Contrairement aux AGs, le croisement peut utiliser plus que deux individus parents (ρ individus) choisis aléatoirement pour créer un seul individu enfant. On rencontre deux classes de recombinaison utilisées par les SEs :

La recombinaison discrète, appelée aussi recombinaison dominante, consiste à choisir les différentes variables $X_r = [x_r^1, x_r^2, \dots, x_r^n]$ caractérisant l'individu enfant de manière aléatoire parmi les ρ individus parents.

La recombinaison intermédiaire : Ce croisement prend en compte les ρ individus parents de manière équivalente : l'individu recombiné $X_r = [x_r^1, x_r^2, \dots, x_r^n]$ est le centre des ρ individus parents :

$$\forall k, \quad x_r^k = \frac{1}{\rho} \sum_{j=1}^{\rho} x_j^k \quad (\text{I.4})$$

Où x_j^k est la $k^{\text{ième}}$ variable définissant le $j^{\text{ième}}$ individu parent.

I.3.2.2.3 Mutation

L'opérateur de mutation est l'opérateur de base d'une stratégie d'évolution donc il est la source principale de la variation génétique. Il garantit à la fois la globalité de la recherche, en permettant d'explorer l'ensemble de l'espace de recherche, et la convergence vers un optimum, en utilisant, le plus souvent, une mutation gaussienne auto-adaptative [Do, 2006].

I.3.3 Les AEs mono-objectif et les AEs multi-objectifs

On distingue deux catégories d'AEs, les mono-objectif et les multi-objectifs. Selon Justesen [Justesen, 2009], la plus grande différence entre ces deux familles, est que dans le cas mono-objectif, il est simple de retourner la solution optimale d'une population, celle ayant la valeur

minimale de la fonction coût, alors que dans le cas multi-objectifs (AEMO – Algorithme Évolutionnaire Multi-Objectifs), la situation est très différente. Suivant la dimension de l'espace fonctionnel, F , les individus d'une population peuvent être incomparables entre eux, chacun représentant un compromis entre plusieurs objectifs. Le résultat d'un AEMO est donc en général l'ensemble des solutions non dominées (SOP).

I.4 SE-META

Dans ce paragraphe, on présente l'algorithme, SE-META, actuellement intégré dans le logiciel éléments finis, forge 2009, pour résoudre les problèmes d'optimisation mono-objectif de mise en forme. L'algorithme SE-META (en anglais Metamodel Assisted Evolution Strategies (MAES)) est un couplage entre une $(\mu + \lambda)$ -SE et le métamodèle de Krigage pour réduire le nombre d'évaluations exactes de la fonction coût. Il est proposé par Emmerich et al. [Emmerich et al, 2002] et est également décrit dans [Fourment et al, 2009], [Fourment et al, 2010] et [Massé, 2010]. Un aperçu de SE-META est représenté sur la Figure 8. SE-META commence par un choix aléatoire d'une population initiale de taille deux fois le nombre de paramètres d'optimisation ($2n$) pour construire le métamodèle initial. Il utilise une population parent de taille, μ , fixée à 2 fois le nombre de paramètres d'optimisation, tandis que le nombre d'enfants est $\lambda = \max(20, 3\mu)$.

Après le calcul exact de la population initiale, les μ individus sont utilisés pour donner les λ enfants. Ces derniers sont approximés par le métamodèle initial. Les valeurs de la fonction coût f ne sont pas directement approchées par \tilde{f} , mais par la fonction mérite $f_{\text{mérite}} = \tilde{f} - \Delta\tilde{f}$ où $\Delta\tilde{f}$ est l'erreur d'approximation locale du modèle de Krigage (Eq (38)). La fonction mérite représente une estimation de la valeur la plus basse possible de f . Les meilleurs μ individus ayant $\min(\tilde{f} - \Delta\tilde{f})$ parmi les individus parents et enfants seront évalués exactement pour enrichir le métamodèle de krigage, et cette procédure est répétée jusqu'à ce que le nombre maximal de calculs exacts soit atteint.

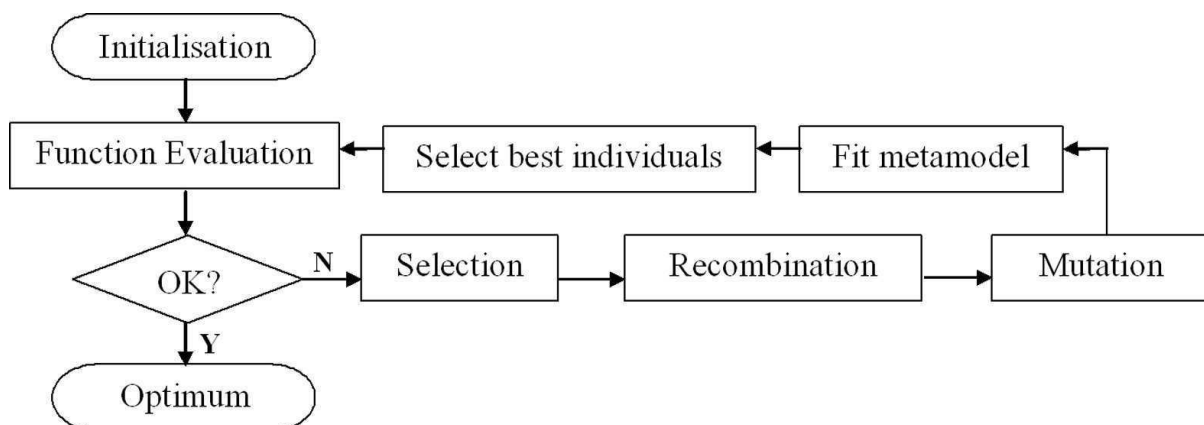


Figure 8– Fonctionnement de SE-META [Bonte et al, 2010]

I.5 ALGORITHME GENETIQUE ÉLITISTE DE TRI NON-DOMINE (NSGA-II)

Dans cette partie, on présente l'Algorithme Génétique élitiste de Tri Non-Dominé (Elitist Non-Dominated Sorting Genetic Algorithm), NSGA-II, qui sera utilisé dans la suite de cette thèse. NSGA-II, a été développé par Kalyanmoy Deb et ses étudiants à partir de 2000 ([Deb et al, 2000a] [Deb et al, 2002a]). Il est également décrit dans [Deb, 2001] et [Goal et al, 2007], et apparaît comme l'un des algorithmes de référence pour trouver l'ensemble optimal de Pareto avec une excellente variété de solutions. Il est certainement le plus populaire algorithme génétique multi-objectifs [Justesen, 2009]. Il est basé sur les trois caractéristiques suivantes : il utilise le principe de l'élitisme, il favorise les solutions non dominées, et il utilise une variété explicite des solutions, grâce au critère de distance d'encombrement.

Définition 4 : *Pour estimer la densité des solutions autour d'une solution particulière i de la population courante, on utilise la distance moyenne normalisée des deux solutions les plus proches de i , dans les sens croissant et décroissant, et pour chacune des fonctions coût. Cette quantité d_i , est appelée **distance d'encombrement**. Elle constitue une estimation du périmètre de l'hyper cube formée par les plus proches voisins de i .*

L'algorithme suivant est utilisé pour calculer la distance d'encombrement de chaque point d'un ensemble Γ .

Procédure de calcul de la distance d'encombrement :

Étape 1 : soit $l = |\Gamma|$ le nombre d'individus de Γ . Pour tout $i = 1, \dots, l$, initialiser $d_i = 0$.

Étape 2 : Pour chaque fonction coût, c'est-à-dire pour tout $m = 1, \dots, M$, trier cet ensemble Γ suivant l'ordre décroissant de la fonction f_m en construisant le vecteur $I^m = \text{trier}(\Gamma, f_m, >)$ de classement.

Étape 3 : Pour tout $m = 1, \dots, M$, attribuer une grande valeur numérique à la distance d'encombrement des solutions extrêmes (c'est-à-dire correspondant à une valeur minimale ou maximale de f_m) : $d_{I_1^m} = d_{I_l^m} = \infty$, et pour les autres individus, c'est-à-dire pour tout $j = 2$ à $(l-1)$ calculer :

$$d_j = \sum_{m=1}^M d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{\max} - f_m^{\min}} \quad (\text{I.5})$$

Où f_m^{\max} et f_m^{\min} sont respectivement la valeur maximale et minimale de la fonction f_m et $f_m^{I_j^m}$ est la valeur de la fonction f_m du vecteur I^m de l'individu j .

L'algorithme génétique NSGA-II commence par une génération aléatoire d'une population initiale P_0 de N individus parents. Ensuite, cette population est classée en plusieurs fronts de rangs différents, de la manière suivante. Chaque individu de P_0 est comparé à tous les autres individus par le concept de dominance. Les individus non dominés appartiennent au front de

rang 1, $r=1$, le front optimal de Pareto. En éliminant, temporairement, ces individus de l'ensemble de recherche, l'algorithme est relancé pour fournir le front de rang 2, $r=2$, et ainsi de suite. Puis, pour chaque solution de chaque front, on calcule sa distance d'encombrement. Ensuite, on applique le critère de sélection. Ici la *sélection par tournoi* est utilisée. Elle consiste à tirer aléatoirement deux solutions de la population parent et à garder la solution ayant le rang inférieur. Si les deux solutions ont le même rang, on garde celle ayant la plus grande distance d'encombrement. Après cette phase de sélection, les deux autres opérateurs évolutionnaires de croisement et de mutation sont appliqués pour créer la population enfant Q_0 de N individus. Aux générations suivantes t , on applique de même les opérateurs de sélection par tournoi, de croisement et de mutation à la population parent P_t de taille N pour générer une population enfant Q_t de N individus. Ensuite, les deux populations sont combinées pour former une nouvelle population R_t de taille $2N$. La recherche des solutions non dominées permet de classer les individus de R_t en plusieurs fronts de rangs différents. La nouvelle population parent P_{t+1} est alors construite avec les N meilleurs individus de la population R_t , i.e. les individus appartenant aux fronts de rang les plus faibles. Pour le dernier front, il y a généralement plus de solutions que de places restantes dans la nouvelle population P_{t+1} . Les individus sont alors triés selon leur distance d'encombrement et par ordre décroissant. Ce choix permet d'offrir la meilleure distribution des individus sur le front de rang le plus élevé.

Les algorithmes évolutionnaires multi-objectifs (AEMOs) apparaissent comme les techniques les plus générales et les plus efficaces pour la détermination de l'ensemble optimal de Pareto avec une bonne diversité de points, même pour des problèmes particulièrement difficiles [Zitzler et al, 2003]. Cependant, ils nécessitent un nombre important d'évaluations des fonctions coût, ce qui rend ces approches inutilisables pour des problèmes gourmands en terme de temps de calcul, comme en mise en forme des matériaux, le cadre de notre étude. Ce nombre important d'évaluation des fonctions peut être considérablement réduit en remplaçant une grande partie de celles-ci par des approximations construites à partir d'un métamodèle.

I.6 METAMODELES

I.6.1 Définition

Le terme «métamodèle» caractérise l'utilisation d'une équation mathématique pour exprimer la relation entre les variables d'entrée et de sortie d'un modèle complexe [Tunali and Batmaz, 2000]. Il permet de remplacer l'évaluation exacte des fonctions par des approximations, en connaissant la valeur exacte de ces fonctions en certain nombre de points, connus sous le nom de *points maîtres*. Il permet également de se faire une idée sur l'allure des fonctions étudiées, et de comprendre l'influence des variables d'entrées (ou de décision) sur les variables de sortie (les fonctions coût ou les contraintes), jouant ainsi un rôle important pour l'analyse du problème d'optimisation [Queipo et al, 2005].

Dans cette thèse, on présente les trois métamodèles les plus utilisées : la Méthodologie des Surfaces de Réponse (MSR), le Krigeage et les Fonctions à Bases Radiales (FBR).

I.6.2 La Méthodologie des Surfaces de Réponse (MSR)

En se référant à Myers et al. [Myers et al, 2009], la Méthodologie des Surfaces de Réponse (MSR) (en anglais : Response Surface Methodology) est une combinaison des techniques statistiques et mathématiques utiles pour le développement, l'amélioration et l'optimisation

des processus. Cette méthode est largement utilisée dans le monde industriel, particulièrement lorsque plusieurs variables d'entrée influencent les résultats (les variables de sortie). La Méthodologie des Surfaces de Réponse est connue sous d'autres noms, par exemple dans [Queipo et al, 2005] elle est nommée « Modèle de régression polynomiale » et dans [Jin, 2005] « Modèle polynomial ». Dans la suite de cette section, on présente la construction du modèle puis ses méthodes de validation.

I.6.2.1 Construction de la Méthodologie des Surfaces de Réponse

La Méthodologie des Surfaces de Réponse (MSR) consiste à représenter une réponse f comme la somme d'un métamodèle polynomial de faible degré (ordre un ou deux) et d'un terme d'erreur ε ayant une distribution normale avec une moyenne nulle ($E(\varepsilon) = 0$) et une variance égal à σ^2 . Si on considère un modèle polynomial d'ordre deux avec deux variables d'entrée (de régression), x_1 et x_2 , la réponse f sera représentée de la manière suivante :

$$f = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \quad (\text{I.6})$$

Si on pose le changement de variable suivant $x_1^2 = x_3$, $x_2^2 = x_4$, $x_1 x_2 = x_5$, $\beta_{11} = \beta_3$, $\beta_{22} = \beta_4$ et $\beta_{12} = \beta_5$, la relation entre la sortie f et les 5 variables de régression s'écrit :

$$f = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \varepsilon \quad (\text{I.7})$$

L'équation (Eq (I.7)) est appelée *modèle de régression linéaire multiple* [Myers et al, 2009] et les coefficients β_i , $i = 0, \dots, 5$ sont appelés des *coefficients de régression*. D'une manière générale, f s'écrit en fonction des n variables de régression (ici, $n = 5$):

$$f = \beta_0 + \sum_{j=1}^n \beta_j x_j + \varepsilon \quad (\text{I.8})$$

L'équation (Eq (I.8)) est un modèle de régression linéaire en β . En général, chaque modèle de régression linéaire en β est un modèle de régression linéaire quelque soit la forme de la surface de réponse qu'il génère.

Si on suppose maintenant que l'on connaît les valeurs des $m > n$ réponses de la fonction f à m valeurs différentes des variables d'entrées, pour chacune des i observations, on note f_i la valeur de la réponse et x_{ij} la $j^{\text{ème}}$ variable de régression de l'instance $X_i = (x_{ij})_{j=1,n}$. En considérant alors que la moyenne de l'erreur est nulle $E(\varepsilon) = 0$ et que sa variance vaut

$\text{var}(\varepsilon) = \sigma^2$, la solution f_i a une moyenne égal à $E(f_i) = \beta_0 + \sum_{j=1}^n \beta_j x_{ij}$ et une variance égal à $\text{var}(f_i) = \sigma^2$, et donc :

$$f_i = \beta_0 + \sum_{j=1}^n \beta_j x_{ij} + \varepsilon_i, \quad \forall i = 1, \dots, m \quad (\text{I.9})$$

Sous forme matricielle, l'équation (Eq (I.9)) s'écrit :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (\text{I.10})$$

$$\text{Avec: } \mathbf{Y} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}, \text{ et } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_m \end{bmatrix}.$$

Une valeur approchée, $\tilde{\boldsymbol{\beta}}$, du vecteur des coefficients de régression peut être déterminée en minimisant la fonction des moindres carrés de l'erreur, L , par rapport aux $\beta_{j=0, \dots, n}$:

$$L = \sum_{i=1}^n \varepsilon_i^2 = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \quad (\text{I.11})$$

D'où :

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (\text{I.12})$$

Une fois les coefficients de régression estimés, le métamodèle, Méthodologie des Surfaces de Réponse (MSR), devient une fonction mathématique explicite qui peut être utilisée pour déterminer une valeur approchée \tilde{f}_0 de la réponse f_0 en tout point X_0 :

$$\tilde{f}_0 = X_0^T \tilde{\boldsymbol{\beta}} \quad (\text{I.13})$$

Avec $X_0^T = [1 \ x_{01} \ x_{02} \ \dots \ x_{0n}]$. Il est aussi possible de déterminer la variance de la réponse \tilde{f}_0 [Myers et al, 2009] :

$$\text{var}(\tilde{f}_0) = \sigma^2 X_0^T (\mathbf{X}^T \mathbf{X})^{-1} X_0 \quad (\text{I.14})$$

La variance σ^2 peut être estimée comme *la Moyenne de la Somme des Carrés des résidus ou de l'Erreur : MSCE*

$$\tilde{\sigma}^2 = MSCE = \frac{\sum_{i=1}^m (f_i - \tilde{f}_i)^2}{m - (k + 1)} \quad (\text{I.15})$$

Où k est le nombre des coefficients de régression -1 , i.e. le nombre de variables ($k = n$).

I.6.2.2 Validation de la Méthodologie des Surfaces de Réponse

Pour valider le métamodèle, Méthodologie des Surfaces de Réponse, il faut en considérer ses deux aspects :

- Les hypothèses proposées pour la construction du métamodèle.
- La précision du métamodèle en fonction des mesures.

I.6.2.2.1 Validation des hypothèses proposées pour la construction de la MSR

Les hypothèses proposées pour la construction du métamodèle, MSR, sont [Bonte, 2007] [Tunali and Batmaz, 2000] :

1. Les réponses doivent dépendre des variables de régression d'une manière linéaire et additive.
2. La moyenne de l'erreur est nulle : $E(\varepsilon) = 0$
3. La variance de l'erreur est homogène : elle est constante pour toutes les instances d'évaluation X_i : $\text{var}(\varepsilon_i) = \sigma^2$
4. Les erreurs sont indépendantes : $\text{cov}(\varepsilon_i, \varepsilon_j) = 0$ pour $i \neq j$
5. L'erreur est distribuée normalement.

Les trois premières hypothèses peuvent être validées par la courbe d'analyse des résidus [Bonte, 2007] [Tunali and Batmaz, 2000], qui trace l'erreur standard en fonctions des réponses mesurées. Si les résidus varient aléatoirement autour de zéro, les trois premières hypothèses sont validées [Tunali and Batmaz, 2000]. La quatrième hypothèse est validée par la courbe de la corrélation des erreurs [Tunali and Batmaz, 2000], et la cinquième, la distribution normale de l'erreur, par la courbe normale des résidus.

I.6.2.2.2 Validation de la précision de la MSR

Une fois le métamodèle, MSR, construit et ses hypothèses validées, il est très important d'étudier sa précision, par exemple en utilisant la méthode de la validation croisée (cross validation). Pour cela, chaque solution i calculée exactement est successivement retirée du métamodèle, et un nouveau métamodèle est reconstruit en son absence. La différence entre la solution exacte f_i et celle \tilde{f}_{-i} obtenue par le métamodèle sans son concours (Eq (I.16)) donne une idée de l'erreur de prédiction du métamodèle [Batmaz and Tunali, 2003] [Myers et al, 2009] :

$$e_{-i} = f_i - \tilde{f}_{-i} \quad (\text{I.16})$$

Si le métamodèle est performant, l'erreur de prédiction (Eq (16)) est nulle pour tous les points considérés. Les résultats de la validation croisée peuvent être visualisés par une courbe représentant la solution approximée \tilde{f}_{-i} en fonction de la solution exacte f_i , pour tous les points utilisés (voir Figure 9). Plus le nuage de points s'approche de la droite $\tilde{f}_{-i} = f_i$, plus le métamodèle est performant.

Pour étudier la précision du métamodèle, MSR, on peut aussi utiliser la technique d'analyse de la variance (ANOVA, ANalysis Of VAriance), qui est un outil statistique très performant. Dans la suite de cette partie, on le présente rapidement et pour plus de détails le lecteur peut consulter [Batmaz and Tunali, 2003].

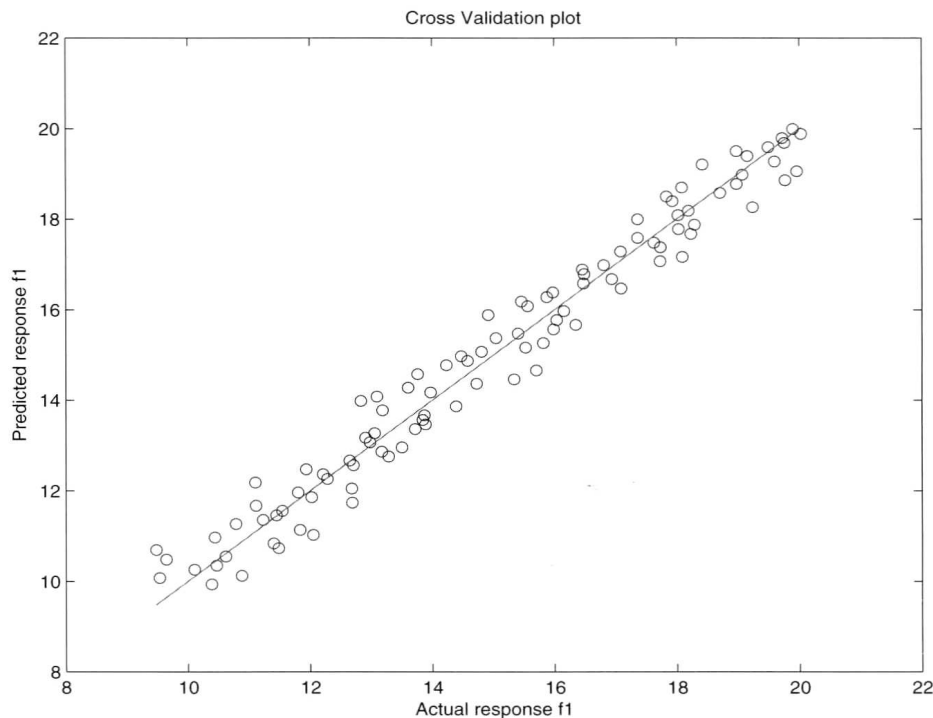


Figure 9 : Courbe de la validation croisée [Bonte, 2007].

L'étude de l'erreur repose sur trois nombres :

La somme des carrés des réponses mesurées ou la *Somme des Carrés Totaux* : SCT

$$SCT = \sum_{i=1}^m (f_i - \bar{f})^2 \quad (\text{I.17})$$

Où $\bar{f} = \frac{1}{m} \sum_{i=1}^m f_i$.

La somme des carrés des réponses prédites ou la *Somme des Carrés résultants de la Régression* : SCR

$$SCR = \sum_{i=1}^m (\tilde{f}_i - \bar{f})^2 \quad (\text{I.18})$$

La *Somme des Carrés des résidus* ou de l'Erreur : SCE

$$SCE = \sum_{i=1}^m (f_i - \tilde{f}_i)^2 \quad (\text{I.19})$$

Selon l'analyse de la variance :

$$SCT = SCR + SCE \quad (\text{I.20})$$

Si le métamodèle est performant, la variabilité due à la régression est grande par rapport à celle due à l'erreur. Pour évaluer la précision du métamodèle, on établit le *coefficient de corrélation multiple* ou *coefficient de détermination* R^2 (Eq (I.21)), qui donne une indication sur la qualité de l'approximation. En effet, il représente le rapport entre la variance due à la régression (SCR) et la variance totale (SCT). Il varie entre 0 et 1. S'il est proche de 1, c'est que le métamodèle est très précis.

$$R^2 = \frac{SCR}{SCT} = 1 - \frac{SCE}{SCT} \quad (\text{I.21})$$

On peut utiliser aussi le critère R^2 ajusté, R_A^2 , pour évaluer la précision du métamodèle :

$$R_A^2 = 1 - \frac{m-1}{m-q} (1 - R^2) \quad (\text{I.22})$$

Où $q (= n + 1)$ est le nombre de coefficients de régression. Plus R_A^2 s'approche de 1, plus le métamodèle est précis.

On peut aussi calculer la *racine carrée de la moyenne de l'erreur quadratique calculée par la technique de la validation croisée* $RMSE_{CV}$ (Cross validation Root Mean Squared Error):

$$RMSE_{CV} = \sqrt{\sum_{i=1}^m \frac{(f_i - \tilde{f}_{-i})^2}{m}} \quad (\text{I.23})$$

Une valeur faible du $RMSE_{CV}$ signifie que le métamodèle est précis.

La Méthodologie des Surfaces de Réponse est une méthode d'approximation simple, rapide et bien établie. Elle permet le lissage des bruits numériques. Toutefois, il est difficile de choisir l'ordre polynomial à utiliser, et cette méthode perd de son efficacité pour un nombre élevé de variables ($n > 10$) [Bonte, 2007].

1.6.3 Le Krigeage

Le krigeage est une technique d'interpolation spatiale d'une fonction f à partir d'un ensemble de m mesures $(X_i, f(X_i))_{i=1, \dots, m}$. Le mot krigeage est dérivé du nom de l'ingénieur Sud-Africain Daniel Gerhardus Krige, qui a proposé, dans les années cinquante, le principe fondamental de cette méthode statistique pour trouver la dispersion de l'or à Witwatersrand [Krige, 1951]. Plus tard, le mathématicien français Georges Matheron a mis en forme la théorie de cette technique, dans les années soixante, en l'appelant « Krigeage » [Matheron, 1962].

1.6.3.1 Construction du krigeage

En se référant à D. R. Jones [Jones, 2001] et Jones et al. [Jones et al, 1998], le modèle du Krigeage peut être construit en suivant la démarche ci-dessous :

On suppose que la fonction mesurée $f(X_i)$ au point X_i s'écrit :

$$f(X_i) = \sum_h \beta_h \varphi_h(X_i) + \varepsilon_i \quad (\text{I.24})$$

φ_h : Une fonction de régression,

β_h : Le coefficient inconnu à estimer de cette fonction,

ε_i : Une erreur distribuée normalement avec une moyenne nulle et une variance égal à σ^2 .

Si f est continue, alors ε est aussi continue. Si en plus, on a deux points X_i et X_j qui sont proches, alors les erreurs $\varepsilon(X_i)$ et $\varepsilon(X_j)$ devraient être proches. En bref, il n'est pas logique de supposer que les erreurs $\varepsilon(X_i)$ et $\varepsilon(X_j)$ soient indépendantes, il est plus raisonnable de supposer qu'elles sont reliées ou corrélées. Cette corrélation est d'autant plus importante que les deux points X_i et X_j sont proches et d'autant plus faible qu'ils sont éloignés. Ainsi, dans le krigeage, nous ne supposons pas que les erreurs sont indépendantes, mais plutôt que la corrélation, entre les erreurs, est liée à la distance entre les points [Jones et al, 1998] :

$$d(X_i, X_j) = \sum_{h=1}^n \theta_h |x_i^h - x_j^h|^{p_h} \quad (\theta_h \geq 0, p_h \in [1, 2]) \quad (\text{I.25})$$

En utilisant ce modèle (Eq (I.25)), la corrélation entre les erreurs aux points X_i et X_j s'écrit :

$$\text{Corr}[\varepsilon(X_i), \varepsilon(X_j)] = \exp(-d(X_i, X_j)) \quad (\text{I.26})$$

L'utilisation de cette corrélation nous permet de remplacer les termes de régression dans l'équation (Eq (I.24)) par un terme constant, μ , qui est la moyenne du métamodèle, ce qui permet d'écrire le krigeage sous la forme suivante :

$$f(X_i) = \mu + \varepsilon(X_i) \quad (\text{I.27})$$

Ce modèle est aussi appelé *modèle d'un processus stochastique*. Il est devenu courant d'appeler ce modèle de processus stochastique (Eq (I.24))-(Eq (I.26)) «DACE», une abréviation de « Design and Analysis of Computer Experiments », le titre de l'article qui a popularisé ce modèle dans son application aux expériences numériques [Sacks et al, 1989].

Le krigeage a $2n+2$ paramètres. On estime la valeur de ces paramètres en maximisant la fonction de vraisemblance (*Likelihood*). Soit $\mathbf{Y} = (f_1, \dots, f_m)^T$ le vecteur des m mesures, \mathbf{R} la matrice de taille $(m \times m)$, dont chaque composante (i, j) est donnée par l'équation (Eq (I.26)), et $\mathbf{1}$ le vecteur unitaire de taille m . La fonction de vraisemblance s'écrit :

$$\frac{1}{(2\pi)^{\frac{m}{2}} (\sigma^2)^{\frac{m}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp \left[\frac{-(\mathbf{Y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{1}\mu)}{2\sigma^2} \right] \quad (\text{I.28})$$

Il est plus pratique de maximiser le log de la fonction de vraisemblance, *log-Likelihood*, qui ignore les termes constants :

$$-\frac{m}{2}\log(\sigma^2) - \frac{1}{2}\log(|\mathbf{R}|) - \frac{(\mathbf{Y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{1}\mu)}{2\sigma^2} \quad (\text{I.29})$$

En écrivant que les dérivés de cette fonction par rapport à μ et σ^2 sont nulles et en résolvant les équations correspondantes, il est possible de déterminer les valeurs optimales de μ et σ^2 en fonction de \mathbf{R} :

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{Y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \quad (\text{I.30})$$

$$\hat{\sigma}^2 = \frac{(\mathbf{Y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{1}\hat{\mu})}{m} \quad (\text{I.31})$$

Si on injecte ces deux équations (Eq (I.30)) et (Eq (I.31)) dans l'équation (Eq (I.29)), on obtient la fonction réduite de log de la fonction de vraisemblance « log-likelihood réduite ». Sans les termes constants, cette fonction s'écrit :

$$-\frac{m}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log(|\mathbf{R}|) \quad (\text{I.32})$$

Cette fonction dépend seulement de \mathbf{R} , et par conséquent des paramètres de corrélations θ et p . En pratique, l'estimation de ces paramètres, $\hat{\theta}_h$ et \hat{p}_h ($h=1, \dots, n$), est obtenue en maximisant la fonction « log-likelihood réduite ». Une fois ces paramètres estimés, on peut utiliser les deux équations (Eq (I.30)) et (Eq (I.31)) pour calculer les deux valeurs de $\hat{\mu}$ et $\hat{\sigma}^2$.

Pour déterminer la valeur approchée \tilde{f} au point X^* , on ajoute ce point (X^*, \tilde{f}) à l'ensemble des points observés (mesurés) comme une $(m+1)^{\text{ième}}$ mesure. On calcule alors la fonction de vraisemblance augmentée en utilisant les paramètres obtenus pour la maximisation de la fonction de vraisemblance. Soit $\tilde{\mathbf{Y}} = (\mathbf{Y}^T \tilde{f})$, le vecteur des fonctions mesurées, augmenté par (X^*, \tilde{f}) . Soit \mathbf{r} le vecteur de corrélation de $\varepsilon(X^*)$ avec $\varepsilon(X_i)$:

$$\mathbf{r} = \begin{pmatrix} \text{Corr}[\varepsilon(X^*), \varepsilon(X_1)] \\ \vdots \\ \text{Corr}[\varepsilon(X^*), \varepsilon(X_m)] \end{pmatrix} \quad (\text{I.33})$$

La matrice de corrélation de système augmenté est :

$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix} \quad (\text{I.34})$$

Finalement, la fonction log-Likelihood pour le système augmenté s'écrit :

$$\left[\frac{-1}{2\hat{\sigma}^2(1-\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})} \right] (\tilde{f} - \hat{\mu})^2 + \left[\frac{\mathbf{r}^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1-\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})} \right] (\tilde{f} - \hat{\mu}) + \text{termes sans } \tilde{f} \quad (\text{I.35})$$

Cette fonction est une fonction quadratique en \tilde{f} . La valeur de \tilde{f} qui maximise (Eq (I.35)) est obtenue en écrivant que sa dérivée par rapport à \tilde{f} est égal à zéro :

$$\left[\frac{-1}{\hat{\sigma}^2(1-\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})} \right] (\tilde{f} - \hat{\mu}) + \left[\frac{\mathbf{r}^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1-\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})} \right] = 0 \quad (\text{I.36})$$

Enfin, le modèle de krigeage s'écrit :

$$\tilde{f}(X^*) = \hat{\mu} + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{1}\hat{\mu}) \quad (\text{I.37})$$

Un avantage supplémentaire du krigeage comme technique d'interpolation est la possibilité d'estimer l'erreur d'interpolation :

$$\hat{s}(X^*) = \sqrt{\hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right]} \quad (\text{I.38})$$

I.6.3.2 Validation du Krigeage

Comme la MSR, la validation du métamodèle, Krigeage, repose sur la validation des hypothèses proposées pour sa construction et sur l'évaluation de sa précision.

I.6.3.2.1 Validation des hypothèses

Les hypothèses de la construction du modèle de krigeage sont :

1. L'erreur est distribuée normalement.
2. La moyenne de cette erreur est nulle.
3. La variance de l'erreur est constante et égal à σ^2 .

Ces hypothèses peuvent être évaluées par la méthode de la validation croisée (Eq (I.16) et Figure 9), qui permet de créer les courbes d'analyse des résidus, de la corrélation des erreurs et de la distribution normale de l'erreur [Bonte, 2007].

I.6.3.2.2 Validation de la précision du krigeage

Pour valider la précision de krigeage, on utilise le coefficient de corrélation multiple R^2 (Eq (I.21)), R_A^2 (Eq (I.22)) et RMSE_{CV} (Eq (I.23)). Comme pour la MSR, si R^2 et R_A^2 sont proches de 1 et si la RMSE_{CV} tends vers 0, alors le métamodèle, krigeage, est précis. On peut aussi utiliser la courbe de la validation croisée comme critère d'évaluation de la précision du métamodèle.

Le krigeage est une méthode d'approximation flexible et déterministe. En revanche, c'est un modèle complexe qui est sensible au bruit numérique. Sa construction est coûteuse en terme de temps de calcul et si les points maîtres (points mesurés) sont proches, la matrice de corrélation peut devenir singulière [Ahmed and Qin, 2009]. Enfin, cette méthode perd de son efficacité pour des problèmes ayant un nombre élevé de variables ($n > 10$) [Bonte, 2007].

I.6.4 Les Fonctions à Bases Radiales

La méthode des Fonctions à Bases Radiales (FBR) utilise une approche d'interpolation classique basée sur la distance entre les points maîtres [Abderrahmane and Regis, 2009]. Cette approximation s'écrit sous la forme :

$$\tilde{f}(X) = \sum_{i=1}^m \omega_i \varphi(\|X - X_i\|) \quad (\text{I.39})$$

Avec $\|\cdot\|$ la norme Euclidienne et φ est une fonction fixé. On trouve dans Carr et al. [Carr et al, 1997] une présentation des quatre types de fonctions radiales φ les plus utilisées.

Les fonctions poids ω_i sont calculées de façon à ce que l'approximation de la fonction au point maître X_i , $\tilde{f}(X_i)$, soit égal à la valeur mesurée $f(X_i)$:

$$\tilde{f}(X_i) = f(X_i) \quad \forall i = 1, \dots, m \quad \text{I.40}$$

Cette équation (Eq (I.40)) implique que les ω_i sont obtenus en résolvant le système linéaire :

$$\mathbf{A}\boldsymbol{\omega} = \mathbf{Y} \quad \text{I.41}$$

Où \mathbf{A} est une matrice de taille $(m \times m)$ avec $A_{ij} = \varphi(\|X_i - X_j\|)$, $i = 1, \dots, m$ et $j = 1, \dots, m$, et $\mathbf{Y} = (f_1 \ \dots \ f_m)^T$ est le vecteur des m mesures.

La méthode des Fonctions à Bases Radiales est une méthode simple qui n'est pas sensible au bruit numérique. Son point faible est la difficulté à estimer sa précision [Bonte, 2007], comme cela est possible avec le modèle de krigeage, et à choisir le type de la fonction radiale à utiliser.

I.7 CONCLUSION

Dans ce chapitre, nous avons présenté les concepts de l'optimisation multi-objectifs, puis les algorithmes évolutionnaires (AEs) comme étant les meilleurs algorithmes de résolution de problèmes d'optimisation multi-objectifs. Ensuite, on a décrit l'algorithme SE-META, qui est utilisé dans cette thèse comme algorithme de référence pour les problèmes d'optimisation mono-objectif, avant de décrire plus en détails l'un des AEs le plus utilisé, l'algorithme génétique NSGA-II qui nous avons utilisé pour le développement d'un nouvel algorithme d'optimisation multi-objectifs. Ce nouvel algorithme est un couplage entre NSGA-II et un métamodèle développé au cours de cette thèse. Dans un second temps, nous avons présenté les trois métamodèles les plus utilisés pour la minimisation des coûts de calcul liés à une utilisation directe des AEs : la Méthodologie des Surfaces de Réponse, le Krigeage et brièvement, les Fonctions à Base Radiale.

Dans ce chapitre, on a vu que l'utilisation de chacun de ces métamodèles se heurte à des difficultés. Pour la Méthodologie des Surfaces de Réponse, le problème concerne l'ordre polynomial à utiliser, le nombre de variables du problème et l'aspect stochastique du métamodèle. L'inconvénient du krigeage est lié à la complexité du modèle (méthode coûteuse en terme de temps de calcul), à la répartition des points maîtres (si les points maîtres sont proches, la matrice de corrélation peut devenir singulière) et à sa sensibilité au bruit numérique. Enfin, la précision de la méthode des Fonctions à Bases Radiales est inconnue. Elle est liée au choix de la fonction radiale.

Pour surmonter ces problèmes et dans le but d'avoir un métamodèle précis, dans le chapitre qui suit (Chapitre II), nous allons décrire le métamodèle basé sur la Méthode de Différences Finies sans Maillage que nous avons utilisé et développé au cours de cette thèse.

CHAPITRE II

METAMODELE A BASE DE DIFFERENCES FINIES SANS MAILLAGE

II.1 INTRODUCTION

Nous avons vu au [Chapitre I](#) que la meilleure méthode (ou la plus répandue) pour la résolution d'un problème d'optimisation multi-objectifs réel (industriel) est l'utilisation d'un couplage entre un métamodèle et un algorithme évolutionnaire multi-objectifs. Les métamodèles les plus connus et les plus utilisés sont la Méthodologie des Surfaces de Réponse (MSR), le Krigeage, les Moindres Carrés Mobiles (MLS, Moving Least Square) et les Fonctions à Bases Radiales (FBR). L'utilisation de chacun de ces métamodèles se heurte à des difficultés. Pour la MSR, le problème concerne l'ordre polynomial à utiliser. Pour le métamodèle, MLS [[Breitkopf et al, 2005](#)] [[Ingarao et al, 2009](#)], l'utilisation d'un rayon d'influence dans la définition des fonctions « fenêtres », de manière à éliminer l'influence des points éloignés, rend la détermination de ce rayon délicate dans les cas les plus généraux. L'inconvénient du krigeage est lié à la complexité du modèle (méthode coûteuse en terme de temps de calcul) et à la répartition des points maîtres (si les points maîtres sont proches, la matrice de corrélation peut devenir singulière). Enfin, la précision de la méthode des FBR est inconnue et elle est liée au choix de la fonction radiale.

Pour surmonter ces problèmes et dans le but d'avoir un métamodèle précis, on a développé une approche basée sur la Méthode de Différences Finies Sans Maillage (MDFSM). Cette méthode a été initialement proposée par Lyszka et Orkisz [[Lyszka and Orkisz, 1980](#)] pour l'utilisation des grilles irrégulières et la résolution des problèmes mécaniques. Elle permet aussi de lisser des données physiques et d'estimer l'écart avec des valeurs numériques calculées, comme proposé par Krok [[Krok, 2002](#)]. Elle peut être vue comme une variante de la méthode MLS avec un choix particulier des fonctions « fenêtres ». Son avantage majeur est la résolution des difficultés introduites par les autres méthodes et de donner lieu à des temps de calcul réduits.

Dans la suite de ce chapitre, on présente une étude détaillée de la construction du métamodèle, qui constitue l'une des contributions principales de ce travail de thèse. Dans la troisième partie, le paramétrage du métamodèle est mis en place. Sa validation compose la quatrième partie. La cinquième partie analyse l'erreur liée au métamodèle, et dans la sixième partie, on développe une méthode pour la prise en compte de bruit numérique.

II.2 CONSTRUCTION DU METAMODELE

Le métamodèle basé sur la MDFSM développée est inspiré de la méthode d'interpolation de Lyszka et Orkisz pour la génération d'un maillage irrégulier [[Lyszka and Orkisz, 1980](#)]. Cette méthode a été utilisée par Do [[Do, 2006](#)] pour construire un métamodèle linéaire appliqué à la résolution des problèmes d'optimisation mono-objectif. Elle est basée sur un développement global en séries de Taylor à l'ordre un [[Do, 2006](#)] ou deux [[Lyszka and Orkisz, 1980](#)], et utilise toutes les informations disponibles pour construire une

approximation aussi précise et fiable que possible, et ce au fur et à mesure de l'enrichissement du métamodèle (ajout d'informations au cours des itérations). Elle consiste à calculer l'approximation $\tilde{f}(X_i)$ de la fonction $f(X_i)$ d'un individu i situé en X_i , à partir de la connaissance des valeurs $f(X_j)$ de la fonction f aux λ points maîtres j du domaine paramétrique. Cette approximation $\tilde{f}(X_i)$ minimise l'erreur moyenne quadratique E des approximations effectuées en tous les points maîtres j (voir plus bas pour plus de détails).

Au début, on a travaillé avec un modèle linéaire basé sur un développement en séries de Taylor à l'ordre un, M-MDFSM-1, mais on a constaté que cette approche manque de précision à cause des interpolations linéaires. Pour l'améliorer, on est passé au second ordre, M-MDFSM-2, ce qui s'est accompagnée de problèmes de singularités. Pour les surmonter, et dans le but d'avoir un modèle performant et fiable, on a construit un modèle régularisé, M-MDFSM, qui combine les avantages des deux modèles précédents.

II.2.1 Métamodèle linéaire : M-MDFSM-1

Dans cette section, on commence par la construction du métamodèle linéaire, M-MDFSM-1 [Ejday and Fourment, 2009a] [Ejday and Fourment, 2009d], qui sera appliqué sur des fonctions analytiques afin de tester sa performance et sa robustesse.

II.2.1.1 Formulation du métamodèle linéaire

Au point i de coordonnées X_i , le modèle linéaire, M-MDFSM-1, est obtenu en écrivant un développement en séries de Taylor à l'ordre un de la fonction f en tout point maître j de coordonnées X_j où la valeur $f(X_j)$ est connue :

$$\forall j, f_j = f_i + (X_j - X_i)^T \nabla f_i + O(\|X_j - X_i\|^2) \quad (\text{II.1})$$

En notant $f_j = f(X_j)$, $\forall j$, $f_i = f(X_i)$ et $\nabla f_i = \frac{df}{dX}(X_i)$ la valeur inconnue de la fonction et son gradient au nœud i .

Si on note : $a = f_i$, $\mathbf{b} = (b_k)_{k=1,n} = \nabla f_i$, $X_{ji} = (x_{ji}^k)_{k=1,n} = X_j - X_i$, $\omega_j = \|X_{ji}\|^{-4}$

Où n est le nombre de paramètres et ω_j est la fonction poids associée au nœud j . L'équation (Eq (II.1)) peut se mettre alors sous la forme d'une interpolation linéaire globale :

$$\forall j, f_j = a + \sum_{k=1}^n b_k x_{ji}^k + O(\|X_{ji}\|^2) \quad (\text{II.2})$$

La solution (a, \mathbf{b}) est obtenue en minimisant l'erreur moyenne quadratique $E(a, \mathbf{b})$, pour tous les λ points maîtres:

$$E(a, \mathbf{b}) = \frac{1}{2} \sum_{j=1}^{\lambda} \omega_j \left(a + \sum_{k=1}^n b_k x_{ji}^k - f_j \right)^2 \quad (\text{II.3})$$

Minimiser $E(a, \mathbf{b})$ revient à résoudre :

$$\begin{cases} \frac{\partial E}{\partial a} = 0 \Rightarrow a \sum_{j=1}^{\lambda} \omega_j + \sum_{k=1}^n \left(b_k \sum_{j=1}^{\lambda} \omega_j x_{ji}^k \right) = \sum_{j=1}^{\lambda} \omega_j f_j \\ \forall l = 1, n, \quad \frac{\partial E}{\partial b_l} = 0 \Rightarrow a \sum_{j=1}^{\lambda} \omega_j x_{ji}^l + \sum_{k=1}^n \left(b_k \sum_{j=1}^{\lambda} \omega_j x_{ji}^k x_{ji}^l \right) = \sum_{j=1}^{\lambda} \omega_j x_{ji}^l f_j \end{cases} \quad (\text{II.4})$$

L'équation (Eq (II.4)) est un système linéaire de $(n+1)$ équations à $(n+1)$ inconnues :

$$\mathbf{AX} = \mathbf{B} \quad (\text{II.5})$$

Où \mathbf{A} est une matrice symétrique de taille $(n+1) \times (n+1)$,

$$\mathbf{A} = \begin{bmatrix} \sum_{j=1}^{\lambda} \omega_j & \sum_{j=1}^{\lambda} \omega_j x_{ji}^1 & \sum_{j=1}^{\lambda} \omega_j x_{ji}^2 & \cdots & \sum_{j=1}^{\lambda} \omega_j x_{ji}^n \\ \sum_{j=1}^{\lambda} \omega_j x_{ji}^1 & \sum_{j=1}^{\lambda} \omega_j (x_{ji}^1)^2 & \sum_{j=1}^{\lambda} \omega_j x_{ji}^1 x_{ji}^2 & \cdots & \sum_{j=1}^{\lambda} \omega_j x_{ji}^1 x_{ji}^n \\ \sum_{j=1}^{\lambda} \omega_j x_{ji}^2 & \sum_{j=1}^{\lambda} \omega_j x_{ji}^2 x_{ji}^1 & \sum_{j=1}^{\lambda} \omega_j (x_{ji}^2)^2 & \cdots & \sum_{j=1}^{\lambda} \omega_j x_{ji}^2 x_{ji}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{j=1}^{\lambda} \omega_j x_{ji}^n & \sum_{j=1}^{\lambda} \omega_j x_{ji}^n x_{ji}^1 & \sum_{j=1}^{\lambda} \omega_j x_{ji}^n x_{ji}^2 & \cdots & \sum_{j=1}^{\lambda} \omega_j (x_{ji}^n)^2 \end{bmatrix}$$

\mathbf{X} , le vecteur des inconnues de taille $(n+1)$: $\mathbf{X}^T = [a \quad b_1 \quad b_2 \quad \cdots \quad b_n]$

\mathbf{B} , le vecteur second membre de taille $(n+1)$:

$$\mathbf{B}^T = \left[\sum_{j=1}^{\lambda} \omega_j f_j \quad \sum_{j=1}^{\lambda} \omega_j x_{ji}^1 f_j \quad \sum_{j=1}^{\lambda} \omega_j x_{ji}^2 f_j \quad \cdots \quad \sum_{j=1}^{\lambda} \omega_j x_{ji}^n f_j \right]$$

Soit le vecteur \mathbf{V}_j de dimension $(n+1)$ contenant les composantes de la différence entre les coordonnées du point maître j et d'un point inconnu i :

$$\mathbf{V}_j^T = [1 \quad x_{ji}^1 \quad x_{ji}^2 \quad \cdots \quad x_{ji}^n].$$

En utilisant ce vecteur, \mathbf{A} et \mathbf{B} s'écrivent :

$$\forall k, l = 1, n, \begin{cases} \mathbf{A}_{kl} = \sum_{j=1}^{\lambda} \omega_j \mathbf{V}_j^k \cdot \mathbf{V}_j^l \\ \mathbf{B}_k = \sum_{j=1}^{\lambda} \omega_j \mathbf{V}_j^k \cdot f_j \end{cases}$$

Le vecteur inconnu \mathbf{X} est alors donné par :

$$\mathbf{X} = \mathbf{A}^{-1} \mathbf{B} \quad (\text{II.6})$$

Notons que pour que \mathbf{A} soit inversible, le nombre de points maîtres doit être au moins de $(n+1)$.

II.2.1.2 Performance du métamodèle linéaire

Dans cette partie, une série des fonctions analytiques est utilisée pour tester la capacité du modèle, M-MDFSM-1, à les approcher en présence d'un nombre de points maîtres (points noirs) répartis de différentes manières (uniforme, aléatoire et particulière). Les fonctions utilisées sont:

Fonction linéaire :

$$f(x) = x, \quad x \in [-2, 2] \quad (\text{II.7})$$

Fonction quadratique 1D :

$$f(x) = x^2, \quad x \in [-2, 2] \quad (\text{II.8})$$

Fonction bi-extrema :

$$f(x) = (x - 0.5)^2 - 0.2 \exp\left(-\left(\frac{x - 0.173}{0.1}\right)^2\right) - 0.1 \exp\left(-\left(\frac{x - 0.75}{0.2}\right)^2\right), x \in [0 \ 1] \quad (\text{II.9})$$

Fonction quadratique 2D :

$$f(x, y) = x^2 + y^2, x, y \in [-2 \ 2] \quad (\text{II.10})$$

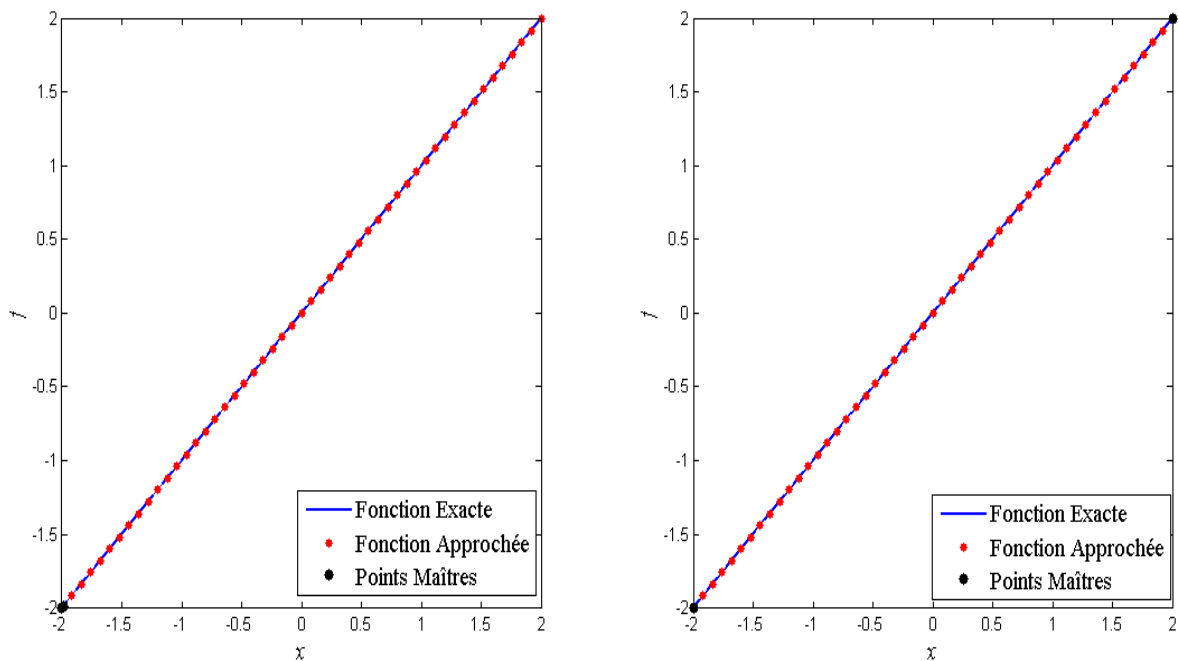


Figure 10 : Approximation de la fonction linéaire avec le M-MDFSM-1 construit avec deux points maîtres répartis d'une manière : particulière (à gauche) et uniforme (à droite).

Dans le cas d'une fonction linéaire, la Figure 10 montre qu'en présence du nombre minimal de points maîtres (2 points) et indépendamment de leurs répartitions, le métamodèle linéaire trouve la courbe analytique.

La courbe à gauche de la Figure 10 montre la performance du M-MDFSM-1 pour l'approximation des fonctions linéaires même pour des cas difficiles, comme le cas présenté où on a une mauvaise dispersion des points maîtres, qui sont localisés dans la partie gauche de l'espace paramétrique et ils sont très proches $(-2, -1.99)$.

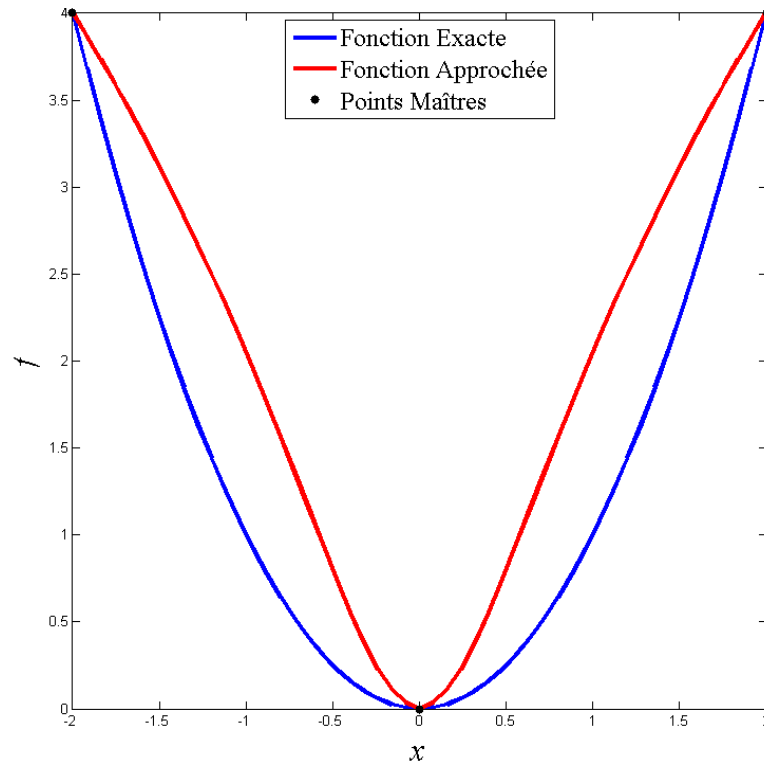


Figure 11 : Approximation de la fonction quadratique 1D avec M -MDFSM-1 construit avec trois points maîtres répartis uniformément.

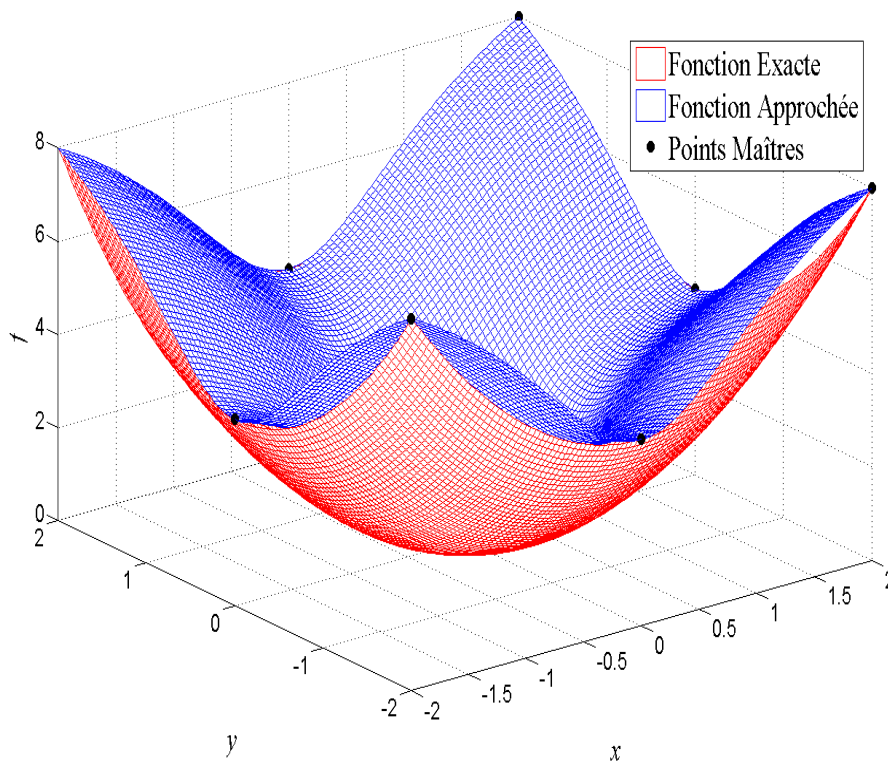


Figure 12 : Approximation de la fonction quadratique 2D avec M -MDFSM-1 construit avec neuf points maîtres générés d'une manière uniforme.

Pour une dispersion uniforme des points maîtres et en présence des informations nécessaires (nombre minimal de points maîtres respecté), la [Figure 11](#) et la [Figure 12](#) montrent que dans le cas d'une fonction quadratique, le métamodèle d'ordre un donne des approximations peu satisfaisantes.

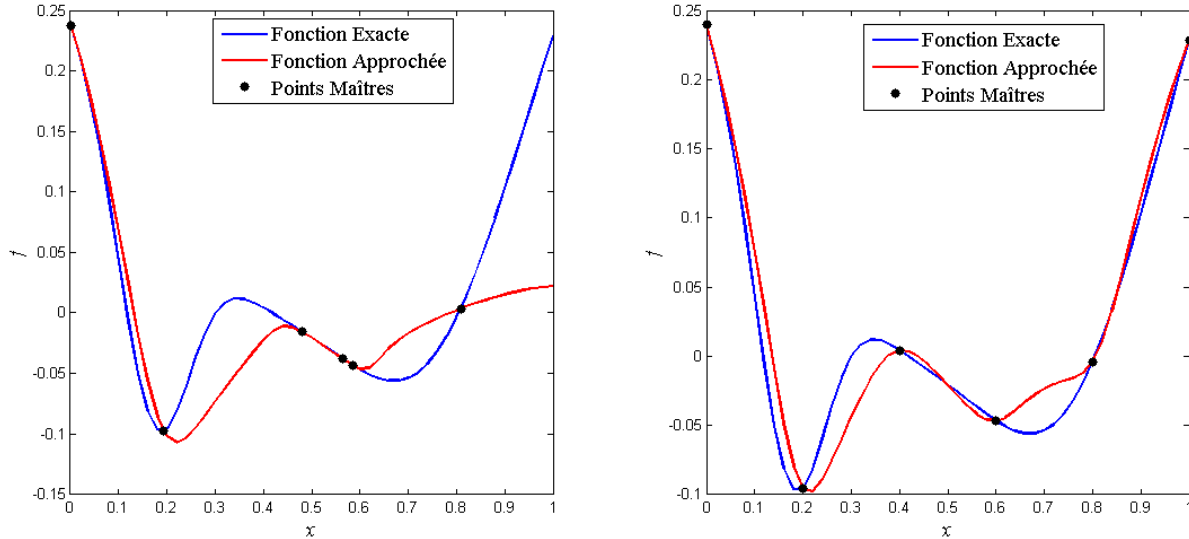


Figure 13 : Approximation de la fonction bi-extrema avec M-MDFSM-1 construit avec six points maîtres générés : Aléatoirement (à gauche) et uniformément (à droite).

Le manque de précision du métamodèle est confirmé par la fonction bi-extrema ([Figure 13](#)), où l'on constate que le métamodèle est bien moins « lisse » que la fonction initiale, ce qui est regrettable – on souhaite que le métamodèle soit le plus lisse possible, c'est-à-dire que dans l'incertitude, le métamodèle propose la solution la plus « simple », la plus « lisse », avec le moins d'extrema, ce qui n'est clairement pas le cas.

II.2.2 Métamodèle quadratique : M-MDFSM-2

Comme nous venons de le voir, le modèle linéaire, M-MDFSM-1, ne permet pas d'obtenir des interpolations satisfaisantes, ce qui peut se traduire par un manque de performance de l'algorithme d'optimisation couplé au métamodèle, et nous conduit à passer au développement en séries de Taylor à l'ordre deux (métamodèle M-MDFSM-2) [[Ejday and Fourment, 2010a](#)] [[Ejday and Fourment, 2010b](#)].

II.2.2.1 Formulation du modèle quadratique

En suivant sur le principe utilisé pour le modèle linéaire, au point i courant de coordonnées X_i , nous écrivons un développement en séries de Taylor à l'ordre deux de la fonction f en tout point maître j de coordonnées X_j :

$$\forall j, f_j = f_i + (X_j - X_i)^T \nabla f_i + \frac{1}{2} (X_j - X_i)^T \nabla^2 f_i (X_j - X_i) + \square \left(\|X_j - X_i\|^3 \right) \quad (\text{II.11})$$

Où $\nabla^2 f_i = \frac{d^2 f}{dX^2}(X_i)$. En utilisant les mêmes notations que précédemment, avec les variations et additions suivantes : $\omega_j = \|X_{ji}\|^{-6}$, $\mathbf{c} = (c_{kl})_{k,l=1,n} = \nabla^2 f_i$ l'équation (Eq (II.11)) donne une approximation globale de degré deux :

$$\forall j, f_j = a + \sum_{k=1}^n b_k x_{ji}^k + \frac{1}{2} \sum_{k,l=1}^n c_{kl} x_{ji}^k x_{ji}^l + O(\|X_{ji}\|^3) \quad (\text{II.12})$$

La solution $(a, \mathbf{b}, \mathbf{c})$ est obtenue en minimisant l'erreur moyenne quadratique $E(a, \mathbf{b}, \mathbf{c})$, pour tous les λ points maîtres :

$$E(a, \mathbf{b}, \mathbf{c}) = \frac{1}{2} \sum_{j=1}^{\lambda} \omega_j \left(a + \sum_{k=1}^n b_k x_{ji}^k + \frac{1}{2} \sum_{k,l=1}^n c_{kl} x_{ji}^k x_{ji}^l - f_j \right)^2 \quad (\text{II.13})$$

Minimiser $E(a, \mathbf{b}, \mathbf{c})$ revient à résoudre :

$$\left\{ \begin{array}{l} \frac{\partial E}{\partial a} = 0 \Rightarrow a \sum_{j=1}^{\lambda} \omega_j + \sum_{k=1}^n \left(b_k \sum_{j=1}^{\lambda} \omega_j x_{ji}^k \right) + \frac{1}{2} \sum_{k,l=1}^n \left(c_{kl} \sum_{j=1}^{\lambda} \omega_j x_{ji}^k x_{ji}^l \right) = \sum_{j=1}^{\lambda} \omega_j f_j \\ \forall \mu = 1, n, \frac{\partial E}{\partial b_{\mu}} = 0 \Rightarrow a \sum_{j=1}^{\lambda} \omega_j x_{ji}^{\mu} + \sum_{k=1}^n \left(b_k \sum_{j=1}^{\lambda} \omega_j x_{ji}^k x_{ji}^{\mu} \right) \\ \quad + \frac{1}{2} \sum_{k,l=1}^n \left(c_{kl} \sum_{j=1}^{\lambda} \omega_j x_{ji}^k x_{ji}^l x_{ji}^{\mu} \right) = \sum_{j=1}^{\lambda} \omega_j x_{ji}^{\mu} f_j \\ \forall \psi, \mu = 1, n, \frac{\partial E}{\partial c_{\psi\mu}} = 0 \Rightarrow \frac{a}{2} \sum_{j=1}^{\lambda} \omega_j x_{ji}^{\psi} x_{ji}^{\mu} + \frac{1}{2} \sum_{k=1}^n \left(b_k \sum_{j=1}^{\lambda} \omega_j x_{ji}^k x_{ji}^{\psi} x_{ji}^{\mu} \right) \\ \quad + \frac{1}{4} \sum_{k,l=1}^n \left(c_{kl} \sum_{j=1}^{\lambda} \omega_j x_{ji}^k x_{ji}^l x_{ji}^{\psi} x_{ji}^{\mu} \right) = \frac{1}{2} \sum_{j=1}^{\lambda} \omega_j x_{ji}^{\psi} x_{ji}^{\mu} f_j \end{array} \right. \quad (\text{II.14})$$

L'équation (Eq (II.14)) est un système linéaire de $n' = \frac{(n+1)(n+2)}{2}$ équations à n' inconnues :

$$\mathbf{AX} = \mathbf{B} \quad (\text{II.15})$$

Où :

\mathbf{X} est un vecteur d'inconnues de taille n' : $\mathbf{X}^T = [a \quad b_1 \quad \dots \quad b_n \quad c_{11} \quad \dots \quad c_{1n} \quad \dots \quad c_{mn}]$

\mathbf{B} est un vecteur second membre de taille n' .

\mathbf{A} est une matrice symétrique de taille $n' \times n'$.

Soit le vecteur \mathbf{V}_j de dimension n' contenant les composantes de la différence entre les coordonnées d'un point maître j et un point inconnu i :

$$\mathbf{V}_j^T = \left[1 \quad x_{ji}^1 \quad \dots \quad x_{ji}^n \quad \frac{1}{2}(x_{ji}^1)^2 \quad x_{ji}^1 x_{ji}^2 \quad \dots \quad x_{ji}^1 x_{ji}^n \quad \dots \quad \frac{1}{2}(x_{ji}^n)^2 \right]$$

En utilisant ce vecteur, \mathbf{A} et \mathbf{B} s'écrivent :

$$\forall k, l = 1, n, \begin{cases} \mathbf{A}_{kl} = \sum_{j=1}^{\lambda} \omega_j \mathbf{V}_j^k \cdot \mathbf{V}_j^l \\ \mathbf{B}_k = \sum_{j=1}^{\lambda} \omega_j \mathbf{V}_j^k \cdot f_j \end{cases}$$

Enfin, le vecteur inconnu \mathbf{X} est donné par :

$$\mathbf{X} = \mathbf{A}^{-1} \mathbf{B} \quad (\text{II.16})$$

Pour que la matrice \mathbf{A} soit inversible, n' points maîtres bien distribués dans l'espace paramétrique sont nécessaires, ce qui n'est toutefois toujours pas une condition suffisante.

II.2.2.2 Performance du métamodèle quadratique

On applique le métamodèle quadratique, M-MDFSM-2, sur les mêmes fonctions tests utilisées pour le métamodèle linéaire, M-MDFSM-1.

La **Figure 14** montre qu'avec 3 points maîtres (nombre minimal de points pour construire l'interpolation quadratique), on retrouve la courbe analytique.

Dans le cas d'une fonction quadratique, quelque soit la répartition des points maîtres (**Figure 15** et **Figure 17**), on retrouve également la fonction analytique, tant en 1D qu'en 2D.

Pour la fonction bi-extrema, l'approximation obtenue (**Figure 16**) avec les mêmes points maîtres utilisés pour la construction du métamodèle linéaire est plus « lisse »: elle présente moins de tendance à faire apparaître plus d'extrema que la fonction originale

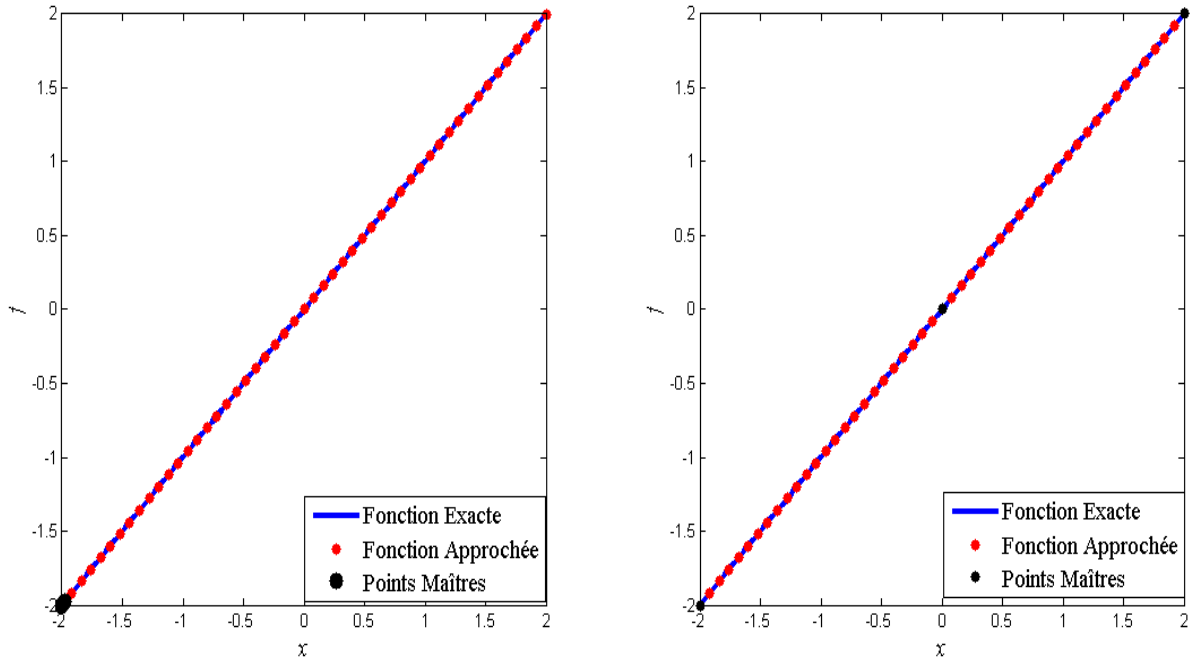


Figure 14 : Approximation de la fonction linéaire avec M -MDFSM-2 construit avec trois points maîtres répartis d'une manière : particulière (à gauche) et uniforme (à droite).

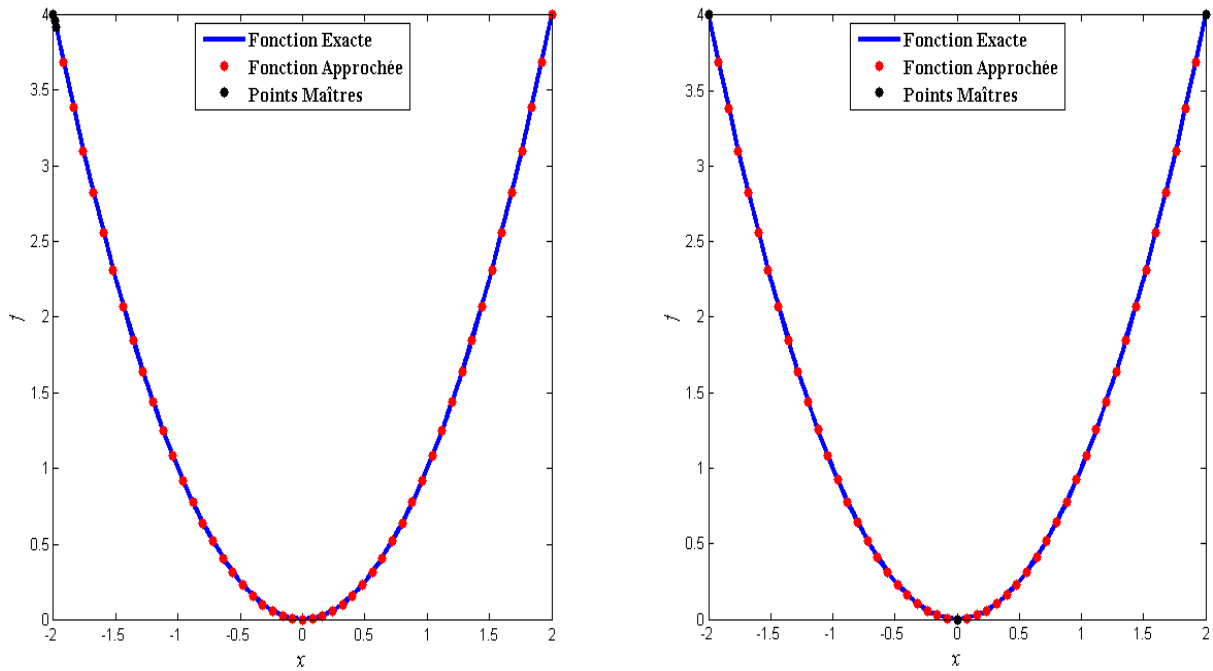


Figure 15 : Approximation de la fonction quadratique 1D avec le métamodèle, M -MDFSM-2, construit avec trois points maîtres s répartis d'une manière : particulière (à gauche) et uniforme (à droite).

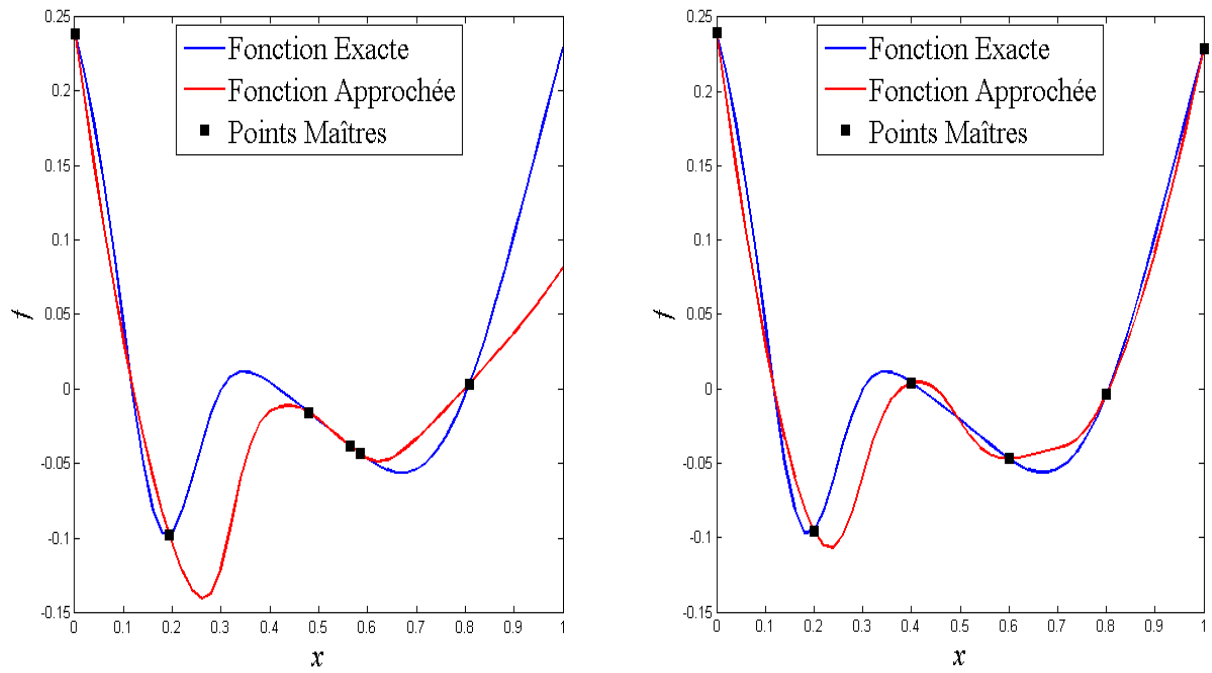


Figure 16 : Approximation de la fonction bi-extrema avec le métamodèle, *M-MDFSM-2*, construit avec six points maîtres répartis : aléatoirement (à gauche) et uniformément (à droite).

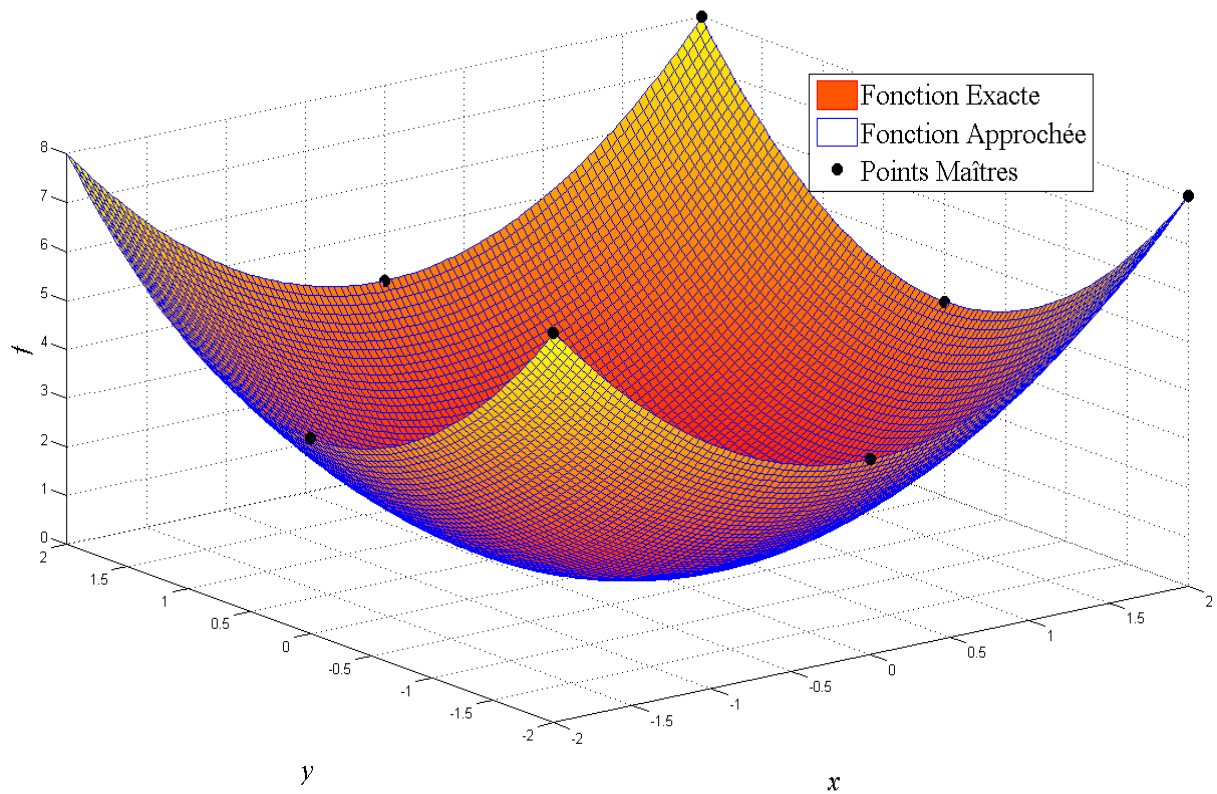


Figure 17 : Approximation de la fonction quadratique 2D avec le métamodèle, *M-MDFSM-2*, construit avec neuf points maîtres générés uniformément.

II.2.3 Approximation mixte : M-MDFSM

Le métamodèle d'ordre deux, M-MDFSM-2, est plus précis, mais pour que sa matrice \mathbf{A} soit inversible, il nécessite un plus grand nombre de points maîtres ($\geq n'$). Ce nombre de points maîtres est nécessaire mais insuffisant car ces points peuvent être mal répartis dans l'espace de recherche, de sorte que le risque de ne pouvoir construire cette interpolation est bien plus grand qu'avec l'ordre un. Par exemple, on souhaite utiliser M-MDFSM-2 pour approximer la fonction quadratique 2D (Eq (II.10)) à partir des six points maîtres particuliers donnés dans le Tableau 1, et qui pourraient correspondre à un plan d'expérience tronqué.

x	-2	-2	0	0	2	2
y	-2	2	-2	2	-2	2

Tableau 1 : Coordonnées des points maîtres mal répartis dans l'espace de recherche

Bien que l'on dispose du nombre minimal de points nécessaires, la Figure 18 et le Tableau 1 montrent qu'ils ne sont suffisamment distribués dans la direction y , avec seulement deux valeurs distinctes qui ne permettent pas d'effectuer une interpolation quadratique. La matrice \mathbf{A} s'avère non inversible. Il faudrait donc utiliser une interpolation d'ordre un, mais on remarque que dans la direction x , les points maîtres sont bien répartis et qu'il serait possible d'utiliser une interpolation de degré deux, ce qui nous conduit à associer les deux interpolations en utilisant une formulation régularisée (métamodèle M-MDFSM) [Ejday and Fourment, 2009b] [Ejday and Fourment, 2009c].

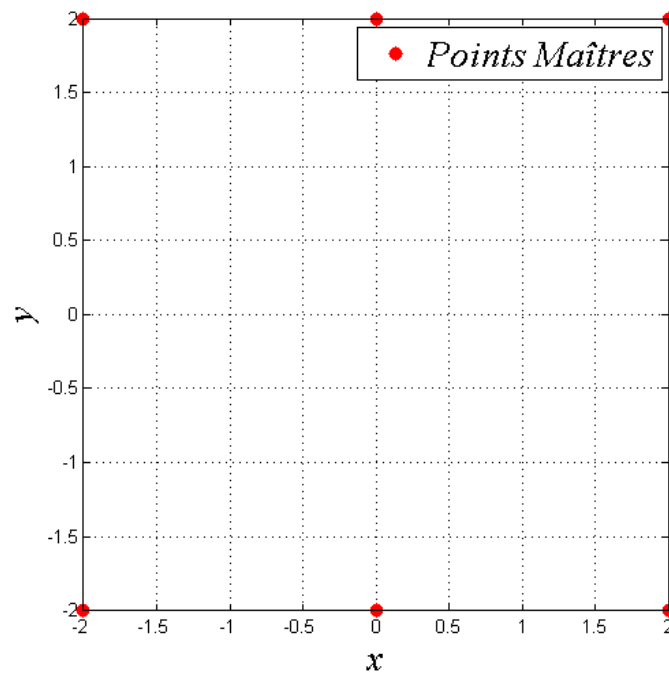


Figure 18 : Répartition des points maîtres dans l'espace de recherche

II.2.3.1 Formulation du métamodèle régularisé

Le métamodèle régularisé, M-MDFSM, est obtenu en modifiant la définition de l'erreur moyenne quadratique $E(a, \mathbf{b}, \mathbf{c})$ (Eq (II.13)) par l'ajout d'un terme de régularisation des coefficients d'interpolation du second ordre:

$$E_r(a, \mathbf{b}, \mathbf{c}) = E(a, \mathbf{b}, \mathbf{c}) + \rho \|c\|^2 = E(a, \mathbf{b}, \mathbf{c}) + \rho_0 \rho_a \sum_{k,l=1}^n (c_{kl})^2 \quad (\text{II.17})$$

Où :

ρ_0 est un coefficient numérique très faible.

ρ_a est un terme d'adimensionnement déterminé de la manière suivante :

Soit la notation suivante : # : de même dimension

$$\text{On a } E(a, \mathbf{b}, \mathbf{c}) \# \sum_{j=1}^{\lambda} \omega_j \left(\sum_{k,l=1}^n c_{kl} x_{ji}^k x_{ji}^l \right)^2 \# \sum_{j=1}^{\lambda} \omega_j \sum_{k,l=1}^n (c_{kl})^2 (x_{ji}^k x_{ji}^l)^2$$

$$\text{Et } \rho_a \sum_{k,l=1}^n (c_{kl})^2 \# E(a, \mathbf{b}, \mathbf{c}) \Rightarrow \rho_a \# \sum_{j=1}^{\lambda} \omega_j \sum_{k,l=1}^n (x_{ji}^k x_{ji}^l)^2$$

$$\text{Or } \omega_j = \left(\sum_{k=1}^n (x_{ji}^k)^2 \right)^{-3} \Rightarrow \sum_{k,l=1}^n (x_{ji}^k x_{ji}^l)^2 \# \left(\sum_{k=1}^n (x_{ji}^k)^2 \right)^2 \# (\omega_j)^{-\frac{2}{3}}$$

$$\text{D'où on obtient } \rho_a \# \sum_{j=1}^{\lambda} \omega_j (\omega_j)^{-\frac{2}{3}} \# \sum_{j=1}^{\lambda} (\omega_j)^{\frac{1}{3}}$$

La minimisation de cette erreur $E_r(a, \mathbf{b}, \mathbf{c})$ donne:

$$\left\{ \begin{array}{l} \frac{\partial E_r}{\partial a} = \frac{\partial E}{\partial a} = 0 \\ \forall \mu = 1, n, \quad \frac{\partial E_r}{\partial b_\mu} = \frac{\partial E}{\partial b_\mu} = 0 \\ \forall \psi, \mu = 1, n, \quad \frac{\partial E_r}{\partial c_{\psi\mu}} = 0 \Rightarrow \frac{\partial E}{\partial c_{\psi\mu}} + 2\rho \sum_{k,l=1}^n c_{kl} = 0 \end{array} \right. \quad (\text{II.18})$$

L'équation (II.18) est un système linéaire de n' équations à n' inconnues :

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B} \quad (\text{II.19})$$

La matrice \mathbf{A} est la somme de deux matrices, $\bar{\mathbf{A}}$, qui est identique à la matrice \mathbf{A} du modèle quadratique, et \mathbf{A}_r , qui contient les termes de régularisation :

$$\mathbf{A} = \bar{\mathbf{A}} + \mathbf{A}_r$$

\mathbf{A}_r est une matrice diagonale qui peut s'écrire sous la forme :

$$\mathbf{A}_r = \text{Diag}(\mathbf{V}_r)$$

Le vecteur \mathbf{V}_r est de même taille que \mathbf{X} . Il est obtenu à partir de \mathbf{X} en remplaçant les termes d'ordre un (différents de c_{ij}) par des zéros et les : c_{ij} ($i \neq j$) par 4ρ et c_{ii} par 2ρ

$$\mathbf{X}^T = [a \quad b_1 \quad \dots \quad b_n \quad c_{11} \quad \dots \quad c_{1n} \quad \dots \quad c_m]$$

$$\mathbf{V}_r^T = [0 \quad 0 \quad \dots \quad 0 \quad 2\rho \quad \dots \quad 4\rho \quad \dots \quad 2\rho]$$

II.2.3.2 Performance du métamodèle régularisé

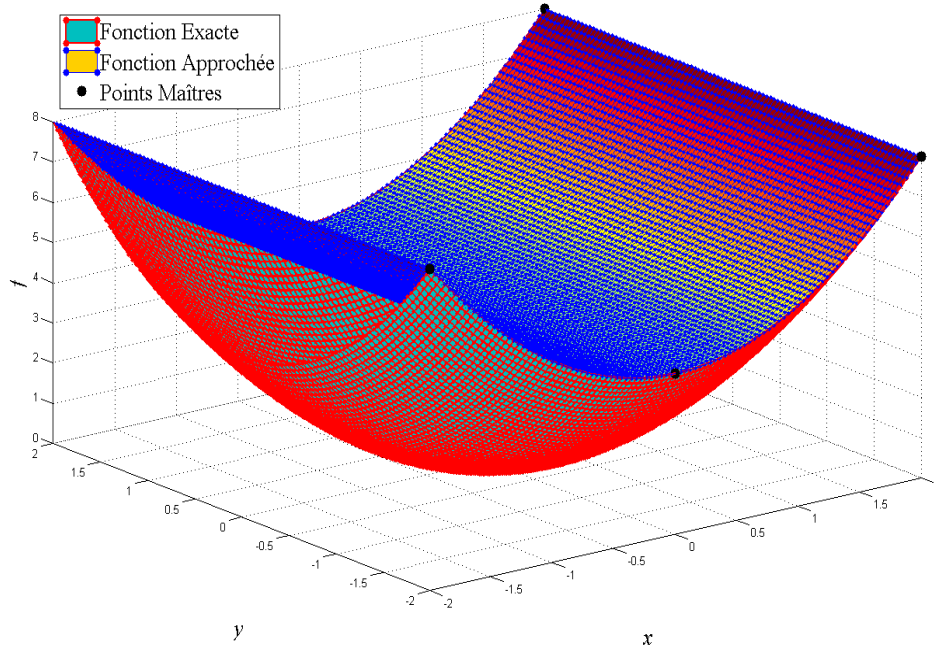


Figure 19 : Approximation de la fonction quadratique 2D avec le modèle régularisé, M-MDFSM.

Pour l'exemple précédant (voir [Tableau 1](#) et [Figure 18](#)), l'utilisation du modèle régularisé, M-MDFSM, permet d'obtenir une approximation quadratique suivant l'axe x , et linéaire suivant l'axe y , comme le montre la [Figure 19](#). Dans les directions propres de la matrice \mathbf{A} où les valeurs propres sont nulle, l'interpolation est de degré un, et de degré deux dans les autres.

II.3 PARAMETRAGE DU METAMODELE M-MDFSM

II.3.1 Critère de singularité

La formulation mixte, M-MDFSM qui sera utilisée dans le reste de cette thèse, est très séduisante, mais dans la pratique, il s'avère difficile de trouver un choix adéquat du coefficient de régularisation ρ_0 , même avec l'adimensionalisation proposée, qui donne des résultats satisfaisants quelque soit la répartition des points maîtres. On préfère donc n'y avoir recours que lorsque la matrice \mathbf{A} n'est pas inversible, et utiliser la formulation non régularisée dans les autres cas. La difficulté est alors de définir le seuil de singularité de cette matrice.

Une première idée consiste à évaluer son déterminant. Les erreurs d'arrondis font que celui-ci n'est jamais parfaitement nul, il faut alors définir une valeur seuil pour laquelle il peut être considéré comme nul. Malheureusement, cette valeur est délicate à adimensionaliser, dépendant de la norme de la matrice \mathbf{A} et de sa taille, mais surtout le déterminant n'apparaît pas comme le critère le plus fiable pour évaluer la singularité de la matrice

On préfère utiliser la décomposition de la matrice \mathbf{A} en valeurs singulières (DVS), qui est une méthode plus générale. Une matrice carrée peut être décomposée de la manière suivante [[Golub and Van Loan, 1989](#)] [[Cours, 2001](#)] :

$$\mathbf{A} = \mathbf{USV}^T$$

Où \mathbf{U} et \mathbf{V} sont deux matrices orthogonales de taille $n' \times n'$ et \mathbf{S} est une matrice diagonale de taille $n' \times n'$ contenant les valeurs singulières : $s_1 \geq s_2 \geq \dots \geq s_{n'}$

Pour tester la singularité de \mathbf{A} , on utilise le critère de singularité suivant :

$$\mathbf{A} \text{ est singulière si et seulement si : } \frac{s_{n'}}{s_1} \leq \varepsilon = 10^{-16}$$

II.3.2 Ordre de la fonction de pondération η

L'influence des différents points maîtres j est contrôlée par la fonction poids donnée par $\omega_j = \left(\|x_{ji}\|^2 \right)^{-(1+\eta)}$, où η est l'ordre d'interpolation qui vaut 1 ou 2. Lorsque le point i tend vers un point maître j , la valeur du poids tend vers l'infini. Krok [[Krok, 2002](#)] propose alors de régulariser ce poids, en utilisant la fonction suivante :

$$\omega_j = \left(\|x_{ji}\|^2 + \frac{g^4}{\|x_{ji}\|^2 + g^2} \right)^{-(1+\eta)}$$

Où g est un coefficient de lissage au voisinage des points maîtres.

La valeur de η a été déterminée à partir de l'erreur de développement en séries de Taylor, mais la méthode des MLS utilise d'autres fonctions « fenêtres », par exemple de type exponentiel. Il semble donc intéressant, dans ce contexte, d'évaluer l'influence de la valeur de η pour voir si le choix effectué est optimal.

Cette étude est faite sur deux fonctions: la fonction bi-extrema (Eq (II.9)) et la fonction inverse donnée par l'équation suivante:

$$f(x) = \frac{1}{x}, \quad x \in [0.1, 2] \quad (\text{II.20})$$

La fonction inverse est approximée avec un métamodèle construit à partir de quatre points maîtres, et la fonction bi-extrema avec six points maîtres, pour des valeurs de η variant de 0 à 3. La Figure 20 et la Figure 21 montrent que l'utilisation de $\eta=2$ ou $\eta=3$ donne des approximations un peu plus satisfaisantes qu'avec les valeurs de $\eta=0$ ou $\eta=1$ (permettant d'étendre le rayon d'influence des points maîtres). Dans la suite, on conserve donc $\eta=2$.

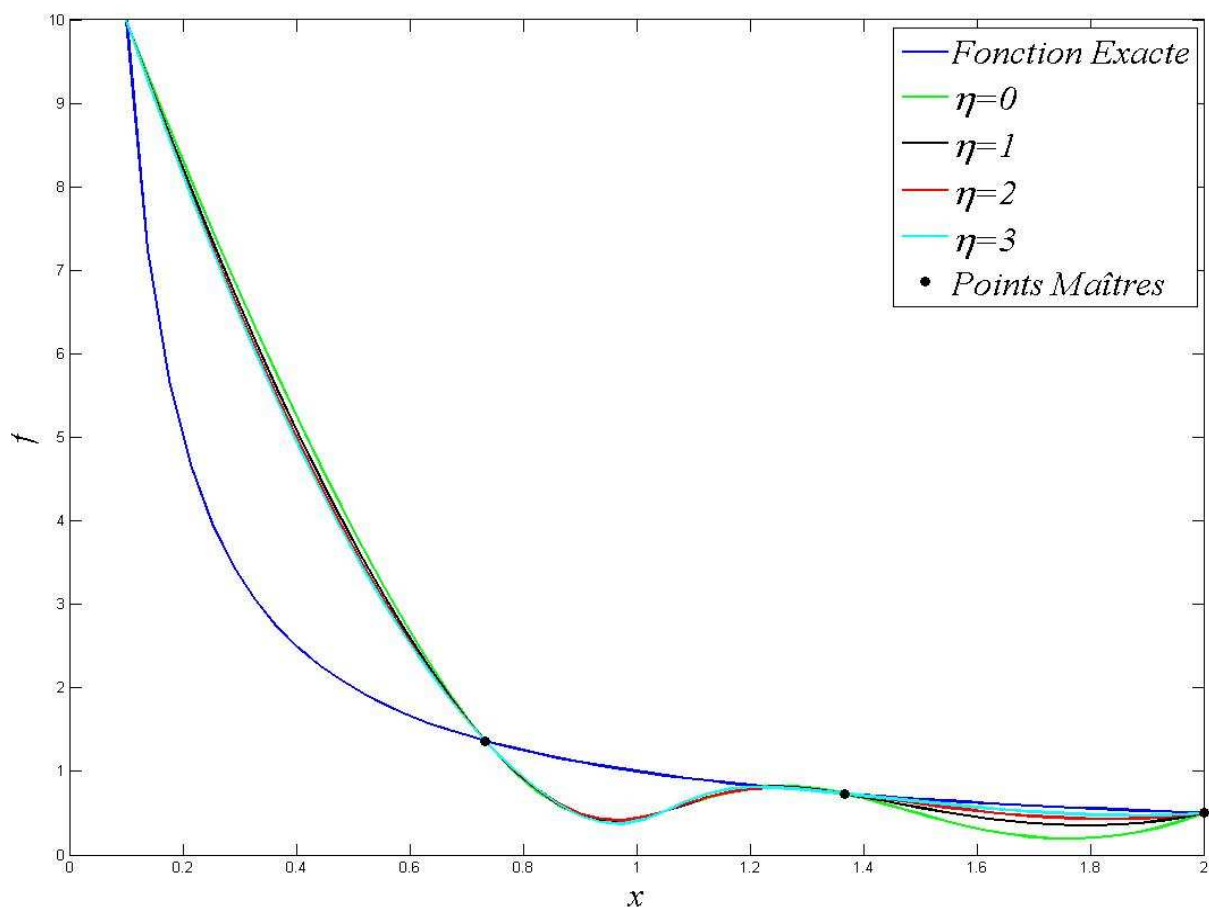


Figure 20 : Influence de la fonction de pondération sur l'approximation de la fonction inverse

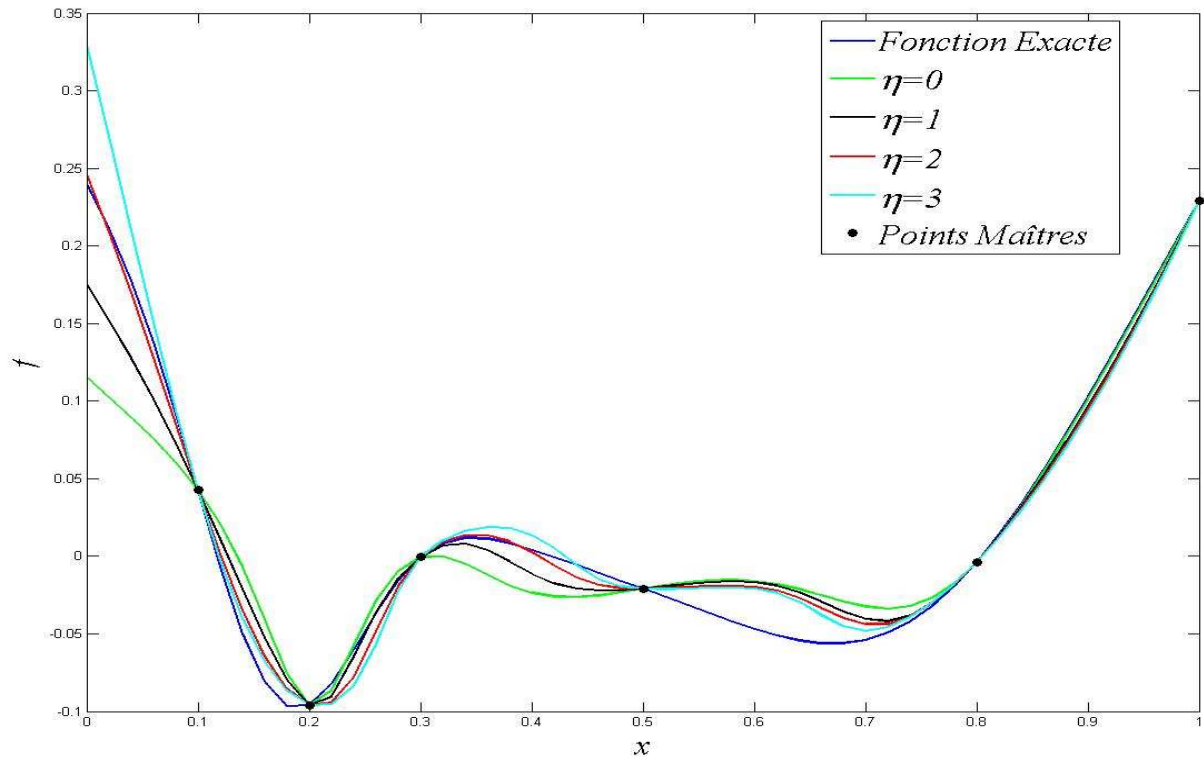


Figure 21 : Influence de la fonction de pondération sur l'approximation de la fonction bi-extrema

II.3.3 Valeur de la fonction de lissage g

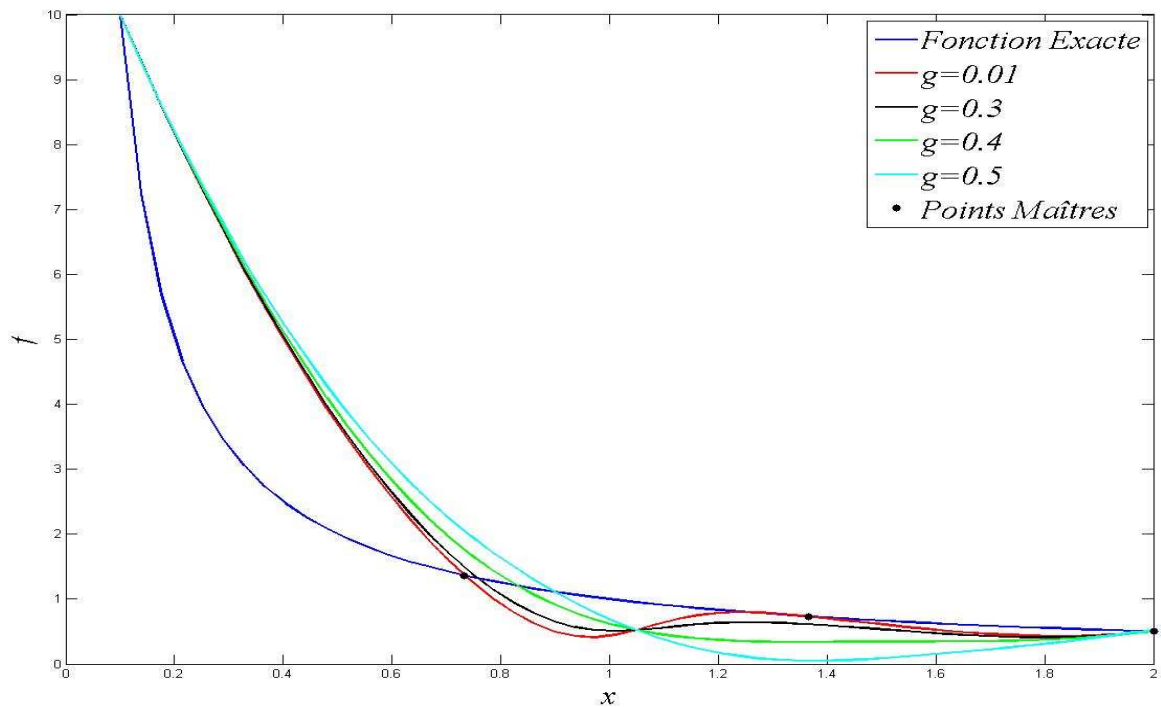


Figure 22 : Influence de la fonction de lissage sur l'approximation de la fonction inverse

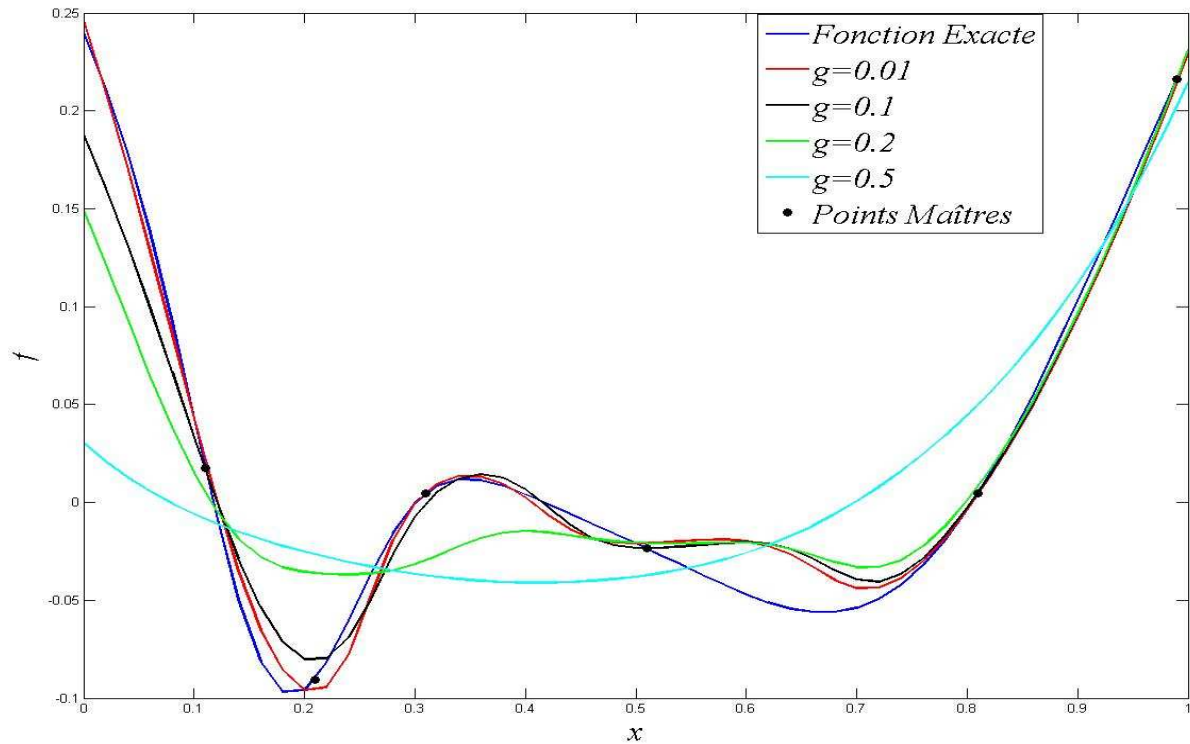


Figure 23 : Influence de la fonction de lissage sur l'approximation de la fonction bi-extrema

Sur les mêmes fonctions et avec les mêmes points maîtres, on étudie l'influence de la valeur du coefficient g en la faisant varier entre 0,01 et 0,5. La Figure 22 et la Figure 23 montrent comment la fonction g contrôle le lissage de l'approximation. Si sa valeur tend vers zéro, on obtient une courbe plus précise, passant exactement par les points maîtres, mais qui oscille davantage. Si g devient grand, l'approximation est plus lisse, mais moins précise. Dans la suite, nous privilégions la précision en prenant $g = 0.01$.

II.4 VALIDATION DU METAMODELE M-MDFSM

Après la construction et le paramétrage du métamodèle, il est intéressant de le valider. Pour ce faire, On utilise les deux méthodes de validation de métamodèles, qui sont la validation croisée et la technique d'analyse de la variance (voir I.6.2.2). Pour la validation de notre métamodèle, M-MDFSM, on utilise les deux fonctions, bi-extrema et Camel Back.

Validation du métamodèle sur la fonction bi-extrema

SCT	SCR	SCE	R^2	R^2_A	$RMSE_{CV}$
0.760328	0.751398	0.000226074	0.999703	0.9988714	0.00149611

Tableau 2 : Analyse de la variance de la fonction bi-extrema

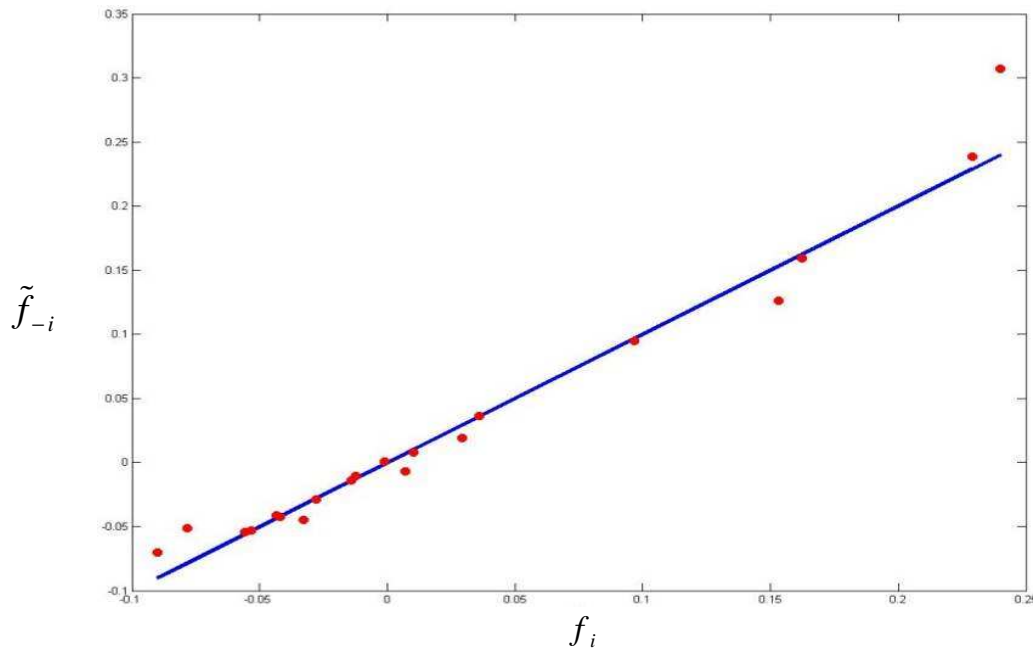


Figure 24 : Courbe de la validation croisée de la fonction bi-extrema

Validation du métamodèle sur la fonction Camel Back

La fonction Camel Back est donnée par l'équation suivante :

$$f(x, y) = \left(4 - 2,1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2, \quad x, y \in [-5, 5] \quad (\text{II.21})$$

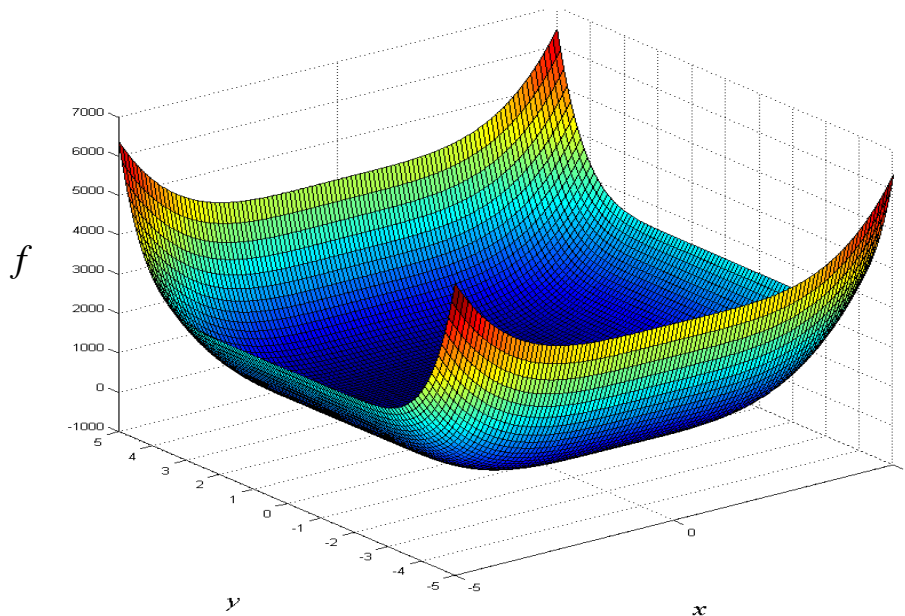


Figure 25 : courbe exacte de la fonction Camel Back

SCT	SCR	SCE	R2	R2A	RMSECV
1.4985E+10	1.5244E+10	4262780	0.999716	0.9773368	20.4421

Tableau 3 : Analyse de la variance de la fonction Camel Back

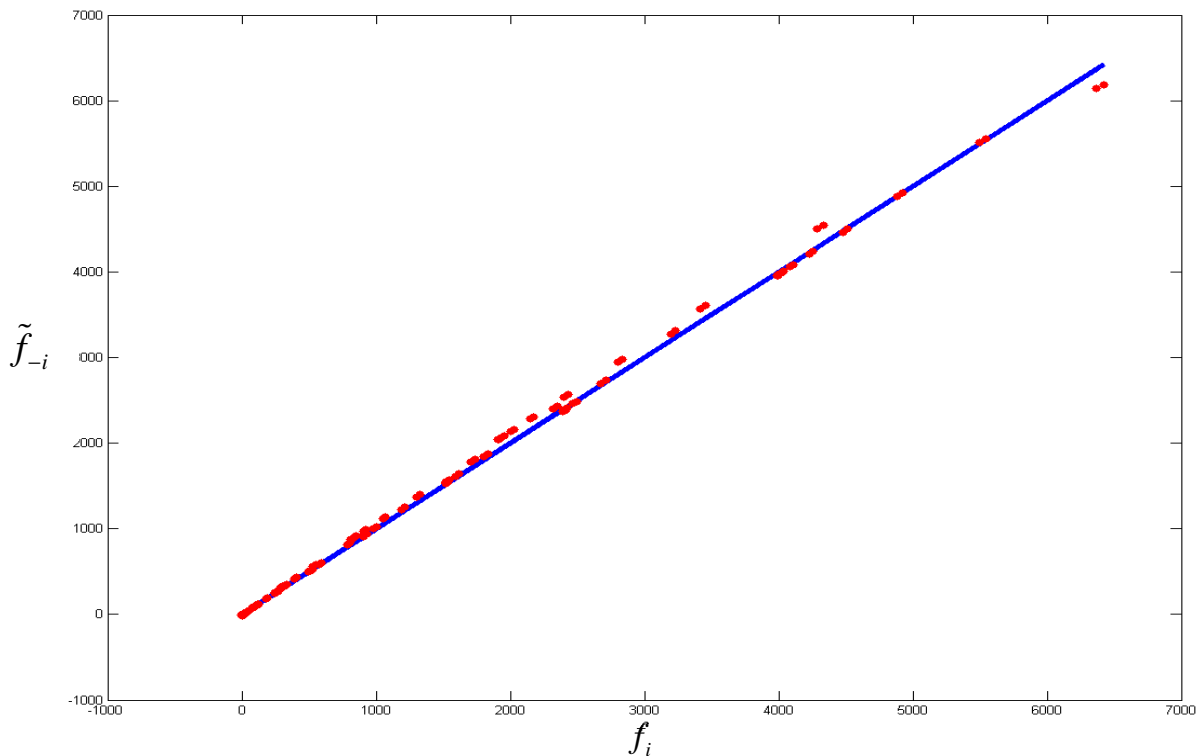


Figure 26 : courbe de la validation croisée de la fonction Camel Back

La courbe de la validation croisée de la fonction bi-extrema, obtenue avec 20 points maîtres, (Figure 24) et celle de Camel Back, obtenue avec 400 points maîtres (Figure 26), montrent que les solutions approchées \tilde{f}_{-i} convergent vers les droites $\tilde{f}_{-i} = f_i$. Cette convergence confirme la précision de notre métamodèle.

Les résultats de la technique d'analyse de la variance de la fonction bi-extrema (Tableau 2) et Camel back (Tableau 3) montrent que les variables R^2 et R_A^2 sont très proches de 1 et que $RMSE_{CV}$ tend vers 0 pour la première fonction et vers 20.4421 pour la deuxième, qui est considérée comme une valeur faible car les valeurs de cette dernière varie entre -1.316 et 6420 (Figure 25). Ces résultats confirment la précision de notre métamodèle

II.5 ÉTUDE DE L'ERREUR DU METAMODELE M-MDFSM

L'utilisation de l'erreur d'approximation locale $\Delta\tilde{f}$ permet à la méthode d'optimisation d'estimer la précision de la valeur de la fonction approchée \tilde{f} et d'exploiter davantage cette information. Donc il est très important de trouver une bonne formulation de l'erreur d'approximation locale, qui joue un rôle majeur dans la suite de cette thèse. L'erreur d'approximation locale sera utilisée pour l'enrichissement du métamodèle. Elle aura servi

pour le choix des nouveaux points maîtres. Cette erreur nous permet d'appliquer la notion de fonction mérite, qui est la différence entre la valeur approchée de la fonction coût \tilde{f}_i et l'erreur d'approximation locale $\Delta\tilde{f}_i$ au point i :

$$f_{merit}^i = \tilde{f}_i - \Delta\tilde{f}_i \quad (\text{II.22})$$

Si on utilise seulement la fonction approchée comme critère de sélection, les solutions les plus prometteuses seront sélectionnées. Par contre, l'utilisation de la fonction mérite permet d'étendre la recherche dans le reste de domaine inexploré, en tenant compte de l'estimation de l'erreur. Donc l'utilisation de la fonction mérite permet de sélectionner les individus avec un potentiel de bonne performance.

Dans cette partie, on s'intéresse à la recherche de l'erreur d'approximation locale $\Delta\tilde{f}_i$ commise par le métamodèle au point d'approximation i . Pour aboutir à un meilleur choix, différents types d'erreur sont testés.

II.5.1 Erreur MC1

En se basant sur la méthode de moindre carrée, on approche l'erreur d'approximation locale, MC1, au point i produit par le développement en séries de Taylor à l'ordre deux (Eq (II.12)) d'un point j au voisinage de ce point i par :

$$\forall j, \frac{(\text{MC1})_i}{\|X_{ji}\|^3} = \frac{a + \sum_{k=1}^n b_k x_{ji}^k + \frac{1}{2} \sum_{k,l=1}^n c_{kl} x_{ji}^k x_{ji}^l - f_j}{\|X_{ji}\|^3} \Rightarrow \forall j, \frac{((\text{MC1})_i)^2}{\|X_{ji}\|^6} = \frac{\left(a + \sum_{k=1}^n b_k x_{ji}^k + \frac{1}{2} \sum_{k,l=1}^n c_{kl} x_{ji}^k x_{ji}^l - f_j \right)^2}{\|X_{ji}\|^6}$$

Pour tous les λ points maîtres l'erreur s'écrit :

$$\forall j, ((\text{MC1})_i)^2 \sum_{j=1}^{\lambda} \omega_j = \sum_{j=1}^{\lambda} \omega_j \left(a + \sum_{k=1}^n b_k x_{ji}^k + \frac{1}{2} \sum_{k,l=1}^n c_{kl} x_{ji}^k x_{ji}^l - f_j \right)^2 = 2E(a, \mathbf{b}, \mathbf{c})$$

Finalement l'erreur d'approximation locale MC1 au point i est donnée par :

$$(\text{MC1})_i = \sqrt{2 \left(\sum_{j=1}^{\lambda} \omega_j \right)^{-1} E(a, \mathbf{b}, \mathbf{c})} \quad (\text{II.23})$$

Où a , \mathbf{b} et \mathbf{c} sont respectivement \tilde{f}_i , $\nabla\tilde{f}_i$ et $\nabla^2\tilde{f}_i$, qui sont les solutions de l'équation (Eq (II.19)).

II.5.2 Erreur EAP

Lorsque on effectue un développement de Taylor à l'ordre deux, une erreur d'approximation se produit. Cet erreur est de l'ordre de $\|X_{ji}\|^3$. On peut formuler l'erreur a priori, EAP, de la manière suivante :

$$(\text{EAP})_i = \left(\sum_{j=1}^{\lambda} \|X_{ji}\|^{-3} \right)^{-1}$$

Cette formulation de l'erreur au point i est indépendante des fonctions approximées \tilde{f}_i . Pour rendre cette erreur proportionnelle aux fonctions \tilde{f}_i , elle sera multipliée par un facteur d'adimensionnement. Ce facteur est composé de deux termes :

Au début, l'erreur est adimensionnée par sa multiplication par l'inverse de sa valeur maximale :

$$\Delta \tilde{f}_{\max} = \max \left(\left(\sum_{j=1}^{\lambda} \|X_{ji}\|^{-3} \right)^{-1} \right) \Rightarrow (\text{EAP})_i = \left(\sum_{j=1}^{\lambda} \|X_{ji}\|^{-3} \right)^{-1} \times \frac{1}{\Delta \tilde{f}_{\max}}$$

Ensuite, pour la rendre proportionnelle à la fonction approchée \tilde{f}_i , on la multiplie par l'écart type :

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\tilde{f}_i - \bar{\tilde{f}})^2}$$

Où $\bar{\tilde{f}}$ est la moyenne des N solutions approchées.

Finalement l'erreur d'approximation locale, EAP, au point i est donnée par :

$$(\text{EAP})_i = \left(\sum_{j=1}^{\lambda} \|X_{ji}\|^{-3} \right)^{-1} \times \frac{\sigma}{\Delta \tilde{f}_{\max}} \quad (\text{II.24})$$

II.5.3 Erreur MC

La formulation de l'erreur MC est semblable à celle de l'erreur MC1. Cette dernière dépend de l'erreur moyenne quadratique $E(a, \mathbf{b}, \mathbf{c})$ résultant d'un développement en séries de Taylor à l'ordre deux d'un point maître j au voisinage d'un point inconnu i . Par contre, l'erreur MC dépend de l'erreur moyenne quadratique $E(\tilde{f})$ résultant d'un développement en séries de Taylor à l'ordre deux d'un point inconnu i au voisinage d'un point maître j :

$$\forall j, \tilde{f}_i = f_j + X_{ij}^T \nabla f_j + \frac{1}{2} X_{ij}^T \nabla^2 f_j X_{ij} + O(\|X_{ij}\|^3)$$

L'erreur moyenne quadratique $E(\tilde{f}_i)$ s'écrit :

$$E(\tilde{f}_i) = \frac{1}{2} \sum_{j=1}^{\lambda} \left(\frac{f_j + X_{ij}^T \nabla f_j + \frac{1}{2} X_{ij}^T \nabla^2 f_j X_{ij} - \tilde{f}_i}{\|X_{ij}\|^3} \right)^2$$

Où : \tilde{f}_i est l'approximation de la valeur de la fonction f_i au point i obtenu par la résolution de l'équation (Eq (II.19)) et $X_{ij} = X_i - X_j$.

Pour calculer l'erreur moyenne quadratique $E(\tilde{f})$, on a besoin de la connaissance des valeurs du gradient et de la hessienne au point maître j . Ces valeurs ne sont pas toujours disponibles. Pour surmonter ce problème, on suppose qu'une petite perturbation du point maître j (j devient $j' = j + \varepsilon$) est accompagnée d'une variation négligeable de valeurs de la fonction, du gradient et de la hessienne :

$$f_j \approx \tilde{f}_{j'}, \nabla f_j \approx \nabla \tilde{f}_{j'} \text{ et } \nabla^2 f_j \approx \nabla^2 \tilde{f}_{j'}$$

Les valeurs de $\tilde{f}_{j'}$, $\nabla \tilde{f}_{j'}$ et $\nabla^2 \tilde{f}_{j'}$ aux λ points $j' = j + \varepsilon$ sont approximées par le métamodèle (Eq (II.19)) construit par les λ points maîtres j .

En remplaçant les valeurs exactes de f_j , ∇f_j et $\nabla^2 f_j$ aux points maîtres j par les valeurs de leurs approximations $\tilde{f}_{j'}$, $\nabla \tilde{f}_{j'}$ et $\nabla^2 \tilde{f}_{j'}$ au point $j' = j + \varepsilon$ dans la formulation précédente de l'erreur moyenne quadratique $E(\tilde{f})$, on trouve :

$$E(\tilde{f}_i) = \frac{1}{2} \sum_{j=1}^{\lambda} \left(\frac{\tilde{f}_{j'} + X_{ij'}^T \nabla \tilde{f}_{j'} + \frac{1}{2} X_{ij'}^T \nabla^2 \tilde{f}_{j'} X_{ij'} - \tilde{f}_i}{\|X_{ij'}\|^3} \right)^2$$

Enfin, l'erreur d'approximation locale MC au point i est calculée à partir de l'équation suivante :

$$(MC)_i = \sqrt{2 \left(\sum_{j=1}^{\lambda} \|X_{ij'}\|^{-6} \right)^{-1}} E(\tilde{f}_i) \quad \text{(II.25)}$$

II.5.4 Erreur H1

L'utilisation d'une erreur d'approximation locale résultant de développement en séries de Taylor à l'ordre deux abouti à des erreurs très faibles qui sont de l'ordre de $\|X_{ji}\|^3$. Ces erreurs sont généralement négligeables devant les valeurs des fonctions approximées. Donc l'utilisation de la fonction mérite (Eq (II.22)) revient à utiliser \tilde{f} . Pour trouver une erreur d'approximation locale significative, on propose de chercher l'erreur liée au développement en séries de Taylor à l'ordre un, qui est de l'ordre de $\|X_{ji}\|^2$. On suppose que l'erreur d'approximation locale H1 au point i , qui est liée à l'ordre un, est la différence entre la solution approchée \tilde{f}_i obtenue par le développement en séries de Taylor à l'ordre deux et celle obtenue par un développement d'ordre un. En prenant compte de cette hypothèse et de l'ordre de l'erreur, on peut écrire l'erreur H1 sous la forme :

$$(H1)_i = \sqrt{\left(\sum_{j=1}^{\lambda} \|X_{ji}\|^{-4} \right)^{-1} \sum_{j=1}^{\lambda} \left(\frac{\frac{1}{2} X_{ji}^T \nabla^2 \tilde{f}_i X_{ji}}{\|X_{ji}\|^2} \right)^2} \quad (II.26)$$

II.5.5 Analyse de l'erreur

Afin de sélectionner la meilleure erreur d'approximation locale parmi les quatre erreurs présentées précédemment, on utilise le critère de la convergence asymptotique comme critère de choix.

II.5.5.1 Convergence asymptotique

La convergence asymptotique consiste à discrétiser l'espace de recherche en ψ parties avec un pas de discrétisation h . Puis, on calcule l'erreur de discrétisation $e(h)$:

$$e(h) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta f_i - \tilde{\Delta f}_i)^2} \quad (II.27)$$

Où $\Delta f_i = f_i - \tilde{f}_i$ est l'erreur exacte au point i .

$\tilde{\Delta f}_i$ est l'erreur d'approximation locale au point i .

N est le nombre de points approximés.

L'erreur de discrétisation $e(h)$ est calculée pour différents pas. La meilleure erreur d'approximation locale retenue est celle respectant le critère suivant :

Si le pas de discrétisation h décroît, l'erreur de discrétisation $e(h)$ décroît.

II.5.5.2 Application de critère de convergence asymptotique

Pour étudier la convergence asymptotique, on approxime la fonction bi-extrema (Eq (II.9)) en utilisant $N=10000$ points. On commence avec $h=0.25$ (5 points maîtres) et pour chaque nouvelle approximation, on divise le pas h précédent par 2 (le nombre final de points maîtres est 129).

Pour sélectionner la meilleure erreur, on étudie la convergence asymptotique de chacune des quatre erreurs précédentes et son évolution avec l'erreur exacte Δf_i (ici, on présente l'évolution des erreurs d'approximation locales avec l'erreur exacte pour le métamodèle, M-MDFSM, construit avec neuf points maîtres ($h=0.125$) (Figure 27)).

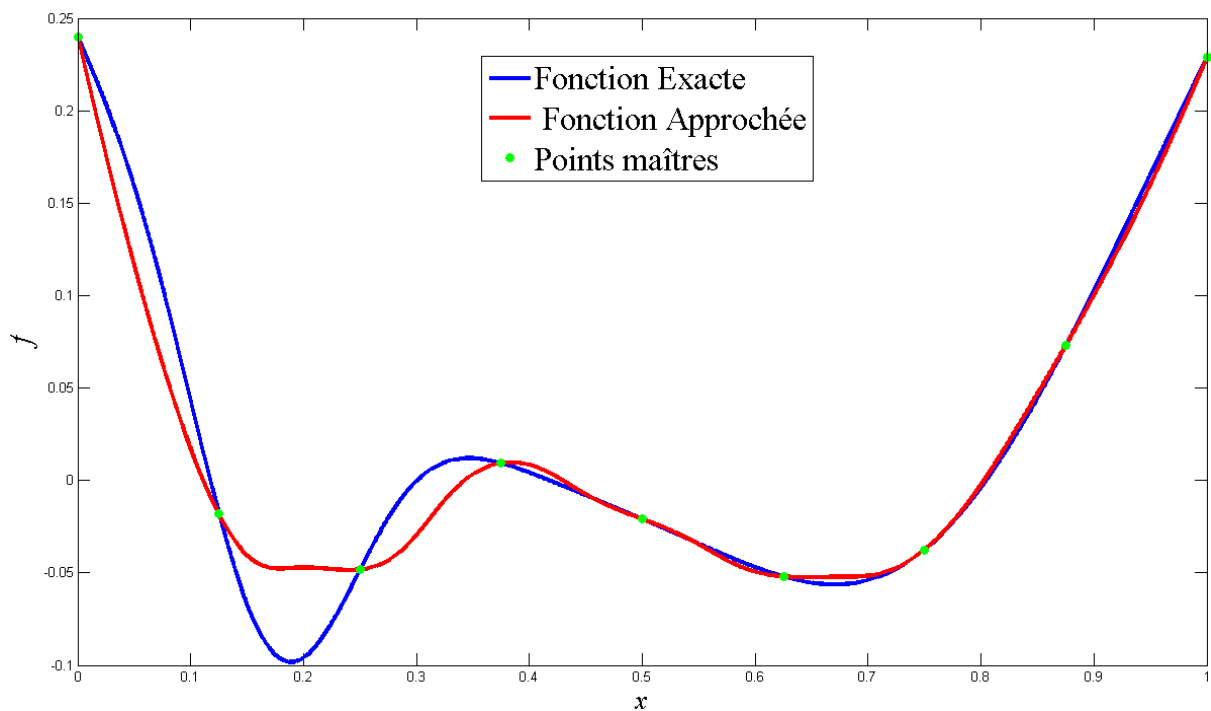


Figure 27 : Approximation de la fonction bi-extrema avec neuf points maîtres

L'étude de l'évolution des quatre erreurs d'approximation locale (MC1, EAP, MC et H1) avec l'erreur exacte (Figure 28) montre que :

- L'erreur MC1 est négligeable devant l'erreur exacte (Figure 28-(a)) et devant les valeurs de la fonction. On peut déduire que l'utilisation de la fonction mérite $(\tilde{f} - MC1)$ pour l'amélioration du métamodèle en ajoutant le point ayant $\min(\tilde{f} - MC1)$ revient à utiliser $\min(\tilde{f})$ comme critère de choix. Donc notre recherche se limite à l'exploitation de l'espace de recherche.
- L'erreur EAP évolue d'une manière uniforme (Figure 28-(b)). Cet erreur ne peut pas être retenue car elle dépend seulement du l'erreur a priori et pas de la fonction approximée.
- Les deux erreurs MC et H1 évoluent de la même manière que l'erreur exacte (Figure 28-(c) et (d)).

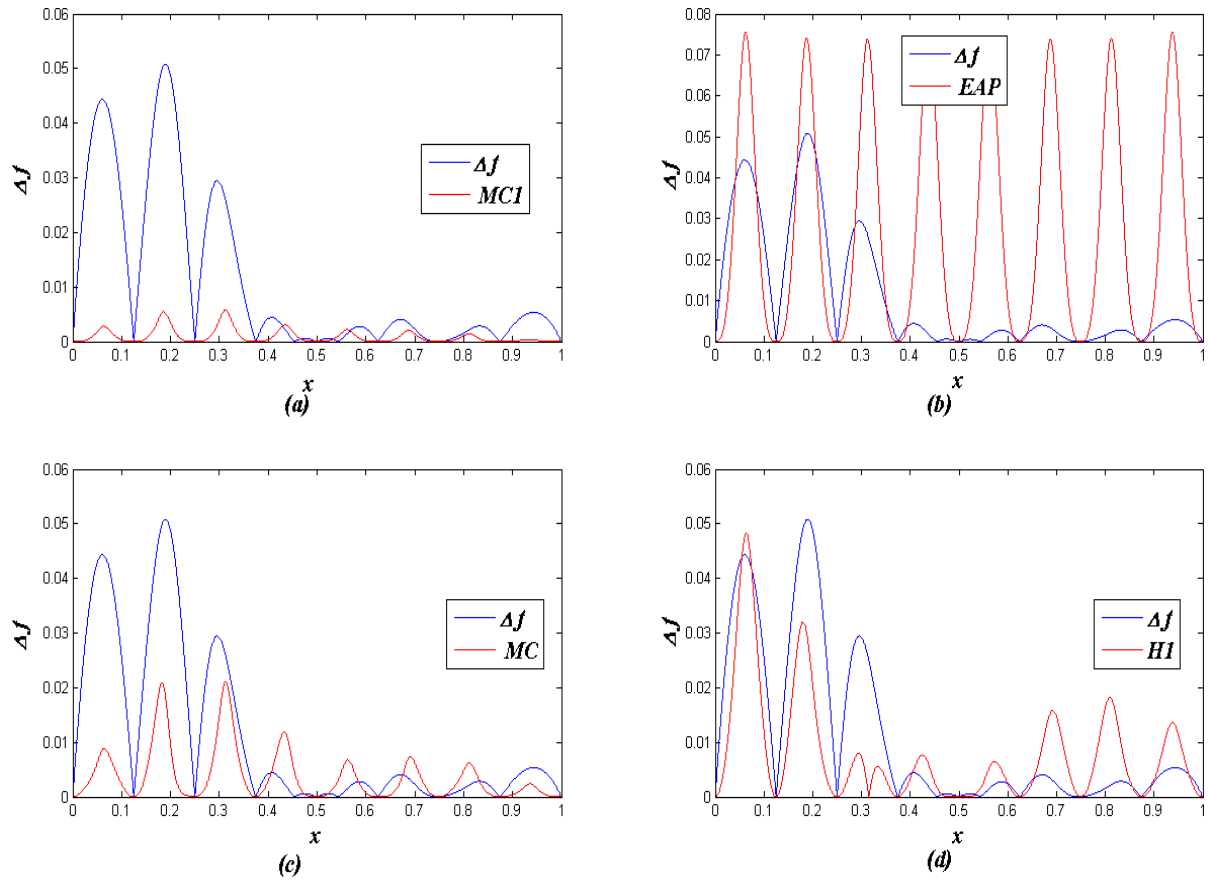


Figure 28 : Comparaison entre l'évolution de l'erreur exacte et des erreurs d'approximation locale : (a) erreur MCI, (b) erreur EAP, (c) erreur MC, (d) erreur HI.

La convergence asymptotique des erreurs d'approximation locale, présenté sur la Figure 29, montre que l'erreur EAP est la seule qui viole le critère de la convergence asymptotique.

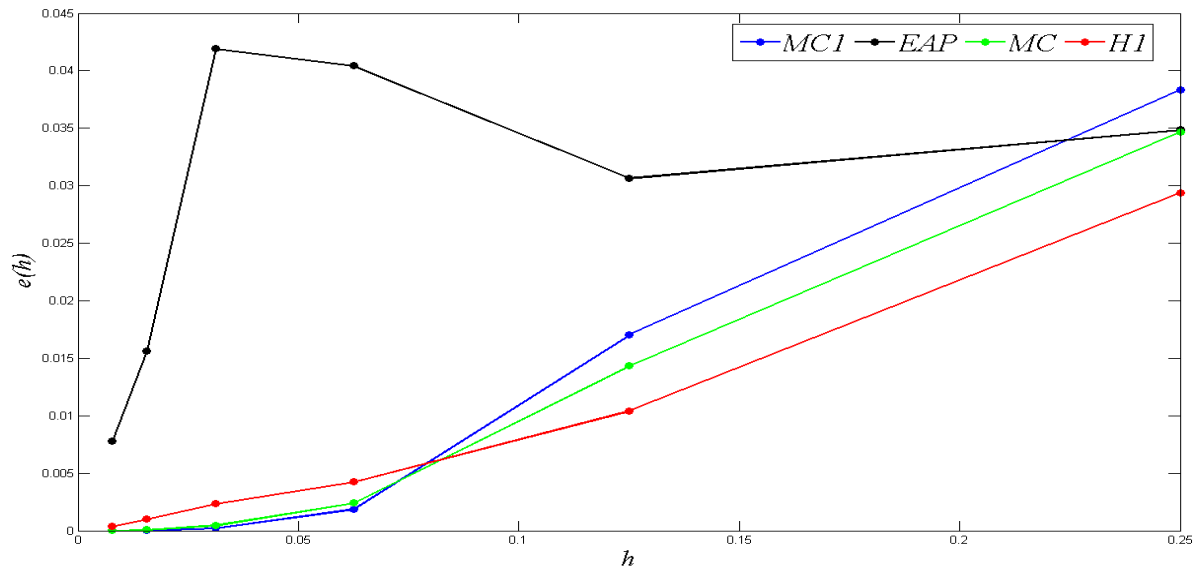


Figure 29 : Convergence asymptotique des erreurs d'approximation locale

On peut conclure que les deux erreurs MC et H1 sont les meilleures erreurs d'approximation locale parmi les quatre erreurs proposées. Ces erreurs respectent le critère de convergence asymptotique. Elles évoluent d'une manière proche de l'évolution de l'erreur exacte.

Pour sélectionner la meilleure erreur, les erreurs MC et H1 seront utilisées dans le chapitre suivant pour l'enrichissement du métamodèle, M-MDFSM, qui sera couplé avec NSGA-II pour la résolution des problèmes d'optimisation multi-objectifs analytiques. Suivant les résultats obtenus, on choisira l'erreur la plus prometteuse.

II.6 ÉTUDE DE BRUIT

II.6.1 Introduction

Dans le domaine de mise en forme, les résultats des simulations (MEF) représentent généralement des bruits. Si on lance deux fois le même cas avec des maillages différents, on obtient des résultats différents, principalement du fait que l'erreur de discrétisation élément fini n'est pas la même dans les deux cas ; cette erreur peut être quantifiée en effectuant les calculs sur un maillage beaucoup plus fin. La variation des résultats à l'intérieur de cette précision éléments finis provient donc simplement de la différence des discrétisations. Elle n'a pas de sens physique. On peut donc la considérer comme un bruit numérique, et la traiter comme telle.

La construction de notre métamodèle avec des solutions bruitées conduit à des mauvaises approximations des fonctions. L'application du couplage de notre métamodèle, moins précis, avec NSGA-II ([Chapitre III](#)) sur un problème d'optimisation en mise en forme ne garantit pas la convergence vers des solutions optimales. Ce problème de divergence peut être surmonté par l'amélioration des approximations de notre métamodèle. Ces améliorations peuvent s'atteindre par la prise en compte du bruit. Pour ce faire, on cherche des nouvelles positions des λ points maîtres qui nous permettent d'avoir des interpolations lisses :

$$\forall j, f_j = \bar{f}_j + \delta_j \quad (\text{II.28})$$

Où :

f_j : la nouvelle valeur de la fonction prenant en compte le bruit au point maître j .

\bar{f}_j : la valeur initiale de la fonction bruitée au point maître j .

δ_j : la valeur du bruit au point maître j .

II.6.2 Métamodèle avec bruit

On suppose que le bruit a une distribution gaussienne de probabilité $(0, \delta_{\max})$:

$$\forall j, p(\delta_j) = e^{-\left(\frac{\delta_j}{\delta_{\max}}\right)^2} \quad (\text{II.29})$$

Où δ_{\max} est la valeur maximale du bruit au point maître j .

L'idée est de trouver les solutions δ_j , aux λ points maîtres, les plus probables qui permettent la minimisation de la hessienne de \tilde{f} :

$$\Phi(\delta) = \int_{\Omega} \|\nabla^2 \tilde{f}\|^2 d\omega \quad (\text{II.30})$$

Minimiser la hessienne (chercher des interpolations lisses) revient à trouver $\delta = (\delta_j), j = 1, \dots, \lambda$ qui minimise $\Phi(\delta)$:

$$\forall j = 1, \dots, \lambda, \quad \frac{\partial \Phi}{\partial \delta_j}(\delta) = 0 \quad (\text{II.31})$$

Selon L'équation (Eq (II.19)), la fonction, le gradient et la hessienne au point i sont donnés par :

$$\begin{pmatrix} \tilde{f}_i \\ \nabla \tilde{f}_i \\ \nabla^2 \tilde{f}_i \end{pmatrix} = \mathbf{A}^{-1} \mathbf{B}$$

Et plus particulièrement, la hessienne est donné par la notation suivante :

$$\forall i, \quad \nabla^2 \tilde{f}_i = (\mathbf{A}^{-1} \mathbf{B})_{\nabla^2 \tilde{f}}$$

$$\mathbf{B} = \sum_{j=1}^{\lambda} \omega_j f_j \mathbf{V}_j \Rightarrow \forall i, \quad \nabla^2 \tilde{f}_i = \left(\mathbf{A}^{-1} \sum_{j=1}^{\lambda} \omega_j f_j \mathbf{V}_j \right)_{\nabla^2 \tilde{f}} = \left(\mathbf{A}^{-1} \sum_{j=1}^{\lambda} \omega_j \mathbf{V}_j (\bar{f}_j + \delta_j) \right)_{\nabla^2 \tilde{f}}$$

Donc on obtient :

$$\forall j = 1, \dots, \lambda \text{ et } \forall i, \quad \frac{\partial (\nabla^2 \tilde{f}_i)}{\partial \delta_j} = \omega_j (\mathbf{A}^{-1} \mathbf{V}_j)_{\nabla^2 \tilde{f}}$$

L'équation (Eq (II.31)) devient:

$$\begin{aligned}
\forall j=1,\dots,\lambda, \quad \frac{\partial \Phi}{\partial \delta_j}(\delta) = 0 &\Leftrightarrow \int_{\Omega} \omega_j (A^{-T} V_j)_{\nabla^2 \bar{f}} \cdot \sum_{j'=1}^{\lambda} \omega_{j'} (\bar{f}_{j'} + \delta_{j'}) (A^{-T} V_{j'})_{\nabla^2 \bar{f}} d\omega = 0 \\
&\Leftrightarrow \sum_{j'=1}^{\lambda} \left(\int_{\Omega} \omega_j \omega_{j'} (A_i^{-1} V_j^i)_{\nabla^2 \bar{f}} \cdot (A_i^{-1} V_{j'}^i)_{\nabla^2 \bar{f}} d\omega \right) (\bar{f}_{j'} + \delta_{j'}) = 0 \quad \text{(II.32)} \\
&\Leftrightarrow \sum_{j'=1}^{\lambda} C_{jj'} (\bar{f}_{j'} + \delta_{j'}) = 0
\end{aligned}$$

Si on note $\forall j=1,\dots,\lambda$, $D_j = \sum_{j'} C_{jj'} \bar{f}_{j'}$, l'équation (Eq (II.32)) s'écrit sous la forme matricielle suivante :

$$C \delta = -D \quad \text{(II.33)}$$

La recherche des solutions δ_j les plus probables revient à maximiser la probabilité de distribution qui est équivalent à minimiser la fonction suivante :

$$\varphi(\delta) = \sum_j e^{\left(\frac{\delta_j}{\delta_{\max}} \right)^2} \quad \text{(II.34)}$$

Enfin, minimiser le bruit revient à minimiser la fonction suivante :

$$\hat{\Phi}(\delta) = \Phi(\delta) + \omega \varphi(\delta) \quad \text{(II.35)}$$

ω est un coefficient de pondération qui peut être sous la forme : $\omega = \omega_0 \frac{\Phi(0)}{\lambda}$. Où ω_0 est un coefficient numérique proche de 1.

La minimisation de la fonction (Eq (II.35)) donne :

$$\frac{\partial \hat{\Phi}}{\partial \delta_j}(\delta) = C \delta + D + \omega E(\delta) \delta \quad \text{(II.36)}$$

Où :

$$\forall j, j' = 1, \dots, \lambda, E_{jj'}(\delta) = \begin{cases} \frac{2}{\delta_{\max}^2} e^{\left(\frac{\delta_j}{\delta_{\max}}\right)^2} & \text{si } j = j' \\ 0 & \text{si non} \end{cases}$$

Pour résoudre l'équation (Eq (II.36)), on utilise l'algorithme de Newton Raphson avec :

$$\begin{aligned} R(\delta) &= C\delta + D + \omega E(\delta)\delta \\ \frac{\partial R(\delta)}{\partial \delta} &= C + \left(\frac{2\delta_j^2}{\delta_{\max}^2} + 1\right)\omega E(\delta) \end{aligned} \quad (\text{II.37})$$

II.6.3 Application

Notre méthode de calcul de bruit est validée sur les quatre fonctions suivantes :

Fonction sinus : $f(x) = \sin(x)$, $x \in [-4, 4]$, $\delta_{\max} = 0.3$

Fonction linéaire : $f(x) = x$, $x \in [-4, 4]$, $\delta_{\max} = 0.8$

Fonction quadratique 1D : $f(x) = x^2$, $x \in [-4, 4]$, $\delta_{\max} = 0.5$

Et la fonction bi-extrema (Eq (II.9)) avec un bruit maximal $\delta_{\max} = 0.03$.

Pour les trois premières fonctions, le métamodèle est construit avec 21 points maîtres \bar{f}_j et pour la fonction bi-extrema, le métamodèle est construit avec 20 points. Pour créer du bruit, certains points maîtres ont été perturbés (les points verts) ($\bar{f}_j = \bar{f}_j + \delta_j$). Le métamodèle résultant de cette perturbation (métamodèle sans prise en compte de bruit) est utilisé pour approximer $N = 1000$ points. Ensuite, pour chacune des fonctions, on cherche les nouveaux points maîtres (les points rouges) ($f_j = \bar{f}_j + \delta_j$) qui permettent de minimiser le bruit. Enfin on reconstruit le nouveau métamodèle avec ces nouveaux points maîtres, qui sera utilisé pour l'approximation de chacune des fonctions (les courbes rouges).

Les figures (Figure 30, Figure 31, Figure 32 et Figure 33) qui présentent les solutions exactes (courbes bleues), les métamodèles sans prise en compte de bruit (courbes vertes) et les métamodèles avec prise en compte de bruit (courbes rouges), montrent l'efficacité de notre méthode. En effet, pour les quatre fonctions on arrive à trouver les nouvelles positions des points maîtres ($f_j = \bar{f}_j + \delta_j$) qui permettent à notre métamodèle d'effectuer des interpolations lisses et sans bruit (courbes rouges) toute en respectant l'intervalle de variation de bruits $\forall j = 1, \dots, 21, |\delta_j| \leq \delta_{\max}$.

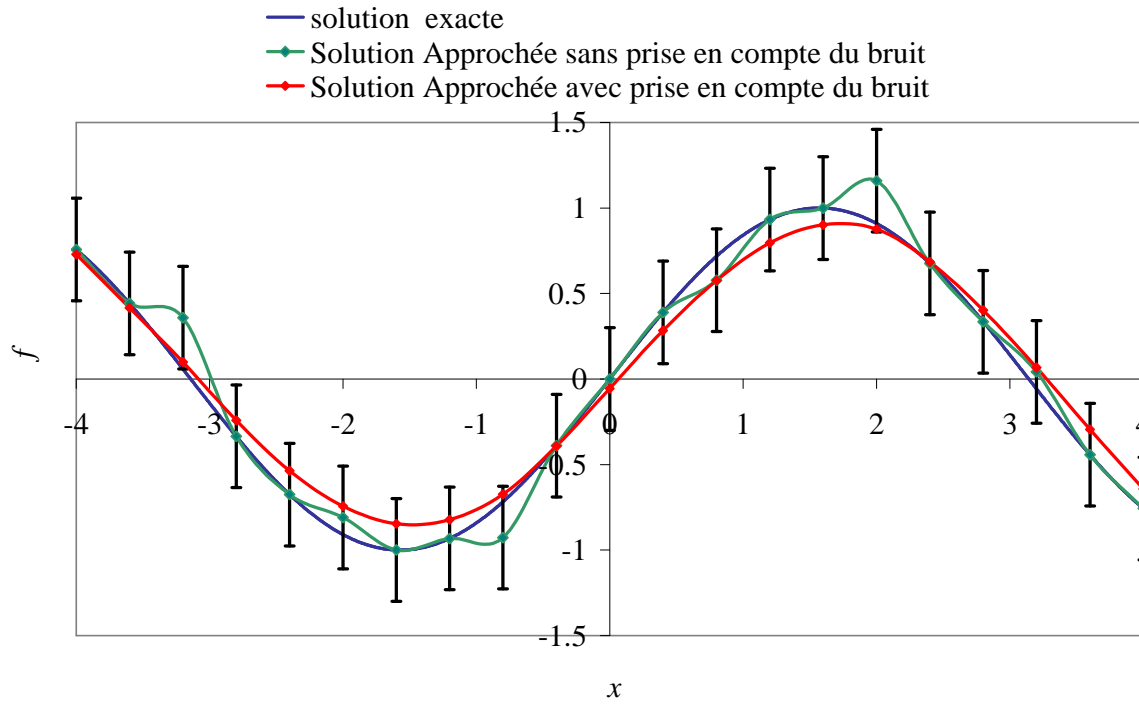


Figure 30 : Calcul de bruit de la fonction sinus

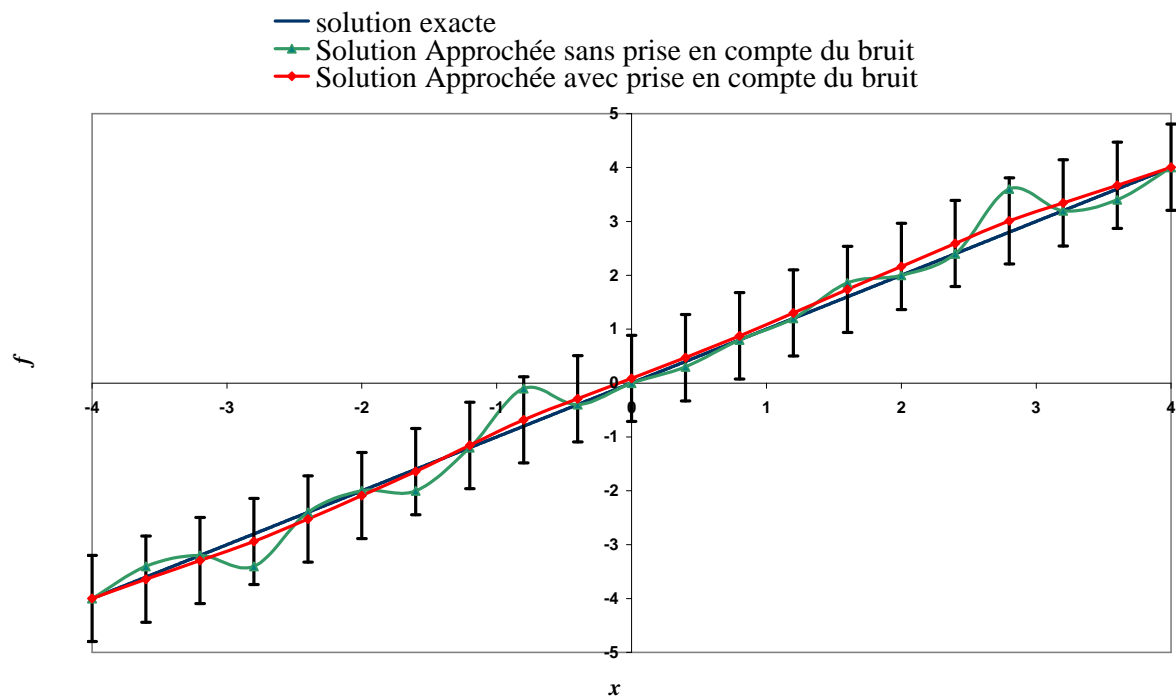


Figure 31 : Calcul de bruit de la fonction linéaire

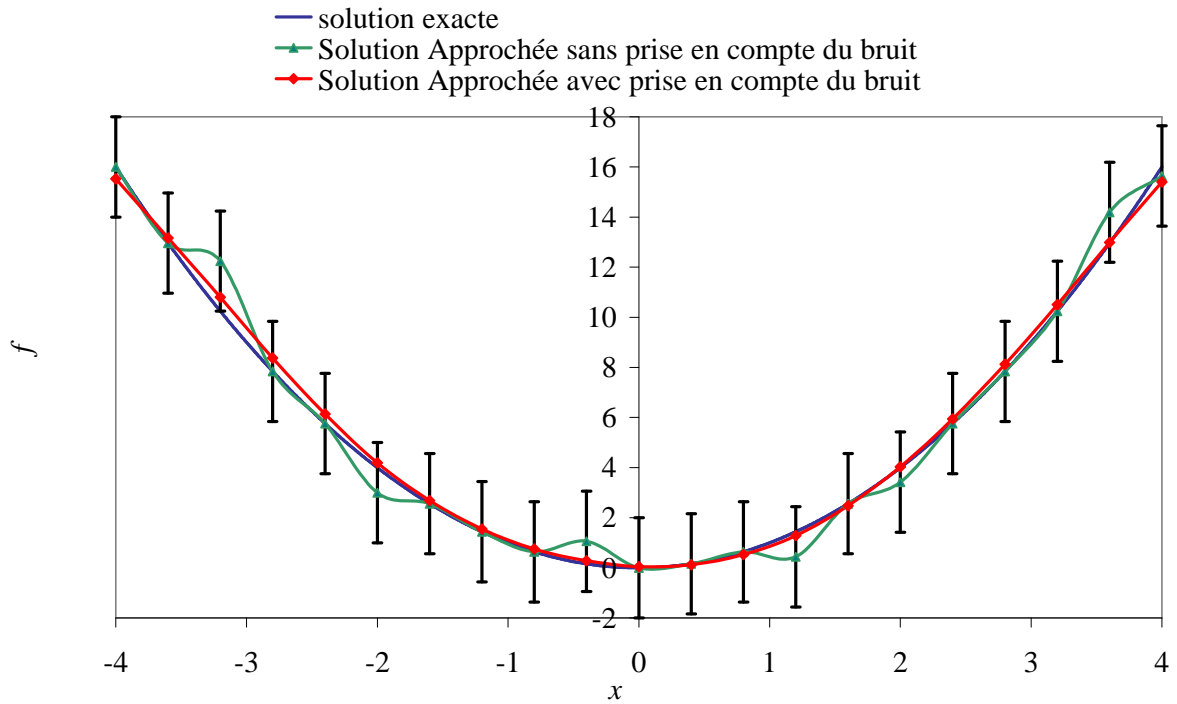


Figure 32 : Calcul de bruit de la fonction quadratique 1D

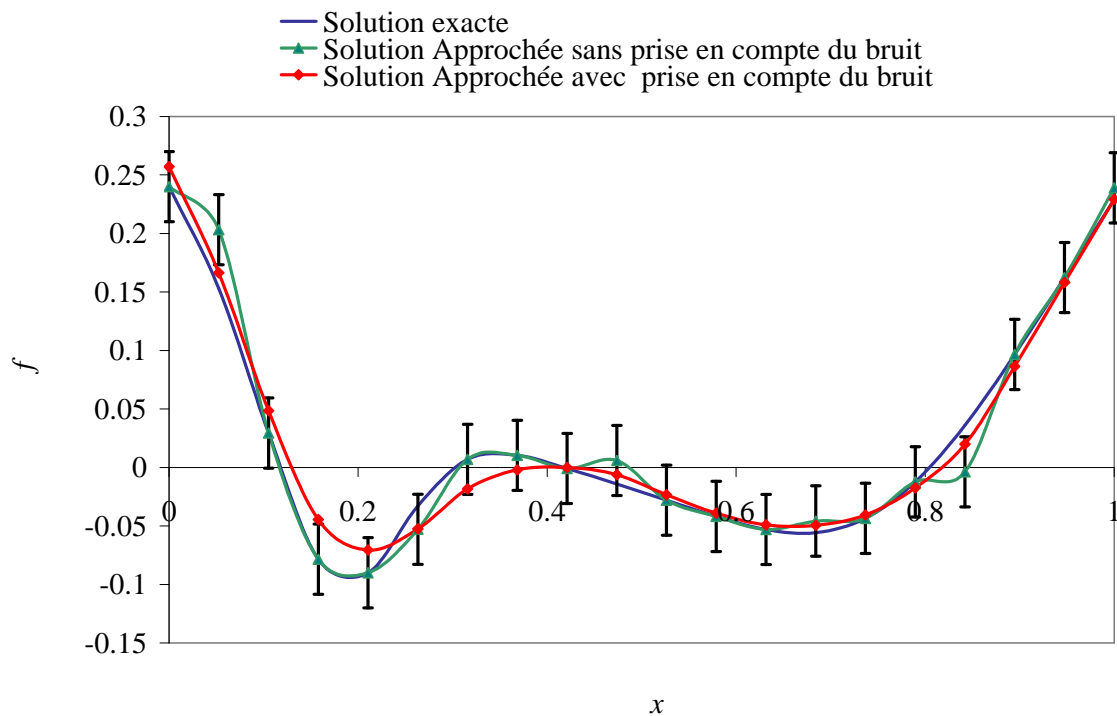


Figure 33 : Calcul de bruit de la fonction bi-extrema

II.7 CONCLUSION

Dans ce chapitre, on a détaillé la construction du métamodèle basé sur la méthode de différences finies sans maillage. En effet, on a commencé par la construction du métamodèle d'ordre un, M-MDFSM-1. L'inconvénient de ce dernier est le manque de précision des approximations. Ce manque de précision est dû à la nature des interpolations, qui sont des interpolations linéaires. Pour améliorer la précision de notre métamodèle, on a introduit un développement en série de Taylor à l'ordre deux. Le métamodèle d'ordre deux, M-MDFSM-2, a augmenté la précision des résultats mais il rencontre un problème de singularité de sa matrice A dû au manque d'information, comme le non respect du nombre minimal de points maîtres et la mauvaise dispersion de ces derniers dans l'espace de recherche. Pour surmonter cette faiblesse et obtenir un modèle précis, on a combiné les deux métamodèles précédents pour aboutir à un métamodèle régularisé, M-MDFSM. Ce nouveau métamodèle est un modèle quadratique en présence des informations nécessaires. Dans le cas contraire (manque d'informations), il devient un modèle linéaire. Ensuite ce métamodèle régularisé a été validé sur deux fonctions analytiques (Camel Back et bi-extrema). Après la construction et la validation de notre métamodèle, on a cherché les meilleures formulations de l'erreur d'approximation locale liée à ce dernier, MC et H1, parmi les quatre types d'erreur testées. Enfin, on a développé une méthode pour la prise en compte de bruit par notre métamodèle. Cette méthode a été validée sur des fonctions analytiques.

Pour atteindre l'objectif de la thèse, développement d'un algorithme d'optimisation multi-objectifs pour la résolution des problèmes de mise en forme, dans le chapitre suivant, on présente le couplage entre ce métamodèle, M-MDFSM, et l'algorithme génétique multi-objectifs, NSGA-II.

CHAPITRE III

COUPLAGE DU M-MDFSM AVEC NSGA-II

III.1 INTRODUCTION

Les problèmes d'optimisation multi-objectifs en mise en forme sont très coûteux en terme de temps de calcul. Ce coût de calcul très élevé rend alors l'optimisation très difficile à mettre en œuvre. Pour l'appliquer dans ce secteur, un couplage entre un métamodèle et un algorithme d'optimisation multi-objectifs est exigé. Ce couplage permet de bénéficier des avantages des algorithmes multi-objectifs, tels que l'obtention d'un ensemble de solutions optimales en une seule itération, la convergence vers le front optimal de Pareto (FOP) avec une bonne dispersion des points sur celui-ci même pour des problèmes particulièrement difficiles [Deb, 2001] [Emmerich et al, 2005] [Beume et al, 2007], toute en surmontant le nombre important d'évaluations des fonctions coût, qui est l'inconvénient majeur de cette famille d'algorithmes. Ce nombre important d'évaluations peut être considérablement réduit en utilisant des approximations construites à partir d'un métamodèle. Dans le cadre de ce travail, nous proposons de coupler le métamodèle inspiré de la méthode de différences finies sans maillage, M-MDFSM, développé au cours de cette thèse avec l'algorithme génétique élitiste de tri non dominé (NSGA-II) développé par Kalyanmoy Deb et ses étudiants. Le couplage de NSGA-II avec M-MDFSM (M-AEMO) peut être effectué de plusieurs manières, dont l'étude constitue le corps de ce chapitre. Ces algorithmes sont ensuite évalués sur plusieurs fonctions tests analytiques, avant d'être appliqués à des problèmes d'optimisation en mise en forme au quatrième chapitre.

Le reste de ce chapitre est structuré de la manière suivante : On commence par une description de différents types de couplage M-AEMO. Ensuite, ces approches sont comparées sur une série des problèmes tests analytiques dans la troisième partie.

III.2 COUPLAGE DE NSGA-II AVEC LE METAMODELE M-MDFSM

Dans cette section, nous considérons trois manières de coupler le métamodèle, M-MDFSM, à l'algorithme NSGA-II; elles diffèrent par le choix des points maîtres au cours des itérations et par l'évolution du métamodèle.

III.2.1 Couplage constant : C-Constant

Comme cela est souvent le cas avec les métamodèles utilisant la Méthodologie des Surfaces de Réponse, le couplage constant, C-Constant, présenté par l'organigramme de la **Figure 34** consiste tout simplement à utiliser un nombre fixe de points maîtres pour construire un métamodèle fixe. Une fois construit, il est utilisé pour remplacer les évaluations exactes durant toutes les itérations de NSGA-II. Ce dernier est construit à partir de λ_{\max} (nombre maximal de points maîtres) premiers individus de la population initiale P_0 , de taille N , générée aléatoirement par NSGA-II. Il est noté \mathbf{M}_0 . Autrement dit, tous les λ_{\max} premiers individus de la population initiale P_0 sont évalués exactement alors que les autres individus de cette population et de toutes les générations suivantes sont approximatés par le métamodèle fixe \mathbf{M}_0 .

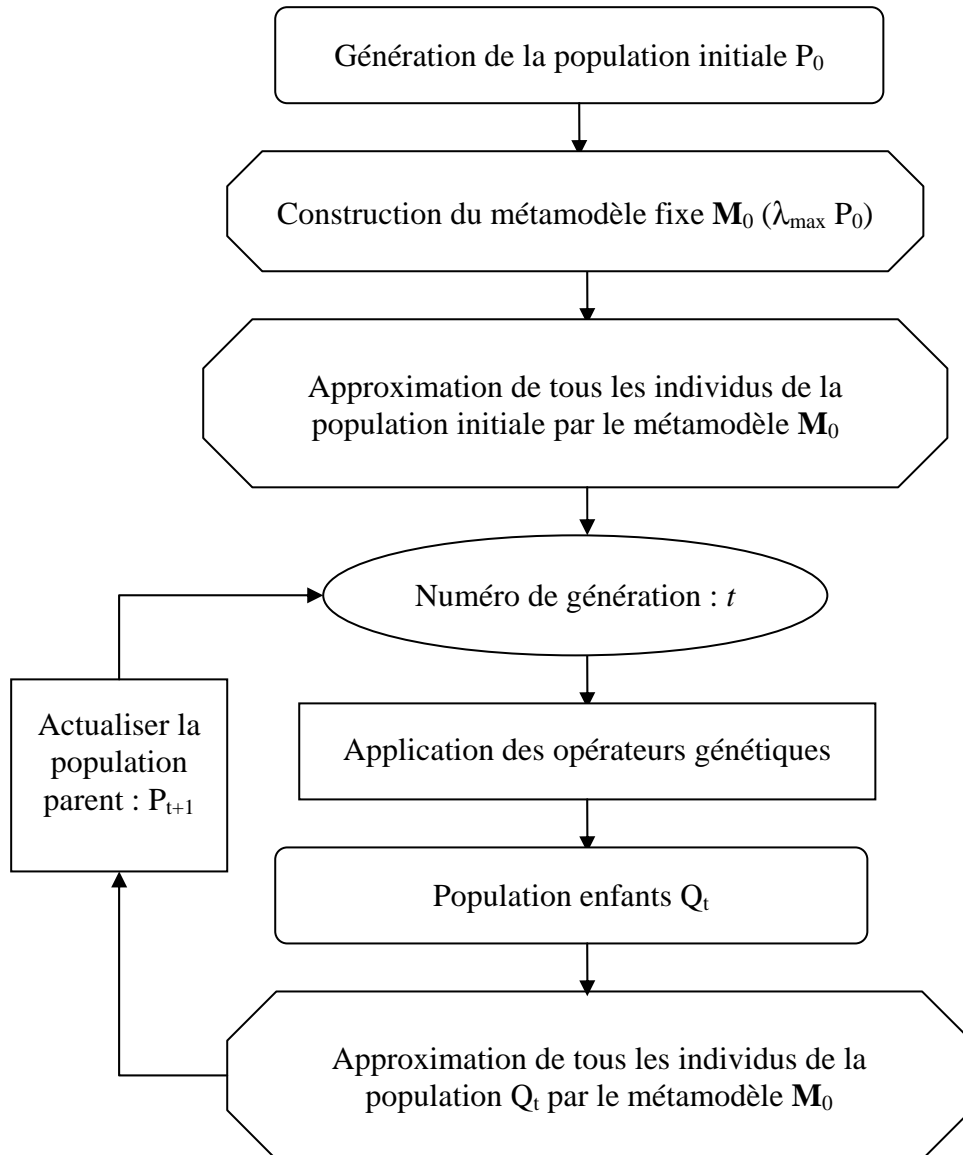


Figure 34– Organigramme de fonctionnement du C-Constant

III.2.2 Couplage Actualisé aléatoirement: C-Actualisé

Comme cela a été constaté dans la littérature [Bonte et al, 2010], et comme nous le verrons par la suite, le métamodèle constant est mal adapté aux fonctions complexes ayant de fortes variations locales. Un premier type d'amélioration est apporté en enrichissant le métamodèle, M-MDFSM, avec de nouveaux points maîtres localisés au voisinage de l'optimum, c'est-à-dire du front optimal de Pareto pour un problème d'optimisation multi-objectifs. Nous le nommons C-Actualisé. Son principe de fonctionnement est présenté par l'organigramme de la Figure 35. Dans le but d'avoir un métamodèle initial riche, on construit notre métamodèle initial, M_0 , avec seulement un nombre initial de points maîtres, bien répartis dans l'espace paramétrique, générés par le plan d'expérience latin hypercube (Figure 36). Nous retenons donc les $\lambda = \frac{(n+1) \times (n+2)}{2}$ individus, sous la contrainte que $\lambda \leq \lambda_{\max}$ (λ_{\max} est le nombre maximal de points maîtres). Les fonctions coût sont alors évaluées exactement en ces points.

Le métamodèle \mathbf{M}_0 est ensuite utilisé pour approximer les valeurs de tous les individus de la population parent initiale P_0 . À chaque nouvelle génération t de l'algorithme génétique, et tant que $t \leq N_u$, N_u étant le nombre maximal de générations où l'enrichissement est autorisé (Eq (III.1)), le métamodèle \mathbf{M}_t est amélioré en lui ajoutant $\theta=2^{M-1}$ nouveaux points maîtres, choisis aléatoirement dans la population enfant courante, où les M fonctions coût sont évaluées exactement pour construire le métamodèle \mathbf{M}_{t+1} . Ce nouveau métamodèle est alors utilisé pour évaluer les valeurs des fonctions pour tous les individus de la population enfant Q_t , ainsi que pour corriger les évaluations précédentes de la population parent P_t .

$$N_u = \frac{\lambda_{\max} - \lambda}{\theta} \quad (\text{III.1})$$

Les θ premiers individus de la population enfant Q_t , sont retenus, dans l'ordre résultant de leur génération par les opérateurs génétiques. Ainsi, à chaque génération, le métamodèle s'enrichit de nouveaux points maîtres, potentiellement plus proches du front de Pareto, dans la mesure où NSGA-II converge vers celui-ci.

III.2.3 Couplage évolutif avec fonction mérite: C-Évolutif-FM

Le principe de fonctionnement de cette approche, C-Évolutif-FM, diffère de C-Actualisé uniquement par la manière de sélectionner les θ nouveaux points maîtres parmi la population P_t des parents. Plutôt que de les choisir de manière quasi-aléatoire, les individus sont classés suivant le critère de dominance et de distance d'encombrement, en fonction de leur position croissante sur les différents fronts et de leur distance décroissante sur chaque front. Cela permet de sélectionner les points les plus intéressants de P_t , au sens du métamodèle \mathbf{M}_t , et donc d'améliorer \mathbf{M}_t dans cette partie de l'espace. Toutefois, \mathbf{M}_t est seulement une approximation du modèle réel. Il est donc possible que l'amélioration obtenue à partir de ces points ne soit pas aussi intéressante qu'attendu, alors que des points non retenus auraient pu s'avérer bien plus intéressants. Cependant, ayant évalué une indication de cette erreur d'approximation locale $\Delta\tilde{f}$ nous pouvons estimer la fiabilité des prédictions et orienter notre choix vers les points potentiellement les plus intéressants. Pour cela, nous utilisons la fonction mérite, $f_{\text{mérite}} = \tilde{f} - \Delta\tilde{f}$, déjà utilisée par de nombreux auteurs [Emmerich et al, 2002] [Emmerich and. Naujoks, 2004] [Emmerich et al, 2006], à la place de \tilde{f} lors du calcul des critères de dominance et de distance d'encombrement. $f_{\text{mérite}}$ représente en effet la meilleure valeur que f puisse atteindre compte tenu de l'erreur d'approximation locale. Les valeurs minimales de $f_{\text{mérite}}$ correspondent à des valeurs faibles de \tilde{f} ou à des valeurs fortes de $\Delta\tilde{f}$, c'est-à-dire à des zones inexplorées de l'espace paramétrique. C-Évolutif-FM permet ainsi d'associer une meilleure exploration de l'espace paramétrique et une exploitation des zones où le front optimal de Pareto semble être localisé.

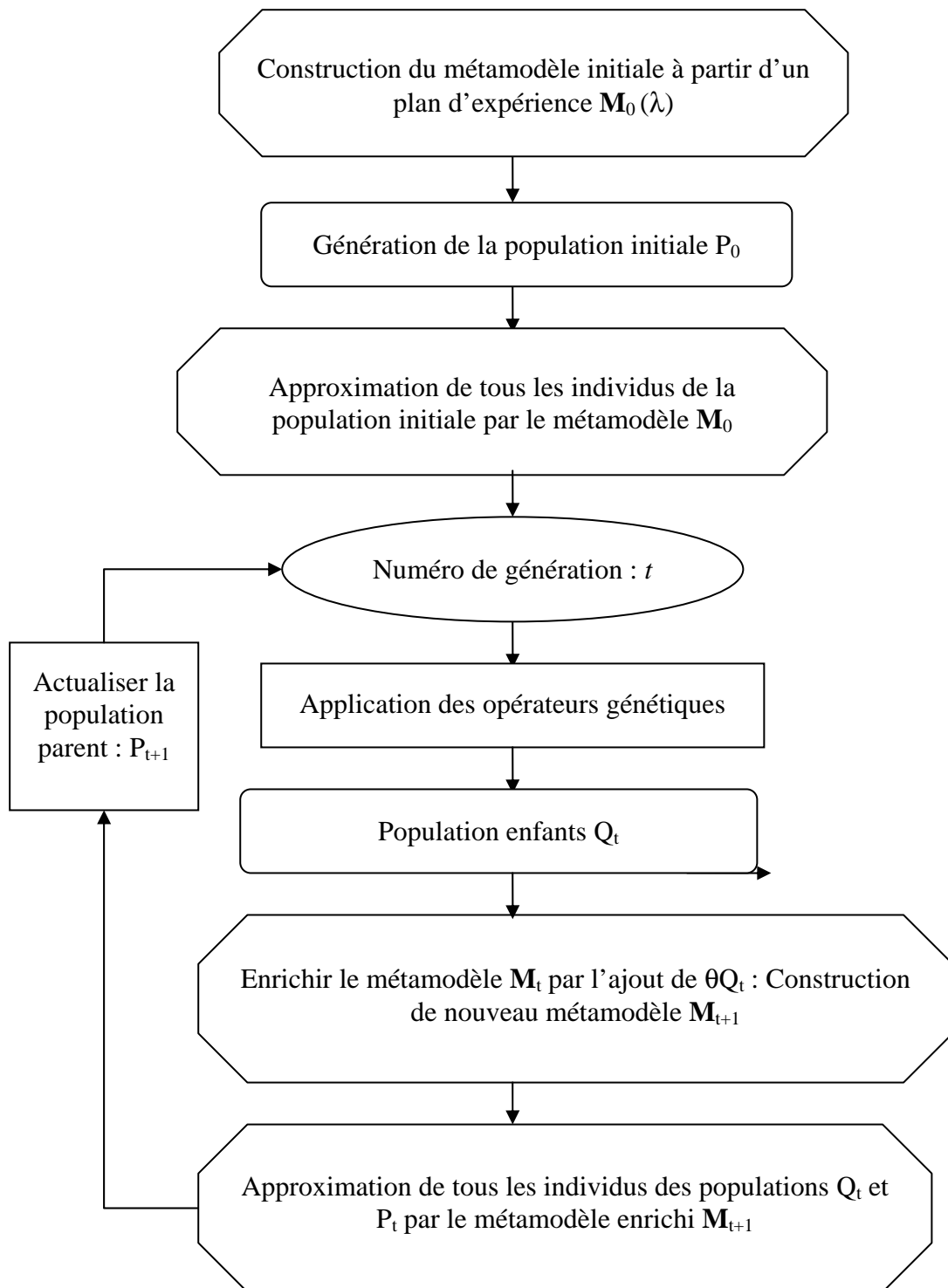


Figure 35- Organigramme de fonctionnement du C-Actualisé.

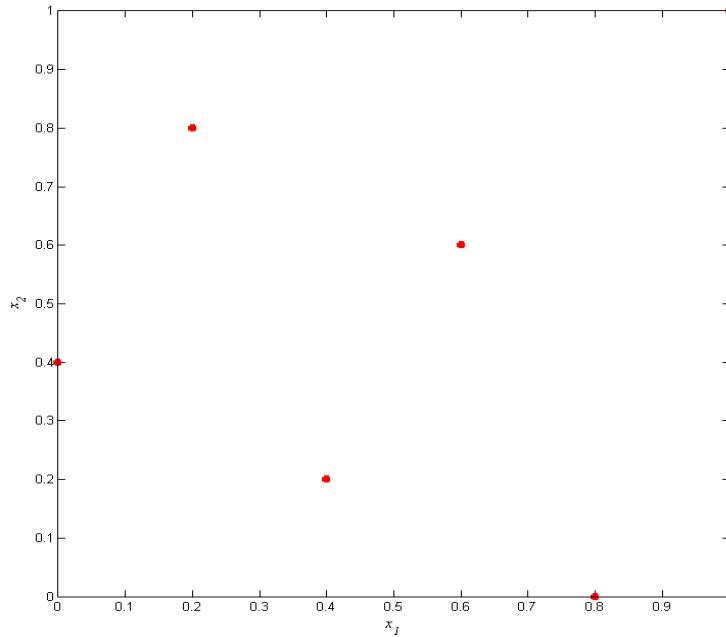


Figure 36– Plan d’expérience latin hypercube pour $n=2$

III.3 PROBLEMES TESTS ANALYTIQUES

Dans cette partie, on s’intéresse à la comparaison de quatre types d’algorithmes : C-Constant, C-Actualisé et deux versions de C-Évolutif-FM. Ces deux versions diffèrent par le choix de la fonction mérite utilisée. Pour C-Évolutif-H1, la fonction mérite utilise l’erreur H1 (Eq (II.26)) et pour C-Évolutif-MC, elle utilise l’erreur MC (Eq (II.25)).

III.3.1 Problèmes tests d’optimisation mono-objectif

Dans ce paragraphe, on compare les résultats d’optimisation des problèmes mono-objectif de Camel-Back, Griewank, Rastrigin et Rosenbrock (Figure 37) obtenus avec les quatre algorithmes, C-Constant, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC avec ceux obtenus par l’algorithme standard NSGA-II, SE-META et les solutions analytiques.

On utilise une population de taille $N = 50 \times n$, un nombre total de générations $n_g = 100$, un

nombre initial de points maîtres $\lambda = \frac{(n+1) \times (n+2)}{2}$ et un nombre maximal de points maîtres

$\lambda_{\max} = 50 \times n$.

III.3.1.1 Fonction de Camel-Back

Le problème de minimisation de la fonction Camel-Back est un problème utilisant deux variables. Il a deux optima de même valeur -1.0316 situé en $(-0.0898, 0.7126)$ et en $(0.0898, -0.7126)$. Le minimum est situé dans une zone de très faible gradient :

$$\left\{ \begin{array}{l} \text{minimiser } f(x_1, x_2) = \left(4 - 2, 1(x_1)^2 + \frac{(x_1)^4}{3} \right) (x_1)^2 + x_1 x_2 + \left(-4 + 4(x_2)^2 \right) (x_2)^2 \\ -5 \leq x_1, x_2 \leq 5 \end{array} \right. \quad (\text{III.2})$$

Algorithmes	x_1	x_2	f_{\min}
Solutions analytiques	-0.0898	0.7126	-1.0316
	0.0898	-0.7126	
NSGA-II	0.0877	-0.7134	-1.0316
C-Constant	-0.0137	0.8145	-0.9036
C-Actualisé	0.0034	-0.7383	-0.9943
C-Évolutif-H1	-0.0312	0.7046	-1.0180
C-Évolutif-MC	0.0974	-0.6963	-1.0291
SE-META	0.0075	-0.7322	-1.0001

Tableau 4- Comparaison des résultats du problème Camel-Back obtenus avec les différents algorithmes

III.3.1.2 Fonction de Griewank

La minimisation de la fonction de Griewank est un problème ayant deux variables :

$$\left\{ \begin{array}{l} \text{minimiser } f(x_1, x_2) = \frac{(x_1)^2}{50} + \frac{(x_2)^2}{50} - \left(\cos(x_1) + 1 \right) \left(\cos\left(\frac{x_2}{\sqrt{2}}\right) + 1 \right) \\ -15 \leq x_1, x_2 \leq 15 \end{array} \right. \quad (\text{III.3})$$

Ce problème possède des minima locaux et un minimum global en (0, 0) de valeur -4.

Algorithmes	x_1	x_2	f_{\min}
Solution analytique	0	0	-4
NSGA-II	-0.0169	-0.0040	-3.9997
C-Constant	-5.7539	1.0739	-2.5293
C-Actualisé	0.0966	0.0171	-3.9903
C-Évolutif-H1	-6.0822	0.0504	-3.2186
C-Évolutif-MC	0.0047	0.0993	-3.9949
SE-META	-0.1305	-1.3089	-3.1544

Tableau 5- Comparaison des résultats du problème Griewank obtenus avec les différents algorithmes

III.3.1.3 Fonction de Rastrigin

La minimisation de la fonction de Rastrigin (Eq (III.4)) est un problème à n variables. Il a plusieurs minima locaux susceptibles de ralentir la convergence des algorithmes. Son minimum global est en $x_i = 0.5, i = 1, \dots, n$ de valeur $-100n$ (pour $n = 2$ voir la Figure 37).

$$\left\{ \begin{array}{l} \text{minimiser } f(x_1, \dots, x_n) = \sum_{i=1}^n (50(x_i - 0.5))^2 - 100 \cos(25(x_i - 0.5)) \\ 0 \leq x_i \leq 1, i=1, \dots, n \end{array} \right. \quad \text{(III.4)}$$

Algorithmes	x_1	x_2	f_{\min}
Solution analytique	0.5	0.5	-200
NSGA-II	0.5005	0.5000	-199.9913
C-Constant	0.4986	0.5814	-38.4740
C-Actualisé	0.5006	0.5000	-199.9890
C-Évolutif-H1	0.4999	0.5003	-199.9960
C-Évolutif-MC	0.4999	0.4976	-199.8080
SE-META	0.4964	0.4946	-198.5750

Tableau 6- Comparaison des résultats du problème Rastrigin ($n=2$) obtenus avec les différents algorithmes

III.3.1.4 Fonction de Rosenbrock

La minimisation de la fonction de Rosenbrock est un problème ayant 2 variables :

$$\left\{ \begin{array}{l} \text{minimiser } f(x_1, x_2) = 100((x_1)^2 - x_2)^2 + (1 - x_1)^2 \\ -2 \leq x_1, x_2 \leq 2 \end{array} \right. \quad \text{(III.5)}$$

Ce problème n'a qu'un seul optimum situé en $(1, 1)$ de valeur 0. La particularité de la fonction de Rosenbrock est que le minimum est situé dans une zone de très faible gradient.

Algorithmes	x_1	x_2	f_{\min}
Solution analytique	1	1	0
NSGA-II	0.9827	0.9656	0.0003
C-Constant	0.9679	0.9527	0.0264
C-Actualisé	1.1098	1.2299	0.0123
C-Évolutif-H1	1.1645	1.3486	0.0326
C-Évolutif-MC	1.0769	1.1557	0.0076
SE-META	0.3256	0.0700	0.5843

Tableau 7- Comparaison des résultats du problème Rosenbrock obtenus avec les différents algorithmes

Les quatre tableaux de comparaison de résultats obtenus par les différents algorithmes (Tableau 4, Tableau 5, Tableau 6 et Tableau 7) montrent que l’algorithme standard NSGA-II utilisant 10100 calculs exacts donne les meilleurs résultats pour les quatre fonctions. Pour les cinq algorithmes à base de métamodèle, C-Constant, C-Actualisé, C-Évolutif-H1, C-Évolutif-MC et SE-META, qui utilisent seulement 100 calculs exacts, on constate que C-Évolutif-MC donne les meilleurs résultats, qui sont comparables à ceux obtenus par NSGA-II. Les mauvais résultats sont obtenus par C-Constant. Les deux approches, C-Actualisé et C-Évolutif-H1, sont plus performant que SE-META.

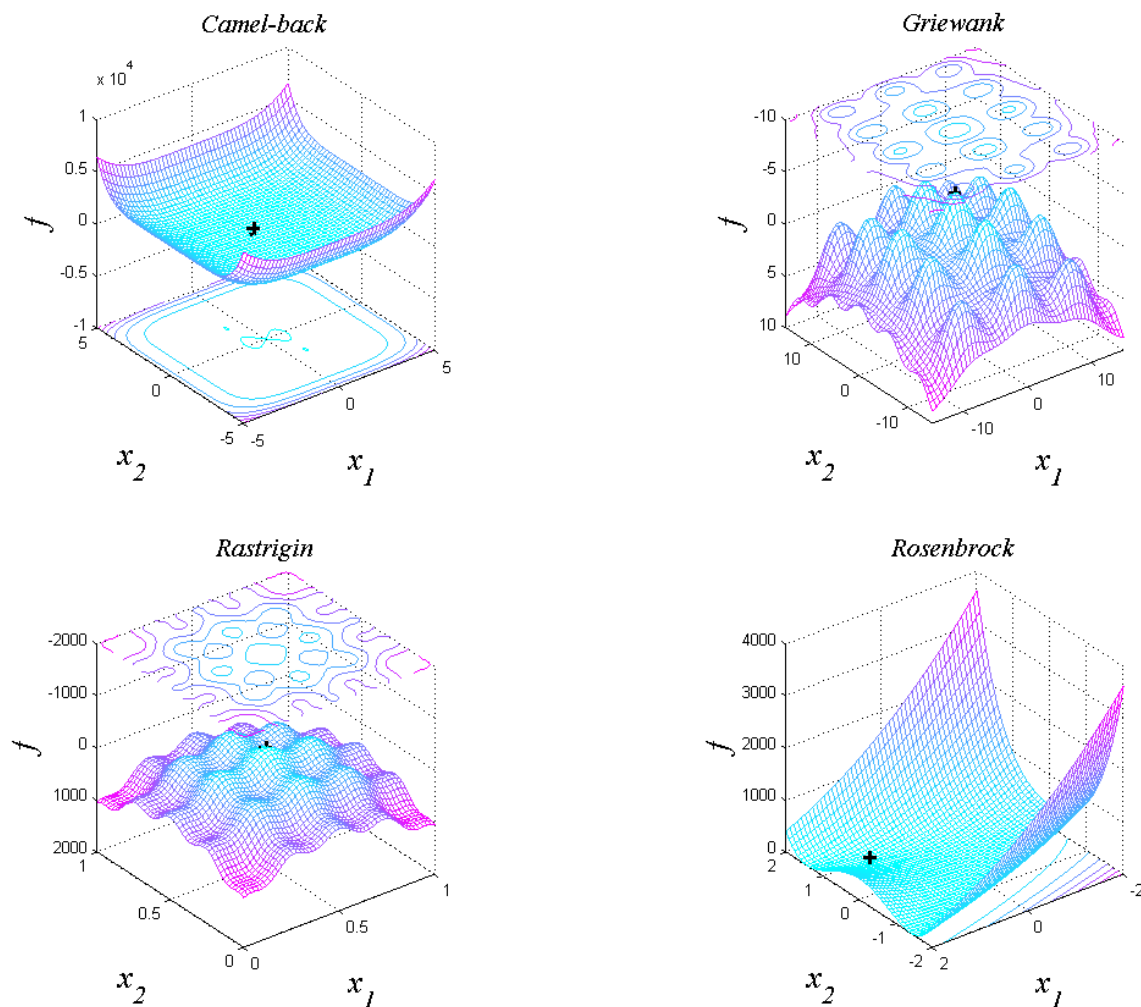


Figure 37 : Représentation des fonctions de Camel-Back, Griewank, Rastrigin et Rosenbrock (les croix indiquent les minima).

III.3.2 Problèmes tests d’optimisation multi-objectifs

Les problèmes tests d’optimisation multi-objectifs sont fréquemment utilisés pour tester, comparer et évaluer la performance des nouveaux algorithmes évolutionnaires multi-objectifs (AEMOs). Ils ont pour propriétés d’être simples à utiliser et de permettre le calcul analytique du front optimal de Pareto [Deb, 1999] [Deb et al, 2002b] [Deb et al, 2005]. Un problème test devrait avoir certaines caractéristiques, comme la simplicité de construction, la connaissance

de la forme exacte de son front optimal de Pareto et une meilleure dispersion des solutions sur ce front.

Pour tester un algorithme évolutionnaire multi-objectifs, Deb et al [Deb et al, 2002b] [Deb et al, 2005] proposent qu'un problème test doive satisfaire les critères suivants :

1. Il devrait être facile à construire.
2. Il devrait être extensible à n'importe quel nombre de variables de décision.
3. Il devrait être extensible à n'importe quel nombre de fonctions coût.
4. La facilité de la compréhension, la connaissance de la forme et de la localisation de son front optimal de Pareto.
5. Il devrait introduire une fonction de contrôle pour converger vers le front optimal exact et aussi pour trouver un ensemble optimal de solutions suffisamment réparties.

Il y a différentes approches pour construire un problème test pour l'optimisation multi-objectif.

III.3.2.1 Construction d'un problème test

En se basant sur les principes ci dessus, il existe différentes méthodes pour construire un problème test pour l'optimisation multi-objectifs. Deb et al [Deb et al, 2002b] [Deb et al, 2005] ont discuté trois méthodes :

III.3.2.1.1 L'approche des fonctions mono-objectif multiple

Cette approche est la plus intuitive et elle été implicitement utilisée par les anciens chercheurs dans le domaine des AEMOs pour construire des problèmes tests. Dans cette approche, M différentes fonctions coût mono-objectif sont utilisées pour construire un problème test multi-objectifs. Pour simplifier la construction, dans plusieurs cas, des fonctions coût sont utilisées comme différentes translations d'une seule fonction coût. Par exemple, le problème de minimisation SCH1 utilise les deux fonctions coût mono-objectif suivantes :

$$f_1(x) = x^2, f_2(x) = (x-2)^2$$

III.3.2.1.2 L'approche ascendante

Pour cette approche, une fonction mathématique décrivant le front optimal de Pareto est formulée dans l'espace des objectifs. Ensuite, un espace de fonctions (espace fonctionnel) est construit à partir de ce front pour définir le problème test. Pour deux objectifs, cette construction a été brièvement suggérée par Deb [Deb, 1999] et a été étendu par lui-même à un nombre élevé d'objectifs [Deb, 2001]. Pour plus de détails le lecteur peut consulter [Deb et al, 2002b].

III.3.2.1.3 L'approche de surface avec contrainte

Contrairement à l'approche ascendante qui commence par un front optimal de Pareto prédéfini, l'approche de surface avec contrainte commence par un espace fonctionnel prédéfini. Par exemple, une géométrie simple, une boîte rectangulaire, peut être utilisée. Chaque valeur de la fonction coût est comprise entre une limite inférieure et une autre supérieure. Le problème qui en résulte est le suivant:

Minimiser $f_1(x)$
 Minimiser $f_2(x)$
 \vdots
 Minimiser $f_M(x)$
 Satisfaisant $f_i^{\min} \leq f_i(x) \leq f_i^{\max}, i = 1, 2, \dots, M$

Il est clair que l'ensemble optimal de Pareto du problème ci-dessus a une seule solution (la borne inférieure de chaque fonction coût $(f_1^{\min}, f_2^{\min}, \dots, f_M^{\min})^T$). Le problème est maintenant rendu plus intéressant par l'ajout d'un certain nombre de contraintes :

$$g_j(f_1, f_2, \dots, f_M) \geq 0, j = 1, 2, \dots, J$$

Ceci est fait pour éliminer des portions du rectangle original. Le reste du rectangle forme le front optimal de Pareto.

III.3.2.2 Critère de comparaison des algorithmes

Afin de déterminer l'algorithme le plus performant, qui sera utilisé pour la résolution des problèmes de mise en forme, nous comparons les résultats obtenus avec les différentes méthodes de couplage du métamodèle, C-Constant, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC, à NSGA-II standard (AEMO sans métamodèle) sur plusieurs problèmes analytiques, et présentons les résultats les plus caractéristiques.

Pour évaluer précisément l'efficacité des algorithmes d'optimisation multi-objectifs, nous utilisons des mesures quantitatives ou des mesures de performance [Deb, 2001] [Deb et al, 2000b]. Ces mesures sont proposées pour évaluer les deux objectifs : (i) la convergence vers le front optimal de Pareto et (ii) la préservation de la diversité de solutions sur ce front. Le premier index, la *moyenne de la métrique de convergence* $\bar{\gamma}$ (Eq (III.6)), mesure la convergence de solutions obtenues (ensemble Q^* de taille N) vers un ensemble connu P^* de solutions optimales de Pareto (SOP) consistant en H (ici égal à 1000) solutions uniformément réparties.

$$\bar{\gamma} = \frac{\left(\sum_{i=1}^N d_i^2 \right)^{1/2}}{N} \quad \text{(III.6)}$$

Où d_i est la distance Euclidienne (dans l'espace des fonctions coût) entre la solution obtenue i et celle de P^* la plus proche :

$$d_i = \min_{k=1}^H \sqrt{\sum_{m=1}^M (f_m^i - f_m^{*k})^2} \quad (\text{III.7})$$

Où f_m^{*k} est la valeur de la $m^{\text{ième}}$ fonction coût de la $k^{\text{ième}}$ solution de P^* .

La seconde métrique, la *moyenne de la métrique de diversité* $\bar{\Delta}$, mesure la dispersion des solutions obtenues sur le front. Elle est donnée par :

$$\bar{\Delta} = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + (N-1)\bar{d}} \quad (\text{III.8})$$

Où le paramètre d_i est la distance Euclidienne entre deux solutions voisines de l'ensemble Q^* , et \bar{d} est la moyenne de ces distances. d_m^e est la distance entre les solutions extrêmes de P^* et Q^* pour la $m^{\text{ième}}$ fonction coût.

III.3.2.3 Problèmes tests multi-objectifs

Dans le reste de ce chapitre, nous utilisons une série des fonctions tests multi-objectifs qui diffèrent par leurs fronts de Pareto. Parmi les Problèmes traités, on distingue ceux sans contraintes, avec contraintes, ayant des fronts convexes, concaves et discontinus.

Au début, les quatre approches, C-Constant, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC sont comparées sur des problèmes d'optimisation multi-objectifs simples tels que Min-Ex, Max-Ex, SCH1, SCH2 et Const-Min-Ex. Ensuite, le meilleur algorithme est comparé à NSGA-II sur des problèmes d'optimisation multi-objectifs plus difficiles, comme les problèmes POL, VNT, TNK et KUR.

Pour la résolution des problèmes Min-Ex, Max-Ex, SCH1, SCH2 et Const-Min-Ex, l'algorithme NSGA-II utilise une population de taille $N = 50 \times n$ et un nombre total de générations $n_g = 50$. Pour les autres algorithmes le paramétrage est le suivant :

- Taille de population: $N = 50 \times n$
- Nombre total de générations: $n_g = 100$
- Nombre initial de points maîtres: $\lambda = \frac{(n+1)(n+2)}{2}$
- Nombre total de points maîtres: $\lambda_{\max} = 20 \times n$

III.3.2.3.1 Min-Ex

Le problème de minimisation Min-Ex (Eq (III.9)) porte sur deux fonctions coût et deux variables. Il est proposé par Deb [Deb, 2001], et a comme solution un front optimal de Pareto convexe.

$$\text{Min - Ex:} \begin{cases} \text{Minimiser} & f_1(x_1, x_2) = x_1 \\ \text{Minimiser} & f_2(x_1, x_2) = \frac{1+x_2}{x_1} \\ & 0.1 \leq x_1 \leq 1 \\ & 0 \leq x_2 \leq 5 \end{cases} \quad (\text{III.9})$$

Bien que ce problème ait l'air simple, les fonctions coût sont conflictuelles ce qui conduit à un ensemble de solutions optimales. Le problème (Eq (III.9)) revient à minimiser $\frac{1}{x_1}$ puisque la minimisation de $f_2(x_1, x_2)$ impose $x_2 = 0$. Le front analytique de Pareto s'écrit donc :

$$f_2(x_1, x_2) = \frac{1}{f_1(x_1, x_2)} \quad (\text{III.10})$$

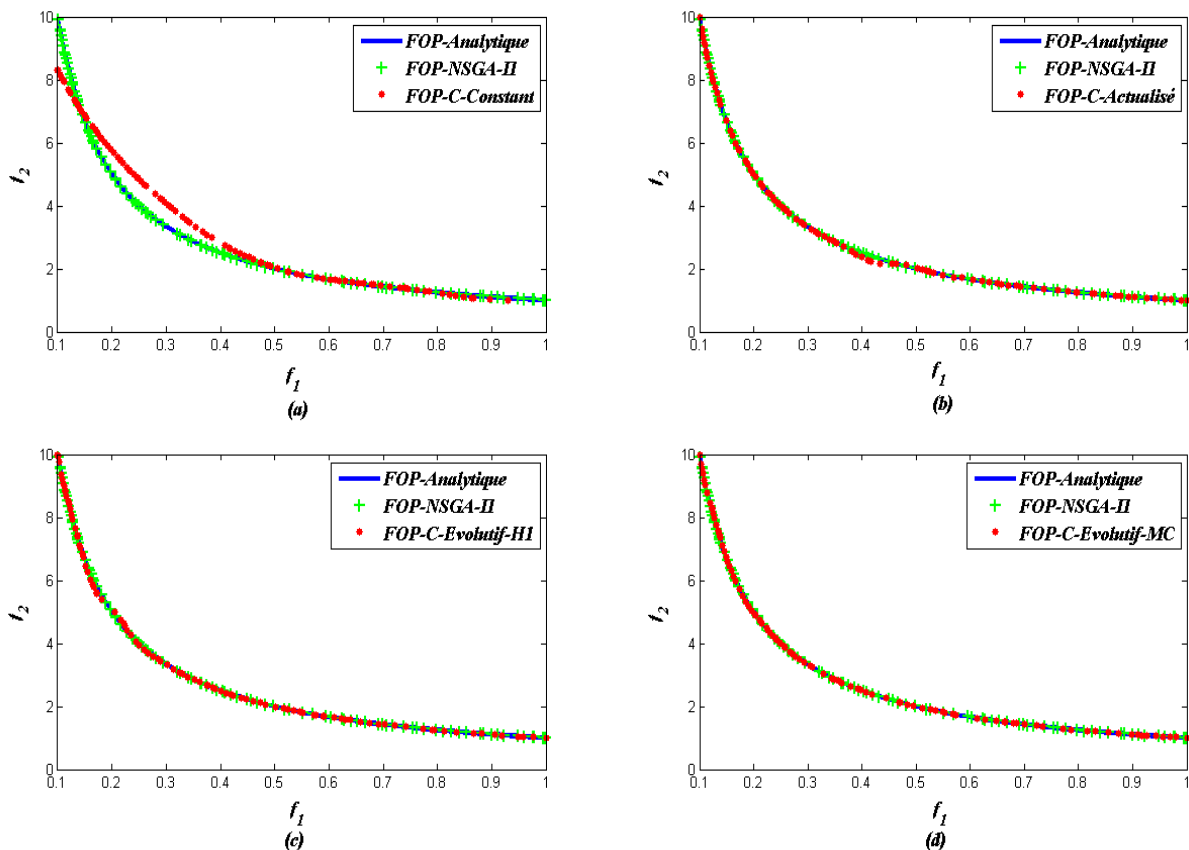


Figure 38 – Comparaison entre le FOP du POMO Min-Ex obtenu avec NSGA-II et ceux obtenus avec (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC.

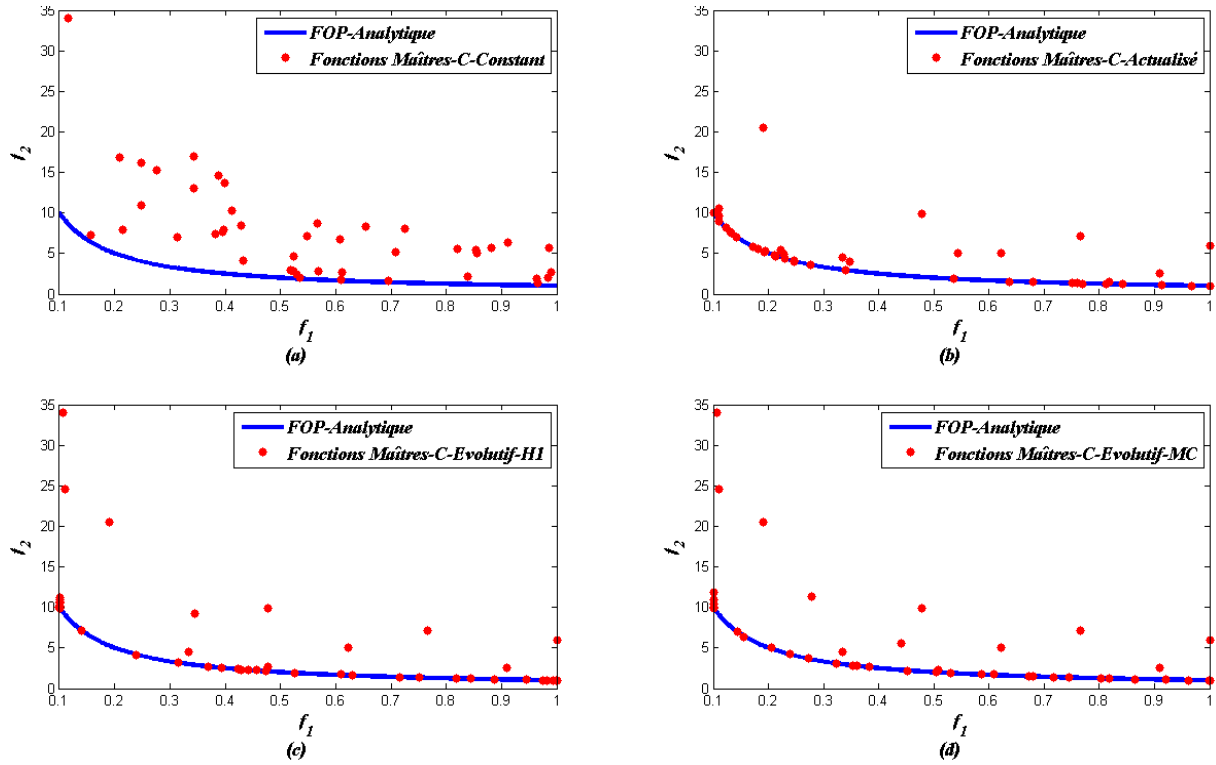


Figure 39 – Comparaison de la convergence de (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-HI et (d) C-Évolutif-MC vers le FOP analytique du POMO Min-Ex.

III.3.2.3.2 Max-Ex

Le problème Max-Ex est un problème de maximisation à deux fonctions coût et deux variables. Il est proposé par Deb [Deb, 2001]. Il a comme solution un front optimal de Pareto non convexe. Il est donnée par :

$$\text{Max-Ex:} \begin{cases} \text{Maximiser} & f_1(x_1, x_2) = 1.1 - x_1 \\ \text{Maximiser} & f_2(x_1, x_2) = 60 - \left(\frac{1 + x_2}{x_1} \right) \\ & 0.1 \leq x_1 \leq 1 \\ & 0 \leq x_2 \leq 5 \end{cases} \quad (\text{III.11})$$

Ce problème d'optimisation multi-objectifs a un ensemble optimal de Pareto en $x_2 = 0$ et $0.1 \leq x_1 \leq 1$ comme celui du problème Min-Ex bien que les fonctions coût soient différentes. Ici on a comme front optimal analytique:

$$f_2(x_1, x_2) = 60 - \frac{1}{1.1 - f_1(x_1, x_2)} \quad (\text{III.12})$$

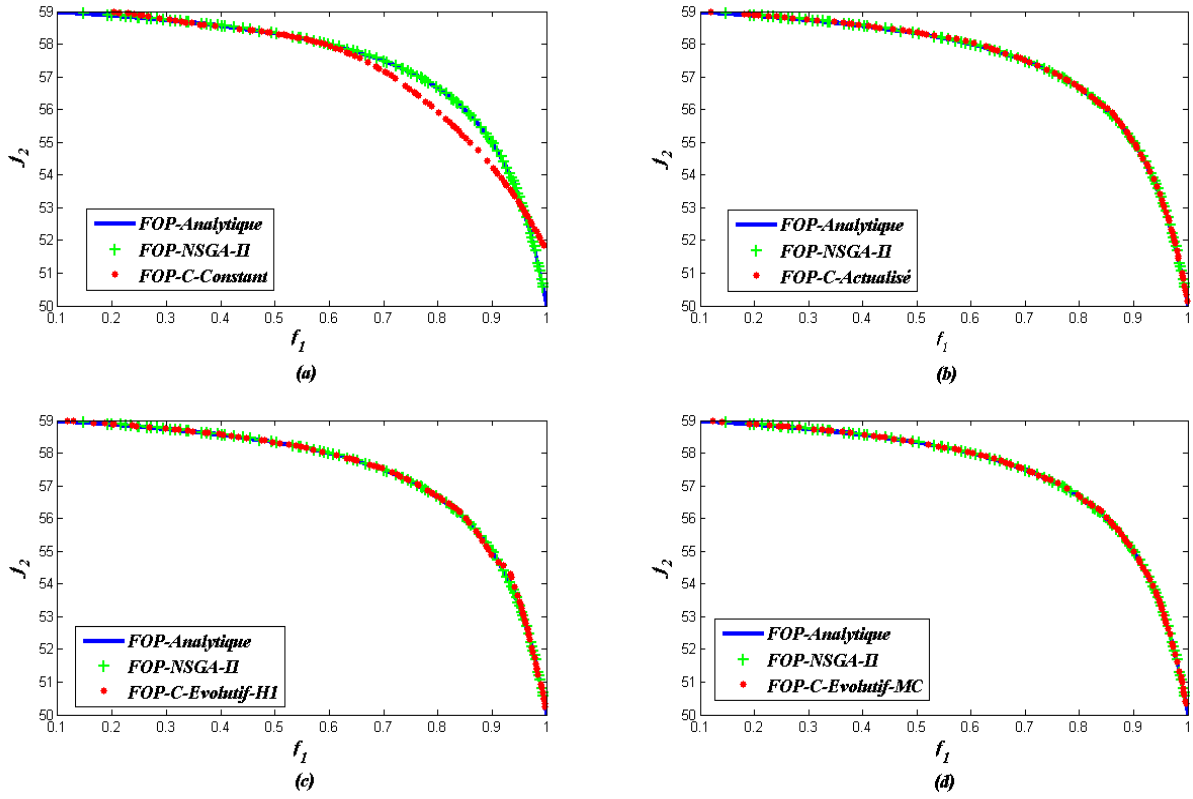


Figure 40– Comparaison entre le FOP du POMO Max-Ex obtenu avec NSGA-II et ceux obtenus avec (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC.

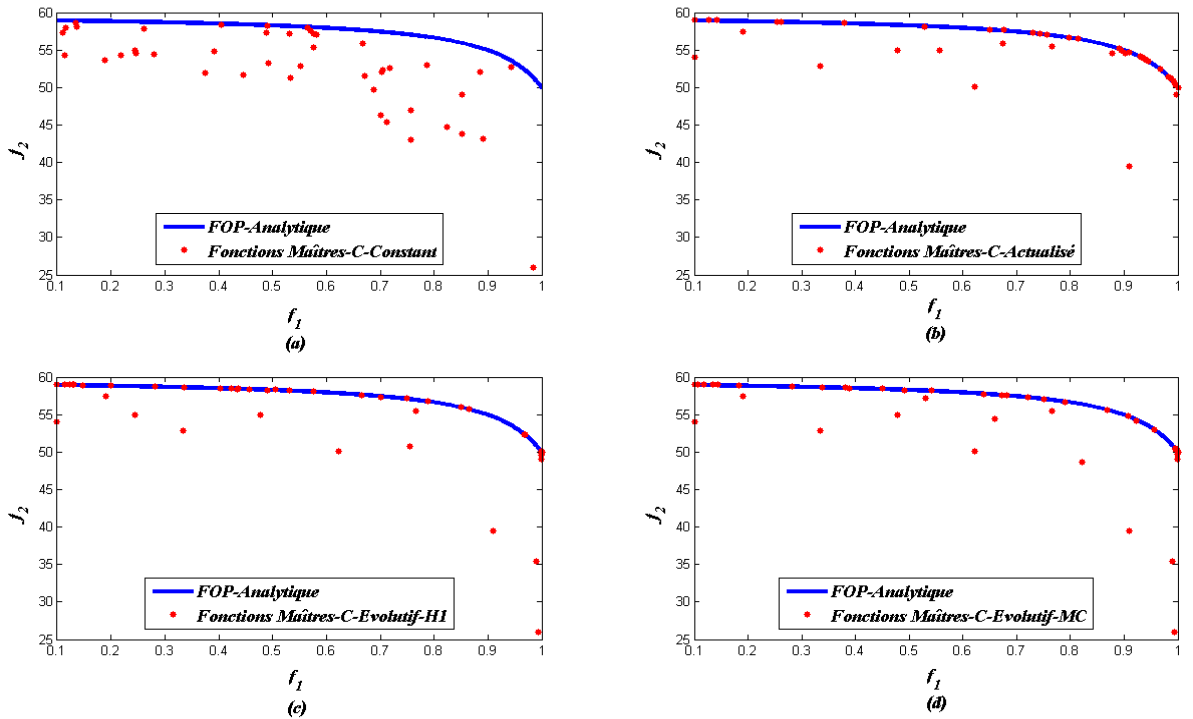


Figure 41 – Comparaison de la convergence de (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC vers le FOP analytique du POMO Max-Ex.

III.3.2.3.3 SCH1

La fonction SCH1 est un problème de minimisation ayant deux fonctions coût et une seule variable de décision. Elle est proposée par Schaffer et utilisée par Deb [Deb, 2001]. Malgré sa simplicité, elle est parmi les fonctions tests les plus utilisées. Elle est donnée par :

$$\text{SCH1:} \begin{cases} \text{Minimiser } f_1(x) = x^2 \\ \text{Minimiser } f_2(x) = (x-2)^2 \\ -5 \leq x \leq 5 \end{cases} \quad (\text{III.13})$$

Ce problème a un ensemble optimal de Pareto (SOP) en $0 \leq x \leq 2$ et ayant un front optimal de Pareto convexe :

$$f_2(x) = (\sqrt{f_1} - 2)^2 \quad (\text{III.14})$$

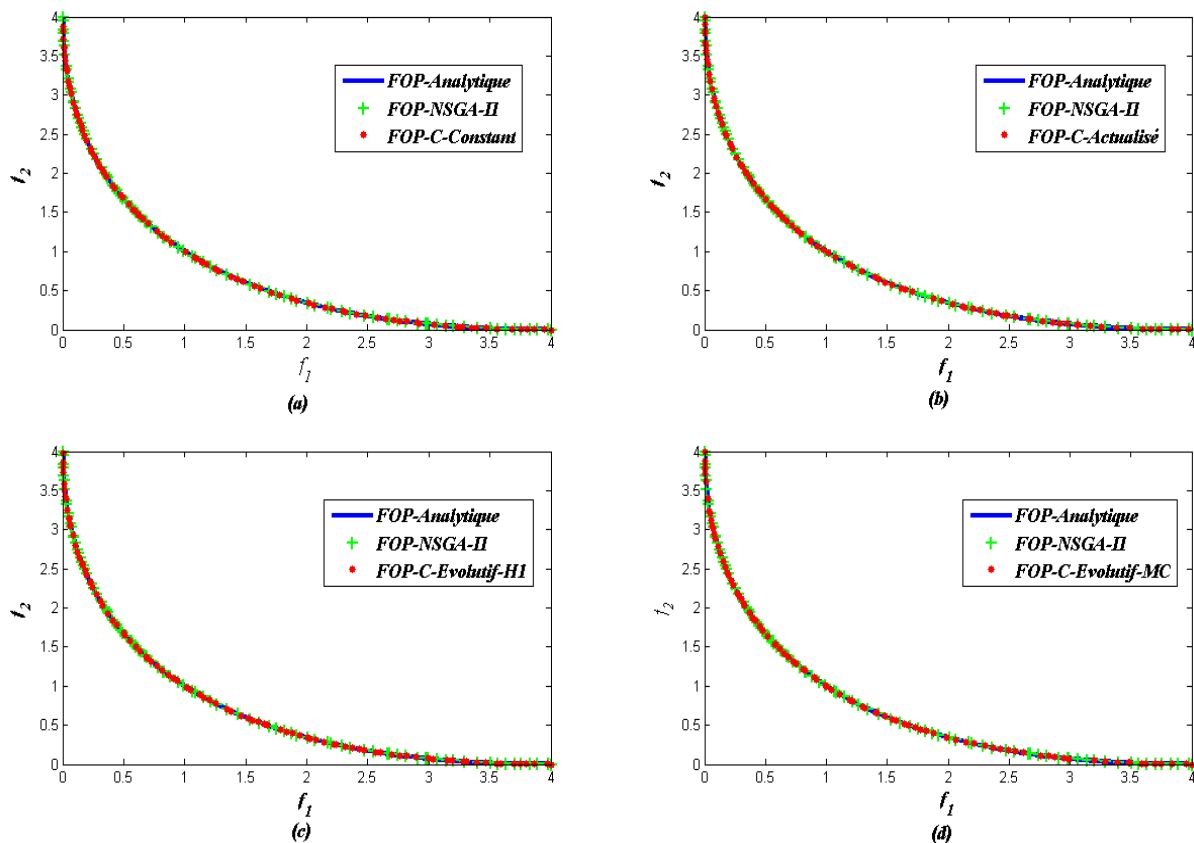


Figure 42–Comparaison entre le FOP du POMO SCH1 obtenu avec NSGA-II et ceux obtenus avec (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC.

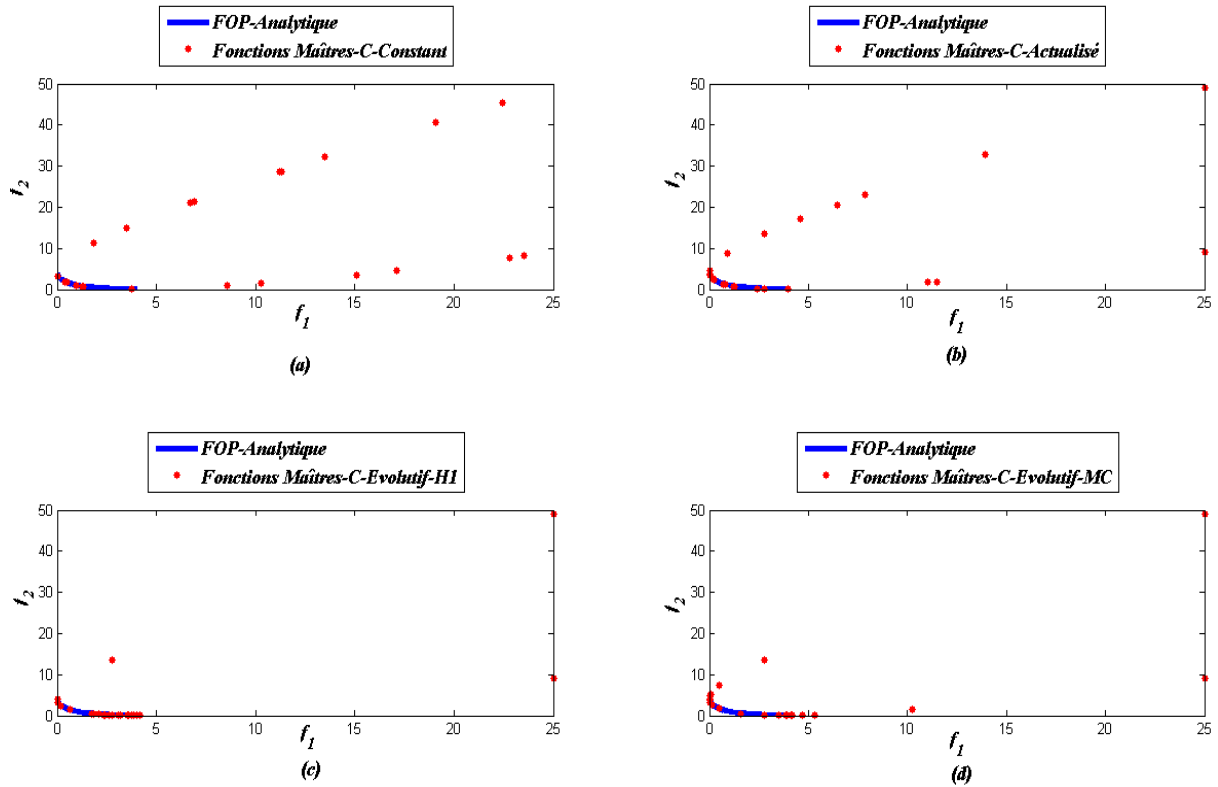


Figure 43 – Comparaison de la convergence de (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC vers le FOP analytique du POMO SCH1.

III.3.2.3.4 SCH2

Ce problème est aussi proposé par Schaffer dont lequel il minimise deux fonctions coût avec une seule variable. Il est aussi largement utilisé [Deb, 2001] pour tester les performances et la robustesse des algorithmes évolutionnaires multi-objectifs. Il est donné par :

$$\text{SCH2:} \left\{ \begin{array}{l} \text{Minimiser } f_1(x) = \begin{cases} -x & \text{if } x \leq 1 \\ x - 2 & \text{if } 1 < x \leq 3 \\ 4 - x & \text{if } 3 < x \leq 4 \\ x - 4 & \text{if } x > 4 \end{cases} \\ \text{Minimiser } f_2(x) = (x - 5)^2 \\ -5 \leq x \leq 10 \end{array} \right. \quad (\text{III.15})$$

L'ensemble optimal de Pareto est formé par deux espaces discontinus : $x \in \{[1,2] \cup [4,5]\}$ ce qui résulte en un front optimal de Pareto discontinu.

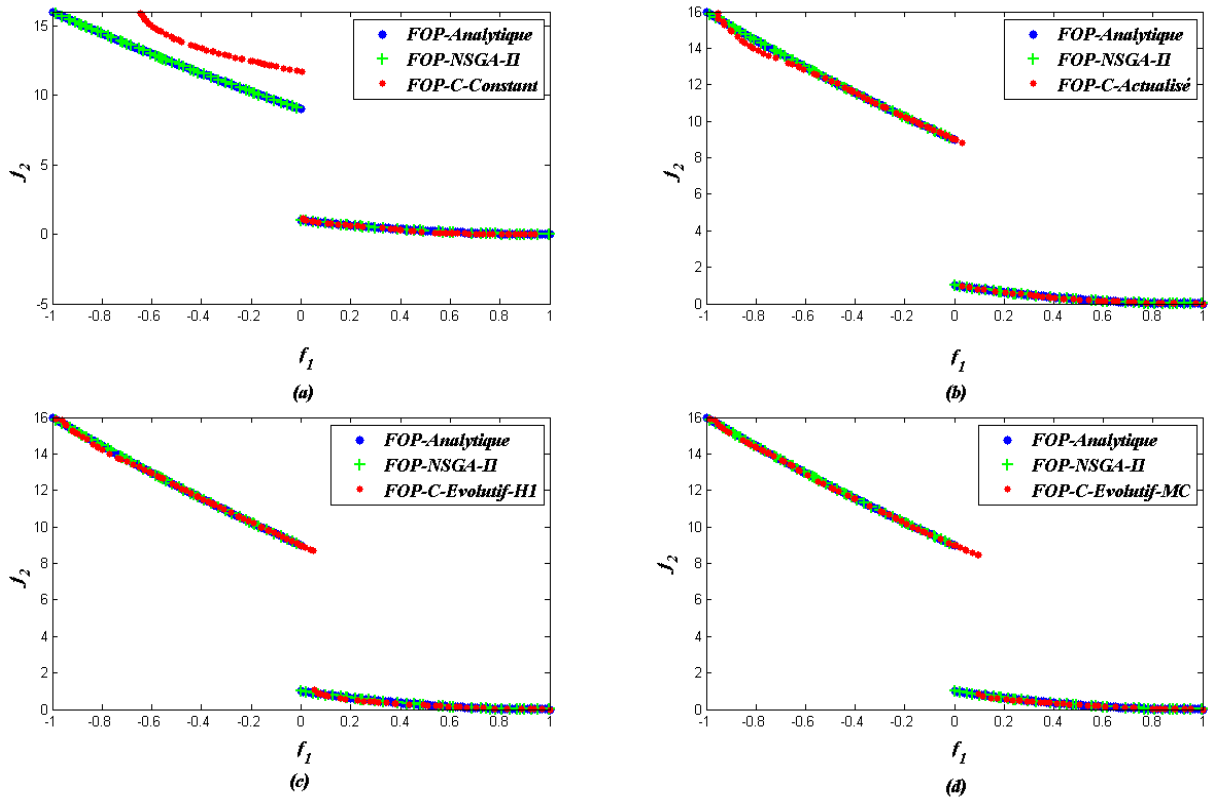


Figure 44 – Comparaison entre le FOP du POMO SCH2 obtenu avec NSGA-II et ceux obtenus avec (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-HI et (d) C-Évolutif-MC.

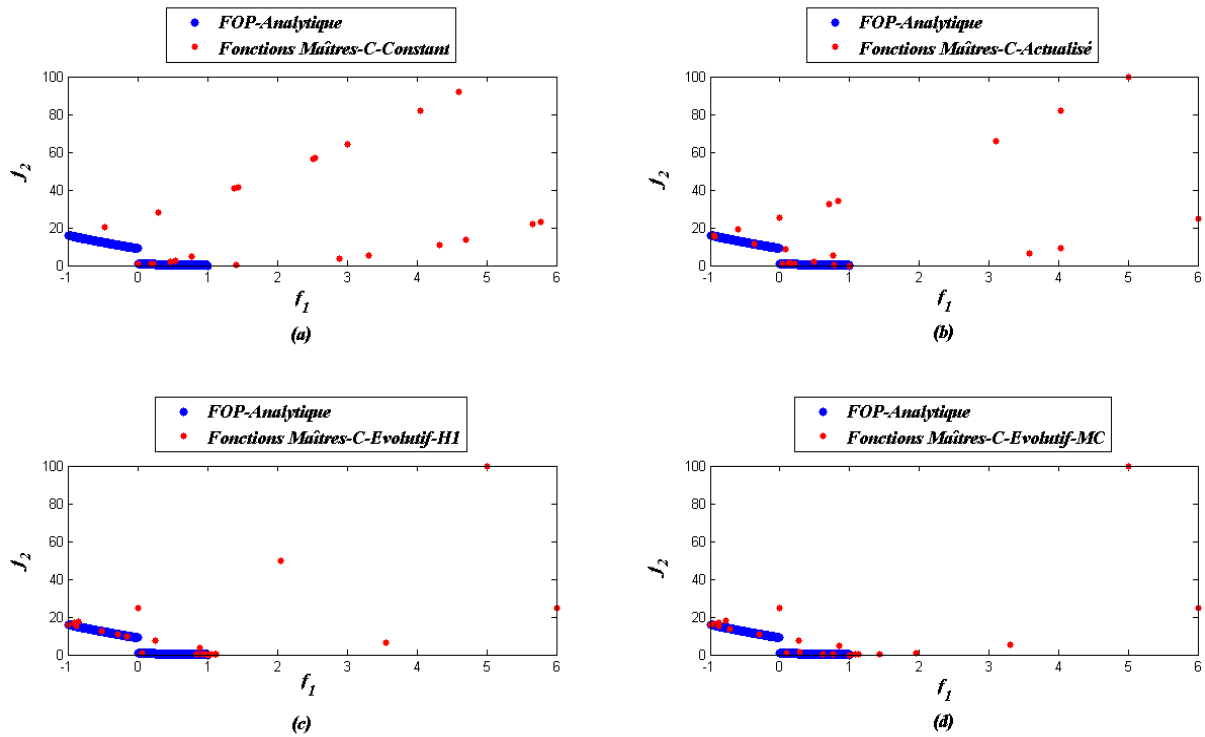


Figure 45 – Comparaison de la convergence de (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-HI et (d) C-Évolutif-MC vers le FOP analytique du POMO SCH2.

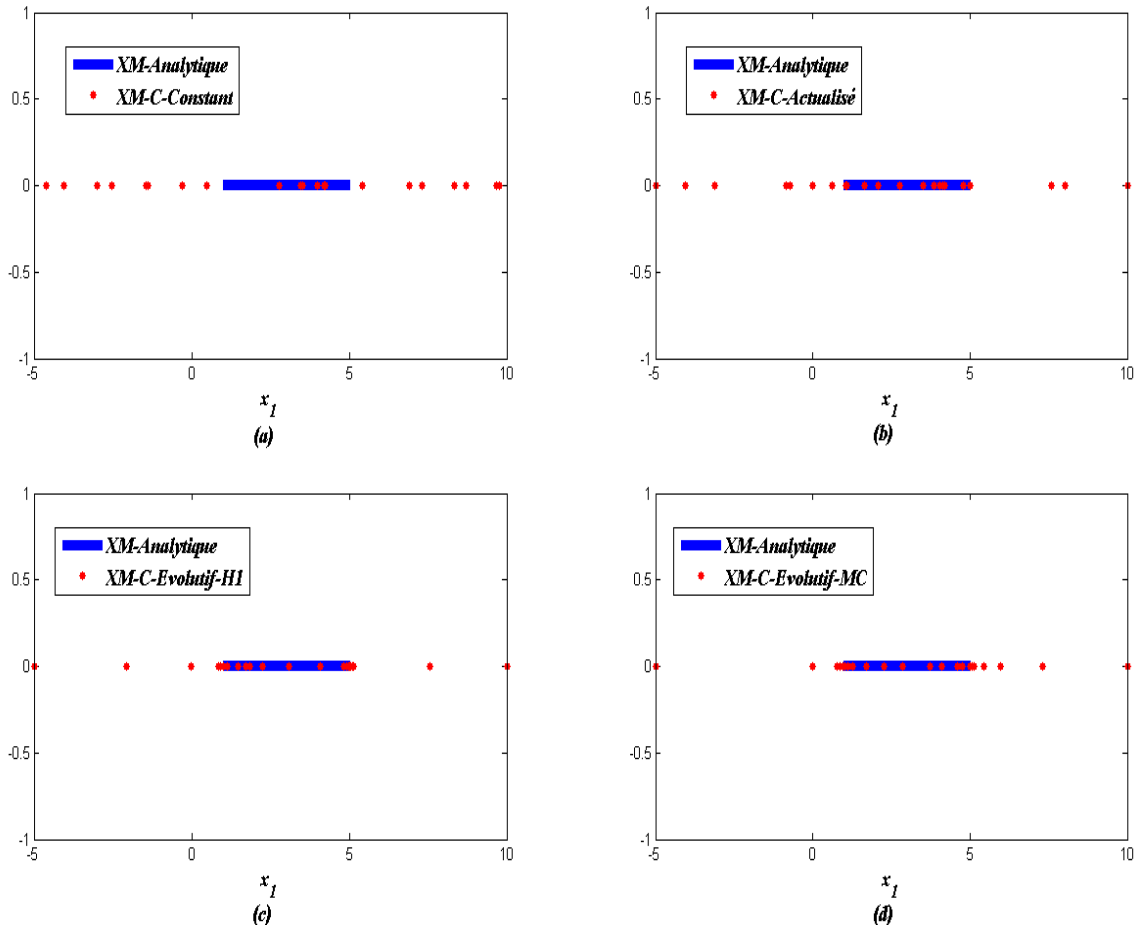


Figure 46 – Comparaison de la convergence de (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC vers le SOP analytique du POMO SCH2.

III.3.2.3.5 Const-Min-Ex

Le problème Const-Min-Ex est un problème de minimisation à deux fonctions coût et deux variables comme le problème Min-Ex mais avec deux contraintes. Il est proposé par Deb [Deb, 2001] :

$$\text{Const-Min-Ex : } \left\{ \begin{array}{l} \text{Minimiser } f_1(x_1, x_2) = x_1 \\ \text{Minimiser } f_2(x_1, x_2) = \frac{1+x_2}{x_1} \\ \text{satisfaisant } g_1(x_1, x_2) = x_2 + 9x_1 \geq 6 \\ \quad \quad \quad g_2(x_1, x_2) = -x_2 + 9x_1 \geq 1 \\ \quad \quad \quad 0.1 \leq x_1 \leq 1; 0 \leq x_2 \leq 5 \end{array} \right. \quad (\text{III.16})$$

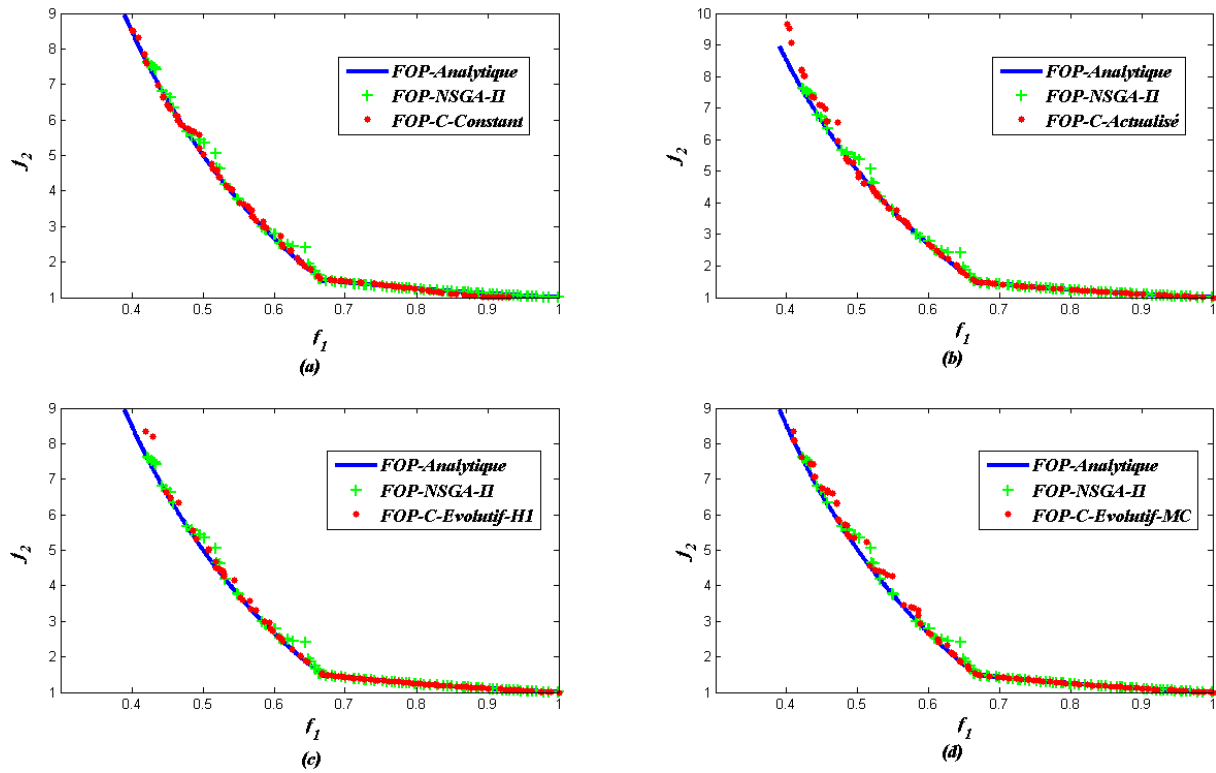


Figure 47 – Comparaison entre le FOP du POMO Const-Min-Ex obtenu avec NSGA-II et ceux obtenus avec (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC.

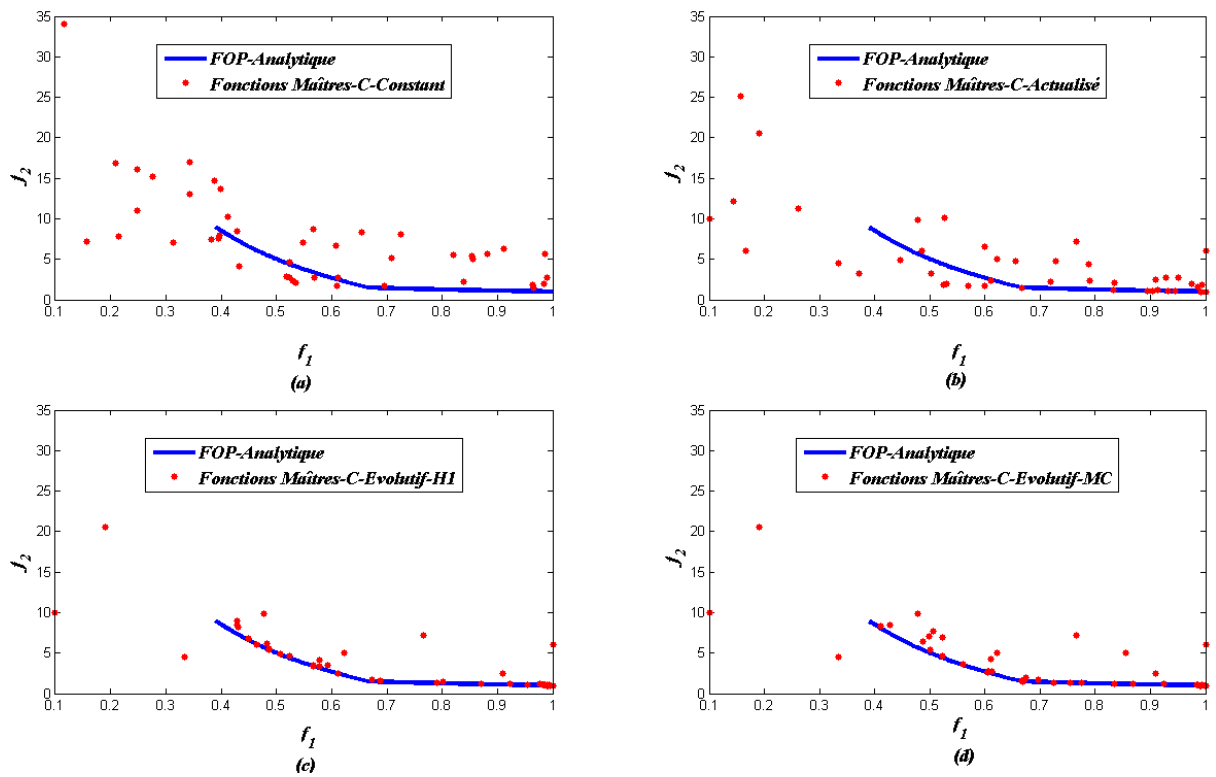


Figure 48 – Comparaison de la convergence de (a) C-Constant, (b) C-Actualisé, (c) C-Évolutif-H1 et (d) C-Évolutif-MC vers le FOP analytique du POMO Const-Min-Ex.

Les cinq figures de comparaison des fronts optimaux de Pareto de problèmes tests multi-objectifs précédents (Figure 38, Figure 40, Figure 42, Figure 44 et Figure 47) montrent que les différentes approches permettent toutes d'assez bien retrouver les fronts optimaux de Pareto. Certaines sont aussi précises que NSGA-II standard (superposé au front analytique), qui requiert pourtant 127 fois plus de calculs exacts (5100 dans ce cas) pour les problèmes Min-Ex, Max-Ex et Const-Min-Ex et 255 fois plus de calculs exacts dans le cas de SCH1 et SCH2. Nous y voyons aussi l'importance d'actualiser le métamodèle, car avec seulement 40 calculs exacts, C-Constant ne permet pas d'obtenir des approximations satisfaisantes de fronts optimaux de Pareto de problèmes Min-Ex et Max-Ex et celui de SCH2 (avec 20 calculs exacts). Cela vient du fait que les métamodèles initiaux (métamodèles fixes) de chacun de ces problèmes ne sont pas assez précis au voisinage des fronts optimaux de Pareto. Les figures de comparaison de la convergence des différents algorithmes vers les fronts optimaux de Pareto (Figure 39, Figure 41, Figure 43, Figure 45 et Figure 48) montrent la faiblesse du couplage C-Constant qui génère des fonctions maîtres (valeurs des fonctions coût aux points maîtres) loin des fronts optimaux de Pareto (Figure 39-(a), Figure 41-(a) et Figure 45-(a)). En revanche, pour les trois métamodèles évolutifs, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC, on obtient des fronts, pour tous les problèmes, tout aussi précis qu'avec NSGA-II standard. Cette précision s'explique par l'efficacité de chacune de ces approches de converger vers les fronts optimaux de Pareto avec une bonne distribution des points (Figure 39, Figure 41, Figure 43, Figure 45 et Figure 48). Pour le problème SCH2 qui a un ensemble optimal de Pareto (SOP) discontinu, on constate que contrairement aux autres couplages, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC, le couplage C-Constant est incapable de générer assez de points maîtres dans cet ensemble (Figure 46).

Pour évaluer avec précision l'efficacité des algorithmes, nous utilisons les mesures de performance (III.3.2.2) :

Algorithmes		Min-Ex	Max-Ex	Const-Min-Ex	SCH1	SCH2-I	SCH2-II
NSGA-II	$\bar{\gamma}$	0.0008865	0.00078302	0.0007918	0.0011385	0.00053077	0.00015386
	σ_{γ}^2	3.83E-08	2.62E-08	1.65E-08	4.48E-07	3.02E-10	1.02E-12
C-Constant	$\bar{\gamma}$	0.00328438	0.00318896	0.00218532	0.00158462	0.0445485	0.00682722
	σ_{γ}^2	1.14E-06	1.27E-06	9.31E-07	2.12E-06	0.0060604	1.04E-05
C-Actualisé	$\bar{\gamma}$	0.00110855	0.00103447	0.00939296	0.00128647	0.00553504	0.0017314
	σ_{γ}^2	6.21E-08	5.57E-08	0.00362447	8.22E-07	3.41E-05	3.47E-08
C-Évolutif-H1	$\bar{\gamma}$	0.00097135	0.00097338	0.00067722	0.00133595	0.00640919	0.00523542
	σ_{γ}^2	5.36E-08	5.26E-08	8.78E-09	1.42E-06	0.00018516	2.04E-06
C-Évolutif-MC	$\bar{\gamma}$	0.00093976	0.00084894	0.000689644	0.00124797	0.000661602	0.000263835
	σ_{γ}^2	5.56E-08	2.31E-08	4.76E-009	1.19E-006	7.16E-010	1.4E-011

Tableau 8- Les valeurs de la moyenne $\bar{\gamma}$ et la variance σ_{γ}^2 de la métrique de convergence obtenues par les différents algorithmes

Pour les problèmes tests présentés précédemment, le Tableau 8 présente la moyenne et la variance de la métrique de la convergence. Il montre que les valeurs obtenues avec les cinq algorithmes sont tout à fait comparables et proches de zéro. On constate que les améliorations

apportées par le C-Évolutif-MC sont plus claires. En effet cet algorithme fournit des valeurs plus petites que celles obtenues par C-Constant, C-Actualisé et C-Évolutif-H1 dans tous les cas à l'exception de la première partie (à gauche) du front de Pareto de SCH2. Ces valeurs sont très proches de celles fournies par NSGA-II.

Algorithmes		Min-Ex	Max-Ex	Const-Min-Ex	SCH1	SCH2-I	SCH2-II
NSGA-II	$\bar{\Delta}$	0.678869	0.672993	1.17954	0.423538	0.559692	0.459693
	σ_{Δ}^2	0.00603596	0.00487025	0.0203184	0.00119833	0.00574808	0.00050207
C-Constant	$\bar{\Delta}$	0.671883	0.695437	0.994816	0.431457	0.724963	0.694849
	σ_{Δ}^2	0.00334704	0.00346088	0.0116465	0.00125522	0.00507459	0.00088065
C-Actualisé	$\bar{\Delta}$	0.715425	0.600515	1.08151	0.463992	0.511989	0.538293
	σ_{Δ}^2	0.00801004	0.00512069	0.0235629	0.00147839	0.00579437	0.00062841
C-Évolutif-H1	$\bar{\Delta}$	0.695238	0.653625	1.25225	0.417752	0.500693	0.476571
	σ_{Δ}^2	0.00664457	0.00509286	0.0364761	0.00111923	0.00429516	0.00104834
C-Évolutif-MC	$\bar{\Delta}$	0.659351	0.599278	1.10627	0.478161	0.549424	0.417146
	σ_{Δ}^2	0.00543168	0.0042376	0.013814	0.00169633	0.00555779	0.00050834

Tableau 9- Les valeurs de la moyenne $\bar{\Delta}$ et la variance σ_{Δ}^2 de la métrique de diversité obtenues par les différents algorithmes

Le **Tableau 9** présente la moyenne et la variance de la métrique de la diversité. Il montre que les valeurs obtenues sont tout à fait comparables, et donc le M-AEMO fournit des résultats de même qualité que NSGA-II et avec un coût très réduit. En particulier, C-Évolutif-MC a une bonne diversité de points sur le front de Pareto, qui est meilleure que celle obtenue avec NSGA-II.

Dans le reste de ce chapitre, le meilleur algorithme, C-Évolutif-MC, sera utilisé pour la résolution des problèmes tests multi-objectifs plus difficiles tels que les problèmes POL, VNT, TNK et KUR.

Problèmes	N	n_g	λ	λ_{\max}
POL	100	100	6	100
VNT	100	100	6	100
TNK	100	250	6	200
KUR	150	250	10	418

Tableau 10- Paramétrage de C-Évolutif-MC et NSGA-II pour les quatre problèmes tests

III.3.2.3.6 POL

Poloni et al [Poloni et al, 2000] ont utilisé un problème ayant deux variables et deux fonctions à minimiser (Eq (III.17)) Ce problème a un ensemble optimal de Pareto non convexe et discontinu qui est difficile à trouver. Il a aussi un ensemble de solutions optimal discontinu dans l'espace des fonctions coût comme celui dans l'espace des variables.

$$\text{POL: } \left\{ \begin{array}{l} \text{Minimiser } f_1(X) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2 \\ \text{Minimiser } f_2(X) = (x_1 + 3)^2 + (x_2 + 1)^2 \\ \text{Avec } \quad A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2 \\ \quad \quad A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2 \\ \quad \quad B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2 \\ \quad \quad B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2 \\ \quad \quad -\pi \leq (x_1, x_2) \leq \pi \end{array} \right. \quad (\text{III.17})$$

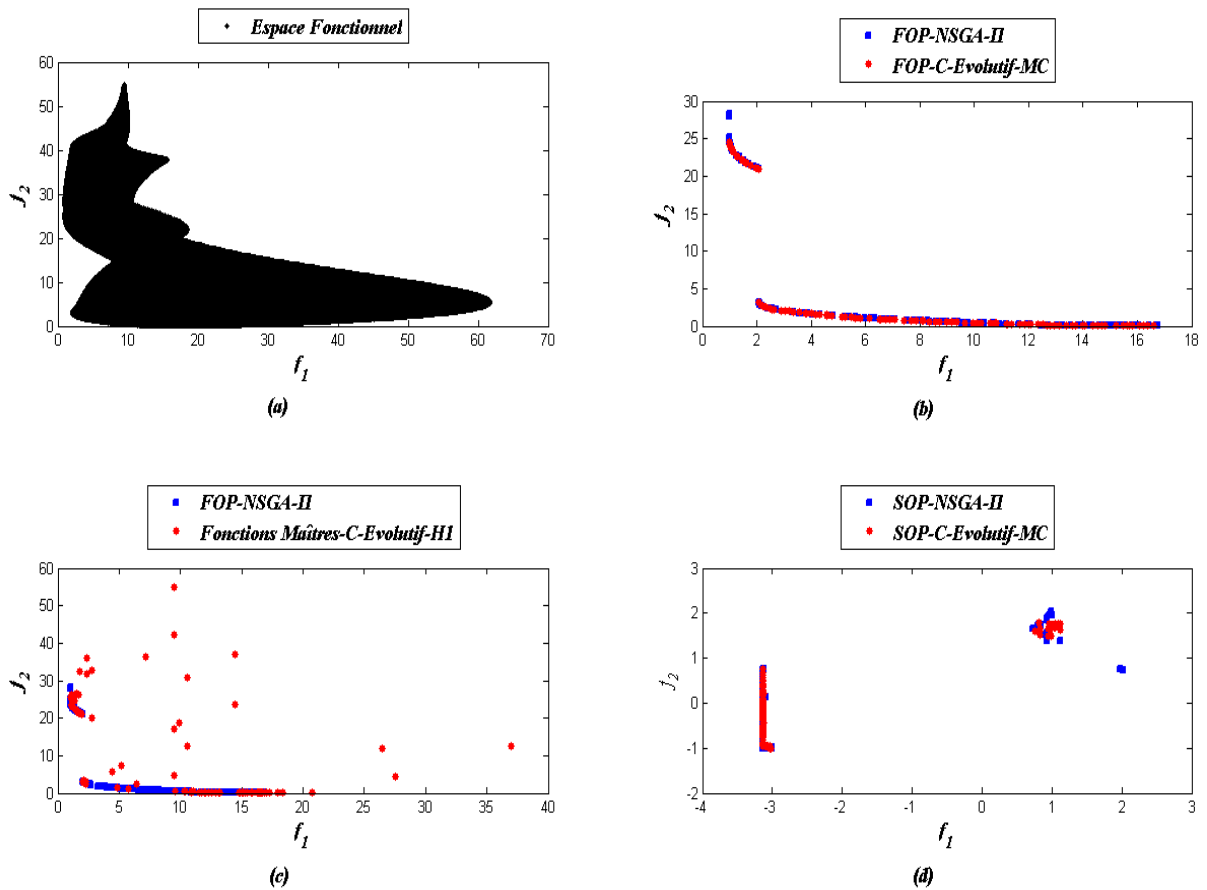


Figure 49– Représentation de (a) l'espace fonctionnel, (b) des FOP obtenus avec NSGA-II et C-Évolutif-MC, la convergence de C-Évolutif-MC vers (c) FOP et (d) SOP obtenus avec NSGA-II pour le POMO POL

III.3.2.3.7 VNT

Viennet [Viennet, 1996] a utilisé un problème d'optimisation multi-objectifs possédant trois fonctions coût à minimiser en présence de deux variables :

$$\text{VNT : } \begin{cases} \text{Minimiser } f_1(X) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \\ \text{Minimiser } f_2(X) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \\ \text{Minimiser } f_3(X) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp[-(x_1^2 + x_2^2)] \\ -3 \leq (x_1, x_2) \leq 3 \end{cases} \quad (\text{III.18})$$

Ce problème a un ensemble de solutions optimal discontinu dans l'espace fonctionnel comme celui dans l'espace de décision. Pour plus de détails le lecteur peut consulter [Deb, 2001].

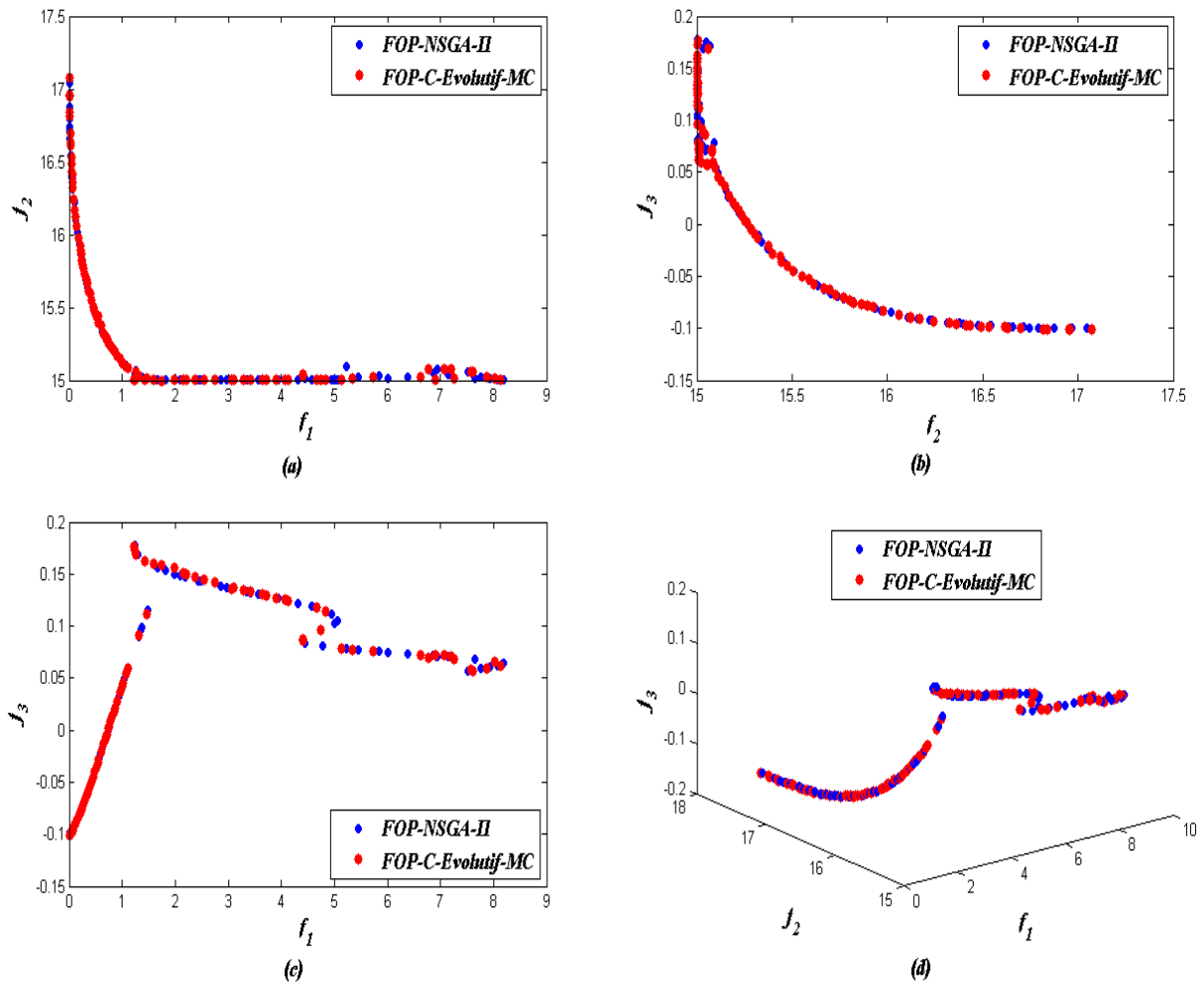


Figure 50 – Comparaison des FOP du POMO VNT obtenus avec NSGA-II et C-Évolutif-MC

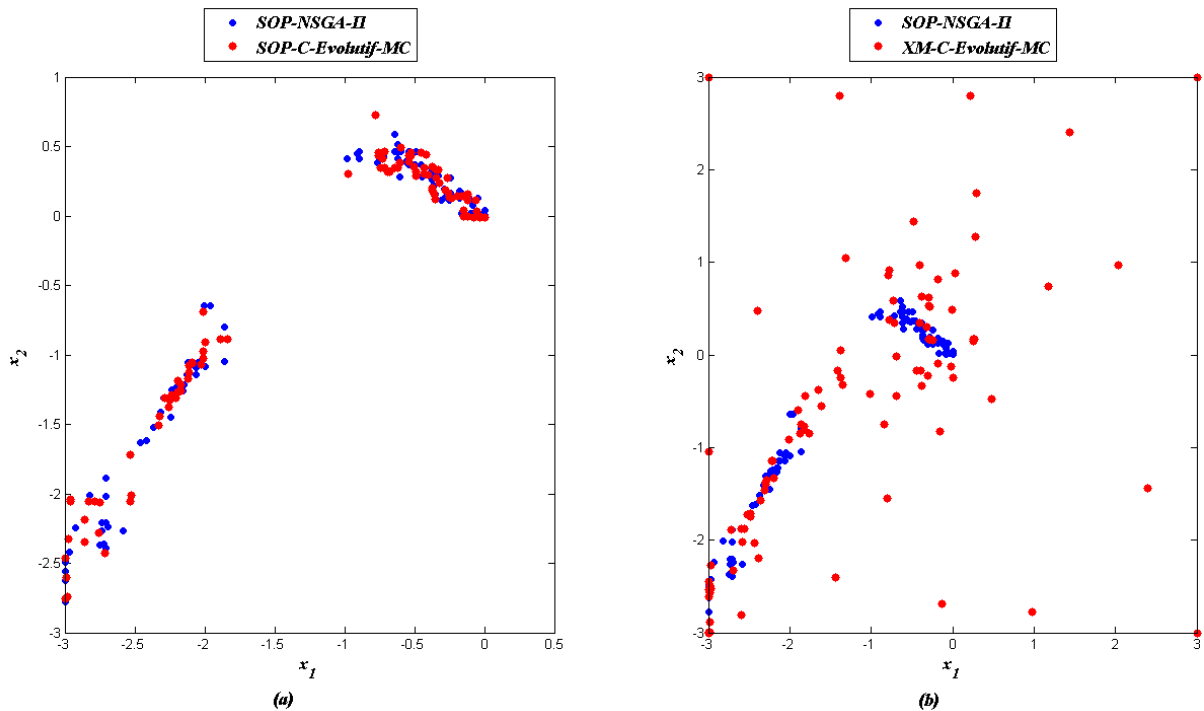


Figure 51 – (a) Comparaison des SOP obtenus avec C-Évolutif-MC et NSGA-II et (b) la répartition des points maîtres(XM) au voisinage du SOP obtenu avec NSGA-II pour le POMO VNT.

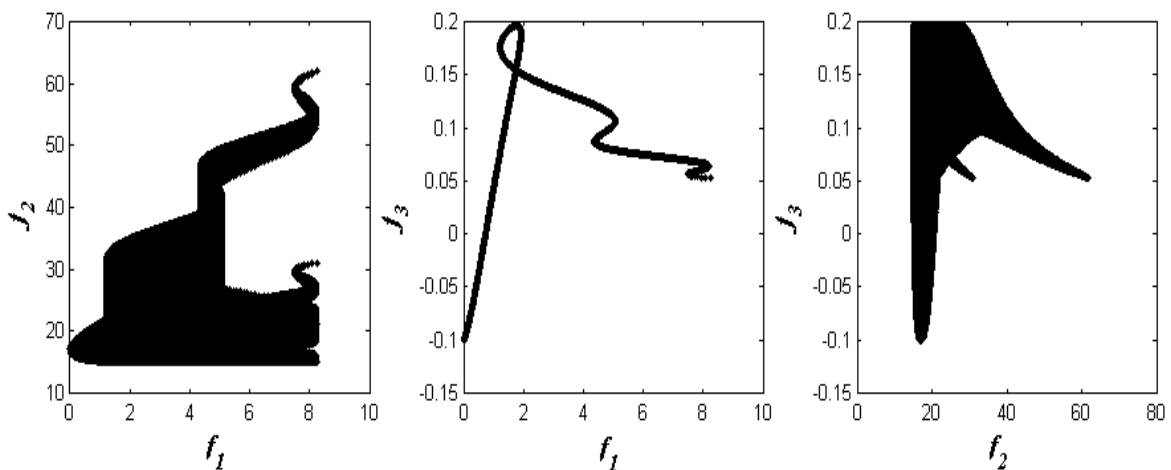


Figure 52– Représentation des espaces fonctionnels du POMO VNT.

III.3.2.3.8 TNK

Tanaka [Tanaka et al, 1995] propose un problème d'optimisation multi-objectifs très compliqué a deux variables et deux fonctions à minimiser en présence de deux contraintes. Pour plus de détails le lecteur peut consulter [Deb, 2001] [Deb et al, 2001] :

$$\text{TNK : } \left\{ \begin{array}{l} \text{Minimiser } f_1(x_1, x_2) = x_1 \\ \text{Minimiser } f_2(x_1, x_2) = x_2 \\ \text{satisfaisant } g_1(x_1, x_2) = x_1^2 + x_2^2 - 1 - 0.1 \cos \left(16 \arctan \frac{x_1}{x_2} \right) \geq 0 \\ g_2(x_1, x_2) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5 \\ 0 \leq x_1, x_2 \leq \pi \end{array} \right. \quad (\text{III.19})$$

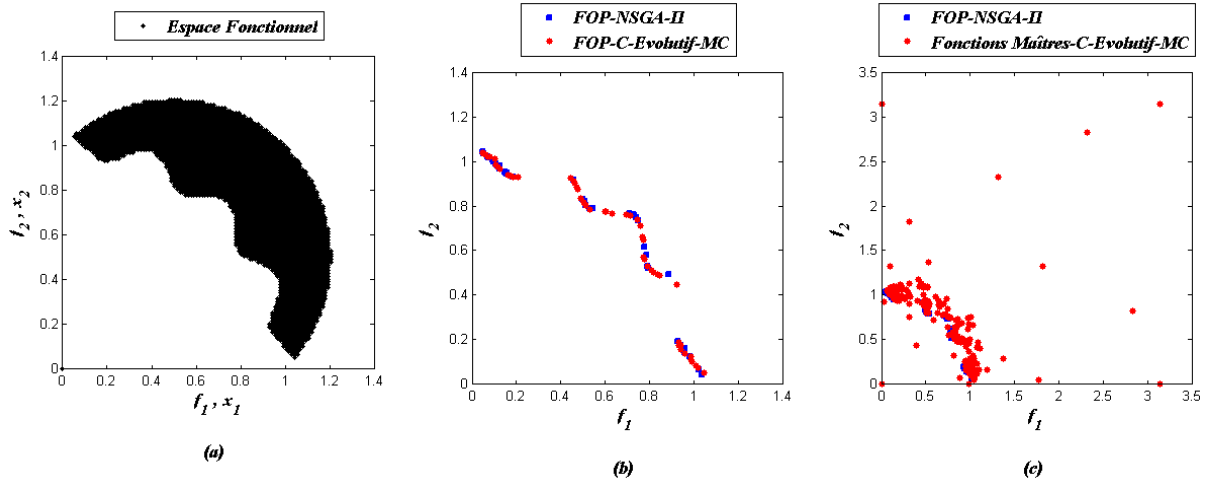


Figure 53– Représentation de (a) l'espace fonctionnel, (b) des FOP obtenus avec NSGA-II et C-Évolutif-MC et (c) la convergence de C-Évolutif-MC vers le FOP du POMO TNK obtenu avec NSGA-II.

III.3.2.3.9 KUR

KUR est un problème d'optimisation multi-objectifs très compliqué qui possède deux fonctions coût et trois variables de décisions. Il été proposé par Kursawe [Kursawe, 1990] et modifiée par Deb [Deb, 2001]. Il est donné par :

$$\text{KUR : } \left\{ \begin{array}{l} \text{Minimiser } f_1(X) = \sum_{i=1}^2 \left[-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right] \\ \text{Minimiser } f_2(X) = \sum_{i=1}^3 \left[|x_i|^{0.8} + 5 \sin(x_i^3) \right] \\ -5 \leq x_i \leq 5, \quad i = 1, 2, 3 \end{array} \right. \quad (\text{III.20})$$

Dans ce cas, le front optimal de Pareto est non convexe et discontinu. Il y a trois fronts différents et discontinus. Les valeurs des variables de décision correspondants à l'ensemble optimal analytique sont difficiles à connaître.

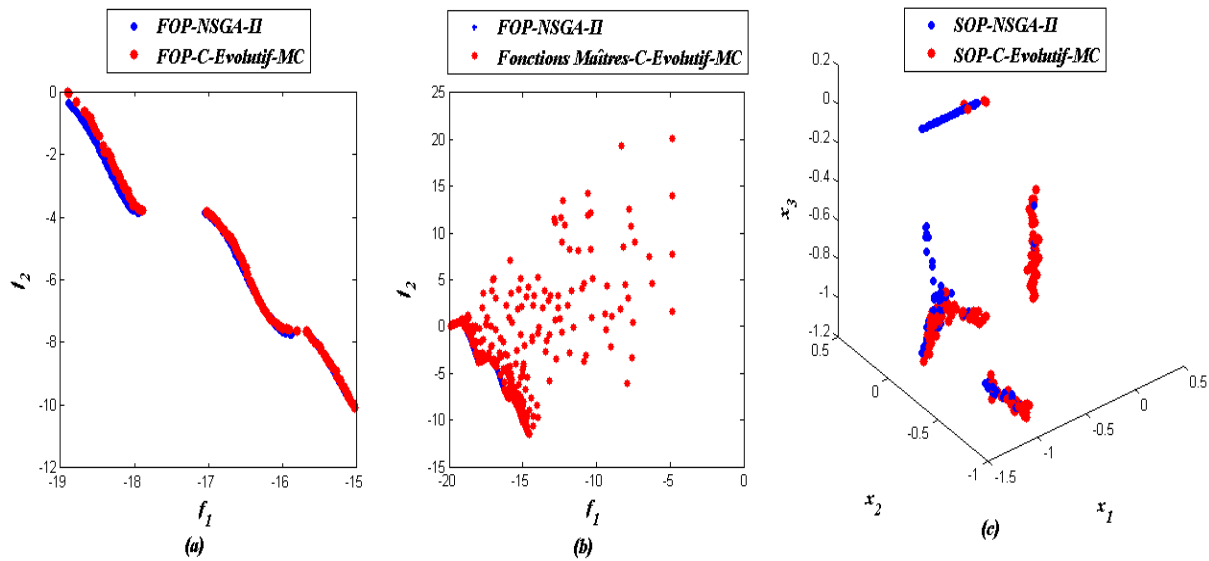


Figure 54– Comparaison des (a) FOP obtenus avec NSGA-II et C-Évolutif-MC, la convergence de C-Évolutif-MC vers (b) le FOP et (c) SOP obtenus avec NSGA-II pour POMO KUR.

Pour tester l'efficacité de notre couplage, C-Évolutif-MC, on a utilisé des problèmes d'optimisation multi-objectifs (POMOs) très compliqués. Ces problèmes possèdent des fronts optimaux de Pareto (FOP) et des ensembles optimaux de Pareto (SOP) discontinus. Ces discontinuités posent des problèmes aux algorithmes évolutionnaires multi-objectifs pour trouver les différentes parties de FOP ainsi au niveau de dispersion des SOP sur ce front.

La Figure 49 et la Figure 50 montrent la performance de notre algorithme, C-Évolutif-MC, qui permet de résoudre les deux POMOs, POL et VNT, en utilisant seulement 100 calculs exacts tout en retrouvant toutes les parties de FOP et de SOP (Figure 49-(b), Figure 49-(d), Figure 50 et Figure 51-(a)). Cet algorithme est aussi précis que NSGA-II standard, qui requiert pourtant 101 fois plus de calculs exacts (10100 dans ce cas).

Pour le POMO, TNK, l'espace fonctionnel est le même que celui de décision (Figure 53-(a)). Cette similarité de deux espaces conduit à un FOP identique au SOP. Le FOP représenté par la Figure 53-(b) est composé de trois fronts discontinus et non linéaires. Malgré ce non linéarité, qui empêche les algorithmes d'optimisation de trouver une bonne dispersion des solutions sur le FOP [Deb et al, 2001], C-Évolutif-MC, avec seulement 200 calculs exacts, trouve les trois parties du FOP (SOP) avec une bonne diversité des points sur ces dernières. La Figure 53-(b) montre que ces résultats très satisfaisants, sont aussi précis que ceux obtenus avec NSGA-II avec 25100 calculs exacts.

Les résultats de la Figure 54 confirment la performance et l'efficacité de notre algorithme, C-Évolutif-MC, qui permet de résoudre le POMO compliqué, KUR, en utilisant seulement 418 calculs exacts. Il ne permet pas seulement de déterminer les trois fronts discontinus et non linéaires mais aussi il assure une bonne diversité de points sur ces fronts. Il donne des résultats de même qualité que NSGA-II (utilisant 37650 calculs exacts) et avec un coût très réduit.

La robustesse et la performance de notre approche, C-Évolutif-MC, peuvent s'expliquer par l'efficacité de l'utilisation de la fonction mérite pour l'enrichissement du métamodèle

construit initialement avec un nombre fixe de points maîtres, générés par un plan d'expérience (latin hypercube), bien dispersés dans l'espace de recherche. Cette stratégie d'ajout de nouveaux points maîtres assure une exploration de l'espace de recherche dans les zones inexplorées et une exploitation au voisinage de FOP (Figure 49-(c), Figure 51-(b), Figure 53-(c) et Figure 54-(b)).

III.4 CONCLUSION

Le couplage entre un métamodèle et un algorithme évolutionnaire multi-objectifs (AEMO) pour la résolution des problèmes d'optimisation multi-objectifs (POMOs) en mise en forme est aujourd'hui en plein développement. Ceci permet de mettre en œuvre l'optimisation dans le secteur industriel tout en surmontant le coût important en terme de temps de calcul. Dans la recherche d'un couplage efficace et robuste pour la résolution des problèmes d'optimisation en mise en forme, on a étudié la performance de quatre algorithmes développés dans cette thèse, C-Constant, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC, sur des problèmes d'optimisation tests (mono-objectif et multi-objectifs). Ces algorithmes correspondent aux différents couplages de notre métamodèle à base de différences finies sans maillage, M-MDFSM, avec l'algorithme NSGA-II. Ils diffèrent par le choix des points maîtres, par l'évolution du métamodèle et par la fonction mérite utilisée.

Pour la comparaison de différents types de couplage, on a utilisé plusieurs types de problèmes tests. On trouve ceux sans contraintes ayant un front optimal de Pareto (FOP) convexe (Min-Ex et SCH1), concave (Max-Ex), discontinu (SCH2, Pol, VNT et KUR) et ceux sous contraintes comme Const-Min-Ex et TNK.

Au début, les quatre algorithmes sont comparés sur des problèmes tests mono-objectif (Camel-Back, Griewank, Rastrigin et Rosenbrock) et des POMOs moins compliqués (Min-Ex, Max-Ex, SCH1, SCH2 et Const-Min-Ex). Les résultats obtenus montrent que C-Évolutif-MC est plus performant que les autres algorithmes proposés.

Ensuite, C-Évolutif-MC est comparé à NSGA-II sur des POMOs très compliqués ayant des fronts et des ensembles optimaux de Pareto non linéaires et discontinus tels que POL, VNT, TNK et KUR. Les résultats obtenus confirment l'efficacité de notre algorithme, C-Évolutif-MC, qui permet de retrouver des résultats aussi précis que NSGA-II pour tous les problèmes proposés et avec un gain considérable en terme de temps de calcul.

CHAPITRE IV

OPTIMISATION DES PROCÉDES DE MISE EN FORME

IV.1 INTRODUCTION

Après sa validation sur des problèmes d'optimisation tests analytiques (voir Chapitre III), notre algorithme, C-Évolutif-MC, est intégré dans le logiciel éléments finis (EF), Forge2009®, pour la résolution des problèmes d'optimisation réels proposés par nos partenaires industriels. Donc les points maîtres utilisés pour construire notre métamodèle sont des simulations EF très coûteuses en terme de temps de calcul. Dans ce chapitre, on a traité deux types de problèmes d'optimisation en mise en forme. Le premier est un problème mono-objectif, tels que le cas de tréfilage proposé par ArcelorMittal, de remplissage d'une couronne proposé par Transvalor et de remplissage d'une bielle proposé par Atelier des Janves. Pour ces problèmes d'optimisation mono-objectif, l'apport de notre algorithme, C-Évolutif-MC, est évalué par des comparaisons avec l'algorithme préalablement intégré dans Forge2009®, SE-META. Le deuxième est un problème multi-objectifs, qui traite le cas de tréfilage. Les résultats obtenus montrent la performance et l'efficacité de notre algorithme pour résoudre des problèmes de mise en forme avec un temps de calcul raisonnable.

Le reste de ce chapitre est structuré de la manière suivante : On commence par le paramétrage de l'algorithme C-Évolutif-MC. Ensuite, le paramétrage des procédés de mise en forme constitue la troisième partie. Dans la quatrième partie, on résout un problème d'optimisation mono-objectif, remplissage d'une bielle proposé par Atelier des Janves. La cinquième partie est consacrée à l'étude du problème d'optimisation mono-objectif de remplissage d'une couronne, proposé par Transvalor. Enfin, l'étude des problèmes de tréfilage, mono-objectif et multi-objectifs, proposés par ArcelorMittal constitue la sixième partie.

IV.2 PARAMETRAGE DE L'ALGORITHME C-ÉVOLUTIF-MC

Pour les problèmes d'optimisation mono-objectif ayant :

- un paramètre ($n = 1$), on utilise un nombre maximal de simulations (points maîtres) égal à $\lambda_{\max} = 25$.

- plus qu'un paramètre, on utilise un nombre maximal de simulations égal à $\lambda_{\max} = 20 \times n$.

Pour les problèmes d'optimisation multi-objectifs, on utilise un nombre maximal de simulations égal à $\lambda_{\max} = 50 \times n$.

Pour ces différents problèmes d'optimisation, mono-objectif ou multi-objectifs, on commence avec un métamodèle, M-MDFSM, construit à partir de $\lambda = \frac{(n+1)(n+2)}{2}$ simulations

généérées par le plan d'expérience, latin hypercube, et enrichi par l'ajout de $\theta=2^{M-1}$ nouveaux points maîtres durant les $N_u = \frac{\lambda_{\max} - \lambda}{\theta}$ générations. Ici, M est le nombre des fonctions coût.

Pour les problèmes mono-objectif, les simulations sont lancées en séquentiel. Par contre, les simulations des problèmes d'optimisation multi-objectifs sont lancées en parallèle sur θ processeurs.

IV.3 PARAMETRAGE DES PROCEDES DE MISE EN FORME

Dans cette section, on présente la loi de comportement, de frottement, le maillage, le pas de temps et le nombre des incréments utilisés pour la simulation des trois procédés de mise en forme : procédé de remplissage d'une bielle, de remplissage d'une couronne et de tréfilage.

IV.3.1 Procédé de remplissage d'une bielle

Le procédé de remplissage 3D d'une bielle à chaud (Figure 56), proposé par Atelier des Janves, utilise un lopin en acier (38MnSi4) avec un maillage 3D composé de 40637 éléments tétraédriques et de 162548 nœuds et deux outils rigides. Chacun de ces outils utilise un maillage triangulaire de 20656 éléments et 61968 nœuds.

Une simulation de ce procédé est composée de trois phases : Répartition, ébauche et finition.

On utilise 180 incréments de temps pour la répartition, 100 pour l'ébauche et 45 pour la finition ; un pas de temps correspond à 10^{-4} seconde.

Pour la simulation de ce procédé de mise en forme, on utilise :

La loi d'écoulement visco-plastique suivant le modèle de Hansel et Spittel. Elle est thermo-dépendante et, tient compte des phénomènes d'écrouissage et d'adoucissement. Elle prend en compte la vitesse de déformation du matériau. Elle se présente sous la forme :

$$\sigma = A e^{m_1 T} T^{m_9} \varepsilon^{m_2} e^{\frac{m_4}{\varepsilon}} (1 + \varepsilon)^{m_5 T} e^{m_7 \varepsilon} \dot{\varepsilon}^{m_3} \dot{\varepsilon}^{m_8 T} \quad (\text{IV.1})$$

Où

σ est la contrainte.

T est la température donnée en Celsius.

m_1 et m_9 définissent la sensibilité du matériau à la température

m_5 terme couplé température et déformation.

m_8 terme couplé température et vitesse de déformation.

m_2 , m_4 et m_7 définissent la sensibilité du matériau à la déformation.

m_3 dépend de la sensibilité du matériau à la vitesse de déformation.

Pour notre cas le matériau utilisé présente des valeurs nulles pour les paramètres m_5 à m_9 . On simplifie donc l'équation comme ceci :

$$\sigma = Ae^{m_1 T} \varepsilon^{m_2} e^{\frac{m_4}{\varepsilon}} \dot{\varepsilon}^{m_3} \quad (\text{IV.2})$$

Où : $A=1809.2421$, $m_1=-0.0029$, $m_2=-0.14227$, $m_3=0.14102$ et $m_4=-0.05764$.

La loi de frottement de Coulomb :

$$\tau = \mu \sigma_n \begin{cases} \text{si } \mu \sigma_n < \bar{m} \frac{\sigma_0}{\sqrt{3}} \text{ alors } \tau = \mu \sigma_n \\ \text{si } \mu \sigma_n \geq \bar{m} \frac{\sigma_0}{\sqrt{3}} \text{ alors } \tau = \bar{m} \frac{\sigma_0}{\sqrt{3}} \end{cases} \quad (\text{IV.3})$$

Où

σ_n est la contrainte normale (pression de contact).

σ_0 est la contrainte équivalente de von Mises.

μ est le coefficient de frottement. Ici il est égal à 0.4.

\bar{m} est un nombre compris entre 0 et 1. Ici il est égal à 0.8.

IV.3.2 Procédé de remplissage d'une couronne

Pour le forgeage 2D à chaud d'une couronne (Figure 61), on utilise un lopin cylindrique en acier (C45) avec un maillage composé de 4791 éléments triangulaires et 19164 nœuds et deux outils, chacun a 500 éléments triangulaires et 1500 nœuds.

Pour ce procédé, une simulation est composée de trois opérations. La première est composée de 451 incréments de temps avec un pas de $6 \cdot 10^{-2}$ s. La deuxième est composée de 566 incréments de temps avec un pas de $5 \cdot 10^{-2}$ s. Et la dernière est composée de 367 incréments avec un pas de temps égal à $3 \cdot 10^{-2}$ s.

On utilise :

La loi d'écoulement visco-plastique suivant le modèle de Hansel et Spittel (Eq (IV.2)) avec les paramètres suivants : $A= 1521.306$, $m_1=-0.00269$, $m_2=-0.12651$, $m_3= 0.14542$ et $m_4=-0.05957$.

La loi de frottement de Coulomb (Eq (IV.3)) avec : $\mu =0.15$ et $\bar{m}=0.3$.

IV.3.3 Procédé de tréfilage

Pour le procédé de tréfilage 2D à froid d'un fil en acier (C75) avec un maillage de 3528 éléments triangulaires et 14112 nœuds à travers une ou plusieurs filières avec un maillage de 8 éléments triangulaires et 24 nœuds (Figure 67), on utilise :

La loi linéaire puissance :

$$\sigma = \sqrt{3}^{m+1} K e^{\frac{\beta}{T}} \dot{\epsilon}^m (1 + a\bar{\epsilon}^n) \quad (\text{IV.4})$$

Où

σ est la contrainte.

T est la température donnée en Kelvin.

m est le coefficient de sensibilité à la vitesse de déformation.

K est la consistance du matériau.

β est le terme de thermo-dépendance.

n est un terme d'écrouissage.

a est un terme constant d'écrouissage.

$\bar{\epsilon}$ est la déformation plastique.

$\dot{\epsilon}$ est la vitesse de déformation plastique.

On utilise les valeurs suivantes : $a=7.32$, $\beta=0$, $n=0.13$ et $K=100.64$.

Pour le frottement entre le fil et la (les) filière (s), on utilise la **loi de Tresca** :

$$\tau = \bar{m} \frac{\sigma_0}{\sqrt{3}} \quad (\text{IV.5})$$

Où

σ_0 est la contrainte équivalente de Von Mises.

\bar{m} est un nombre compris entre 0 et 1. Ici il est égal à 0.02.

IV.4 OPTIMISATION DU PROCEDE DE REMPLISSAGE D'UNE BIELLE

Une première application de C-Évolutif-MC en mise en forme des matériaux est réalisée sur un problème d'optimisation de forme en 3D d'un lopin en acier pour le forgeage d'une bielle (**Figure 56**). Ce lopin cylindrique de volume initial $V_0=80753 \text{ mm}^3$ est défini par ses diamètres initiaux, D_i , $i=1,\dots,4$ et sa longueur L , qui est considérée constante (**Figure 55** et **Tableau 11**).

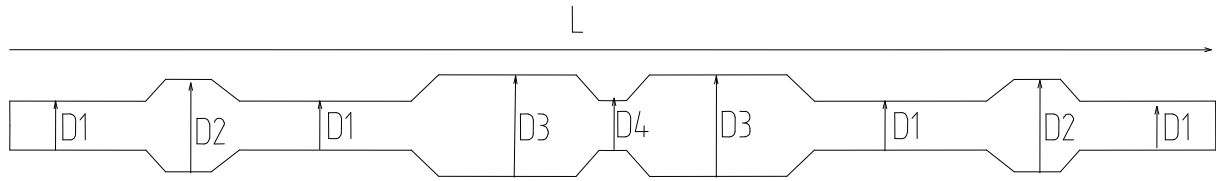


Figure 55- Géométrie du lopin et définition des paramètres de l'optimisation

D ₁	D ₂	D ₃	D ₄	L
38 mm	72 mm	80 mm	40 mm	262 mm

Tableau 11- Paramétrage du lopin

Le problème d'optimisation mono-objectif consiste à minimiser le volume de matière du lopin, V , en respectant l'écart entre la pièce désirée et celle forgée. Ce dernier est évalué par le non remplissage des gravures de forgeage (Figure 57) et la distance pièce/outil en dehors des zones de bavure (Figure 58). Le non remplissage est calculé de la manière suivante :

$$\text{non-remplissage outil} = \frac{S_{\text{outil}} - S_{\text{outil/pièce}}}{S_{\text{outil}}} \quad (\text{IV.6})$$

Où S_{outil} et $S_{\text{outil/pièce}}$ sont respectivement les surfaces de l'outil et de contact de l'outil avec la pièce.

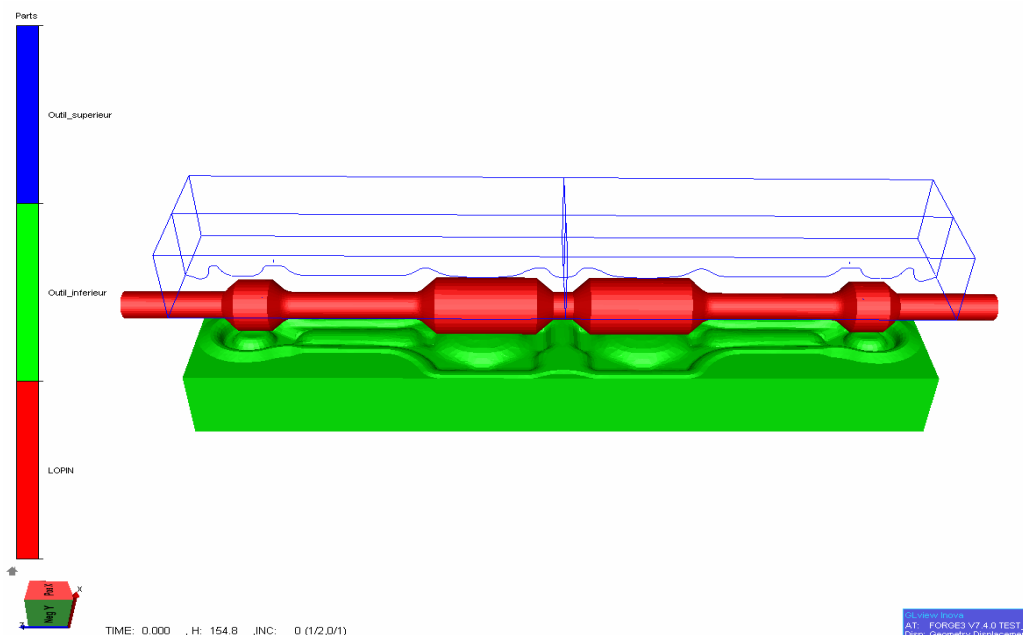


Figure 56- Visualisation de la simulation du forgeage d'une bielle sous GLView

du problème PB2 en utilisant l'algorithme C-Évolutif-MC demande un temps CPU = **121h34min57s**, qui est clairement inférieur à celui nécessaire pour la résolution du problème PB1 avec SE-META (CPU = **145h18min17s**) ou avec C-Évolutif-MC (CPU = **130h23min41s**) qui nécessite moins de temps que SE-META.

Le **Tableau 12** de comparaison des résultats montre l'efficacité de C-Évolutif-MC par rapport à SE-META pour la résolution du problème PB1. Notre algorithme, C-Évolutif-MC, permet de trouver la solution optimale après 8 simulations avec un temps CPU-simulation optimale = **39h11min11s**. Par contre, SE-META trouve la solution optimale à la dernière simulation (simulation 25) avec un temps CPU = **145h18min17s**.

La résolution du problème mono-objectif sous contrainte, PB2, avec C-Évolutif-MC permet un gain important au niveau de temps CPU par rapport à la résolution du problème PB1, où on utilise une somme pondérée de la fonction coût et des contraintes, avec C-Évolutif-MC. Pour le problème PB2 la solution optimale est obtenue à la quatrième simulation avec un temps CPU-simulation optimale = **15h56min17s** qui est inférieur à la moitié du temps CPU-simulation optimale = **39h11min11s** nécessaire pour obtenir la solution optimale du problème PB1.

La **Figure 59** montre que le métamodèle, M-MDFSM, fournit une assez bonne estimation de la fonction coût et des contraintes. Ce métamodèle permet d'observer plus finement comment le volume de matière est relié au défaut de remplissage des gravures pour estimer, par exemple les intervalles de sécurité à considérer pour garantir la robustesse de l'opération de forgeage.

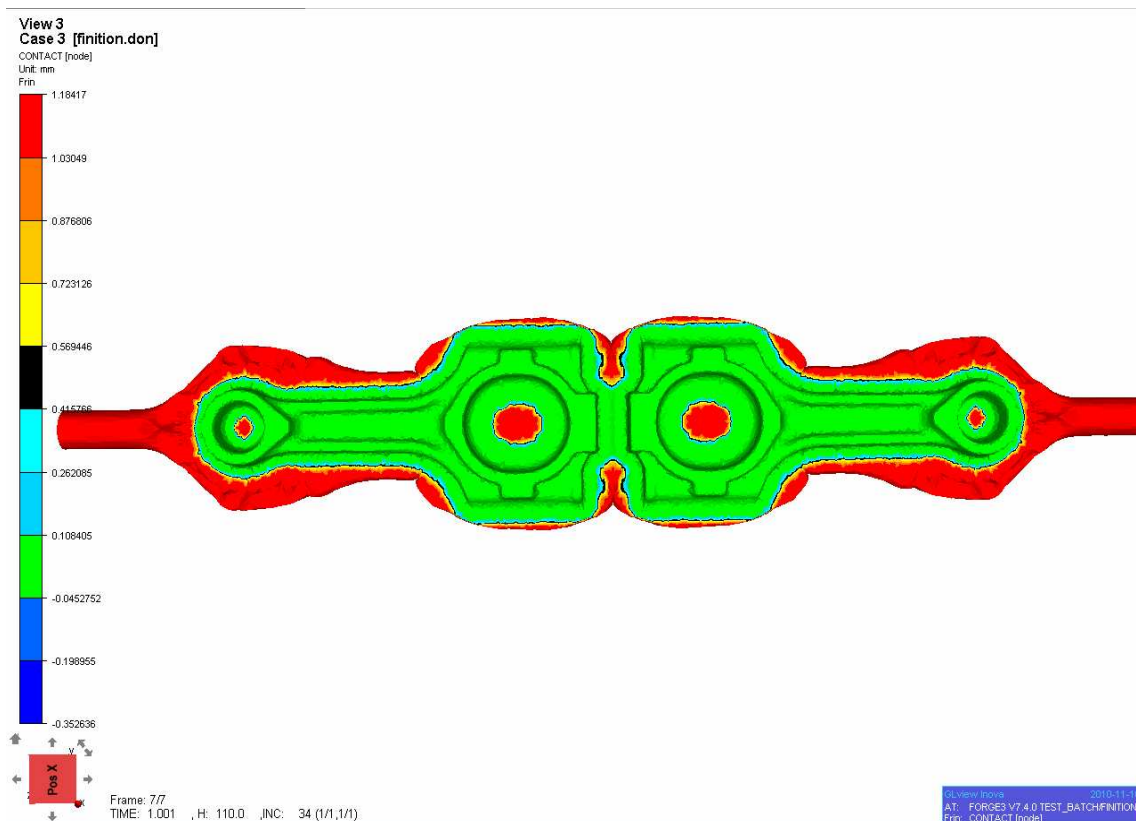


Figure 57 – Forgeage d'une bielle avec Forge3® : iso valeurs de contact de la solution optimale du PB2

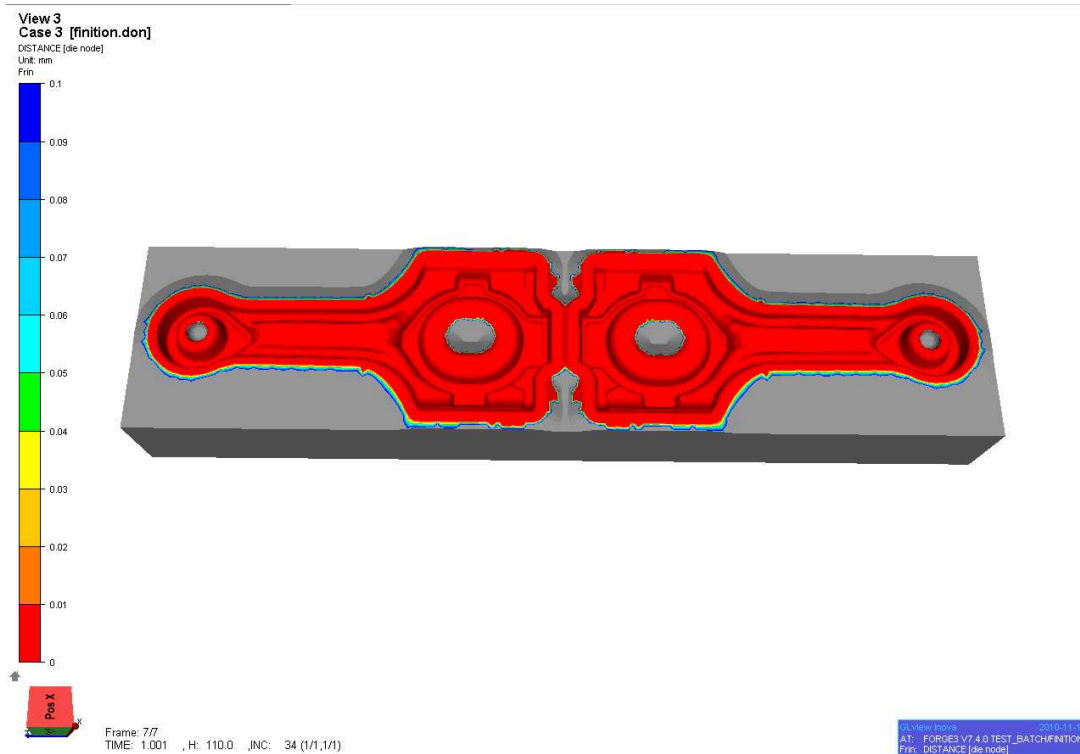


Figure 58 – Forgeage d’une bielle avec Forge3® : iso valeurs distance pièce/outil de la solution optimale du PB2

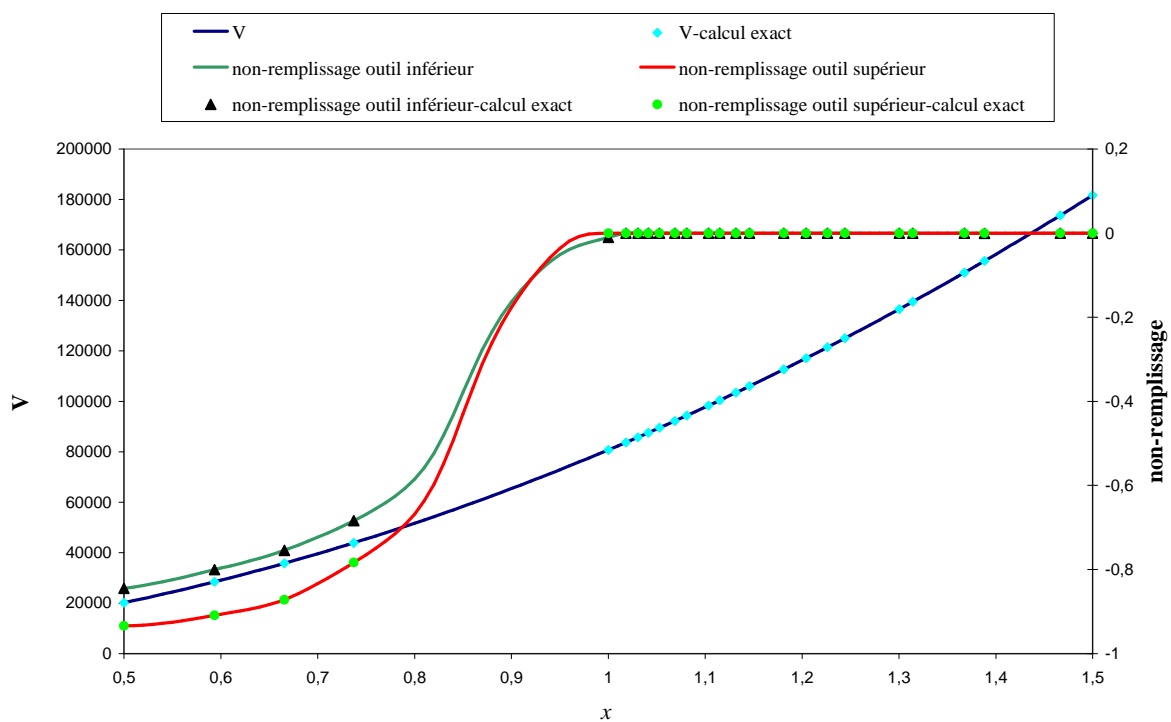


Figure 59 – Approximation de la fonction coût volume, V , de la contrainte non-remplissage outil inférieur et de la contrainte non-remplissage outil supérieur du PB2 avec le métamodèle M-MDFSM

figures, **Figure 62** et **Figure 63**, montrent que l'écart entre la pièce désirée et celle forgée est satisfaisant. La **Figure 64** montre que l'utilisation du métamodèle, M-MDFSM, pour le traçage de la fonction coût et des contraintes permet d'observer plus finement comment le volume de matière est relié au défaut de remplissage des gravures.

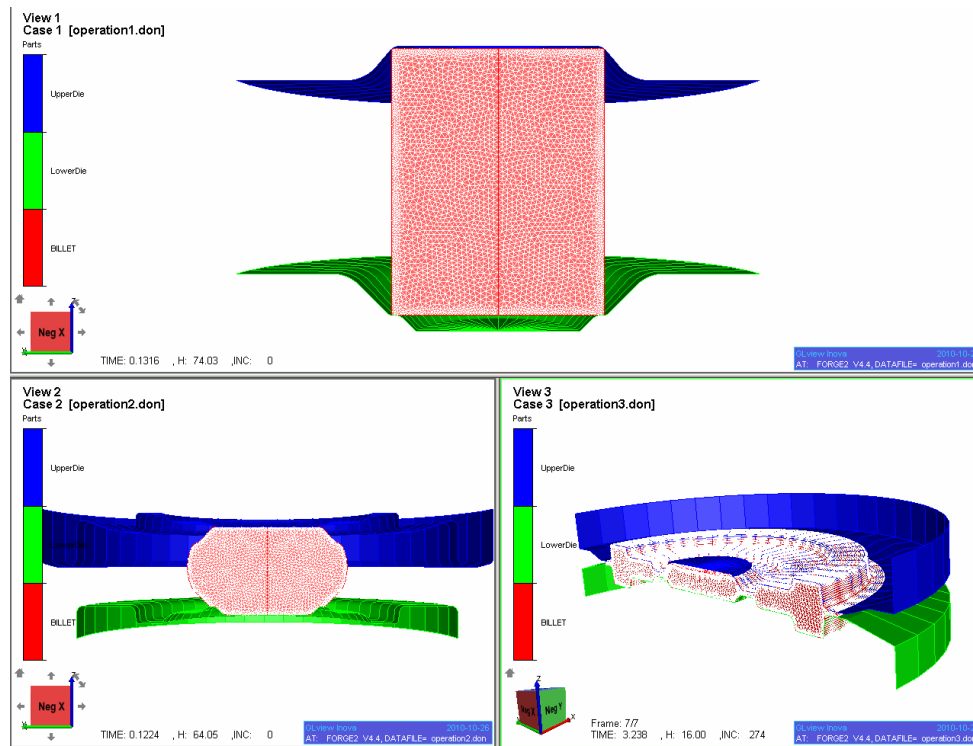


Figure 60- Visualisation 3D de la simulation du forgeage d'une couronne sous GLView

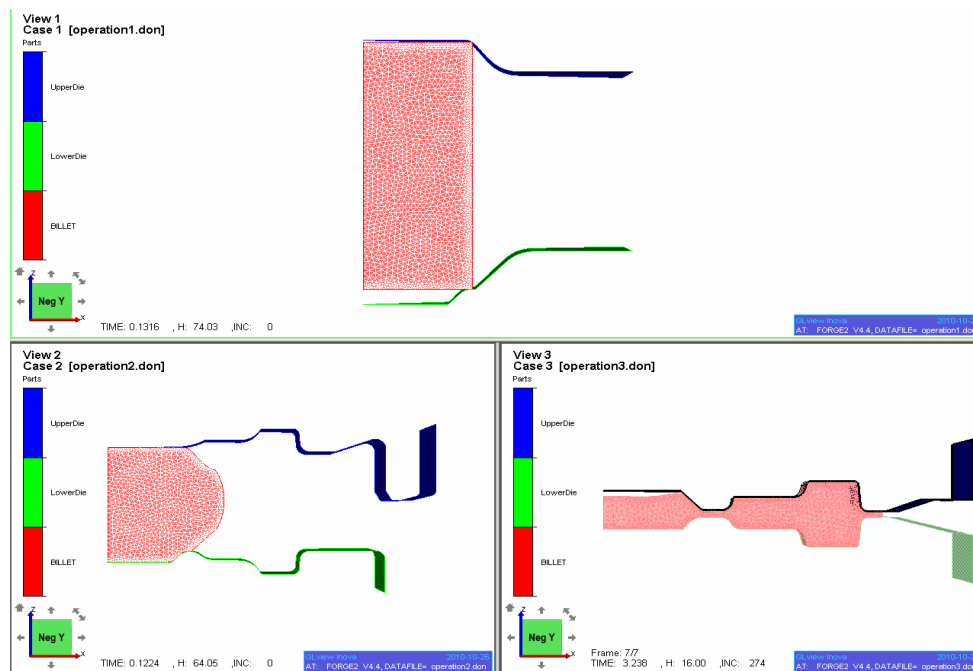


Figure 61- Visualisation 2D de la simulation du forgeage d'une couronne sous GLView

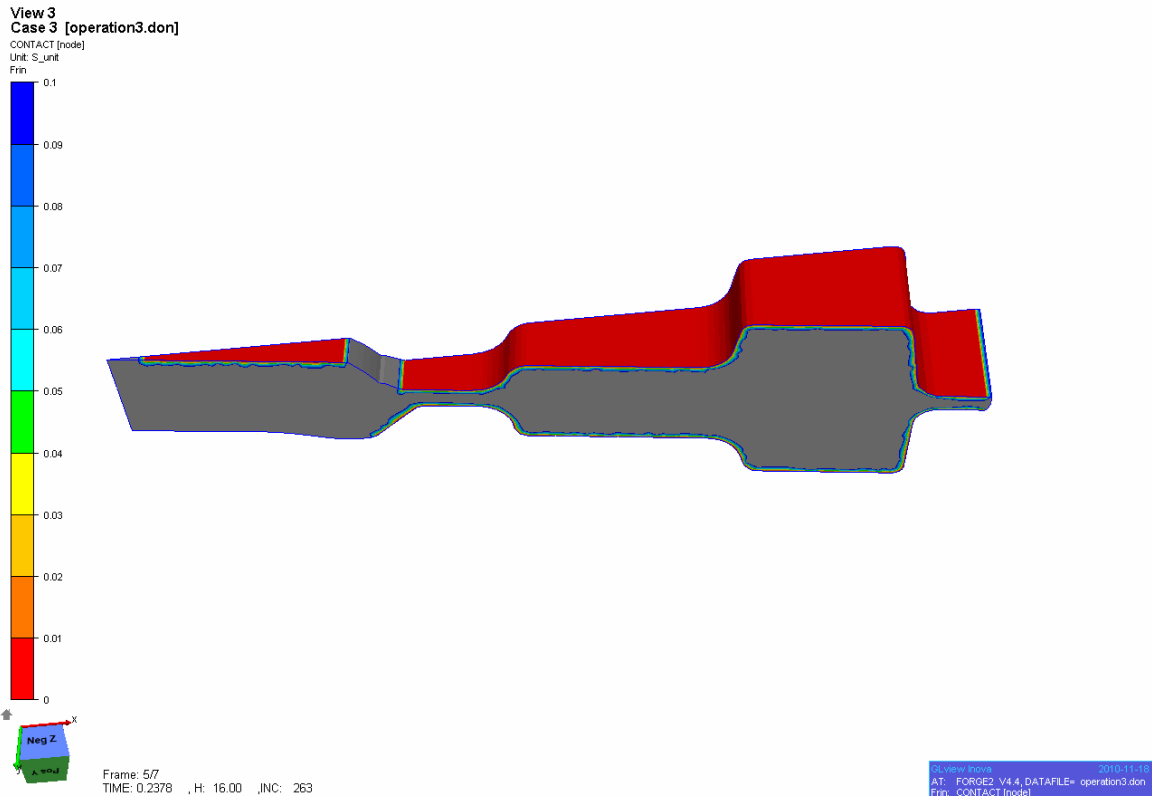


Figure 62 – Forgeage d'une couronne avec Forge2® : iso valeurs de contact de la solution optimale du PCR2

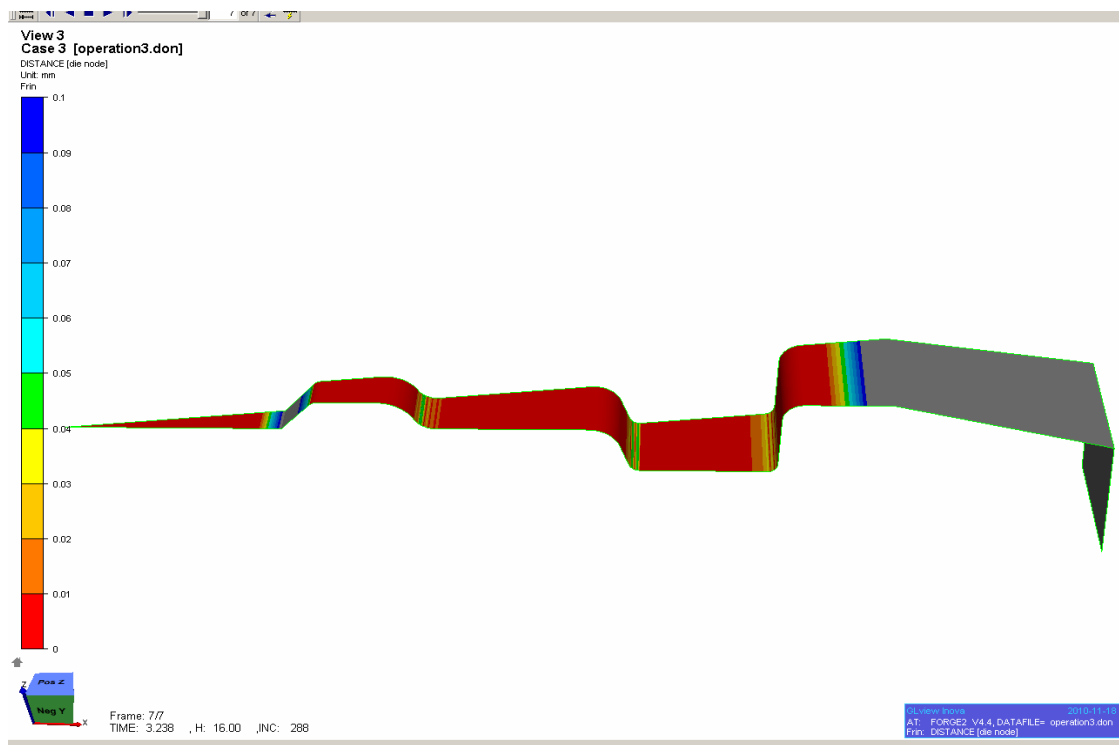


Figure 63 – Forgeage d'une couronne avec Forge2® : iso valeurs distance pièce/outil de la solution optimale du PCR2

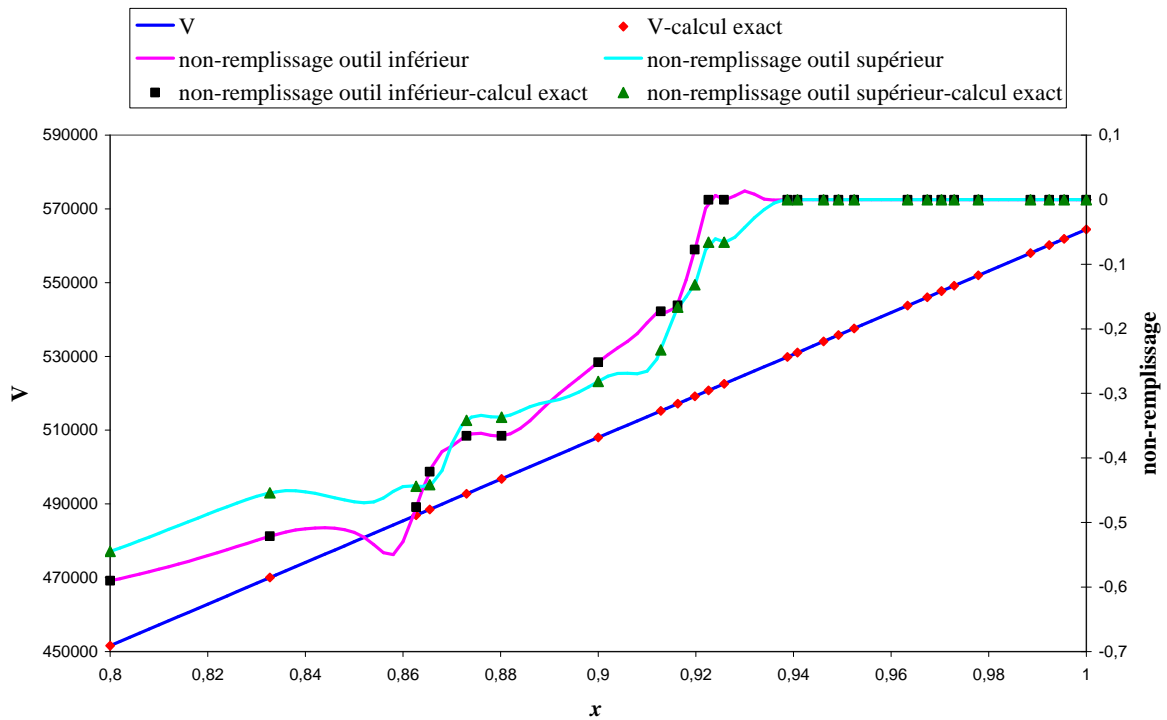


Figure 64 – Approximation de la fonction coût volume, V , de la contrainte non-remplissage outil inférieur et de la contrainte non-remplissage outil supérieur du PCr2 avec le métamodèle M-MDFSM

IV.6 OPTIMISATION DU PROCÉDE DE TRÉFILAGE

IV.6.1 Description

Ce procédé de mise en forme, tréfilage à froid, est proposé par ArcelorMittal dans le cadre du projet LOGIC. C'est un problème 3D (Figure 66) ayant une symétrie axiale. Il est traité comme un problème 2D (Figure 67). Il consiste à réduire la section d'un fil par sa traction à travers une ou plusieurs filières (voir Figure 65). À la fin du processus, un niveau de résistance élevé est atteint par le durcissement dû à la réduction de la section initiale du fil.

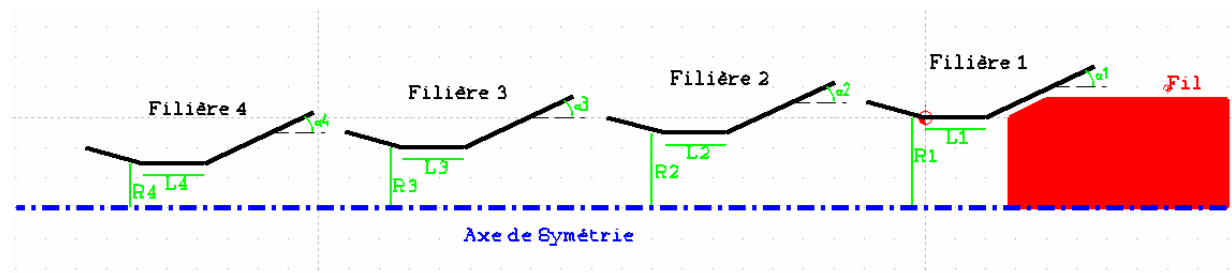


Figure 65- Schéma d'un cas de tréfilage à 4 filières.

L'optimisation du procédé de tréfilage à froid vise à chercher la forme optimale des filières qui permet de réduire l'endommagement maximal du fil et de minimiser la force de traction maximale.

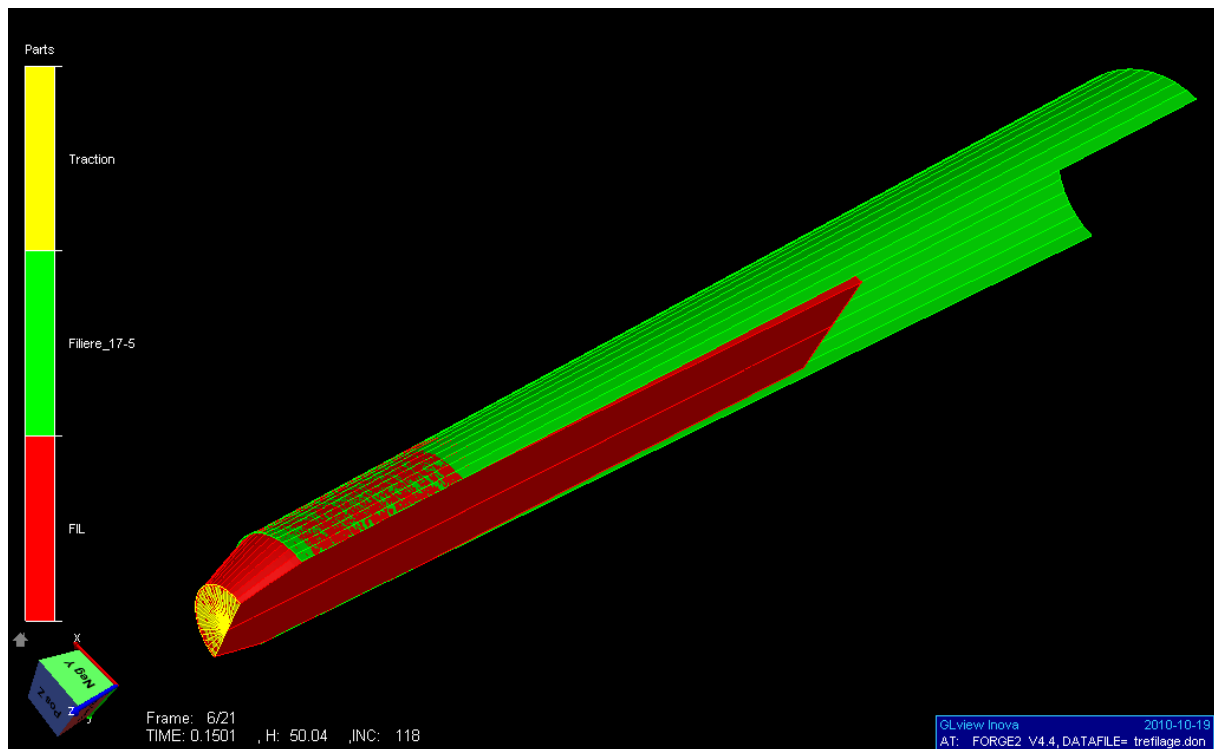


Figure 66- Visualisation de la simulation 3D de tréfilage à deux paramètres sous GLView

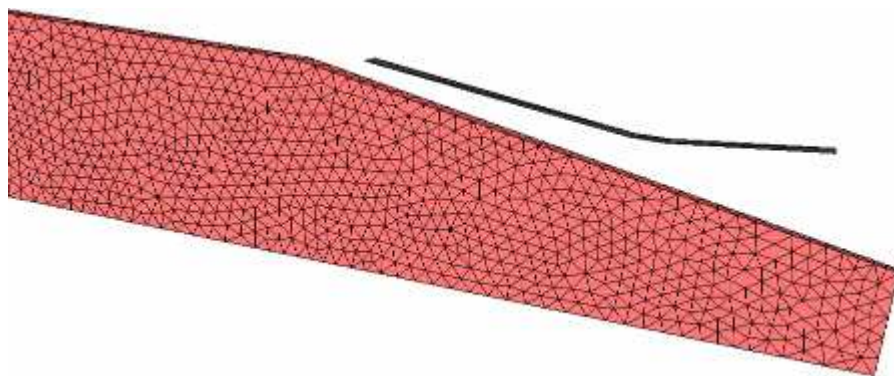


Figure 67- Visualisation de la simulation 2D de tréfilage à deux paramètres sous GLView

IV.6.2 Choix des paramètres d'optimisation

Trouver la forme optimale du procédé de tréfilage revient à déterminer la géométrie optimale de chaque filière. Cette géométrie est caractérisée par les trois paramètres présentés à la Figure 68 :

- Le demi angle d'entrée de la filière α (en $^\circ$) où sa valeur correspond à une dilatation non isotrope de centre C dans la direction X .
- La longueur de la portée L (en mm) qui est la zone de contact entre la filière et le fil. Elle a une valeur qui correspond à une dilatation isotrope de centre C dans les deux directions X et Z .

- Le taux de réduction τ qui est défini par le rapport $\frac{S_0 - S_f}{S_0}$, où S_0 et S_f sont respectivement la section d'entrée et de sortie du fil pour chaque filière. Il correspond à une translation suivant la direction X (variation du rayon R (en mm) de la filière).

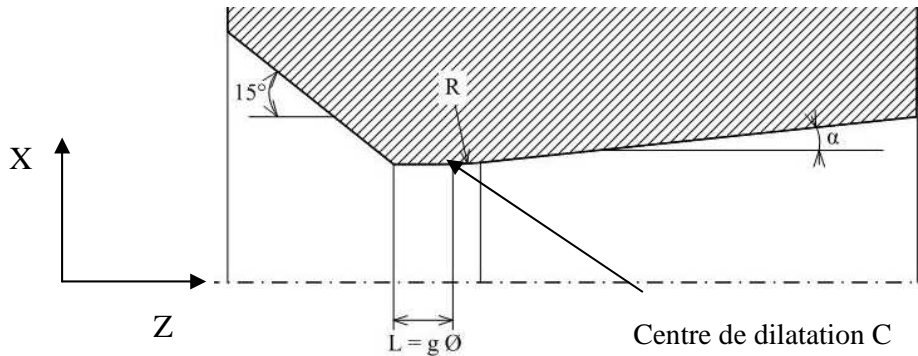


Figure 68- Géométrie de la filière et définition des paramètres de l'optimisation

IV.6.3 Choix des fonctions coût

Notre objectif est d'améliorer le procédé de tréfilage en minimisant l'endommagement maximal D_{\max} et la force de traction maximale (force de tréfilage) F_{\max} . Pour évaluer l'endommagement, on utilise le critère de Latham & Cockcroft car il donne une évolution correcte de l'endommagement au cours d'un procédé de mise en forme à froid [Massé, 2010], il s'écrit sous la forme suivante :

$$D_c = \int_0^{\varepsilon_{eq,f}} \sigma_I d\varepsilon_{eq} \quad (IV.11)$$

D_c est l'endommagement critique.

σ_I est la contrainte principale maximale.

ε_{eq} est la déformation équivalente.

$\varepsilon_{eq,f}$ est la déformation équivalente à la rupture.

IV.6.4 Étude de l'optimisation du procédé de tréfilage

Dans cette section, on étudie différents types de problèmes d'optimisation du procédé de tréfilage. Selon la ou les fonctions coût utilisées, on distingue deux types de problèmes. Un problème d'optimisation mono-objectif où on minimise une seule fonction coût, soit l'endommagement maximal ou la force de traction maximale, ou une somme pondérée des deux fonctions coût. Le deuxième problème est un problème d'optimisation multi-objectifs ayant deux fonctions coût.

IV.6.4.1 Optimisation mono-objectif du procédé de tréfilage

Selon le type et le nombre des paramètres utilisés, on étudie trois types de problèmes d'optimisation mono-objectif. Chacun de ces problèmes est traité de trois manières : (1) optimisation de l'endommagement maximal (PT-n-D), (2) optimisation de la force maximale (PT-n-F) et (3) optimisation de la somme pondérée des deux fonctions coût (PT-n-SP).

IV.6.4.1.1 Tréfilage à deux paramètres

Le problème d'optimisation du procédé de tréfilage, représenté par la **Figure 67**, est connu sous le nom de tréfilage à une passe. Ce procédé consiste à utiliser une seule filière. La géométrie optimale de cette dernière dépend de la longueur de la portée L et de demi angle d'entrée α . On commence avec une filière de longueur de portée $L=2.8841$ mm, de demi angle $\alpha=11.7204^\circ$ et de rayon $R=8.5$ mm et un fil de rayon 9.5 mm.

Optimisation de l'endommagement maximal

Ce problème, (PT-2-D), consiste à chercher la forme optimale de la filière qui permet de minimiser l'endommagement maximal D_{\max} . Il est formulé de la manière suivante :

$$\text{PT-2-D: } \begin{cases} \text{Minimiser } f(L, \alpha) = D_{\max}(L, \alpha) \\ 0.2884 \leq L \leq 5.7682 \\ 1.1885 \leq \alpha \leq 22.5346 \end{cases} \quad (\text{IV.12})$$

Optimisation de la force de traction maximale

Ce problème, PT-2-F, consiste à déterminer le jeu des paramètres optimaux (L, α) qui minimise la force de traction maximale F_{\max} . Il est formulé de la manière suivante :

$$\text{PT-2-F: } \begin{cases} \text{Minimiser } f(L, \alpha) = F_{\max}(L, \alpha) \\ 0.2884 \leq L \leq 5.7682 \\ 1.1885 \leq \alpha \leq 22.5346 \end{cases} \quad (\text{IV.13})$$

Optimisation de la somme pondérée de l'endommagement et de la force de traction

Ce problème, PT-2-SP, consiste à déterminer le jeu des paramètres optimaux (L, α) qui minimise la somme $D_{\max} + 0.1 \times F_{\max}$. Il est formulé de la manière suivante :

$$\text{PT-2-SP:} \begin{cases} \text{Minimiser } f(L, \alpha) = D_{\max}(L, \alpha) + 0.1 \times F_{\max}(L, \alpha) \\ 0.2884 \leq L \leq 5.7682 \\ 1.1885 \leq \alpha \leq 22.5346 \end{cases} \quad (\text{IV.14})$$

		L	α	f	Numéro de la Simulation optimale	CPU- Simulation optimale	λ_{\max}	CPU-total
PT-2-D	SE-META	5.7682	1.1885	0.0472	28	2h29min3s	40	3h8min30s
	C-Évolutif-MC	5.7577	1.1896	0.0473	22	1h59min30s		3h23min53s
PT-2-F	SE-META	2.8193	4.3244	7.5151	13	1h21mn39s		4h1mn57s
	C-Évolutif-MC	1.9714	4.7485	7.4754	8	56mn7s		4h10mn5s
PT-2-SP	SE-META	2.9526	3.0470	0.8252	34	3h2mn24s		3h30mn52s
	C-Évolutif-MC	1.4142	3.0876	0.8136	32	2h55mn15s		3h33mn57s

Tableau 15- Comparaison entre les résultats obtenus avec C-Évolutif-MC et SE-META pour PT-2-D, PT-2-F et PT-2-SP

Le **Tableau 15** confirme la performance de notre algorithme, C-Évolutif-MC, pour résoudre des problèmes d'optimisation mono-objectif en mise en forme. Il trouve des solutions meilleures que celles obtenues par SE-META pour les deux problèmes, PT-2-F et PT-2-SP, et très proche pour le problème PT-2-D. Notre méthode permet une réduction considérable au niveau de de temps de calcul. Ce gain au niveau de temps CPU-Simulation optimale est dû à une convergence rapide vers les solutions optimales.

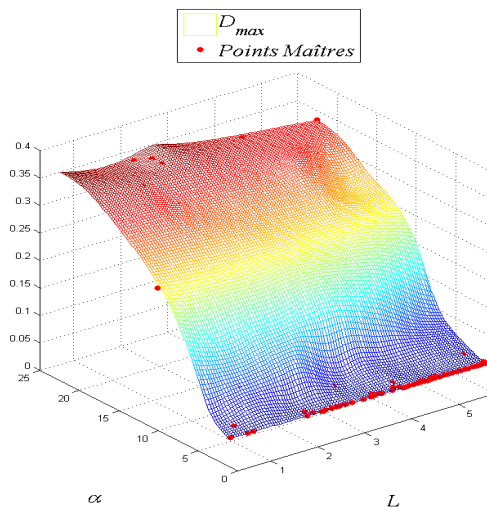


Figure 69- Approximation de la fonction coût, D_{\max} , du PT-2-D avec le métamodèle M-MDFSM

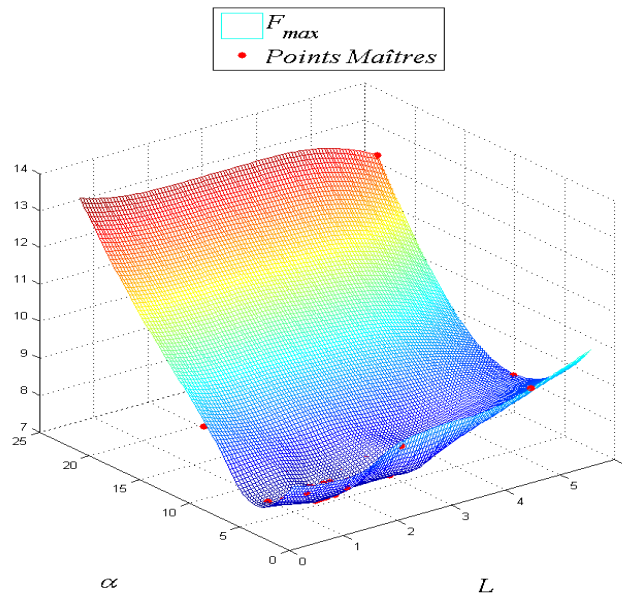


Figure 70- Approximation de la fonction coût, F_{max} , du PT-2-F avec le métamodèle M-MDFSM

Les deux figures, Figure 69 et Figure 70, montrent l'importance d'utiliser un métamodèle pour la compréhension du comportement des fonctions coût en fonction des paramètres. Ces deux figures montrent que quelque soit la fonction coût, la longueur de portée à une influence mineure sur l'optimum par rapport au demi angle d'entrée de la filière.

IV.6.4.1.2 Tréfilage à trois paramètres

Le procédé de tréfilage à trois paramètres consiste à utiliser quatre filières. Chaque filière ($i=1,\dots,4$) est caractérisée par sa longueur de portée fixe L_i , son demi angle d'entrée fixe α_i et son rayon R_i . Les géométries initiales de filières sont données par le tableau suivant :

	Filière 1	Filière 2	Filière 3	Filière 4
L	2.6246	3.3510	5.7680	2.8840
α	1.1887	1.1887	1.1918	0.7025
R	8.7	7.5	6.5	5.9

Tableau 16- Géométries initiales des quatre filières.

Le fil a un rayon initial de 9.5 mm.

L'optimisation de tréfilage à trois paramètres consiste à chercher les rayons optimaux des trois premières filières ($R_i, i=1,2,3$) qui minimise une fonction coût donnée (D_{max}, F_{max} ou $D_{max} + 0.1 \times F_{max}$).

Optimisation de l'endommagement maximal

Ce problème, PT-3-D, est donné sous la forme suivante :

$$\text{PT-3-D:} \begin{cases} \text{Minimiser } f(R_1, R_2, R_3) = D_{\max}(R_1, R_2, R_3) \\ 8.4 \leq R_1 \leq 9.4 \\ 7.05 \leq R_2 \leq 7.95 \\ 6 \leq R_3 \leq 7 \end{cases} \quad (\text{IV.15})$$

Optimisation de la force de traction maximale

Ce problème, PT-3-F, est donné sous la forme suivante :

$$\text{PT-3-F:} \begin{cases} \text{Minimiser } f(R_1, R_2, R_3) = F_{\max}(R_1, R_2, R_3) \\ 8.4 \leq R_1 \leq 9.4 \\ 7.05 \leq R_2 \leq 7.95 \\ 6 \leq R_3 \leq 7 \end{cases} \quad (\text{IV.16})$$

Optimisation de la somme pondérée de l'endommagement et de la force de traction

Ce problème, PT-3-SP, est donné sous la forme suivante :

$$\text{PT-3-SP:} \begin{cases} \text{Minimiser } f(R_1, R_2, R_3) = D_{\max} + 0.1 \times F_{\max} \\ 8.4 \leq R_1 \leq 9.4 \\ 7.05 \leq R_2 \leq 7.95 \\ 6 \leq R_3 \leq 7 \end{cases} \quad (\text{IV.17})$$

		R1	R2	R3	f	Numéro de la Simulation optimale	CPU- Simulation optimale	λ_{\max}	CPU-total
PT-3-D	SE-META	8.5795	7.5729	6.6009	0.3499	46	3h23min56s	60	4h26min0s
	C-Évolutif-MC	8.4007	7.5142	6.6233	0.3470	36	2h40min19s		4h27min12s
PT-3-F	SE-META	8.5919	7.6235	6.5912	9.3995	55	4h6mn57s		4h29mn24s
	C-Évolutif-MC	8.6048	7.7867	6.9812	8.7941	21	1h34mn13s		4h29mn12s
PT-3-SP	SE-META	8.5919	7.6235	6.5912	1.2898	55	4h3mn39s		4h25mn48s
	C-Évolutif-MC	8.5980	7.7671	6.9137	1.1964	42	3h10mn40s		4h32mn24s

Tableau 17- Comparaison entre les solutions optimales obtenues avec C-Évolutif-MC et SE-META pour les trois problèmes, PT-3-D, PT-3-F et PT-3-SP

		τ_1 (%)	τ_2 (%)	τ_3 (%)	τ_4 (%)
PT-3-D	SE-META	18.4398	22.0893	24.0222	20.1096
	C-Évolutif-MC	21.8036	19.9923	22.3073	20.6479
PT-3-F	SE-META	18.2033	21.2724	25.2490	19.8734
	C-Évolutif-MC	17.9591	18.1095	19.6195	28.5764
PT-3-SP	SE-META	18.2033	21.2724	25.2490	19.8734
	C-Évolutif-MC	18.0882	18.3942	20.7662	27.1752

Tableau 18- Comparaison entre les pourcentages des taux de réduction obtenus avec C-Évolutif-MC et SE-META pour les trois problèmes, PT-3-D, PT-3-F et PT-3-SP

Le **Tableau 17** confirme la robustesse de notre algorithme, C-Évolutif-MC, qui nécessite moins d'itérations que SE-META pour converger vers l'optimum de chacun des problèmes, PT-3-D, PT-3-F et PT-3-SP. C-Évolutif-MC offre un gain au niveau de temps CPU (CPU-simulation optimale).

Les résultats du **Tableau 18** montrent que SE-META permet d'obtenir la même évolution des pourcentages de taux de réduction pour les trois problèmes ($\tau_1 < \tau_4 < \tau_2 < \tau_3$). Cette évolution n'est pas la même pour les pourcentages de taux de réduction obtenus avec C-Évolutif-MC. Pour le problème PT-3-D, on a ($\tau_2 < \tau_4 < \tau_1 < \tau_3$) et pour les deux autres problèmes, l'évolution est monotone ($\tau_1 < \tau_2 < \tau_3 < \tau_4$).

IV.6.4.1.3 Tréfilage à quatre paramètres

Le procédé de tréfilage à quatre paramètres consiste à réduire la section d'un fil de rayon initial 9.5 mm par sa traction à travers quatre filières. L'optimisation consiste à chercher les quatre demi angles d'entrée optimaux qui permettent de minimiser une fonction coût donnée (D_{\max} , F_{\max} ou $D_{\max} + 0.1 \times F_{\max}$). La géométrie initiale de chacune des quatre filières est donnée par le **Tableau 19** :

	Filière 1	Filière 2	Filière 3	Filière 4
L	2.884	2.884	2.884	2.884
α	11.7231	11.7231	11.7231	11.7231
R	8.434	7.487	6.646	5.9

Tableau 19- Géométries initiales des quatre filières.

Optimisation de l'endommagement maximal

Ce problème, PT-4-D, est donné par l'équation suivante :

$$PT-4-D: \begin{cases} \text{Minimiser } f(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = D_{\max}(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ 0.7266 \leq \alpha_i \leq 22.5395, i=1, \dots, 4 \end{cases} \quad (IV.18)$$

Optimisation de la force de traction maximale

Ce problème, PT-4-F, est formulé de la manière suivante :

$$\text{PT-4-F: } \begin{cases} \text{Minimiser } f(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = F_{\max}(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ 0.7266 \leq \alpha_i \leq 22.5395, i=1, \dots, 4 \end{cases} \quad (\text{IV.19})$$

Optimisation de la somme pondérée de l'endommagement et de la force de traction

Ce problème, PT-4-SP, est donné sous la forme suivante :

$$\text{PT-4-SP: } \begin{cases} \text{Minimiser } f(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = D_{\max} + 0.1 \times F_{\max} \\ 0.7266 \leq \alpha_i \leq 22.5395, i=1, \dots, 4 \end{cases} \quad (\text{IV.20})$$

		α_1	α_2	α_3	α_4	f
PT-4-D	SE-META	10.5649	14.3818	8.0599	2.4089	0.8808
	C-Évolutif-MC	1.2023	1.4694	1.7890	1.4889	0.3401
PT-4-F	SE-META	4.5715	6.7290	9.9405	10.6793	7.9130
	C-Évolutif-MC	4.3154	3.4525	1.2781	9.6300	7.904
PT-4-SP	SE-META	10.1699	13.2006	7.6269	2.9156	1.8123
	C-Évolutif-MC	2.6572	1.8058	1.3244	1.8266	1.1753

Tableau 20- Comparaison entre les solutions obtenues avec C-Évolutif-MC et SE-META pour les problèmes PT-4-D, PT-4-F et PT-4-SP

		Numéro de la Simulation optimale	CPU- Simulation optimale	λ_{\max}	CPU-total
PT-4-D	SE-META	77	8h5mn44s	80	8h24mn40s
	C-Évolutif-MC	35	3h13mn40s		7h22mn40s
PT-4-F	SE-META	68	7h59mn12s		9h23mn46s
	C-Évolutif-MC	71	7h26mn7s		8h22mn40s
PT-4-SP	SE-META	80	8h28mn26s		8h28mn26s
	C-Évolutif-MC	21	1h42mn1s		6h28mn40s

Tableau 21- Comparaison entre les temps CPU de C-Évolutif-MC et SE-META pour les problèmes PT-4-D, PT-4-F et PT-4-SP

Pour les trois problèmes d'optimisation mono-objectif de procédé de tréfilage à quatre paramètres, PT-4-D, PT-4-F et PT-4-SP, notre algorithme, C-Évolutif-MC, donne des solutions meilleures que celles obtenues avec SE-META (Tableau 20) et avec un gain important au niveau de temps CPU (CPU- Simulation Optimale) (Tableau 21)

IV.6.4.2 Optimisation multi-objectifs du procédé de tréfilage

La résolution d'un problème d'optimisation multi-objectifs comme un problème mono-objectif utilisant la méthode de la somme pondérée des fonctions coût rencontre plusieurs difficultés. La première concerne le choix des coefficients de pondération (fonctions poids). La deuxième difficulté concerne la performance de la solution optimale, qui est une solution

parmi un ensemble de solutions qui peut être obtenu par la résolution du vrai problème d'optimisation multi-objectifs.

IV.6.4.2.1 Tréfilage à deux paramètres

On utilise le même projet (procédé de tréfilage) décrit dans la partie IV.6.4.1.1. Dans cette partie, on résout un problème d'optimisation multi-objectifs (PT-2). Ce problème consiste à déterminer les valeurs optimales, de la longueur de la portée L et de demi-angle d'entrée α , qui permettent de minimiser les deux fonctions coût, l'endommagement maximal D_{max} et la force de traction maximale F_{max} . Il est formulé de la manière suivante :

$$PT-2: \begin{cases} \text{Minimiser } D_{max}(L, \alpha) \\ \text{Minimiser } F_{max}(L, \alpha) \\ 0.2884 \leq L \leq 5.7682 \\ 1.1885 \leq \alpha \leq 22.5346 \end{cases} \quad (IV.21)$$

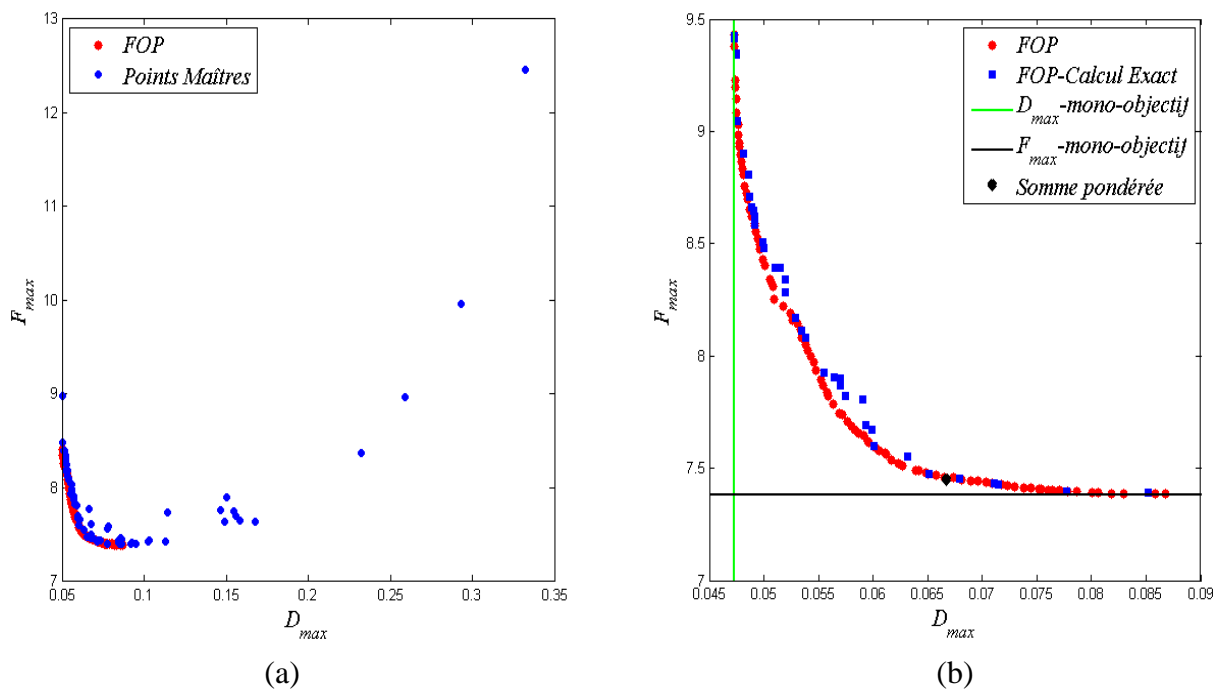


Figure 71- (a) convergence de calculs exacts vers le FOP du PT-2, et (b) représentation des FOP approchés (FOP) et exact (FOP-calcul exact) obtenus avec C-Évolutif-MC

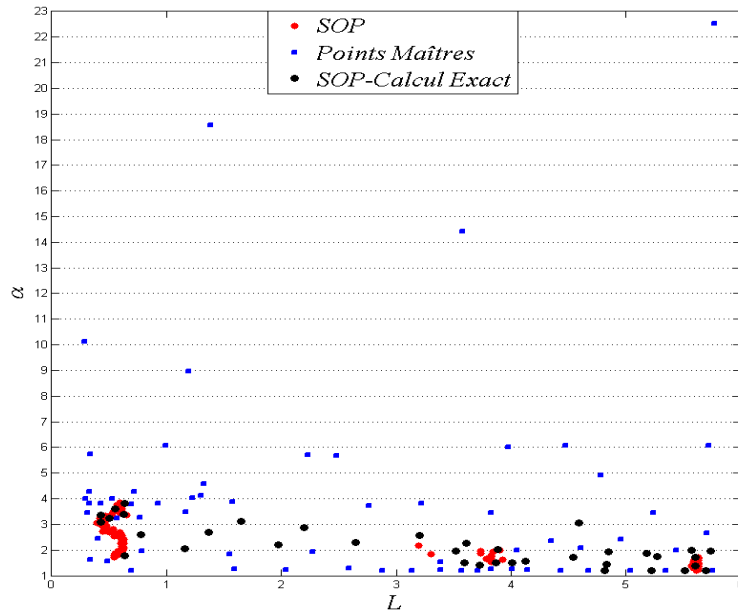


Figure 72- Répartition des solutions optimales du problème PT-2 obtenues avec C-Évolutif-MC dans l'espace de recherche

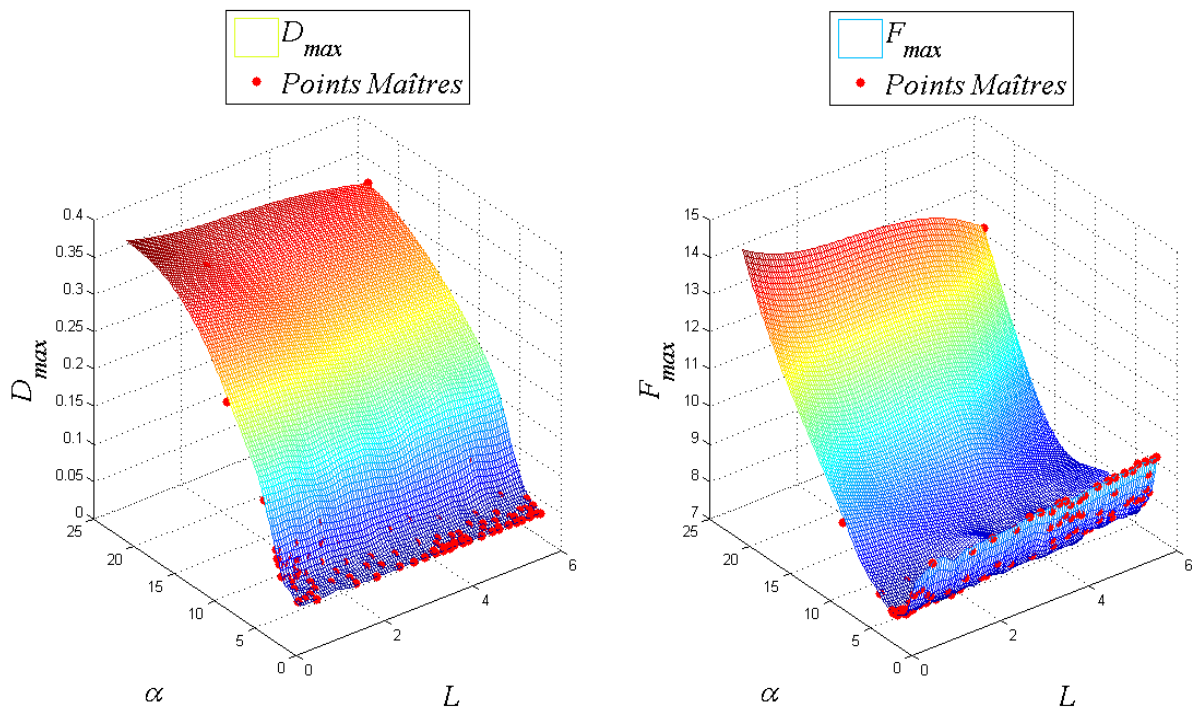


Figure 73- Approximation des fonctions coût, D_{max} et F_{max} du problème PT-2 avec le métamodèle M-MDFSM

IV.6.4.2.2 Tréfilage à quatre paramètres

On utilise le même projet (procédé de tréfilage) décrit dans la partie IV.6.4.1.3. Dans cette partie, on résout un problème d'optimisation multi-objectifs (PT-4). Ce problème consiste à déterminer les valeurs optimales, des demi-angles d'entrée α_i , $i=1,2,3,4$, qui permettent de

minimiser les deux fonctions coût, l'endommagement maximal D_{max} et la force de traction maximale F_{max} . Il est formulé de la manière suivante :

$$PT-4: \begin{cases} \text{Minimiser } D_{max}(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \text{Minimiser } F_{max}(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ 0.7266 \leq \alpha_i \leq 22.5395, i=1, \dots, 4 \end{cases} \quad (IV.22)$$

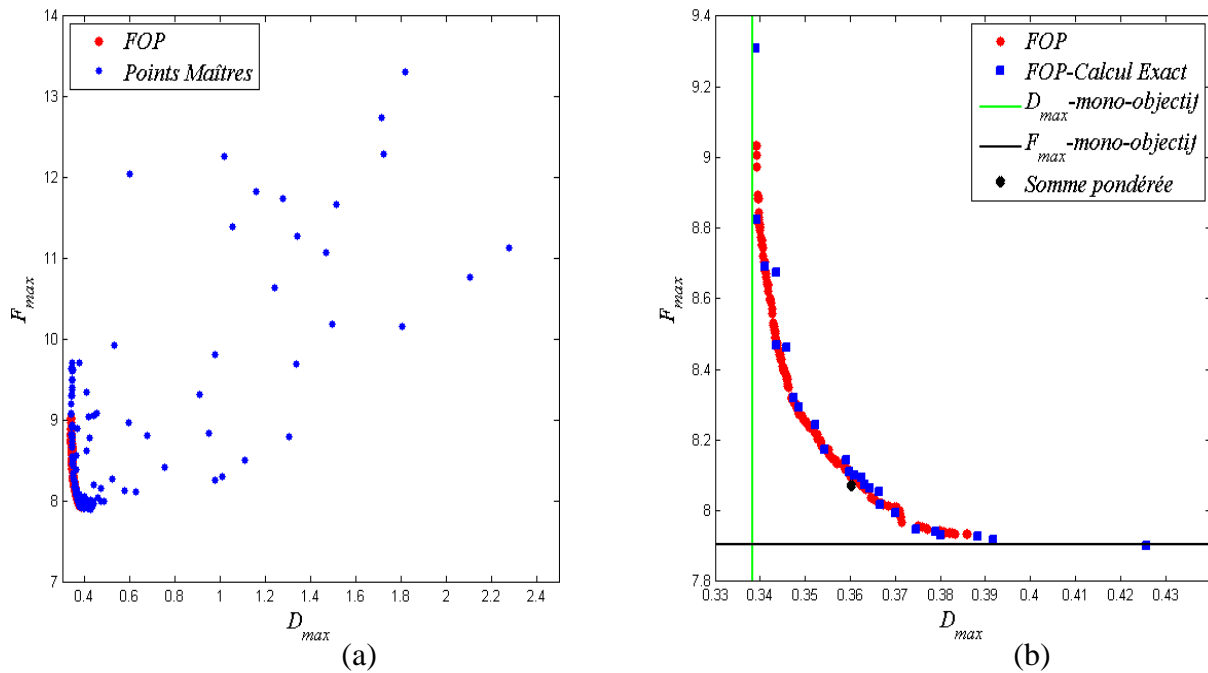


Figure 74- (a) convergence de calculs exacts vers le FOP du PT-4, et (b) représentation des FOP approchés (FOP) et exact (FOP-calcul exact) obtenus avec C-Évolutif-MC

La résolution des deux problèmes d'optimisation multi-objectifs de tréfilage, à deux paramètres (PT-2) avec un temps CPU=3h46mn30s et à quatre paramètres (PT-4) avec un temps CPU=7h44mn37s, en utilisant C-Évolutif-MC, permet de retrouver les solutions mono-objectif obtenues précédemment. On constate que les valeurs minimales, de la force de traction maximale F_{max} et de l'endommagement D_{max} , de la Figure 71-(b) correspondent respectivement aux solutions optimales des deux problèmes mono-objectif PT-2-F et PT-2-D et ceux de la Figure 74-(b) correspondent aux solutions des problèmes PT-4-F et PT-4-D. Les solutions du problème PT-2 (Figure 72 et Figure 73) confirment l'influence secondaire de la longueur de portée par rapport au demi-angle d'entrée de la filière.

Cette résolution nous permet d'obtenir un ensemble de solutions optimales pour chacun des deux problèmes, PT-2 et PT-4 où la solution optimale de chacun des problèmes mono-objectif, PT-2-SP et PT-4-SP est une solution parmi un ensemble optimal de solutions (Figure 71-(b) et Figure 74-(b)). Donc l'optimisation multi-objectifs offre à l'utilisateur plusieurs choix possibles. La Figure 71-(b) et la Figure 74-(b) permettent à l'utilisateur de choisir s'il préfère donner de l'importance soit à l'endommagement, D_{max} , soit à la force de

tréfilage, F_{\max} . En tout cas ces courbes montrent qu'il est possible de diminuer fortement l'endommagement sans trop augmenter la force de tréfilage.

Les résultats obtenus pour les problèmes PT-2 et PT-4 confirment la robustesse et l'efficacité de notre algorithme pour la résolution des problèmes d'optimisation multi-objectifs en mise en forme. Cette performance peut s'expliquer par la convergence vers l'ensemble optimal de Pareto au fur et à mesure des itérations (Figure 71-(a) et Figure 74-(a)). Cette convergence est due à un métamodèle, M-MDFSM, précis (Figure 73). Cette précision est due à une bonne stratégie d'enrichissement par l'ajout des nouveaux points maîtres avec un potentiel de bonne performance. Cet ajout est dû à l'utilisation de la fonction mérite, qui permet d'étendre la recherche dans le reste des domaines de recherche inexplorés, en tenant compte de l'erreur d'approximation locale.

IV.7 CONCLUSION

Dans ce chapitre, on a validé la robustesse et l'efficacité de notre algorithme, C-Évolutif-MC sur des vrais problèmes d'optimisation en mise en forme. Au début on a commencé par la résolution du problème 3D d'optimisation mono-objectif de remplissage d'une bielle (PB1 et PB2), proposé par notre partenaire industriel, Atelier des Janves. Ensuite, on a traité le problème 2D d'optimisation mono-objectif de remplissage d'une couronne (PCr1 et PCr2), proposé par Transvalor. Enfin, le reste de ce chapitre a été consacré à l'optimisation du tréfilage, qui est proposée par ArcelorMittal. Ce procédé de tréfilage est étudié sous forme de plusieurs problèmes d'optimisation mono-objectif, qui diffèrent par leur nombre de paramètres et par la fonction coût utilisée, tels que PT-2-D, PT-2-F, PT-2-SP, PT-3-D, PT-3-F, PT-3-SP, PT-4-D, PT-4-F et PT-4-SP et sous forme de deux problèmes d'optimisation multi-objectifs, PT-2 et PT-4.

Pour les problèmes d'optimisation mono-objectif, les résultats obtenus avec C-Évolutif-MC sont comparés avec ceux obtenus avec l'algorithme préalablement intégré dans Forge2009, SE-META. Ces comparaisons montrent l'apport de notre algorithme qui donne des meilleurs résultats que SE-META pour tous les problèmes et avec un temps de calcul CPU moins élevé.

La résolution des deux problèmes d'optimisation multi-objectifs de tréfilage, PT-2 et PT-4, avec C-Évolutif-MC permet de donner les fronts de Pareto correspondants. Contrairement à la solution optimale de PT-2-SP et celle de PT-4-SP, qui sont deux choix uniques, les fronts obtenus donnent différents choix possibles d'une solution optimale. Ces fronts optimaux de Pareto confirment et précisent les points obtenus précédemment (solutions optimales, PT-2-D, PT-2-F, PT-2-SP, PT-4-D, PT-4-F et PT-4-SP). Ils montrent qu'il est possible de diminuer fortement l'endommagement sans trop augmenter la force de tréfilage.

CONCLUSION GENERALE

L'objectif de ce travail de thèse était de développer un algorithme d'optimisation multi-objectifs qui permet la résolution de problèmes d'optimisation en mise en forme, mono-objectif et multi-objectifs, avec un coût de calcul raisonnable. Plusieurs aspects ont été abordés pendant cette étude. On a commencé par la présentation de l'optimisation multi-objectifs, d'algorithmes évolutionnaires, en particulier l'algorithme mono-objectif, SE-META et l'algorithme génétique multi-objectifs, NSGA-II et de trois métamodèles, la Méthodologie des Surfaces de Réponse, le Krigeage et les Fonctions à Base Radiales. Dans le premier chapitre, on a vu que la meilleure méthode (ou la plus répandue) pour la résolution d'un problème d'optimisation réel, généralement multi-objectifs, avec un coût de calcul réduit est l'utilisation d'un algorithme évolutionnaire multi-objectifs couplé à un métamodèle. Les métamodèles les plus connus et les plus utilisés sont la Méthodologie des Surfaces de Réponse, le Krigeage, les Moindres Carrés Mobiles et les Fonctions à Base Radiales. L'utilisation de chacun de ces métamodèles se heurte à des difficultés. Pour la Méthodologie des Surfaces de Réponse, le problème concerne l'ordre polynomial à utiliser. Pour la méthode des Moindres Carrés Mobiles, l'utilisation d'un rayon d'influence dans la définition des fonctions « fenêtres », de manière à éliminer l'influence des points éloignés, rend la détermination de ce rayon délicate dans les cas les plus généraux. L'inconvénient du krigeage est lié à la complexité du modèle (méthode coûteuse en terme de temps de calcul) et à la répartition de points maîtres (si les points maîtres sont proches, la matrice de corrélation peut devenir singulière). Enfin, la précision de la méthode des Fonctions à Base Radiales est inconnue et elle est liée au choix de la fonction radiale.

Pour surmonter ces problèmes et dans le but d'avoir un métamodèle précis, on a développé un métamodèle basé sur la Méthode de Différences Finies Sans Maillage (**Chapitre II**). Son avantage majeur est la résolution des difficultés introduites par les autres méthodes et de donner lieu à des temps de calcul réduits. Au début, on a commencé par la construction du métamodèle d'ordre un, M-MDFSM-1 (un développement en série de Taylor à l'ordre un). L'inconvénient de ce dernier est le manque de précision des approximations. Ce manque de précision est dû à la nature des interpolations, qui sont des interpolations linéaires. Pour améliorer la précision de notre métamodèle, on a introduit un développement en série de Taylor à l'ordre deux. Le métamodèle d'ordre deux, M-MDFSM-2, a augmenté la précision des résultats mais il rencontre un problème de singularité de sa matrice \mathbf{A} dû au manque d'information, comme le non respect du nombre minimal de points maîtres et la mauvaise dispersion de ces derniers dans l'espace de recherche. Pour surmonter cette faiblesse et obtenir un modèle précis, on a combiné les deux métamodèles précédents pour aboutir à un métamodèle régularisé, M-MDFSM. Ce nouveau métamodèle est un modèle quadratique en présence des informations nécessaires. Dans le cas contraire (manque d'informations), il devient un modèle linéaire. Ensuite ce métamodèle régularisé a été validé sur deux fonctions analytiques (Camel Back et bi-extrema). Après la construction et la validation de notre métamodèle, on a cherché les meilleures formulations de l'erreur d'approximation locale liée à ce dernier, l'erreur MC et H1, parmi quatre types d'erreur testées. Enfin, on a développé une méthode pour la prise en compte de bruit par notre métamodèle. Cette méthode a été validée sur des fonctions analytiques.

Pour aboutir à un algorithme robuste et efficace pour la résolution des problèmes d'optimisation multi-objectifs en mise en forme, qui sont très coûteux en terme de temps de calcul, on a étudié dans le **Chapitre III** différents types de couplages entre NSGA-II et notre métamodèle, M-MDFSM tels que C-Constant, C-Actualisé, C-Évolutif-H1 et C-Évolutif-MC. Ces couplages diffèrent par le choix des points maîtres au cours des itérations, par l'évolution du métamodèle et par le type d'erreur d'approximation locale utilisée. Ces algorithmes sont ensuite évalués sur plusieurs fonctions tests analytiques. On trouve ceux sans contraintes ayant un front optimal de Pareto convexe (Min-Ex et SCH1), concave (Max-Ex), discontinu (SCH2, Pol, VNT et KUR) et ceux sous contraintes comme Const-Min-Ex et TNK.

Au début, les quatre algorithmes sont comparés sur des problèmes tests mono-objectif (Camel-Back, Griewank, Rastrigin et Rosenbrock) et des POMOs moins compliqués (Min-Ex, Max-Ex, SCH1, SCH2 et Const-Min-Ex). Les résultats obtenus montrent que C-Évolutif-MC est plus performant que les autres algorithmes proposés. Ensuite, C-Évolutif-MC est comparé à NSGA-II sur des POMOs très compliqués ayant des fronts et des ensembles optimaux de Pareto non linéaires et discontinus tels que POL, VNT, TNK et KUR. Les résultats obtenus confirment l'efficacité de notre algorithme, C-Évolutif-MC, qui permet de retrouver des résultats aussi précis que NSGA-II pour tous les problèmes proposés et avec un gain considérable en terme de temps de calcul.

Après sa validation sur des problèmes d'optimisation tests analytiques, notre algorithme, C-Évolutif-MC, est intégré dans le logiciel EF, Forge2009®, pour la résolution des problèmes d'optimisation réels proposés par nos partenaires industriels. Donc les points maîtres utilisés pour construire notre métamodèle sont des simulations EF très coûteuses en terme de temps de calcul (**Chapitre IV**). Dans ce chapitre, on a traité deux types de problèmes d'optimisation en mise en forme. Le premier est un problème mono-objectif, tels que le cas de tréfilage proposé par ArcelorMittal, de remplissage d'une couronne proposé par Transvalor et de remplissage d'une bielle proposé par Atelier des Janves. Pour ces problèmes d'optimisation mono-objectif, l'apport de notre algorithme, C-Évolutif-MC, est évalué par des comparaisons avec l'algorithme préalablement intégré dans Forge2009®, SE-META. Le deuxième est un problème multi-objectifs, qui traite deux problèmes multi-objectifs du cas de tréfilage.

Pour les problèmes d'optimisation mono-objectif, les résultats obtenus avec C-Évolutif-MC sont comparés avec ceux obtenus avec l'algorithme préalablement intégré dans Forge2009, SE-META. Ces comparaisons montrent l'apport de notre algorithme qui donne des meilleurs résultats que SE-META pour tous les problèmes et avec un temps de calcul CPU moins élevé.

La résolution des deux problèmes d'optimisation multi-objectifs de tréfilage permet d'obtenir les fronts de Pareto correspondants. Contrairement à la solution optimale d'un problème mono-objectif, qui est un choix unique, les fronts obtenus donnent différent choix possibles d'une solution optimale.

A l'avenir, il serait intéressant de tester l'algorithme C-Évolutif-MC sur des autres problèmes d'optimisation multi-objectifs en mise en forme. Il serait aussi intéressant de tester la possibilité d'ajouter autant de points maîtres à chaque génération que de points utilisés pour l'initialisation de notre métamodèle ($\lambda=(n+1)(n+2)/2$) et de lancer ces simulations sur λ processeurs.

Un autre axe de recherche pourrait être l'étude de nouveaux plans d'expérience pour initialiser le métamodèle. On peut aussi étudier des nouvelles formulations de l'erreur d'approximation locale.

BIBLIOGRAPHIE

- [Abderrahmane and Regis, 2009] Abderrahmane, B., and D. Regis, "Review and Analysis of some Metamodeling Techniques used in Optimization", Institut National de Recherche en Informatique et en Automatique (INRIA), 2009.
- [Abedrabbo et al, 2004] Abedrabbo, N., N. Zafar, R. Averill, F. Pourboghrat, and R. Sidhu, "Optimization of a Tube Hydroforming Process", In: *Proceedings of NUMIFORM*, Columbus, OH, 13-17 June 2004.
- [Ahmed and Qin, 2009] Ahmed, M. Y. M., and N. Qin, "Comparison of Response Surface and Kriging Surrogates in Aerodynamic Design Optimization of Hypersonic Spiked Blunt Bodies", *13th International Conference on AEROSPACE SCIENCES & AVIATION TECHNOLOGY, ASAT- 13*, 26 – 28 May 2009.
- [Batmaz and Tunali, 2003] Batmaz, I., and S. Tunali, "Small response surface designs for metamodel estimation", *European Journal of Operational Research* 145, pp.455–470, 2003.
- [Ben Ayed et al, 2005] Ben Ayed, L., A. Delamézière, J. L. Batoz, et C. KnopfLenoir, "Contrôle de la striction et du plissement en emboutissage par optimisation des efforts serre-flan", *Septième Colloque National En Calcul Des Structures*, Giens, 2005.
- [Beume et al, 2007] Beume, N., B. Naujoks and M. Emmerich, "SMS-EMOA: Multi-objective selection based on dominated hypervolume", *European Journal Research* 18, pp.1653-1669, 2007.
- [Beyer and Schwefel, 2002] Beyer, H. G., and H. P. Schwefel, "Evolution strategies: A comprehensive introduction", *Natural Computing* 1, pp.3–52, 2002.
- [Bonte, 2007] Bonte, M. H. A., "Optimisation Strategies for Metal Forming Processes", thèse de doctorat, University of Twente, 2007.
- [Bonte et al, 2007] Bonte, M. H. A., A. H. Van den Boogaard, and J. Huétink, "A metamodel based optimisation algorithm for metal forming processes", In: Banabic D, eds., *Advanced methods in material forming*, Springer Verlag, Heidelberg, Germany, pp.55–72, 2007.
- [Bonte et al, 2008] Bonte, M. H. A., A. H. Van den Boogaard, and J. Huétink, "An optimisation strategy for industrial metal forming processes : Modelling, screening and solving of optimisation problems in metal forming", *Structural and Multidisciplinary optimization* 35, pp.571–586, 2008.
- [Bonte et al, 2010] Bonte, M. H. A., L. Fourment, T. T. Do, A. H. Van den Boogaard, and J. Huetink, "Optimisation of metal forming processes using Finite Element simulations: A sequential approximate optimization algorithm and its comparison to other algorithms by application to forging", *Structural and Multidisciplinary optimization* 42, pp.797-810, 2010.
- [Breitkopf et al, 2005] Breitkopf, P., H. Naceur, A. Rassineux, and P. Villon, "Moving Least squares response surface approximation: Formulation and metal forming applications", *Computers and structures* 83, pp.1411-1428, 2005.
- [Büche, 2003] Büche, D., "Multi-Objectives Evolutionary Optimization of Gas Turbine Components", thèse de doctorat, Swiss Federal Institute of Technology, Zürich, 2003.

- [Büche et al, 2005] Büche, D., N. N. Schraudolph, and P. Koumoutsak, "Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models", *IEEE Transactions on Systems, Man, Cybern*, pp.184-194, 2005.
- [Cao et al, 2001] Cao, J., S. Li, Z. Xia, and S. Tang, "Analysis of an axisymmetric deep-drawn part forming using reduced forming steps", *Journal of Materials Processing Technology* 117, pp.193–200, 2001.
- [Carr et al, 1997] Carr, J. C., W. R. Fright, and R. K. Beaston, "Surface Interpolation With Radial Basis Functions for Medical Imaging", *IEEE Transactions on Medical Imaging*, pp.96-107, 1997.
- [Castro et al, 2004] Castro, C., C. António, and L. Sousa, "Optimisation of shape and process parameters in metal forging using genetic algorithms", *Journal of Materials Processing Technology* 146, pp.356–364, 2004.
- [Chambers and Mount-Campbell, 2002] Chambers, M., and C. A. Mount-Campbell, "Process optimization via neural network metamodeling", *International Journal of Production Economics* 79, pp.93-100, 2002.
- [De Jong, 1975] De Jong, K., "The analysis of the behaviour of class of genetic adaptative systems", thèse de doctorat, Department of computer Science, University of Michigan, Ann Arbor, Michigan, 1975.
- [De Jong and Spears, 1992] De Jong, K., and W. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms", *Annals of Mathematics and Artificial Intelligence* 5, pp.1-26, 1992.
- [Deb, 1999] Deb, K., "Multi-objective genetic algorithms: Problem difficulties and construction of test problems", *Evolutionary Computation Journal* 7, pp.205-230, 1999.
- [Deb, 2001] Deb, K., "*Multi-Objective Optimisation using Evolutionary Algorithms*", Edited by John Wiley & Sons, Chichester, 2001.
- [Deb, 2004] Deb, K., "Single and Multi-Objective Optimization Using Evolutionary Algorithms", Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL), 2004.
- [Deb et al, 2001] Deb, K., A. Pratap, and T. Meyarivan, "Constrained Test Problems for Multi-objective Evolutionary Optimization", In: *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001)*, Berlin, Germany, pp.284-298, 2001.
- [Deb et al, 2000a] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II", Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL), 2000.
- [Deb et al, 2000b] Deb, K., S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", In: *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI)*, Berlin, pp.849–858, 2000.
- [Deb et al, 2002a] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, pp.182–197, 2002.
- [Deb et al, 2002b] Deb, K., L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-

Objective Optimization Test Problems", *Congress on Evolutionary Computation*, Piscataway, New Jersey, pp.825–830, 2002.

[Deb and Agrawal, 1999] Deb, K., and S. Agrawal, "Understanding Interactions among Genetic Algorithm Parameters", Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL), 1999.

[Deb et al, 2005] Deb, K., L.Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization", *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, US, pp.105-145, 2005.

[Do, 2006] Do, T T., "Optimisation de forme en forgeage 3D", thèse de doctorat, Mines ParisTech, Sophia-Antipolis, 2006.

[Ejday and Fourment, 2010a] Ejday, M., and L. Fourment, "Metamodel assisted evolutionary algorithm for multi-objective optimization of non-steady metal forming problems", *International Journal of Material Forming* 3, pp.5-8., 2010.

[Ejday and Fourment, 2010b] Ejday, M., and L. Fourment, "Optimisation Multi-Objectifs à base de Métamodèle pour des applications en Mise en Forme des Métaux", *Mécanique & Industries*, 2010.

[Ejday and Fourment, 2009a] Ejday, M., and L. Fourment, "Meta-model assisted Multi-Objective Optimization for non-steady 3D metal forming processes", *International Journal of Material Forming* 2, pp.335-338, 2009.

[Ejday and Fourment, 2009b] Ejday, M., and L. Fourment, "Meta-model assisted Multi-Objective Optimization – Application to nonsteady 3D metal forming processes", *8th World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, 2009.

[Ejday and Fourment, 2009c] Ejday, M., and L. Fourment, "Meta-model assisted Multi-Objective Optimization for non-steady 3D metal forming applications", *7th EUROMECH Solid Mechanics Conference*, Lisbon, Portugal, 2009.

[Ejday and Fourment, 2009d] Ejday, M., and L. Fourment, "Optimisation multi-objectifs à base de métamodèle pour les problèmes de mise en forme", *9e colloque national en calcul des structures*, Giens, France, 2009.

[Emmerich et al, 2002] Emmerich, M., A. Giotis, M. Ozdemir, T. Back, and K. Giannakoglou, "Metamodel assisted evolution strategies", *International Conference on parallel problem solving from nature*, Berlin, Germany, 2002.

[Emmerich and. Naujoks, 2004] Emmerich, M., and B. Naujoks, "Metamodel assisted multiobjective optimisation strategies and their application in airfoil design", In: Parmee, I.C., eds., *Adaptive Computing in Design and Manufacture VI*, Springer, London, pp.249–260, 2004.

[Emmerich et al, 2006] Emmerich, M., K. Giannakoglou and B. Naujoks, "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels", *IEEE Transactions on Evolutionary Computation*, pp.421-439, 2006.

[Emmerich et al, 2005] Emmerich, M., N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion", *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Berlin, pp.62–76, 2005.

[Fonseca et al, 2003] Fonseca, D. J., D. O. Navesse, and G. P. Moynihan, "Simulation metamodeling through artificial neural networks", *Engineering Applications of Artificial*

Intelligence 16, pp.177-183, 2003.

[Fonseca and. Fleming, 1995] Fonseca, M. C., and P. J. Fleming, "Multi-objective genetic algorithms made easy: Selection, sharing and mating restrictions", *In: Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Application*, pp.45-52, 1995.

[Fourment, 2007] Fourment, F., "Gradient, Non-Gradient and Hybrid Algorithms for Optimizing 3D Forging Sequences with Uncertainties", *Proceedings of the 9th International Conference on Numerical Methods in Industrial Forming Processes*, pp.475-480, 2007.

[Fourment and Chenot, 1996] Fourment, L., and J. L. Chenot, "Optimal design for non-steady-state metal forming processes-I. Shape optimization method", *International journal for numerical methods in engineering* 39, pp.33-50, 1996.

[Fourment et al, 1996] Fourment, L., T. Balan, and J. L. Chenot, "Optimal design for non-steady-state metal forming processes-II. Application of shape optimization in forging", *International journal for numerical methods in engineering* 39, pp.51-65, 1996.

[Fourment et al, 2009] Fourment, L., T. Massé, S. Marie, R. Ducloux, M. Ejday, C. Bobadilla, and P. Montmitonnet, "Optimisation of a range of 2D and 3D bulk forming processes by a meta-model assisted evolution strategy", *International Journal of Material Forming* 2, pp.343-346, 2009.

[Fourment et al, 2010] Fourment, L., R. Ducloux, S. Marie, M. Ejday, D. Monnereau, T. Massé, and P. Montmitonnet, "Mono and multi-objective optimization techniques applied to a large range of industrial test cases using Metamodel assisted Evolutionary Algorithms", *In: Proceedings of the 10th International Conference on Numerical Methods in Industrial Forming Processes*, pp.833-840, 2010.

[Ghanmi et al, 2005] Ghanmi, S., M. L. Bouazizi, et N. Bouhaddi, "Optimisation multi-objectifs en mécanique des structures : Approche basée sur les surfaces de réponse adaptative", *7ème colloque national en calcul des structures*, Giens, France, 2005.

[Gildemyn, 2008] Gildemyn, E., "Caractérisation des procédés de fabrication de pièces de sécurité automobile. Optimisation multiobjectifs de la mise en forme", thèse de doctorat, Ecole Nationale Supérieure d'Arts et Métiers, 2008.

[Goal et al, 2007] Goal, T., R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, "Response surface approximation of Pareto optimal front in multi-objective optimization", *Comput. Methods Appl. Mech. Engrg* 196, pp.879-893, 2007.

[Goal et al, 2004] Goel, T., R. Vaidyanathan, R. T. Haftka, N. V. Queipo, W. Shyy, and K. Tucker, "Response surface approximation of Pareto optimal front in multi-objective optimization", *In: Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, 2004.

[Goldberg, 1989] Goldberg, D E., "*Genetic Algorithms in Search, Optimization, and Machine Learning*", Addison-Wesley Publishing company., 1989.

[Goldberg and Holland, 1988] Goldberg, D. E., and J. H. Holland, "Genetic Algorithms and Machine Learning", *Machine Learning* 3, pp.95-99, 1988.

[Golub and Van Loan, 1989] Golub, G. H., and C. F. Van Loan, "Matrix Computations", Johns Hopkins University Press, Baltimore, 1989.

[Holland, 1962] Holland, J. H., "Outline for a Logical Theory of Adaptive Systems", *Journal*

of the association of computing machinery 3, pp.297-314, 1962.

[Ingarao et al, 2009] Ingarao, G., R. D. Lorenzo, and F. Micari, "Internal pressure and counter-punch action design in Y-shaped tube hydroforming processes: A multi-objective optimisation approach", *Computers and Structures* 87, pp.591-602, 2009.

[Jensen et al, 1998] Jensen, M. R., F. F. Damborg, K. B. Nielsen, and J. Danckert, "Optimization of the draw-die design in conventional deep-drawing in order to minimise tool wear", *Journal of Materials Processing Technology* 83, pp.106–114, 1998.

[Jin, 2005] Jin, Y., "A comprehensive Survey of Fitness Approximation in Evolutionary Computation", *Soft Computing* 9, pp.3-12, 2005.

[Jones, 2001] Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces", *Journal of Global Optimization* 21, pp.345-383, 2001.

[Jones et al, 1998] Jones, D. R., M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions", *Journal of Global Optimization* 13, pp.455–492, 1998.

[Jourdan, 2009] Jourdan, A., "How to repair a second-order surface for computer experiments by Kriging", *Chemometrics and Intelligent Laboratory Systems* 96, pp.108-116, 2009.

[Jurkovic et al, 2006] Jurkovic, Z., M. Jurkovic, and S. Buljan, "Optimization of extrusion force prediction model using different techniques", *Journal of Achievements in Materials and Manufacturing Engineering* 17, pp.353-356, 2006.

[Justesen, 2009] Justesen, P. D., "Multi-objective Optimization using Evolutionary Algorithms", University of Aarhus, Department of Computer Science, Denmark, 2009.

[Kadkhodaei et al, 2007] Kadkhodaei, M., M. Salimi, and M. Poursina, "Analysis of asymmetrical sheet rolling by a genetic algorithm", *International Journal of Mechanical Sciences* 49, pp.622–634, 2007.

[Kim et al, 2001] Kim, N., K. Choi, and J. Chen, "Die shape design optimization of sheet metal stamping process using meshfree method", *International Journal for Numerical Methods in Engineering* 51, pp.1385–1405, 2001.

[Kleinermann and Ponthot, 2003] Kleinermann, J. P., and J. P. Ponthot, "Parameter identification and shape/process optimization in metal forming simulation", *Journal of Materials Processing Technology* 139, pp.521–526, 2003.

[Knowles and Hughes, 2005] Knowles, J., and E. J. Hughes, "Multiobjective Optimization on a Budget of 250 Evaluations", *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pp.176-190, 2005.

[Krige, 1951] Krige, D. G., "A statistical approach to some basic mine valuation problems on the Witwatersrand", *Journal of the Chemical, Metallurgical and Mining Society of the South Africa* 52, pp.119-139, 1951.

[Krok, 2002] Krok, J., "An Extended Approach to Error Control in Experimental and Numerical Data Smoothing and Evaluation Using the Meshless FDM", *Meshfree Computational Mechanics* 11, pp.913-945, 2002.

[Kursawe, 1990] Kursawe, F., "A variant of evolution strategies for vector optimization", *In: Proceedings of the Parallel Problem Solving from Nature I (PPSN-I)*, pp.193-197, 1990.

- [Ledoux et al, 2010] Ledoux, Y., P. Sébastian, and S. Samper, "Optimization method for stamping tools under reliability constraints using genetic algorithms and finite element simulations", *Journal of Materials Processing Technology* 210, pp.474–486, 2010.
- [Lepadatu, 2006] Lepadatu, D., "Optimisation Des Procédés De Mise En Forme Par Approche Couplée Plans D'expériences, Éléments Finis Et Surface De Réponse", thèse de doctorat, Institut des Sciences et Techniques de l'Ingénieur d'Angers, 2006.
- [Le Riche et al, 2007] Le Riche, R., M. Schoenauer and M. Sebag, "Un état des lieux de l'optimisation évolutionnaire et de ses implications en sciences pour l'ingénieur", *chapter in Modélisation Numérique: défis et perspectives, 2nd volume, Traité Mécanique et Ingénierie des Matériaux*, P. Breitkopf and C. Knopf-Lenoir eds., Hermes publ., pp. 187-259, 2007
- [Le Riche et al, 2009] Le Riche, R., V. Picheny, D. Ginsbourger, A. Meyer and N. H. Kim, "Gears design with shape uncertainties using Monte Carlo simulations and kriging", *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, USA, May 4-7, 2009*, paper No. AIAA-2009-2257
- [Li et al, 2008] Li, R., M. T. M. Emmerich, J. Eggermont, E. G. P. Bovenkamp, T. Bäck, J. Dijkstra, and J. H. C. Reiber, "Metamodel-Assisted Mixed Integer Evolution Strategies and Their Application to Intravascular Ultrasound Image Analysis", *IEEE Congress on Evolutionary Computation (CEC 2008)*, pp.2764-2771, 2008.
- [Lin et al, 2003] Lin, Z., X. Juchen, W. Xinyun, and H. Guoan, "Optimization of die profile for improving die life in the hot extrusion process", *Journal of Materials Processing Technology* 142, pp.659–664, 2003.
- [Liszka and Orkisz, 1980] Liszka, T., and J. Orkisz, "The finite difference method at arbitrary irregular grids and its application in applied mechanics", *Computers and Structures* 11, pp.83-95, 1980.
- [Massé, 2010] Massé, T., "Study and Optimization of High Carbon Steel Flat Wires", thèse de doctorat, Mines ParisTech, Sophia-Antipolis, 2010.
- [Matheron, 1962] Matheron, G., "Traité de géostatistique appliquée", 1962.
- [Cours, 2001] "Méthodes numériques pour la régression linéaire", Cours STAT 2430, 2001.
- [Myers et al, 2009] Myers, R. H., D. C. Montgomery, C. M. Anderson, and C. M. Anderson-Cook, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Edited by John Wiley and Sons, Hoboken, New Jersey, USA, 2009.
- [Naceur et al, 2008] Naceur, H., S. Ben-Elechi, J. L. Batoz, and C. Knopf-Lenoir, "Response surface methodology for the rapid design of aluminum sheet metal forming parameters", *Materials and Design* 29, pp.781–790, 2008.
- [Naceur et al, 2006] Naceur, H., Y. Q. Guo, and S. Ben-Elechi, "Response surface methodology for design of sheet forming parameters to control springback effects", *Computers and Structures* 84, pp.1651–1663, 2006.
- [Naceur et al, 2001] Naceur, H., Y. Q. Guo, J. L. Batoz, and C. Knopf-Lenoir, "Optimization of drawbead restraining forces and drawbead design in sheet metal forming process", *International Journal of Mechanical Sciences* 43, pp.2407–2434, 2001.
- [Naceur et al, 2004] Naceur, H., Y. Q. Guob, and J. L. Batoz, "Blank optimization in sheet metal forming using an evolutionary algorithm", *Journal of Materials Processing Technology* 151, pp.183–191, 2004.

- [Naujoks et al, 2005] Naujoks, B., N. Beume, and M. Emmerich, "Multi-objective Optimization using S-metric Selection: Application to three-dimensional Solution Spaces", *IEEE Congress on Evolutionary Computation (CEC'2005) (IEEE Service Center) 2*, pp.1282–1289, 2005.
- [Oudjene et al, 2009] Oudjene, M., L. Ben-Ayed, A. Delamézière, and J. L. Batoz, "Shape optimization of clinching tools using the response surface methodology with Moving Least-Square approximation", *Journal of materials processing technology* 209, pp.289–296, 2009.
- [Pawan and Deb, 2005] Pawan, K., S. Nain, and K. Deb, "A Multi-Objective Optimization Procedure with Successive Approximate Models", Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL), 2005.
- [Pawan and Deb, 2004] Pawan, K., S. Nain, and K. Deb, "A Multi-Objective Search and Optimization Procedure with Successive Approximate Models", Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL), 2004.
- [Poloni et al, 2000] Poloni, C., A. Giurgevich, L. Onesti, and V. Pedirod, "Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics", *Computer Methods in Applied Mechanics and Engineering* 186, pp.403-420, 2000.
- [Ponthot and Kleinermann, 2006] Ponthot, J. P., and J. P. Kleinermann, "A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation", *Computer Methods in Applied Mechanics and Engineering* 195, pp.5472–5508, 2006.
- [Poursina et al, 2006] Poursina, M., J. Parvizian, and C. A. C. Antonio, "Optimum pre-form dies in two-stage forging", *Journal of Materials Processing Technology* 174, pp.325–333, 2006.
- [Queipo et al, 2005] Queipo, N. V., R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization", *Progress in Aerospace Sciences* 41, pp.1–28, 2005.
- [Raza and Kim, 2008] Raza, W., and K. Y. Kim, "Shape optimization of wire-wrapped fuel assembly using Kriging metamodeling technique", *Nuclear Engineering and Design* 238, pp.1332–1341, 2008.
- [Rechenberg, 1965] Rechenberg, I., "Cybernetic solution path of an experimental problem", Royal Aircraft Establishment, UK, 1965.
- [Rippa, 1999] Rippa, S., "An algorithm for selecting a good value for the parameter c in radial basis function interpolation", *Advances in Computational Mathematics* 11, pp.193–210, 1999.
- [Sacks et al, 1989] Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments (with discussion)", *Statistical Science* 4, pp.409–435, 1989.
- [Schenk and Hillmann, 2004] Schenk, O., and M. Hillmann, "Optimal design of metal forming die surfaces with evolution strategies", *Computers and structures* 82, pp.1695–1705, 2004.
- [Sobester et al, 2005] Sobester, A., S. J. Leary, and A. J. Keane, "On the Design of Optimization Strategies Based on Global Response Surface Approximation Models", *Journal of Global Optimization* 33, pp.31–59, 2005.

- [Spears and De Jong, 1991] Spears, W., and K. De Jong, "On the virtues of parameterised uniform crossover", *Proceedings of the 4th International Conference on Genetic Algorithms*, La Jolla, CA, pp.230-236, 1991.
- [Syswerda, 1989] Syswerda, G., "Uniform crossover in genetic algorithms", *In: Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufman, pp.2-8, 1989.
- [Tanaka et al, 1995] Tanaka, M., H. Watanabe, T. Furukawa, and T. Yanino, "GA-based decision support system for multi-criteria optimization", *In: Proceedings of the International Conference on Systems, Man and Cybernetics*, pp.1556-1561, 1995.
- [Tunali and Batmaz, 2000] Tunali, S., and I. Batmaz, "Dealing with the least squares regression assumptions in simulation metamodeling", *Computers & Industrial Engineering* 38, pp.307–320, 2000.
- [Viennet, 1996] Viennet, R., "Multicriteria optimization using a genetic algorithm for determining a Pareto set", *International Journal Of Systems Science* 27, pp.255-260, 1996.
- [Wei and Yuying, 2008] Wei, L., and Y. Yuying, "Multi-objective optimization of sheet metal forming process using Pareto-based genetic algorithm", *Journal of Materials Processing Technology* 208, pp.499–506, 2008.
- [Wu and Hsu, 2002] Wu, C. Y., and Y. C. Hsu, "Optimal Shape Design of an Extrusion Die Using Polynomial Networks and Genetic Algorithms", *The International Journal of Advanced Manufacturing Technology* 19, pp.79–87, 2002.
- [Zitzler et al, 2000] Zitzler, E., K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation* 2, pp.173-195, 2000.
- [Zitzler et al, 2003] Zitzler, E., M. Laumanns, and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization. Metaheuristics for Multiobjective Optimisation", *Workshop on Multiple Objective Metaheuristics (MOMH 2002)*, Springer-Verlag, Berlin, Germany, pp.3-38, 2003.