



HAL
open science

Méthodes efficaces de capture de front de pareto en conception mécanique multicritère : applications industrielles

Aalae Benki

► **To cite this version:**

Aalae Benki. Méthodes efficaces de capture de front de pareto en conception mécanique multicritère : applications industrielles. Mathématiques générales [math.GM]. Université Nice Sophia Antipolis, 2014. Français. NNT : 2014NICE4000 . tel-00959099

HAL Id: tel-00959099

<https://theses.hal.science/tel-00959099>

Submitted on 13 Mar 2014

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.

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR
SCIENCES
ECOLE DOCTORALE EDSFA

THESE

pour obtenir le titre de

DOCTEUR EN SCIENCES

De l'UNIVERSITE de Nice-Sophia Antipolis

Spécialité « Mathématiques appliquées »

présentée et soutenue par

Aalae BENKI

MÉTHODES EFFICACES DE CAPTURE DE
FRONT DE PARETO EN CONCEPTION
MÉCANIQUE MULTICRITÈRE.
APPLICATIONS INDUSTRIELLES

Thèse dirigée par **Abderrahmane HABBAL**
soutenue le 30/01/2014 devant le jury composé de :

M ^{me}	R. ABOULAICH	Professeure, Ecole Mohamadia d'ingénieurs	(Rapporteur)
M.	P. VANNUCCI	Professeur, Université Versailles St Quentin	(Rapporteur)
M.	G. MATHIS	ArcelorMittal R&D	(Examineur)
M.	J.A. DESIDERI	Professeur, INRIA Sophia	(Examineur)
M.	L. FOURMENT	Professeur, Ecole des Mines Sophia	(Examineur)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

162. قُلْ إِنَّ صَلَاتِي وَنُسُكِي وَمَحْيَايَ وَمَمَاتِي لِلَّهِ رَبِّ الْعَالَمِينَ

163. لَا شَرِيكَ لَهُ وَبِذَلِكَ أُمِرْتُ وَأَنَا أَوَّلُ الْمُسْلِمِينَ

162. Dis: «En vérité, ma Salāt, mes actes de dévotion, ma vie et ma mort appartiennent à Allah, Seigneur de l'Univers.

163. A Lui nul associé! Et voilà ce qu'il m'a été ordonné, et je suis le premier à me soumettre.»

Surate les bestiaux

سُورَةُ الْأَنْعَامِ

*A l'âme de mon grand père moulay habib, à mes
chers parents et ma chère femme*

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 Abderrahmane Habbal, Professeur à l'Université de Nice Sophia Antipolis, 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 grand merci à Monsieur Gael Mathis, Ingénieur de recherche à ArcelorMittal Global R&D , et Monsieur Olivier Beigneux, Ingénieur de recherche en métallurgie (aciers pour emballage) à ArcelorMittal Maizières , pour leurs aides, conseils et orientations. Je remercie aussi tous le personnel d'ArcelorMittal.

REMERCIEMENTS et profonde gratitude vont également aux membres de la commission d'examen. J'exprime ainsi toute ma reconnaissance à Madame Rajae Aboulaich, Professeure à l'École Mohammadia d'ingénieurs, qui a accepté d'être rapporteur de ce mémoire. J'adresse aussi mes sincères remerciements à Monsieur Paolo Vannucci, Professeur à l'Université Versailles Saint-Quentin, 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 Lionel Fourment, HDR à l'École Nationale Supérieure des Mines de Paris (ENSMP) au Centre de Mise en Forme des Matériaux (CEMEF) 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 Madame Montserrat Argente.

IL me serait impossible, enfin, de ne pas dire un grand MERCI à ma femme **Lala Zineb**, mes parents **Lala aicha et Moulay Ahmed**, mes beaux parents **Lala Souad et Sidi Mohamed** et mes deuxième parents **Lala Khadija et moulay Mokhtar**, ainsi qu'à toute ma famille, particulièrement ma chère grand mère **LALA CHRIF**, 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 arriver à finir cette thèse.

ENFIN, je remercie mes cher(e)s ami(e)s Z. saad, A. wissam , B. Youness, B. Mohammed, B. Mohammed, A. Jawad, N. Anas, T. issam, B. issam, P. Zakaria, Af. Asmae, Bo. Nabila et T. Mehdi.

Résumé Dans le domaine d'optimisation de forme de structures, 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 multicritère 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 de capture de front de Pareto (NBI ou NNCM) avec un métamodèle (RBF), c'est-à-dire des approximations des résultats des simulations coûteuses. D'après l'ensemble des résultats obtenus par cette approche, il est intéressant de souligner que la capture de front de Pareto génère un ensemble des solutions non dominées. Pour savoir lesquelles choisir, le cas échéant, il est nécessaire de faire appel à des algorithmes de sélection, comme par exemple Nash et Kalai-Smorodinsky. Ces deux approches, issues de la théorie des jeux, ont été utilisé pour notre travail. L'ensemble des algorithmes sont validés sur deux cas industriels proposés par notre partenaire industriel. Le premier concerne un modèle 2D du fond de la canette (elasto-plasticité) et le second est un modèle 3D de la traverse (élasticité linéaire). Les résultats obtenus confirment l'efficacité de nos algorithmes développés.

Mots-clés Optimisation multicritère, Méthode NBI, Méthode NNCM, Métamodèle RBF, Couplage, Equilibre de Nash, Equilibre de Kalai-Smorodinsky.

Abstract One of the current challenges in the domain of the multiobjective shape optimization is to reduce the calculation time required by conventional methods. The high computational cost is due to the high number of simulation or function calls required by these methods. Recently, several studies have been led to overcome this problem by integrating a metamodel in the overall optimization loop. In this thesis, we perform a coupling between the Normal Boundary Intersection -NBI- algorithm and The Normalized Normal constraint Method -NNCM- algorithm with Radial Basis Function -RBF- metamodel in order to have a simple tool with a reasonable calculation time to solve multicriteria optimization problems. First, we apply our approach to academic test cases. Then, we validate our method against two industrial cases, namely, shape optimization of the bottom of a can undergoing nonlinear elasto-plastic deformation and an optimization of an automotive twist beam. Then, in order to select solutions among the Pareto efficient ones, we use the same surrogate approach to implement a method to compute Nash and Kalai-Smorodinsky equilibria.

Keywords Multicriteria optimization problems, Normal Boundary Intersection, Normalized Normal constraint Method, Radial Basis Function metamodel, Coupling, Nash and Kalai-Smorodinsky equilibria.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	viii
LISTE DES FIGURES	x
PRÉSENTATION GÉNÉRALE DE LA THÈSE - MOTIVATION, OBJECTIF ET ORGANISATION	1
1 Outils de base en optimisation multicritère	5
1.1 État de l'art en optimisation multicritère	7
1.1.1 Optimisation monocritère	7
1.1.2 Optimisation multicritère	12
1.2 Revue des méthodes de capture de front de Pareto	15
1.2.1 La méthode Normal Boundary Intersection	15
1.2.2 La méthode Normalized Normal Constraint	20
1.2.3 Validation	23
1.3 Nécessité et revue des métamodèles	31
1.3.1 Généralités	31
1.3.2 Le métamodèle RBF	33
1.3.3 Validation	38
CONCLUSION	42
2 Modèles mécaniques étudiés	43
2.1 Élasticité en grande déformation pour le packaging (2D)	45
2.1.1 Généralités	45
2.1.2 Conception du modèle du fond de la canette 2D	47
2.1.3 Critères à optimiser : DRP vs DG	48
2.2 Élasticité linéaire pour une pièce automobile (3D)	50
2.2.1 Généralités	50
2.2.2 Conception du modèle de traverse 3D	51
2.2.3 Critères à optimiser : VM vs FR34	52
2.3 Techniques pour la déformation des formes en 2D et 3D	54
2.3.1 Interpolation Spline cubique pour la déformation des formes 2D	54
2.3.2 Interpolation par FFD+RBF pour la déformation des formes 3D	55
CONCLUSION	59
3 Étude de capture de front de Pareto via les métamo- dèles - Application industrielles	61

3.1	COUPLAGE DES MÉTHODES NBI NNCM ET DU MÉTAMODÈLE RBF	63
3.2	VALIDATION DE COUPLAGE (CAS ACADÉMIQUES)	63
3.2.1	Validation de couplage NBI et RBF	63
3.2.2	Validation de couplage NNCM et RBF	67
3.3	VALIDATION DE COUPLAGE NBI RBF POUR LES CAS INDUSTRIELS	70
3.3.1	Fond de la canette 2D	70
3.3.2	Pièce automobile : Traverse 3D	82
	CONCLUSION	90
4	SÉLECTION DE SOLUTIONS EFFICACES	91
4.1	SÉLECTION PAR JEUX DE NASH	93
4.1.1	Introduction	93
4.1.2	Split arbitraire -choix arbitraire de partage-	94
4.1.3	Aperçu d'autres stratégies de partage	99
4.2	SÉLECTION PAR JEU DE KALAI-SMORODINSKY	99
4.2.1	Introduction	99
4.2.2	Validation	101
	CONCLUSION	107
	CONCLUSION GÉNÉRALE	109
A	ANNEXES	111
A.1	VALIDATION DE LA MÉTHODE NBI	113
A.2	USAGE D'UN FILTRE POUR LA MÉTHODE NBI	114
A.3	VALIDATION DE LA MÉTHODE NNCM	115
A.4	USAGE D'UN FILTRE POUR LA MÉTHODE NNCM	116
A.5	ETUDE COMPARATIVE ENTRE LA DISTRIBUTION UNIFORME ET LE LATIN HYPERCUBE POUR LE CHOIX DES ÉLÉMENTS DE LA BASE RBF (POINTS MAÎTRES)	117
A.6	VALIDATION DU MÉTAMODÈLE RBF POUR DES FONCTIONS 1D .	121
A.7	VALIDATION DU MÉTAMODÈLE RBF POUR DES FONCTIONS 2D .	122
A.8	VALIDATION DE L'APPROCHE NBI RBF	123
A.9	OPTIMISATION DES PROFILS DE LA CANETTE 2D	124
A.10	EQUILIBRE DE NASH -PARTAGE ARBITRAIRE-	127
A.11	EQUILIBRE DE KALAI-SMORODINSKY	128
	BIBLIOGRAPHIE	129
	NOTATIONS	135

LISTE DES FIGURES

1.1	Les différents espaces de recherche	7
1.2	les différents types de minima pour une fonction : Minimum global (point rouge) et Minimum local (point bleu)	8
1.3	Illustration d'une itération de l'algorithme de la méthode générale pour la recherche d'un minimum	9
1.4	Exemple de la représentation de l'ensemble de Pareto (en bleu) relatif au problème de Tanaka	13
1.5	Illustration de la notion de dominance pour un cas bicritère, f_1 et f_2 sont deux fonctions à minimiser. la solution A domine les solutions C et E	14
1.6	Génération des éléments CHIM pour un problème à trois critères ($n=3$) et une discrétisation à 7 points ($m_i=7$)	18
1.7	Les étapes de la capture du front de Pareto par la Méthode NBI : (a) Calcul des minima individuels, (b) Détermination d'enveloppe convexe ou CHIM, (c) Discrétisation du CHIM, (d) Détermination du champs des directions normales, (e) Maximisation de l'amplitude descente le long des normales et (f) Capture de front de Pareto	19
1.8	Les étapes de la capture du front de Pareto par la Méthode NNCM : (a) Calcul des minima individuels et le point uTopia, (b) Passage à l'espace normalisé à l'aide des longueurs L_1 et L_2 , (c) Détermination de la line utopia, (d) Discrétisation de la line utopia, (e) Minimisation de la fonction f_2 normalisée et (f) Capture de front de Pareto	22
1.9	Résolution du problème SCH2 par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	25
1.10	Résolution du problème Min-Ex par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	26
1.11	Résolution du problème TNK par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	26
1.12	Résolution du problème POL par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	27
1.13	Résolution du problème FON par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	28

1.14	Résolution du problème P4 par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	28
1.15	Usage d'un filtre pour éliminer les solutions dominées obtenues par les méthodes (a) NBI et (b) NNCM pour les problèmes SCH2, TNK et POL	30
1.16	Ojectif des métamodèles : approximation de la fonction exacte f en un point donné x(m+1) (point rouge) à partir d'un échantillon de m points connus (x(1),...,x(m)) (points verts)	31
1.17	Illustration de l'influence du facteur d'atténuation a_f choisi pour le calcul de la fonction approchée (rouge) d'une fonction exacte donnée (noir) par le métamodèle RBF en utilisant différentes valeurs pour ce facteur en gardant toujours les même points maîtres (points bleus) pour la construction du métamodèle	35
1.18	Génération des points maîtres par les différentes méthodes citées : Génération aléatoire (bleu), Hypercube latin (vert) et Distribution uniforme (rouge)	37
1.19	Illustration de l'influence du choix des points maîtres générés par les méthodes (a) Hypercube latin,(b) Génération aléatoire et(c) Distribution uniforme, pour le calcul de la fonction approchée (rouge) d'une fonction exacte donnée (noir) par RBF	37
1.20	Approximation de la fonction Rastrigin 1D par RBF avec différents nombre total N de points maîtres (a) N=5 et (b) N=15	39
1.21	Approximation de la fonction Rastrigin 2D par RBF avec différents nombre total N de points maîtres (a) N=25 et (b) N=49	39
1.22	Approximation de la fonction Schwefel 1D par RBF avec différents nombre total N de points maîtres (a) N=6 et (b) N=10	40
1.23	Approximation de la fonction Schwefel 2D par RBF avec différents nombre total N de points maîtres (a) N=9 et (b) N=49	40
2.1	La boîte boisson dans tous ses formats	45
2.2	Coupe d'une boîte boisson et différentes zones de ses composantes	46
2.3	Passage de géométrie 3D au 2D pour un type de canette	47
2.4	Déroulement du pied de la canette causé par la pression interne	49
2.5	Exemple d'un calcul de DRP à l'aide de Ls Dyna	50
2.6	Exemple d'un calcul de DG à l'aide de Ls Dyna	50
2.7	Critères à optimiser pour la traverse 3D	53
2.8	Exemple d'un calcul de FR_{34} à l'aide de Ls Dyna : les deux forces FR_3 et FR_4 sont calculé à l'instant final de la simulation (t=1) et $FR_{34} = \sqrt{FR_3^2 + FR_4^2}$	53

2.9	Exemple d'un calcul de VM à l'aide de Ls Dyna : VM est indiqué en haut à gauche	54
2.10	Géométrie 3D à déformer (Ensemble des points en bleu et les points maîtres en rouge)	58
2.11	Surface déformée (b) par Free Form Deformation (FFD + RBF) à partir d'une surface initiale plane (a)	58
3.1	Comparaison des résultats obtenus par la méthode NBI (bleu) avec ceux obtenus par l'approche NBI RBF (rouge) pour les problèmes test de (a) SCH ₁ , (b) HannaNC, (c) Fon et (d) P ₄	66
3.2	Comparaison des résultats obtenus par la méthode NNCM (bleu) avec ceux obtenus par l'approche NNCM RBF (rouge) pour les problèmes test de (a) SCH ₂ , (b) TNK, (c) P ₃ et (d) P _{4bis}	69
3.3	Les trois types de profils de canettes étudiés	71
3.4	Illustration de la stratégie pour le choix de la variable d'optimisation pour chaque profil de la canette 2D (cas 1, cas 2 et cas 3) : Division de la forme en partie fixe (Zone entre les points bleus) et en partie variable caractérisée par quatre points de contrôle (Points verts) qui constituent la variable de l'optimisation	72
3.5	Illustration de la stratégie suivie pour le choix des points maîtres pour la construction du métamodèle RBF et déterminer les contraintes de notre problèmes d'optimisation (Contraintes de bornes) pour chaque profil de la canette 2D (cas 1, cas 2 et cas 3) : les points de contrôle initiaux avec ces trois situations possibles (points verts) permettent de générer $3^4 = 81$ profils représentant les points maîtres pour le RBF	73
3.6	Organigramme de fonctionnement du couplage NBI RBF pour le cas industriel du fond de la canette 2D - Exemple pour 81 éléments pour la base RBF	75
3.7	Comparaison entre les résultats obtenus par l'approche NBI RBF, et l'évaluation exacte de ces résultats pour différentes valeur N de discrétisation du CHIM pour le profil de canette -Cas 1-	76
3.8	Comparaison entre les résultats obtenus par l'approche NBI RBF, et l'évaluation exacte de ces résultats pour différentes valeur N de discrétisation du CHIM pour le profil de canette -Cas 2-	77
3.9	Comparaison entre les résultats obtenus par l'approche NBI RBF, et l'évaluation exacte de ces résultats pour différentes valeur N de discrétisation du CHIM pour le profil de canette -Cas 3-	78
3.10	Nouveaux Profils des canettes 2D vérifiant les caractéristiques requises par l'industriel pour les trois types de canette. Les valeurs de DRP et DG montrent que ces solutions sont deux à deux non dominées	80

3.11	Superposition des résultats obtenus par notre approche NBI RBF après filtrage (rouge), les éléments constituant la base RBF (noir) et la solution initiale (bleu) pour les trois types de canette (cas 1, cas 2 et cas 3)	81
3.12	Visualisation du profil de la traverse 3D sur le logiciel LSprepost de LSDyna	82
3.13	Zoom sur la traverse 3D permettant de mettre en évidence le ventre de la traverse sur le logiciel LSprepost de LSDyna	83
3.14	(a) Le ventre de la traverse 3D capturé par le logiciel Scilab et (b) La projection 2D du ventre 3D	83
3.15	Une projection 2D du ventre de la traverse 3D permettant de mettre en évidence l'ensemble des noeuds (en bleu) fixés durant l'optimisation et l'ensemble des noeuds RBF (en rouge) pour la génération des nouveaux profils pour la traverse à l'aide de la méthode FFD par RBF.	84
3.16	Illustration de la simplification effectuée afin de réduire la dimension de notre variable d'optimisation à trois. Au début, le choix était de neuf points de contrôle et pour chaque point, nous avons cinq positions possibles (points noirs), A la fin, nous avons caractérisé chaque point par la longueur de l'intervalle dans les trois directions au lieu des cinq positions possibles sur l'intervalle. Pour l'ensemble de points, nous avons exigé la même longueur pour les intervalles. . .	84
3.17	Organigramme de fonctionnement du couplage NBI RBF pour le cas industriel du travesrse 3D - Exemple pour une base RBF de 125 éléments	86
3.18	Résultats de l'optimisation concurrente de forme de la traverse 3D pour différentes valeur N de discrétisation du CHIM	87
3.19	Nouveaux profils (a, b et c) de la Traverse 3D vérifiant les caractéristiques requises par l'industriel. Les valeurs de VM et FR_{34} montrent que ces solutions sont deux à deux non dominées entre elles.	89
4.1	Les équilibres de Nash (point bleu) pour les problèmes Constmin, TNK et POL par les deux partages arbitraires possibles	96
4.2	Les équilibres de Nash (points rouges) pour le problème du fond de canette 2D par des différents partages arbitraires pour les trois types de profil de canette	98
4.3	La solution de Kalai-Smorodinsky KS (point bleu) pour un problème bicritère	100
4.4	Les étapes de l'identification de l'équilibre de KS : Calcul des minima individuels et le point utopia (point noir), calcul du point nadir (point vert), détermination de la line reliant les points utopia et nadir, et l'identification de l'équilibre de KS (point bleu).	101
4.5	Identification de l'équilibre de Kalai-Smorodinsky (point bleu) pour les problèmes (a) Constmin, (b) TNK et (c) POL .	102

4.6	Equilibre de Kalai-Smorodinsky calculé à l'aide des critères approchés par RBF (figures à gauche : KS approché est le point rouge et les points bleus présentent le front de Pareto approché par RBF) et sa projection réelle sur l'espace fonctionnel (figures à droites : KS est le point rouge) pour les trois types de profil de la canette 2D	104
4.7	Comparaison entre les positions de l'équilibre de Kalai-Smorodinsky (point rouge sur les figures à droite) et les équilibres de Nash (points rouges sur les figures à gauche) par rapport à l'espace fonctionnel (points noirs) et la solution initiale (point bleu) pour les trois types de profil de la canette 2D.	105
4.8	Profils correspondant aux équilibres de Kalai-Smorodinsky pour les trois types de profil de la canette.	106
A.1	Résolution des problèmes (a) SCH ₁ , (b) Messac, (c) Hanna non convexe, (d) ZDT ₁ , (e) Deb et (f) Hanna discontinu par la méthodes NBI, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	113
A.2	Usage d'un filtre pour éliminer les solutions dominées obtenues par la méthode NBI pour les problèmes (a) Deb et (b) Hanna discontinu	114
A.3	Résolution des problèmes (a) Max-Ex, (b) Const-Min-Ex, (c) P ₃ , (d) ZDT ₂ , (e) Messac-mod et (f) Tanaka-mod par la méthodes NBI, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge	115
A.4	Usage d'un filtre pour éliminer les solutions dominées obtenues par la méthode NNCM pour les problèmes (a) Tanakamod et (b) Tanaka	116
A.5	Approximation des fonctions (a) Rastrigin 1D et (b) Schwefel par RBF avec le même nombre total (N = 15) de points maîtres avec les méthodes de génération de points : distribution uniforme (rouge) et latin hypercube (bleu)	117
A.6	Approximation des fonctions Rastrigin 2D et Branin par RBF avec le même nombre total (N = 81) de points maîtres avec les méthodes de génération de points : distribution uniforme (figures à gauches) et latin hypercube (figures à droites) -fonctions exactes en rouge et fonctions approchées en bleu-	118
A.7	Approximation du critère DRP avec le même nombre total (N = 81) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas2.	119

A.8	Approximation du critère DRP avec le même nombre total (N = 81) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas3	119
A.9	Approximation du critère DG avec le même nombre total (N = 81) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas2	120
A.10	Approximation du critère DG avec le même nombre total (N = 81) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas3	120
A.11	Approximation des fonctions (a) Rastrigin , (b) Perm, (c) Zakharov et (d) Ackbley, avec le métamodèle RBF construit avec deux points maîtres répartis d'une manière uniforme (distribution uniforme).	121
A.12	Approximation des fonctions (a) Matyas , (b) Perm, (c) Beale et (d) Branin, avec le métamodèle RBF construit avec deux points maîtres répartis d'une manière uniforme (distribution uniforme).	122
A.13	Comparaison entre les solutions obtenues par l'approche NBI RBF (* rouge) et celles obtenus par NBI (points bleus) pour les problèmes (a) Tanaka, (b) Messac et (c) Const-Min-Ex.	123
A.14	Optimisation de la forme du font de la canette de type cas 1 avec l'approche NBI RBF : (a) Présentation du problème à traiter avec la variable d'optimisation et les contraintes, (b) Les résultats obtenus par l'approche NBI RBF (points rouge) et les éléments de la base RBF (points noirs), (c) et (d) des nouveaux profils pour le fond de la canette vérifiant les exigences industriels.	124
A.15	Optimisation de la forme du font de la canette de type cas 2 avec l'approche NBI RBF : (a) Présentation du problème à traiter avec la variable d'optimisation et les contraintes, (b) Les résultats obtenus par l'approche NBI RBF (points rouge) et les éléments de la base RBF (points noirs), (c) et (d) des nouveaux profils pour le fond de la canette vérifiant les exigences industriels.	125
A.16	Optimisation de la forme du font de la canette de type cas 3 avec l'approche NBI RBF : (a) Présentation du problème à traiter avec la variable d'optimisation et les contraintes, (b) Les résultats obtenus par l'approche NBI RBF (points rouge) et les éléments de la base RBF (points noirs), (c) et (d) des nouveaux profils pour le fond de la canette vérifiant les exigences industriels.	126

A.17	Equilibre de Nash pour les problèmes Min-Ex, Max-Ex et P_4 pour deux différents partages arbitraires	127
A.18	Equilibre de Kalai-Smorodinsky (point bleu) pour les problèmes (a) Min-Ex, (b) Hanna non discontinu, (c) Fonseca et (d) P_4	128

PRÉSENTATION GÉNÉRALE DE LA THÈSE - MOTIVATION, OBJECTIF ET ORGANISATION. . .

CADRE GÉNÉRAL

Dans l'ingénierie relative à la conception des pièces métalliques dans plusieurs domaines (comme l'aérodynamique, l'hydrodynamique, l'acoustique, l'électromagnétisme, l'électronique, l'optique et autres), il existe une étape primordiale parmi les étapes de la conception qui est l'optimisation de la forme (i.e optimisation de structures) de la pièce. En général, un problème d'optimisation de structures (ou de formes) est défini par trois données :

- un modèle qui permet d'évaluer (on dit aussi d'analyser) le comportement mécanique d'une structure.
- un critère que l'on cherche à minimiser ou maximiser, et éventuellement plusieurs critères.
- un ensemble admissible de variables d'optimisation qui tient compte d'éventuelles contraintes que l'on impose aux variables.

L'optimisation de forme est faite à l'aide d'un ensemble de méthodes permettant de trouver la « *meilleure forme* » à donner à une pièce pour qu'elle remplisse l'ensemble de ses critères. Ces dernières sont établies par l'analyse fonctionnelle. La capacité de la pièce à remplir un critère peut en général s'exprimer sous forme numérique.

En réalité, pour les problèmes de forme en industrie, nous sommes souvent obligés d'optimiser plusieurs critères simultanément qui sont aussi concurrents (i.e on ne peut pas améliorer un critère sans détériorer l'autre). Nous pouvons considérer ces problèmes comme des problèmes d'optimisation multicritère. En principe, la résolution de ce genre de problème nous mène à l'identification de front de Pareto (ensemble des solutions non dominées), une identification que nous pouvons effectuer en faisant appel à un ensemble d'algorithmes d'optimisation ad hoc comme les algorithmes génétiques, le recuit simulé, la méthode Normal Boundary Intersection (NBI) (Deb.1996) (Shukla.2007) et Normalized Normal constraint Methods (NNCM) (Messac.2004) (Martinez.2007). Les deux dernières méthodes citées sont de plus en plus utilisées de nos jours pour résoudre des problèmes industriels (Siddiqui.2012) (Logist.2012)

(Ganesana.2013) (Lim.2001) (Jia.2007) (Datta.2005).

Les Méthodes NBI et NNCM nous permettent à travers l'identification du front de Pareto d'avoir un ensemble de bonnes solutions. Lorsqu'il est nécessaire d'effectuer un choix parmi les solutions Pareto-optimales, souvent beaucoup trop nombreuses, nous pouvons utiliser des algorithmes de sélection parmi lesquels nous considérons ici ceux issus de la théorie des jeux, comme l'équilibre de Nash (Nash.1950) (Leyton-Brown.2008) ou celui de Kalai-Smorodinsky (Kalai.1975) (Nagahisa.2002) (Bozbay.2012).

L'identification du front de Pareto pour nos problèmes d'optimisation industriels est très coûteuse en temps de calcul, à cause de la nécessité d'un grand nombre d'évaluations de critères à optimiser, nécessitant à chaque fois le calcul coûteux de la réponse mécanique de la structure, réponse parfois non-linéaire. Pour remédier à cela, une approche courante revient à développer des algorithmes efficaces qui consistent à coupler les méthode de résolution (i.e méthode de capture de front de Pareto ou méthode de sélection) pour ces problèmes d'optimisation avec un métamodèle (Wang.2007) (Duvigneau.2007). Parmi les travaux relatifs à cette approche, nous pouvons citer : (Lian.2004) (Tahk.2003) (Fang.2004) (Chung.2004) (Farina.2002) (Raza.2008) (Sobester.2005) (Wu.2002) (Coelho.2007) (Coelho.2008) (Jones.2001).

MOTIVATION, PROBLÉMATIQUE ET OBJECTIFS

Dans cette thèse, nous allons valider nos approches et les appliquer à deux cas industriels proposés par notre partenaire industriel **Arcelor Mittal**. Nous avons deux cas différents pour deux différents services de l'entreprise, le modèle du fond de la canette fourni par **l'équipe Cans & Components du centre Packaging d'ArcelorMittal Research SA** et qui est un modèle 2D, et le modèle 3D de la traverse fourni par **ArcelorMittal R&D MontataireAutomotive Applications** et **ArcelorMittal Tubular Product Vitry**.

Pour les deux cas traités, nous allons réaliser une optimisation de forme afin de chercher les meilleurs profils qui, à la fois, optimisent les critères et respectent les contraintes exigées par l'organisme industriel. Comme nous l'avons déjà signalé, la résolution de ce genre de problème exige de coupler les méthodes de résolution (capture de front de Pareto ou sélection de solutions) avec des métamodèles. Dans le cadre de notre travail, nous avons choisi, dans un premier temps, de résoudre les problèmes en identifiant le front de Pareto avec les méthodes NBI et NNCM couplées avec le métamodèle de type fonctions à base radiale (RBF) (Rippa.1999) (Duvigneau.2007). Dans un deuxième temps, nous utiliserons un couplage du métamodèle RBF avec les algorithmes de Nash et Kalai-Smorodinsky afin de sélectionner les solutions optimales considérées comme des équilibres des algorithmes cités.

Suite à ces données et ces exigences, nous nous sommes fixés comme

objectifs :

- Modéliser les deux cas industriels à l'aide du logiciel LS Dyna et programmer un code scilab permettant d'extraire les critères à optimiser à travers les résultats de LS Dyna.
- Construire des métamodèles précis pour les critères à optimiser à l'aide du métamodèle RBF.
- Etudier l'ensemble des couplages entre les méthodes classiques et le métamodèle RBF (NBI RBF, NNCM RBF, Nash RBF, Kalai-Smorodinsky RBF).
- Appliquer ces approches à la fois à la résolution des cas tests académiques et aux cas industriels.

CONTENU DE LA THÈSE

Ce manuscrit de thèse est composé de quatre parties. La première est consacrée à la présentation complète des notions des outils et méthodes utilisées durant la thèse (NBI, NNCM et RBF) et la validation des codes Scilab relatifs. La deuxième partie, est une présentation des modèles mécaniques des cas traités (fond de canette 2D et traverse 3D), ainsi que les méthodes utilisées pour la déformation des modèles durant l'optimisation (Spline cubique et FFD par RBF respectivement pour les cas 2D et 3D). Dans la troisième partie, nous présentons une contribution à l'optimisation multicritère à base de métamodèle à l'aide des algorithmes développés durant la thèse à savoir (NBI RBF et NNCM RBF), ces derniers sont validés à la fois pour des cas tests académiques connus et pour les applications industrielles qui représentent des problèmes de conception multicritère des structures mécaniques. Finalement, la dernière partie a pour objet de faire une comparaison entre les deux équilibres utilisés pour la sélection de solutions à savoir Nash et Kalai-Smorodinsky respectivement, une comparaison faite à la base de la résolution de certains problèmes tests académiques et un seul problème industriel qui est le fond de la canette 2D.

Nous concluons le manuscrit en synthétisant les principales contributions présentées, et en dénombrant de nouvelles perspectives sur la base des travaux effectués.

OUTILS DE BASE EN OPTIMISATION MULTICRITÈRE



SOMMAIRE

1.1	ÉTAT DE L'ART EN OPTIMISATION MULTICRITÈRE	7
1.1.1	Optimisation monocritère	7
1.1.2	Optimisation multicritère	12
1.2	REVUE DES METHODES DE CAPTURE DE FRONT DE PARETO	15
1.2.1	La méthode Normal Boundary Intersection	15
1.2.2	La méthode Normalized Normal Constraint	20
1.2.3	Validation	23
1.3	NECESSITÉ ET REVUE DES MÉTAMODÈLES	31
1.3.1	Généralités	31
1.3.2	Le métamodèle RBF	33
1.3.3	Validation	38
	CONCLUSION	42

DANS ce chapitre, nous allons présenter à la fois le contexte de notre travail, ainsi que l'ensemble des outils utilisés (méthodes et techniques); nous commencerons par le contexte méthodologique général de la thèse qui est les problèmes d'optimisation multicritère, les méthodes de résolutions de ce genre de problème (parmi lesquelles NBI et NNCM) et les techniques d'approximation ou encore métamodèles, et nous terminerons par une présentation de l'implémentation de l'ensemble des outils (en Scilab).

1.1 ETAT DE L'ART EN OPTIMISATION MULTICRITÈRE

1.1.1 Optimisation monocritère

Un problème d'optimisation monocritère est défini comme la recherche du minimum ou du maximum (optimum) d'une fonction donnée. Nous pouvons aussi trouver certains problèmes d'optimisation de ce genre pour lesquels les variables de la fonction à optimiser doivent vérifier un certain nombre de contraintes. Dans ce cas, nous sommes devant un cas particulier que nous appelons un problème d'optimisation avec contraintes.

En général, un problème d'optimisation monocritère peut s'écrire sous la forme mathématique suivante :

$$\begin{aligned} & \min_x f(x) \\ s/c \quad & \begin{cases} g_j(x) \geq 0 & j = 1; \dots; J \quad (*) \\ h_k(x) = 0 & k = 1; \dots; K \quad (**) \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \quad (***) \end{cases} \end{aligned} \quad (1.1)$$

Pour ce genre de problème, il y a trois types de contraintes :

- Des contraintes de bornes (***) : les valeurs de x vérifiant ces contraintes définissent l'*espace de recherche*. Cet espace est représenté à la figure 1.1-a (pour le cas $m=2$).
- Des contraintes de type égalités et inégalités ((* et (**)) : les valeurs de x vérifiant ces contraintes définissent l'*espace des valeurs réalisables* ou *espace admissible*. Cet espace est représenté à la figure 1.1-b (pour le cas $m=2$).

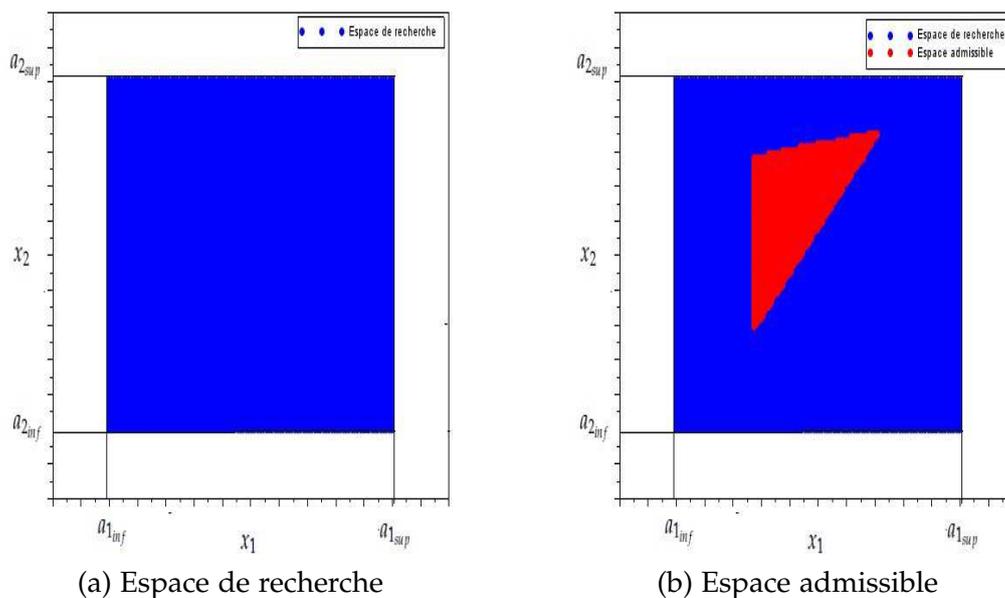


FIGURE 1.1 – Les différents espaces de recherche

Dans la formulation (1.1), nous nous limitons à un problème de minimisation, puisque la maximisation d'une fonction $f(x)$ peut facilement être transformée en un problème de minimisation :

$$\max_x f(x) = - \min_x (-f(x)) \quad (1.2)$$

Dans toute la suite, nous supposerons que la fonction f est à minimiser.

La fonction f a plusieurs dénominations comme la *fonction critère*, *fonction objectif* ou encore *fonction coût*.

Le vecteur x est appelé *variable de décision* ou encore *variable d'optimisation*. C'est en faisant varier ce vecteur que l'on recherche le minimum de la fonction f .

Pour la fonction critère, on peut définir deux types de minima :

- Un minimum global : un point x^* est un minimum global de la fonction f si on a : $f(x^*) < f(x)$ quel que soit x tel que $x^* \neq x$.
- Un minimum local : un point x^* est un minimum local de la fonction f si on a : $f(x^*) < f(x)$ quel que soit $x \in V(x^*)$ et $x^* \neq x$, où $V(x^*)$ définit un voisinage de x .

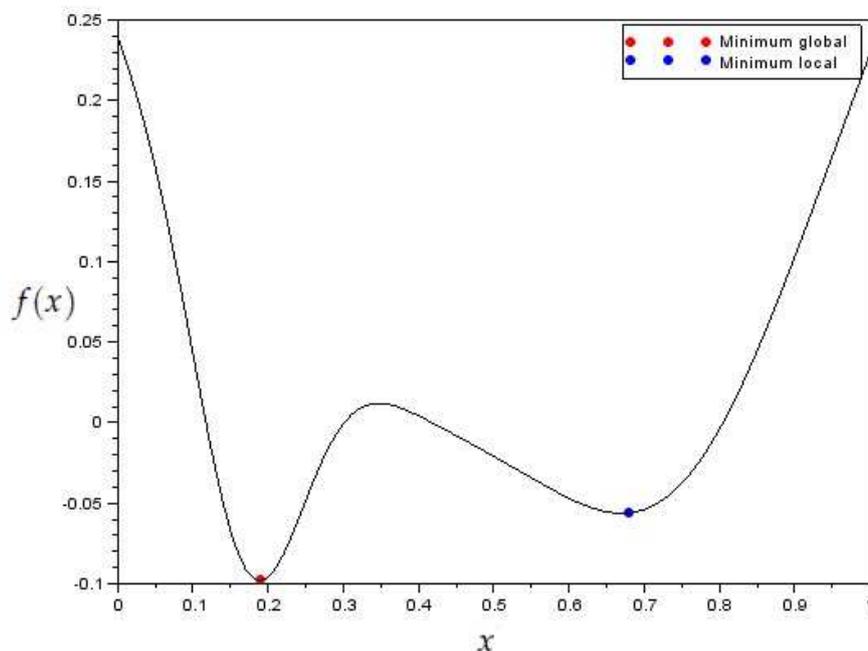


FIGURE 1.2 – les différents types de minima pour une fonction : Minimum global (point rouge) et Minimum local (point bleu)

Pour identifier ces minima, nous pouvons faire appel à un ensemble de méthodes dédiées pour ce faire, et parmi ces dernières, nous citons certaines méthodes pour chaque catégorie de minimisation (locale et globale).

Méthodes locales

Optimiser localement, c'est chercher une solution à un problème qui soit proche d'une solution de départ (optimisation locale), mais qui soit meilleure en terme de coût (fonction critère). Pour cela, on recherche une meilleure solution par itérations successives, cette classe de méthodes peut se diviser en deux catégories : méthodes de minimisation sans calcul de gradient et celles avec le calcul de gradient.

Méthode de minimisation sans calcul de gradient

Méthode générale : C'est la méthode la plus simple, très semblable à la méthode de bisection pour trouver un zéro, elle se base sur l'idée de d'encadrer le minimum entre 3 points $a < b < c$ tel que $f(b) < f(a)$ et $f(b) < f(c)$ et on répète l'encadrement jusqu'à la convergence (voir l'algorithme 1).

Algorithm 1 Méthode générale

- 1: **Données :** les points a b c et la fonction f tel que $f(b) < f(a)$ et $f(b) < f(c)$
 ▷ f la fonction à minimiser
 - 2: Soient $x_1 \in [a, b]$ et $x_2 \in [b, c]$
 - 3: Si $f(x_1) < f(a)$ alors $a = x_1$ ▷ Test x_1 à gauche
 - 4: Si $f(x_1) < f(b)$ alors $b = x_1$
 - 5: Si $f(x_2) < f(c)$ alors $c = x_2$ ▷ Test x_2 à droite
 - 6: Si $f(x_2) < f(b)$ alors $b = x_2$
 - 7: Retourner en 2
-

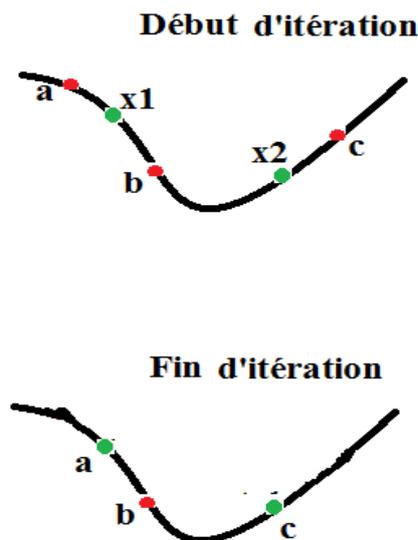


FIGURE 1.3 – Illustration d'une itération de l'algorithme de la méthode générale pour la recherche d'un minimum

Méthode de minimisation avec calcul de gradient

Parmi les méthodes qui font appel au calcul de gradient et /ou de hessien afin de faire la minimisation d'une fonction, nous trouvons la famille des méthodes de minimisation à directions de descente.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction continûment différentiable, et x un vecteur de \mathbb{R}^n . Le vecteur $d \in \mathbb{R}^n$ est appelé direction de descente de f en x si et seulement si $\nabla f(x)^T d < 0$.

Le principe général de ces méthodes est le suivant : pour un x_1 donné, on construit un processus x_k satisfaisant $f(x_{k+1}) < f(x_k)$ avec :

$$x_{k+1} = x_k + \alpha_k d_k$$

où d_k est une direction de descente et α_k minimise $f(x_{k+1})$ dans la direction d_k .

Algorithm 2 Algorithme général des méthodes de descente

- 1: **Initialisation** x_0, α_0 et d_0
 - 2: Tant que test d'arrêt n'est pas vérifié faire
 - 3: $x_{k+1} = x_k + \alpha_k d_k$
 - 4: Fin Tant que
-

Les algorithmes à directions de descente portent en général le nom de leur direction de descente comme :

- **l'algorithme du gradient** est celui qui utilise la direction du gradient.

$$d_k = -\nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Il s'agit bien d'une direction de descente si $f'(x_k) \neq 0$.

Si on prend $\alpha_k = \text{constante}$, alors on est devant l'algorithme du gradient à pas fixe.

Si on prend $\alpha_k = \arg \min_{\alpha} f(x_k + \alpha d_k)$, alors on est devant l'algorithme du gradient à pas optimal.

- **l'algorithme du gradient conjugué** est celui qui utilise les directions du gradient conjugué.

$$d_k = \nabla f(x_k) + \frac{\|\nabla f(x_k)\|}{\|\nabla f(x_{k-1})\|} d_{k-1}$$

$$\alpha_k = \frac{-\nabla f(x_k)}{(\nabla^2 f(x_k) d_k) \cdot d_k}$$

À la première itération, il n'y a pas de direction précédente et l'algorithme prend alors simplement l'opposé du gradient.

- **l'algorithme de Newton** est celui qui utilise la direction de Newton. La direction de Newton est définie en un point x_k en lequel le hessien de f est inversible par :

$$d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Toutes ces méthodes convergent vers un minimum local. Pour trouver le minimum global, il faut faire appel à des méthodes plus complexes que nous présentons dans la suite.

Méthodes globales

Ces dernières années, une forte demande dans le domaine des applications industrielles a obligé les chercheurs à donner plus d'importance à l'optimisation globale. Et pour ce faire, plusieurs travaux sont effectués pour l'évolution des algorithmes permettant de chercher l'optimum global. Ces méthodes se divisent en deux familles, les méthodes déterministes et celles non-déterministes.

Méthodes déterministes

Ces méthodes se basent sur des techniques déterministes, comme le "branch and bound" utilisant la programmation par intervalle. Ces techniques permettent, lorsqu'elles sont utilisables, de trouver les optima de fonctions de façon certaine. Leur domaine d'application s'étend lentement, puisque l'on est passé au fil des années de fonctions à quelques variables, à des fonctions pouvant compter jusqu'à une vingtaine de variables, au delà de ce nombre de variables, les méthodes peuvent avoir des difficultés pour trouver les minima. Et parmi ces méthodes déterministes (Floudas.2000) (Horst.1996), nous pouvons citer :

- Méthodes de séparation-évaluation (Clausen.1999) (Branch-and-Bound methods) .
- Méthodes à Recherche par Quadrillage (Elster.1995) (Grid Search Methods).
- Méthodes des Trajectoires (Stryk.1992) (Trajectory Methods).

Méthodes non-déterministes ou stochastiques

Ces méthodes se basent sur des techniques stochastiques, comme le recuit simulé ou les algorithmes évolutionnaires. Ces techniques, principalement développées par John Holland et popularisées par David Goldberg, sont particulièrement utiles sur des problèmes trop complexes pour être résolus par les méthodes d'optimisation classiques. Et parmi ces méthodes, nous pouvons citer :

- **Le recuit simulé** : C'est la méthode la plus utilisée aujourd'hui. Très populaire ces dernières années, elle a permis de résoudre des problèmes très complexes du type « voyageur de commerce » où les méthodes déterministes sont rapidement piégées dans des minima locaux. Elle s'inspire de la thermodynamique et c'est une méthode

de type « Monte Carlo » donc nécessite des nombres aléatoires (voir (Kirkpatrick.1983) (Cerny.1985)).

- **Les algorithmes de colonie de fourmis** : Il s'agit d'algorithmes inspirés du comportement des fourmis et qui constituent une famille de métaheuristiques d'optimisation. Le premier algorithme de colonies de fourmis proposé est appelé le système fourmi. Il vise notamment à résoudre le problème du voyageur de commerce, où le but est de trouver le plus court chemin permettant de relier un ensemble de villes. L'algorithme général est relativement simple, et repose sur un ensemble de fourmis, chacune parcourant un trajet parmi ceux possibles. À chaque étape, la fourmi choisit de passer d'une ville à une autre en fonction de quelques règles (voir (Dorigo.1996) (Colomi.1991) (Dorigo.1997)) (source de définition wikipédia).
- **Les algorithmes génétiques** : Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes. Leur but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné. La solution est approchée par « bonds » successifs, comme dans une procédure de séparation et évaluation, à ceci près que ce sont des formules qui sont recherchées et non plus directement des valeurs (voir (Holland.1975) (Goldberg.1994)) (source de définition wikipédia).

1.1.2 Optimisation multicritère

Selon la formulation (1.1), la résolution d'un problème d'optimisation monocritère consiste à chercher un minimum pour une seule fonction objectif. Cependant, lorsqu'on modélise un problème, on cherche souvent à satisfaire plusieurs critères. Par exemple, on veut un système performant et on veut aussi que ce système consomme peu. Dans ce cas, on parle d'un problème d'optimisation multicritère (ou multiobjectif). Celui-ci s'écrit de la manière suivante :

$$\begin{aligned} \min_x F(x) &= (f_1(x); f_2(x); \dots; f_n(x))^T \\ \text{s/c } (A) &= \begin{cases} g_j(x) \geq 0 & j = 1; \dots; J & (*) \\ h_k(x) = 0 & k = 1; \dots; K & (**) \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m & (***) \end{cases} \quad (1.3) \end{aligned}$$

Contrairement aux problèmes d'optimisation monocritère, où la notion d'optimum a un sens grâce à la relation d'ordre total sur les réels, dans le cas multicritère, la relation d'ordre total n'existe pas et la notion d'optimum devient arbitraire. Une solution est alors de chercher un ensemble de solutions nommées *solutions non dominées*, parmi lesquelles nous ne

pouvons pas décider si l'une est meilleure que l'autre, aucune ne permet systématiquement l'optimum pour tous les critères.

Un concept intéressant, qui nous permettra de définir ces solutions, est le *compromis*. En effet, le but de la résolution d'un problème d'optimisation multicritères est de chercher l'ensemble des meilleurs compromis sur les nombreux critères de l'étude, cet ensemble est encore nommé *l'ensemble de Pareto*, et sa représentation graphique est le *front de Pareto* (Figure 1.4).

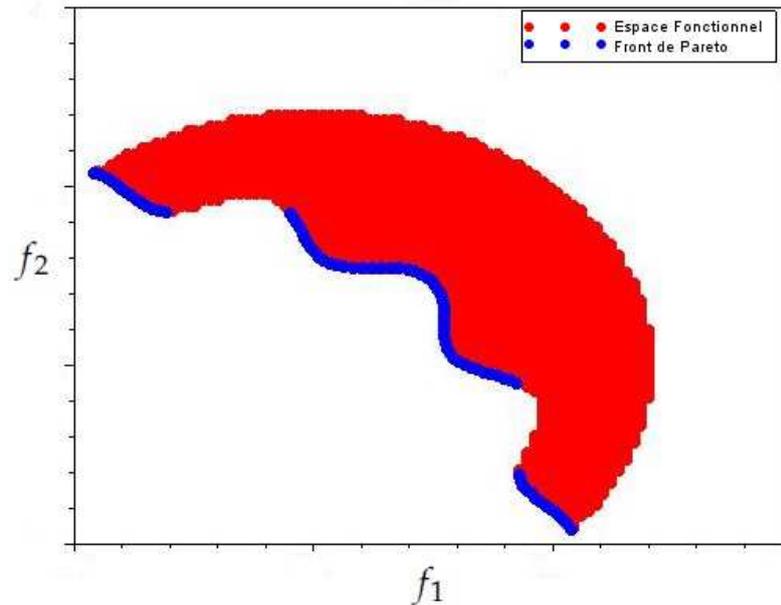


FIGURE 1.4 – Exemple de la représentation de l'ensemble de Pareto (en bleu) relatif au problème de Tanaka

Comme il a été signalé ci-dessus, l'ensemble de Pareto est constitué des solutions non dominées, ou autrement dit, elles vérifient la relation de dominance comme suit :

Soient x_1 et x_2 deux solutions potentielles au problème multicritère (1.3).

La solution $x_1 \in (A)$ est dite dominer $x_2 \in (A)$ si :

$$\forall i, f_i(x_1) \leq f_i(x_2)$$

Avec au moins un i tel que $f_i(x_1) < f_i(x_2)$

Remarque :

Une solution x_1 est dite **faiblement non dominée** s'il n'existe pas de solution $x_2 \in (A)$ telle que :

$$\forall i, f_i(x_1) < f_i(x_2)$$

Une solution x_1 est dite **fortement non dominée** s'il n'existe pas de solution $x_2 \in (A)$ telle que :

$$f_i(x_2) \leq f_i(x_1)$$

Avec au moins un i tel que $f_i(x_1) < f_i(x_2)$

où $i=1,..,n$ avec n le nombre des fonctions critère.

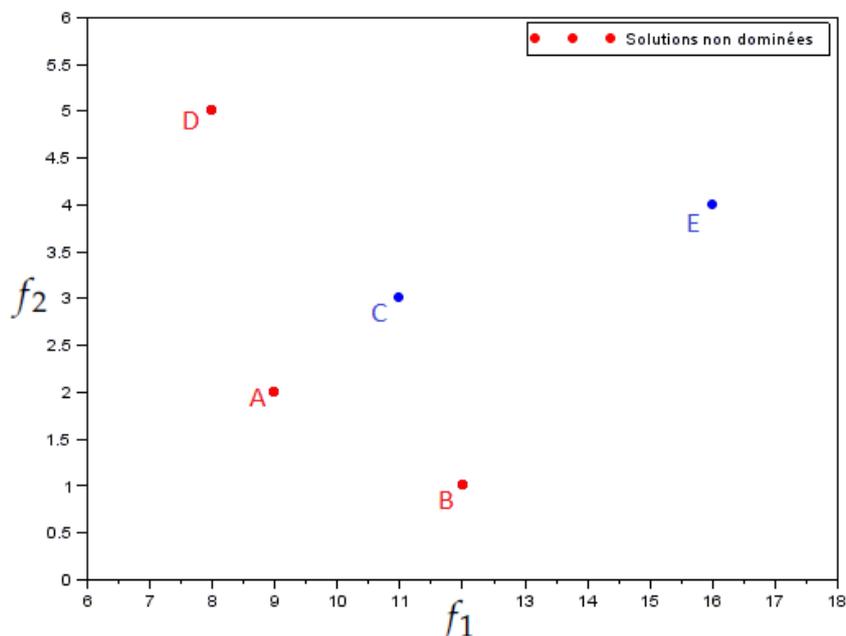


FIGURE 1.5 – Illustration de la notion de dominance pour un cas bicritère, f_1 et f_2 sont deux fonctions à minimiser. la solution A domine les solutions C et E

La figure 1.5 illustre le concept de dominance dans le cas d'un problème d'optimisation bicritère, un problème où nous cherchons à minimiser deux fonctions critère f_1 et f_2 . Dans ce cas, les solutions représentées par les points A et B dominent la solution représentée par le point C, par contre, les solutions représentées par les points A, B et D ne sont dominées par aucune solution. Les points représentatifs de ces solutions non dominées constituent le front de Pareto.

Nous remarquons toujours à travers La figure 1.5 la diversité des solutions. On peut ainsi distinguer :

- celles qui sont uniformément améliorables (points bleus) : il est possible de trouver des solutions meilleures pour l'ensemble des fonctions critère.
- celles qui ne sont pas uniformément améliorables (points rouges) : il est impossible de trouver une solution meilleure qu'elles pour l'ensemble des fonctions critère.

Ce sont les solutions entrant dans ce deuxième cas de figure que l'on désigne comme optimaux au sens de Pareto, ou Pareto-optimaux ou ensemble de Pareto. La notion du front de Pareto ne permet donc pas de les comparer entre elles. Pour savoir lesquelles sont les plus souhaitables, il est nécessaire de faire appel à d'autres critères d'évaluation, d'un point de vue qualitatif ou quantitatif où bien d'utiliser autres approches comme les jeux de Nash.

En conclusion de cette partie, la notion adéquate lors de la résolution d'un problème d'optimisation multicritère est celle du front de Pareto, lieu géométrique, dans l'espace des critères (*Espace Fonctionnel*), de tous les points non dominés. Pour la détermination numérique de front de Pareto, plusieurs méthodes peuvent être adoptées.

1.2 REVUE DES METHODES DE CAPTURE DE FRONT DE PARETO

Compte tenu de l'importance de la capture du front de Pareto, plusieurs recherches ont été consacrées à développer des méthodes efficaces permettant cette capture. Le livre (Yann.2002) de Yann Collette et Patrick Siarry traite l'ensemble de ces algorithmes classés comme suit :

- Les méthodes scalaires (Coello.1998) : Pondération des fonctions objectif, Keeney-Raiffa, Distance à un objectif de référence, Compromis, Corley, But à atteindre, But programmé, Algorithme de Lin-Giesy et algorithme de Lin-Tabak.
- Les méthodes interactives (Miettinen.1999) : Compromis par substitution, Fandel, STEP, Jahn, Geoffrion et la méthode du simplexe.
- Les méthodes floues (Sakawa.1988) : La méthode de Sakawa et celle de Reardon.
- Les méthodes exploitant une métaheuristique (Sait.1999) : Le recuit simulé, La méthode P.A.S.A (Pareto Archived Simulated Annealing), La méthode M.O.S.A (Multiple Objective Simulated Annealing), La recherche Tabou et Les algorithmes génétiques (La méthode Multiple Objective Genetic Algorithm (M.O.G.A.), La méthode Non dominated Sorting Genetic Algorithm (N.S.G.A.), La méthode Niche Pareto G.A. (N.P.G.A.), La méthode Weighted Average Ranking G.A. (W.A.R.G.A)).
- Les méthodes d'aide à la décision (Maystre.1994) : ELECTRE I, ELECTRE II, ELECTRE IS, ELECTRE TRI et PROMETHEE I.

Dans cette thèse, nous considérons deux méthodes très importantes : la méthode **Normal Boundary Intersection** ou NBI développée par DAS et DENNIS en 1996 (Deb.1996) (Shukla.2007), et la méthode **Normalized Normal Constraint method**. Cette dernière, couramment appelée NNCM est une méthode développée par Messac et Mattson en 2004 (Messac.2004) (Martinez.2007).

1.2.1 La méthode Normal Boundary Intersection

La méthode Normal Boundary Intersection (NBI) permet de construire une approximation consistante du Front de Pareto, donnant en général une répartition bien uniforme des points sur le Front. Nous commençons par donner une description rapide des notions de base utilisées dans cette méthode.

Notions de base NBI

En principe, la méthode NBI se base sur les six étapes suivantes :

*Individual Minima : nous obtenons ces points, à partir de la résolution de n sous problème d'optimisation ($P(i)$) de notre problème d'optimisation initial (POMO).

$$(P(i))$$

$$\min_x f_i(x); \quad i = 1; \dots; n$$

$$s/c \begin{cases} g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases} \quad (1.4)$$

Pour chaque sous problème ($P(i)$), nous avons un x_i^* comme solution, alors le point anchor est le suivant :

$$F_i^* = [f_1(x_i^*), f_2(x_i^*), \dots, f_n(x_i^*)] \quad (1.5)$$

Après la résolution de tous les sous problèmes, nous obtenons n points qui présentent la notion de **Anchor points**.

Remarque : En reliant l'ensemble des n points, nous avons un hyperplan nommé **Convex Hull of Individual Minima** ou CHIM.

*Matrice de Shadow Minima : c'est une matrice d'ordre $n \times 1$ contenant les n valeurs minimales de chaque fonction objectif.

$$F^* = [f_1(x_1^*), f_2(x_2^*), \dots, f_n(x_n^*)] \quad (1.6)$$

*Matrice de Pay-off : la matrice ϕ est d'ordre $n \times n$ (n est le nombre des fonctions objectif) et elle a comme composantes de la i eme ligne les éléments :

$$F(x_i^*) - F^*$$

Alors la matrice sera de la forme suivante :

$$\phi = \begin{pmatrix} 0 & f_1(x_2^*) - f_1(x_1^*) & \dots & f_1(x_n^*) - f_1(x_1^*) \\ f_2(x_1^*) - f_2(x_2^*) & 0 & \dots & \dots \\ \dots & f_3(x_2^*) - f_3(x_3^*) & \dots & \dots \\ \dots & \dots & \dots & f_{n-1}(x_n^*) - f_{n-1}(x_{n-1}^*) \\ f_n(x_1^*) - f_n(x_n^*) & f_n(x_2^*) - f_n(x_n^*) & \dots & 0 \end{pmatrix}$$

*L'hyperplan CHIM :

$$CHIM = \left\{ \phi\beta; \beta = (\beta_1, \beta_2, \dots, \beta_n)^T \text{ avec } \sum_{i=1}^n \beta_i = 1, \beta_i \geq 0. \right\} \quad (1.7)$$

*Le vecteur normal :

$$\mathbf{n} = -\phi.e \quad \text{et} \quad e = (1, 1, 1, \dots, 1)^T. \quad (1.8)$$

*L'identification de front de pareto : pour chaque point de notre hyperplan, nous aurons un point de pareto correspondant que nous obtenons par la résolution du problème suivant :

$$(P_{\text{pareto}} \text{ NBI})$$

$$\max_{x,t} \quad t \quad (1.9)$$

$$s/c \begin{cases} \phi.\beta + t.\mathbf{n} = F(x) - F^* \\ g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{\text{inf}}} \leq x_i \leq a_{i_{\text{sup}}} & i = 1; \dots; m \end{cases}$$

Méthode pour la génération des points β_i

Cette méthode a comme finalité la génération des points constituant l'hyperplan *CHIM*, l'utilisateur choisissant le nombre exact de ces points. Dans ce sens, il faut signaler que la génération de ces points n'est qu'une simple génération des valeurs β_{kj} .

* Soit m_i le nombre de points discrétisant le segment $[F_i^* F_{i+1}^*]$ et $K_i = \frac{1}{m_i - 1} \quad i = 1, \dots, n - 1$.

Alors β_{1j} est la discrétisation uniforme suivante :

$$\beta_{1j} \in \{0; K_1; 2K_1; \dots; k_1 K_1 = 1\}$$

avec

$$k_1 = m_1 - 1$$

* Pour chaque valeur β_{1j} , nous pouvons affecter les valeurs β_{2j} suivantes :

$$\beta_{2j} \in \{0; K_2; 2K_2; \dots; k_2 K_2\}$$

avec

$$K_2 = \frac{1}{m_2 - 1}$$

$$k_2 = E\left(\frac{1 - \beta_{1j}}{K_2}\right)$$

* De même, les valeurs β_{3j} correspondantes à chacune des valeurs β_{1j} et β_{2j} sont les suivantes :

$$\beta_{3j} \in \{0; K_3; 2K_3; \dots; k_3 K_3\}$$

avec

$$K_3 = \frac{1}{m_3 - 1}$$

$$k_3 = E\left(\frac{1 - \beta_{1j} - \beta_{2j}}{K_3}\right)$$

* Nous continuons avec la même technique jusqu'aux valeurs β_{n-1j} correspondantes à chacune des valeurs β_{1j} et β_{2j} et... et β_{n-1j} et qui seront les valeurs suivantes :

$$\beta_{n-1j} \in \{0; K_{n-1}; 2K_{n-1}; \dots; k_{n-1}K_{n-1}\}$$

avec

$$K_{n-1} = \frac{1}{m_{n-1} - 1}$$

$$k_{n-1} = E\left(\frac{1 - \sum_{i=1}^{n-2} \beta_{ij}}{K_{n-1}}\right)$$

*Enfin , nous avons la valeur β_{nj} à l'aide de la formule :

$$\beta_{nj} = 1 - \sum_{i=1}^{n-1} \beta_{ij}$$

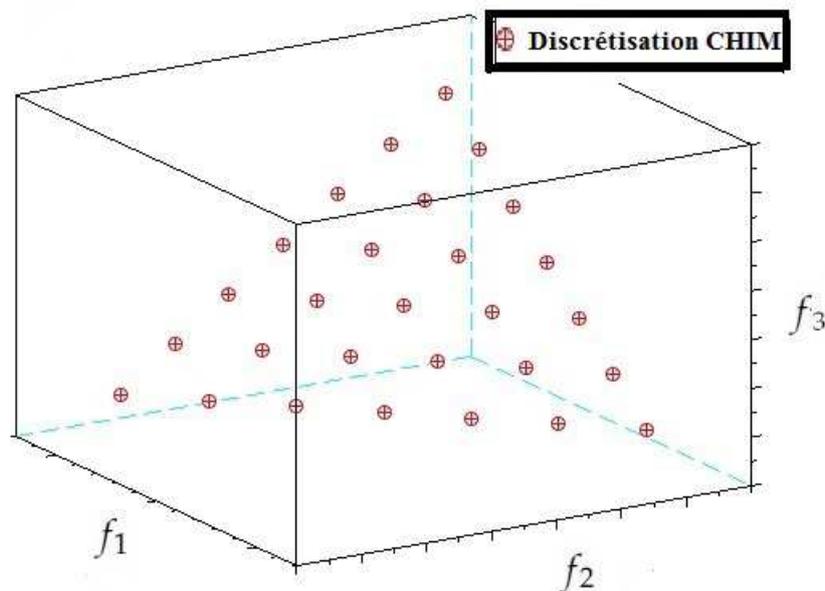


FIGURE 1.6 – Génération des éléments CHIM pour un problème à trois critères ($n=3$) et une discrétisation à 7 points ($m_i=7$)

Pour mieux illustrer la méthode NBI, nous faisons appel à un problème d'optimisation biobjectif que nous allons résoudre par NBI, chaque étape

figure dans le schéma suivant :

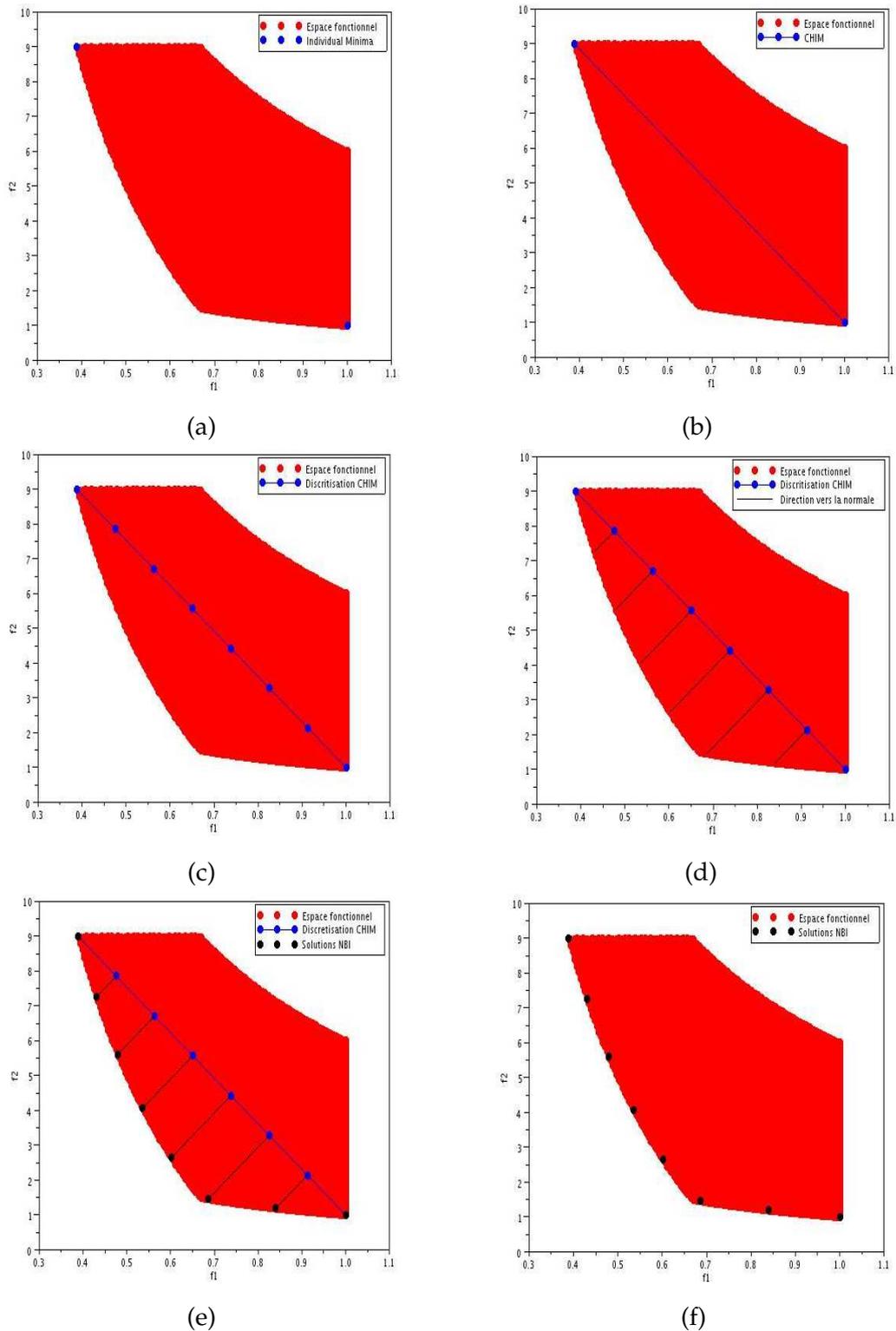


FIGURE 1.7 – Les étapes de la capture du front de Pareto par la Méthode NBI : (a) Calcul des minima individuels, (b) Détermination d'enveloppe convexe ou CHIM, (c) Discretisation du CHIM, (d) Détermination du champ des directions normales, (e) Maximisation de l'amplitude descente le long des normales et (f) Capture de front de Pareto

1.2.2 La méthode Normalized Normal Constraint

Bien qu'ayant le même objectif que la méthode précédente, celui de capturer le mieux possible le Front de Pareto, la méthode Normalized Normal Constraint (NNCM) diffère sensiblement de la méthode NBI. Nous allons commencer par présenter les notions de base de cette méthode.

Notions de base NNCM

En principe, la méthode NNCM se base sur les sept étapes suivantes :

*Comme la méthode NBI, la première étape de la méthode NNCM consiste à résoudre les n sous problèmes ($P(i)$) (problème (1.4)) afin d'identifier les points Anchor.

*Les fonctions objectif normalisées : on fait un changement d'échelle, de l'espace normal à l'espace normalisé, et pour cela, nous introduisons la notion des fonctions objectif normalisées \bar{f}_i .

Soient f^U et f^N , le **point Utopia** et **point Nadir** définis respectivement comme suit :

$$f^U = [f_1(x_1^*), f_2(x_2^*), \dots, f_n(x_n^*)] \quad (1.10)$$

$$f^N = [f_1^N, f_2^N, \dots, f_n^N] \quad (1.11)$$

Avec

$$f_i^N = \max[f_i(x_1^*), f_i(x_2^*), \dots, f_i(x_n^*)]$$

En nous basant sur ces deux derniers points, nous définissons la matrice L des longueurs de la manière suivante :

$$L = f^N - f^U \quad (1.12)$$

L est donc de la forme :

$$L = [L_1, L_2, \dots, L_n]$$

Avec

$$L_i = f_i^N - f_i(x_i^*)$$

Maintenant, nous pouvons définir les n fonctions objectif normalisées :

$$\bar{f}_i(x) = \frac{f_i(x) - f_i(x_i^*)}{L_i} \quad i = 1; \dots; n \quad (1.13)$$

*les vecteurs hyperplan : pour un problème d'optimisation multiobjectif (n fonctions), nous avons $(n-1)$ vecteurs hyperplan à définir comme suit :

$$\bar{N}_k = \bar{F}_n^* - \bar{F}_k^* \quad k = 1; \dots; n - 1 \quad (1.14)$$

Avec

$$\bar{F}_i^* = [\bar{f}_1(X_i^*), \bar{f}_2(X_i^*), \dots, \bar{f}_n(X_i^*)]$$

*La discrétisation des vecteurs hyperplan : nous commençons tout d'abord par la discrétisation du premier vecteur hyperplan, en choisissant le nombre de points m_1 que nous voulons, et nous obtenons $m_1 - 1$ segments de même longueur :

$$K_1 = \frac{1}{m_1 - 1}$$

Et à partir de m_1 , nous pouvons déduire le nombre de points pour les autres vecteurs hyperplan selon la formule suivante :

$$m_k = m_1 \frac{\|\bar{N}_k\|}{\|\bar{N}_1\|} \quad k = 1; \dots, n - 1$$

*La génération des points constituant l'hyperplan : nous obtenons l'ensemble de ces points à l'aide de la formule classique suivante :

$$\bar{X}_{kj} = \sum_{k=1}^n \beta_{kj} \bar{F}_k^*$$

Avec

$$\sum_{k=1}^n \beta_{kj} = 1 \quad \text{et} \quad 0 \leq \beta_{kj} \leq 1$$

Remarque : Pour la génération des points \bar{X}_{kj} , nous nous basons sur la méthode utilisée pour la génération des points β_i (section NBI).

*L'identification de front de pareto : pour chaque point de notre hyperplan, nous identifions un point du Front de Pareto correspondant que nous obtenons par la résolution du problème suivant :

$$\begin{aligned} & (P_{pareto} \text{ NNCM}) \\ & \min_x \bar{f}_d(x); \quad d = n \\ & s/c \begin{cases} g_j(X) \geq 0 & j = 1; \dots; J \\ h_k(X) = 0 & k = 1; \dots; K \\ a \leq X_i \leq b & i = 1; \dots; m \\ \bar{N}_k(\bar{f} - \bar{X}_{kj})^T \leq 0 & k = 1; \dots; n - 1 \\ \text{avec } \bar{f} = [\bar{f}_1(x), \bar{f}_2(x), \dots, \bar{f}_n(x)] \end{cases} \end{aligned} \quad (1.15)$$

*Retour à l'espace normal : après avoir identifié notre front sur l'espace normalisé, nous devons le ramener dans l'espace normal, et cela en faisant un transfert entre les deux espaces à l'aide de la formule simple suivante :

$$f_i(x) = \bar{f}_i(x) \times L_i + f_i(x_i^*) \quad i = 1; \dots; n \quad (1.16)$$

De même, nous faisons appel à un problème d'optimisation biobjectif que nous allons résoudre par NNCM afin de bien illustrer les étapes clés de la méthode NNCM.

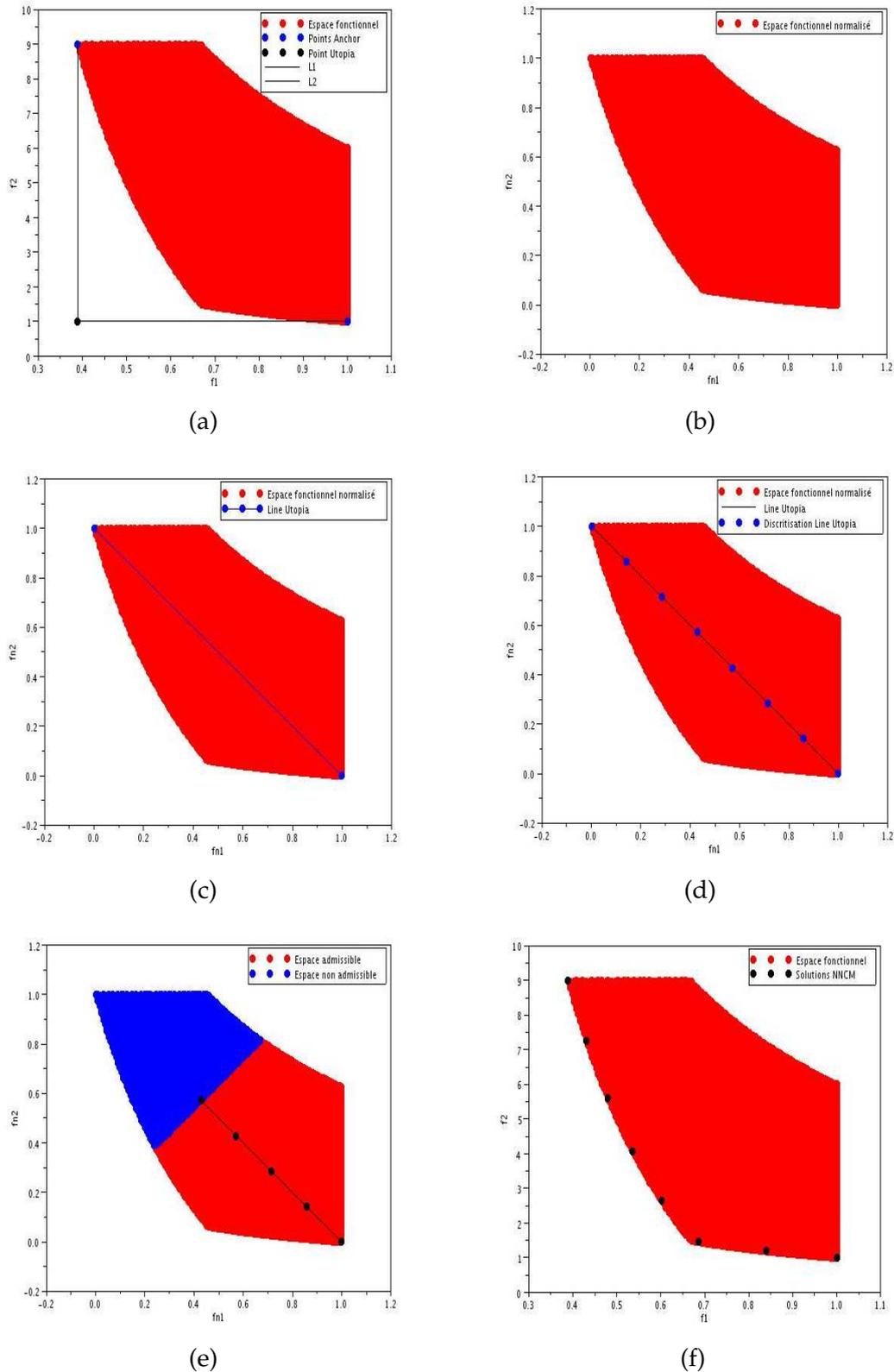


FIGURE 1.8 – Les étapes de la capture du front de Pareto par la Méthode NNCM : (a) Calcul des minima individuels et le point u^{Topia} , (b) Passage à l'espace normalisé à l'aide des longueurs L_1 et L_2 , (c) Détermination de la ligne utopia, (d) Discrétisation de la ligne utopia, (e) Minimisation de la fonction f_2 normalisée et (f) Capture de front de Pareto

1.2.3 Validation

Après une présentation détaillée du principe de deux méthodes NBI et NNCM pour le cas général, nous allons les valider dans le cadre des problèmes bicritère (vu que tous les problèmes traités durant cette thèse rentrent dans ce cadre d'étude). Pour ce faire, nous allons nous baser sur les algorithmes (5 3). Cette validation s'applique à un ensemble de tests académiques classiques et va montrer par la suite aussi bien les avantages que les inconvénients des deux méthodes.

Le logiciel utilisé pour la programmation des deux méthodes est *Scilab*.

Algorithm 3 NNCM - cas bicritère -

- 1: $x_i^* = \min_x f_i(x)$ $i = 1 : 2$ \triangleright Minimisation de chaque fonction objectif
- 2: $F_i^* = [f_1(x_i^*), f_2(x_i^*)]^T$ $i = 1 : 2$ \triangleright Individual minima
- 3: $F^u = [f_1(x_1^*), f_2(x_2^*)]^T$ \triangleright Point Utopia
- 4: $F^n = [f_1(x_2^*), f_2(x_1^*)]^T$ \triangleright Point Nadir
- 5: $L = F^u - F^n$ \triangleright Matrice des longueurs
- 6: $\bar{f}_i(x) = \frac{f_i(x) - f_i(x_i^*)}{L_i}$ $i = 1 : 2$ \triangleright Fonctions objectif normalisées
- 7: $\bar{F}_i^* = [\bar{f}_1(x_i^*), \bar{f}_2(x_i^*)]^T$ $i = 1 : 2$
- 8: $\bar{N}_1 = \bar{F}_2^* - \bar{F}_1^*$ $i = 1 : 2$
- 9: $\bar{X}_{kj} = \sum_{k=1}^2 \beta_k \bar{F}_k^*$, $\beta_i \geq 0$, $\sum_{i=1}^2 \beta_i = 1$ \triangleright Discrétisation Utopia
- 10: Pour chaque $\beta = (\beta_1, \beta_2)^T$ \triangleright Généré par l'algorithme 4
- 11: On résoudre le problème :

$$\min_x \bar{f}_2(x)$$

$$s/c \begin{cases} g_j(X) \geq 0 & j = 1; \dots; J \\ h_k(X) = 0 & k = 1; \dots; K \\ a \leq X_i \leq b & i = 1; \dots; m \\ \bar{N}_1(\bar{f} - \bar{X}_{kj})^T \leq 0 \\ avec \bar{f} = [f_1(x), f_2(x)] \end{cases}$$

Algorithm 4 Une discrétisation uniforme de l'enveloppe convexe (CHIM) en h points

- 1: **Pour** $i = 1 : h$
 - 2: $\beta_{i1} = \frac{i-1}{h-1}, \beta_{i2} = 1 - \beta_{i1}$
 - 3: **Fin Pour**
-

Algorithm 5 NBI - cas bicritère -

- 1: $x_i^* = \min_x f_i(x)$ $i = 1 : 2$ ▷ Minimisation de chaque fonction objectif
- 2: $F_i^* = [f_1(x_i^*), f_2(x_i^*)]^T$ $i = 1 : 2$ ▷ Individual minima
- 3: $F^* = [f_1(x_1^*), f_2(x_2^*)]^T$ ▷ Shadow Minima
- 4: ϕ ▷ La matrice Pay-off d'ordre(2 × 2)
- 5: $\phi_i = F(x_i^*) - F^*$ ▷ La i^{eme} colonne de ϕ
- 6: $\mathbf{n} = -\phi.e, e = (1, 1)^T$ ▷ vecteur normal
- 7: $\beta = (\beta_1, \beta_2)^T, \beta_i \geq 0, \sum_{i=1}^2 \beta_i = 1$ ▷ Discrétisation CHIM
- 8: Pour chaque $\beta = (\beta_1, \beta_2)^T$ ▷ Généré par l'algorithme 4
- 9: On résoudre le problème :

$$s/c \begin{cases} \max_{x,t} t \\ \phi.\beta + t.\mathbf{n} = F(x) - F^* \\ g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases}$$

Problèmes tests bicritère

Dans cette partie de la validation, nous utilisons une série de problèmes tests bicritère qui diffèrent par leurs fronts de Pareto. Parmi les problèmes traités, nous distinguons ceux sans contraintes, avec contraintes, ayant des fronts convexes, concaves ou discontinus.

Les deux méthodes, NBI et NNCM, sont validées sur des problèmes d'optimisation multicritère connus et difficiles tels que SCH2, Min-Ex, POL, TNK, FON et P4.

Pour la résolution de ces problèmes, nous avons choisi :

- Le toolbox **Fmincon** de scilab comme optimiseur pour la détermination des minima individuels et la maximisation le long des normales du CHIM.
- $h = 25$ et $h = 100$ comme nombre de solutions souhaité pour les deux méthodes NBI et NNCM.

Schaffer 2 -SCH2-

Ce problème (Deb.2001) est proposé par Schaffer. On minimise deux

fonctions coût avec une seule variable. La solution est un front de Pareto discontinu. Le problème est donné par :

$$\begin{aligned}
 & (SCH2) \\
 \min_x F(x) = & \begin{cases} f_1(x) = \begin{cases} -x & \text{si } x \leq 1 \\ x - 2 & \text{si } 1 < x \leq 3 \\ 4 - x & \text{si } 3 < x \leq 4 \\ x - 4 & \text{si } x > 4 \end{cases} \\ f_2(x) = (x - 5)^2 \end{cases} & (1.17) \\
 \text{s/c } \{ & -5 \leq x \leq 10
 \end{aligned}$$

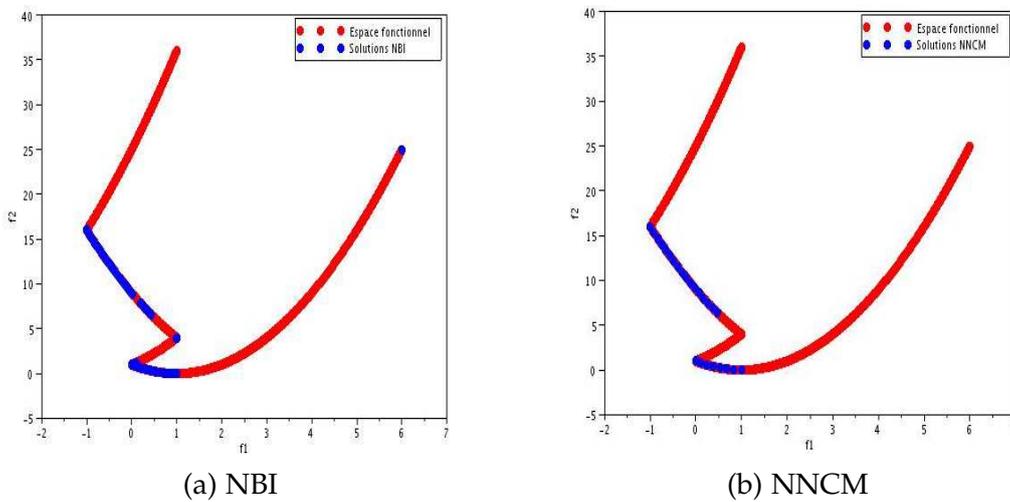


FIGURE 1.9 – Résolution du problème SCH2 par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

Minimization Example -Min-Ex-

Le problème de minimisation Min-Ex porte sur deux fonctions coût et deux variables. Il est proposé par Deb (Deb.2001), et a comme solution un front de Pareto convexe. Il est défini comme suit :

$$\begin{aligned}
 & (Min - Ex) \\
 \min_{x=(x_1, x_2)} F(x) = & \begin{cases} f_1(x) = x_1 \\ f_2(x) = \frac{1+x_2}{x_1} \end{cases} & (1.18) \\
 \text{s/c } \{ & \begin{cases} 0.1 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 5 \end{cases}
 \end{aligned}$$

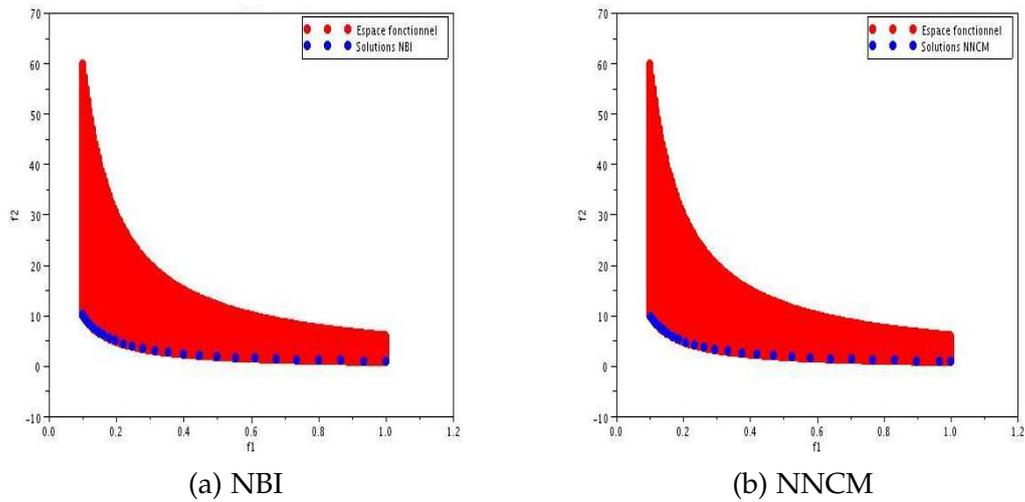


FIGURE 1.10 – Résolution du problème Min-Ex par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

Tanaka -TNK-

Tanaka (Tanaka.1995) propose un problème d'optimisation bicritère très compliqué à deux variables et deux fonctions à minimiser en présence de deux contraintes, il a comme solution un front de Pareto discontinu, le problème est défini comme suit :

$$\begin{aligned}
 & (TNK) \\
 & \min_{x=(x_1, x_2)} F(x) = \begin{cases} f_1(x) = x_1 \\ f_2(x) = x_2 \end{cases} \\
 & s/c \begin{cases} x_1^2 + x_2^2 - 1 - 0.1 \cos(16 \arctang(\frac{x_1}{x_2})) \geq 0 \\ (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5 \\ 0 \leq x_1, x_2 \leq \pi \end{cases}
 \end{aligned} \tag{1.19}$$

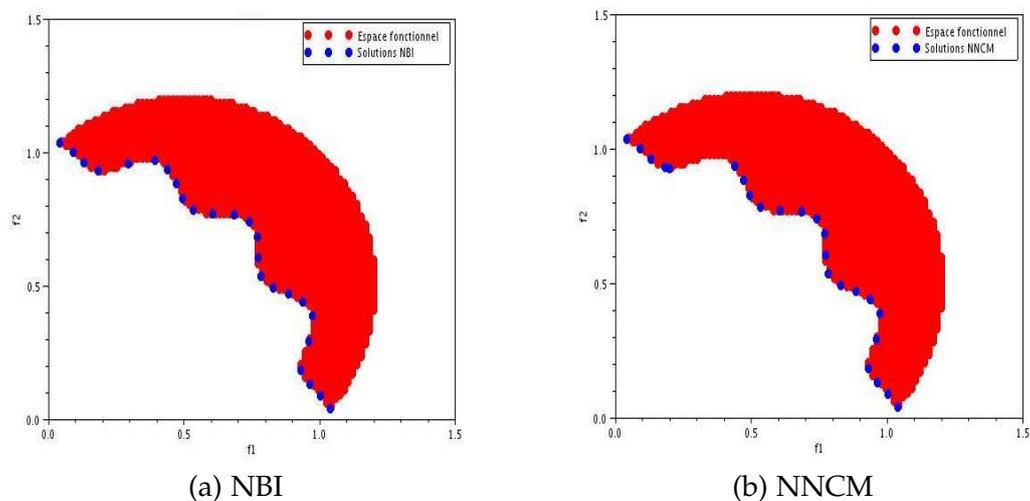


FIGURE 1.11 – Résolution du problème TNK par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

Poloni -POL-

Poloni (Cerny.2000) a utilisé un problème ayant deux variables et deux fonctions à minimiser. Ce problème a un ensemble de Pareto non convexe et discontinu qui est difficile à trouver. Il est défini comme suit :

$$\begin{aligned}
 & \text{(POL)} \\
 & \min_{x=(x_1, x_2)} F(x) = \begin{cases} f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2 \\ f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2 \end{cases} \\
 & \text{s/c } \begin{cases} A_1 = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2) \\ A_2 = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2) \\ B_1 = 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2) \\ B_2 = 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2) \\ -\pi \leq x_1, x_2 \leq \pi \end{cases} \quad (1.20)
 \end{aligned}$$

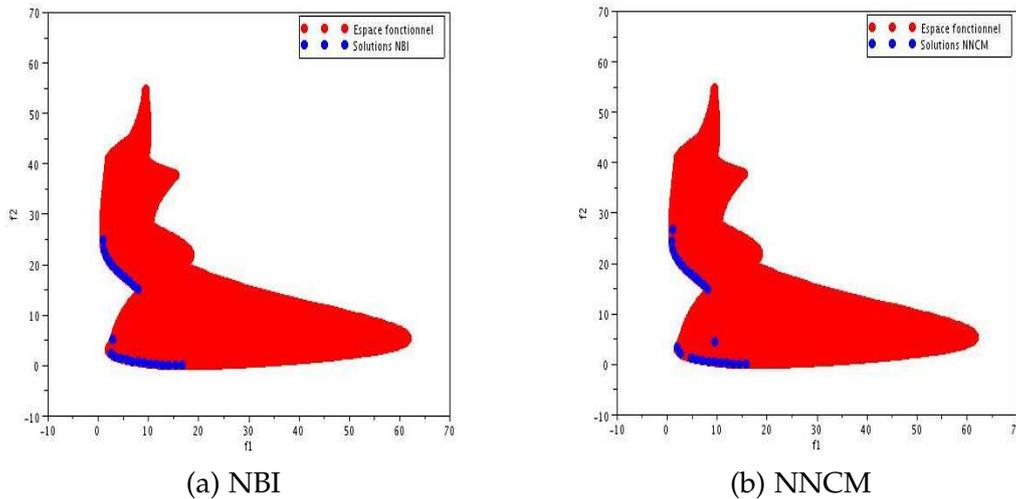


FIGURE 1.12 – Résolution du problème POL par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

Fonseca -FON-

Fonseca (Fonseca.1997) a utilisé un problème ayant trois variables et deux fonctions à minimiser. Ce problème a un ensemble de Pareto concave. Il est donné par :

$$\begin{aligned}
 & \text{(FON)} \\
 & \min_{x=(x_1, x_2, x_3)} F(x) = \begin{cases} f_1(x) = 1 - \exp(-\sum_{i=1}^3 (x_i - \frac{1}{\sqrt{3}})) \\ f_2(x) = 1 - \exp(-\sum_{i=1}^3 (x_i + \frac{1}{\sqrt{3}})) \end{cases} \quad (1.21) \\
 & \text{s/c } \{ -4 \leq x_1, x_2, x_3 \leq 4
 \end{aligned}$$

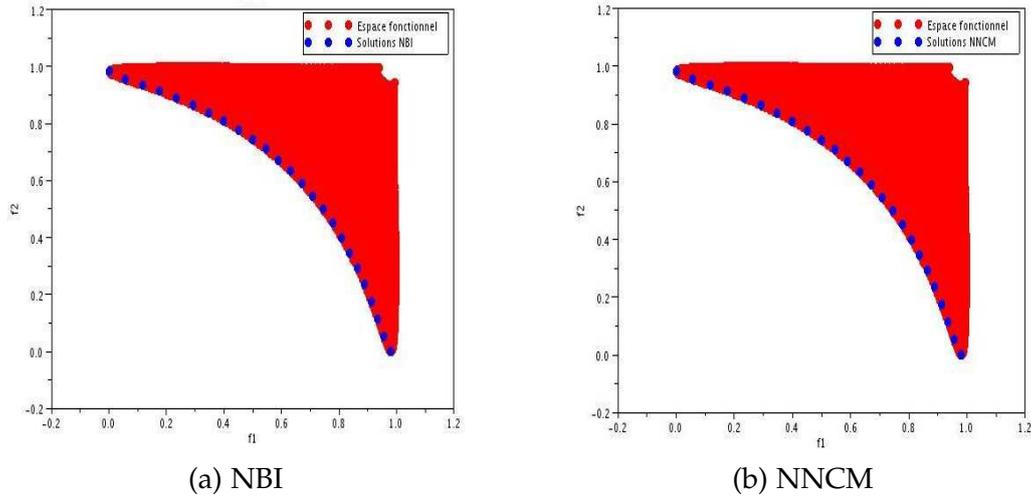


FIGURE 1.13 – Résolution du problème FON par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

Problème à quatre variables -P4-

P4 est un problème ayant quatre variables et deux fonctions à minimiser, et a comme solution un front de Pareto non convexe, il est défini comme suit :

$$\begin{aligned}
 & (P4) \\
 \min_{x=(x_1, x_2, x_3, x_4)} F(x) = & \begin{cases} f_1(x) = -(x_1 - x_3)^4 + (x_2 - x_4)^4 \\ f_2(x) = (x_1 - x_3)^2 + (x_2 - x_4 + 1)^2 \end{cases} & (1.22) \\
 \text{s/c } & \{ 0 \leq x_1, x_2, x_3, x_4 \leq 1
 \end{aligned}$$

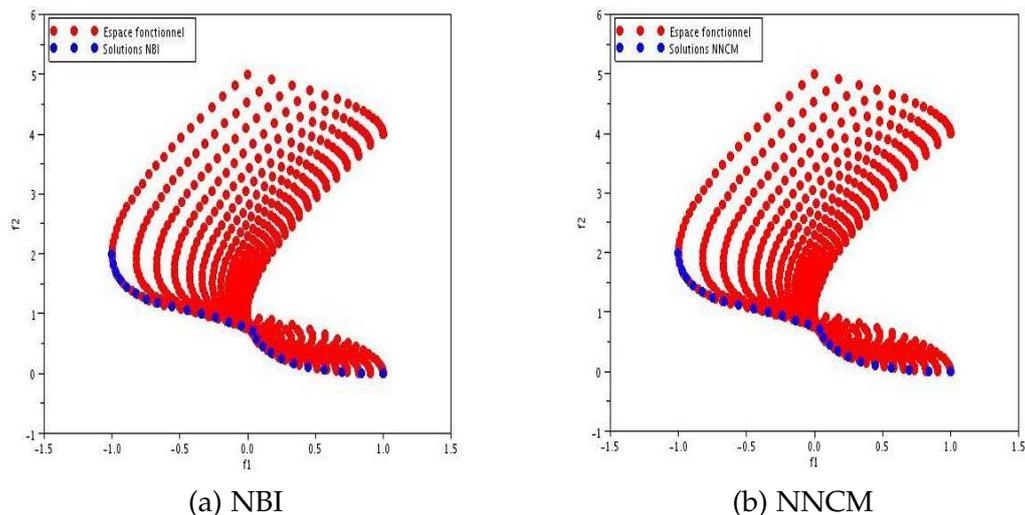


FIGURE 1.14 – Résolution du problème P4 par les méthodes (a) NBI et (b) NNCM, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

Les deux méthodes utilisées arrivent à correctement approcher les problèmes ayant un front de Pareto continu (Figures 1.10 1.13 1.14). Cependant, lorsqu'il s'agit d'un front discontinu, elles rajoutent des solutions

dominées (Figures 1.9 1.11 1.12). Ceci est un premier point faible de NBI et NNCM. Un autre point faible est relatif au grand nombre d'évaluations des fonctions objectif effectuées lors de la résolution du problème (Tableau 1.1).

Remarque : Les solutions dominées générées ne résultent généralement pas toujours de la méthode. Il arrive des fois qu'elles proviennent de défaillance de l'optimiseur **Fmincon** utilisé (Figure 1.11.NNCM).

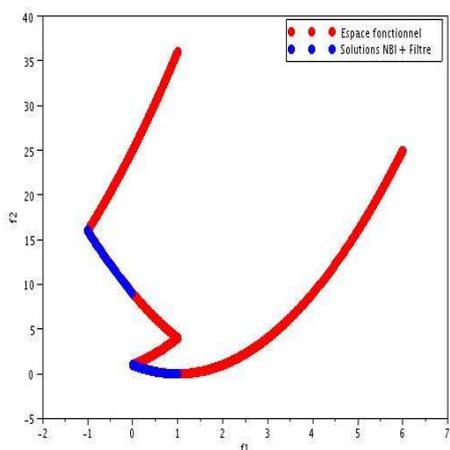
Ceci n'empêche pas que les deux approches présentent un avantage majeur qui est la génération d'un ensemble uniforme de points Pareto représentatifs de l'ensemble des point Pareto et non pas un nombre de points concentrés dans une région donnée, ce qui permet de construire une bonne approximation du front de Pareto.

Problème	Méthode	nb. solutions	nb. évaluations
SCH ₂	NBI	25	2432
	NNCM	25	2670
	NBI	100	10216
	NNCM	100	10518
Min-Ex	NBI	25	403
	NNCM	25	449
	NBI	100	1575
	NNCM	100	1741
TNK	NBI	25	497
	NNCM	25	620
	NBI	100	1905
	NNCM	100	2324
POL	NBI	25	554
	NNCM	25	603
	NBI	100	2008
	NNCM	100	2151
FON	NBI	25	305
	NNCM	25	399
	NBI	100	967
	NNCM	100	1339
P ₄	NBI	25	448
	NNCM	25	508
	NBI	100	1782
	NNCM	100	1912

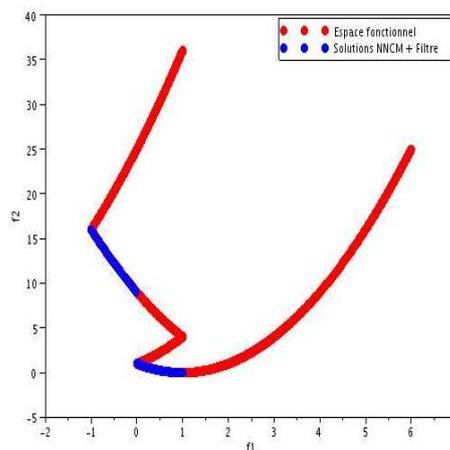
Tableau 1.1 – Nombre d'évaluations des fonctions critère effectuées par les méthodes NBI et NNCM lors de la résolution des problèmes SCH₂, Min-Ex, POL, TNK, FON et P₄

Ayant en vue l'optimisation et l'atténuation des inconvénients, il est possible d'utiliser un filtre pour éliminer les solutions dominées (Figures 1.15) (Messac.2003) (Messac.2002) et intégrer les métamodèles dans la boucle d'optimisation pour pouvoir diminuer le grand nombre d'évalua-

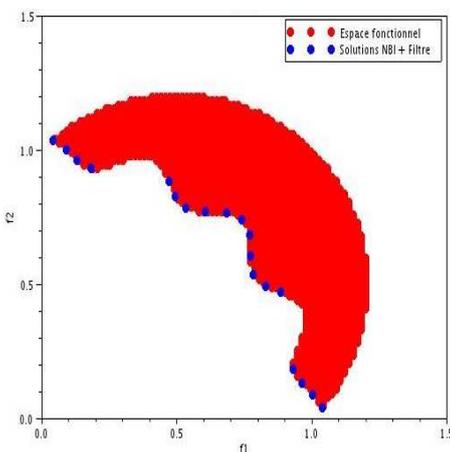
tions des fonctions objectif exactes (l'objet de la prochaine section et du prochain chapitre).



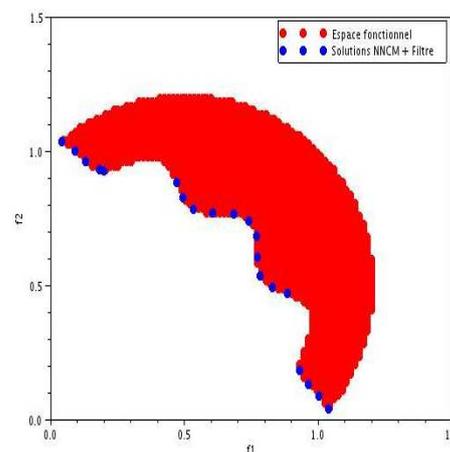
(a) SCH2 par NBI + Filre



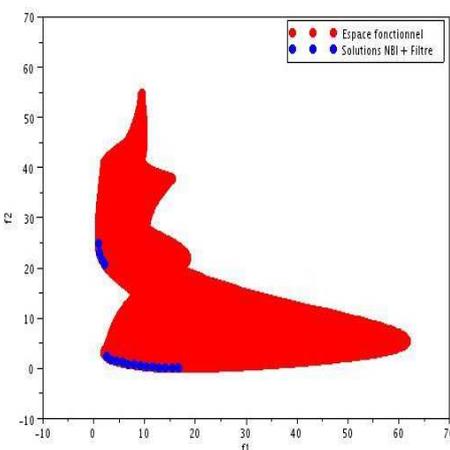
(b) SCH2 par NNCM + Filre



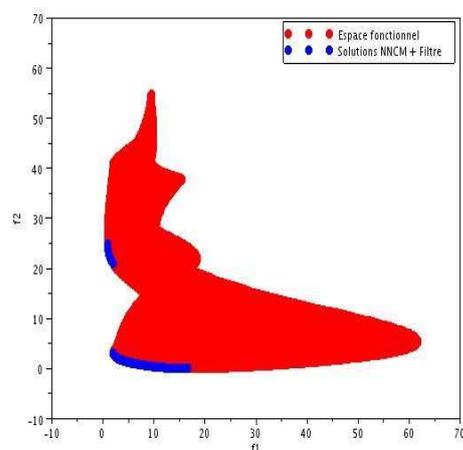
(a) TNK par NBI + Filre



(b) TNK par NNCM + Filre



(a) POL par NBI + Filre



(b) POL par NNCM + Filre

FIGURE 1.15 – Usage d'un filtre pour éliminer les solutions dominées obtenues par les méthodes (a) NBI et (b) NNCM pour les problèmes SCH2, TNK et POL

1.3 NECESSITÉ ET REVUE DES MÉTAMODÈLES

1.3.1 Généralités

Dans les problèmes d'optimisation multicritère, nous devons chercher l'ensemble de front de Pareto, ce qui nécessite de nombreuses évaluations des fonctions objectif à minimiser, ce qui peut être très coûteux en temps de calcul. C'est pour contourner cet obstacle qu'il a été proposé, depuis plusieurs années, de remplacer une partie des évaluations de la fonction à optimiser par un modèle mathématique construit à partir d'un échantillon de valeurs (valeurs exactes de ces fonctions en un certain nombre de points, connus sous le nom de points maîtres). Ce modèle a souvent le nom de métamodèle (Figure 1.16).

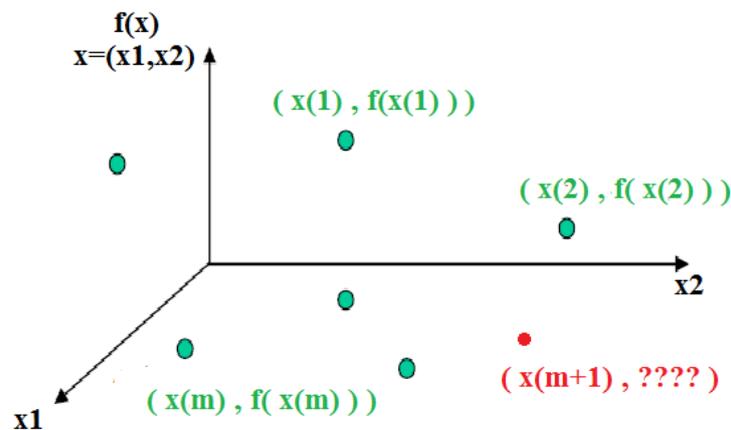


FIGURE 1.16 – Ojectif des métamodèles : approximation de la fonction exacte f en un point donné $x(m+1)$ (point rouge) à partir d'un échantillon de m points connus $(x(1), \dots, x(m))$ (points verts)

Dans la littérature, nous trouvons plusieurs types de métamodèles, parmi lesquels, les plus utilisés sont :

- **La méthode des surfaces de réponse** : 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). Dans une démarche d'analyse en surface de réponse de type polynomiale, la réponse f à une entrée multidimensionnelle $x = (x_1, \dots, x_k)$ constituée des k variables de conception x_i est généralement modélisée par un polynôme du second degré :

$$f = F(x) + e$$

Avec :

e est un terme d'erreur ayant une distribution normale avec une

moyenne nulle ($E(e) = 0$) et une variance égal à σ^2 et $F(x)$ défini comme suit :

$$F(x) = a_0 + \sum_{i=1}^k a_i x_i + \sum_{i=1}^k \sum_{j=1}^k a_{ij} x_i x_j \quad (1.23)$$

Avec un changement de variable au niveau du troisième terme de la formule 1.23, on peut écrire $F(x)$ sous la forme suivante :

$$F(x) = G(y) = a_0 + \sum_{i=1}^h a_i y_i, \quad (h > k) \quad (1.24)$$

où les y_i contiennent les parties quadratiques $x_i x_j$. Par la suite, nous notons (F, x) le couple (G, y) .

Supposons maintenant que nous avons un échantillon de m points connus $(x^i, f(x^i))$, alors chaque réponse f_i peut s'écrire comme suit :

$$f_i = a_0 + \sum_{j=1}^h a_j x_{ij} + e_i, \quad \forall i = 1, \dots, m \quad (1.25)$$

avec x_{ij} est la composante j du point x^i .

les coefficients a_i sont appelés les coefficients de régression.

Sous forme matricielle, l'équation (1.25) s'écrit :

$$\mathbf{f} = \mathbf{X}\mathbf{a} + \mathbf{e} \quad (1.26)$$

avec :

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_m \end{bmatrix}, \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ a_h \end{bmatrix}, \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ e_m \end{bmatrix} \text{ et } \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdot & x_{1h} \\ 1 & x_{21} & x_{22} & \cdot & x_{2h} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_{m1} & x_{m2} & \cdot & x_{mh} \end{bmatrix}.$$

Une valeur approchée \mathbf{a} , du vecteur des coefficients de régression peut être déterminée en minimisant la fonction des moindres carrés de l'erreur, \mathbf{L} , par rapport aux $a_{i=0, \dots, h}$:

$$\mathbf{L} = \sum_{j=1}^m e_j^2 = \mathbf{e}^T \mathbf{e} = (\mathbf{f} - \mathbf{X}\mathbf{a})^T (\mathbf{f} - \mathbf{X}\mathbf{a}) \quad (1.27)$$

d'où :

$$\tilde{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{f} \quad (1.28)$$

Mainenant, le métamodèle Surfaces de Réponse (MSR), devient une fonction mathématique explicite qui peut être utilisée pour déterminer une valeur approchée d'un point x^0 :

$$f_{MSR}(x^0) = \mathbf{X}^0{}^T \tilde{\mathbf{a}} \quad (1.29)$$

Avec :

$$\mathbf{X}^0 = [1, x_{11}, x_{12}, \dots, x_{1h}]$$

- **Le krigeage** : Le métamodèle de krigeage est une méthode d'interpolation probabiliste basée sur la minimisation de la variance de l'estimation en un point X donné, proposée par Krige (Krige.1951) et formalisée par Matheron (Matheron.1962) permettant d'approcher une fonction f en un point donné X à partir d'un échantillon de m points connus $(x^i, f(x^i))$.
On suppose par ailleurs que $f(X)$ est une réalisation d'une variable aléatoire $j(X)$ dont la covariance est connue :

$$\text{cov}(j(X); j(Y)) = c(X; Y) \quad (1.30)$$

En cherchant à minimiser $\text{var}(\hat{j}(X) - j(X))$ tout en imposant $E(\hat{j}(X) - j(X)) = 0$, on aboutit à une relation permettant de déterminer $f_{krigeage}(X)$:

$$f_{krigeage}(X) = K^T C^{-1} F \quad (1.31)$$

où K est le vecteur colonne de terme général $c(x^i; X)$, C est la matrice de terme général $c(x^i; x^j)$, et F le vecteur colonne de terme général $f(x^i)$.

La fonction de corrélation est en général choisie comme étant de type exponentielle :

$$\text{cov}(X; Y) = \theta_1 \exp\left(-\frac{1}{2} \sum_{i=1}^k \frac{(X_i - Y_i)^2}{r_i^2}\right) + \theta_2 \quad (1.32)$$

Les paramètres $\Theta = (\theta_1, \theta_2, r_1, \dots, r_k)$ sont alors déterminés par le principe du maximum de vraisemblance, c'est à dire en maximisant la fonction :

$$\mathbf{L}(\Theta) = p(f(x^1), \dots, f(x^m)) = -\frac{1}{\sqrt{(2\pi)^m \det C}} \exp\left(-\frac{1}{2} F^T C^{-1} F\right) \quad (1.33)$$

(Pour plus de détails, voir (Duvigneau.2007) (Ababou.1994)).

- **La méthode des fonctions à base radiale (RBF)** : C'est le type de métamodèle utilisé durant cette thèse, il sera détaillé dans la section ci-dessous (Duvigneau.2007).

1.3.2 Le métamodèle RBF

Pour un ensemble de N observations $(X_i, f(X_i))$, le but est d'avoir une valeur approchée pour un point X_{i+1} qui n'appartient pas à l'ensemble des observations, pour cela, nous construirons une fonction \tilde{f} , cette dernière va nous permettre d'avoir le résultat recherché et cela en nous basant sur le métamodèle de type fonctions à bases radiales RBF.

En général, il y a plusieurs types de fonctions à bases radiales, à savoir :

- Gaussienne $\phi(r) = e^{-\beta r^2}$

- Multiquadratique $\phi(r) = \sqrt{r^2 + \beta^2}$
- Polyharmonique spline

$$\phi(r) = \begin{cases} r^k & \text{si } k = 1; 3; 5; \dots \\ r^k \cdot \ln(r) & \text{si } k = 2; 4; 6; \dots \end{cases}$$

Notre choix s'est orienté vers les fonctions à base radiale de type gaussiennes.

En général, les points X_i où nous disposons des valeurs exactes de la fonction à approcher sont appelés les centres du métamodèle, et les fonctions à bases radiales ϕ_i sont des fonctions de la forme suivante :

$$\phi_i(X) = \phi(\|X - X_i\|)$$

où $\|X - X_i\|$ est la distance entre les points X et X_i en norme euclidienne.

La valeur approchée $\tilde{f}(X)$ est la combinaison des fonctions à bases radiales en utilisant des paramètres ω_i .

$$\tilde{f}(X) = \sum_{i=1}^n \omega_i \phi(\|X - X_i\|)$$

Le choix des paramètres ω_i doit se faire de manière qu'ils vérifient la condition suivante :

$$\tilde{f}(X_i) = f(X_i), i \in \{1; 2; \dots; N\}$$

En se basant sur la formule précédente, nous aurons le système linéaire suivant :

$$A \cdot \omega = F \quad (1.34)$$

Avec :

$$F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} \text{ et } A = \begin{bmatrix} \phi(\|X_1 - X_1\|) & \cdot & \phi(\|X_1 - X_N\|) \\ \phi(\|X_2 - X_1\|) & \cdot & \phi(\|X_2 - X_N\|) \\ \cdot & \cdot & \cdot \\ \phi(\|X_N - X_1\|) & \cdot & \phi(\|X_N - X_N\|) \end{bmatrix}.$$

La famille de fonctions, gaussiennes, que nous avons choisi pour construire notre métamodèle RBF, peut être écrite sous la forme :

$$\phi(r) = e^{-r^2 a_f}$$

a_f est le coefficient d'atténuation.

Il faut signaler que ce coefficient joue un rôle crucial au niveau de la précision de métamodèle. En effet, un mauvais choix peut nous conduire à des valeurs approchées totalement aberrantes (Figures 1.21).

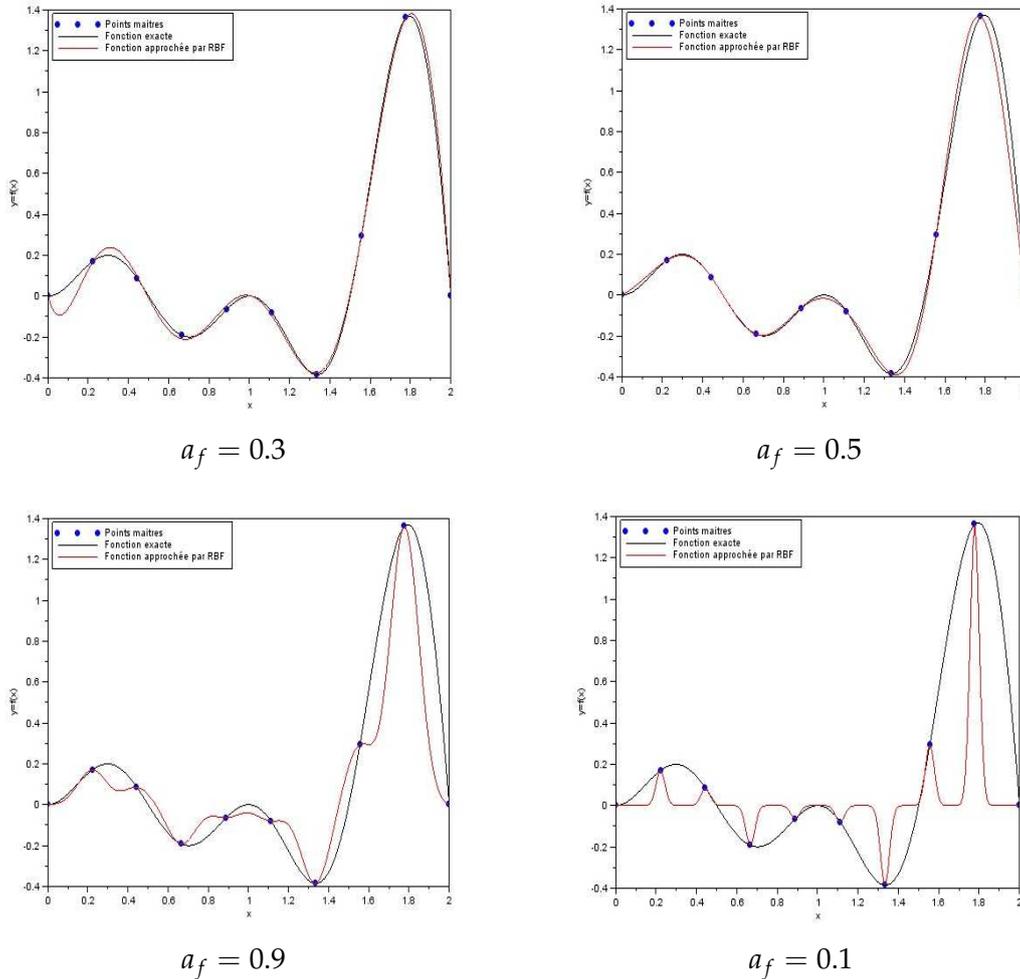


FIGURE 1.17 – Illustration de l’influence du facteur d’atténuation a_f choisi pour le calcul de la fonction approchée (rouge) d’une fonction exacte donnée (noir) par le métamodèle RBF en utilisant différentes valeurs pour ce facteur en gardant toujours les mêmes points maîtres (points bleus) pour la construction du métamodèle

Optimisation du coefficient d’atténuation

La méthode présentée pour l’optimisation du facteur d’atténuation est celle connue par **leave-one-out cross validation (LOOCV)** qui consiste à construire le métamodèle par $(N-1)$ observations puis nous le validons sur la n ème observation et nous répétons cette opération N fois.

Soit $X^{(n)}$ l’ensemble des points $\{X_1, X_2, \dots, X_{n-1}, X_{n+1}, \dots, X_N\}$.

Soit $\tilde{f}^{(n)}(X, a)$ la fonction de l’interpolation RBF construite à base des points $X^{(n)}$:

$$\tilde{f}^{(n)}(X, a) = \sum_{m=1, m \neq n}^N \omega_m^{(n)} \phi(\|X - X_m\|)$$

Soit E_n l'erreur :

$$E_n = f(X_n) - \tilde{f}^{(n)}(X_n, a)$$

Alors, si nous éliminons successivement un point parmi les N points, et nous construisons la fonction d'interpolation RBF, nous aurons un vecteur des erreurs :

$$E(a) = [E_1, E_2, \dots, E_N]^T$$

Notre but est de chercher a_f qui minimise l'ensemble des erreurs obtenues.

$$a_f = \operatorname{argmin} \|E(a)\|$$

Algorithm 6 Optimisation du coefficient d'atténuation

- 1: **Données** : On fixe a réel
 - 2: **Pour** $n = 1 : N$
 - 3: $X^{(n)} = \{X_1, X_2, \dots, X_{n-1}, X_{n+1}, \dots, X_N\}$
 - 4: $\tilde{f}^{(n)}(X, a) = \sum_{m=1, m \neq n}^N \omega_m^{(n)} \phi(\|X - X_m\|)$
 - 5: $E_n = f(X_n) - \tilde{f}^{(n)}(X_n, a)$
 - 6: **Fin Pour**
 - 7: $E(a) = [E_1, E_2, \dots, E_N]^T$
 - 8: $a_f = \operatorname{argmin} \|E(a)\|$
-

Bien que cette mise en oeuvre naïve de l'algorithme leave-one-out (algorithme 6) serait plutôt coûteuse (de l'ordre de N^4), Rippa (Rippa.1999) a montré que l'algorithme peut être simplifié en une seule formule définie comme suit :

$$E_n = \frac{\omega_n}{A^{-1}_{nn}}$$

où ω_n est le n -ième coefficient du vecteur ω défini en (1.34), et A^{-1}_{nn} est le n -ième élément diagonal de l'inverse de la matrice d'interpolation A correspondant.

Choix des points maîtres

Après la discussion sur le facteur d'atténuation et de la méthode Rippa servant à son optimisation, nous allons présenter un autre aspect primordial pour les métamodèles, qui est la manière de choisir les points maîtres. Dans la littérature, nous trouvons plusieurs méthodes pour ce faire, parmi lesquelles :

- Génération aléatoire

- Hypercube latin
- Distribution uniforme

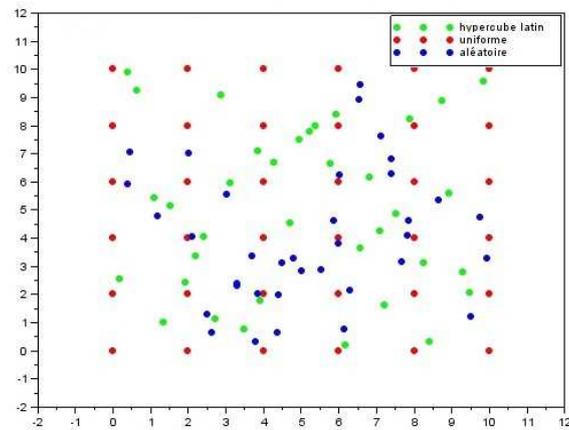


FIGURE 1.18 – Génération des points maîtres par les différentes méthodes citées : Génération aléatoire (bleu), Hypercube latin (vert) et Distribution uniforme (rouge)

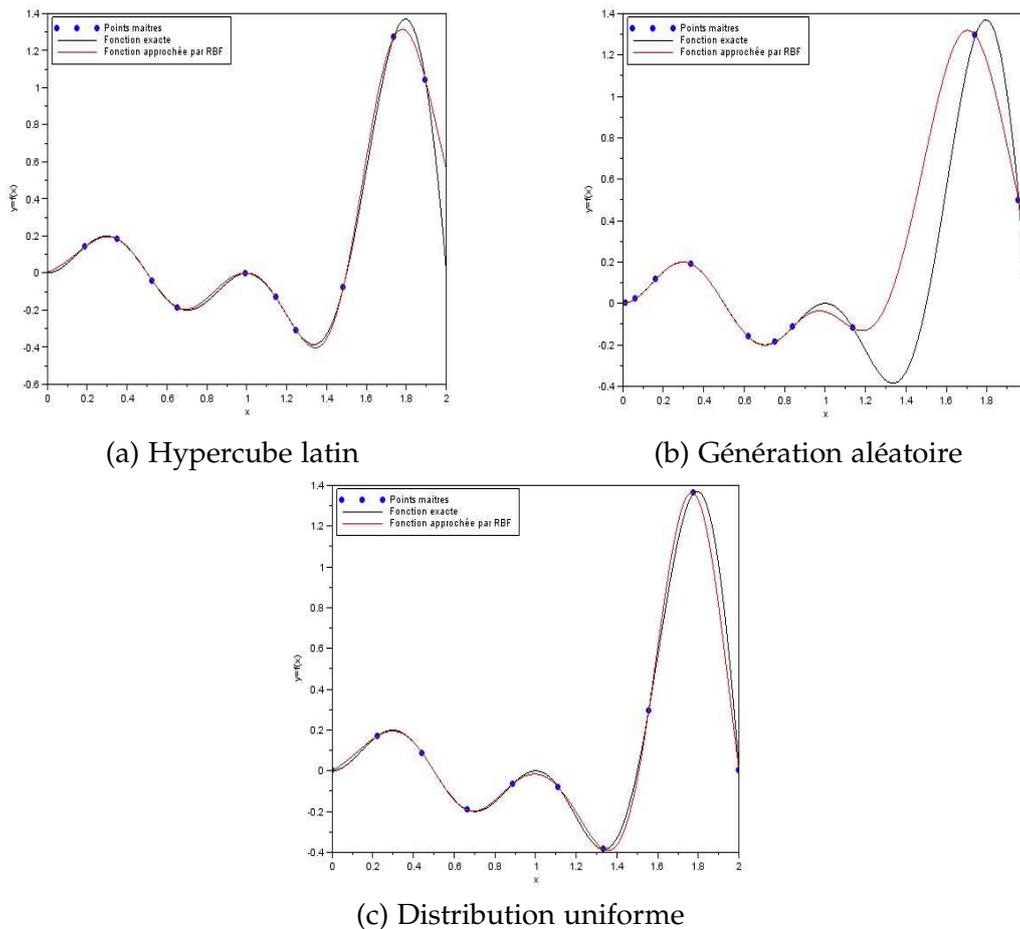


FIGURE 1.19 – Illustration de l'influence du choix des points maîtres générés par les méthodes (a) Hypercube latin, (b) Génération aléatoire et (c) Distribution uniforme, pour le calcul de la fonction approchée (rouge) d'une fonction exacte donnée (noir) par RBF

Les résultats (Figures 1.19) montrent clairement, que le choix des points maîtres à l'aide d'une distribution uniforme ou bien hypercube latin demeure meilleur qu'un choix aléatoire, de même, le choix uniforme est le meilleur (Etude comparative en Annexe).

Pour le reste de notre travail, tous les métamodèles RBF utilisés sont construits à base d'un choix uniforme des points maîtres.

1.3.3 Validation

Après une présentation relativement exhaustive des métamodèles et en particulier, celui du **RBF**, nous allons les valider pour des fonctions test connues comme celles de Rastrigin et Schwefel (Törn.1989).

Pour ce faire, nous allons nous baser sur l'algorithme (7) (implémenté en Scilab).

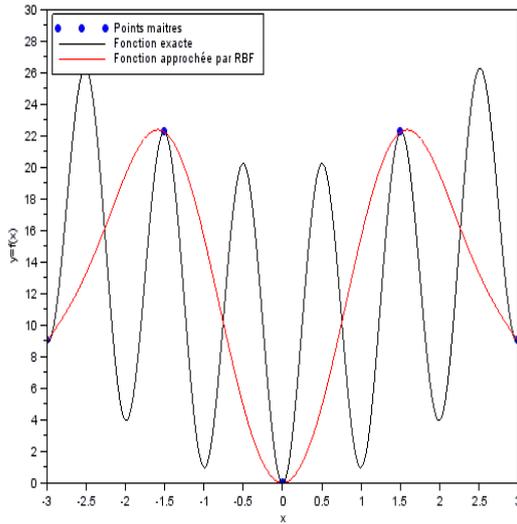
Algorithm 7 RBF

- 1: **Données** : f la fonction à approcher, N le nombre total des points maîtres, ϕ fonction gaussienne et a_0 facteur d'atténuation initial.
 - 2: $X^{(N)} = \{X_1, X_2, \dots, X_N\}$ \triangleright Points maîtres choisis avec distribution uniforme
 - 3: **Pour** $i = 1 : N$
 - 4: $f^{(i)} = f(X_i)$
 - 5: **Fin Pour**
 - 6: $FF = [f^{(1)}, f^{(2)}, \dots, f^{(N)}]$
 - 7: $a_f = \text{Rippaoptim}(X^{(N)}, FF, a_0)$
 - 8: **Pour** $i = 1 : N$ \triangleright Calculer la matrice $A = (A_{ij})$
 - 9: **Pour** $j = 1 : N$
 - 10: $A_{ij} = \phi(\|X_i - X_j\|)$ $\triangleright \phi(r) = e^{-r*a_f}$
 - 11: **Fin Pour**
 - 12: **Fin Pour**
 - 13: $A.\omega = FF$ \triangleright Déterminer la matrice du poids ω en résolvant l'équation
 - 14: $f_{RBF}(X) = \sum_{i=1}^N \omega_i \phi(\|X - X_i\|)$
-

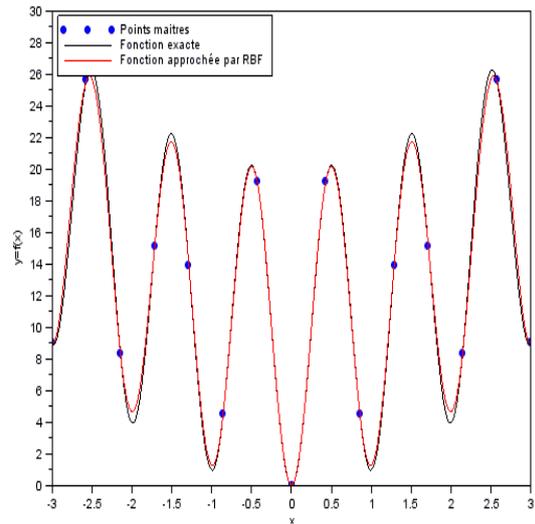
Fonction Rastrigin

$$f(x) = 10n + \sum_{i=1}^n x_i - 10(\cos(2\pi x_i))$$

***1D** $-3 \leq x_1 \leq 3$



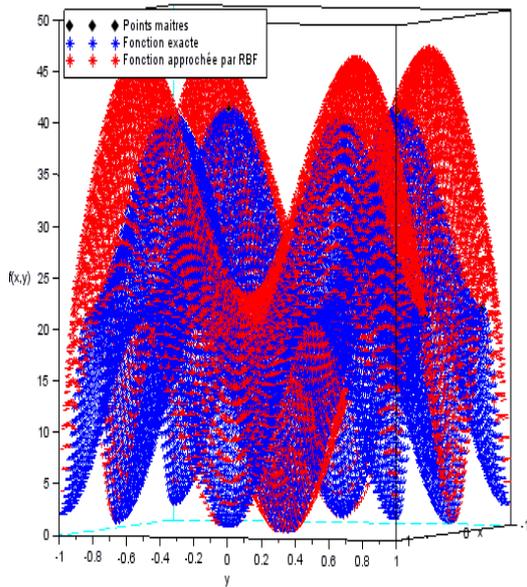
(a) N=5



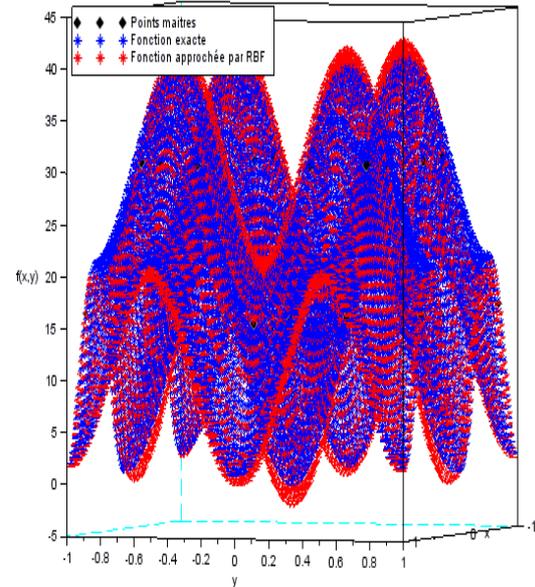
(b) N=15

FIGURE 1.20 – Approximation de la fonction Rastrigin 1D par RBF avec différents nombre total N de points maîtres (a) N=5 et (b) N=15

***2D** $-1 \leq x_1, x_2 \leq 1$



(a) N=25



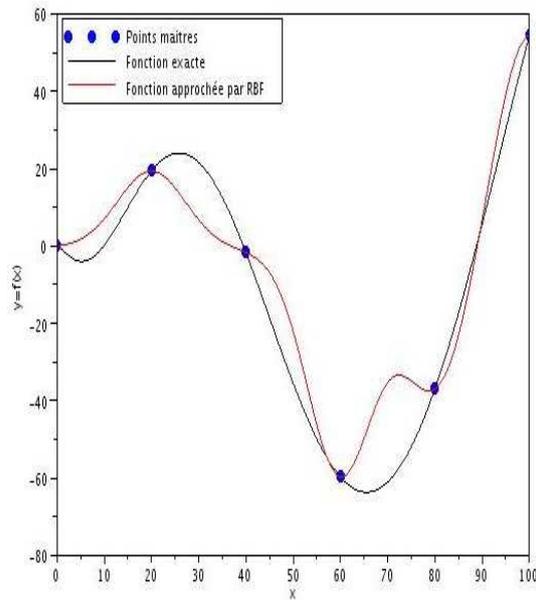
(b) N=49

FIGURE 1.21 – Approximation de la fonction Rastrigin 2D par RBF avec différents nombre total N de points maîtres (a) N=25 et (b) N=49

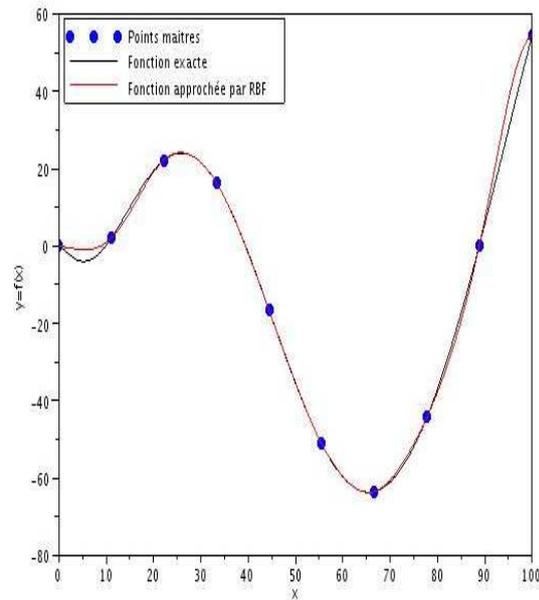
Fonction Schwefel

$$f(x) = - \sum_{i=1}^n x_i \sin(\sqrt{x_i})$$

*1D $0 \leq x_1 \leq 100$



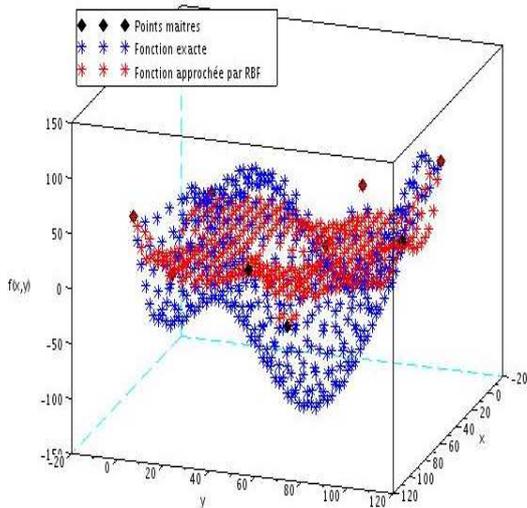
(a) $N=6$



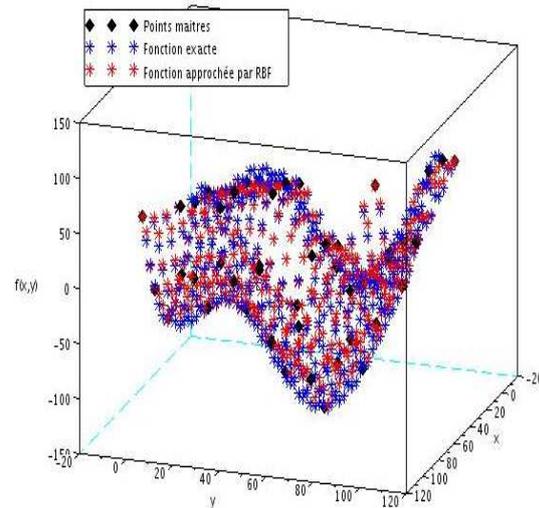
(b) $N=10$

FIGURE 1.22 – Approximation de la fonction Schwefel 1D par RBF avec différents nombre total N de points maîtres (a) $N=6$ et (b) $N=10$

*2D $0 \leq x_1, x_2 \leq 100$



(a) $N=9$



(b) $N=49$

FIGURE 1.23 – Approximation de la fonction Schwefel 2D par RBF avec différents nombre total N de points maîtres (a) $N=9$ et (b) $N=49$

Pour avoir une idée sur la précision de notre métamodèle, nous calculons la valeur exacte et celle approchée par le métamodèle en m points test (f_i et \tilde{f}_i respectivement - $i = 1, \dots, m$), et à partir de ces deux dernières valeurs, nous calculons les facteurs RMSD - Root Mean Square Deviation -, NRMSD -Normalized Root Mean Square Deviation -, MAX et NMAX définis comme suit :

$$RMSD = \sqrt{\frac{\sum_{i=1}^m (\tilde{f}_i - f_i)^2}{m}}$$

$$NRMSD = \sqrt{\frac{\sum_{i=1}^m (\tilde{f}_i - f_i)^2}{\sum_{i=1}^m (f_i)^2}}$$

$$MAX = \max_i |\tilde{f}_i - f_i|$$

$$NMAX = \frac{MAX}{\sqrt{\frac{\sum_{i=1}^m (\tilde{f}_i - \bar{f})^2}{m}}} \quad \bar{f} = \frac{\sum_{i=1}^m f_i}{m}$$

Ces facteurs nous permettent d'analyser la précision locale et globale du métamodèle, en effet, ce dernier est précis si les valeurs des facteurs NRMSD et NMAX tendent vers 0.

Fonction test	Nb Points maîtres	RMSD	NRMSD
Rastrigin 1D	5	9.6	0.64
	15	0.4	0.02
Rastrigin 2D	25	8.5	0.37
	49	1.3	0.06
Schwefel 1D	6	10.2	0.29
	10	2.0	0.05
Schwefel 2D	9	47.4	0.92
	49	9.5	0.18

Tableau 1.2 – Les valeurs RMSD et NRMSD pour l'ensemble des cas traités

Fonction test	Nb Points maîtres	MAX	NMAX
Rastrigin 1D	5	15.8	2.11
	15	1.0	0.13
Rastrigin 2D	25	21.9	2.20
	49	3.2	0.32
Schwefel 1D	6	25.8	0.80
	10	7.8	0.24
Schwefel 2D	9	125.0	2.66
	49	31.3	0.66

Tableau 1.3 – Les valeurs MAX et NMAX pour l'ensemble des cas traités

D'après les résultats obtenus et illustrés dans les Tableaux 1.2 1.3, il est clair que notre choix adopté pour une distribution uniforme des points

maîtres et en présence d'un nombre raisonnable de ces points dans la base RBF, le métamodèle RBF fournit de bonnes approximations pour la fonction à approcher (voir les colonnes NRMSD et NMAX des tableaux), par contre si le nombre de points est faible, le métamodèle reste moins bon à cause du manque de précision dans certaines zones de la fonction (on n'a pas assez de points pour les couvrir).

CONCLUSION DU CHAPITRE

Dans ce premier chapitre, nous avons présenté les outils mathématiques utilisés dans le cadre de cette thèse. Dans un premier temps, nous avons traité de l'optimisation en général, et en particulier celle multicritère. Nous avons présenté les méthodes NBI et NNCM pour la résolution des problèmes de capture de front de Pareto, et nous avons montré que ces méthodes ont deux inconvénients majeurs qui sont la génération de certaines solutions dominées et le grand nombre d'évaluations exigées par ces méthodes, et nous avons présenté des solutions pour éliminer ces deux inconvénients (pour le premier point, nous pouvons utiliser un filtre, et pour le deuxième, nous utilisons les métamodèles dans la boucle d'optimisation). Dans un deuxième temps, nous avons présenté les métamodèles en général et celui du RBF que nous avons utilisé durant cette thèse en mettant en évidence l'ensemble des techniques à régler afin d'avoir une bonne approximation à base de ce métamodèle RBF (facteur d'atténuation et génération des points maîtres).

Dans le prochain chapitre, nous allons présenter des approches algorithmiques que nous avons développées à la base des méthodes NBI, NNCM et le métamodèle RBF permettant la capture du front de Pareto. Les deux méthodes développées consistent à faire un couplage entre les méthodes NBI et NNCM avec le métamodèle RBF. Ces méthodes va nous permettre de résoudre les problèmes d'optimisation multicritère avec un temps de calcul raisonnable surtout pour des applications industrielles complexes en mécanique non linéaire.

MODÈLES MÉCANIQUES ÉTUDIÉS

2

SOMMAIRE

2.1	ÉLASTICITÉ EN GRANDE DÉFORMATION POUR LE PACKAGING (2D)	45
2.1.1	Généralités	45
2.1.2	Conception du modèle du fond de la canette 2D	47
2.1.3	Critères à optimiser : DRP vs DG	48
2.2	ÉLASTICITÉ LINÉAIRE POUR UNE PIÈCE AUTOMOBILE (3D) . . .	50
2.2.1	Généralités	50
2.2.2	Conception du modèle de traverse 3D	51
2.2.3	Critères à optimiser : VM vs FR34	52
2.3	TECHNIQUES POUR LA DÉFORMATION DES FORMES EN 2D ET 3D	54
2.3.1	Interpolation Spline cubique pour la déformation des formes 2D	54
2.3.2	Interpolation par FFD+RBF pour la déformation des formes 3D	55
	CONCLUSION	59

DANS ce chapitre, nous présentons brièvement les modèles mécaniques étudiés durant cette thèse. En effet, le but n'est pas de présenter de manière détaillée les équations utilisées (élasticité en grandes déformation, élasticité linéaire) ni le cadre mathématique et les approximations numériques (éléments finis, intégration en temps, etc). Ces modèles représentent des cas industriels proposés par notre partenaire industriel **ArcelorMittal**. Nous avons deux cas différents pour deux différents services de l'entreprise, le modèle du fond de la canette fourni par **l'équipe Cans & Components du centre Packaging d'ArcelorMittal Research SA** et qui est un modèle 2D, et le modèle 3D de la traverse fourni par **ArcelorMittal R&D Montataire Automotive Applications** et **ArcelorMittal Tubular Product Vitry**. Dans une dernière partie de ce chapitre, nous allons présenter l'ensemble des techniques utilisées pour la déformation de la géométrie des cas d'études cités (spline cubique pour la déformation 2D et FFD+RBF pour la déformation de 3D), des techniques que nous allons utiliser dans un prochain chapitre pour faire de l'optimisation de forme.

2.1 ELASTICITÉ EN GRANDE DÉFORMATION POUR LE PACKAGING (2D)

Dans le cadre de l'optimisation des boîtes boisson, l'équipe Cans & Components du centre Packaging d'ArcelorMittal Research SA étudie plusieurs propriétés de ces emballages telles que la tenue à la pression, la tenue aux variations de températures ou encore la résistance aux chocs des boîtes boisson. L'ensemble de ces critères correspondent à un cahier des charges précis qui doit être respecté. Ces études permettent de caractériser précisément les causes de détérioration des boîtes, quelles soient de source mécanique ou chimique, le but étant d'améliorer la performance des aciers pour emballages (APE). Cette partie de thèse s'est concentrée sur la tenue à la pression du fond de la boîte boisson via la recherche des nouveaux profils à l'aide de l'optimisation géométrique.

Ce chapitre est soumis à restriction de confidentialité.

2.1.1 Généralités

La boîte boisson, communément appelée canette, est un emballage métallique de masse fabriqué à base d'aluminium ou d'acier étamé. Les fonctionnalités visées de ce genre de boîte boisson fabriqué grâce au procédé d'emboutissage / repassage de la paroi sont multiples, et voici une liste non exhaustive :

- Coût compétitif et logistique performante
- Recyclabilité
- Légèreté
- Capacité de remplissage et d'impression
- Bon aspect de surface
- Capacité d'emboutissage élevée

En principe, la canette présente de nombreux avantages tant pour les consommateurs que pour les industriels. Elle est souvent légère, résistante, compacte et on peut la trouver sous plusieurs formats (Figure 2.1).



FIGURE 2.1 – La boîte boisson dans tous ses formats

Composition de la canette

La boîte boisson est composée de deux parties : le corps qui est subdivisé en trois parties (paroi épaisse, paroi fine et une zone de transition), et le fond qui est constitué de (dôme, pied de la boîte) (Figure 2.2).

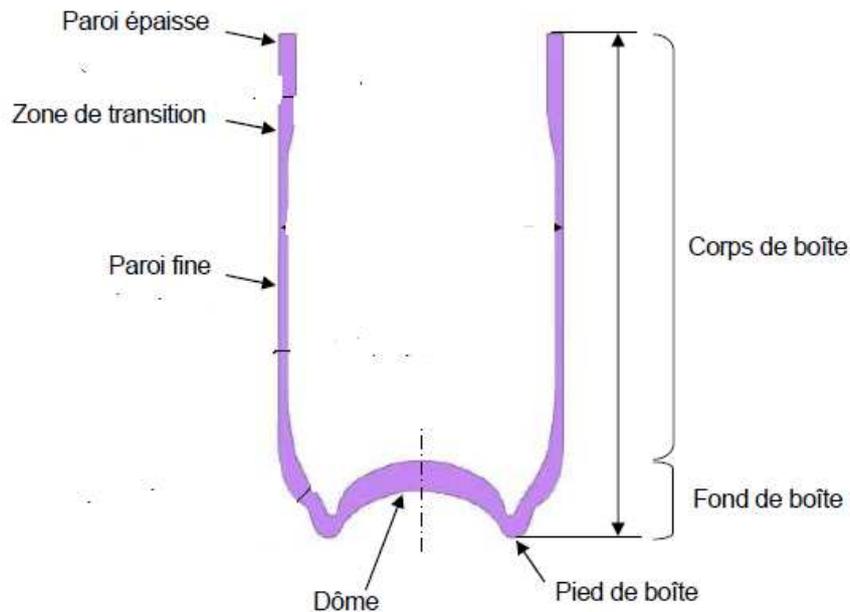


FIGURE 2.2 – Coupe d'une boîte boisson et différentes zones de ses composantes

Cahier de charge pour la conception des canettes

Le respect strict du cahier des charges s'impose lors de la fabrication d'une canette fabriquée suivant le procédé d'emboutissage/repassage de la paroi. Ce cahier de charge porte sur 3 caractéristiques de la boîte boisson :

- **La résistance axiale de la boîte vide** : Ce paramètre permet de définir la tenue de la boîte durant l'opération de sertissage. La cible visée est de 90 kg.
- **La porosité de la boîte** : Ce paramètre est important quant à l'aptitude de la boîte à résister à la corrosion créée par l'acidité des boissons carbonatées de types colas, limonades etc.
- **La tenue du fond à la pression interne** : Ce paramètre est très important. En effet, au cours de sa fabrication et de sa distribution, la pression interne d'une boîte boisson peut augmenter. De plus, son stockage à températures élevées (comme par exemple le coffre d'une voiture au soleil) peut également provoquer une augmentation de pression interne. Pour éviter tout problème, la tenue du fond à la pression interne doit être garantie.

Ce dernier critère est étudié tout au long de ce chapitre. En effet, lors de la montée en pression de la boîte boisson, le pied de la boîte se déroule petit à petit jusqu'à ce que la pression soit telle que le dôme se retourne

subitement. Afin de caractériser la tenue du fond de la canette à la pression interne, deux critères sont définis : le déroulement du pied (en anglais, dome growth [DG]) et la pression de retournement du fond (en anglais, dome reversal pressure [DRP]). Nous mettons en évidence que ces deux derniers critères sont bien concurrents, et nous réalisons une étude de front de Pareto leur correspondant.

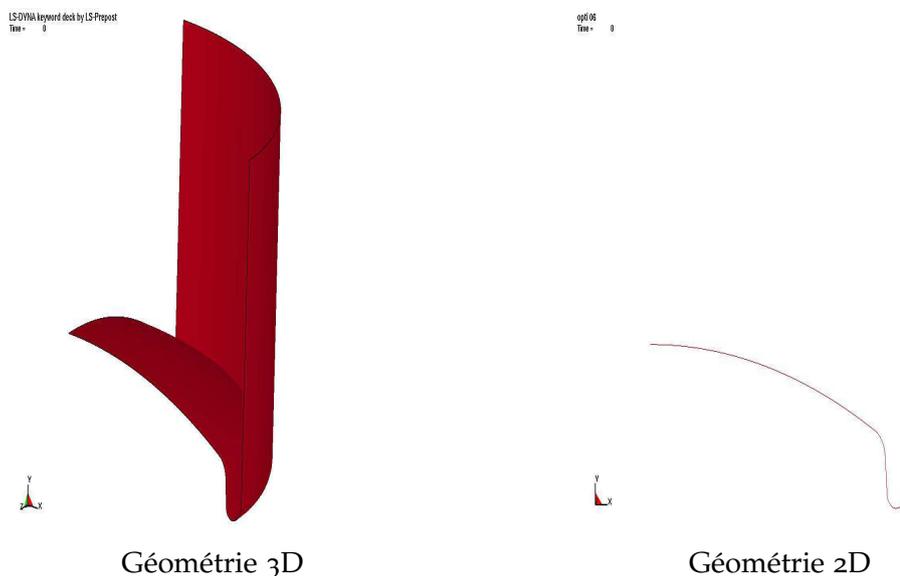
2.1.2 Conception du modèle du fond de la canette 2D

Le but principal de cette partie est de prédire de manière fiable la pression de retournement du fond d'une boîte boisson . Pour ce faire, la modélisation par éléments finis sous **LS dyna** a été choisie. Ce paragraphe est destiné à exposer les paramètres choisis pour réaliser la mise en calcul (la forme canette) ainsi que les sorties exploitées (critères à optimiser).

Modélisation

La mise en place du modèle a été faite à partir d'une seule géométrie de fond de canette mais le modèle est destiné à être fonctionnel avec toutes les géométries possibles (pour notre travail, nous allons traiter trois géométries différentes, voir chapitre suivant, Figure 3.3).

La canette étudiée est considérée axisymétrique, par conséquent, seule une demie génératrice (2D) sera suffisante pour représenter l'ensemble de la boîte. L'hypothèse de calculs en 2D ne s'éloigne pas trop de la réalité. Le modèle étudié sera uniquement un modèle du fond de la canette 2D (Figure 2.3).



Géométrie 3D

Géométrie 2D

FIGURE 2.3 – Passage de géométrie 3D au 2D pour un type de canette

Données d'entrée du modèle 2D

Les entrées du modèle peuvent être une grande source d'erreurs si elles sont mal choisies. Il est donc important de connaître précisément les paramètres des matériaux et du procédé (caractéristiques mécaniques et

caractéristiques d'acier) afin de modéliser au mieux la réalité. Pour la mise en données des caractéristiques mécaniques, le logiciel de simulation LS Dyna traite séparément la partie **élastique** et la partie **plastique** de la courbe de traction. Pour la partie élastique, les données à renseigner sont le module d'Young E et le coefficient de Poisson ν . La loi d'Hollomon (Equation 2.1) a été utilisée pour modéliser la partie plastique de la courbe de traction à partir de la limite d'élasticité (Re), de la résistance à la traction (Rm) et du coefficient d'écrouissage n . Toutes ces données sont obtenues à partir des essais de traction uniaxiale. Un dernier argument d'entrée est celui de l'épaisseur d'acier utilisé e .

La loi d'Hollomon ou loi puissance permet d'écrire la contrainte sous la forme suivante :

$$\sigma = K\epsilon^n \quad (2.1)$$

Pour notre modèle, nous avons choisi les valeurs suivantes pour nos caractéristiques :

- $\nu = 0.3$
- $E = 210$ GPa
- $e = 0.46$ mm
- $n = 0.2$
- $Re = 270$ MPa
- $Rm = 380$ MPa
- $K = e^{(Rm \cdot (n \cdot (1 - \ln(n))))}$

Toutes les modélisations ont été faites sous le logiciel de simulation LS Dyna, où nous avons choisi une méthode de résolution **implicite explicite** des systèmes d'équations. Cette méthode débute par une résolution implicite, et si la convergence des itérations d'équilibre échoue, elle passera automatiquement à une résolution explicite pour un intervalle de temps déterminé. Après cet intervalle de temps, la méthode revient au mode implicite et tente de continuer ([Ls dyna manual.2012](#)).

2.1.3 Critères à optimiser : DRP vs DG

Comme nous l'avons déjà signalé, il y a deux critères d'optimisation à prendre en compte simultanément : le déroulement du pied (DG) qui est un critère mesuré par le déplacement vertical du pied de la canette (exprimé en mm) tandis que la pression de retournement (DRP) représente la pression à laquelle le dôme s'est retourné (exprimé en bar). Ces critères peuvent être décrits comme suit :

Pression de retournement du fond

Comme nous l'avons déjà signalé, la méthode utilisée pour la résolution est bien celle implicite explicite, qui consiste à passer de la méthode implicite à celle explicite à un certain temps (**temps de switch**). A ce moment là, le dôme de la canette se retourne. Le DRP n'est que la valeur de la pression à ce temps de switch (T_{switch}).

D'autre part, pour notre modèle, la pression appliquée est une fonction linéaire par rapport au temps :

$$Pression(t) = 4 t \quad (2.2)$$

d'où :

$$DRP = Pression(Tswitch) = 4 Tswitch \quad (2.3)$$

Déroulement du pied

Ce critère n'est autre que le déplacement vertical Y du point de contact avec un support plan X_c au moment où le code de calcul passe de la méthode implicite à celle explicite ($Tswitch$). D'où la formule mathématique de DG peut être définie comme suit :

$$DG = Y(X_c)_{t=Tswitch} \quad (2.4)$$

Le but est de chercher des nouveaux profils pour le fond du canette permettant de :

- Maximiser la valeur de Pression de Retournement du Fond.
- Minimiser la valeur de Déroulement du Pied.

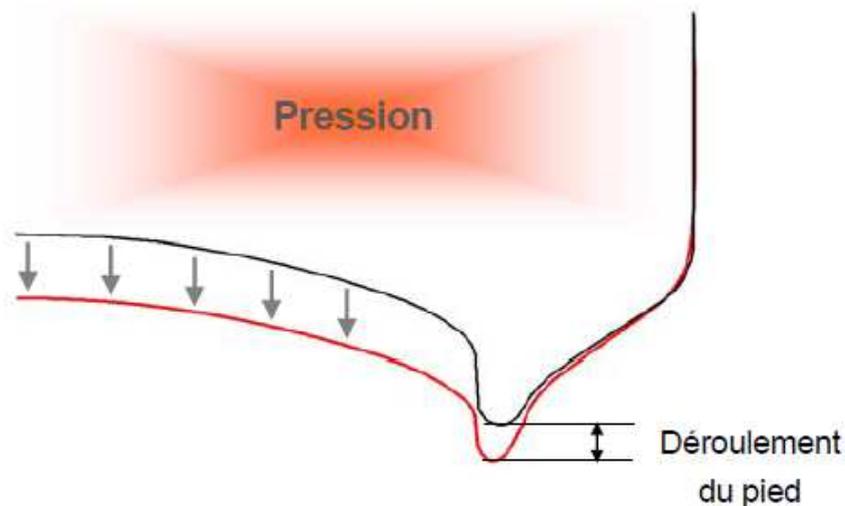


FIGURE 2.4 – Déroulement du pied de la canette causé par la pression interne

Extraction des valeurs DRP et DG via les fichiers des résultats de Ls dyna

Si nous disposons d'un profil du fond de canette 2D, nous pouvons extraire les valeurs correspondants à nos critères à optimiser DRP et DG à l'aide de LS prepost de LS dyna comme il est montré sur les figures ci

dessous :

* DRP

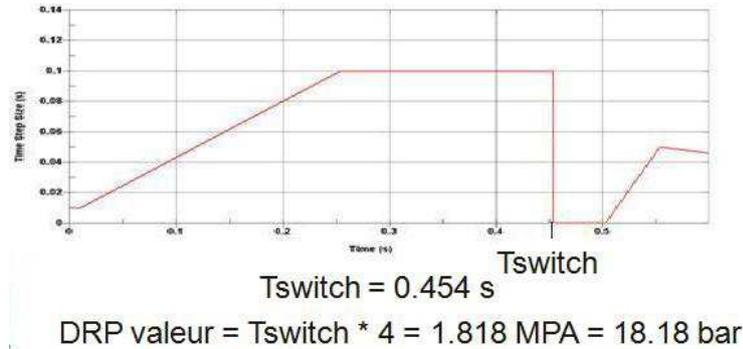


FIGURE 2.5 – Exemple d'un calcul de DRP à l'aide de Ls Dyna

* DG

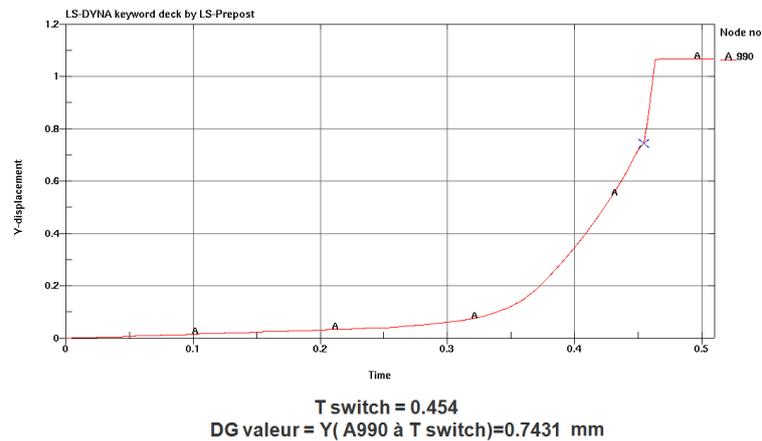


FIGURE 2.6 – Exemple d'un calcul de DG à l'aide de Ls Dyna

2.2 ELASTICITÉ LINÉAIRE POUR UNE PIÈCE AUTOMOBILE (3D)

Afin d'atteindre l'objectif de satisfaire les utilisateurs finaux du domaine de l'automobile, différentes activités R&D de modélisation ou d'expérimentation sont réalisées au centre de **techniques sur les applications automobile**, comme par exemple la mise en forme des aciers (emboutissage, profilage, hydroformage), leur assemblage (soudage par point, laser, brasage...), ou encore l'évaluation des propriétés d'usage des structures automobiles aciers (performances en crash, raideur de caisses...), ou aussi l'optimisation de forme de certaines pièces automobiles (berceau moteur, bras de suspension, traverse...). Dans cette partie de thèse, nous travaillons sur l'optimisation de la forme de la traverse.

2.2.1 Généralités

La Traverse pour véhicule automobile est destinée à être tenue par ses extrémités sur des éléments de châssis, tels que des prolonges. La

traverse est essentiellement constituée d'une structure tubulaire dont la section droite est conformée sur une partie au moins de sa longueur. Cette conformation comprend, à partir de positions choisies situées de part et d'autre de sa partie centrale, un repli sur elle-même de la structure tubulaire, lequel repli s'accroît lorsque l'on va vers les extrémités.

Les caractéristiques mécaniques importantes pour la traverse en fonction sont :

- Rigidité
- Résistance aux efforts maxi
- Résistance aux efforts exceptionnels
- Résistance à la fatigue
- Acoustique et vibratoire
- Résistance à la corrosion

2.2.2 Conception du modèle de traverse 3D

Le but principal de cette partie est la mise en place d'outils de prédiction de manière fiable du comportement mécanique de la traverse 3D. Pour ce faire, la modélisation par éléments finis sous **LS dyna** a été choisie. Ce paragraphe est destiné à exposer les paramètres choisis pour réaliser la modélisation (la traverse) ainsi que les sorties exploitées (critères à optimiser).

Modélisation

La mise en place du modèle 3D a été faite à partir d'une seule géométrie réelle de la traverse qui entre dans le cadre de l'élasticité linéaire. Pour la mise en données des caractéristiques mécaniques, nous devons renseigner les données suivantes au logiciel de simulation LS Dyna à savoir le module d'Young E et le coefficient de Poisson ν . La loi de Hooke (Equation 2.5) a été utilisée pour modéliser le comportement linéaire du matériau utilisé (acier) qui est homogène et isotrope. Un dernier argument d'entrée est celui de l'épaisseur d'acier utilisé e .

Pour la loi de Hooke, deux aspects sont importants : la linéarité et l'élasticité. Ces deux aspects ne sont pas identiques, la linéarité exprime « l'allongement est proportionnel à la force », l'élasticité exprime que cet effet est réversible et permet donc de revenir à l'état initial tel un ressort soumis à de faibles forces. L'élasticité a une limite, qui est indépendante de la notion de linéarité, Hooke n'a considéré que la phase élastique et linéaire, donc proportionnelle et réversible.

$$\sigma = 2\mu\epsilon + \lambda tr(\epsilon) \quad (2.5)$$

Avec :

$$\mu = \frac{E}{2(1 + \nu)}$$

$$\lambda = \frac{vE}{(1+v)(1-2v)}$$

Pour notre modèle, nous avons choisi les valeurs suivantes pour nos caractéristiques :

- $v = 0.3$
- $E = 210 \text{ GPa}$
- $e = 3 \text{ mm}$

2.2.3 Critères à optimiser : VM vs FR₃₄

Comme nous l'avons déjà signalé, il y a deux critères d'optimisation simultanés à prendre en compte à savoir la valeur maximale de la contrainte Von Mises sur toute la pièce (exprimée en MPA) et les forces de réaction en deux points choisis de la pièces (voir la figure 2.7) (exprimées en N).

Contrainte de Von Mises

On peut identifier sur une courbe de traction une partie élastique et une partie plastique. Considérons σ_0 comme étant la contrainte d'écoulement et $\sigma_{V\text{mises}}$ comme la contrainte de Von Mises (ou contrainte équivalente). On reste dans le domaine élastique si $\sigma_{V\text{mises}} < \sigma_0$.

La contrainte de Von Mises peut être exprimée en fonction des contraintes principales ($\sigma_1, \sigma_2, \sigma_3$) à l'aide de la formule suivante :

$$\sigma_{V\text{mises}} = \frac{1}{\sqrt{2}} [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_1 - \sigma_3)^2]^{\frac{1}{2}} \quad (2.6)$$

La contrainte de Von Mises est calculée sur tous les éléments de la pièce. Elle renseigne sur les zones en surface ou à coeur pouvant être endommagées par déformation plastique. Notre but est de minimiser la valeur maximale de cette contrainte sur toute la pièce. Autrement dit, si $P_i, i=1:N$ sont les éléments constituant la pièce de la traverse et $\sigma_{V\text{mises}}^i$ les valeurs de la contrainte de Von Mises calculées sur ces éléments respectivement, notre critère à optimiser est défini comme suit :

$$VM = \max_i \sigma_{V\text{mises}}^i \quad i = 1 : N$$

Forces de réaction

Dans notre cas d'étude, nous avons deux forces exerçant leurs actions en deux points appelés points d'applications (ou points d'impacts, point 3 et 4 sur la figure 2.7). Ces deux forces (FR_3 et FR_4) sont des forces de réactions et elles ont presque la même valeur avec deux signes différents (+ et -). Notre but étant de maximiser ces deux valeurs, pour ce faire, nous avons défini le critère suivant :

$$FR_{34} = \sqrt{FR_3^2 + FR_4^2}$$

A partir de ces données, nous faisons de l'optimisation géométrique pour chercher des nouveaux profils permettant de :

- Maximiser la valeur des forces de réaction (FR_{34}).
- Minimiser la valeur maximale de la contrainte de Von Mises (VM).

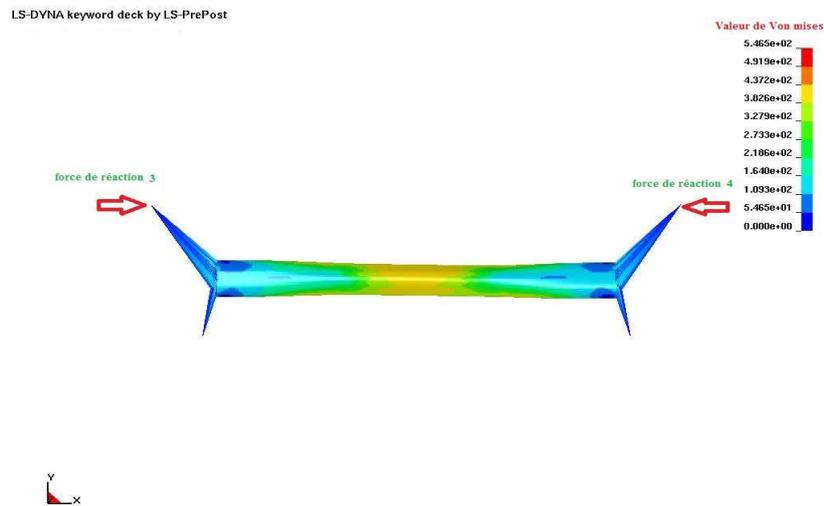


FIGURE 2.7 – Critères à optimiser pour la traverse 3D

Extraction des valeurs VM et FR_{34} via les fichiers des résultats de Ls dyna

Si nous disposons d'un profil du traverse 3D, nous pouvons extraire les valeurs correspondants à nos critères à optimiser VM et FR_{34} à l'aide de LS prepost de LS dyna comme il est montré sur les figures ci dessous :

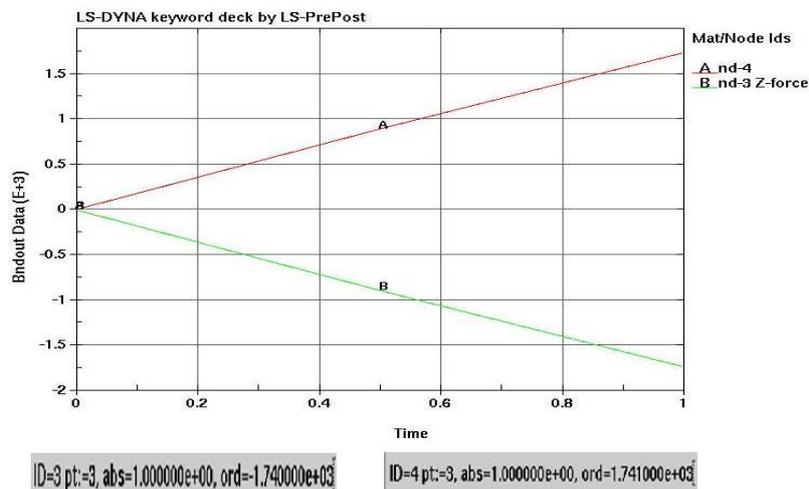


FIGURE 2.8 – Exemple d'un calcul de FR_{34} à l'aide de Ls Dyna : les deux forces FR_3 et FR_4 sont calculé à l'instant final de la simulation ($t=1$) et $FR_{34} = \sqrt{FR_3^2 + FR_4^2}$

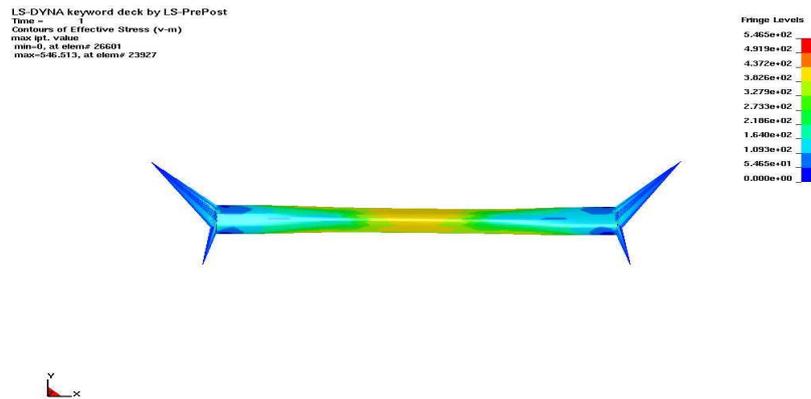


FIGURE 2.9 – Exemple d’un calcul de VM à l’aide de Ls Dyna : VM est indiqué en haut à gauche

2.3 TECHNIQUES POUR LA DÉFORMATION DES FORMES EN 2D ET 3D

Dans cette partie, nous allons présenter des techniques pour la déformation des formes en 2D et 3D dont nous aurons besoin pour la déformation des modèles présentés auparavant à savoir le fond de la canette 2D et la traverse de la voiture 3D quand nous allons faire l’optimisation de forme de ces deux derniers (chapitre 4). Dans cette section, nous allons présenter la formulation théorique des techniques utilisées, et nous allons valider les codes développés pour ces techniques sur des cas tests simples avant de les appliquer pour les cas industriels dans un prochain chapitre.

2.3.1 Interpolation Spline cubique pour la déformation des formes 2D

Généralités

Dans ce qui suit, nous allons présenter une méthode pour déformer une géométrie bidimensionnelle (2D) en utilisant une technique reposant sur les fonctions splines cubiques (Splin sur scilab). Ces fonctions mathématiques qui peuvent être utilisées pour la création d’une interpolation de classe C_0^2 (Bookstein.1989) (Choi.2000) (Balci 1.2007) (Balci 2.2007) (Shackelford.2012).

Interpolation cubique

Comme son nom l’indique, on utilise ici une équation cubique pour modéliser localement la courbe. Quatre points sont nécessaires pour évaluer la fonction qui remplace la courbe discrète. Tout dépend des conditions de continuité utilisées, la forme de la cubique peut varier et donner une interpolation différente (ex : interpolation cubique de Keys ou interpolation cubique splines). La tangente à chaque point d’indice i possède la même pente que le segment reliant les points d’indice $i-1$ et $i+1$, ce qui signifie que chaque pic de la courbe peut être dépassé par la courbe interpolée.

Fonction Splin de Scilab

Cette fonction calcule une spline cubique ou sous-spline \mathbf{s} qui interpole les points maîtres (x_i, y_i) , c'est à dire, nous avons $\mathbf{s}(x_i) = y_i$ pour tout $i = 1, \dots, M$. Le résultat spline \mathbf{s} est complètement défini par le triplet (x, y, d) où d est le vecteur avec les dérivés aux x_i : $\mathbf{s}'(x_i) = d_i$ (c'est ce qu'on appelle la forme Hermite). Plusieurs types de splines peuvent être calculées à l'aide de Scilab en sélectionnant le paramètre *spline_{type}* approprié, et parmi ces types, nous trouverons :

not a knot

C'est le cas par défaut utilisé par Scilab, la spline cubique est calculée en utilisant les conditions suivantes (en considérant les M points x_1, \dots, x_M) :

$$\begin{aligned}\mathbf{s}'''(x_2) &= \mathbf{s}'''(x_2^+) \\ \mathbf{s}'''(x_{M-1}^-) &= \mathbf{s}'''(x_{M-1}^+)\end{aligned}$$

natural

La spline cubique est calculée en utilisant les conditions suivantes :

$$\begin{aligned}\mathbf{s}''(x_1) &= 0 \\ \mathbf{s}''(x_n) &= 0\end{aligned}$$

periodic

La spline cubique périodique est calculée en utilisant les conditions suivantes :

$$\begin{aligned}y_1 &= y_n \\ \mathbf{s}'(x_1) &= \mathbf{s}'(x_n) \\ \mathbf{s}''(x_1) &= \mathbf{s}''(x_n)\end{aligned}$$

monotone

Dans ce cas, la spline est calculée en utilisant un schéma local pour d_i tel que \mathbf{s} est monotone sur chaque intervalle :

$$\begin{aligned}\text{Si } y_i \leq y_{i+1}, & \text{ alors } \mathbf{s} \text{ est croissante sur l'intervalle } [x_i, x_{i+1}] \\ \text{Si } y_i \geq y_{i+1}, & \text{ alors } \mathbf{s} \text{ est décroissante sur l'intervalle } [x_i, x_{i+1}]\end{aligned}$$

2.3.2 Interpolation par FFD+RBF pour la déformation des formes 3D

Généralités

Dans ce qui suit, nous allons présenter une méthode FFD (free form deformation) pour déformer une géométrie tridimensionnelle (3D) en utilisant une technique reposant sur les fonctions à base radiales (RBF). Ces fonctions mathématiques qui peuvent être utilisées pour la création d'une interpolation lisse entre les valeurs connues seulement à un ensemble discret de positions. Le terme radial est utilisé parce que l'argument d'entrée

des fonctions de base est généralement calculé comme la distance entre une position fixe dans l'espace 3D et une autre position à laquelle nous aimerions évaluer une certaine quantité (Lassila.2010) (Manzoni.2010) (Lassila.2011).

Interpolation par RBF

Soit $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ une fonction à valeur scalaire. Supposons que cette fonction soit connue en M points discrets distincts \mathbf{x}_i dans l'espace tridimensionnel. Soit $g(r_i) : \mathbb{R} \rightarrow \mathbb{R}$ une fonction base radiale où r_i est la distance entre le point $\mathbf{x} = (x, y, z)$ que nous cherchons à évaluer et \mathbf{x}_i un point déjà évalué (un parmi les M points).

Une fonction RBF F nous fournit un moyen pour créer une fonction d'interpolation lisse de f dans l'ensemble du domaine de \mathbb{R}^3 . Cette fonction est écrite comme une somme de M évaluations de la fonction base radiale $g(r_i)$, cette fonction RBF est donnée par la formulation suivante :

$$F(\mathbf{x}) = \sum_{i=1}^M w_i g(\|\mathbf{x} - \mathbf{x}_i\|) + c_0 + c_1 x + c_2 y + c_3 z, \quad \mathbf{x} = (x, y, z) \quad (2.7)$$

Avec w_i des coefficients scalaires et les quatre derniers termes constituent un polynôme du premier degré avec des coefficients c_0 à c_3 . Ces termes décrivent une transformation affine qui ne peut être réalisée par les fonctions radiales seules. Depuis les M valeurs connues de la fonction $f(x_i, y_i, z_i) = F_i$, nous pouvons assembler un système d'équations linéaires d'ordre $M+4$:

$$\mathbf{G}\mathbf{A} = \mathbf{F} \quad (2.8)$$

où $\mathbf{F} = (F_1, F_2, \dots, F_M, 0, 0, 0, 0)$, $\mathbf{A} = (w_1, w_2, \dots, w_M, c_0, c_1, c_2, c_3)$ et \mathbf{G} est une matrice d'ordre $(M+4) \times (M+4)$ définie comme suit :

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & \bullet & \bullet & \bullet & g_{1M} & 1 & x_1 & y_1 & z_1 \\ g_{21} & g_{22} & \bullet & \bullet & \bullet & g_{2M} & 1 & x_2 & y_2 & z_2 \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \\ g_{M1} & g_{M2} & \bullet & \bullet & \bullet & g_{MM} & 1 & x_M & y_M & z_M \\ 1 & 1 & \bullet & \bullet & \bullet & 1 & 0 & 0 & 0 & 0 \\ x_1 & x_2 & \bullet & \bullet & \bullet & x_M & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \bullet & \bullet & \bullet & y_M & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \bullet & \bullet & \bullet & z_M & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.9)$$

Avec $g_{ij} = g(\|\mathbf{x}_i - \mathbf{x}_j\|)$.

Pour le choix de la fonction à base radiale, il y en a plusieurs connues dans la littérature scientifique comme par exemple :

- $g(t) = \sqrt{\log(t^2 + k^2)}, \quad k^2 \geq 1$
- $g(t) = \exp(-(k t)^2), \quad k^2 \geq 1$
- $g(t) = \sqrt{\log(t^2 + k^2)}, \quad k^2 \geq 1$

Pour notre cas, nous utilisons la deuxième formule (type gaussienne) avec $k = 3$. Maintenant, nous pouvons facilement résoudre l'équation (2.8), afin d'obtenir les $M+4$ coefficients à utiliser dans l'expression (2.7) lors de l'interpolation entre les valeurs connues.

Déformation d'une géométrie 3D à l'aide de l'interpolation RBF

Dans cette partie, nous allons essayer de répondre à la question suivante : Comment pouvons-nous utiliser l'interpolation RBF pour effectuer une déformation pour une géométrie 3D ?

Soit FG une géométrie 3D que nous cherchons à déformer.

Dans un premier temps, nous choisissons certains points de la géométrie initiale, et nous commençons par la déformation de ces points (déformation = déplacement des points en espace 3D), alors, cette déformation sera représentée par un vecteur décrivant le déplacement 3D \mathbf{u}_i du point de la géométrie qui a été positionné à \mathbf{x}_i à la géométrie initiale (état non déformé).

Dans l'état actuel, il y a uniquement les points choisis (point maîtres) dont nous avons effectué un déplacement et leurs nouvel état est décrit par le vecteur $\mathbf{x}_i + \mathbf{u}_i$, alors à ce niveau, nous pouvons faire appel à l'interpolation RBF pour effectuer le déplacement du reste des points constituant la géométrie initiale. Soit $\mathbf{x}_i = (x_i, y_i, z_i)$ et $\mathbf{u}_i = (u_i^x, u_i^y, u_i^z)$ les points de contrôle (points maîtres) choisis et leur déplacement en espace 3D respectivement. Nous considérons maintenant les trois systèmes linéaires suivants :

$$\mathbf{G}\mathbf{A}_x = (u_1^x, u_2^x, \dots, u_M^x, 0, 0, 0, 0)^T \quad (2.10)$$

$$\mathbf{G}\mathbf{A}_y = (u_1^y, u_2^y, \dots, u_M^y, 0, 0, 0, 0)^T \quad (2.11)$$

$$\mathbf{G}\mathbf{A}_z = (u_1^z, u_2^z, \dots, u_M^z, 0, 0, 0, 0)^T \quad (2.12)$$

où \mathbf{G} est assemblé comme décrit en (2.9). Les solutions de \mathbf{A}_x , \mathbf{A}_y , et \mathbf{A}_z nous donnent les coefficients d'interpolation des déplacements dans les trois directions de l'expression (2.7).

Validation du code de FFD + RBF : Déformation par RBF

En nous basant sur les notions de la méthode présentée auparavant pour la déformation d'une géométrie 3D, nous avons écrit un code scilab

pour ce faire, et ci-dessous, nous présentons les résultats obtenus pour un cas simple à savoir une surface 3D qui est un carré.

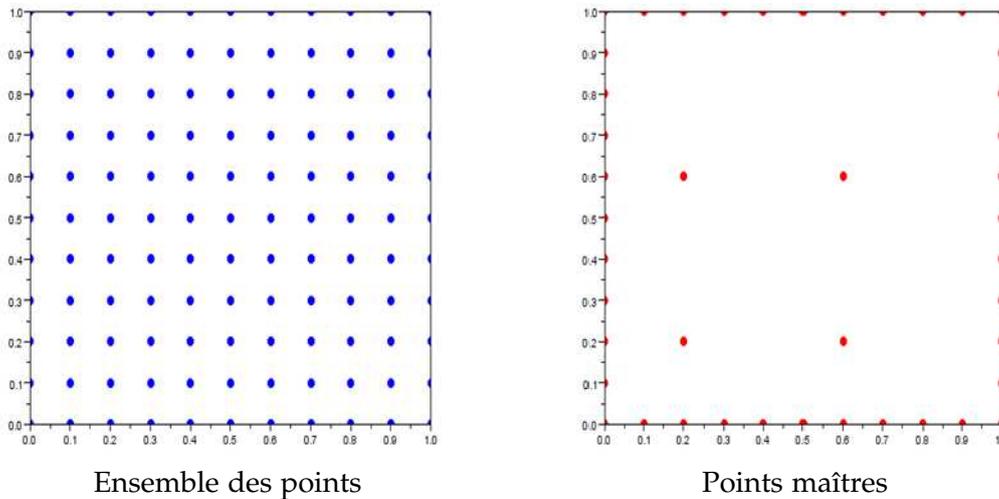


FIGURE 2.10 – Géométrie 3D à déformer (Ensemble des points en bleu et les points maîtres en rouge)

Pour appliquer notre FFD, nous avons choisi parmi l'ensemble des points constituant notre géométrie à déformer un échantillon des points maîtres définis comme suit :

- Ensemble des points de la frontière dont nous faisons un déplacement nul dans toutes les directions (points fixés).
- Ensemble des points au centre de la géométrie auquel on applique un déplacement dans toutes les directions.

Les deux figures 2.10 montrent clairement ce choix effectué, et après nous avons utilisé notre code afin de déformer l'ensemble des points et par la suite nous obtenons la déformation totale de la géométrie initiale 2.11.

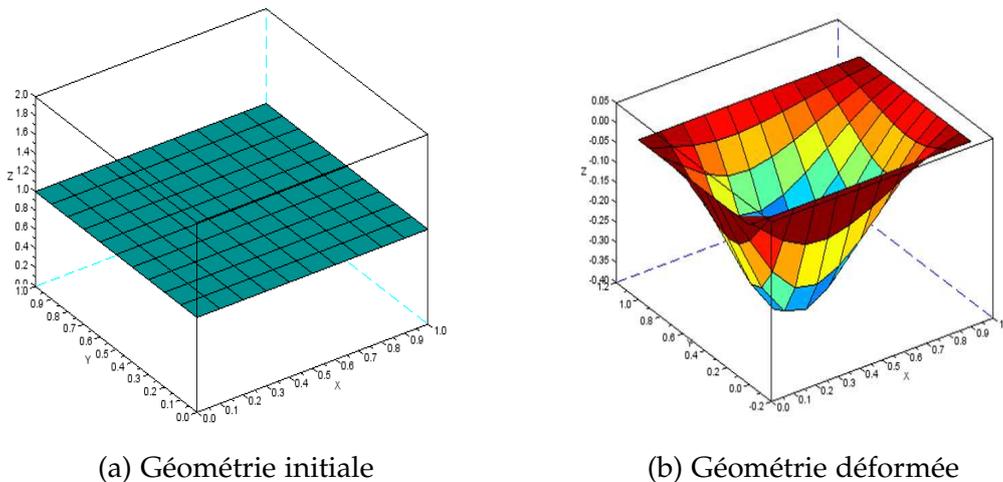


FIGURE 2.11 – Surface déformée (b) par Free Form Deformation (FFD + RBF) à partir d'une surface initiale plane (a)

CONCLUSION DU CHAPITRE

Dans ce chapitre, nous avons succinctement décrit les modèles mécaniques, avec les détails des procédures de calcul effectif de réponses elasto-plastique et élastiques, ainsi que celles de calcul de critères. Nous avons également décrit les méthodes de la variation ou la déformation de la forme en 2D (Spline cubique) et en 3D (FFD par RBF). L'ensemble de ces méthodes présentées nous sera nécessaire pour réaliser le capture du front de Pareto lors de l'optimisation concurrente de forme en plasticité (Fond de la canette 2D) et en élasticité (Traverse 3D).

ÉTUDE DE CAPTURE DE FRONT DE PARETO VIA LES MÉTAMODÈLES - APPLICATION INDUSTRIELLES

SOMMAIRE

3.1	COUPLAGE DES MÉTHODES NBI NNCM ET DU MÉTAMODÈLE RBF	63
3.2	VALIDATION DE COUPLAGE (CAS ACADÉMIQUES)	63
3.2.1	Validation de couplage NBI et RBF	63
3.2.2	Validation de couplage NNCM et RBF	67
3.3	VALIDATION DE COUPLAGE NBI RBF POUR LES CAS INDUSTRIELS	70
3.3.1	Fond de la canette 2D	70
3.3.2	Pièce automobile : Traverse 3D	82
	CONCLUSION	90

DANS ce chapitre, nous présentons un couplage effectué entre les deux méthodes NBI et NNCM avec le métamodèle RBF. Ce couplage représente un outil efficace pour la capture de front de Pareto pour des problèmes d'optimisation multicritère en faisant appel à un nombre réduit d'évaluations (souvent coûteuses en terme de temps de calcul) pour les fonctions critère constituant les problèmes d'optimisation. Cette réduction d'évaluations à travers le couplage rend cet outil de capture de front de Pareto raisonnablement coûteux en temps de calcul.

Dans un premier temps, nous validons les deux couplages **NBI RBF** et **NNCM RBF** sur des cas test académiques, et dans un deuxième temps, nous résolvons les deux problèmes d'optimisation de forme présentés auparavant, le fond de la canette 2D et la traverse 3D, à l'aide du couplage NBI RBF, et nous terminons par une discussion sur les résultats obtenus.

3.1 COUPLAGE DES MÉTHODES NBI NNCM ET DU MÉTAMODÈLE RBF

Les problèmes d'optimisation multicritère en général et, en particulier, les problèmes liés à l'optimisation de 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 oeuvre. Pour l'appliquer à des études de cas industriels, un couplage entre un métamodèle et un algorithme d'optimisation multicritère est souvent incontournable.

Ce couplage permet de bénéficier des avantages des méthodes de capture de front de Pareto (NBI et NNCM), tels que l'identification de l'ensemble de solutions optimales avec une bonne dispersion des solutions sur le front de Pareto même pour des problèmes particulièrement difficiles, tout 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 RBF avec les deux méthodes étudiées ici, NBI et NNCM.

L'idée principale de notre couplage est de résoudre le problème d'optimisation initial en remplaçant les fonctions objectif exactes dont l'évaluation est souvent coûteuse par des fonctions approchées par le métamodèle RBF construit à la base d'un échantillon des valeurs exactes de la fonction. Nous rappelons que les points maîtres constituant l'échantillon sont choisis à partir d'une distribution uniforme et que le facteur d'atténuation du métamodèle est optimisé à l'aide de la technique de Rippa.

Dans un premier temps, nous allons valider les deux approches pour des cas test académiques, en faisant une comparaison entre les résultats obtenus par les deux approches développées avec ceux obtenus par la résolution des problèmes uniquement par les méthodes NBI et NNCM sans couplage. Dans un deuxième temps, nous allons appliquer nos approches pour les cas industriels présentés en chapitre 3 à savoir l'optimisation concurrente de forme du fond de la canette 2D et celle de la traverse 3D.

3.2 VALIDATION DE COUPLAGE (CAS ACADÉMIQUES)

3.2.1 Validation de couplage NBI et RBF

Algorithme NBI RBF

L'algorithme (8) présente le couplage que nous avons effectué pour la méthode NBI et le métamodèle RBF. En nous basant sur ce dernier, nous avons programmé l'approche NBI RBF à l'aide du logiciel *Scilab*.

Algorithm 8 Algorithme de l'approche NBI RBF pour le cas bicritère

1: **Données** : (P) le problème à résoudre, N le nombre des points maîtres.

$$(P)$$

$$\min_x F(x) = (f_1(x), f_2(x))$$

$$s/c \begin{cases} g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases}$$

2: $x^{(N)} = \{x_1, x_2, \dots, x_N\}$ \triangleright Points maîtres choisis d'une manière uniforme

3: $(\tilde{f}_1, \tilde{f}_2) = \text{RBF}(f_1, f_2, x^{(N)})$ \triangleright Construction des métamodèles pour chaque fonction critère à l'aide de l'algorithme 7

4: $x_i^* = \min_x \tilde{f}_i(x)$ $i = 1 : 2$ \triangleright Minimisation de chaque fonction critère

5: $\tilde{F}_i^* = [\tilde{f}_1(x_i^*), \tilde{f}_2(x_i^*)]^T$ $i = 1 : 2$ \triangleright Individual minima

6: $\tilde{F}^* = [\tilde{f}_1(x_1^*), \tilde{f}_2(x_2^*)]^T$ \triangleright Shadow Minima

7: ϕ \triangleright La matrice Pay-off d'ordre (2×2)

8: $\phi_i = \tilde{F}(x_i^*) - \tilde{F}^*$ \triangleright Le i^{eme} colonne de ϕ

9: $\mathbf{n} = -\phi.e, e = (1, 1)^T$ \triangleright vecteur normal

10: $\beta = (\beta_1, \beta_2)^T, \beta_i \geq 0, \sum_{i=1}^2 \beta_i = 1$ \triangleright Discrétisation CHIM

11: Pour chaque $\beta = (\beta_1, \beta_2)^T$ \triangleright Généré par l'algorithme 4

12: On résout le problème :

$$\max_{x,t} t$$

$$s/c \begin{cases} \phi.\beta + t.\mathbf{n} = \tilde{F}(x) - \tilde{F}^* \\ g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases}$$

Cas test

Cette validation s'applique à un ensemble de tests académiques classiques comme SCH1, HannaNC, Fon et P4, les quatre problèmes sont des problèmes à une, deux, trois et quatre variables d'optimisation respectivement. Les deux derniers problèmes ont été présentés auparavant, par contre les deux premiers sont définis dans ce qui suit.

Schaffer 1 -SCH1-

Le problème SCH1 (Deb.2001) est proposé par Schaffer dans lequel il minimise deux fonctions coût avec une seule variable, et a comme solution un front de Pareto convexe. Il est donné par :

$$\begin{array}{c}
 (SCH1) \\
 \min_x F(x) = \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x - 2)^2 \end{cases} \\
 s/c \left\{ \begin{array}{l} -5 \leq x \leq 5 \end{array} \right.
 \end{array} \quad (3.1)$$

Hanna Non convexe -HannaNC-

Le problème HannaNC (Deb.2001) porte sur deux fonctions coût et deux variables. Il est proposé par Deb, et a comme solution un front de Pareto concave. Il est défini comme suit :

$$\begin{array}{c}
 (HannaNC) \\
 \min_{x=(x_1, x_2)} F(x) = \begin{cases} f_1(x) = \sqrt{x_1} \\ f_2(x) = \sqrt{x_2} \end{cases} \\
 s/c \left\{ \begin{array}{l} x_1 + x_2 \leq 5 \\ 0 \leq x_1 \\ 0 \leq x_2 \end{array} \right.
 \end{array} \quad (3.2)$$

Résultats

D'après les graphiques de la figure (3.1), il est clair que notre approche du couplage NBI RBF a donné de bons résultats. En effet, nous avons pu converger vers les solutions exactes (celles fournies par la méthode NBI) en utilisant le métamodèle RBF dans la boucle d'optimisation. Cette convergence était obtenue uniquement en faisant appel à un nombre minimal d'évaluations exactes. Le tableau 3.1 montre que pour l'ensemble des problèmes traités, le nombre d'appels de fonctions exactes lors de la résolution à l'aide de l'approche NBI RBF est largement inférieur au nombre d'évaluations exigées par la méthode NBI. Par exemple, pour le problème HannHc, l'approche NBI RBF a effectué 50 évaluations exactes contre 7997 évaluations effectuées par la méthode NBI, autrement dit, le nombre d'évaluations NBI RBF présente 0.62% d'évaluations NBI.

Pour les problèmes SCH₁, FON et P₄, le nombre d'évaluations NBI RBF présente 12.75%, 41.96% et 36.16% d'évaluations NBI respectivement. D'où une réduction remarquable pour le nombre d'évaluations et par la suite un gain considérable en terme de temps de calcul.

Problème	Méthode	nb.points maîtres	nb.évaluations
SCH ₁	NBI	-	314
	NBI RBF	20	40
HannNC	NBI	-	7997
	NBI RBF	25	50
Fon	NBI	-	305
	NBI RBF	64	128
P ₄	NBI	-	448
	NBI RBF	81	162

Tableau 3.1 – Nombre d'évaluations exactes effectuées par les méthodes NBI et NBI RBF pour une même discrétisation du CHIM

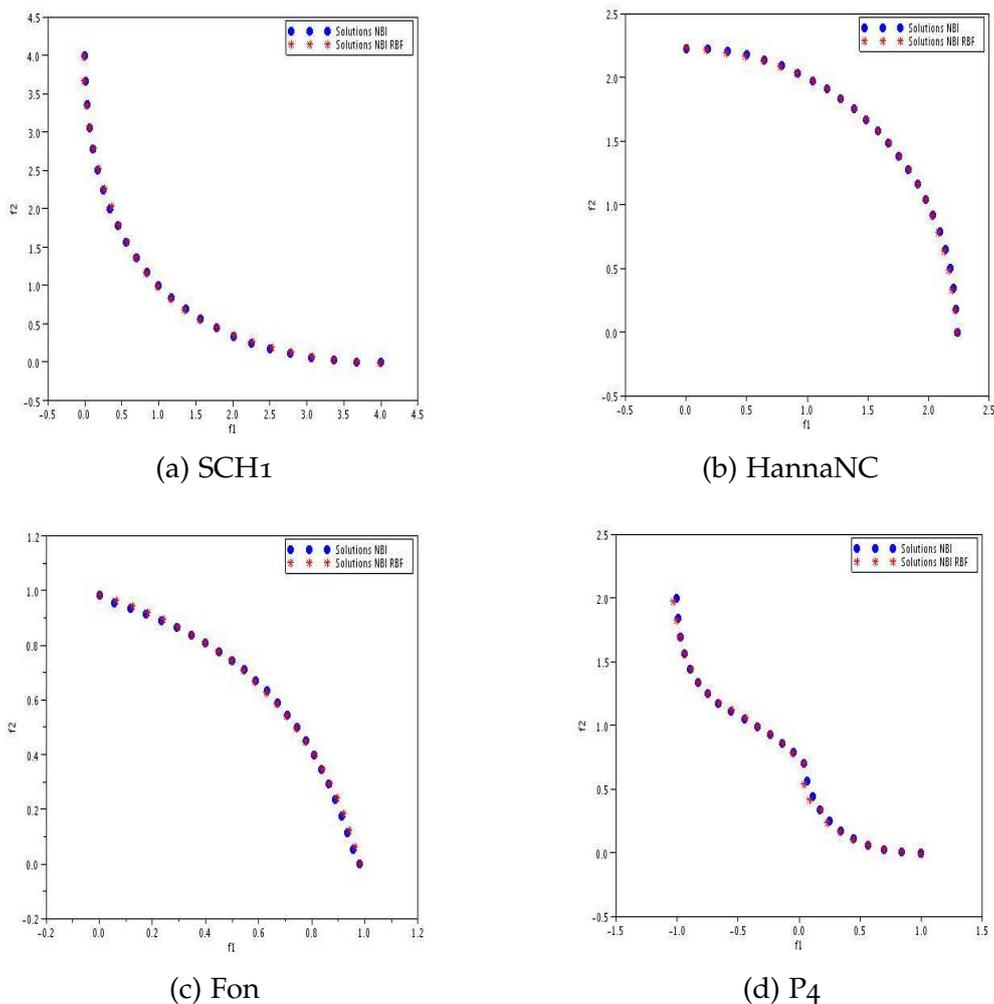


FIGURE 3.1 – Comparaison des résultats obtenus par la méthode NBI (bleu) avec ceux obtenus par l'approche NBI RBF (rouge) pour les problèmes test de (a) SCH₁, (b) HannaNC, (c) Fon et (d) P₄

3.2.2 Validation de couplage NNCM et RBF

Algorithme NNCM RBF

L'algorithme (9) présente le couplage que nous avons effectué pour la méthode NNCM et le métamodèle RBF. En se basant sur ce dernier, nous avons programmé l'approche NNCM RBF à l'aide du logiciel *Scilab*.

Cas test

Cette validation s'applique à un ensemble de tests académiques classiques comme SCH2, TNK, P3 et P4bis, les quatre problèmes sont des problèmes à une, deux, trois et quatre variables d'optimisation respectivement. Les deux premiers problèmes ont été présentés auparavant, par contre les deux derniers sont définis dans ce qui suit.

Problème à trois variables -P3-

Le problème P3 (Deb.2001) est un problème ayant trois variables et deux fonctions à minimiser. Ce problème a un ensemble optimal de Pareto convexe qui est difficile à trouver. Il est donné par :

$$\begin{array}{c}
 (P3) \\
 \min_{x=(x_1, x_2, x_3)} F(x) = \begin{cases} f_1(x) = x_1^2 + (x_3 - x_2)^2 \\ f_2(x) = x_3^2 + (x_1 - x_2 - 1)^2 \end{cases} \\
 s/c \begin{cases} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \\ 0 \leq x_3 \leq 1 \end{cases}
 \end{array} \quad (3.3)$$

Problème à quatre variables bis -P4bis-

le problème P4bis (Deb.2001) est un problème ayant quatre variables et deux fonctions à minimiser, et a comme solution un front optimal de Pareto convexe, il est défini comme suit :

$$\begin{array}{c}
 (P4bis) \\
 \min_{x=(x_1, x_2, x_3, x_4)} F(x) = \begin{cases} f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 \\ f_2(x) = 3x_1 + 2x_2 + \frac{x_3}{3} + 10(x_4)^3 \end{cases} \\
 s/c \begin{cases} 4x_1 - 2 * x_2 - 0.8x_3 + 0.5x_1^2 = 0 \\ x_1 + 2x_2 - x_3 + 0.5x_4 = 2 \\ -1 \leq x_1 \leq 1 \\ -1 \leq x_2 \leq 1 \\ -1 \leq x_3 \leq 1 \\ -1 \leq x_4 \leq 1 \end{cases}
 \end{array} \quad (3.4)$$

Algorithm 9 Algorithme de l'approche NNCM RBF pour le cas bicritère

1: **Données** : (P) le problème à résoudre, N le nombre des points maîtres.

(P)

$$\min_x F(x) = (f_1(x), f_2(x))$$

$$s/c \begin{cases} g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases}$$

2: $x^{(N)} = \{x_1, x_2, \dots, x_N\}$ \triangleright Points maîtres choisis d'une manière uniforme

3: $(\tilde{f}_1, \tilde{f}_2) = \text{RBF}(f_1, f_2, x^{(N)})$ \triangleright Construction des métamodèles pour chaque fonction objectif à l'aide de l'algorithme 7

4: $x_i^* = \min_x \tilde{f}_i(x)$ $i = 1 : 2$ \triangleright Minimisation de chaque fonction objectif

5: $\tilde{F}_i^* = [\tilde{f}_1(x_i^*), \tilde{f}_2(x_i^*)]^T$ $i = 1 : 2$ \triangleright Individual minima

6: $\tilde{F}^u = [\tilde{f}_1(x_1^*), \tilde{f}_2(x_2^*)]^T$ \triangleright Point Utopia

7: $\tilde{F}^n = [\tilde{f}_1(x_2^*), \tilde{f}_2(x_1^*)]^T$ \triangleright Point Nadir

8: $L = \tilde{F}^u - \tilde{F}^n$ \triangleright Matrice des longueurs

9: $\bar{f}_i(x) = \frac{\tilde{f}_i(x) - \tilde{f}_i(x_i^*)}{L_i}$ $i = 1 : 2$ \triangleright Fonctions objectif normalisées

10: $\bar{F}_i^* = [\bar{f}_1(x_i^*), \bar{f}_2(x_i^*)]^T$ $i = 1 : 2$

11: $\bar{N}_1 = \bar{F}_2^* - \bar{F}_1^*$ $i = 1 : 2$

12: $\bar{X}_{kj} = \sum_{k=1}^2 \beta_k \bar{F}_k^*$, $\beta_i \geq 0$, $\sum_{i=1}^2 \beta_i = 1$ \triangleright Discrétisation Utopia

13: Pour chaque $\beta = (\beta_1, \beta_2)^T$ \triangleright Généré par l'algorithme 4

14: On résout le problème :

$$\min_x \bar{f}_2(x)$$

$$s/c \begin{cases} g_j(X) \geq 0 & j = 1; \dots; J \\ h_k(X) = 0 & k = 1; \dots; K \\ a \leq X_i \leq b & i = 1; \dots; m \\ \bar{N}_1(\bar{f} - \bar{X}_{kj})^T \leq 0 \\ \text{avec } \bar{f} = [\bar{f}_1(x), \bar{f}_2(x)] \end{cases}$$

Résultats

Les graphiques de la Figure (3.2) montrent clairement les bons résultats obtenus avec notre approche du couplage NNCM RBF. En effet, nous avons pu converger vers les solutions exactes pour l'ensemble des cas traités (même pour le problème SCH2 caractérisé par son front discontinu, nous avons pu capturer un grand morceau de son front) en utilisant le métamodèle RBF dans la boucle d'optimisation. Cette convergence était obtenue uniquement en faisant appel à un nombre minimal d'évaluations exactes.

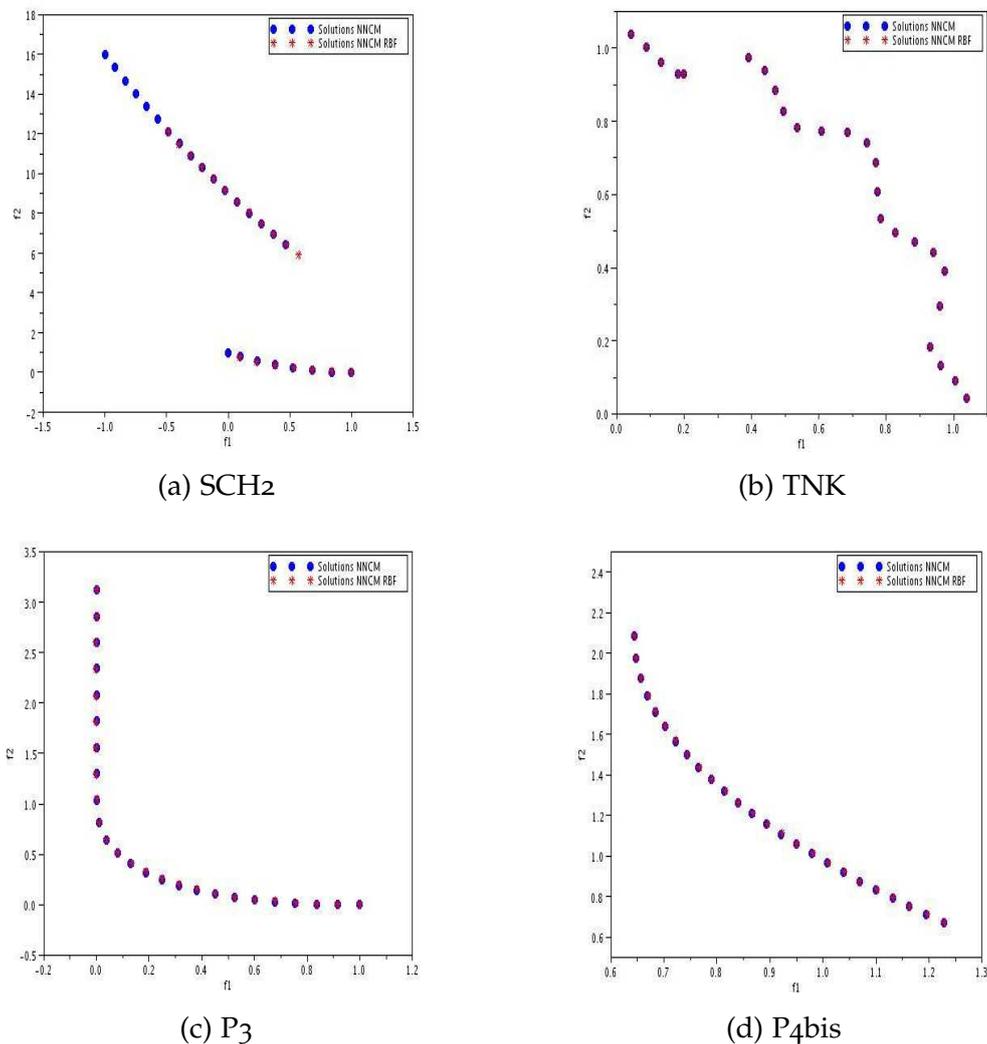


FIGURE 3.2 – Comparaison des résultats obtenus par la méthode NNCM (bleu) avec ceux obtenus par l'approche NNCM RBF (rouge) pour les problèmes test de (a) SCH2, (b) TNK, (c) P3 et (d) P4bis

Le tableau (3.2) montre que pour l'ensemble des problèmes traités, le nombre d'appels de fonctions exactes lors de la résolution à l'aide de l'approche NBI RBF est largement inférieur au nombre d'évaluations exigées

par la méthode NNCM. Par exemple, pour le problème SCH₂, l'approche NNCM RBF a effectué uniquement 50 évaluations exactes contre 2670 évaluations effectuées par la méthode NNCM, autrement dit, le nombre d'évaluations NNCM RBF présente 1.87% d'évaluations NNCM. Pour les problèmes TNK, P₃ et P_{4bis}, le nombre d'évaluations NNCM RBF présente 5.16% 6.42% et 17.60% d'évaluations NNCM respectivement. D'où, une réduction remarquable pour le nombre d'évaluations et par la suite un gain considérable en terme de temps de calcul.

Problème	Méthode	nb.points maîtres	nb.évaluations
SCH ₂	NNCM	-	2670
	NNCM RBF	25	50
TNK	NNCM	-	620
	NNCM RBF	16	32
P ₃	NNCM	-	840
	NNCM RBF	27	54
P _{4bis}	NNCM	-	920
	NNCM RBF	81	162

Tableau 3.2 – Nombre d'évaluations exactes effectuées par les méthodes NNCM et NNCM RBF pour une même discrétisation de la ligne Utopia

Conclusion

Dans cette partie, nous avons validé les deux couplages pour un ensemble de cas tests académiques (voir autres cas traités en Annexe), et nous avons prouvé qu'à l'aide de notre couplage, il y a possibilité de capturer le front exact en diminuant le nombre d' évaluations exactes des fonctions si nous utilisons uniquement les méthodes classiques, et si nous parlons de la diminution des fonctions exactes, nous parlons de la réduction du temps de calcul exigé pour la résolution de ce genre de problème d'optimisation. Maintenant, nous passons au traitement du cas industriel, mais dans cette partie, nous avons utilisé uniquement un seul couplage qui est le couplage NBI RBF.

3.3 VALIDATION DE COUPLAGE NBI RBF POUR LES CAS INDUSTRIELS

Des restrictions de confidentialité industrielle s'appliquent à un certain nombre de résultats de calcul, transmis à notre partenaire industriel, mais non présentés ici.

3.3.1 Fond de la canette 2D

Dans le cadre de la validation de notre couplage NBI RBF, nous avons étudié un cas industriel fourni par notre partenaire industriel par **Arcelor-Mittal** qui est le fond de la canette 2D. Le but est de faire l'optimisation géométrique afin de chercher des nouveaux profils qui vont satisfaire les besoins industriels demandés par les clients d'ArcelorMittal.

Pour ce cas d'études, nous avons trois types différents de profils. Pour chaque profil, nous avons mené une étude de capture de front de Pareto par l'approche NBI RBF.

Profils canette 2D

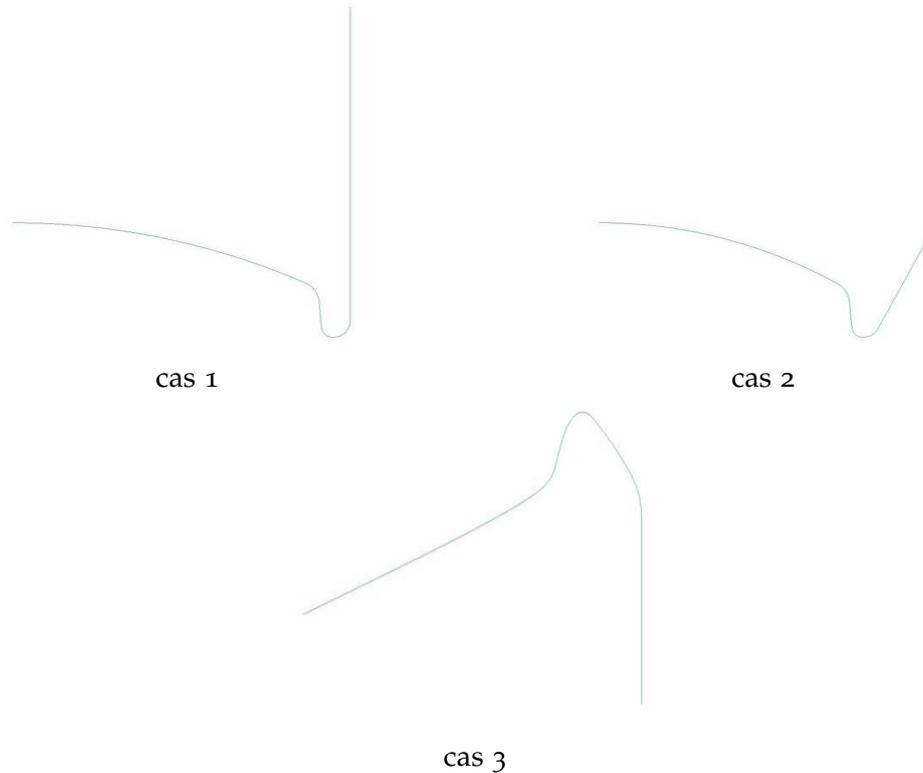


FIGURE 3.3 – Les trois types de profils de canettes étudiés

Formulation du problème d'optimisation

Variable d'optimisation

La première étape de l'optimisation est bien le choix de nos variables d'optimisation pour les trois types de formes. En effet, pour chaque forme, nous avons divisé le profil en deux parties :

- Partie fixe : cette partie de profil va être fixée durant tout le processus d'optimisation.
- Partie variable : cette partie de profil sera notre variable d'optimisation qui sera caractérisée par un certain nombre de points dits points de contrôle (figure 3.4). Cet ensemble des points de contrôle constitue notre variable d'optimisation (formule 3.5).

Soit φ un profil de canette, alors nous pouvons écrire ce dernier comme suit :

$$\varphi = (\varphi_1, \varphi_2, \varphi_3, \varphi_4) \quad (3.5)$$

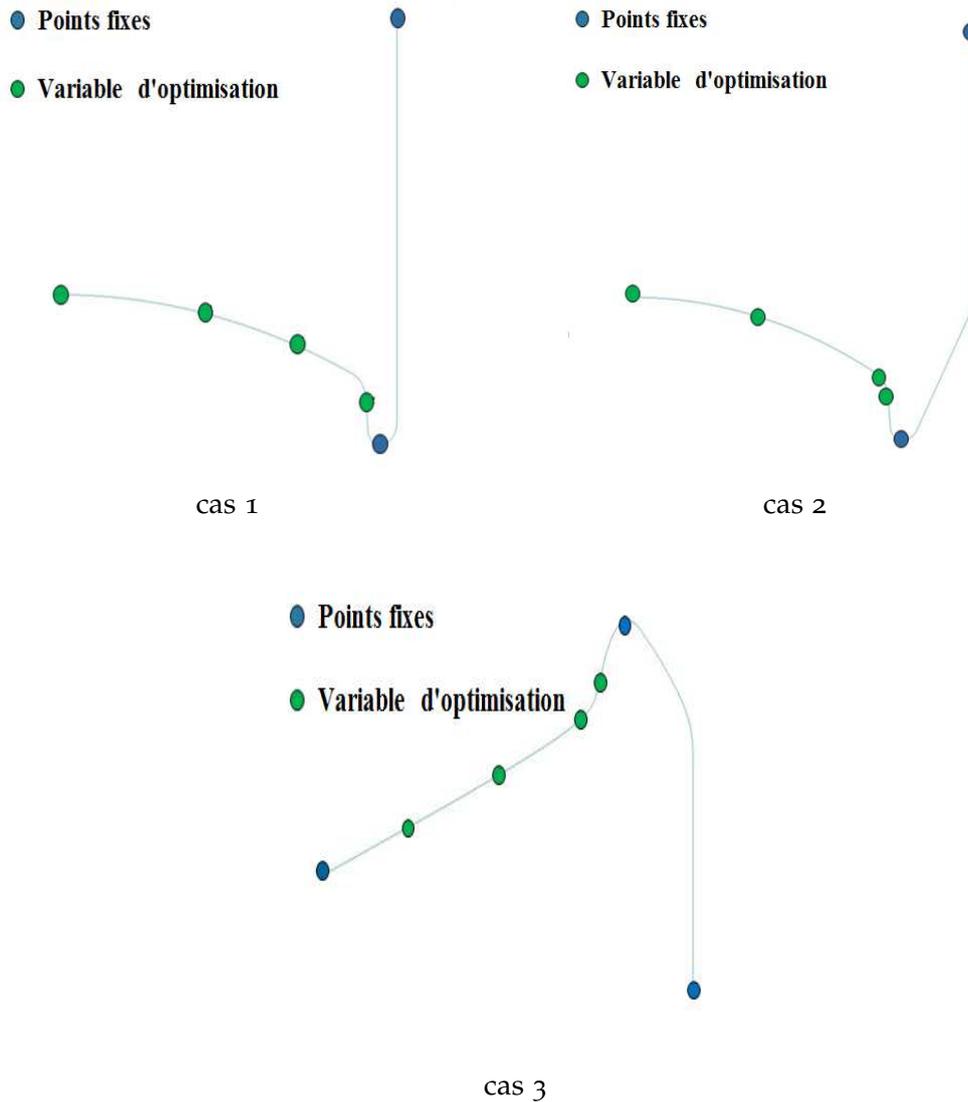


FIGURE 3.4 – Illustration de la stratégie pour le choix de la variable d'optimisation pour chaque profil de la canette 2D (cas 1, cas 2 et cas 3) : Division de la forme en partie fixe (Zone entre les points bleus) et en partie variable caractérisée par quatre points de contrôle (Points verts) qui constituent la variable de l'optimisation

Points Maîtres RBF / Contraintes NBI

Pour les points choisis comme variable d'optimisation, nous allons autoriser uniquement un déplacement vertical durant l'optimisation, par contre, ils seront fixés selon l'axe horizontal. Le déplacement vertical sera dans un intervalle bien précis de telle façon que le point initial sera le milieu de cet intervalle (contraintes NBI). De même, il faut signaler que le point initial et les points extrêmes de cet intervalle vont constituer les points maîtres pour le métamodèle RBF.

L'ensemble des points de contrôle choisis comme variable d'optimisation seront fixés selon l'axe horizontal, uniquement le déplacement vertical est autorisé durant le processus d'optimisation pour ces points. Ce déplacement vertical est autorisé dans un intervalle bien précis de telle

façon que le point de contrôle initial sera au milieu de l'intervalle. Il faut aussi signaler que le point milieu et les points extrêmes de l'intervalle vont nous servir pour le choix des points maîtres pour le métamodèle RBF. En effet, nous avons une variable d'optimisation avec quatre points de contrôle. Pour chaque point, nous choisissons trois positions possibles (le milieu et les extrêmes de l'intervalle), d'où, nous pouvons générer $3^4 = 81$ profils pour chaque forme parmi les trois types de canettes (chaque profil représente un point maître) qui vont nous servir pour la construction de notre métamodèle RBF afin d'approcher les critères à optimiser : la pression de retournement du fond **DRP** et le déroulement du pied **DG**.

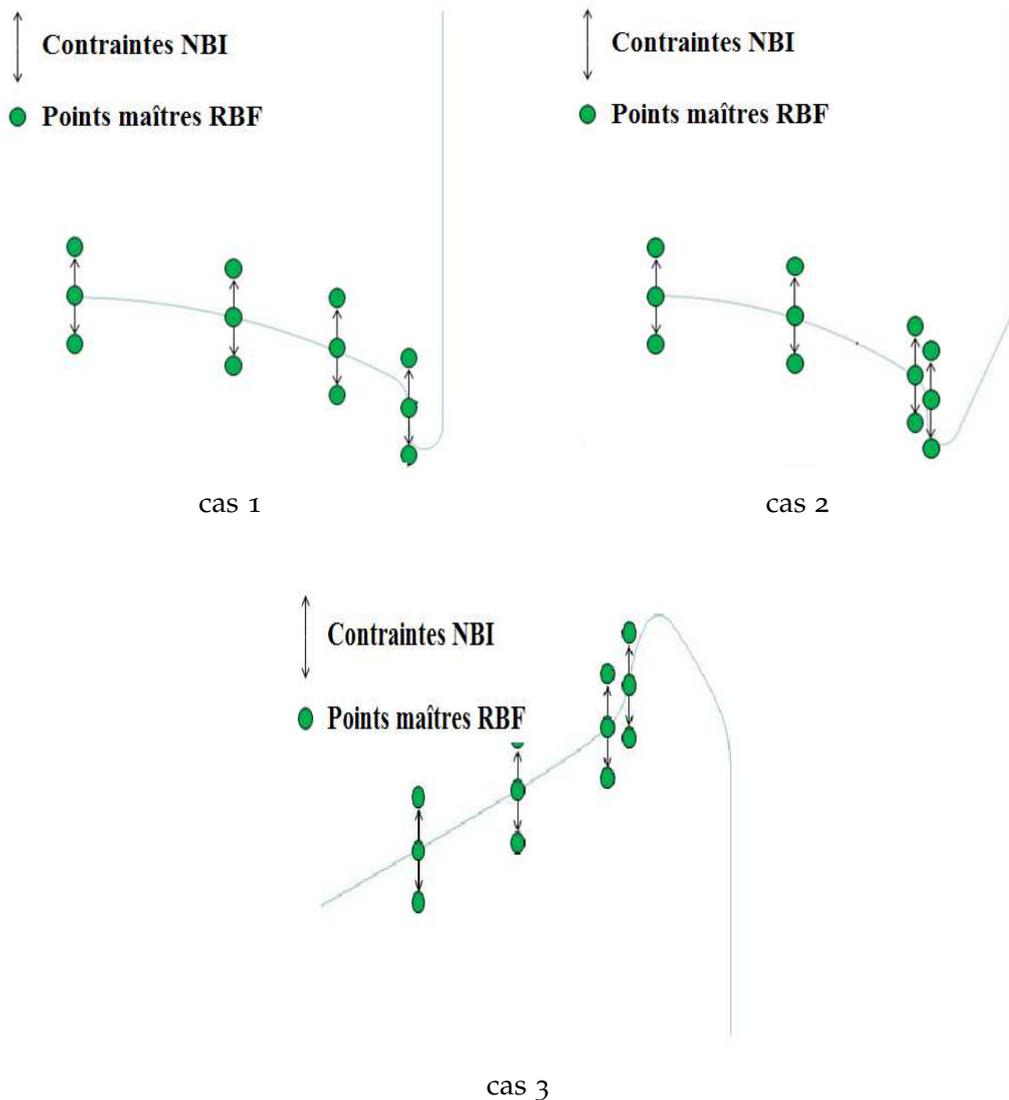


FIGURE 3.5 – Illustration de la stratégie suivie pour le choix des points maîtres pour la construction du métamodèle RBF et déterminer les contraintes de notre problèmes d'optimisation (Contraintes de bornes) pour chaque profil de la canette 2D (cas 1, cas 2 et cas 3) : les points de contrôle initiaux avec ces trois situations possibles (points verts) permettent de générer $3^4 = 81$ profils représentant les points maîtres pour le RBF

Soit $\varphi_0 = (\varphi_{01}, \varphi_{02}, \varphi_{03}, \varphi_{04})$ la forme initiale de la canette, et α un nombre réel positif (dans notre cas d'étude $\alpha = 0.5$). Les contraintes NBI qui sont uniquement des contraintes de bornes pour les cas traités peuvent s'écrire à l'aide des bornes max et min de φ_0 définis comme suit :

- $\varphi^{lower} = (\varphi_{01} - \alpha, \varphi_{02} - \alpha, \varphi_{03} - \alpha, \varphi_{04} - \alpha)$
- $\varphi^{upper} = (\varphi_{01} + \alpha, \varphi_{02} + \alpha, \varphi_{03} + \alpha, \varphi_{04} + \alpha)$

Formulation mathématique

Après avoir déterminé les critères à optimiser (chapitre 3), la maximisation **DRP** et la minimisation du **DG**, et les contraintes à vérifier (contraintes de bornes), nous pouvons écrire notre problème d'optimisation bicritère sous la forme mathématique suivante :

$$\begin{array}{c}
 (Pcan) \\
 \max_{\varphi=(\varphi_1, \varphi_2, \varphi_3, \varphi_4)} \text{DRP}(\varphi) / \min_{\varphi=(\varphi_1, \varphi_2, \varphi_3, \varphi_4)} \text{DG}(\varphi) \\
 s/c (D can) \{ \varphi^{lower} \leq \varphi \leq \varphi^{upper}
 \end{array} \quad (3.6)$$

Résolution du problème d'optimisation

Pour la résolution de ce problème d'optimisation bicritère de la canette (problème 3.6), nous avons choisi le couplage NBI RBF. Les étapes suivies par l'algorithme NBI RBF utilisé dans ce cas industriel sont décrites dans l'organigramme de la figure (3.6). En effet, les étapes principales sont les suivantes :

* Construction d'une fonction approchée par le métamodèle RBF pour chaque critère à optimiser. Le nouveau problème d'optimisation étant sous la forme :

$$\begin{array}{c}
 (PcanRBF) \\
 \max_{\varphi=(\varphi_1, \varphi_2, \varphi_3, \varphi_4)} \widetilde{\text{DRP}}(\varphi) / \min_{\varphi=(\varphi_1, \varphi_2, \varphi_3, \varphi_4)} \widetilde{\text{DG}}(\varphi) \\
 s/c (D can) \{ \varphi^{lower} \leq \varphi \leq \varphi^{upper}
 \end{array} \quad (3.7)$$

* Le problème (3.7) sera résolu à l'aide de la méthode NBI.

* La dernière étape sera de faire les évaluations exactes des solutions trouvées par notre approche, et faire une comparaison entre les deux afin de tester, au moins partiellement, l'efficacité de notre approche (en effet,

nous ne disposons pas de front de Pareto pour les cas industriels).

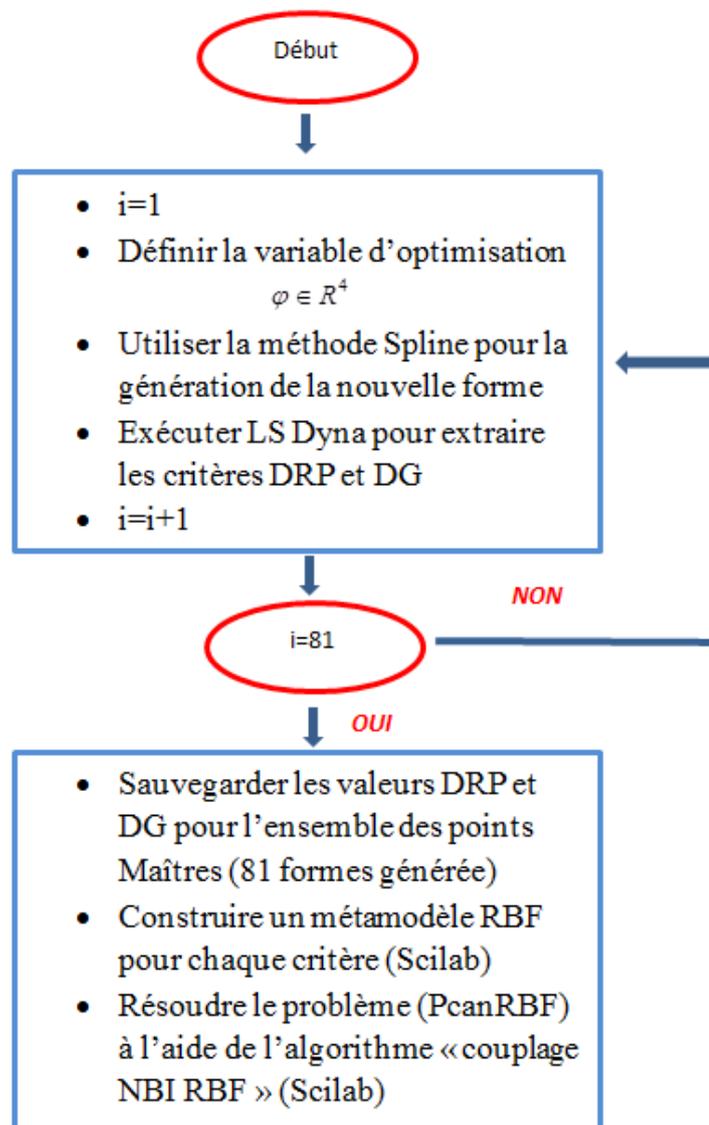


FIGURE 3.6 – Organigramme de fonctionnement du couplage NBI RBF pour le cas industriel du fond de la canette 2D - Exemple pour 81 éléments pour la base RBF

Résultats

Pour discuter de l'efficacité de notre approche pour cette application industrielle, nous analysons les résultats obtenus à plusieurs niveaux :

- Comparaison de solutions approchées par RBF, et leur calcul exact.
- Temps de calcul.
- But ou objectif industriel (forme du fond de canette avec des valeurs DRP et DG opérationnelles données par **Arcelor Mittal**) :

Cas 1 : Profils avec une DRP supérieure à **15 bars** et à une DG inférieure à **1 mm**).

Cas 2 : Profils avec une DRP supérieure à **19 bars** et à une DG inférieure à **1 mm**).

Cas 3 : Profils avec une DRP supérieure à **20 bars** et à une DG inférieure à **1 mm**).

- Vérification que les résultats sont effectivement des solutions non dominées.

Résultats - Cas 1 -

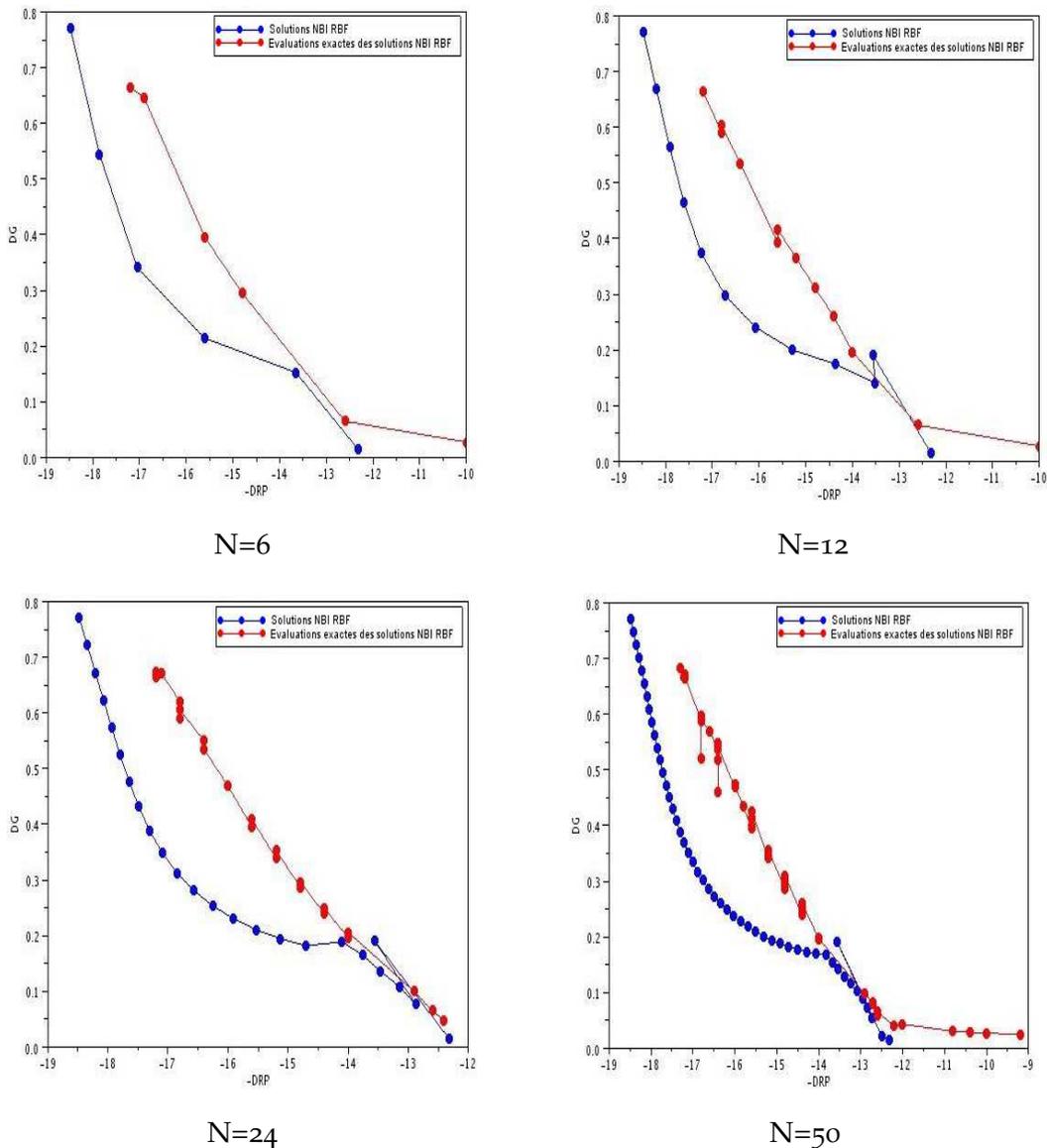


FIGURE 3.7 – Comparaison entre les résultats obtenus par l’approche NBI RBF, et l’évaluation exacte de ces résultats pour différentes valeur N de discrétisation du CHIM pour le profil de canette -Cas 1-

N	CPU total***	Fonction exacte		Fonction approchée	
		nb.éval	CPU**	nb.éval	CPU*
6	4h 21 min 07 s	87	3h 57 min 34 s	76524	23 min 33 s
12	4h 38 min 53 s	93	4h 11 min 00 s	78246	27 min 53 s
24	5h 11 min 31 s	105	4h 38 min 45 s	80784	32 min 46 s
50	6h 19 min 40 s	131	5h 47 min 20 s	80405	32 min 20 s

Tableau 3.3 – Comparaison entre le temps de calcul nécessaire pour l'évaluation des fonctions objectif exactes et des fonctions approchées au niveau de notre approche NBI RBF - Cas 1 - ((***)=(*)+(**))

Résultats - Cas 2 -

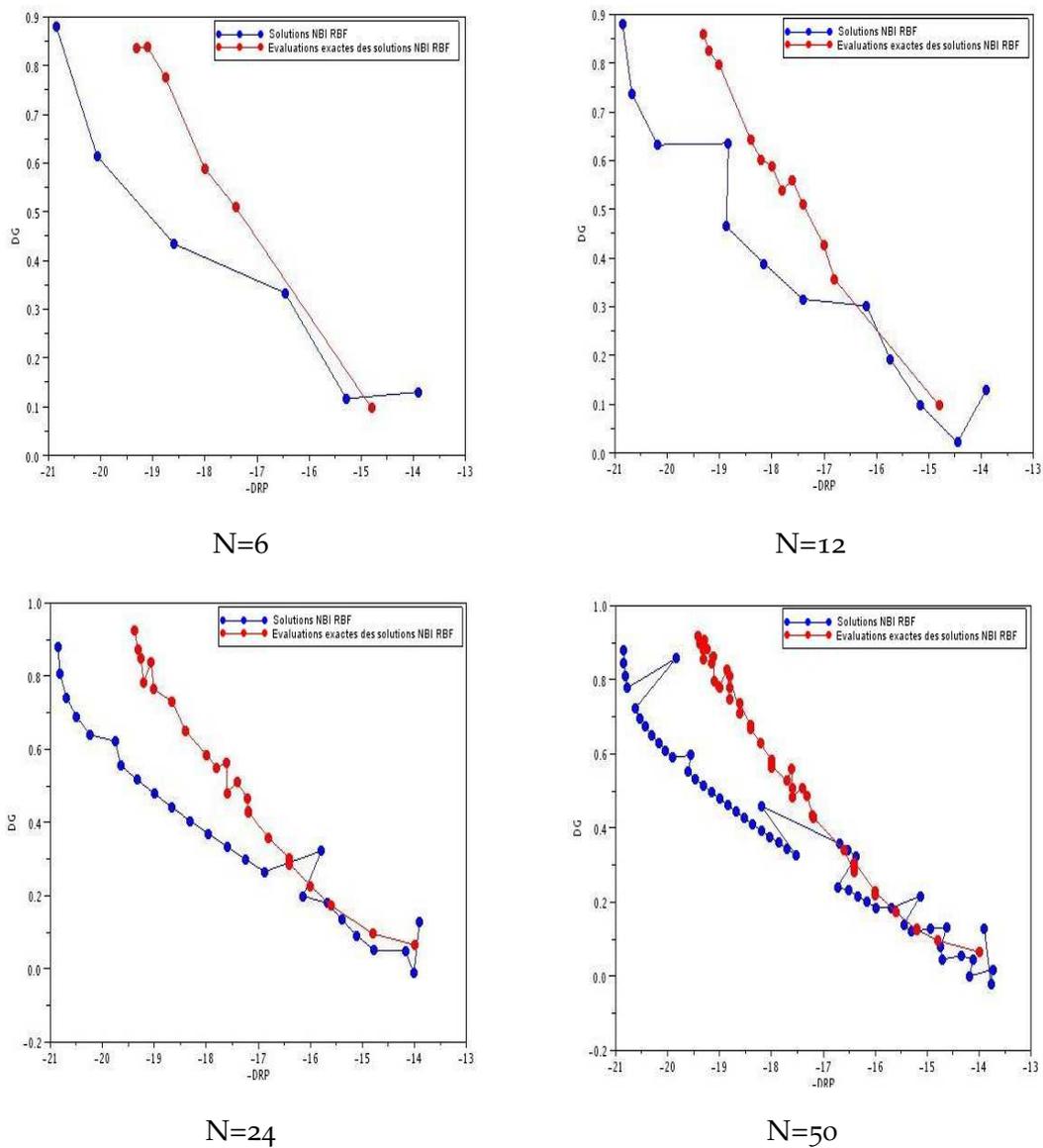


FIGURE 3.8 – Comparaison entre les résultats obtenus par l'approche NBI RBF, et l'évaluation exacte de ces résultats pour différentes valeur N de discrétisation du CHIM pour le profil de canette -Cas 2-

N	CPU total***	Fonction exacte		Fonction approchée	
		nb.éval	CPU**	nb.éval	CPU*
6	3h 46 min 10 s	87	3h 21 min 24 s	86835	24 min 46 s
12	4h 07 min 28 s	93	3h 37 min 06 s	87444	30 min 22 s
24	4h 27 min 39 s	105	4h 00 min 13 s	85193	27 min 26 s
50	5h 43 min 23 s	131	5h 06 min 55 s	90883	36 min 28 s

Tableau 3.4 – Comparaison entre le temps de calcul nécessaire pour l'évaluation des fonctions objectif exactes et des fonctions approchées au niveau de notre approche NBI RBF - Cas 2 - ((***)=(*)+(**))

Résultats - Cas 3 -

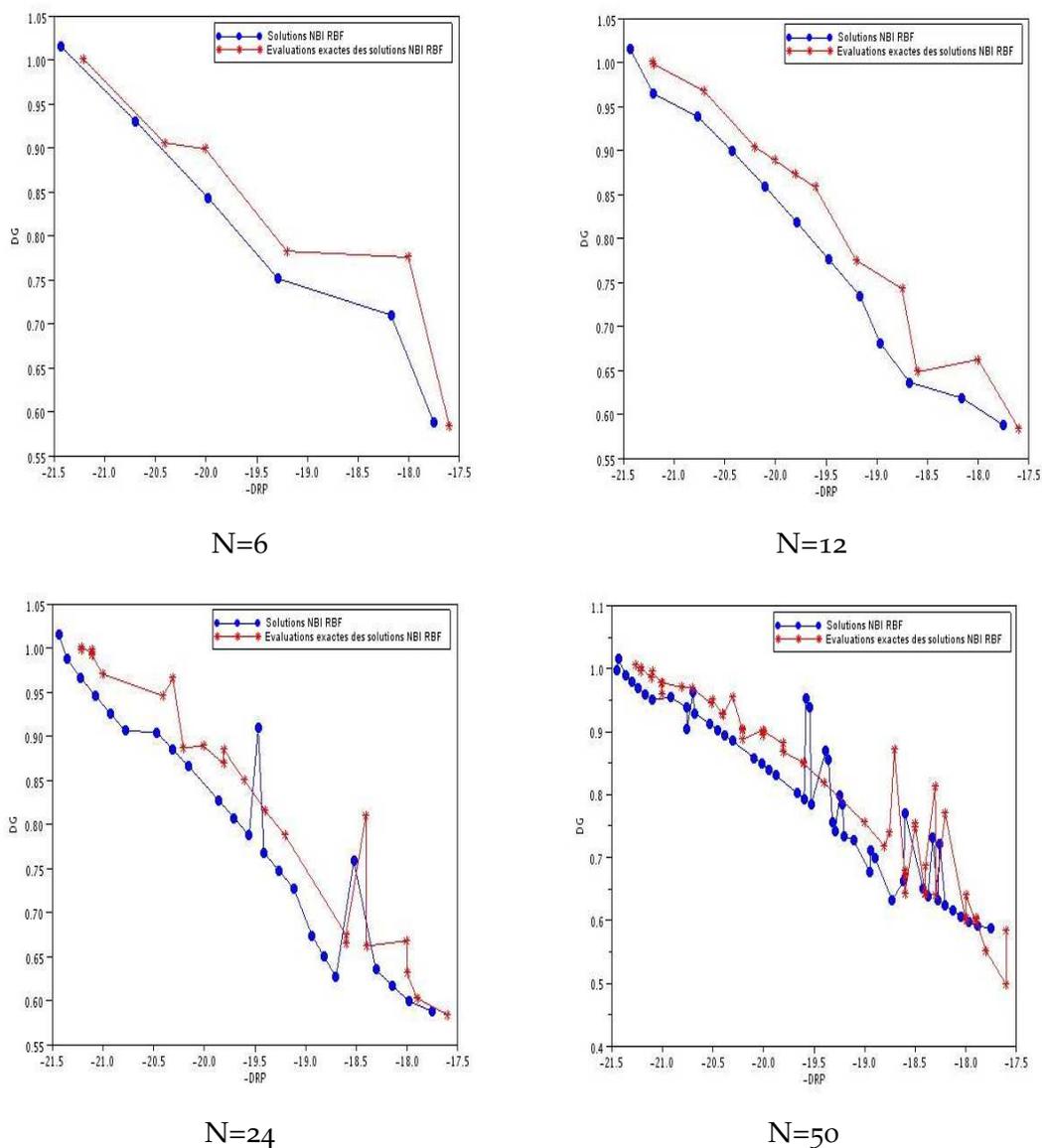


FIGURE 3.9 – Comparaison entre les résultats obtenus par l'approche NBI RBF, et l'évaluation exacte de ces résultats pour différentes valeur N de discrétisation du CHIM pour le profil de canette -Cas 3-

N	CPU total***	Fonction exacte		Fonction approchée	
		nb.éval	CPU**	nb.éval	CPU*
6	6h22min47s	87	6h01min00s	462	21min06s
12	6h40min36s	93	6h19min30s	738	21min47s
24	7h25min01s	105	7h01min37s	1472	23min24s
50	10h28min02s	131	10h01min02s	2992	27min00s

Tableau 3.5 – Comparaison entre le temps de calcul nécessaire pour l'évaluation des fonctions objectif exactes et des fonctions approchées au niveau de notre approche NBI RBF - Cas 3 - ((***)=(*)+(**))

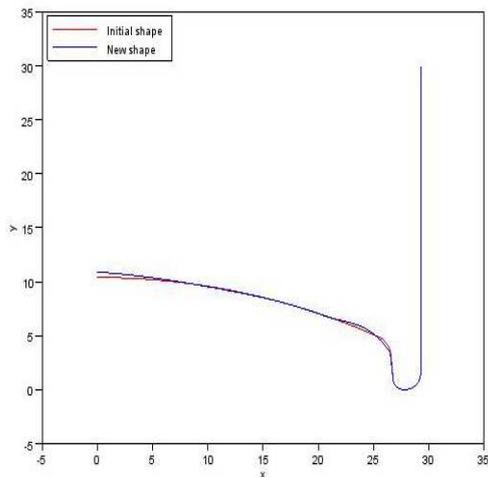
Discussion des résultats

Une simple comparaison entre les résultats obtenus par notre approche et l'évaluation exacte de ces solutions pour l'ensemble des cas traités (Figures 3.7 3.8 3.9), nous permet de juger que nos résultats restent très bons par rapport à la complexité mécanique de nos cas d'étude.

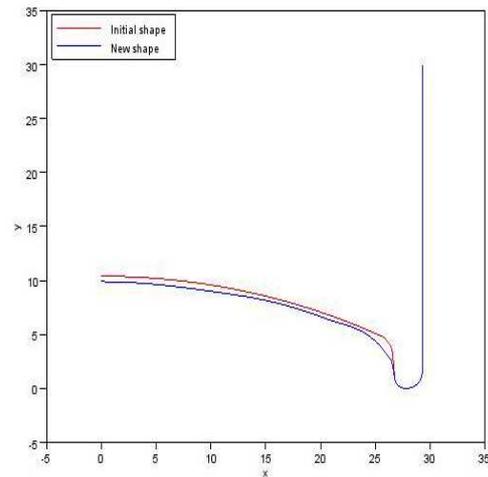
De même, il faut signaler que la résolution des problèmes avec notre approche, nous a permis de faire un gain remarquable au niveau du temps de calcul par rapport au temps exigé si nous avions choisi de les résoudre uniquement en faisant appel à la méthode NBI avec des évaluations exactes. Pour bien illustrer ce point, nous prenons par exemple le troisième cas traité avec un nombre de solutions $N=50$ représentant le nombre de points sur le front, il est clair à partir du tableau (3.5) que notre approche nous a permis de gagner un temps de calcul remarquable. En effet, la résolution du problème a demandé 131 évaluations exactes et 2992 évaluations approchées, respectivement, qui représentent 4.2% et 95.8% des évaluations totales utilisées dans notre approche, mais en même temps, nous constatons que seul ce 4.2% d'évaluations exactes prend 95.71% du temps de calcul total requis.

Il est important de souligner que l'ensemble des résultats obtenus pour la capture du front de Pareto pour ce cas industriel présente une preuve numérique de caractère concurrent des deux critères DRP et DG.

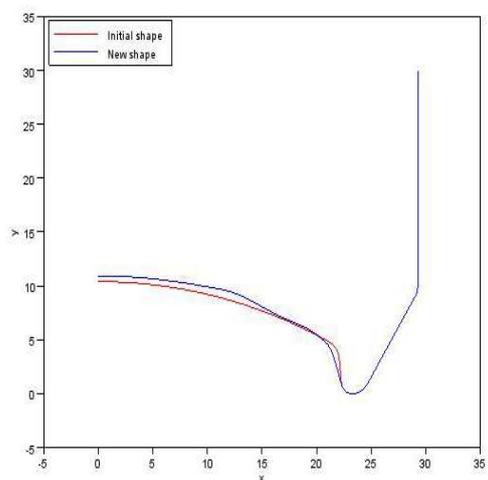
Au début de cette partie, nous avons présenté l'objectif industriel qui était la recherche de nouveaux profils pour le fond de canettes satisfaisant certaines caractéristiques, et cela pour les trois profils étudiés (rappel des caractéristiques : une valeur DRP supérieure à 15, 19 et 20 bars respectivement pour les cas 1, 2 et 3, et une valeur DG inférieure à 1 mm pour tous les cas), un objectif que nous atteignons à l'aide de notre approche utilisée qui est - **le couplage NBI RBF** -, et nous notons que dans tous les cas étudiés ($N = 6, 12, \dots, 50$), il y a toujours parmi les solutions obtenus certains profils qui vérifient les caractéristiques requises. Les profils sur la figure (3.10) présentent certains nouveaux profils proposés pour les trois cas traités.



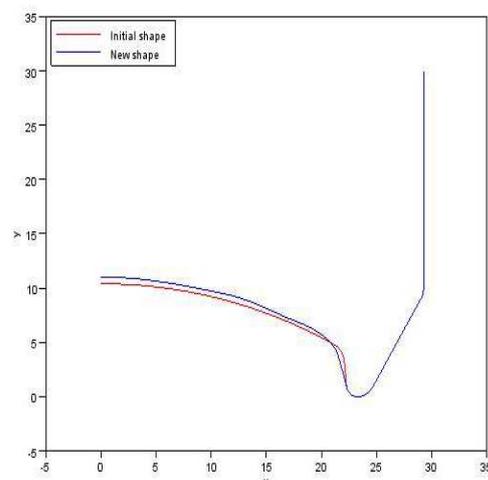
(a) $DRP = 16.1001$, $DG = 0.4646$



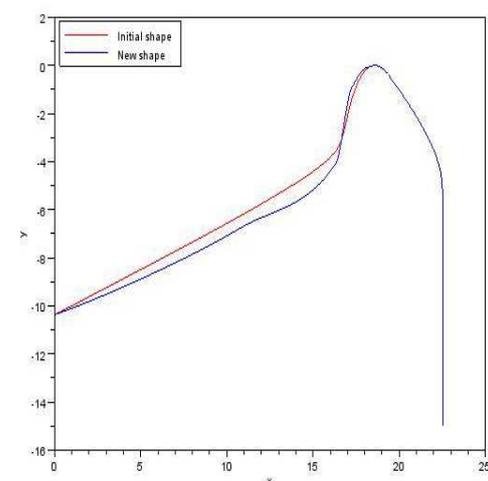
(b) $DRP = 17.2212$, $DG = 0.6754$



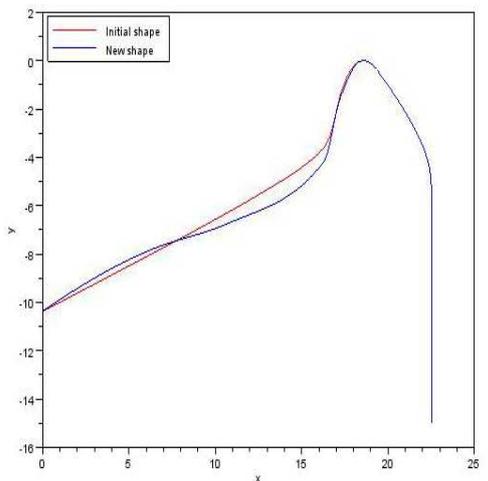
(c) $DRP = 19.4712$, $DG = 0.9312$



(d) $DRP = 19.2514$, $DG = 0.8787$



(e) $DRP = 20.2092$, $DG = 0.8977$



(f) $DRP = 21.2123$, $DG = 0.9880$

FIGURE 3.10 – Nouveaux Profils des canettes 2D vérifiant les caractéristiques requises par l'industriel pour les trois types de canette. Les valeurs de DRP et DG montrent que ces solutions sont deux à deux non dominées

Nous avons utilisé un filtre pour éliminer toutes les solutions dominées, et nous avons remarqué que l'ensemble des solutions non dominées sont presque situées sur la frontière de l'espace formé par les points maîtres de la base RBF (Figures 3.11), et en nous basant sur ces résultats, nous pouvons conclure que les solutions obtenues peuvent être considérées comme le front de Pareto capturé par la méthode NBI et que notre approche était capable de résoudre ce problème industriel avec un temps de calcul très raisonnable. Nous n'avons cependant pas de certitude qu'il n'existe pas de points dont les critères exactes dominent le front approché.

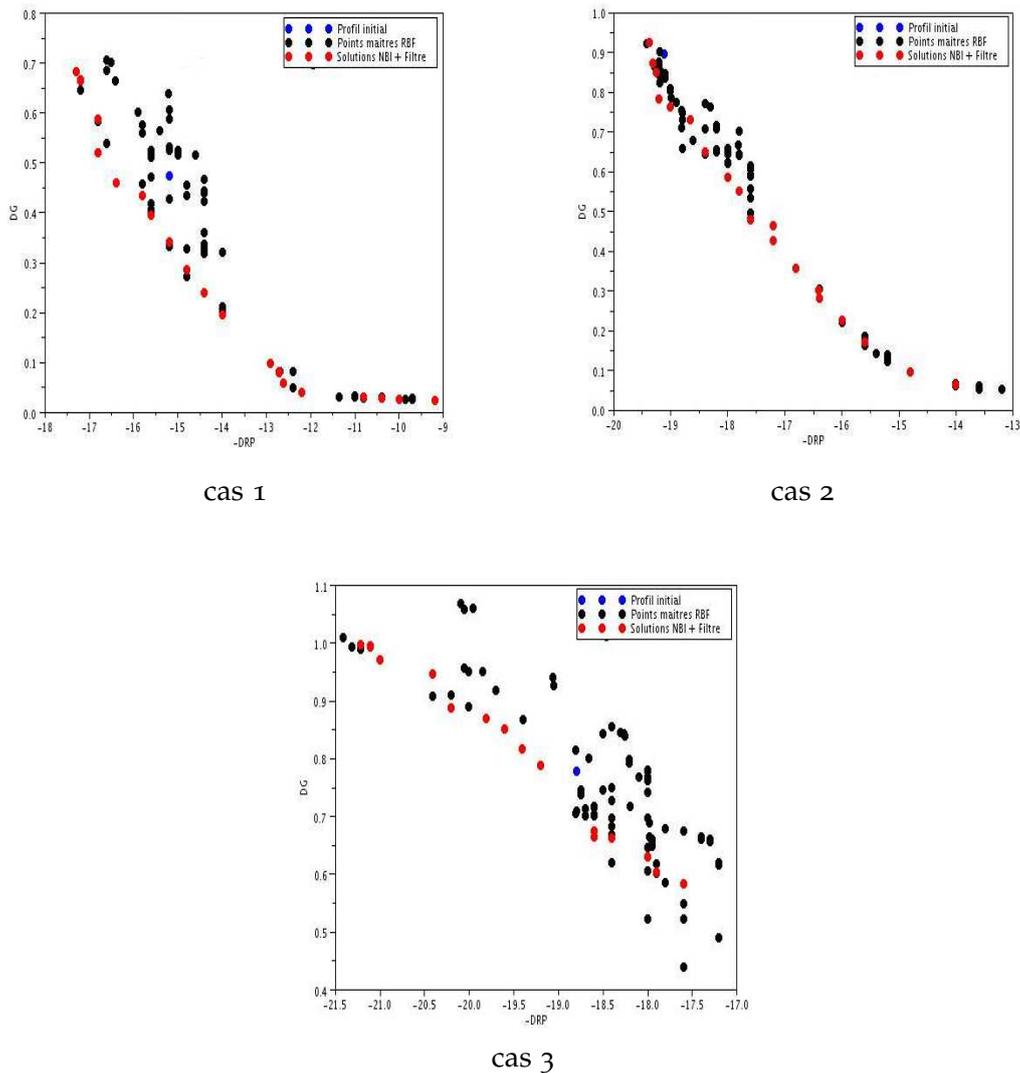


FIGURE 3.11 – Superposition des résultats obtenus par notre approche NBI RBF après filtrage (rouge), les éléments constituant la base RBF (noir) et la solution initiale (bleu) pour les trois types de canette (cas 1, cas 2 et cas 3)

Après avoir validé notre approche de **Couplage NBI RBF** pour le premier cas industriel 2D qui est l'optimisation du fond de canette avec ses trois

différents profils, nous passerons à un deuxième cas industriel, dans le domaine automobile, qui est l'optimisation de forme de la Traverse 3D.

3.3.2 Pièce automobile : Traverse 3D

Dans le cadre de la validation de notre couplage NBI RBF, nous avons utilisé un autre cas industriel qui est la traverse de voiture 3D. Le but est de réaliser une optimisation géométrique concurrente afin de chercher de nouveaux profils qui vont satisfaire les besoins des utilisateurs finaux des solutions acier proposées par **ArcelorMittal**.

Profil traverse 3D

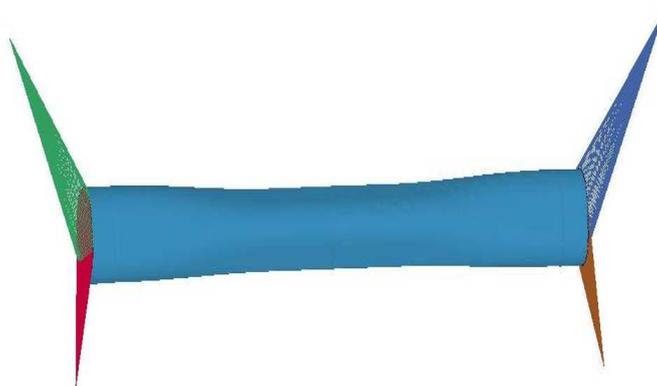


FIGURE 3.12 – Visualisation du profil de la traverse 3D sur le logiciel LSprepost de LSDyna

Formulation du problème d'optimisation

Variable d'optimisation

La première étape de l'optimisation est bien le choix de nos variables d'optimisation de notre forme, en effet, nous avons divisé le profil traverse en deux parties :

- Partie variable : c'est le ventre de la traverse et cette partie constitue réellement notre variable d'optimisation qui sera caractérisée par le choix d'un certain nombre de points de contrôle.
- Partie fixe : c'est le reste du profil qui va être fixé durant tout le processus d'optimisation.

Nous présentons, par la suite, certaines figures qui vont nous faciliter la compréhension du profil traverse et le choix des variables d'optimisation pour ce cas d'études. La figure (3.13) présente donc un zoom sur la traverse pour mettre en évidence le ventre de cette pièce et la figure (3.14) présente le ventre de la traverse 3D obtenu à l'aide d'un code Scilab et sa projection en 2D, celle-ci va nous servir d'éclaircissement du point

essentiel de cette partie à savoir le choix des variables d'optimisation.

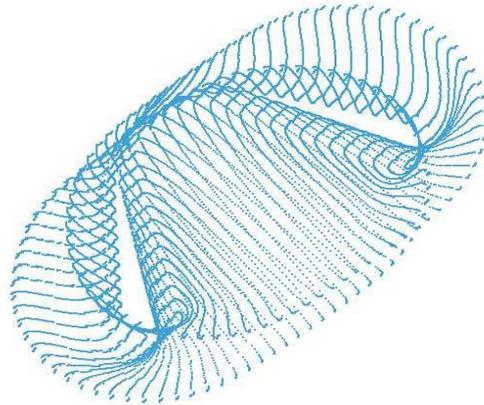
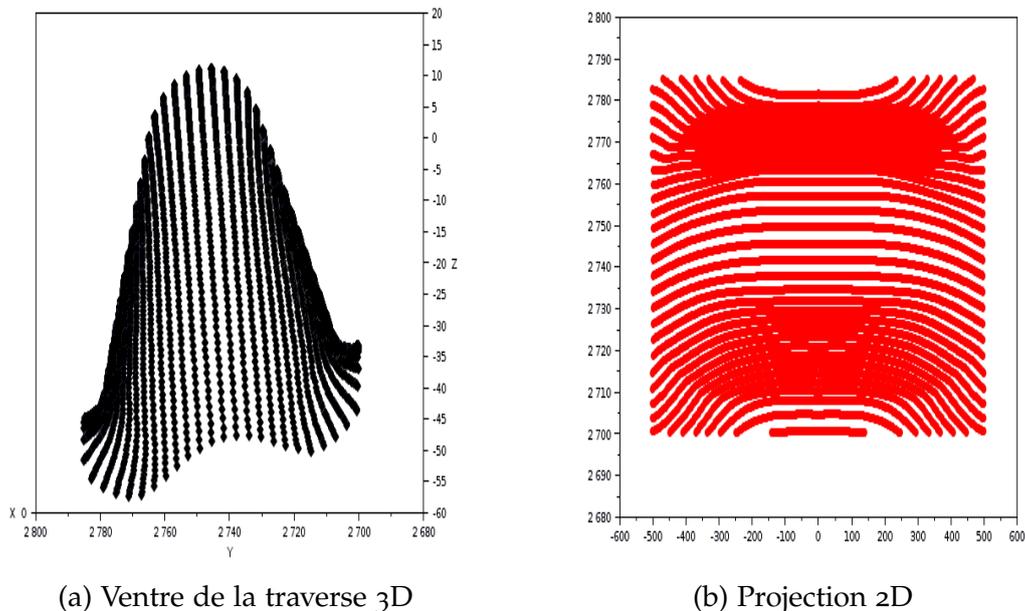


FIGURE 3.13 – Zoom sur la traverse 3D permettant de mettre en évidence le ventre de la traverse sur le logiciel LSprepost de LSDyna



(a) Ventre de la traverse 3D

(b) Projection 2D

FIGURE 3.14 – (a) Le ventre de la traverse 3D capturé par le logiciel Scilab et (b) La projection 2D du ventre 3D

Comme nous l'avons déjà signalé dans le chapitre 3, la génération des formes sera à la base de la méthode FFD par RBF expliquée auparavant, et pour bien l'appliquer dans ce cas, la frontière ou bien le bord de la partie variable de la traverse aura un déplacement nul dans toutes les directions (les noeuds constituant cette zone seront ainsi fixés), et nous allons choisir d'autres points maitres pour construire la base RBF pour la génération des formes (dans notre cas d'étude, nous avons choisi neuf points) (voir les figures 3.15).

Soit φ un profil de la traverse 3D, alors nous pouvons écrire ce dernier comme suit :

$$\varphi = (\varphi_1, \dots, \varphi_9) \quad (3.8)$$

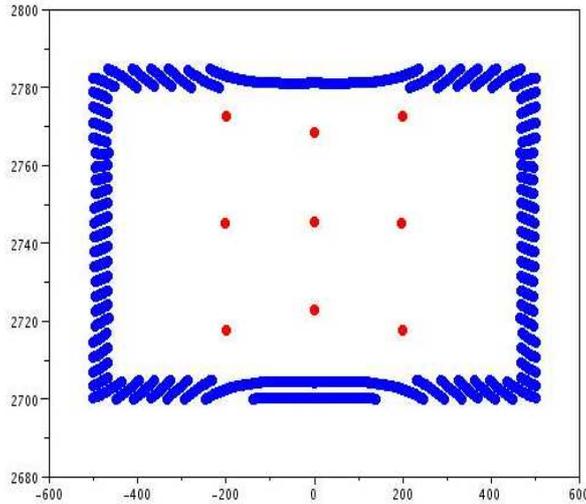


FIGURE 3.15 – Une projection 2D du ventre de la traverse 3D permettant de mettre en évidence l'ensemble des noeuds (en bleu) fixés durant l'optimisation et l'ensemble des noeuds RBF (en rouge) pour la génération des nouveaux profils pour la traverse à l'aide de la méthode FFD par RBF.

Points Maîtres RBF / Contraintes NBI

Pour les neuf points choisis comme points de contrôle constituant notre variable d'optimisation, nous allons autoriser un déplacement dans les trois directions (selon les axes x, y et z) durant tout le processus d'optimisation. De même, les coordonnées spatiales des points initiaux seront le milieu de l'intervalle présentant les contraintes NBI dans toutes les trois directions (voir la figure 3.16). Il est intéressant de signaler que ces points initiaux et les points extrêmes des intervalles vont nous servir pour constituer les points maîtres pour la construction de notre métamodèle RBF.

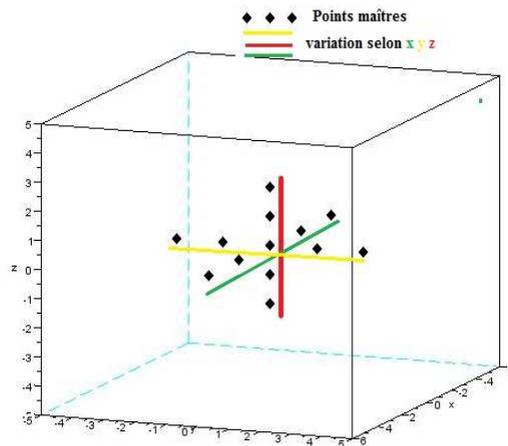


FIGURE 3.16 – Illustration de la simplification effectuée afin de réduire la dimension de notre variable d'optimisation à trois. Au début, le choix était de neuf points de contrôle et pour chaque point, nous avons cinq positions possibles (points noirs), A la fin, nous avons caractérisé chaque point par la longueur de l'intervalle dans les trois directions au lieu des cinq positions possibles sur l'intervalle. Pour l'ensemble de points, nous avons exigé la même longueur pour les intervalles.

En réalité, nous avons une variable d'optimisation avec neuf points de contrôle, et pour chaque point, nous avons choisi cinq positions possibles, d'où, nous pouvons générer $5^9 = 1953125$ formes possibles pour la traverse. Vu ce grand nombre des formes dont nous devons effectuer un calcul exact pour construire la base RBF, nous avons fait une simplification (Figure 3.16), au lieu que les neuf noeuds de la traverse soient notre variable d'optimisation, nous allons choisir plutôt la distance de l'intervalle autorisée pour les déplacements dans les trois axes comme des nouvelles variables, d'où nous pouvons choisir la même distance pour les neuf noeuds selon les trois axes, ainsi, nous aurons uniquement $5^3 = 125$ formes pour calculer la base RBF. Cette dernière va nous être utile pour la construction de notre métamodèle RBF pour approcher les deux critères à optimiser à savoir la valeur maximale de la contrainte de Von Mises sur toute la pièce (VM) et les forces de réaction (FR_{34}) (voir chapitre 3).

Après avoir donné une idée sur la manière dont nous allons gérer les variables d'optimisation, nous pouvons remplacer la formule déjà présentée pour identifier un profil de traverse (formule 3.8) par la nouvelle formule :

$$\varphi = (\varphi_{ax}, \varphi_{ay}, \varphi_{az}) \quad (3.9)$$

Soit $\varphi_0 = (\varphi_{ax0}, \varphi_{ay0}, \varphi_{az0})$ la forme initiale de la traverse, et ax , ay et az des nombres réels positifs (Dans notre cas d'étude $ax = 5$, $ay = 3$ et $az = 2$). Alors, les contraintes NBI qui sont uniquement des contraintes de bornes pour ce cas traité peuvent s'écrire à l'aide des bornes max et min de φ_0 définis comme suit :

- $\varphi^{lower} = (\varphi_{ax0} - ax, \varphi_{ay0} - ay, \varphi_{az0} - az)$
- $\varphi^{upper} = (\varphi_{ax0} + ax, \varphi_{ay0} + ay, \varphi_{az0} + az)$

Formulation mathématique

Connaissant les critères à optimiser et les contraintes à vérifier (contraintes de bornes), nous pouvons écrire le problème d'optimisation de la forme de la traverse 3D qui est un problème d'optimisation bicritère sous la forme mathématique suivante :

$$\begin{array}{c} (PTraverse) \\ \max_{\varphi=(\varphi_{axx}, \varphi_{ayy}, \varphi_{azz})} FR_{34}(\varphi) / \min_{\varphi=(\varphi_{axx}, \varphi_{ayy}, \varphi_{azz})} VM(\varphi) \\ s/c (D\ traverse) \{ \varphi^{lower} \leq \varphi \leq \varphi^{upper} \end{array} \quad (3.10)$$

Résolution du problème d'optimisation

Pour la résolution de ce problème d'optimisation bicritère de la traverse, nous avons choisi le couplage NBI RBF. L'algorithme détaillé utilisé

dans ce cas industriel est présenté dans la figure (3.17). En effet, les étapes principales sont les mêmes que celles que nous avons déjà appliquées pour le premier cas de la canette 2D à savoir :

* Construction de métamodèle pour chaque critère à optimiser et formuler Le problème approché du problème initial (Problème 3.11).

$$\begin{array}{c}
 (P_{\text{TraverseRBF}}) \\
 \max_{\varphi=(\varphi_{axx},\varphi_{ayy},\varphi_{azz})} \widetilde{FR}_{34}(\varphi) / \min_{\varphi=(\varphi_{axx},\varphi_{ayy},\varphi_{azz})} \widetilde{VM}(\varphi) \\
 s/c (D \text{ traverse}) \{ \varphi^{lower} \leq \varphi \leq \varphi^{upper}
 \end{array} \quad (3.11)$$

* Ce dernier problème sera résolu à l'aide de la méthode NBI classique.

* Dans la dernière étape, nous allons faire les évaluations exactes des solutions trouvées par notre approche, et faire une comparaison entre les deux afin de tester l'efficacité de notre approche.

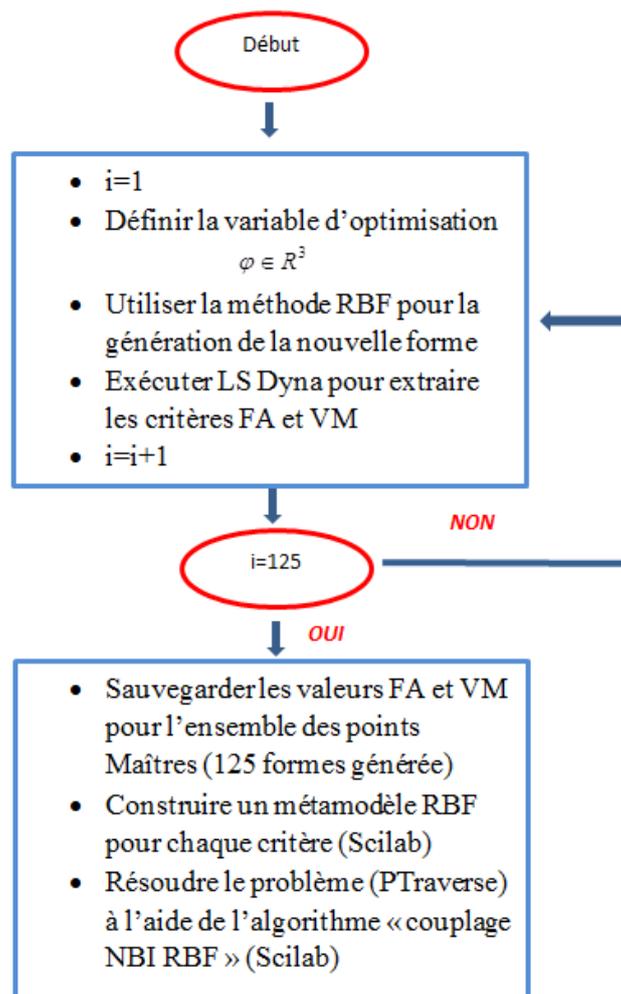


FIGURE 3.17 – Organigramme de fonctionnement du couplage NBI RBF pour le cas industriel du traverse 3D - Exemple pour une base RBF de 125 éléments

Résultats

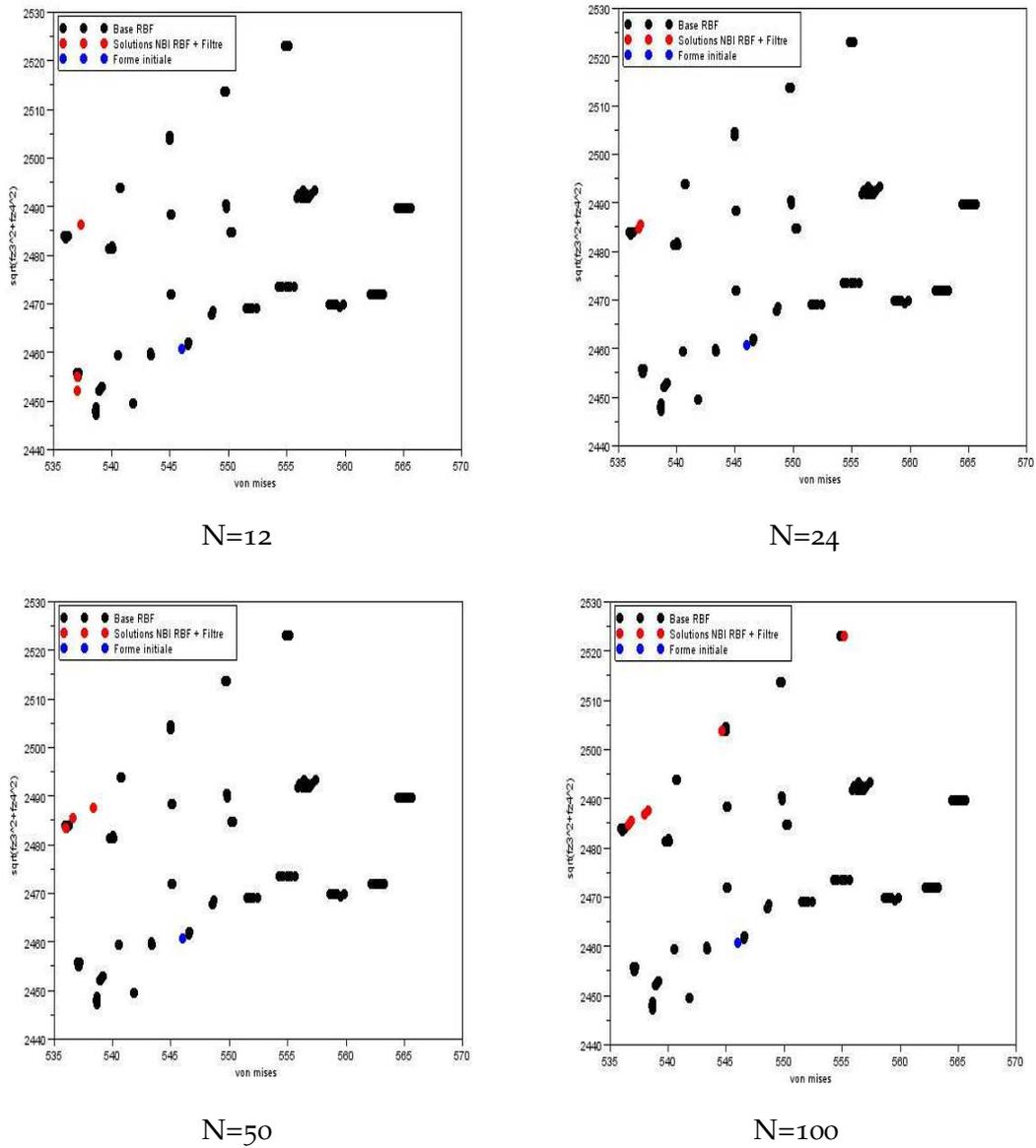


FIGURE 3.18 – Résultats de l'optimisation concurrente de forme de la traverse 3D pour différentes valeur N de discrétisation du CHIM

N	CPU total***	Fonction exacte		Fonction approchée	
		nb.éval	CPU**	nb.éval	CPU*
12	1j 21h 50 min	137	1j 21h 04 min	390	46 min
24	2j 01h 51 min	149	2j 01h 04 min	786	47 min
50	2j 09h 56 min	175	2j 09h 08 min	1584	48 min
100	3j 02h 01 min	225	3j 01h 12 min	3206	49 min

Tableau 3.6 – Comparaison entre le temps de calcul nécessaire pour l'évaluation des fonctions objectif exactes et des fonctions approchées au niveau de notre approche NBI RBF - Cas Traverse - ((***)=(*)+(**))

Discussion des résultats

Pour discuter de l'efficacité de notre approche de couplage NBI et RBF pour cette application industrielle, nous analysons les résultats selon les axes :

- Temps de calcul.
- But ou objectif industriel qui est l'identification des nouveaux profils dominant la forme initiale de la traverse 3D.
- Vérification que les résultats sont effectivement des solutions non dominées.

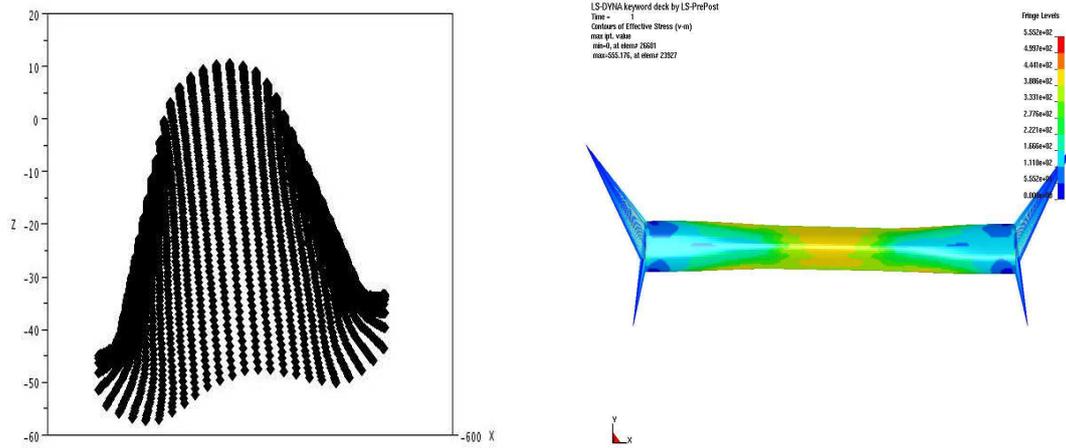
En ce qui concerne le premier point, il faut signaler que la résolution des problèmes avec notre approche, nous a permis de faire un gain remarquable au niveau du temps de calcul par rapport au temps exigé si nous avons choisi de les résoudre en faisant appel à la méthode NBI avec évaluations exactes seulement. Pour illustrer ce point, nous prenons par exemple le quatrième cas traité avec un nombre de solutions ($N=100$), il est clair à partir du tableau (3.6) que notre approche nous a permis de gagner un temps de calcul remarquable, en effet, la résolution du problème a demandé 225 évaluations des fonctions exactes et 3206 celles approchées, respectivement, qui présentent 6.55% et 93.45% des évaluations totales utilisées dans notre approche, mais en même temps, nous constatons que seul ce 6.55% prend 98.9% du temps de calcul total requis.

Cette dernière remarque explique notre choix d'éviter la résolution de ces problèmes industriels uniquement avec la méthode NBI, parce que cette dernière va nécessiter un grand nombre d'évaluations des fonctions exactes représentant les critères à optimiser, ce qui nécessitera un temps de calcul très important.

Pour le deuxième point, nous notons que dans tous les cas étudiés ($N = 12, 24, \dots, 100$), il y a toujours parmi les solutions obtenues, certains profils qui vérifient les caractéristiques requises à savoir des profils dominant le profil initial de la traverse (voir les figures 3.18).

Concernant le dernier point de cette discussion, les résultats de la figure (3.18) mettent en évidence un certain nombre de solutions deux à deux non dominées, qui sont toutes meilleures que (dominant) la solution de départ, initialement utilisée par l'industriel. Ces solutions, Pareto-optimales du point de vue du métamodèle, sont parfois à la lisière de la frontière formée par les points de la base du métamodèle, tout en n'étant dominées par aucun de ces derniers (pas de "mauvaise surprise"). L'obtention de solutions inédites non dominées (par évaluation exacte des critères), répondant aux cahier des charges de l'industriel, dont les valeur exactes des critères les placent suffisamment proche du front de Pareto du métamodèle, peut être considéré comme une preuve d'efficacité de notre approche.

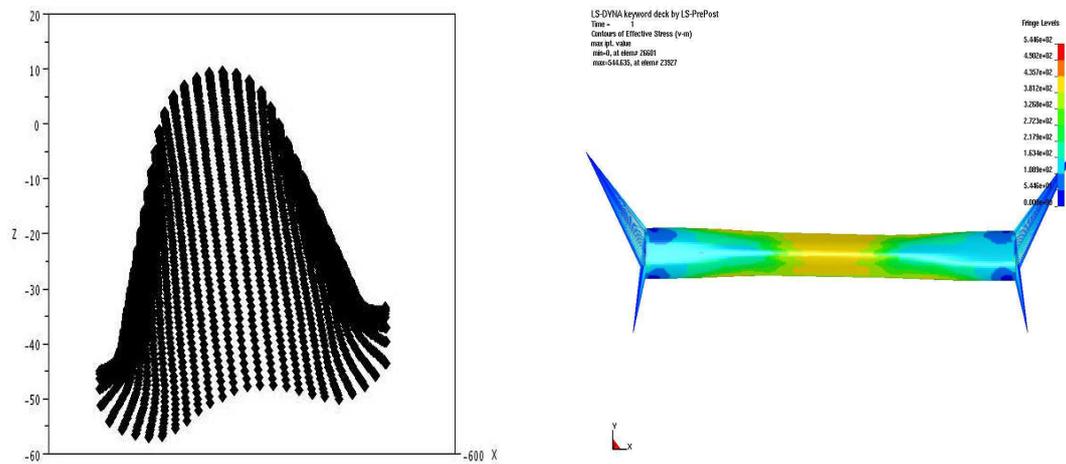
Nous terminons cette partie par la présentation de certains profils dominant la forme de la traverse 3D initiale (figures 3.19).



(a) Le ventre

(a) La traverse

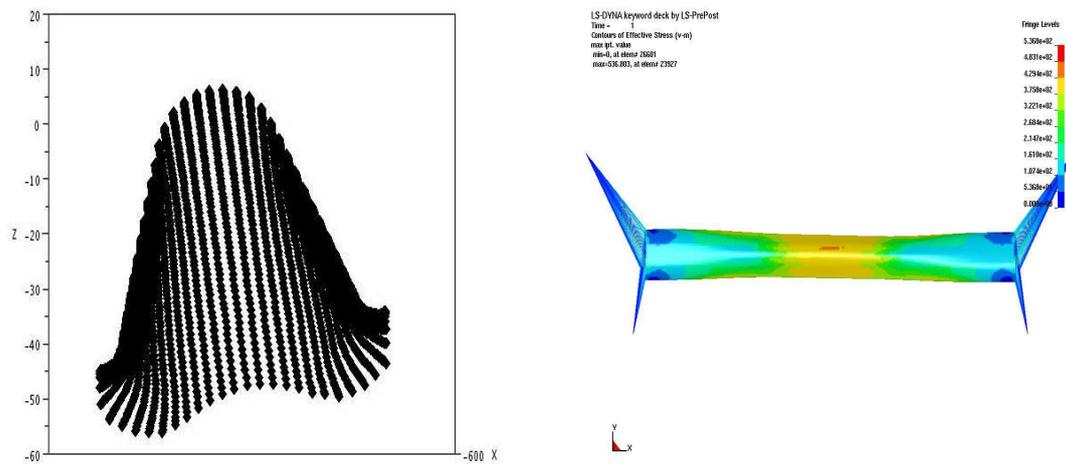
profil (a) : $VM = 555.176$; $FR_{34} = 2523$



(b) Le ventre

(b) La traverse

profil (b) : $VM = 544.635$; $FR_{34} = 2504$



(c) Le ventre

(c) La traverse

profil (c) : $VM = 536.803$; $FR_{34} = 2486$

FIGURE 3.19 – Nouveaux profils (a, b et c) de la Traverse 3D vérifiant les caractéristiques requises par l'industriel. Les valeurs de VM et FR_{34} montrent que ces solutions sont deux à deux non dominées entre elles.

Les résultats obtenus pour le problème de la traverse 3D confirment l'efficacité de notre algorithme pour la résolution des problèmes d'optimisation bicritère industriels. Cette performance peut s'expliquer par la convergence vers l'ensemble optimal de Pareto au fur et à mesure avec l'augmentation des nombres solutions NBI (Figure 3.18). Cette convergence est due surtout à la précision du métamodèle RBF. Cette précision est due à une bonne stratégie du choix des éléments de la base RBF (points maîtres) qui se base sur le principe de la distribution uniforme.

CONCLUSION DU CHAPITRE

Dans ce chapitre, nous avons validé l'efficacité de nos deux algorithmes NNCM RBF et NBI RBF pour un ensemble de cas tests académiques connus, avant de valider le dernier algorithme pour la résolution de problèmes industriels complexes, la forme du fond de la canette 2D et de la traverse 3D, proposés par notre partenaire industriel.

Les résultats obtenus pour les deux couplages effectués au niveau des cas test académiques prouvent qu'il y a possibilité de capturer le front exact en diminuant le nombre d'évaluations exactes des fonctions critères. Pour les cas industriels, nous avons utilisé le couplage NBI RBF pour résoudre les deux cas proposés, le fond de la canette 2D et la traverse 3D, et les résultats obtenus confirment l'efficacité de notre algorithme sur deux niveaux : niveau résolution du problème d'optimisation et la capture du front et le niveau des exigences industrielles.

D'après l'ensemble des résultats obtenus dans ce chapitre, il est intéressant de souligner que la capture de front de Pareto génère un ensemble des solutions non dominées. Pour savoir lesquelles choisir, le cas échéant, il est nécessaire de faire appel à des algorithmes de sélection, comme par exemple Nash et Kalai-Smorodinsky. Ces deux approches, issues de la théorie des jeux, feront l'objet du prochain chapitre.

SÉLECTION DE SOLUTIONS EFFICACES

4

SOMMAIRE

4.1	SÉLECTION PAR JEUX DE NASH	93
4.1.1	Introduction	93
4.1.2	Split arbitraire -choix arbitraire de partage-	94
4.1.3	Aperçu d'autres stratégies de partage	99
4.2	SÉLECTION PAR JEU DE KALAI-SMORODINSKY	99
4.2.1	Introduction	99
4.2.2	Validation	101
	CONCLUSION	107

DANS le chapitre précédent, nous avons pu identifier le front de Pareto pour un ensemble de problèmes académiques et industriels. Nous avons aussi signalé que cette approche permet de fournir beaucoup de bonnes solutions, mais, parfois, il faut en implémenter une seule : Comment choisir une parmi l'ensemble des solutions capturées ? Pour ce faire, nous allons présenter deux approches de sélection des solutions. Dans un premier temps, nous présentons une première approche issue de la théorie des jeux statiques, et de la notion d'équilibre de Nash. Dans un deuxième temps, nous étudierons la possibilité d'appliquer la théorie de l'équilibre de Kalai-Smorodinsky à la sélection de solutions acier efficaces.

4.1 SÉLECTION PAR JEUX DE NASH

4.1.1 Introduction

La théorie des jeux est une théorie riche et pluridisciplinaire qui vise à modéliser et optimiser le comportement d'acteurs qui prennent des décisions dans un environnement concurrentiel (plus précisément, les actions des uns ont des répercussions sur les gains des autres). C'est l'héritage de mathématiciens qui comptent parmi les plus grands du vingtième siècle, tels que Von Neumann, Nash. La théorie des jeux a des ramifications en sociologie, économie, mathématiques, recherche opérationnelle, etc. Dans cette thèse, nous présentons l'un des concepts principaux de cette théorie, qui est l'équilibre de Nash, souvent utilisé dans diverses applications d'ingénierie (finance, gestion de chaînes logistiques, industrie automobile, aérodynamique, et mécanique des structures). Plusieurs papiers ont traité de ce concept, parmi lesquels : (Cachon.2004) (Leng.2005) (Armantier.1998) et (Abou El Majd.2010).

Généralités

Dans le cadre de la théorie des jeux, l'équilibre de Nash fut proposé par John Forbes Nash. Considérons un jeu se jouant à plusieurs joueurs, où chaque joueur souhaite améliorer sa position. Le théorème de Nash stipule qu'il existe sous certaines hypothèses (théorème 1) au moins un point d'équilibre où chaque joueur ne pourra plus améliorer unilatéralement. On nomme ce point **Equilibre de Nash (EN)** (définition 1).

Soient N disciplines (joueurs) telles que chaque joueur (i) dispose d'un espace de contrôles S_i et d'un critère f_i .

$$f_i : s = (s_1, \dots, s_N) \in \prod_i S_i \mapsto f_i(s)$$

L'existence d'un équilibre de Nash dans un jeu est lié à l'existence d'un point fixe dans la correspondance de meilleure réponse. Les théorèmes de points fixes permettent ainsi de caractériser les situations dans lesquelles un équilibre existe. Le résultat le plus communément utilisé est celui de la théorème 1.

Théorème 1. *Si les ensembles S_i sont compacts, et les critères f_i s.c.i. convexes, alors il existe au moins un équilibre de Nash.*

Définition 1. Une stratégie $s^* = (s_i^*)$ est un équilibre de Nash si :

$$s_i^* \text{ est solution de } \min_{s_i \in S_i} f_i(s_i, s_{-i}^*) \quad 1 \leq i \leq N$$

Algorithmes de calcul d'un EN

Comme signalé ci-dessus, un équilibre de Nash dans un jeu est lié à l'existence d'un point fixe. En effet, un équilibre de Nash est un point fixe. Donc, tout algorithme de calcul de point-fixe multivalué peut être essayé pour déterminer l'équilibre de Nash (algorithme 10).

Algorithm 10 Calcul de l'équilibre de Nash

-
- 1: Initialisation : $s^{(0)} \in S_1 \times \dots \times S_N$
 - 2: Calcul de $\hat{s}_i = \min_{s_i \in S_i} f_i(s_i, s_{-i}^k), 1 \leq i \leq N$
 - 3: **Faire**
 - 4: $s^{(k+1)} = t \hat{s} + (1 - t)s^{(k)}$
 - 5: $k \leftarrow k + 1$
 - 6: **jusqu'à convergence**
-

Pour le critère d'arrêt ou encore la convergence, toute une étude a été menée dans (Li.1987) par exemple.

Dans la suite de cette partie de thèse, nous allons considérer uniquement sur des problèmes bicritère.

Cas des problèmes bicritère

Pour le cas bicritère, il s'agit d'un jeu à deux joueurs (critères f_1 et f_2). On considère S_1 (resp. S_2) l'ensemble des stratégies admissibles du joueur (1) (resp. du joueur (2)). Les deux joueurs agissent respectivement comme suit :

$$\begin{aligned} \text{Joueur(1)} : \quad & \min_{s^1 \in S^1} f_1(s^1, s^2) \\ \text{Joueur(2)} : \quad & \min_{s^2 \in S^2} f_2(s^1, s^2) \end{aligned}$$

où f_1 et f_2 sont deux critères définis sur $S^1 \times S^2$. Les deux joueurs cherchent à améliorer leurs propres critères parallèlement. Le but est de trouver une stratégie $(s^{1*}, s^{2*}) \in S^1 \times S^2$ qui définit l'état d'équilibre du jeu comme suit :

(Equilibre de Nash)

$$\begin{aligned} (s^{1*}, s^{2*}) &= \arg \min_{s^1 \in S^1} f_1(s^1, s^{2*}) \\ (s^{1*}, s^{2*}) &= \arg \min_{s^2 \in S^2} f_2(s^{1*}, s^2) \end{aligned}$$

4.1.2 Split arbitraire -choix arbitraire de partage-

Dans notre cas d'étude, nous considérons notre variable de recherche $x = (x_1, x_2, \dots, x_m)$ et les deux ensembles des stratégies de chaque joueur sont des variables appartenant à R^{m_1} et R^{m_2} tel que $m_1 + m_2 = m$. la problématique qui se pose en général, c'est la manière de choisir les stratégies -autrement dit le partage de la variable d'optimisation entre les joueurs-. Dans un premier temps, nous allons tester le fait d'effectuer un partage arbitraire pour des cas tests académiques et industriels déjà présentés.

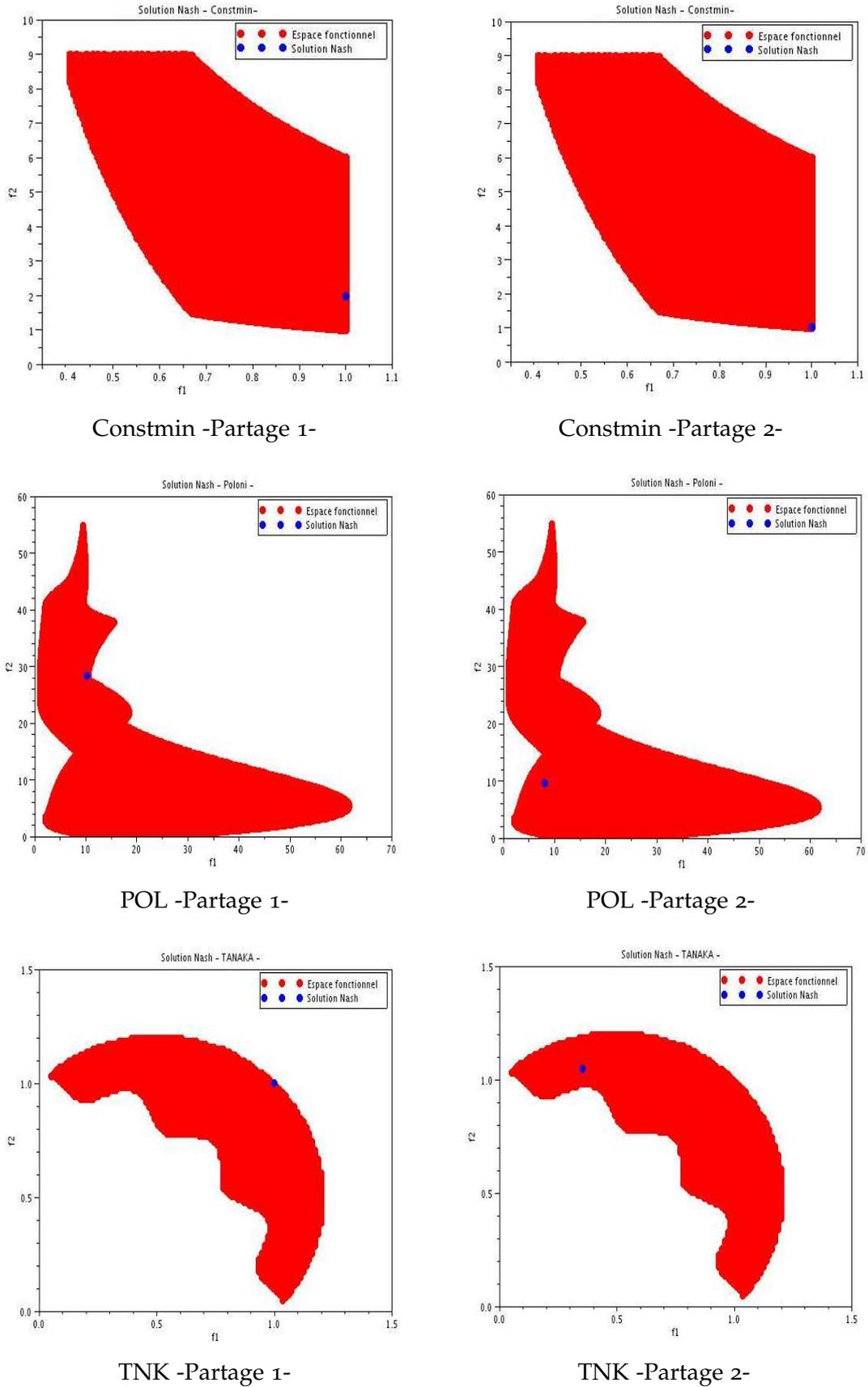


FIGURE 4.1 – Les équilibres de Nash (point bleu) pour les problèmes Constmin, TNK et POL par les deux partages arbitraires possibles

Cas Tests Industriels

Dans cette partie, nous allons calculer l'équilibre de Nash pour le problème d'optimisation du fond de canette, nous utilisons un partage arbitraire entre les deux critères à optimiser (**DRP** et **DG**). Pour faire les calculs, nous avons adopté le principe suivi dans le chapitre 3 pour la capture de front de Pareto avec les méthodes NBI et NNCM auparavant dans ce cas industriel. Nous allons résoudre le problème approché par RBF jusqu'à la convergence vers l'équilibre de Nash et faire le calcul exact de la solution approchée trouvée, comme le montre l'algorithme (12).

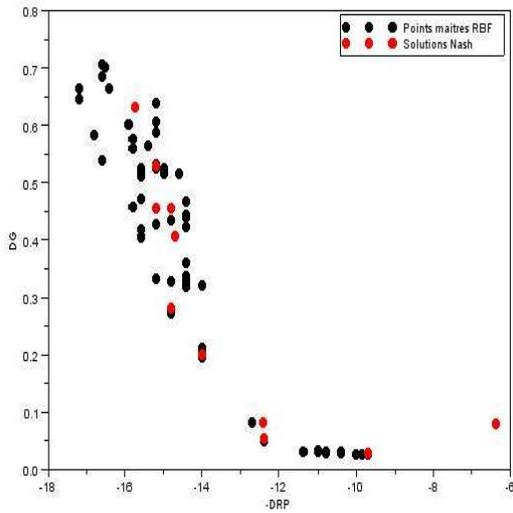
Algorithm 12 Couplage Nash RBF

- 1: $x = (x_1, x_2)$ ▷ Effectuer un partage arbitraire de la variable d'optimisation entre les fonctions objectif
 - 2: Choisir (x_1^0, x_2^0) ▷ Point de départ
 - 3: Choisir $kmax, \epsilon$ ▷ Paramètres de l'équilibre de Nash
 - 4: $k = 0$
 - 5: $x_1^{k+1} = \min_{x_1} \tilde{f}_1(x_1, x_2^k)$ ▷ Minimiser la première fonction objectif
 - 6: $x_2^{k+1} = \min_{x_2} \tilde{f}_2(x_1^k, x_2)$ ▷ Minimiser la deuxième fonction objectif
 - 7: **Tant que** $k < kmax$
 - 8: **Si** $\|x_1^{k+1} - x_1^k\| + \|x_2^{k+1} - x_2^k\| < \epsilon$
 - 9: $k = k + 1$
 - 10: Retourner à l'étape 5 ▷ Retourner à l'étape de minimisation
 - 11: **Sinon**
 - 12: *Arrêt*
 - 13: **Fin Si**
 - 14: **Fin Tant que**
 - 15: $(x_1^*, x_2^*) = (x_1^{k+1}, x_2^{k+1})$ ▷ Equilibre de Nash approché par RBF
 - 16: $(f_1(x_1^*, x_2^*), f_2(x_1^*, x_2^*))$ ▷ Evaluation exacte de l'équilibre approché
-

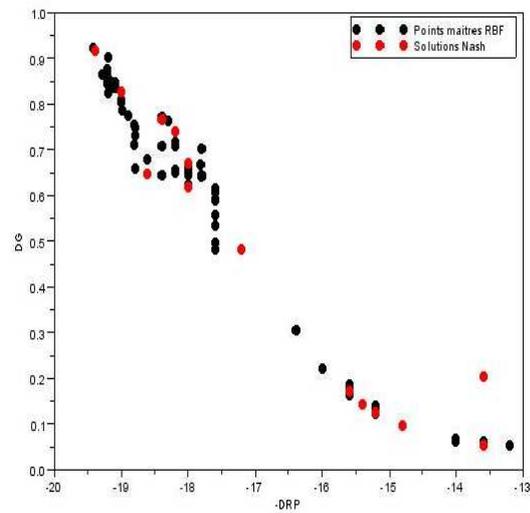
Dans nos cas d'étude, la variable d'optimisation a quatre composantes. Nous pouvons nous permettre de tester l'ensemble des scénarios représentant le partage arbitraire entre les critères (14 scénarios possibles).

Les résultats présentés dans la figure 4.2 indiquent les équilibres de Nash obtenus pour les différents partages de la variable d'optimisation entre les deux joueurs **DRP** et **DG**. Il existe certaines solutions Nash remarquables qui se trouvent sur ou sont près de la frontière de l'espace fonctionnel - front de Pareto-. Mais, malheureusement, loin de la zone d'intérêt pour les applications industrielles (zone supérieure gauche du Front de Pareto) et elles restent minoritaires par rapport aux autres solutions qui demeurent

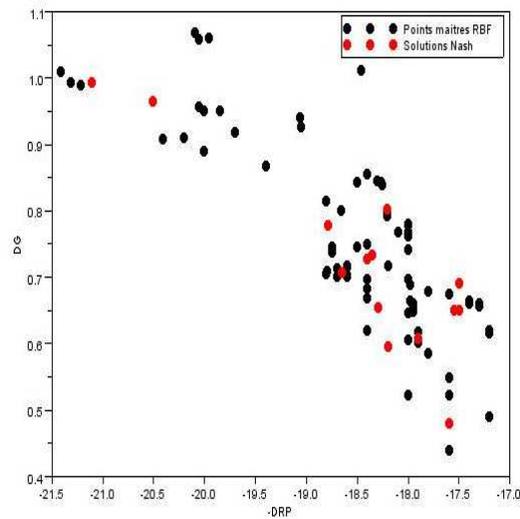
inefficaces (strictement dominée par un optimum de Pareto). D'où, on retient que le partage arbitraire est à proscrire dans ce cas d'étude industriel.



Cas 1



Cas 2



Cas 3

FIGURE 4.2 – Les équilibres de Nash (points rouges) pour le problème du fond de canette 2D par des différents partages arbitraires pour les trois types de profil de canette

Nous avons montré à travers plusieurs cas tests (académiques et industriels) que le choix d'un partage arbitraire pour le calcul d'un équilibre de Nash d'un problème d'optimisation n'est pas toujours souhaitable, car nous ne sommes pas sûr que cet équilibre soit une solution optimale au sens de Pareto, ni même, une solution proche du front de Pareto du problème traité.

4.1.3 Aperçu d'autres stratégies de partage

Ces derniers temps, plusieurs études ont été faites pour développer certains algorithmes de partage pour les jeux de Nash. Parmi ces études, nous pouvons citer :

- J.A. Désidéri a proposé un algorithme de partage de variables d'optimisation utilisant les vecteurs propres de la matrice Hessienne pour le critère le plus important. Cet algorithme de partage qu'on qualifie d'intelligent, permet une dégradation du critère de la discipline principale avec un gain au niveau du critère de la deuxième discipline, tout en garantissant que la dégradation soit la moindre possible. La difficulté majeure de cette approche de partage est le choix de la dimension de chaque espace de la stratégie pour chaque joueur d'une part, et le choix du critère le plus important d'autre part (Désidéri.2007).
- N. Moussaid a présenté une nouvelle approche de partage de variables d'optimisation, qui permet de déterminer numériquement les stratégies entre deux joueurs en utilisant la stratégie de jeux de Nash. Cette approche est basée sur la construction de deux tableaux d'allocations P et Q dont la i^{eme} composante (de P pour le joueur 1, de Q pour le joueur 2) représente la probabilité que le joueur contrôle la i^{eme} composante de la variable de décision. Moussaid a proposé et expérimenté avec succès des algorithmes qui calculent P, Q et les équilibres de Nash, en utilisant la stratégie de jeux de Nash couplée avec une extension de l'approche "Normal Boundary Intersection". Le détail de ces travaux est publiée dans sa thèse (Moussaid.2010).

Conclusion

Dans cette partie, nous avons essayé de calculer les équilibres de Nash pour un ensemble des problèmes d'optimisation académiques et industriels, et à travers les résultats obtenus, nous pouvons conclure que l'avantage majeur de cette approche est le coût de calcul raisonnable (vs celui de capture du front de Pareto), par contre, elle a certains inconvénients comme le risque d'inefficacité de la solution obtenue (EN loin du front de Pareto) ou bien le problème de partage de la variable d'optimisation. Pour éviter ces deux problèmes majeurs de l'EN, nous avons fait appel à une autre alternative à savoir l'**Equilibre de Kalai-Smorodinsky** dont les avantages sont le fait que cet équilibre est vraiment une solution au sens de Pareto et que la résolution ne nécessite pas un partage.

4.2 SÉLECTION PAR JEU DE KALAI-SMORODINSKY

4.2.1 Introduction

L'équilibre de Kalai-Smorodinsky (KS) est considéré comme l'une des alternatives de l'équilibre de Nash, il est défini par Raiffa (1953) et carac-

térisé par Kalai et Smorodinsky (1975). Cet équilibre est lié à la solution donnée par Nash pour le problème de négociation, et dans la plupart des publications traitant ce problème comme (Kalai.1975) (Nagahisa.2002) (Bozbay.2012), nous pouvons facilement observer cette relation car nous trouvons que les solutions présentées par les deux approches ont presque les mêmes caractéristiques (3 caractéristiques communes sur 4) .

L'approche de KS nous permet d'identifier un équilibre qui doit satisfaire quatre axiomes : KS est une solution efficace (solution sur le front de Pareto), indépendante, symétrique et monotone (pour plus de détails, consulter les références cités ci-dessus).

Généralités

La solution de Kalai-Smorodinsky est le point d'intersection entre le segment qui relie le point Utopia F^u et le point Nadir F^n avec le bord de l'ensemble réalisable -Espace fonctionnel- (voir figure 4.3). Le point Utopia est le point qui a comme valeur pour chaque objectif la valeur optimale de l'objectif considéré. Ce point ne correspond pas à une solution réalisable car si c'était le cas, cela sous-entendrait que les objectifs ne sont pas contradictoires et qu'une solution optimisant un objectif, optimise simultanément tous les autres, ce qui ramènerait le problème à un problème ayant une seule solution Pareto optimale. Le point Nadir est en fait à l'opposé du point utopique, ses composantes sont les valeurs des maxima des composantes de f_i sur l'ensemble de Pareto et non pas sur tout l'ensemble faisable. Une visualisation de l'ensemble de ces définitions présentant les étapes du calcul de l'équilibre de Kalai-Smorodinsky est donnée sur la figure 4.4.

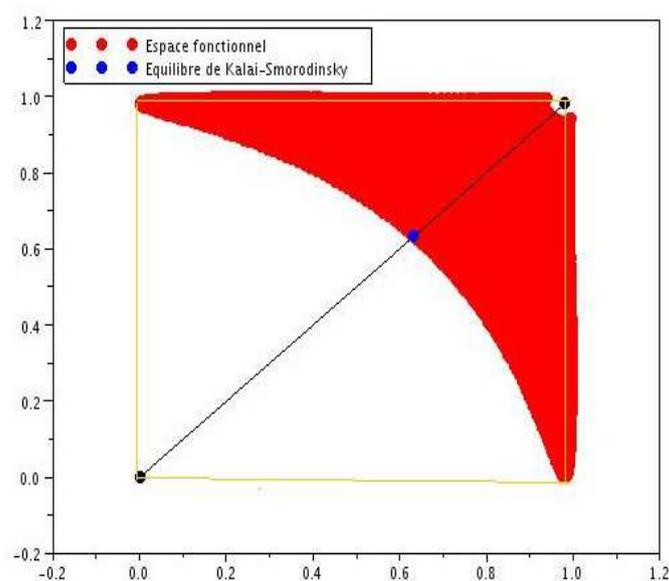


FIGURE 4.3 – La solution de Kalai-Smorodinsky KS (point bleu) pour un problème bicritère

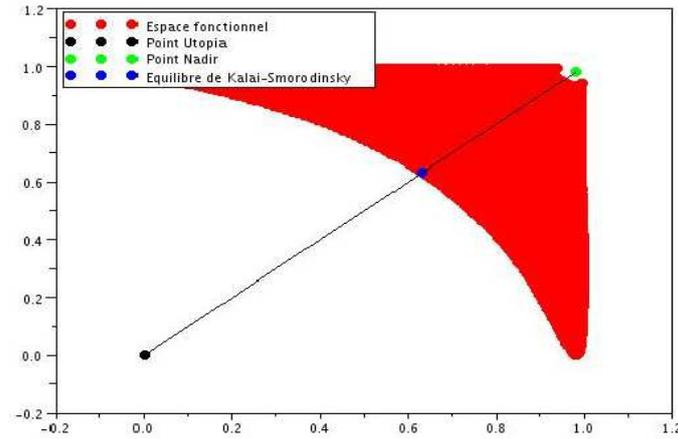


FIGURE 4.4 – Les étapes de l'identification de l'équilibre de KS : Calcul des minima individuels et le point utopia (point noir), calcul du point nadir (point vert), détermination de la ligne reliant les points utopia et nadir, et l'identification de l'équilibre de KS (point bleu).

4.2.2 Validation

Pour la validation de nos codes relatifs au calcul de l'équilibre de Kalai-Smorodinsky, nous avons utilisé les cas tests académiques et industriels utilisés auparavant pour la validation de l'équilibre de Nash afin de faire une comparaison entre les deux équilibres.

Cas Tests Académiques

En nous basant sur l'algorithme(13) qui permet de calculer l'équilibre de Kalai-Smorodinsky, nous allons le tester pour les problèmes académiques utilisés auparavant à savoir Constmin, TNK et POL.

Algorithm 13 algorithme de Kalai-Smorodinsky

- 1: $x_1^* = \min_x f_1(x)$ ▷ Minimiser la première fonction objectif
- 2: $x_2^* = \min_x f_2(x)$ ▷ Minimiser la deuxième fonction objectif
- 3: $F^u = [f_1(x_1^*), f_2(x_2^*)]^T$ ▷ Point Utopia
- 4: $F^n = [f_1(x_2^*), f_2(x_1^*)]^T$ ▷ Point Nadir
- 5: $\tau = \frac{F^u - F^n}{\|F^u - F^n\|}$
- 6: On résout le problème : ▷ Equilibre de Kalai-Smordonsky

$$\max_{x,t} t$$

$$s/c \begin{cases} F^n + t.\tau = F(x) \\ g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases}$$

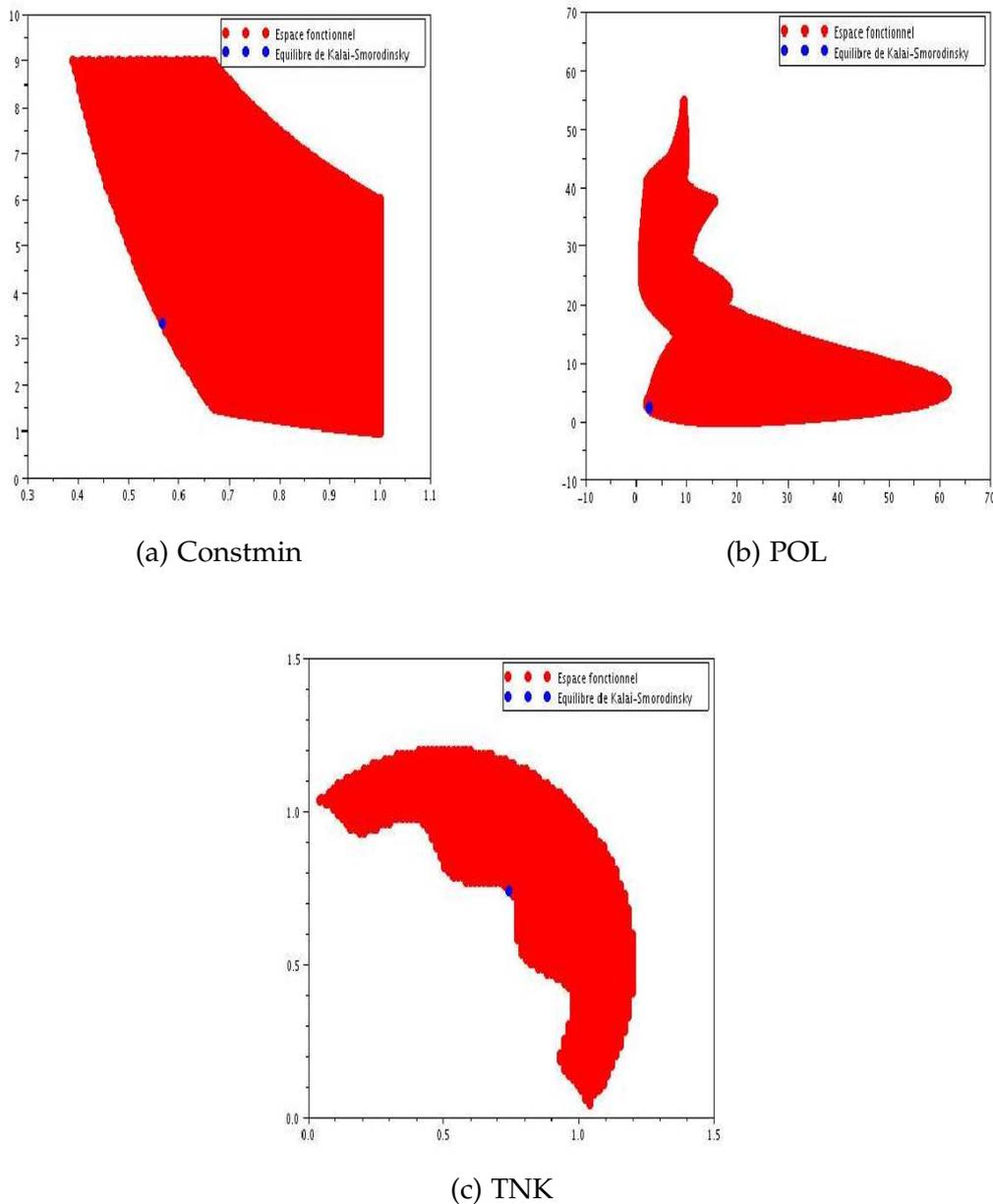


FIGURE 4.5 – Identification de l'équilibre de Kalai-Smorodinsky (point bleu) pour les problèmes (a) Constmin, (b) TNK et (c) POL

Les résultats présentés au niveau de la figure 4.5 illustrent clairement le point que nous avons signalé au début de notre chapitre à propos de l'avantage majeur de l'équilibre de Kalai-Smorodinsky à savoir que cette solution est par définition une solution optimale au sens de Pareto.

Cas Tests Industriels

Dans cette partie, nous allons calculer l'équilibre de Kalai-Smorodinsky pour le problème d'optimisation du fond de canette, et nous allons suivre le même principe que nous avons adopté pour les méthodes NBI,NNCM et Nash dans ce cas industriel à savoir résoudre

le problème approché par RBF jusqu'à la convergence vers l'équilibre de Kalai-Smorodinsky et faire le calcul exact de cette solution approchée trouvée comme le montre l'algorithme (14).

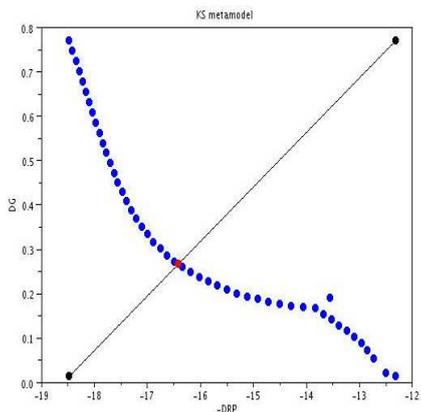
Algorithm 14 Couplage de Kalai-Smorodinsky et RBF

- 1: $x_1^{k+1} = \min_{x_1} \tilde{f}_1(x_1, x_2^k)$ ▷ Minimiser la première fonction objectif
- 2: $x_2^{k+1} = \min_{x_2} \tilde{f}_2(x_1^k, x_2)$ ▷ Minimiser la deuxième fonction objectif
- 3: $\tilde{F}^u = [\tilde{f}_1(x_1^*), \tilde{f}_2(x_2^*)]^T$ ▷ Point Utopia
- 4: $\tilde{F}^n = [\tilde{f}_1(x_2^*), \tilde{f}_2(x_1^*)]^T$ ▷ Point Nadir
- 5: $\tau = \frac{\tilde{F}^u - \tilde{F}^n}{\|\tilde{F}^u - \tilde{F}^n\|}$
- 6: On résout le problème : ▷ Formulation Kalai-Smordonsky

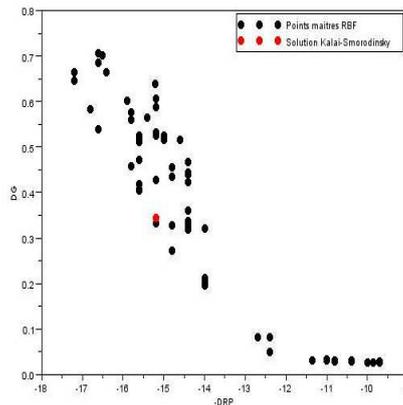
$$\begin{aligned} & \max_{x,t} \quad t \\ & s/c \quad \begin{cases} F^n + t \cdot \tau = F(x) \\ g_j(x) \geq 0 & j = 1; \dots; J \\ h_k(x) = 0 & k = 1; \dots; K \\ a_{i_{inf}} \leq x_i \leq a_{i_{sup}} & i = 1; \dots; m \end{cases} \end{aligned}$$

- 7: x^{KSapp} ▷ Equilibre de Kalai-Smorodinsky approché par RBF
 - 8: $x^{KS} = (f_1(x^{KSapp}), f_2(x^{KSapp}))$ ▷ Evaluation exacte de l'équilibre approché
-

Nous allons présenter les résultats obtenus dans cette partie en deux étapes, une première étape établit la validation de notre approche uniquement au niveau du calcul de l'équilibre de Kalai-Smorodinsky pour les critères approchés par RBF, et pour cela, nous avons effectué une superposition des solutions NBI RBF (présentant le front de Pareto approché dans ce cas) et la solution de Kalai-Smorodinsky calculée par notre code, et nous pouvons conclure à partir des figures à gauche de la figure des résultats 4.6 l'efficacité de notre code, car pour l'ensemble des cas traités, l'équilibre de KS se situe parmi les solutions NBI. Donc, cet équilibre peut être considéré comme une solution optimale au sens de Pareto. Après cette première étape, nous sommes passé à la deuxième étape de la présentation des résultats à savoir la projection réelle de la solution KS sur l'espace fonctionnel (espace constitué par le calcul exact des éléments de la base RBF), et nous remarquons clairement sur les figures à gauche de la figure des résultats 4.6 que la solution KS évaluée exactement pour les cas traités se trouve sur la frontière de l'espace, d'où une validation acceptable de notre code du couplage Kalai-Smorodinsky RBF (KS RBF).

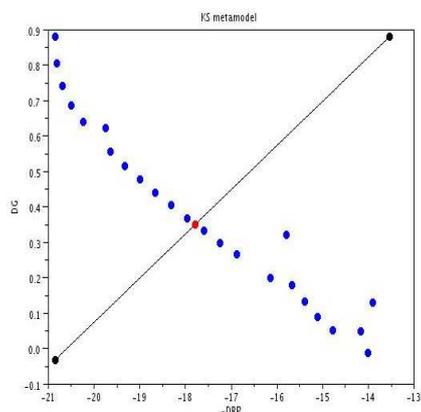


KS RBF

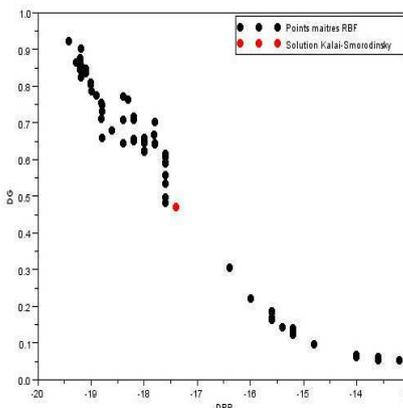


Projection réelle

(a) cas 1

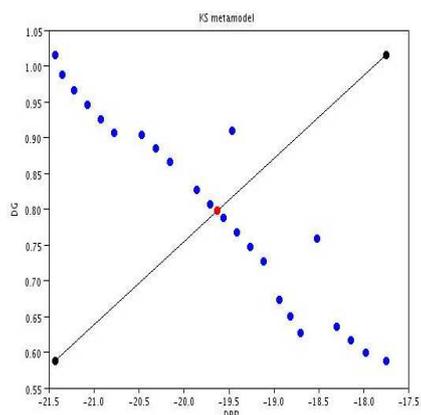


KS RBF

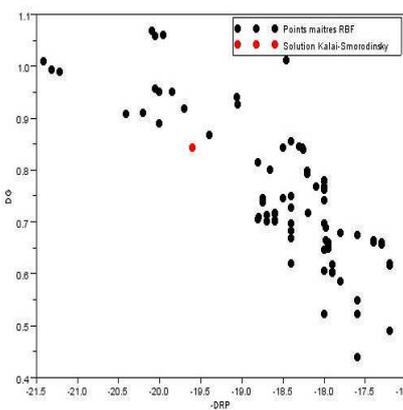


Projection réelle

(b) cas 2



KS RBF



Projection réelle

(c) cas 3

FIGURE 4.6 – Equilibre de Kalai-Smorodinsky calculé à l'aide des critères approchés par RBF (figures à gauche : KS approché est le point rouge et les points bleus présentent le front de Pareto approché par RBF) et sa projection réelle sur l'espace fonctionnel (figures à droites : KS est le point rouge) pour les trois types de profil de la canette 2D

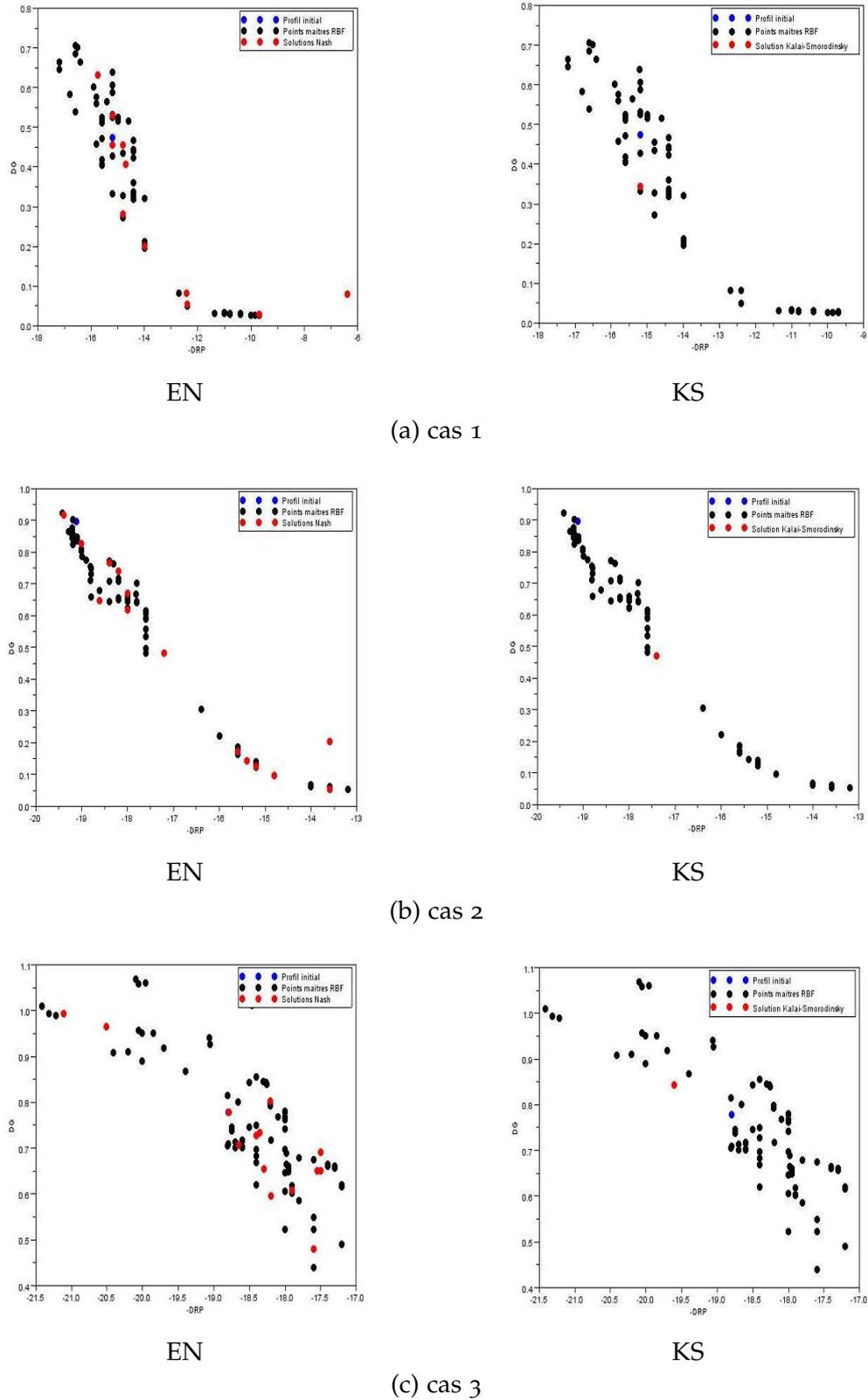
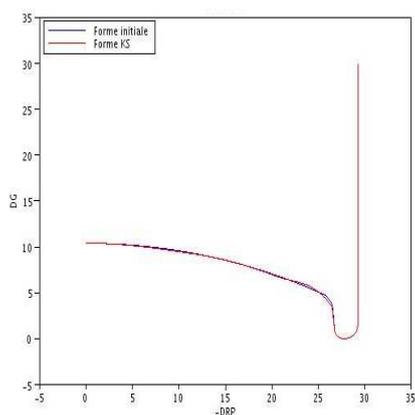


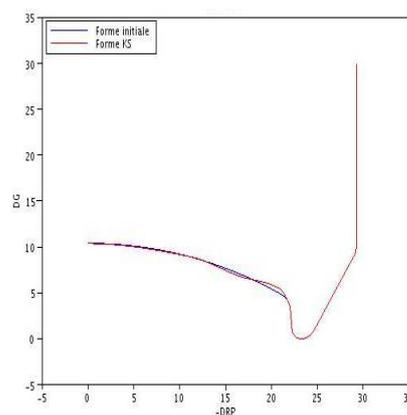
FIGURE 4.7 – Comparaison entre les positions de l'équilibre de Kalai-Smorodinsky (point rouge sur les figures à droite) et les équilibres de Nash (points rouges sur les figures à gauche) par rapport à l'espace fonctionnel (points noirs) et la solution initiale (point bleu) pour les trois types de profil de la canette 2D.

Une simple comparaison entre les résultats obtenus pour le problème de l'optimisation du fond de canette par les deux approches Kalai-Smorodinsky et Nash avec un partage arbitraire (Figure 4.7) nous a permis de relever les points suivants :

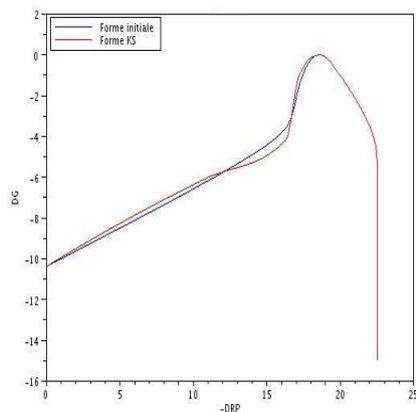
- L'approche KS permet d'avoir directement une solution optimale au sens de Pareto contrairement à l'approche EN dont nous avons effectué 14 scénarios pour le partage sans pour autant obtenir des résultats concluants.
- Parmi les 14 équilibres de Nash calculés pour les trois types de canette, nous trouverons uniquement 4, 6 et 3 solutions Pareto-optimales respectivement pour les cas d'études 1, 2 et 3. Parmi l'ensemble des solutions, il y a uniquement une seule qui se situe dans la zone intéressante pour l'industriel.
- Les solutions KS obtenues pour les cas 1 et cas 3 sont des solutions qui dominent la solution initiale.



(a) cas 1



(b) cas 2



(c) cas 3

FIGURE 4.8 – Profils correspondant aux équilibres de Kalai-Smorodinsky pour les trois types de profil de la canette.

Conclusion

A travers les remarques signalées ci-dessus, nous pouvons conclure que l'approche KS a le même avantage que celle de EN qui est le temps de calcul raisonnable, et en même temps, cette approche élimine les inconvénients majeurs de EN. En effet, la solution KS obtenue est une solution optimale et elle est identifiée sans avoir besoin de partage de la variable d'optimisation.

CONCLUSION DU CHAPITRE

Dans ce chapitre, nous avons étudié la faisabilité de deux approches "équilibre de Nash et équilibre de Kalai-Smorodinsky" pour la sélection des solutions efficaces pour des problèmes d'optimisation multicritère. Les résultats obtenus prouvent que les deux approches ont un grand avantage sur les méthodes de capture de front de Pareto qui est le temps de calcul raisonnable. Par contre, une comparaison entre les deux approches montre clairement que l'approche KS demeure meilleure que celle de EN grâce à deux points :

- La solution KS est une solution par construction Pareto-optimale contrairement à celle de EN qui ne l'est pas nécessairement.
- L'approche KS ne demande pas un partage de la variable d'optimisation comme l'exige la formulation jeux de Nash (EN).

CONCLUSION GÉNÉRALE

L'objectif de ce travail de thèse était d'étudier un ensemble de méthodes d'optimisation multicritère qui permet la résolution de problèmes de conception de forme avec un coût de calcul raisonnable. Plusieurs aspects ont été abordés pendant cette étude. Nous avons commencé par la présentation de l'optimisation multicritère, des algorithmes de capture de front de Pareto, en particulier, les algorithmes NBI et NNCM, et du métamodèle RBF. Pour les deux méthodes NBI et NNCM, nous avons montré, à travers des cas tests, que le point fort de ces deux méthodes est qu'elles génèrent un ensemble uniforme de points Pareto-optimaux équidistribués sur le front de Pareto et non pas un nombre de points concentrés dans une région donnée, ce qui permet de construire une bonne approximation de la surface Pareto. Pour les inconvénients de ces deux approches citées, nous trouverons la possibilité de générer certaines solutions dominées et le grand nombre d'évaluations des fonctions critère, ce que peut être coûteux au niveau du temps de calcul. De même, nous avons montré que la meilleure approche (au moins pour nos applications) pour la construction d'un métamodèle RBF précis est bien de générer sa base d'une manière uniforme, et d'optimiser son facteur d'atténuation à l'aide de la technique de RIPPA. On a aussi vu que la meilleure méthode (ou la plus répandue) pour la résolution d'un problème d'optimisation réel, généralement multicritère, avec un coût de calcul réduit est l'utilisation d'un algorithme de capture de front de Pareto couplé à un métamodèle.

Dans cette thèse, nous avons développé deux algorithmes basés sur ce principe de couplage, le couplage NBI RBF et celui de NNCM RBF. Ces deux algorithmes ont été utilisés dans le troisième chapitre pour résoudre, dans un premier temps, un ensemble des problèmes tests académiques complexes connus comme (SCH₁, SCH₂, HannaNC, POL, TNK, FON, P₃, et P₄). Dans un deuxième temps, nous avons validé et appliqué nos approches à deux cas industriels proposés par notre partenaire industriel. Le premier concerne un modèle 2D du fond de la canette (elasto-plasticité) et le second est un modèle 3D de la traverse (élasticité linéaire). Les résultats obtenus confirment l'efficacité de nos algorithmes développés NBI RBF et NNCM RBF. En effet, les résultats obtenus par les deux couplages pour les cas test académiques prouvent qu'il y a possibilité de converger vers le front de Pareto réel en diminuant le nombre d'évaluations exactes des fonctions critères. Pour les cas industriels, nous avons utilisé uniquement le couplage NBI RBF pour la résolution des deux cas proposés, et les résultats obtenus confirment la robustesse et l'efficacité de notre algorithme sur deux niveaux : la résolution du problème d'optimisation (i.e identification du front de Pareto) et les exigences industrielles, tout en

ayant un gain considérable en terme de temps de calcul.

Pour les cas industriels traités, nous avons succinctement décrit les modèles mécaniques (dans le deuxième chapitre), avec les détails des procédures de calcul effectif de réponses elasto-plastique et élastiques, ainsi que celles de calcul de critères. Nous avons également décrit les méthodes de la variation ou la déformation de la forme du fond de la canette en 2D (Spline cubique) et la traverse en 3D (FFD par RBF).

L'identification du front de Pareto pour l'ensemble de problèmes académiques et industriels nous a permis de fournir beaucoup de bonnes solutions, mais, parfois, il faut en implémenter une seule : comment choisir une parmi l'ensemble des solutions capturées ? Pour ce faire, nous avons introduit, au niveau de chapitre 4, deux approches de sélection des solutions. Dans un premier temps, nous avons présenté une première approche issue de la théorie des jeux statiques qui est l'équilibre de Nash (EN). Dans un deuxième temps, nous avons étudié la possibilité d'appliquer la théorie de l'équilibre de Kalai-Smorodinsky (KS) à la sélection de solutions efficaces. Pour l'équilibre de Nash, nous avons utilisé un partage arbitraire de la variable d'optimisation entre les critères à optimiser, et nous avons prouvé que cette méthode n'est pas assez fiable car il y a le risque d'inefficacité des solutions (EN loin de front de Pareto). Pour surmonter cette problématique, nous avons eu recours à la deuxième alternative qui est l'équilibre de Kalai-Smorodinsky. Ce dernier nous a permis d'avoir un équilibre (solution) efficace. Pour les deux approches, il faut signaler un avantage majeur qui est le coût de calcul qui reste raisonnable par rapport à celui de capture du front de Pareto.

A l'avenir, il serait intéressant de tester le couplage des méthodes NBI et NNCM avec d'autres types de métamodèle pour résoudre les problèmes d'optimisation multicritère industriels. Il serait aussi intéressant de tester la possibilité d'ajouter autant de points maîtres à chaque itération (actualisation et enrichissement des métamodèles).

Un autre axe de recherche pourrait être l'étude d'autres approches pour la partage de la variable d'optimisation pour le calcul de l'équilibre de Nash, comme celles introduites et étudiées par Désidéri (Désidéri.2007) et Moussaid (Moussaid.2010).

ANNEXES

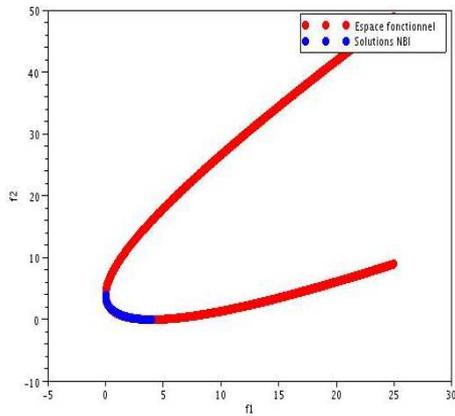
A

SOMMAIRE

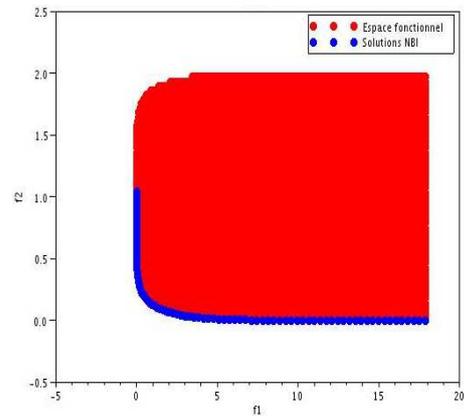
A.1	VALIDATION DE LA MÉTHODE NBI	113
A.2	USAGE D'UN FILTRE POUR LA MÉTHODE NBI	114
A.3	VALIDATION DE LA MÉTHODE NNCM	115
A.4	USAGE D'UN FILTRE POUR LA MÉTHODE NNCM	116
A.5	ÉTUDE COMPARATIVE ENTRE LA DISTRIBUTION UNIFORME ET LE LATIN HYPERCUBE POUR LE CHOIX DES ÉLÉMENTS DE LA BASE RBF (POINTS MAÎTRES)	117
A.6	VALIDATION DU MÉTAMODÈLE RBF POUR DES FONCTIONS 1D .	121
A.7	VALIDATION DU MÉTAMODÈLE RBF POUR DES FONCTIONS 2D .	122
A.8	VALIDATION DE L'APPROCHE NBI RBF	123
A.9	OPTIMISATION DES PROFILS DE LA CANETTE 2D	124
A.10	EQUILIBRE DE NASH -PARTAGE ARBITRAIRE-	127
A.11	EQUILIBRE DE KALAI-SMORODINSKY	128

A.1 VALIDATION DE LA MÉTHODE NBI

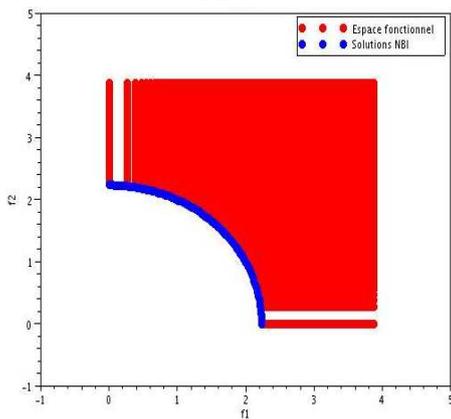
La méthode NBI est validée sur un ensemble de problèmes d'optimisation multicritère connus et difficiles tels que SCH1, Messac, Hanna non convexe, ZDT1, Deb et Hanna discontinu.



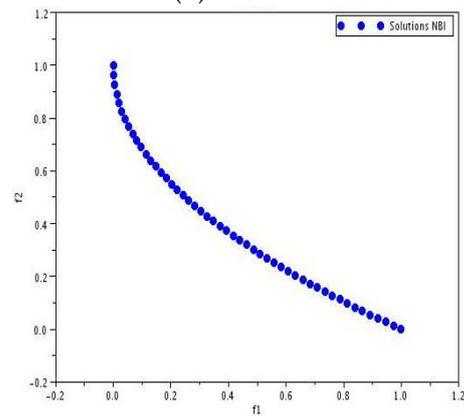
SCH1



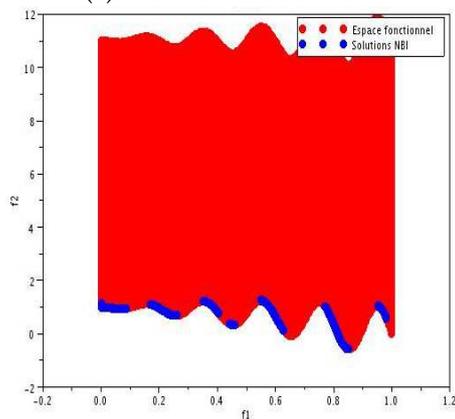
(b) Messac



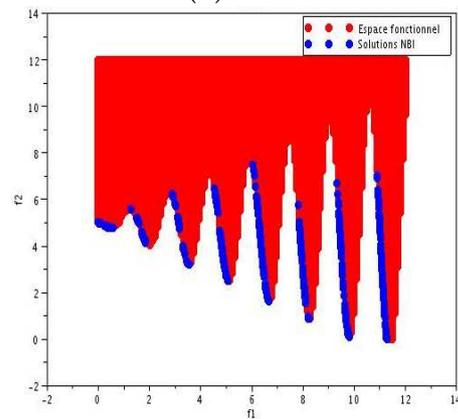
(c) Hanna non convexe



(d) ZDT1



(e) Deb



(f) Hanna discontinu

FIGURE A.1 – Résolution des problèmes (a) SCH1, (b) Messac, (c) Hanna non convexe, (d) ZDT1, (e) Deb et (f) Hanna discontinu par la méthode NBI, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

A.2 USAGE D'UN FILTRE POUR LA MÉTHODE NBI

Comme déjà signalé tout en long des chapitres, la méthode NBI a un inconvénient majeur, elle peut générer des solutions dominées, d'où la nécessité d'un filtre pour éliminer ces solutions (cas de problème de Deb et Hanna discontinu).

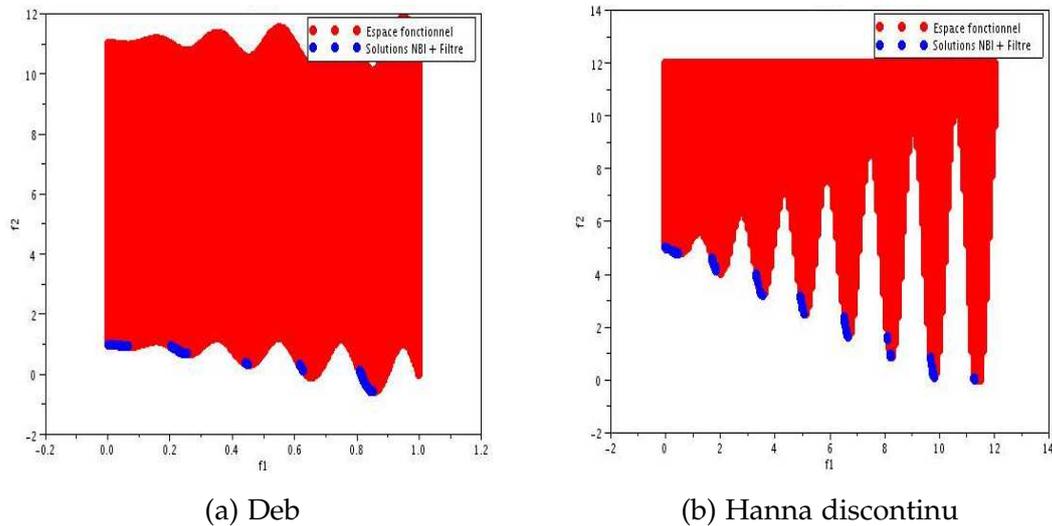


FIGURE A.2 – Usage d'un filtre pour éliminer les solutions dominées obtenues par la méthode NBI pour les problèmes (a) Deb et (b) Hanna discontinu

A.3 VALIDATION DE LA MÉTHODE NNCM

La méthode NNCM est validée sur un ensemble de problèmes d'optimisation multicritère connus et difficiles tels que Max-Ex, Const-Min-Ex, P₃, ZDT2, Messac-mod et Tanaka-mod.

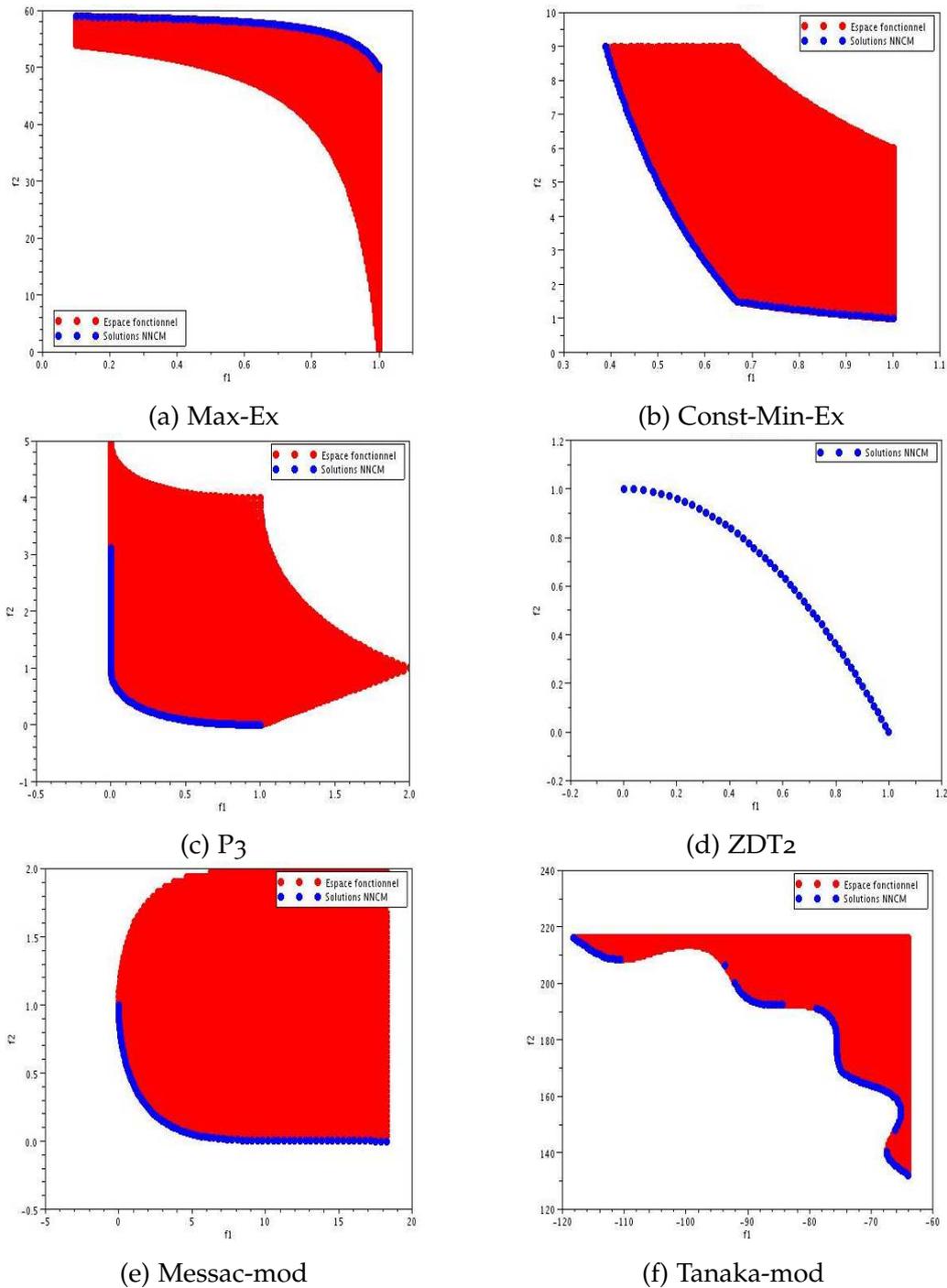


FIGURE A.3 – Résolution des problèmes (a) Max-Ex, (b) Const-Min-Ex, (c) P₃, (d) ZDT2, (e) Messac-mod et (f) Tanaka-mod par la méthode NBI, les solutions obtenues sont en bleu et l'espace fonctionnel en rouge

A.4 USAGE D'UN FILTRE POUR LA MÉTHODE NNCM

Comme la méthode NBI, la méthode NNCM a aussi le même inconvénient majeur qui est la possibilité de générer des solutions dominées, d'où la nécessité d'un filtre pour éliminer ces solutions (Cas de problème de Tanaka-mod et Tanaka).

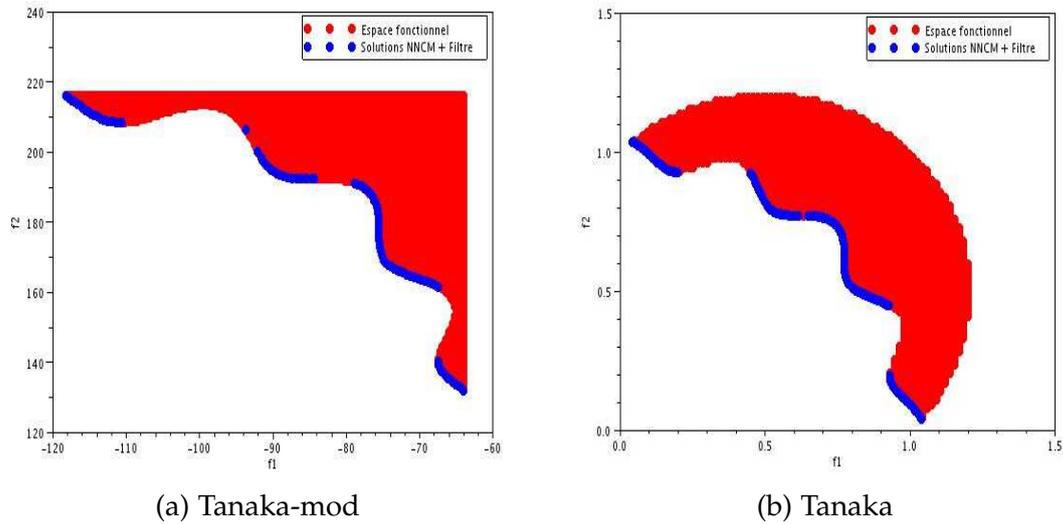


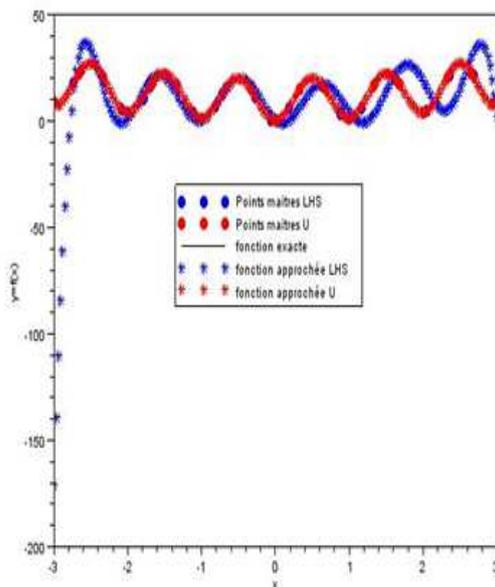
FIGURE A.4 – Usage d'un filtre pour éliminer les solutions dominées obtenues par la méthode NNCM pour les problèmes (a) Tanaka-mod et (b) Tanaka

A.5 ETUDE COMPARATIVE ENTRE LA DISTRIBUTION UNIFORME ET LE LATIN HYPERCUBE POUR LE CHOIX DES ÉLÉMENTS DE LA BASE RBF (POINTS MAÎTRES)

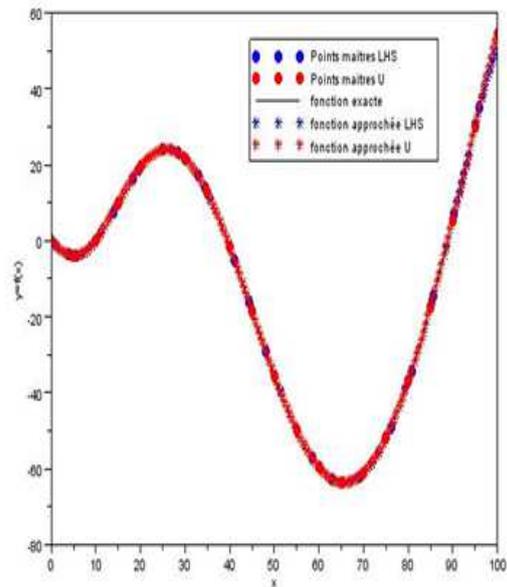
Comme déjà signalé, nous avons choisi la distribution uniforme des points maîtres pour la construction de la base du métamoèle RBF. Ce choix est fait à la base des résultats obtenus par une étude comparative entre les deux méthodes (latin hypercube et distribution uniforme). Cette étude est validé sur des cas tests académiques et des cas industriels (les critères DRP et DG).

* Fonctions académiques

* Fonctions à une variable



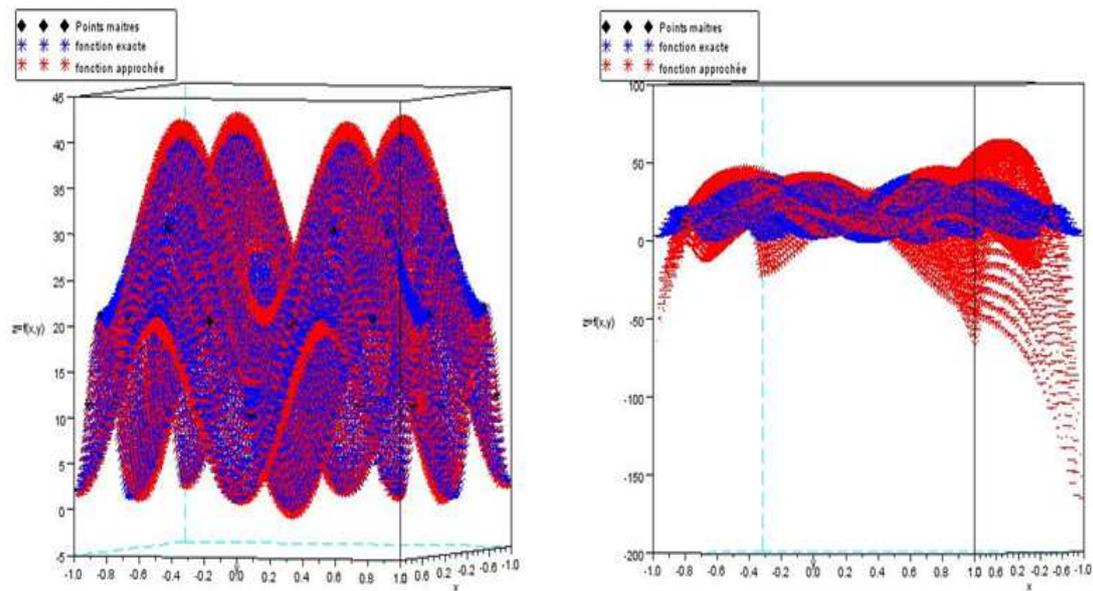
(a) Rastrigin 1D



(b) Schwefel

FIGURE A.5 – Approximation des fonctions (a) Rastrigin 1D et (b) Schwefel par RBF avec le même nombre total ($N = 15$) de points maîtres avec les méthodes de génération de points : distribution uniforme (rouge) et latin hypercube (bleu)

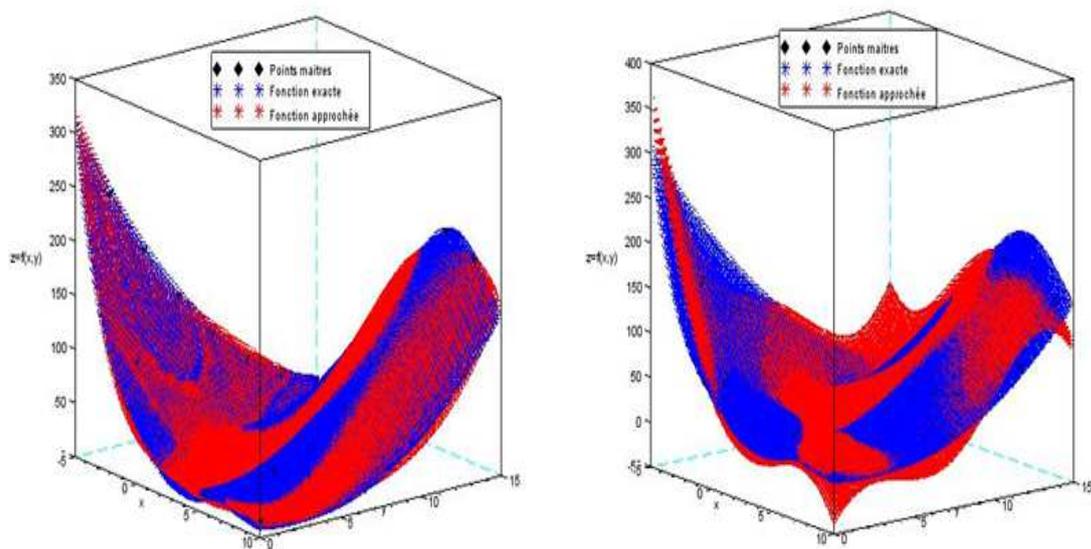
* Fonctions à deux variables



(a) Distribution Uniforme

(b) Latin Hypercube

Fonction Rastrigin 2D



(a) Distribution Uniforme

(b) Latin Hypercube

Fonction Branin

FIGURE A.6 – Approximation des fonctions Rastrigin 2D et Branin par RBF avec le même nombre total ($N = 81$) de points maîtres avec les méthodes de génération de points : distribution uniforme (figures à gauches) et latin hypercube (figures à droites) –fonctions exactes en rouge et fonctions approchées en bleu–

* Critères industriels

* Critère DRP

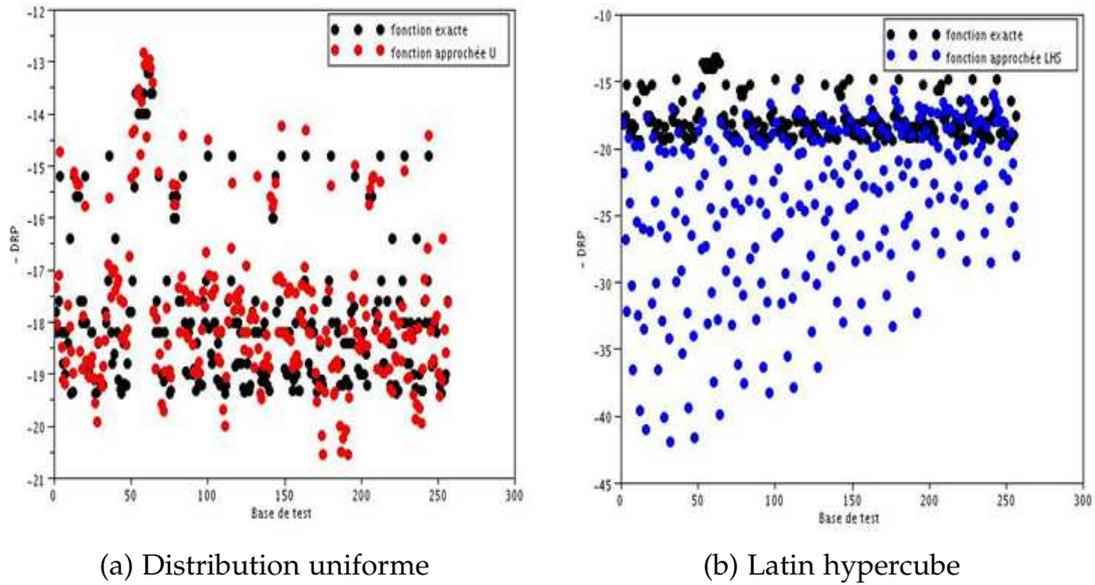


FIGURE A.7 – Approximation du critère DRP avec le même nombre total ($N = 81$) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas2.

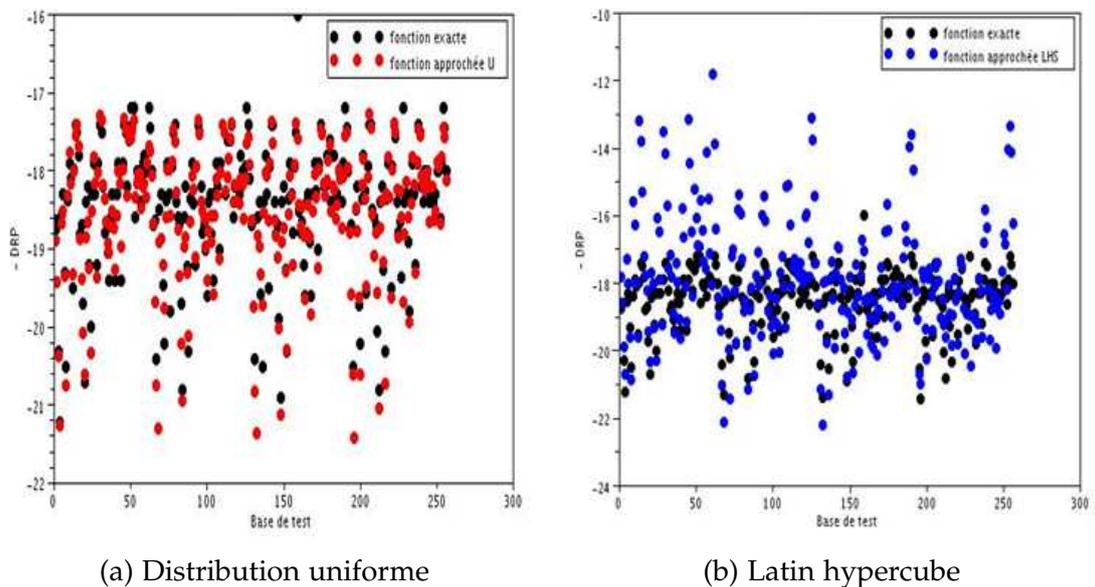
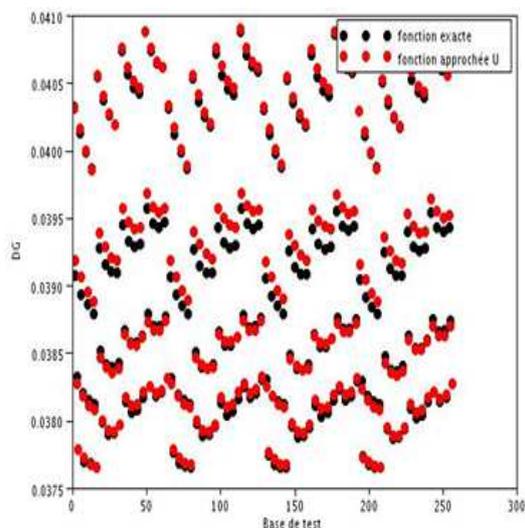
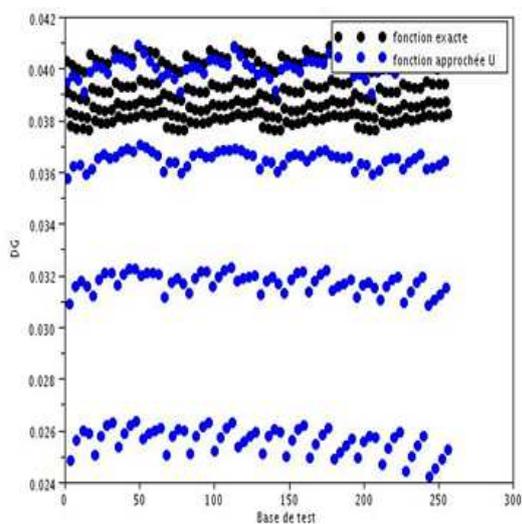


FIGURE A.8 – Approximation du critère DRP avec le même nombre total ($N = 81$) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas3

* Critère DG

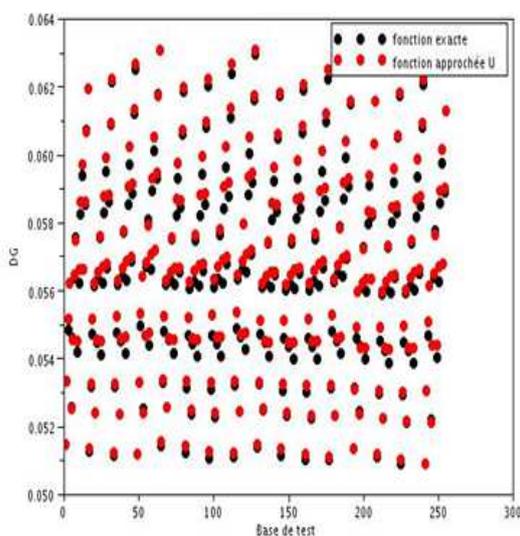


(a) Distribution uniforme

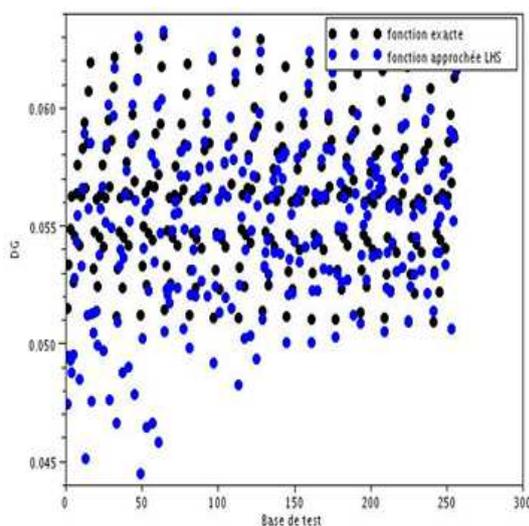


(b) Latin hypercube

FIGURE A.9 – Approximation du critère DG avec le même nombre total ($N = 81$) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas2



(a) Distribution uniforme

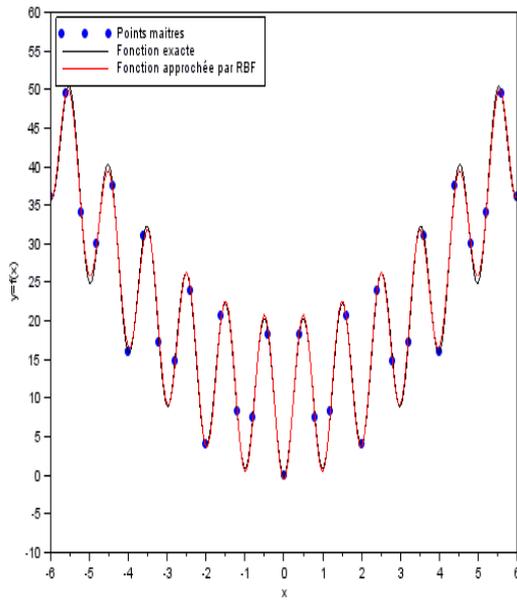


(b) Latin hypercube

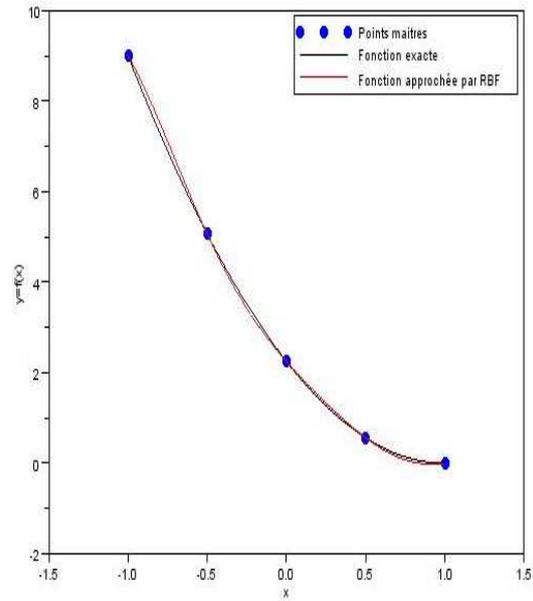
FIGURE A.10 – Approximation du critère DG avec le même nombre total ($N = 81$) de points maîtres avec les méthodes de génération de points : (a) Distribution uniforme (figure à gauches) et (b) Latin hypercube (figures à droites) -fonction exacte en noir et la fonctions approchée (a) en rouge et (b) en bleu, pour la canette de type Cas3

A.6 VALIDATION DU MÉTAMODÈLE RBF POUR DES FONCTIONS 1D

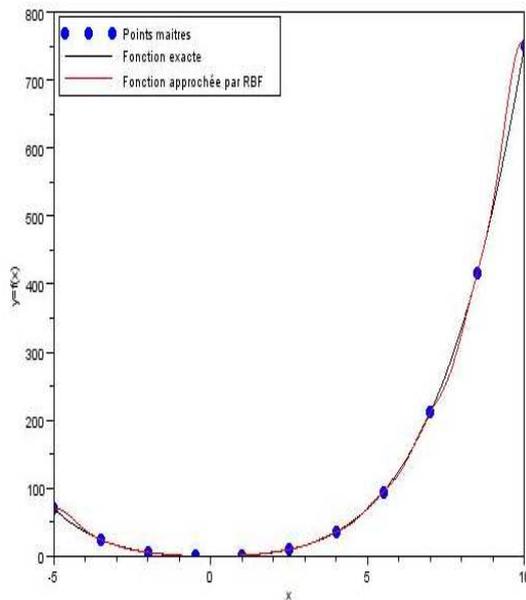
Dans cet annexe, une série des fonctions analytiques 1D est utilisée pour tester la capacité du métamodèle RBF à les approcher en présence d'un nombre de points maîtres (points bleus) répartis de manière. Les fonctions utilisées sont : Rastrigin , Perm, Zakharov et Ackbley.



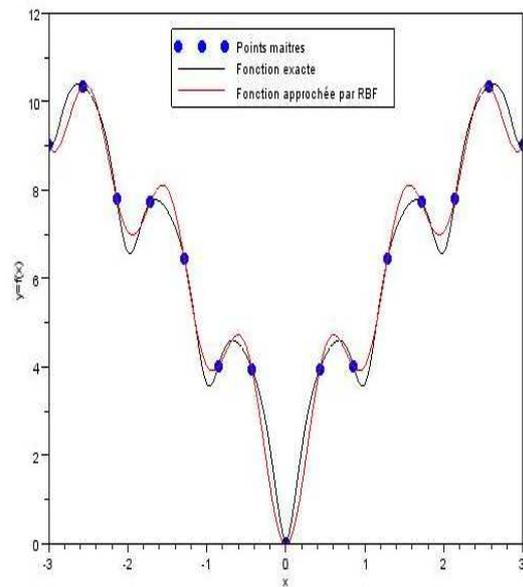
(a) Rastrigin



(b) Perm



(c) Zakharov



(d) Ackbley

FIGURE A.11 – Approximation des fonctions (a) Rastrigin , (b) Perm, (c) Zakharov et (d) Ackbley, avec le métamodèle RBF construit avec deux points maîtres répartis d'une manière uniforme (distribution uniforme).

A.7 VALIDATION DU MÉTAMODÈLE RBF POUR DES FONCTIONS 2D

Dans cet annexe, une série des fonctions analytiques 2D est utilisée pour tester la capacité du métamodèle RBF à les approcher en présence d'un nombre de points maîtres (points bleus) répartis de manière. Les fonctions utilisées sont : Matyas , Perm, Beale et Branin.

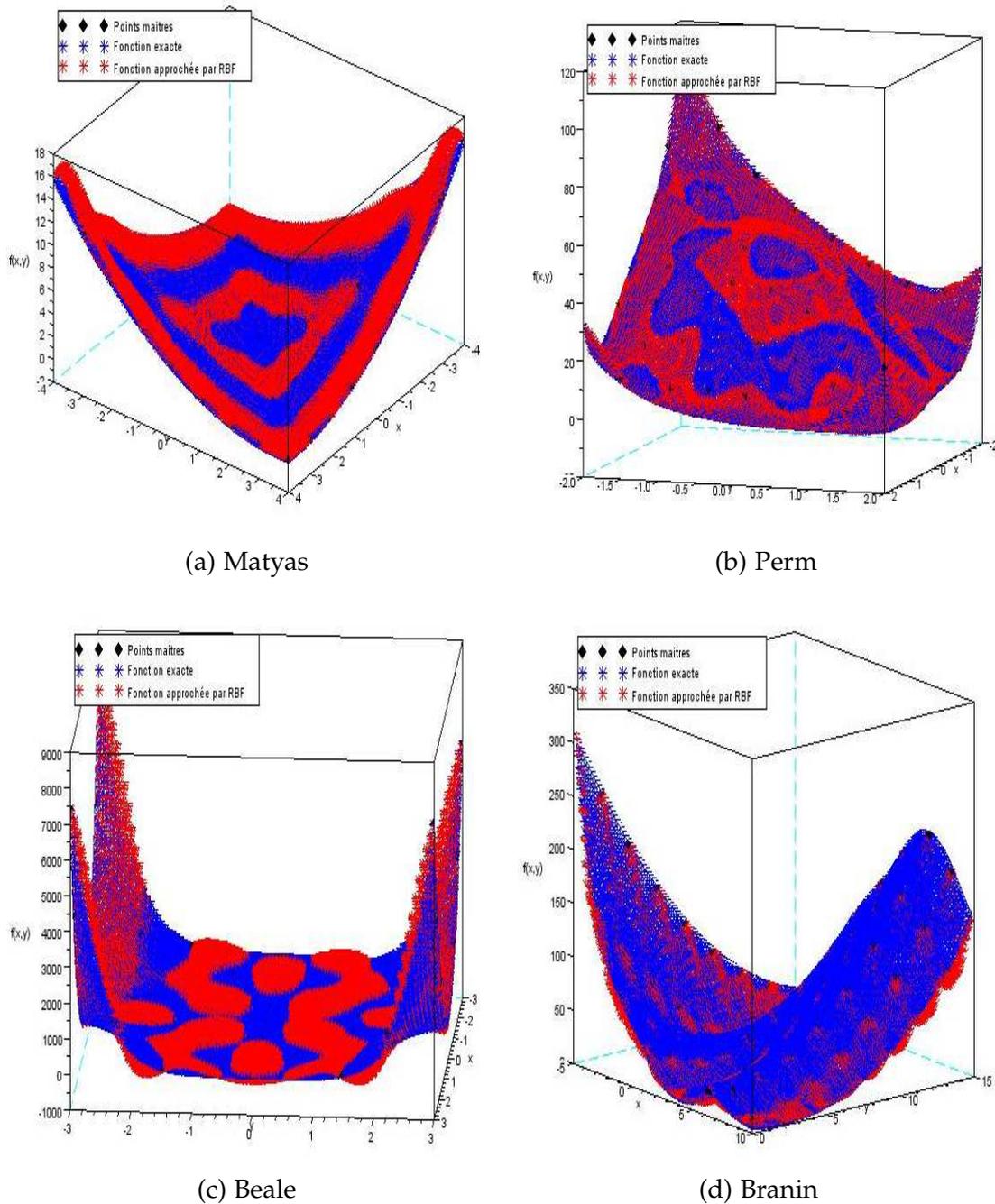


FIGURE A.12 – Approximation des fonctions (a) Matyas , (b) Perm, (c) Beale et (d) Branin, avec le métamodèle RBF construit avec deux points maîtres répartis d'une manière uniforme (distribution uniforme).

A.8 VALIDATION DE L'APPROCHE NBI RBF

Dans cet annexe, nous utilisons une série des problèmes tests multicritère qui diffèrent par leurs fronts de Pareto afin de valider notre approche de couplage NBI RBF.

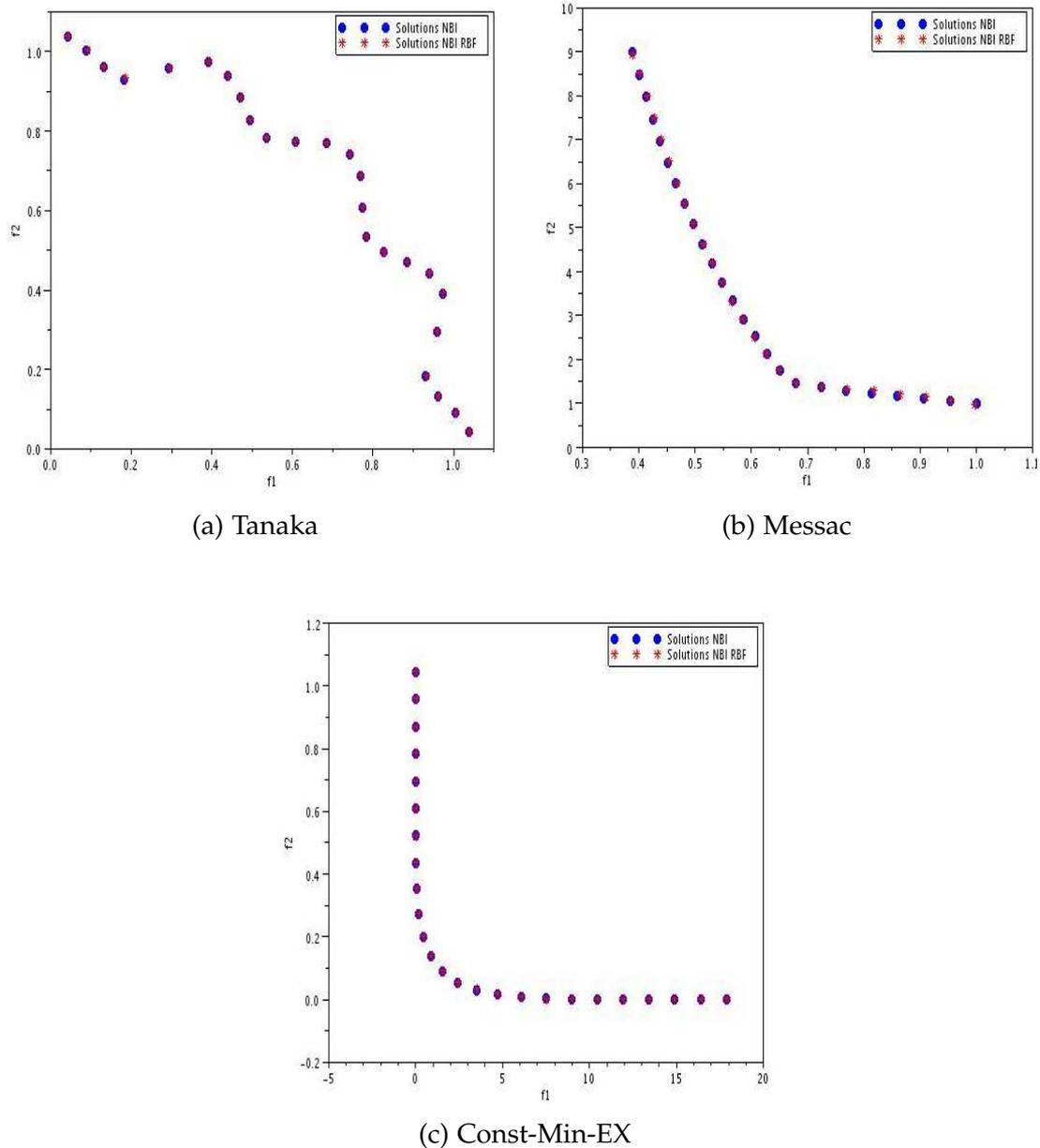


FIGURE A.13 – Comparaison entre les solutions obtenues par l'approche NBI RBF (* rouge) et celles obtenus par NBI (points bleus) pour les problèmes (a) Tanaka, (b) Messac et (c) Const-Min-Ex.

A.9 OPTIMISATION DES PROFILS DE LA CANETTE 2D

Dans cet annexe, nous faisons l'optimisation de forme pour les trois type de profil de la canette 2D, avec une nouvelle variable d'optimisation et des nouvelles contraintes. Les résultats obtenus prouvent toujours l'efficacité de notre couplage NBI RBF d'identifier des nouveaux profils respectant les exigences industriels.

* Optimisation canette -Cas1-

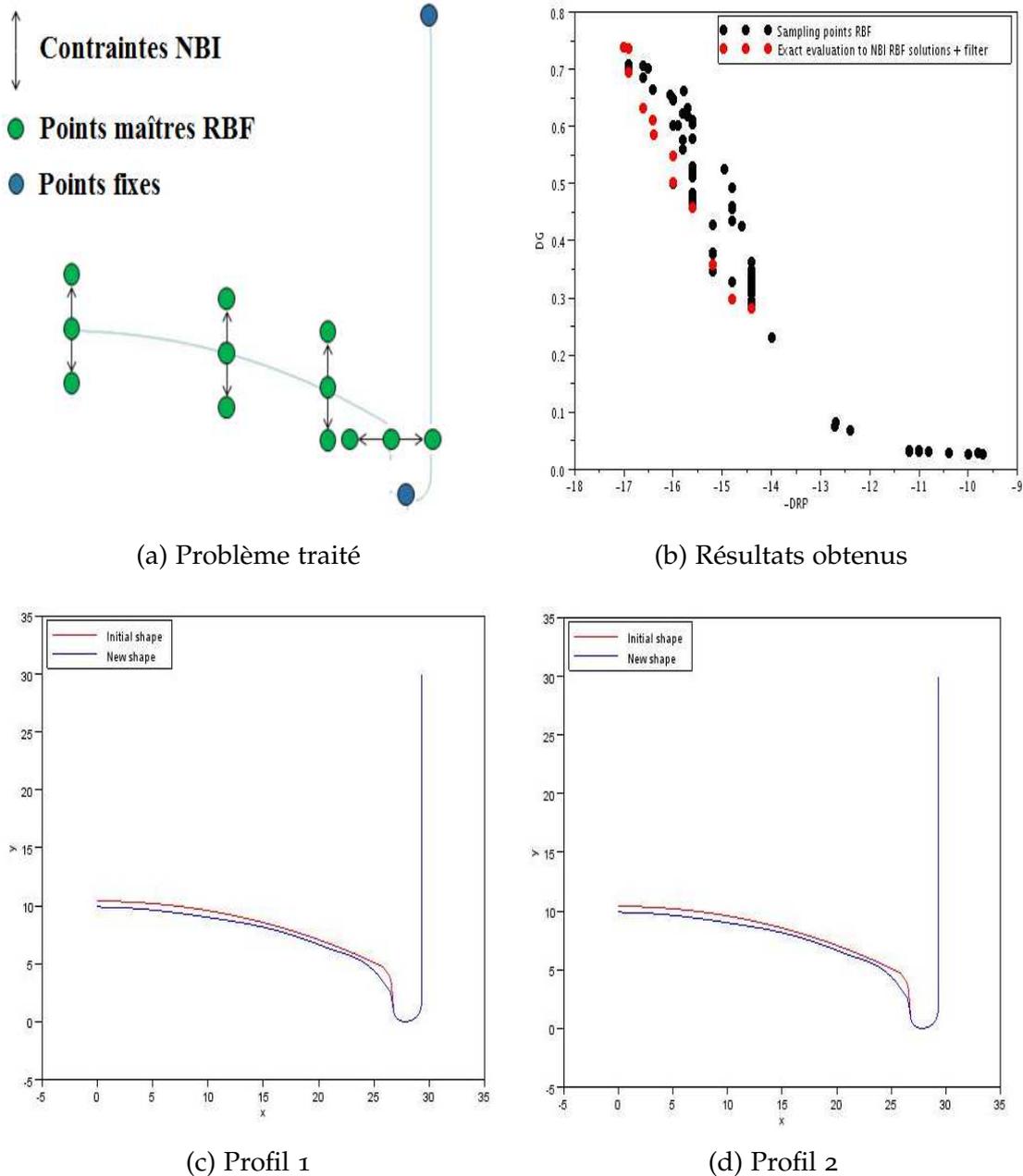


FIGURE A.14 – Optimisation de la forme du fond de la canette de type cas 1 avec l'approche NBI RBF : (a) Présentation du problème à traiter avec la variable d'optimisation et les contraintes, (b) Les résultats obtenus par l'approche NBI RBF (points rouge) et les éléments de la base RBF (points noirs), (c) et (d) des nouveaux profils pour le fond de la canette vérifiant les exigences industriels.

* Optimisation canette -Cas2-

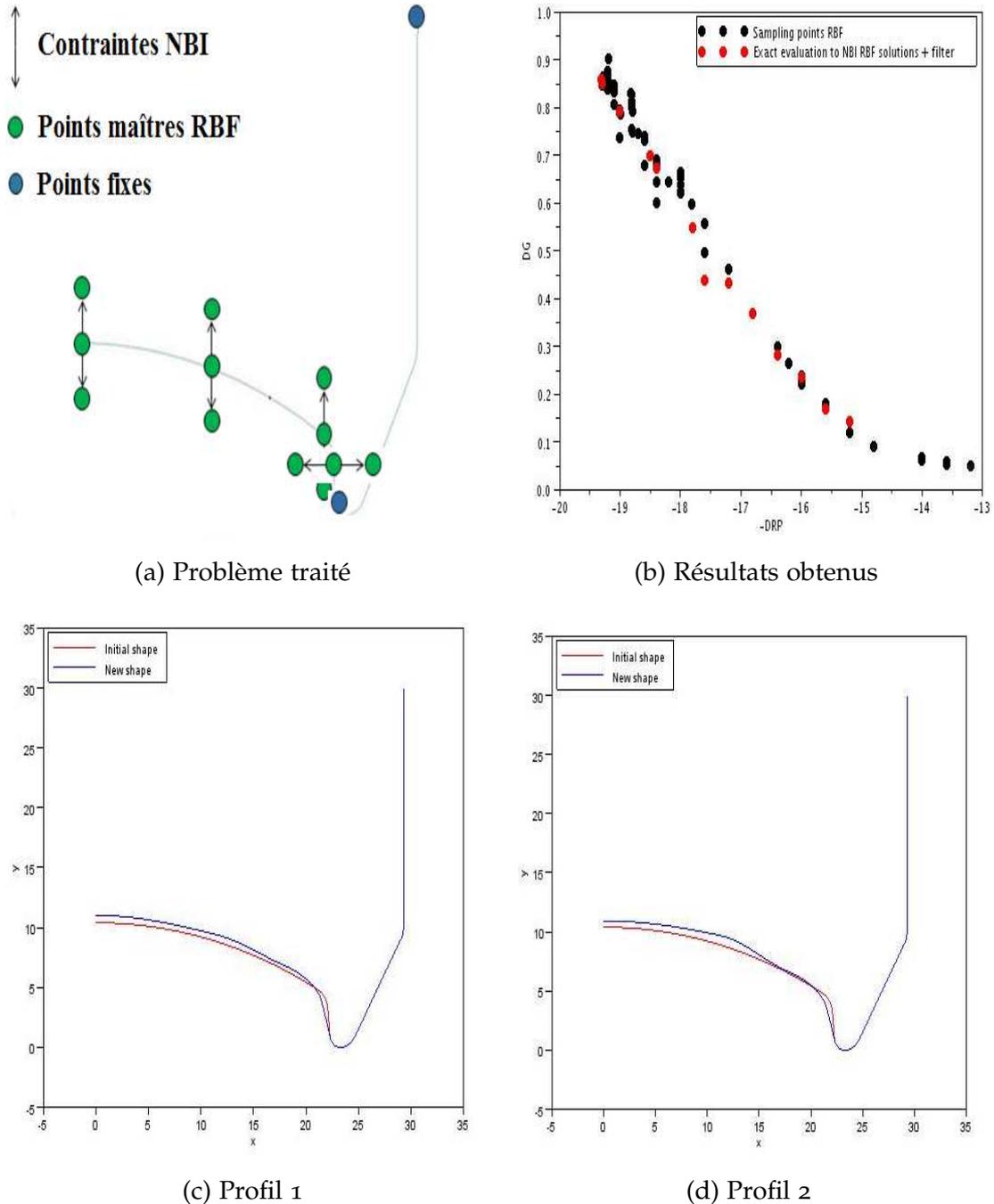
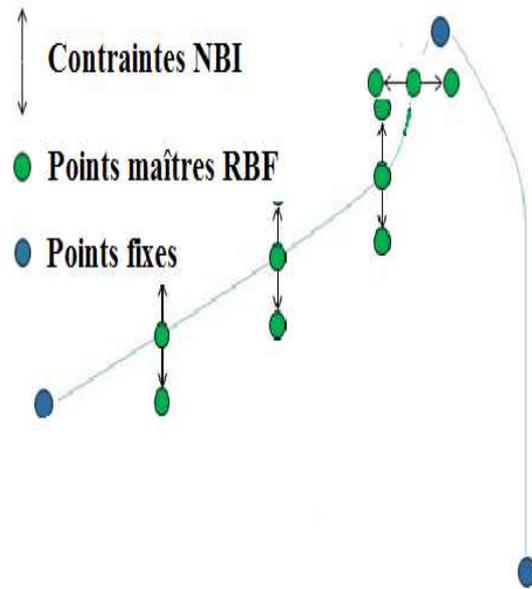
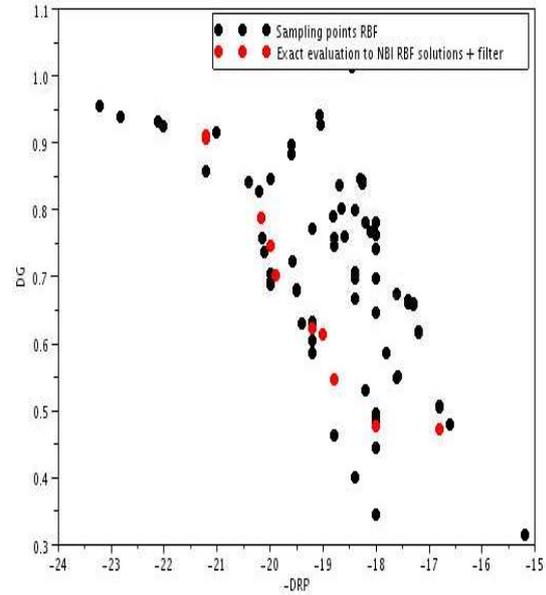


FIGURE A.15 – Optimisation de la forme du fond de la canette de type cas 2 avec l'approche NBI RBF : (a) Présentation du problème à traiter avec la variable d'optimisation et les contraintes, (b) Les résultats obtenus par l'approche NBI RBF (points rouge) et les éléments de la base RBF (points noirs), (c) et (d) des nouveaux profils pour le fond de la canette vérifiant les exigences industriels.

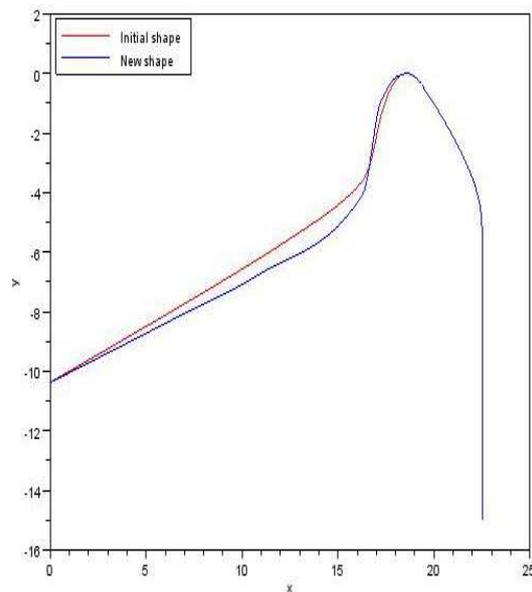
* Optimisation canette -Cas3-



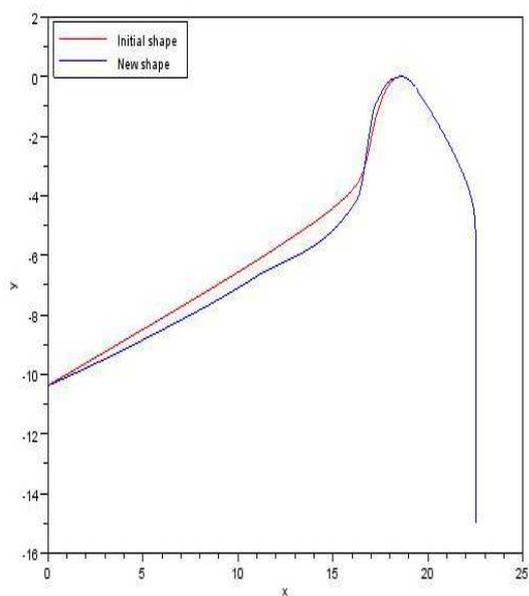
(a) Problème traité



(b) Résultats obtenus



(c) Profil 1



(d) Profil 2

FIGURE A.16 – Optimisation de la forme du fond de la canette de type cas 3 avec l'approche NBI RBF : (a) Présentation du problème à traiter avec la variable d'optimisation et les contraintes, (b) Les résultats obtenus par l'approche NBI RBF (points rouge) et les éléments de la base RBF (points noirs), (c) et (d) des nouveaux profils pour le fond de la canette vérifiant les exigences industriels.

A.10 EQUILIBRE DE NASH -PARTAGE ARBITRAIRE-

Dans cet annexe, nous utilisons une série des problèmes tests multi-critère pour calculer les équilibres de Nash avec des différents partage de la variable d'optimisation.

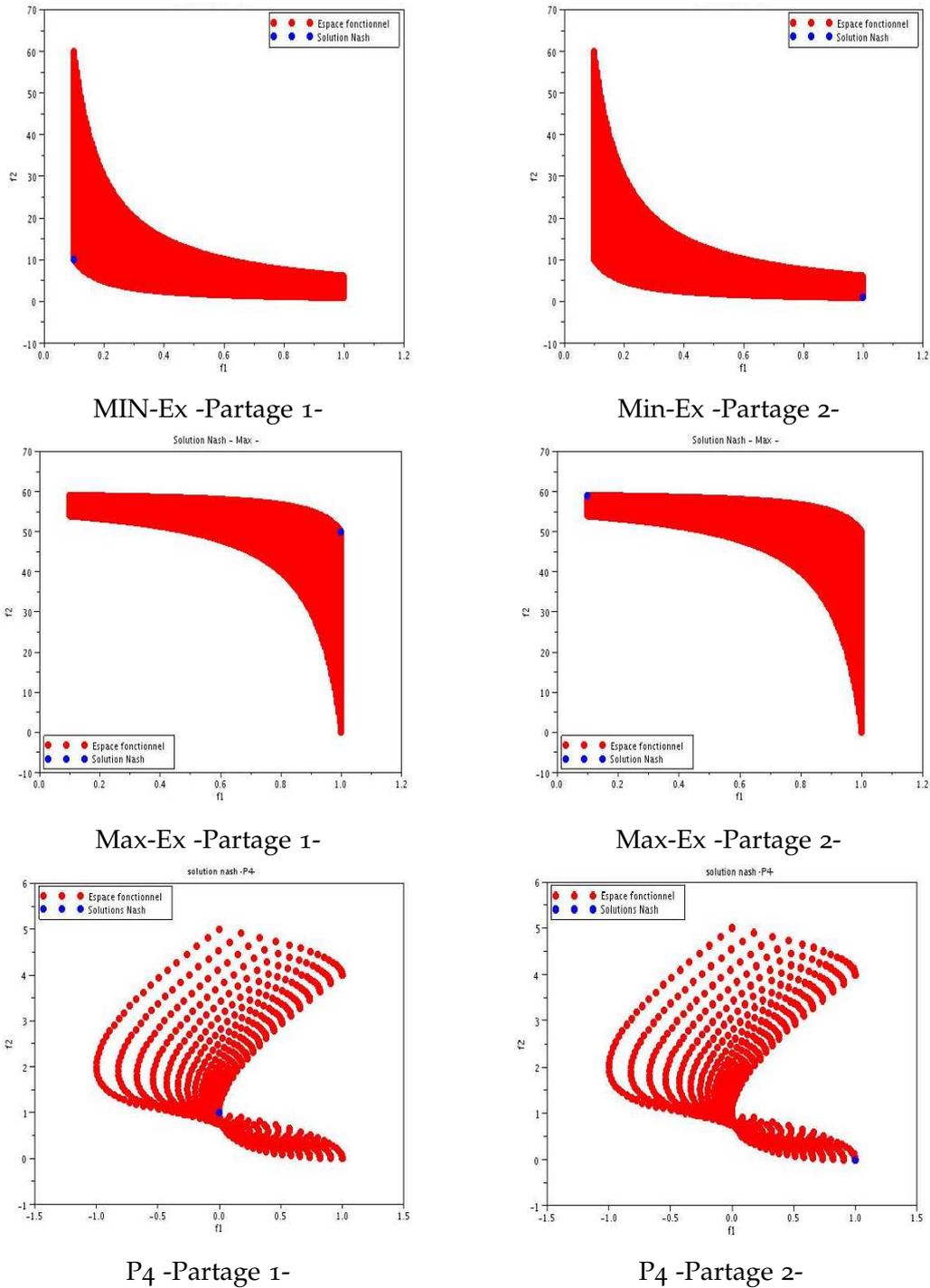
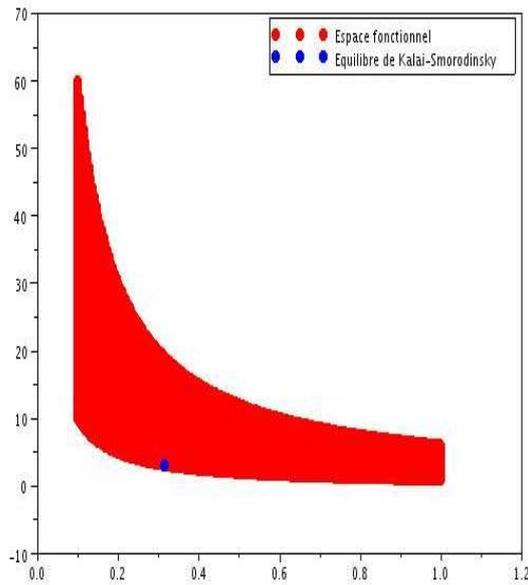


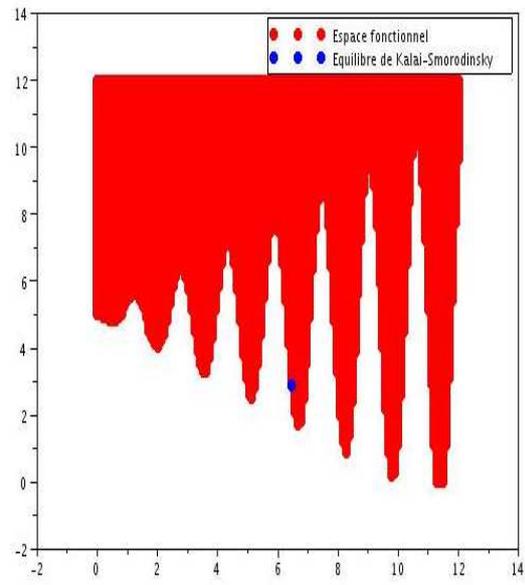
FIGURE A.17 – Equilibre de Nash pour les problèmes Min-Ex, Max-Ex et P4 pour deux différents partages arbitraires

A.11 EQUILIBRE DE KALAI-SMORODINSKY

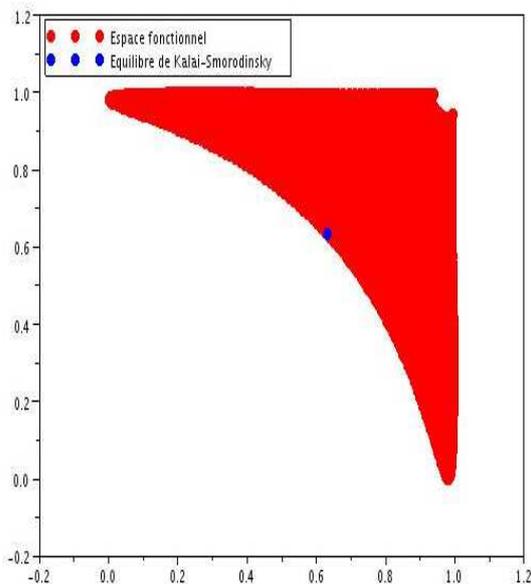
Dans cet annexe, nous utilisons une série des problèmes tests multicritère pour identifier l'équilibre de Kalai-Smorodinsky.



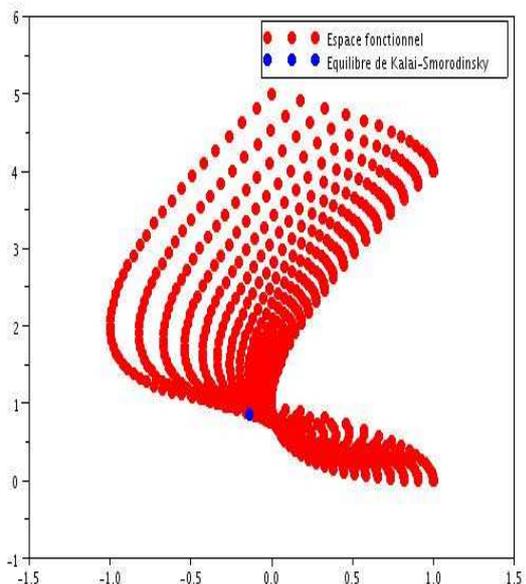
(a) Min-Ex



(b) Hanna non discontinu



(c) Fonseca



(d) P₄

FIGURE A.18 – Equilibre de Kalai-Smorodinsky (point bleu) pour les problèmes (a) Min-Ex, (b) Hanna non discontinu, (c) Fonseca et (d) P₄

BIBLIOGRAPHIE

- [Ababou.1994] R. Ababou, A. Bagtzoglou and E. Wood, *On the condition number of covariance matrices in kriging, estimation, and simulation of random fields*, *Mathematical Geology* 26, 99-133, 1994. (Cité page 33.)
- [Abou El Majd.2010] B. Abou El Majd, J-A. Désidéri, and A. Habbal, *Optimisation de forme fluide-structure par un jeu de Nash*, *Revue ARIMA*, vol. 13, pp. 3-15, 2010. (Cité page 93.)
- [Armantier.1998] O. Armantier, J.P. Florens and J.F. Richard, *Équilibre approximatif et règle intuitive : une application aux appels d'offres dans l'industrie spatiale* In : *Économie prévision*. Numéro 132-133, pp. 179-190, 1998. (Cité page 93.)
- [Balci 1.2007] S. K. Balci, P. Golland, M. Shenton and W. M. Wells, *Free-Form B-spline Deformation Model for Groupwise Registration*, 2007. (Cité page 54.)
- [Balci 2.2007] S. K. Balci, P. Golland, M. Shenton and W. M. Wells, *Non-rigid Groupwise Registration using B-Spline Deformation Model*, 2007. (Cité page 54.)
- [Bookstein.1989] F. L. Bookstein, *Principal warps : Thin-plate Splines and the decomposition of deformations*, *IEEE Transactions on pattern analysis and machine intelligence*, 1989. (Cité page 54.)
- [Bozbay.2012] I. Bozbay, H. Dietrich, Franz and Peters, *Bargaining with endogenous disagreement : The extended Kalai-Smorodinsky solution*, *Games and Economic Behavior*, 407-417, 2012. (Cité pages 2 et 100.)
- [Cachon.2004] G.P. Cachon and S. Netessine, *Game theory in supply chain analysis*. In : D. Simchi-Levi, S.D. Wu and Z.-J. Shen, editors. *Handbook of Quantitative Supply Chain Analysis : Modeling in the eBusiness Era*, Kluwer Academic Publishers, pp. 13-66, 2004. (Cité page 93.)
- [Cerny.1985] V. Cerny, *Thermodynamical approach to the traveling salesman problem : an efficient simulation algorithm*, *J. of Optimization th. and applications*, 45 :4151, 1985. (Cité page 12.)
- [Cerny.2000] C. Poloni, 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. (Cité pages 27 et 95.)

- [Choi.2000] Y. Choi and S. Lee, *Pinjectivity Conditions of 2D and 3D Uniform Cubic B-Spline Functions*, Graphical Models 62, 411427, 2000. (Cité page 54.)
- [Chung.2004] H. Chung and J. Alonso, *Multiobjective Optimization Using Approximation Model-Based Genetic Algorithms*, Multidisciplinary Analysis and Optimization Conference, 2004. (Cité page 2.)
- [Clausen.1999] J. Clausen, *Branch and Bound Algorithms - Principles and Examples*, 1999. (Cité page 11.)
- [Coelho.2007] F. Coelho, P. Breitkopf, C. Knopf-Lenoir, *Model reduction for multidisciplinary optimization Application to a 2D wing*, Structural and Multidisciplinary Optimization, 10.1007/s00158-007-0212-5, 2007. (Cité page 2.)
- [Coelho.2008] F. Coelho, P. Breitkopf, C. Knopf-Lenoir. *Bi-level Model Reduction for Coupled Problems Application to a 3D wing*, Structural and Multidisciplinary Optimization, DOI 10.1007/s00158-008-0335-3, 2008. (Cité page 2.)
- [Coello.1998] C. A. Coello Coello, *An updated survey of G.A. based multiobjective optimization techniques*, Rapport technique Lania-RD-98-08, Xalapa, Veracruz, Mexico, décembre 1998. (Cité page 15.)
- [Colorni.1991] A. Colorni, M. Dorigo and V. Maniezzo, *Distributed Optimization by Ant Colonies*, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991. (Cité page 12.)
- [Datta.2005] R. Datta and K. Deb , *An Adaptive Normalization based Constrained Handling Methodology with Hybrid Bi-Objective and Penalty Function Approach*, KanGAL Report Number 2005. (Cité page 2.)
- [Deb.1996] K. Deb and J. Dennis, *Normal boundary intersection : An alternate method for generating pareto optimal point in multicriteria problems*, ICASE, 25, 1996. (Cité pages 1 et 15.)
- [Deb.2001] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley and Sons, 2001. (Cité pages 24, 25, 65 et 67.)
- [Désidéri.2007] J-A. Désidéri, *Split of Territories in Concurrent Optimization*, Rapport de recherche INRIA , 2007. (Cité pages 99 et 110.)
- [Dorigo.1996] M. Dorigo, V. Maniezzo, and A. Colorni, *Ant system : optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics–Part B, volume 26, numéro 1, pages 29-41, 1996 (Cité page 12.)
- [Dorigo.1997] M. Dorigo and L.M. Gambardella, *Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem*, IEEE Transactions on Evolutionary Computation, volume 1, numéro 1, pages 53-66, 1997 (Cité page 12.)

- [Duvigneau.2007] R. Duvigneau and C. Praveen, *Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization*, INRIA, 26-30, 2007. (Cité pages 2 et 33.)
- [Elster.1995] C. Elster and A. Neumaier, *A grid algorithm for bound-constrained optimization of noisy functions*, IMA Journal of Numerical Analysis, pp. 585-608, 1995. (Cité page 11.)
- [Fang.2004] H. Fang, M. Rais-Rohani and Horstemeyer, *Multiobjective Crashworthiness Optimization With Radial Basis Function*, Multidisciplinary Analysis and Optimization Conference, 2004. (Cité page 2.)
- [Farina.2002] M. Farina, *A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm*, IEEE Congress on Evolutionary Computation, 956-961, 2002. (Cité page 2.)
- [Floudas.2000] C.A. Floudas, *Deterministic Global Optimization. Theory, Methods, and Applications*. Kluwer, Dordrecht, 2000. (Cité page 11.)
- [Fonseca.1997] C. M. Fonseca and P. J. Fleming, *Multiobjective optimization*, in Handbook of Evolutionary Computation (T. Bäck, D. B. Fogel, and Z. Michalewicz, eds.), ch. C4.5, pp. C4.5 :1-C4.5 :9, IOP Publishing and Oxford University Press, 1997. (Cité page 27.)
- [Ganesana.2013] T. Ganesana, P. Vasantb and I. Elamvazuthic, *Normal-boundary intersection based parametric multi-objective optimization of green sand mould system*, Journal of Manufacturing Systems, 197205, 2013. (Cité page 2.)
- [Goldberg.1994] D. Goldberg, *Algorithmes génétiques*, AddisonWesley France, 1994. (Cité page 12.)
- [Holland.1975] J. H. Holland, *Adaptation In Natural And Artificial Systems*, University of Michigan Press, 1975. (Cité page 12.)
- [Horst.1996] R. Horst and H. Tuy, *Global Optimization. Deterministic Approaches*, Springer, Berlin, 1996. (Cité page 11.)
- [Jia.2007] Z. Jia and M. G. Ierapetritou, *Generate pareto optimal solutions of scheduling problems using normal boundary intersection technique*, Computers and Chemical Engineering, 268-280, 2007. (Cité page 2.)
- [Jones.2001] R. Jones. *A Taxonomy of Global Optimization Methods Based on Response Surfaces*, Journal of Global Optimization, 345383, 2001. (Cité page 2.)
- [Kalai.1975] E. Kalai and M. Smorodinsky, *Other Solutions to Nash's Bargaining Problem*, Econometrica, 513-518, 1975. (Cité pages 2 et 100.)
- [Karakasis.2005] M. K. Karakasis and K. C. Giannakoglou, *Metamodel-assisted multi-objective evolutionary optimization*, Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Munich, 2005.
- [Kirkpatrick.1983] S. Kirkpatrick, C. Gelatt and M. P. Vecchi, *Optimization by simulated annealing*. Science, 671680, 1983. (Cité page 12.)

- [Krige.1951] D. G. Krige, *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. (Cité page 33.)
- [Lassila.2010] T. Lassila, G. Rozza. *Parametric free-form shape design with PDE models and reduced basis method*, Comput. Methods Appl. Mech. Engrg., 1583-1592, 2010. (Cité page 56.)
- [Lassila.2011] T. Lassila, A. Manzoni and G. Rozza *Geometrical and computational reduction strategies for the approximation and optimization of viscous flows*, **Workshop on Reduced basis in high dimension**, Université Pierre et Marie Curie, Paris VI, France, June 23-24, 2011. (Cité page 56.)
- [Leng.2005] M. Leng and M. Parlar , *Game theoretic applications in supply chain management : a review*. Information Systems and Operations Research, 43(3), pp. 187-230, , 2005. (Cité page 93.)
- [Leyton-Brown.2008] Leyton-Brown, Kevin, Shoham and Yoav, *Essentials of Game Theory : A Concise, Multidisciplinary Introduction*, The National Academy of Scienc, San Rafael, CA : Morgan and Claypool, 2008. (Cité page 2.)
- [Li.1987] S. Li and T. Basar , *Distributed algorithms for the computation of noncooperative equilibria*, Automatica, pp 523-533, 1987. (Cité page 94.)
- [Lian.2004] Y. Lian and M. Liou, *Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithms*, Multidisciplinary Analysis and Optimization Conference, 2004. (Cité page 2.)
- [Lim.2001] Y. Lim and X. Joulia , *Efficient Implementation of the Normal Boundary Intersection (NBI) Method on Multiobjective Optimization Problems*, Industrial and Engineering Chemistry Research, 648655, 2001. (Cité page 2.)
- [Logist.2012] F. Logist and J. V. Impe , *Novel insights for multi-objective optimisation in engineering using Normal Boundary Intersection and (Enhanced) Normalised Normal Constraint*, Structural and Multidisciplinary Optimization, 417-431, 2012. (Cité page 1.)
- [Ls dyna manual.2012] *LS-DYNA KEYWORD USER'S MANUAL VOLUME I*, page 111-112, 2012. (Cité page 48.)
- [Manzoni.2010] A. Manzoni, A. Quarteroni, G. Rozza. *Shape optimization of cardiovascular geometries by reduced basis methods and free-form deformation techniques*, submitted to Int. J. Num. Meth. Fluids, 2010. (Cité page 56.)
- [Martinez.2007] M. Martinez, J. Sanchez, X. Blasco and J. V. Salcedo, *A new perspective on multiobjective optimization by enhanced normalized normal constraint method*, Structural and Multidisciplinary Optimization, 537-546, 2007. (Cité pages 1 et 15.)

- [Matheron.1962] G. Matheron, *Traité de géostatistique appliquée*, 1962. (Cité page 33.)
- [Maystre.1994] L.-Y. Maystre, J. Pictet and J. Simos, *Méthodes multicritères ELECTRE*, édition Presses Polytechniques et Universitaires Romandes, 1994. (Cité page 15.)
- [Messac.2002] A. Messac, A. A. Mullur and C. A. Mattson , *Minimal Representation of Multiobjective Design Space using a Smart Pareto Filter*, Proceedings of 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-2002-5458, Atlanta, GA, September 2002. (Cité page 29.)
- [Messac.2003] A. Messac, A. Ismail-Yahaya and C. A. Mattson , *The Normalized Normal Constraint Method for Generating the Pareto Frontier*, Structural and Multidisciplinary Optimization, 8698, 2003. (Cité page 29.)
- [Messac.2004] A. Messac and C. A. Mattson , *Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier*, Structural Dynamics, and Materials Conference, Palm Springs, California, April 19-22, 2004. (Cité pages 1 et 15.)
- [Miettinen.1999] K. M. Miettinen, *Nonlinear multiobjective optimization*, éditions Kluwer academic publisher, 1999. (Cité page 15.)
- [Moussaid.2010] N. Moussaid, *Optimisation multidisciplinaire par partage mixte de variable Applications au traitement d'images et à l'optimisation topologique*. Thèse EMI, pp. 31-54, 2010. (Cité pages 99 et 110.)
- [Nagahisa.2002] R. I. Nagahisa and M. Tanaka , *An axiomatization of the Kalai-Smorodinsky solution when the feasible sets can be finite*, Social Choice and Welfare, 751-761, 2002. (Cité pages 2 et 100.)
- [Nash.1950] J. Nash, *Equilibrium points in n -person games*, The National Academy of Scienc, 48-49, 1950. (Cité page 2.)
- [Raza.2008] W. Raza and K. Y. Kim, *Shape optimization of wire-wrapped fuel assembly using Kriging metamodeling technique*, Nuclear Engineering and Design 238, 13321341, 2008. (Cité page 2.)
- [Rippa.1999] S. Rippa, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comp. Math, 11, 1999. (Cité pages 2 et 36.)
- [Sait.1999] S. M. Sait and H. Youssef, *Iterative Computer Algorithms with Applications in Engineering : Solving Combinatorial Optimization Problems*, IEEE Computer Society, 1999. (Cité page 15.)
- [Sakawa.1988] M. Sakawa and H. Yano, *An interactive fuzzy satisficing method for multiobjective linear programming problems with fuzzy parameters*, Fuzzy Sets and Systems, volume 28, 129-144, 1988. (Cité page 15.)
- [Shackleford.2012] J.A. Shackleford, Q. Yang, A. M. Lourenço, N. Shusharina, N. Kandasamy, and G. C. Sharp, *Analytic Regularization of Uniform Cubic B-spline Deformation Fields*, Medical Image Computing and

- Computer-Assisted Intervention MICCAI 2012 Lecture Notes in Computer Science Volume 7511,122-129 , 2012. (Cité page 54.)
- [Shukla.2007] P. K. , *On the Normal Boundary Intersection Method for Generation of Efficient Front*, Computational Science ICCS 2007, 310-317, 2007. (Cité pages 1 et 15.)
- [Siddiqui.2012] S. Siddiqui, S. Azarm and S. A. Gabriel , *On improving normal boundary intersection method for generation of Pareto frontier*, Structural and Multidisciplinary Optimization, 839-852, 2012. (Cité page 1.)
- [Sobester.2005] A. Sobester, 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, 3159, 2005. (Cité page 2.)
- [Stryk.1992] O. V. Stryk and R. Bulirsch ,*Direct and indirect methods for trajectory optimization*, Annals of Operations Research, Volume 37, Issue 1, pp 357-373, 1992. (Cité page 11.)
- [Tahk.2003] M. J. Tahk, Y. S. Hong and H. Lee, *Acceleration of the Convergence Speed of Evolutionary Algorithms Using Multilayer Neural Networks*, Eng.Optimiz, 91-102, 2003. (Cité page 2.)
- [Tanaka.1995] M. Tanaka, 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. (Cité pages 26 et 95.)
- [Törn.1989] A. Törn and A. Zilinskas, *Global Optimization*. Lecture Notes in Computer Science, N° 350, Springer-Verlag, Berlin, 1989. (Cité page 38.)
- [Wang.2007] G. G. Wang and S. Shan, *Review of Metamodelling Techniques in Support of Engineering Design Optimization*, ASME J. Mech. Des., 129, 370-380, 2007. (Cité page 2.)
- [Wu.2002] C. Y. Wu, 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, 7987, 2002. (Cité page 2.)
- [Yann.2002] C. Yann and P. Siarry, *Optimisation multiobjectif*, Eyrolles, isbn = 2-212-11168-1, 2002. (Cité page 15.)

NOTATIONS

DG : Dome Growth /Déroutement du Pied

DRP : Dome Reversal Pressure/Pression de Retournement du Fond

DU : Distribution Uniforme

EN : Equilibre de Nash

FFD : Free Form Deformation

KS : Equilibre Kalai-Smorodinsky

LHS : Latin Hypercub Sampling

MAX : Le facteur Maximum Absolute Error

NBI : La méthode Normal Boundary Intersection

NBI RBF : Couplage de la méthode NBI et le métamodèle RBF

NMAX : Le facteur Normalized Maximum Absolute Error

NNCM : La méthode Normalized Normal constraint Methods

NNCM RBF : Couplage de la méthode NNCM et le métamodèle RBF

NRMSD : Le facteur Normalized Root Mean Square Deviation

RBF : Le métamodèle Radial Basis Function

RMSD : Le facteur Root Mean Square Deviation

