HAL
open science

Représentation de solution en optimisation continue, multi-objectif et applications

Hafid Zidani

To cite this version:

Hafid Zidani. Représentation de solution en optimisation continue, multi-objectif et applications. Mécanique des structures [physics.class-ph]. INSA de Rouen; Université Mohammed V-Agdal (Rabat, Maroc ; 1993-2014), 2013. Français. NNT : 2013ISAM0027 . tel-00939980

HAL Id: tel-00939980

https://theses.hal.science/tel-00939980v1

Submitted on 31 Jan 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.

Université Mohammed V - Agdal
École Mohammadia d'Ingénieurs
Laboratoire d'Étude et Recherche en
Mathématiques Appliquées.
LERMA. Rabat - Maroc

Institut National des Sciences
Appliquées de Rouen
Laboratoire d'Optimisation et Fiabilité
en Mécanique des Structures. Rouen - France
ED : Sciences Physiques, Mathématiques
et de l'Information pour l'Ingénieur

THÈSE DE DOCTORAT EN COTUTELLE

présentée à

L'ECOLE MOHAMMADIA D'INGENIEURS

pour obtenir le grade de

DOCTEUR EN SCIENCES APPLIQUEES DE L'ÉCOLE MOHAMMADIA D'INGENIEURS

Spécialité : Mathématiques appliquées et Informatique.
Option : Optimisation, Analyse Numérique, Statistique.

et le grade de

DOCTEUR DE L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN

Spécialité : Optimisation des structures.
Option : Fiabilité, Structures, Qualité.

présentée par

M. Hafid ZIDANI

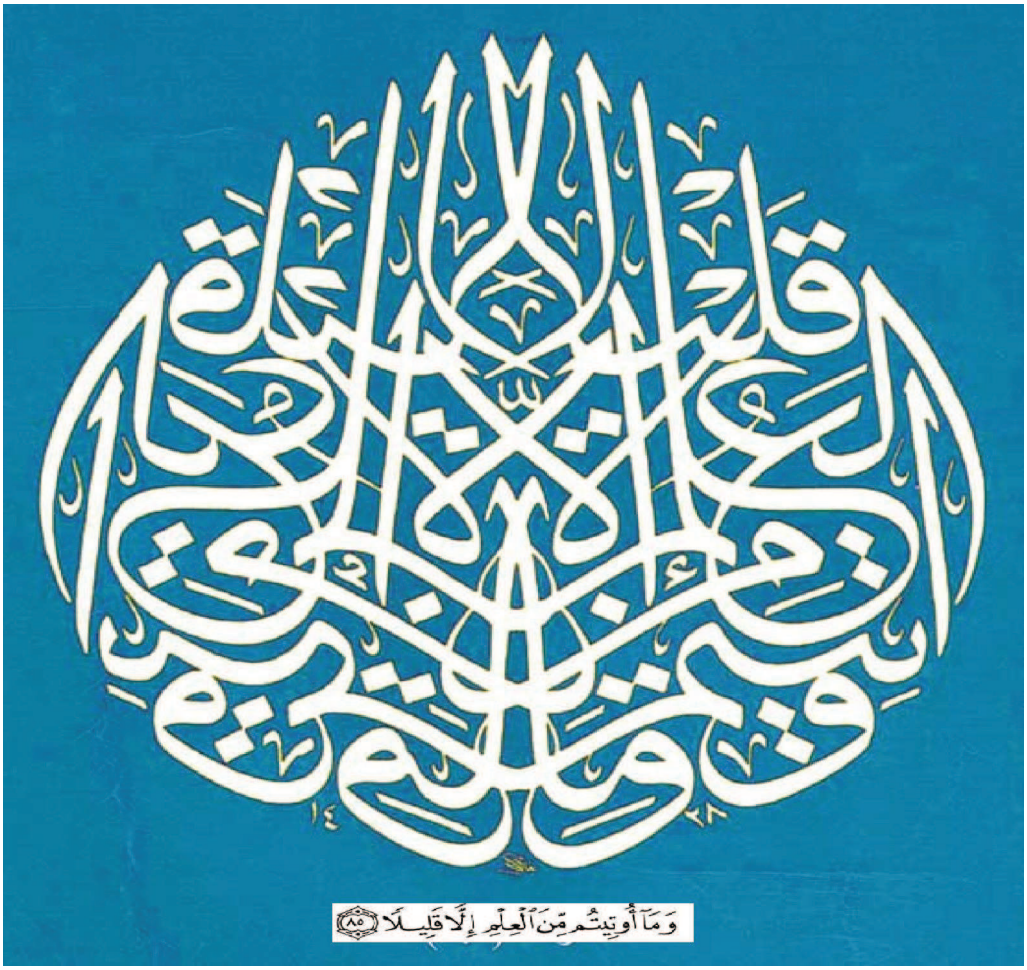
Sujet de la thése :

REPRESENTATION DE SOLUTION EN OPTIMISATION CONTINUE, MULTI-OBJECTIF ET APPLICATIONS

Soutenue publiquement le 26 Octobre 2013 devant le jury composé de :

Mme. Najiba SBIHI	Professeur, EMI, Rabat	Présidente
Mme. Souad EL BERNOUSSI	Professeur, Fac des sciences de Rabat	Rapporteur
M. Piotr BREITKOPF	Professeur, UTC, Compiègne	Rapporteur
M. Mohamed TKIOUAT	Professeur, EMI, Rabat	Rapporteur
M. Abdelkhalak EL HAMI	Professeur, INSA de Rouen	Examinateur
M. Rachid ELLAIA	Professeur, EMI Rabat	Directeur de thèse
M. José Eduardo SOUZA DE CURSI	Professeur, INSA de Rouen	Directeur de thèse

بسم الله الرحمن الرحيم



"Et on ne vous a donné que peu de connaissance." Sourate AL-ISRA (Le voyage Nocturne) versets 85.

"And of knowledge, you (mankind) have been given only a little." Sourate AL-ISRA (The Journey by Night).

A mes parents,

A ma femme Zineb,

A mon frère Rachid,

A Hafsa et Rachid.

REMERCIEMENTS

le prophète Mohammed(saw)(570 - 632) a dit : (Celui qui ne remercie pas les gens, ne remercie pas Allah) rapporté par Ahmed.

La thèse rapportée dans ce mémoire est fruit d'une collaboration scientifique entre le Laboratoire d'Étude et Recherche en Mathématiques Appliquées(LERMA), de l'École Mohammadia d'Ingénieurs, et le Laboratoire d'Optimisation et Fiabilité en Mécanique des Structures (LOFIMS) de l'Institut National de Sciences Appliquées de Rouen. Ce travail n'aurait jamais vu le jour sans le soutien et les encouragements de mon directeur de thèse Prof. ELLAI Rachid. Je tiens à lui exprimer ma profonde reconnaissance pour ses précieux conseils, la qualité de son encadrement, son soutien constant et sa lucidité scientifique tout au long de ces années. Je suis profondément reconnaissant envers Prof. José Eduardo SOUZA DE CURSI, mon directeur de thèse, qui a mis en place les conditions propices au bon développement de mes travaux et a su prodiguer des réflexions inestimables. Ses conseils en matière de recherche ont été toujours précieux. Je suis heureux de pouvoir exprimer ici mes remerciements à chacun des membres du jury qui ont bien accepté d'évaluer ce travail. Je remercie donc tout d'abord Prof. Najiba SBIHI, professeur à l'École Mohammedia d'Ingénieurs, d'avoir accepté de présider le jury de thèse. Ensuite, j'exprime toute ma reconnaissance au Prof. Souad EL BERNOUSSI, professeur à la faculté des sciences de Rabat, Prof. Piotr BREITKOPF, professeur à l'Université Technologique de Compiègne, et Prof. Mohamed TKIOUAT, professeur à l'École Mohammedia d'Ingénieurs, pour l'intérêt qu'ils ont porté à mes travaux et pour avoir accepté la charge d'en être les rapporteurs. Je remercie Prof. Abdelkhalak EL HAMI , professeur à l'Institut Nationale des Sciences Appliquées de Rouen, pour l'honneur qu'il m'a fait d'accepter d'être examinateur de mes travaux. Je tiens aussi à remercier Prof. Rubens Sampaio et D. Emmanuel PAGNACCO pour leur travail collaboratif et leur participation fructueuse dans l'élaboration des applications mécaniques. Remerciements et profonde gratitude vont également à Mme. Rajae ABOULAICH, professeur à l'École Mohammedia d'Ingénieurs (ÉMI) et responsable du laboratoire LERMA, pour son soutien morale et scientifique et pour l'ambiance chaleureuse et agréable qu'elle a toujours maintenue au sein du laboratoire.

Chacun des membres des laboratoires LERMA et LOMIFS mérite ma reconnaissance pour avoir, d'une manière ou d'une autre contribué à ce que la période des travaux soit agréable et enrichissante. Merci à vous tous.

Un immense MERCI, tout à fait particulier, à ma femme Zineb MOUATADIR, je suis conscient que cette thèse ne serait pas ce qu'elle est sans sa patience et soutien.

Résumé. Cette thèse a pour objectif principal le développement de nouveaux algorithmes pour la résolution de problèmes d'optimisation mono et multi-objectif, en se basant sur des formules de représentation ayant la tâche principale de générer des points initiaux appartenant à une zone proche du minimum globale.

Dans ce contexte, une nouvelle approche fondée sur une formule de représentation pour résoudre des problèmes d'optimisation globale a été adoptée. Des expérimentations ont été réalisées pour l'optimisation de fonctions non linéaires, non différentiables et multimodaux utilisant cette formule de représentation, hybridée avec l'algorithme génétique et l'algorithme de Nelder Mead. Des expérimentations approfondies concernant l'effet de différents paramètres ont été réalisées. Des comparaisons avec d'autres algorithmes proposés dans la littérature ont montré l'efficience de notre nouveau algorithme (RFNM). D'autre part, une extension à la dimension infinie a été établie en proposant une démarche pour la recherche du minimum global d'une fonctionnelle sur un sous-ensemble fermé, borné de l'espace de Hilbert séparable. La représentation suggère une méthode numérique simple, ce qui présente l'originalité de n'introduire ni une approximation en dimension finie à priori ni la résolution d'équations algébriques linéaires ou non linéaires. Cette méthode a été testée sur quelques problèmes classiques du calcul des variations, en utilisant différentes stratégies et bases et a conduit à des points de départ convenables pour une amélioration ultérieure par l'optimisation avec des méthodes standards.

Par ailleurs, une méthodologie pour transformer un problème linéaire avec charge gaussienne en un problème multi-objectif a été appliquée à deux problèmes de conception mécanique.

D'autres part, un dispositif de suspension modélisé par un système dynamique linéaire SDOF impliquant une rigidité aléatoire uniforme a été considéré. La conception du raideur est réalisée en respectant une coque de réponse d'amplitude maximale, compte tenu de la probabilité de défaillance. Pour générer le front de Pareto, pour ce problème convexe on a amélioré la méthode NNC pour obtenir la courbe de Pareto en utilisant notre algorithme hybride RFNM.

Enfin, une nouvelle méthode d'optimisation multi-objectif dite RSMO est présentée. Elle permet de résoudre les problèmes d'optimisation multi-objectif en générant un nombre suffisant de points du front de Pareto. Une implémentation de cette méthode, pour des problèmes convexes, a été proposée. Des tests sur quelques problèmes de la littérature ont été réalisés, et ont démontré l'efficacité et l'efficience de cette approche. Des comparaisons avec des méthodes bien connues dans le domaine de l'optimisation multi-objectif ont montré que RSMO est plus efficace et plus efficiente que les autres approches.
Mots clés. Optimisation globale, optimisation multi-objectif, représentation de solution, Nelder Mead, algorithme hybride, algorithme génétique, optimisation par essaim de particules, optimisation des structures, calculs des variations, dynamique des structures, optimisation fiabiliste.

Abstract. The main objective of this work is to develop new global algorithms to solve single and multi-objective optimization problems, based on the representation formulas with the main task to generate initial points belonging to an area close to the global minimum.

In this context, a new approach based on a representation formula for solving global optimization problems has been adopted. Experiments were performed for optimizing nonlinear, non-differentiable and multimodal problems using this representation formula, hybridized with the genetic and the Nelder Mead algorithms. Extensive research on the effect of different parameters were performed. Comparisons with other algorithms proposed in the literature have shown the efficiency of RFNM.

On the other hand, an extension to the infinite dimension was established by proposing an approach for finding the global minimum of a functional of a closed subset, bounded on the Hilbert separable space. The representation suggests a numerical method

based on the generation of samples in order to estimate the means. Three strategies for the implementation are examined, with the originality that they do not involve a priori finite dimensional approximation of the solution and consider a Hilbertian basis or enumerable dense family. Numerical examples involving classical situation issued from calculus of variations are presented and establish that the method is effective to calculate.

Moreover, a methodology to transform a linear problem with Gaussian charge to a multi-objective problem has been applied to two problems of mechanical design. To solve the resulting problem, the RFNM algorithm optimization problem was proposed in collaboration with the NBI method. The two-bar truss problem has an interesting feature : Depending on the frequency of the load, the Pareto curve takes three different forms (one point, a convex and continuous curve and finally a discontinuous Pareto curve). For both problems RFNM is more accurate, reliable and efficient to locate the global optimum than the other analyzed alternatives.

On the other hand, a suspension idealized as a spring-mass-damper system is considered. The amplitude of a nominal system is constrained to satisfy certain limitations in a given frequency band and the design is to be done as a reliability-based optimization. The stiffness is characterized by a uniform random variable, and its mean and standard deviation are the optimization parameters. To generate the Pareto front, the Normal Boundary Intersection (NBI) method is used with the RFNM algorithm. Results show that a not-connected Pareto curve can be obtained for some choice of constraint. Hence, in this simple example, one shows that difficult situations can occur in the design of dynamic systems when prescribing an amplitude response hull.

Finally, a new multi-objective optimization method called RSMO is presented. It solves the multi-objective optimization problems by generating a sufficient number of points in the Pareto front. An implementation of this method for convex problems was proposed. Tests on some problems of literature are made, and demonstrated the effectiveness and efficiency of this approach. Comparisons with well-known multi-objective optimization methods have shown that RSMO is more effective and more efficient than the other approaches.

Keywords. Global optimisation, Multi-objectif optimisation, representation of solution, Nelder Mead, hybrid algorithms, genetic algorithm, Particle swarm optimisation, variational calculus, Structural dynamics, Mechanical design, Random systems, Uncertainty, Multi-objective optimisation, Reliability-based design optimisation.

TABLE DES MATIÈRES

TABLE DES MATIÈRES vi

LISTE DES FIGURES x

LISTE DES TABLEAUX xiii

CONTEXTE, OBJECTIF ET ORGANISATION DE LA THÈSE 2

1 ÉTAT DE L'ART DES MÉTHODES D'OPTIMISATION MONO-OBJECTIF ET MULTI-OBJECTIF 5
1.1 OPTIMISATION MONO-OBJECTIF 6
1.1.1 Introduction 6
1.1.2 Formulation 7
1.1.3 Méthodes énumeratives 7
1.1.4 Méthodes déterministes 8
1.1.4.1 Les méthodes de gradient 8
1.1.4.2 La méthode de Newton 8
1.1.4.3 La méthode multistart 9
1.1.4.4 La méthode de Nelder Mead 9
1.1.5 Méthodes stochastiques 11
1.1.5.1 Les algorithmes génétiques 12
1.1.5.2 La méthode du recuit simulé 13
1.1.5.3 La méthode de Recherche Tabou (RT) 14
1.1.5.4 Algorithme de colonie de fourmis 15
1.1.5.5 Algorithme d'essaim de particules 15
1.1.6 Optimisation avec Contraintes 17
1.2 OPTIMISATION MULTI-OBJECTIF 17
1.2.1 Introduction 17
1.2.2 Concepts de base et terminologie 18
1.2.2.1 Définition du POM 18
1.2.2.2 Relations 18
1.2.2.3 Domination de Pareto 20
1.2.2.4 Optimalité de Pareto 21
1.2.2.5 Ensemble optimal local et global 21
1.2.3 Classification des méthodes 21
1.2.4 Méthodes Agrégées 22
1.2.4.1 Méthodes d'Agrégation 22
1.2.4.2 Méthode ε-contrainte 23
1.2.4.3 Programmation par but 24
1.2.4.4 Discussion sur les méthodes d'agrégation 25
1.2.5 Méthodes Non Agrégées et Non Pareto 25
1.2.5.1 Méthode VEGA 25
1.2.5.2 La Méthode lexicographique 26

1.2.6 Méthodes Pareto 26
1.2.6.1 Méthode MOGA(Multiple Objective Genetic Algorithm) 27
1.2.6.2 Méthode NPGA(Niched Pareto Genetic Algorithm) . 27
1.2.6.3 Méthode SPEA(Strength Pareto Evolutionary Algorithm) 28
1.2.6.4 Méthode PAES(Pareto Archived Evolution Strategy) . 29
1.2.6.5 Méthode NSGA(Non dominated Sorting Genetic Algorithm) 30
1.2.6.6 Méthode NBI(Normal Boundary Intersection) 31
1.2.6.7 Méthode NNC(Normalized Normal Constraint) . . . 34
1.3 CONCLUSION 35

2 REPRESENTATION DE SOLUTION : CONTRIBUTION À L'OPTIMISATION GLOBALE **36**
2.1 INTRODUCTION 37
2.2 FORMULE DE REPRÉSENTATION 38
2.2.1 Introduction 38
2.2.2 Hypothèses et notations 39
2.2.3 Le résultat central 40
2.2.4 Choix de la fonction 41
2.3 HYBRIDATION DE LA MÉTHODE DU SIMPLEXE AVEC LA FORMULE DE REPRÉSENTATION ET L'ALGORITHME GÉNÉTIQUE . . 43
2.3.1 Introduction 43
2.3.2 La Formule de Representation 44
2.3.3 Algorithme : 45
2.4 FONCTIONS TEST 45
2.5 RÉSULTATS NUMÉRIQUES 48
2.5.1 Influence du choix de la distribution de probabilité 49
2.5.2 Influence de la fonction g 49
2.5.3 Influence de la taille de la population pour GA 50
2.5.4 Influence de la taille de l'échantillon pour RF 50
2.5.5 Comparison entre méthodes 50
2.5.6 Comparison avec d'autres algorithmes 52
2.6 CONCLUSION 53

3 APPLICATIONS : PROBLÈMES DU PILIER DE PONT ET D'UNE STRUCTURE EN TREILLIS **56**
3.1 INTRODUCTION 57
3.2 CONCEPTION D'UN PILIER 57
3.3 CONCEPTION DE LA STRUCTURE DE TREILLIS 61
3.4 EFFICACITÉ ET EFFICIENCE DE RFNM 65
3.4.1 Influence de la fonction de Pincus 65
3.4.2 Comparaison avec GA, PSO et NSGA II 66
3.5 CONCLUSION 68

4 NNC AMÉLIORÉ POUR LA CONCEPTION D'UN DISPOSITIF FIABLE DE SUSPENSION DYNAMIQUE AVEC RIGIDITÉ DYNAMIQUE ALÉATOIRE ET UNIFORME **69**
4.1 INTRODUCTION 70
4.2 GÉNÉRALITÉS SUR LA CONCEPTION MÉCANIQUE LORS DE L'EXAMEN DE LA FIABILITÉ DES STRUCTURES 71
4.3 EQUATIONS DU PROBLÈME DE VIBRATION MÉCANIQUE 72

4.4 Formulation du problème d'optimisation sous contrainte fiabiliste . . . 73
4.5 Propagation des incertitudes pour le système mécanique . . . 74
4.6 Amélioration de la méthode NNC . . . 75
4.6.1 Introduction . . . 75
4.6.2 Formulation et amélioration de la méthode NNC . . . 75
4.7 Applications . . . 78
4.7.1 Premier Problème d'optimisation . . . 78
4.7.2 Le deuxième problème d'optimisation . . . 79
4.8 Comparaison avec d'autres méthodes d'optimisation multiobjectif . . . 81
4.9 Conclusion . . . 85

5 Approximation Numérique de La Solution en Optimisation en Dimension Infinie en utilisant la Formule de Representation 86
5.1 Introduction . . . 87
5.2 Notation . . . 88
5.3 Mesures finies dans un espace de Hilbert à dimension infinie . . . 89
5.4 Intégration avec une mesure sur un espace de Hilbert 89
5.5 Hypothèses . . . 91
5.6 Une representation de l'optimum global . . . 91
5.7 Approche numérique . . . 95
5.8 Résultats Expérimentaux . . . 97
5.8.1 Influence de la fonction de Pincus . . . 98
5.8.2 Problèmes élémentaires . . . 98
5.8.3 Problème du Brachistochrone . . . 100
5.8.4 Problème Exponentiel . . . 100
5.8.5 Problème de la chaînette . . . 102
5.8.6 Géodésiques dans un cylindre . . . 102
5.9 Conclusion . . . 103

6 Representation de Solution pour l'optimisation Multi-objectif : Génération suffisante de la courbe de Pareto 105
6.1 Introduction . . . 105
6.2 Representation de Solution pour l'optimisation Multi-Objectif(RSMO) . . . 106
6.2.1 Principe de base . . . 106
6.2.2 Formulation . . . 106
6.2.3 Algorithme . . . 109
6.3 Validation expérimentale . . . 109
6.3.1 Exemple illustratif : problème Zhang . . . 110
6.3.2 Comparaison avec d'autres méthodes multi-objectif . . . 112
6.3.2.1 Problème Deb (1999) . . . 113
6.3.2.2 Problème de Zitzler ZDT1 . . . 116
6.4 Conclusion . . . 118

Conclusion générale et perspectives 119

Bibliographie 123

ANNEXE 131

LISTE DES FIGURES

1.1 Exemple de simplexe dans $\mathbb{R}^2$ (à gauche) et $\mathbb{R}^3$(à droite) 10
1.2 Opération de réflexion du simplexe 10
1.3 Opération d'expansion du simplexe 10
1.4 Opération de contraction interne (à droite) et externe (à gauche) du simplexe 10
1.5 Opération de rétrécissement du simplexe 10
1.6 Concepts de base : Espace des variables de décision (à gauche) et espace des objectifs(à droite) 19
1.7 Illustration de l'optimalité de Pareto dans l'espace des objectifs 19
1.8 Illustration des relations possibles entre les solutions 20
1.9 Illustration des ensembles de solutions localement et globalement optimales dans l'espace des objectifs. 22
1.10 La méthode d'agrégation 23
1.11 Représentation géométrique de la méthode ε-Contrainte 24
1.12 Principe de fonctionnement de VEGA 25
1.13 Principe de base de la NBI : CHIM 32
1.14 Principe de base de la NBI 32
1.15 Génération du vecteur beta 33
1.16 La méthode NNC pour le cas de problème à deux dimensions. 35

3.1 Schéma et dimension du pilier 58
3.2 PSD et PDF de la charge appliquée au pilier 59
3.3 PSD et PDF pour 1,000 kg (à gauche) et 1,500 kg (à droite) du pilier . . 61
3.4 Front de Pareto pour 1,000 kg (à gauche) et 1,500 kg (à droite) du pilier . 61
3.5 Schéma et dimensions du treillis à deux bars 62
3.6 PSD du treillis à deux barres pour quatre extremum ; Ligne continue épaisse : $S_1 = 22$ mm^2 et $S_2 = 50$ mm^2 ; traits interrompus : $S_1 = 22$ mm^2 et $S_2 = 100$ mm^2 ; ligne continue mince : $S_1 = 200$ mm^2 et $S_2 = 50$ mm^2 ; ligne en pointillés : $S_1 = 200$ mm^2 et $S_2 = 100$ mm^2 64
3.7 Surfaces de réponses du treillis à deux barres pour la moyenne (gauche) et l'écart type (droite) pour la bande de fréquence [0-200] Hz 64
3.8 Surfaces de réponses du treillis à deux barres pour la moyenne (gauche) et l'écart type (droite) pour la bande de fréquence [100-300] Hz 64
3.9 Surfaces de réponses du treillis à deux barres pour la moyenne (gauche) et l'écart type (droite) pour la bande de fréquence [600-800] Hz 65
3.10 Front de Pareto pour le treillis à deux barres ; μ_V représente la moyenne de V et σ_V représente son écart type. 65
3.11 Front de Pareto pour treillis à deux barres, en utilisant GA(a) et RFGA(b). 67
3.12 Front de Pareto pour treillis à deux barres, en utilisant PSO(a) et RFPSO(b). 67
3.13 Front de Pareto pour treillis à deux barres, en utilisant GANM(a) et NSGA II(b). 68

4.1 SDOF system 73

4.2 Amplitude de la réponse en fréquence du système à SDOF ; en trait fin la réponse du système nominal ; zone grise pour la dispersion d'amplitude totale ; gris clair est pour une probabilité plus grande que 75%. 75
4.3 Méthode de la normalisation de la contrainte normal (NNC) pour le cas bi-dimensionnel. 76
4.4 Evolution de la méthode NNC, point par point, pour le cas de problème non convex à deux dimensions. 77
4.5 Evolution de la méthode INNC(NNC améliorée), point par point,pour le cas de problème à deux dimensions. 77
4.6 Amplitude de réponse en fréquence pour un système SDOF déterministe et l'amplitude coque réponse maximale pour le premier problème 80
4.7 front de Pareto pour le premier problème ; • : est pour $P_{f1} = 25\,\%$, + : pour $P_{f2} = 0\%$. 80
4.8 Amplitude de réponse du système SDOF aux points de conception H (haut, gauche), I (haut, droite), J (bas). La zone grise montre la dispersion qui respect la contrainte $P_f \leq P_{f1}$; le trait fin indique la réponse du système nominal. 81
4.9 Front de Pareto pour le second problème. 82
4.10 Graphes du **system response amplitude** de SDOF aux points de conception $\tilde{I}_1$ (à gauche) et $\tilde{I}_2$ (à droite). 82
4.11 Front de Pareto généré par la méthode NSGA II 83
4.12 Front de Pareto généré par la méthode SPEA2 84
4.13 Front de Pareto généré par la méthode NBI 84
4.14 Front de Pareto généré par les méthodes NNC et INNC 84

5.1 Interface informatique des tests. 97
5.2 Résultat obtenu par la formule de représentation seule, Résultat amélioré et la
solution exacte pour le problème du polynôme (a) et pour le problème du sinus (b) pour l'essai $N°5$,
tableau 5.1 et essai $N°3$, tableau 5.2 respectivement. 99
5.3 Solution par le formule de représentation, améliorée et la solution exacte pour le problème du brachistochrone pour le test $N°12$, Tableau 5.3. . . . 101

6.1 Méthode RSMO : Principe de base 107
6.2 Courbe de Pareto Pour le problème 1 (Zhang) avec RSMO) 110
6.3 Courbe de Pareto pour 3 et 5 points 111
6.4 Courbe de Pareto pour 5 et 7 points 111
6.5 Courbe de Pareto pour 10 et 20 points 111
6.6 Courbe de Pareto pour 30 et 50 points 111
6.7 Evolution de l'érreur relative en fonction du nombre de points de Pareto générés par RSMO 112
6.8 Front de Pareto généré par la méthode NBI pour le problème Deb 114
6.9 Front de Pareto généré par la méthode NNC(fonction normalisée) pour le problème Deb 114
6.10 Front de Pareto généré par la méthode NSGAII pour le problème Deb . . 115
6.11 Front de Pareto généré par la méthode RSMO pour le problème Deb . . . 115
6.12 Front de Pareto généré par les méthode NBI et NNC pour le problème ZDT1 117
6.13 Front de Pareto généré par la méthode NSGAII pour le problème ZDT1 . 117
6.14 Front de Pareto généré par la méthode RSMO pour le problème ZDT1 . . 118

15 Représentation graphique de Cosine mixture pour d=2 131
16 Représentation graphique de Rosenbrock pour d=2 132

17 Représentation graphique de Goldstein and Price pour d=2 132
18 Représentation graphique de Camel pour d=2 133
19 Représentation graphique de Camel pour d=2 133
20 Représentation graphique de Shubert pour d=2 134
21 Représentation graphique de Rastrigin pour d=2 135
22 Représentation graphique de Griewank pour d=2 136

Liste des tableaux

2.1 Influence de la taille de la population pour GA et GANM 51
2.2 Influence de la taille de la population pour RFGA et RFGANM 52
2.3 Influence de la taille d'échantillon utilisé dans RF 53
2.4 Comparaison entre méthodes : GA, GANM et RF 54
2.5 Comparaison entre méthodes : RFGA, RFGANM et RFNM 54
2.6 Comparison avec d'autre méthodes, liste des méthodes 54
2.7 Comparison avec d'autres méthodes en terme de taux de succès et nombre d'évaluations de fonction 55

3.1 Comparaison entre les méthodes 67

4.1 Coordonnées des points qui définissent la coque de réponse d'amplitude de crête pour le premier problème 78
4.2 Coordonnées des points characteristiques dans le front de Pareto (à gauche pour $P_{f1} = 25\,\%$ et à droite pour $P_{f2} = 0$) pour le premier problème. 78
4.3 Coordonnées des points qui définissent **the amplitude peak response hull** pour le deuxième problème d'optimisation 81
4.4 Coordonnées des points characteristiques du front de Pareto pour le second problème d'optimisation. 81
4.5 Comparison entres le méthodes INNC, NNC, NBI, SPEA2 et NSGA II . 83

5.1 Problème test 1 : Polynôme, pour la stratégie S1 99
5.2 Problem test 1 : Sinus, pour la stratégie S1 100
5.3 Problème test 2 : Brachistochrone, pour la stratégie S2 101
5.4 Problème test 3 : Problème exponential, pour la stratégie S3 102
5.5 Problème test 4 : Catenary, pour la stratégie S3 103
5.6 Problème test 5 : Géodésique, pour les strategies $S1$, $S2$ et $S3$ 103

6.1 Comparaison entres les méthodes NBI, NNC, NSGA II, et RSMO pour le problème Deb 114
6.2 Comparison entres les méthodes NBI, NNC, NSGA II, et RSMO pour le problème ZDT1 116

3 four-dimensional Shekel function 134
4 informations basiques sur la fonction Shekel 135

CONTEXTE, OBJECTIF ET ORGANISATION DE LA THÈSE

CONTEXTE

L'optimisation est au coeur de tout problème lié à la prise de décison, que ce soit en ingénierie ou en économie. Le but ultime de toutes ces décisions est soit de minimiser l'effort requis et/ou maximiser le bénéfice désiré. Les problèmes d'optimisation rencontrés en pratique sont rarement mono-objectif ; plusieurs paramètres contradictoires à satisfaire sont présents. L'optimisation multiobjectif est disponible depuis environ trois décennies, et son application dans les problèmes du monde réel ne cesse d'augmenter. Son but est de chercher à optimiser plusieurs composantes de la fonction objectif. La solution du problème n'est pas un vecteur unique, mais un ensemble de solutions connu comme l'ensemble de solutions Pareto optimales. Il n'existe pas de méthode unique disponible pour résoudre efficacement tous les problèmes d'optimisation. Plusieurs algorithmes d'optimisation ont été proposés, examinés et analysés dans les dernières décennies. Cependant, l'optimisation dans le domaine de l'ingénierie demeure un champ actif de recherches, puisque plusieurs problèmes réels d'optimisation restent très complexes en réalité et difficiles à résoudre par les algorithmes existants. La littérature existante présente des efforts intensifs de recherches pour résoudre certaines difficultés, pour lesquels seulement des réponses partielles ont été obtenues.

PROBLÉMATIQUE

Parmi les difficultés rencontrées lors de la résolution de ces difficultés, nous pouvons citer : la manipulation de la non convexité (particulièrement quand les contraintes d'optimisation sont impliqués), la manipulation de fonctions ou contraintes avec des évaluations incomplètes ou incorrectes, augmentation du nombre de variables d'optimisation au delà de ceux des conceptions réalistes dans la pratique, traitement des fonctions non régulières (discontinues ou non differentiables) et l'identification des points initiaux convenables pour les méthodes itératives. Nous observons que les difficultés liées à la non-convexité et le choix des points de départ sont liés : les méthodes efficaces pour l'optimisation des fonctions régulières sont souvent déterministes et utilisent des gradients, mais dépendent fortement du point initial ; elles peuvent être emprisonnées par des minimum locaux si un mauvais choix de ce point est utilisé. Alternativement, les méthodes

basés sur l'exploration de l'espace des variables de conception impliquent, habituellement, un aspect stochastique (ainsi, une croissance significative en coût de traitement) et sont moins dépendantes du choix initial, mais les améliorations de leurs performances nécessitent la combinaison avec des méthodes déterministes qui peuvent présenter une dépendance à l'égard du choix initial. Cette dernière approche tend au recours à des méthodes hybrides mettant en jeu les deux approches et essaie de tirer bénéfice du meilleur de chaque méthode. C'est pour cette raison que la littérature traitant les méthodes stochastiques/déterministes hybrides s'est développée dans ces dernières années. Ces algorithmes hybrides fonctionnent mieux si le point initial appartient à une zone d'attraction proche de l'optimum. Ceci montre l'importance de l'estimation du point initiale pour les algorithmes d'optimisation.

Objectifs

Cette thèse a été menée au sein du Laboratoire d'Étude et Recherche en Mathématiques Appliquées (LERMA) à l'École Mohammadia d'Ingénieurs, Université Mohammed V, Agdal, Maroc et au sein du laboratoire d'Optimisation et Fiabilité en Mécanique des Structures(LOFIMS), à l'Institut National des Sciences Appliquées (INSA) de Rouen, France.
Elle a pour objectif principal le développement de nouveaux algorithmes globaux pour la résolution de problèmes d'optimisation mono et multi-objectif, en se basant sur des formules de représentation ayant la tâche principale de générer des points initiaux appartenant à une zone proche du minimum globale. Une hybridation avec des algorithmes locaux et globaux a été envisagée, avec des tests de performance pour comparer l'efficacité et l'efficience des algorithmes proposés. Des applications sur des problèmes de mécanique avec des variables aléatoires ont été réalisées.

Organisation de la thèse

La thèse est organisée en six chapitres :
Le chapitre I dresse un état de l'art non exhaustif des méthodes d'optimisation mono-objectif, et multi-objectif en accordant une importance particulière aux méthodes utilisées dans nos travaux de recherche ; notamment l'algorithme de Nelder Mead, l'algorithme génétique, l'algorithme d'optimisation par essaim des particules, ainsi que les méthodes d'optimisation multi-objectif NBI, NNC, SPEA2, NSGAII. D'autres méthodes bien connues seront brièvement décrites en faisant référence à la documentation pour tout besoin d'informations ou de détails supplémentaires.

Le chapitre II présente notre contribution à l'optimisation globale et mono-objectif en proposant l'algorithme RFNM. C'est un couplage de la formule de représentation reformulée par Sauza et l'algorithme de Nelder Mead. Dans un premier temps, la formule de représentation est employée pour trouver la région contenant la solution globale. Afin d'obtenir des résultats plus précis, le résultat est amélioré par un couplage avec l'algorithme de Nelder Mead et l'algorithme génétique sous plusieurs formes. Des tests sur 21 fonctions tests ont été présentés pour comparer la performance des algorithmes obtenus et étudier les effets des différents paramètres mis en jeux.

Le chapitre III constitue des applications de l'algorithme RFNM couplé avec la méthode NBI pour traiter de nouveaux problèmes de conception mécanique. Il s'agit de deux problèmes linéaires avec une charge aléatoire gaussienne transformés en problèmes d'optimisation multi-objectif. Le premier problème est la conception d'une géométrie de pilier à l'égard d'un processus de chargement aléatoire de compression. Le deuxième

problème est la conception d'une structure en treillis par rapport à un processus de chargement aléatoire vertical pour plusieurs bandes de fréquences. l'algorithme RFNM, motivé par la formule de représentation hybridée avec l'algorithme de Nelder-Mead, est proposée pour résoudre les deux problèmes d'optimisation multi-objectif en le couplant avec la méthode NBI. En fonction de la bande de fréquence d'excitation, on peut avoir comme courbe de Pareto un seul point, une courbe convexe de Pareto, ou une courbe discontinue de Pareto. Ce cas est confronté pour la première fois dans la littérature, à notre connaissance. Une comparaison des performances de RFNM couplé avec NBI, avec l'algorithme génétique(GA), l'algorithme essaim de particule(PSO), et NSGA II est réalisée afin de tester l'efficacité et l'efficience de notre algorithme.

Le chapitre IV traite de la conception d'un dispositif de suspension, modélisé en tant qu'un système ressort-masse-amortisseur. L'amplitude d'un système nominal est contrainte de satisfaire certaines limites dans une bande de fréquence donnée et la conception doit être faite comme une optimisation sous contrainte fiabiliste. Pour se concentrer sur l'idée principale, seule la raideur du système sera considéré comme aléatoire. Le problème de conception est énoncée comme une optimisation bi-objectif dont les deux fonctions objectif sont la moyenne et l'écart-type de la raideur. On cherche à minimiser la raideur et à maximiser l'écart-type, alors que la réponse en amplitude doit être comprise dans le domaine acceptable de vibration prescrit. Pour générer le front de Pareto, on propose une amélioration de la méthode NNC dite INNC en utilisant RFNM. Une comparaison des résultats obtenus en termes de temps d'exécution et nombre d'évaluations de la fonction objectif avec d'autres méthodes d'optimisation multi-objectif telles que NBI, NNC, NSGA II et SPEA2 a été réalisé.

Dans le chapitre V, nous proposons une extension de la formule de représentation en dimension infinie. Pour cela, on considère un problème d'optimisation non convexe, impliquant une fonctionnelle régulier J, sur un sous-ensemble fermé et borné S d'un espace de Hilbert séparable V. Aucune hypothèse de convexité n'est introduite. Les solutions sont représentées à l'aide d'une formule close impliquant des moyens de variables aléatoires convenables. La représentation nécessite l'utilisation d'une méthode numérique basée sur la génération d'échantillons en vue d'estimer ces moyens. Trois stratégies pour la mise en oeuvre sont examinées, avec l'originalité qu'ils n'impliquent pas une approximation en dimension finie à priori de la solution et considèrent une base hilbertienne ou une famille dense dénombrable. Les résultats peuvent être améliorés sur un sous-espace de dimension finie par une procédure d'optimisation, afin d'obtenir des solutions de meilleure qualité. Des exemples numériques impliquant des situations classiques issues du calcul des variations sont présentées.

Le chapitre VI présente une contribution à l'optimisation multi-objectif par une nouvelle approche, appelée (RSMO), pour générer un nombre suffisant de points pour représenter la solution optimale le long de la frontière de Pareto. Cette approche peut être considérée comme une formule de représentation de solution du problème d'optimisation multi-objectif. La méthode peut traiter à la fois les problèmes convexes et non convexes. Dans une première étape, nous nous intéresserons aux problèmes d'optimisation multi-objectif convexe, et nous illustrons la méthode et les idées sous-jacentes à travers plusieurs exemples. Des comparaison avec des méthodes d'optimisation multi-objectif connues sont prévues.

Nous conclurons le manuscrit par des conclusions et des perspectives sur les travaux effectués ainsi que des annexes donnant des compléments d'informations.

CHAPITRE 1

ÉTAT DE L'ART DES MÉTHODES D'OPTIMISATION MONO-OBJECTIF ET MULTI-OBJECTIF

SOMMAIRE

2.1 INTRODUCTION 37
2.2 FORMULE DE REPRÉSENTATION 38
2.2.1 Introduction 38
2.2.2 Hypothèses et notations 39
2.2.3 Le résultat central 40
2.2.4 Choix de la fonction 41
2.3 HYBRIDATION DE LA MÉTHODE DU SIMPLEXE AVEC LA FORMULE DE REPRÉSENTATION ET L'ALGORITHME GÉNÉTIQUE . . 43
2.3.1 Introduction 43
2.3.2 La Formule de Representation 44
2.3.3 Algorithme : 45
2.4 FONCTIONS TEST 45
2.5 RÉSULTATS NUMÉRIQUES 48
2.5.1 Influence du choix de la distribution de probabilité 49
2.5.2 Influence de la fonction g 49
2.5.3 Influence de la taille de la population pour GA 50
2.5.4 Influence de la taille de l'échantillon pour RF 50
2.5.5 Comparison entre méthodes 50
2.5.6 Comparison avec d'autres algorithmes 52
2.6 CONCLUSION 53

L'OPTIMISATION est le fait d'obtenir le meilleur résultat dans des circonstances données. Au niveau de la conception, la construction et l'entretien d'un système d'ingénierie, les ingénieurs doivent prendre de nombreuses décisions techniques et de gestion à plusieurs niveaux. Le but ultime de toutes ces décisions est soit de minimiser l'effort requis et/ou maximiser le bénéfice désiré. Puisque l'effort nécessaire ou la prestation souhaitée

dans une situation concrète peut s'exprimer en fonction de certaines variables de décision, l'optimisation peut être définie comme le processus de recherche des conditions qui donnent la valeur maximale ou minimale d'une fonction. Il n'existe pas de méthode unique disponible pour résoudre efficacement tous les problèmes d'optimisation. D'où un certain nombre de méthodes d'optimisation ont été développées pour résoudre différents types de problèmes d'optimisation.
L'existence de méthodes d'optimisation peut remonter à l'époque de Newton, Lagrange et Cauchy. Le développement de méthodes de calcul différentiel d'optimisation a été possible grâce à la contribution de Newton et de Leibniz. Les bases de calcul des variations, qui traite de la minimisation de fonctionnelles, ont été portées par Bernoulli, Euler, Lagrange, et Weirstrass. La méthode d'optimisation pour les problèmes avec contraintes, ce qui implique l'addition des multiplicateurs inconnus, est devenue connue sous le nom de son inventeur, Lagrange. Cauchy a fait la première application de la méthode de la plus forte descente pour résoudre les problèmes de minimisation sans contraintes(Rao 2009).
En dépit de ces premières contributions, très peu de progrès a été fait jusqu'au milieu du XXe siècle, quand les ordinateurs ont permis de mettre en oeuvre les procédures d'optimisation et de stimuler de nouvelles recherches sur de nouvelles méthodes. Des progrès spectaculaires ont suivi, produisant une littérature considérable sur les techniques d'optimisation. Ce progrès a également donné lieu à l'émergence de plusieurs domaines de recherche théorique sur les techniques d'optimisation. La volonté d'optimiser plus d'un objectif ou un but tout en satisfaisant les contraintes physiques a conduit à la mise au point de méthodes de programmation multicritères(Rao 2009).Dans ce chapitre, nous dressons un état de l'art non exhaustif des méthodes d'optimisation mono-objectif et multi-objectif en accordant une importance particulière aux méthodes utilisées dans nos travaux de recherche ; notamment la méthode de Nelder Mead, l'algorithme génétique, la méthode d'optimisation par essaim des particules, ainsi que les méthodes d'optimisation multi-objectif NBI, NNC, SPEA, NSGA. D'autres méthodes bien connues seront brièvement décrites en faisant référence à la documentation pour tout besoin d'informations ou de détails supplémentaires.

1.1 Optimisation mono-objectif

1.1.1 Introduction

Le développement de la méthode du simplexe par Dantzig en 1947 pour des problèmes de programmation linéaire et l'énoncé du principe d'optimalité de Bellman en 1957 pour des problèmes de programmation dynamique a ouvert la voie pour le développement des méthodes d'optimisation sous contraintes. Les travaux de Kuhn et Tucker en 1951, sur les conditions nécessaires et suffisantes pour la solution optimale de problèmes de programmation ont posé les bases d'un grand nombre de recherches plus tard dans la programmation non linéaire. Les contributions des Zoutendijk et Rosen à la programmation non linéaire au cours des années 1960 ont été considérables. Même si aucune technique n'a été trouvée pour être universellement applicable pour les problèmes de programmation non linéaire, les travaux de Carroll et Fiacco et McCormick ont permis de résoudre de nombreux problèmes difficiles en utilisant les techniques bien connues de l'optimisation sans contrainte. La programmation géométrique a été développée dans les années 1960 par Duffin, Zener, et Peterson. Gomory a fait oeuvre de pionnier en programmation en nombres entiers, l'un des domaines les plus passionnants de d'optimisation qui se développent rapidement. La raison en est que la plupart des applications du monde réel entrent dans cette catégorie de problèmes. Dantzig et Charnes et Cooper ont développé des techniques de programmation stochastique et résoudre les problèmes

en supposant que les paramètres de conception sont indépendants et distribués normalement(Rao 2009).

1.1.2 Formulation

Dans sa forme générale, un problème d'optimisation mono-objectif consiste en une fonction objectif à valeurs réels à minimiser, et un certain nombre de contraintes d'égalité ou d'inégalité :

$$\begin{cases} & \min \ f(\mathbf{x}) \\ \text{sujet à} & g_i(\mathbf{x}) \leq 0 \ , \ i = 1, ..., m, \\ & h_j(\mathbf{x}) = 0 \ , \ j = 1, ..., p, \\ & x^L \leq x \leq x^U \end{cases} \tag{1.1}$$

où $x \in \mathbb{R}^n$ est un vecteur de variables de décision, $g_1, .., g_m$ et $h_1, \cdots, h_p$ sont les fonctions de contraintes d'inégalité et des contraintes d'égalité, respectivement ; et x^L, x^U sont respectivement la borne inférieure et supérieure du domaine de recherche. Cet ensemble de contraintes définit l'espace faisable C des variables de décision :

$$C = \{ \ x \in \mathbb{R}^n \ / \ h(x) = 0, \ \ g(x) \leq 0 \ \ et \ \ x^L \leq x \leq x^U \ \}$$

Résoudre un problème d'optimisation consiste à trouver un point $x^* \in C$ tel que :

$$\forall \, x \in C, \ \ f(x^*) \leq f(x)$$

Les problèmes d'optimisation peuvent être classés de plusieurs manières, en fonction de la nature de la fonction, des variables de décision et des contraintes :

1. Classification basée sur l'existence de contraintes : qui concerne l'existence ou non de limitations sur les variables (contraintes) qui peuvent être simplement des bornes géométriques ou des égalités ou inégalités non linéaires.

2. Classification basée sur la nature des variables de conception : Dans certains problèmes d'optimisation, les variables n'ont un sens que si elles sont entières, on parle dans ce cas d'optimisation discrète. L'optimisation continue -objet de cette thèse- a pour objectif de chercher la solution dans un ensemble continue tel que l'ensemble des nombres réels.

3. Classification basée sur la nature des équations impliquées : Une classification importante est basée sur la nature des expressions pour la fonction objectif et des contraintes. Selon cette classification, les problèmes d'optimisation peuvent être classés comme des problèmes de programmation linéaire, non-linéaire, géométrique, et quadratique.

4. Classification basée sur la nature déterministe des variables : Les problèmes d'optimisation peuvent être classés comme des problèmes de programmation déterministes et stochastiques.

Ainsi, il est important de savoir situer le problème d'optimisation posé, afin de choisir la méthode appropriée pour le résoudre.
Il y a trois grandes familles de méthodes d'optimisation : les méthodes énumeratives, les méthodes déterministes, et les méthodes stochastiques.

1.1.3 Méthodes énumeratives

L'évaluation de la valeur de la fonction objectif est faite en chaque point de l'espace des solutions qui peut être fini ou infini mais discrétisé. Cette méthode est intéressante dans le cas où le nombre de points n'est pas très important ce qui n'est pas le cas dans la pratique. Ces méthodes ont deux inconvénients majeurs :

- Inadaptation aux problèmes de grande taille.
- Absence de processus intelligent qui guide la recherche vers un sous espace sans parcourir l'ensemble des solutions possibles.

1.1.4 Méthodes déterministes

Ces méthodes n'utilisent aucun concept aléatoire, et se caractérisent par une exploration systématique de l'espace de recherche. Elles requirent en générale des hypothèses sur la fonction objectif, telles que la continuité et dérivabilité en tout point du domaine de recherche. Les méthodes déterministes se divisent en deux classes principales : les méthodes d'exploration directes et les méthodes d'exploration indirecte. Ces derniers cherchent à atteindre les extrema locaux en résolvant les systèmes d'équations, souvent non linéaires, obtenus en annulant le vecteur gradient de la fonction étudiée. Les méthodes d'exploration directes recherchent les optima locaux en se déplaçant dans une direction qui dépend du gradient de la fonction. Deux inconvénients majeurs se présentent pour ces méthodes :

- Dans la pratique les fonctions à optimiser peuvent ne pas être dérivables et souvent même pas continues.
- Risque de convergence prématurée vers un optimum local, l'optimum global n'est obtenu que lorsque le point initial de départ choisi est proche de cet optimum.

1.1.4.1 Les méthodes de gradient

Les méthodes de gradient sont simples à mettre en oeuvre et donnent souvent de bons résultats. Pour cette raison, ils sont largement utilisés dans des applications pratiques. Ces méthodes procèdent de la façon suivante : à partir d'un point de départ x^0, on calcule le gradient $\nabla f(x^0)$. Ce gradient indique la direction de la plus grande augmentation de f, on se déplace d'une quantité λ^0 dans le sens opposé au gradient et on a le point suivant x^1 :

$$x^1 = x^0 - \lambda^0 \frac{\nabla f(x^0)}{\|\nabla f(x^0)\|}$$

Ainsi on engendre une série de point x^0, x^1, x^2,...,x^k qui se rapprochent de plus en plus de l'optimum.

$$x^{k+1} = x^k - \lambda^k \frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$$

λ^k >0 est le pas de déplacement à chaque itération. Si le pas est fixe, on parle de méthode de gradient à pas prédéterminé. L'inconvénient dans de ce choix est que la convergence dépendra beaucoup du choix du pas et risque de ralentir le processus. Pour remédier à ce défaut, la méthode de la plus forte pente est utilisée. Elle permet de se libérer du choix de λ^k qui doit être calculée en minimisant à chaque itération la fonction : $g(\lambda) = f(x^k - \lambda \nabla f(x_0))$ càd $\lambda^k = Arg \min_\lambda (f(x_k - \lambda \nabla f(x^k)))$(Chong et Zak 2011).Cet algorithme est la base de l'algorithme de Hill-Climbing appelé également algorithme de descente de gradient.

1.1.4.2 La méthode de Newton

L'idée de base de la méthode de Newton pour l'optimisation sans contrainte consiste à utiliser de manière itérative l'approximation quadratique de la fonction objectif f à l'itération courant x_k et de minimiser cette approximation. Cette méthode suppose que la fonction f est continue et deux fois différentiable et le hessien $\nabla^2 f(x^k)$ est défini positif. De la même manière que la méthode du gradient la suite $\{x_k\}$ est définit par :

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} + \nabla^2 f(x^k)$$

. Il existe une littérature abondante sur les méthodes à base de Newton (Kelley 1995; 1999; 2003, Dennis et Schnabel 1987)

1.1.4.3 La méthode multistart

Dans le cas de fonctions non convexes, les méthodes de gradient peuvent converger vers un optimum local. Pour remédier à ce problème la méthode multistart effectue une recherche locale à partir de plusieurs points répartis dans l'espace de recherche. Un maillage adéquat doit être établi au préalable en respectant la forme de la fonction objectif. Si le maillage est trop grand on risque de ne pas traiter certains intervalles. Dans le cas contraire la recherche ne sera pas efficace puisque plusieurs points de départ donneront lieu au même résultat.

1.1.4.4 La méthode de Nelder Mead

1.1.4.4.1 introduction : La première méthode de recherche directe basée sur le principe de simplexe a été proposé par Spendley, Hext et Himsworth en 1962 (Spendley et al. 1962). Elle n'utilise que deux types de transformations pour former un nouveau simplexe à chaque étape : La réflexion loin du mauvais sommet (celui avec la plus grande valeur de la fonction), ou rétricissement vers un meilleure sommet (avec la plus petite valeur de la fonction). Dans ces transformations, les angles entre les arêtes dans chaque simplexe restent constants tout au long des itérations, de sorte que le simplexe de travail peut changer de taille, mais pas en forme. En 1965, Nelder et Mead a modifié la méthode originale de Spendley et al. en incluant deux transformations supplémentaires (expansion et contraction), qui permettent au simplexe de changer non seulement sa taille, mais aussi sa forme (Nelder et Mead 1965). Elle devient une technique directe et simple de recherche qui a été largement employée pour résoudre des problèmes d'optimisation sans contrainte. Une des raisons de sa popularité est que cette méthode est facile à utiliser et n'a pas besoin d'informations sur la dérivée de la fonction objectif. C'est une caractéristique très importante dans de nombreux cas où les informations sur le gradient n'est pas toujours disponible. Cependant, on doit faire beaucoup attention en employant cette méthode puisqu'elle est très sensible au choix des points initiaux et ne garantit pas d'obtenir l'optimum global(Wright 1996, Powell 1998, Kolda et al. 2003).Cette méthode ne doit pas être confondue avec la méthode du simplexe de Dantzig pour la programmation linéaire, qui est complètement différente, car il résout un problème linéaire sous des contraintes linéaires. En dépit de son âge et les progrès récents dans les méthodes de recherche directe, la méthode de Nelder-Mead est toujours parmi les méthodes les plus populaires de recherche directe dans la pratique.

1.1.4.4.2 Description : Cette méthode commence par un simplexe de $n+1$ sommets, qui est un point de $\mathbb{R}^n$, notés $x_0, x_1, \ldots, x_n$, tel que $f(x_0) \leq f(x_1) \leq \ldots \leq f(x_n)$ voir figure 1.1. Puisqu'on cherche à minimiser f, x_0 est le meilleur point et x_n est le mauvais. A chaque itération, le mauvais point x_n est éliminé, et un autre point est accepté dans le simplexe. Ce processus continue jusqu'à ce que la convergence soit réalisée. Cet algorithme utilise quatre paramètres scalaires : coefficient de réflexion (ρ), d'expansion (χ), de contraction (γ), et de rétrécissement (σ). Selon le papier original de Nelder Mead (Nelder et Mead 1965), ces paramètres doivent satisfaire :

$$\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1, \quad and \quad 0 < \sigma < 1.$$

Dans l'algorithme standard de Nelder Mead, ces coefficients sont choisis comme suit :
$\rho = 1, \quad \chi = 2, \quad \gamma = \frac{1}{2} \quad and \quad \sigma = \frac{1}{2}.$

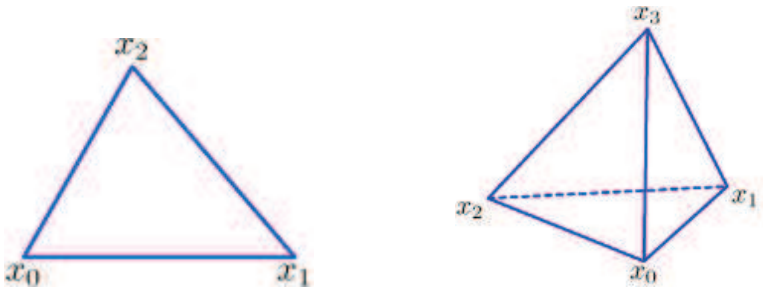


FIGURE 1.1 – *Exemple de simplexe dans* $\mathbb{R}^2$ *(à gauche) et* $\mathbb{R}^3$*(à droite)*

1.1.4.4.3 Algorithme : Etant donné un premier point x_0, l'algorithme construit un simplexe $x_0^1, \ldots, x_0^{n+1}$ en additionnant 5 % à chaque composant x_0^i à x_0, et en utilisant ces n vecteurs comme éléments du simplexe en plus de x_0. Alors l'algorithme modifie le simplexe à plusieurs reprises selon les étapes suivants (voir algorithme 1) :

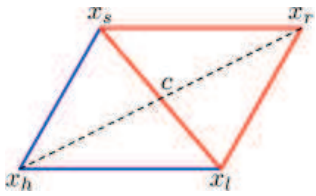


FIGURE 1.2 – *Opération de réflexion du simplexe*

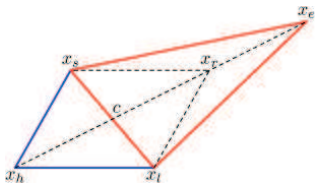


FIGURE 1.3 – *Opération d'expansion du simplexe*

FIGURE 1.4 – *Opération de contraction interne (à droite) et externe (à gauche) du simplexe*

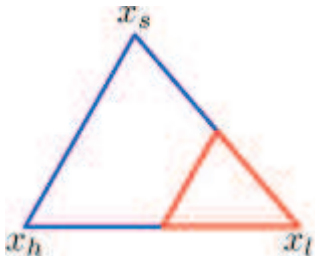


FIGURE 1.5 – *Opération de rétrécissement du simplexe*

Algorithme 1 : Algorithme de Nelder Mead

Etape 1. **Ordonner**. Ordonner les $n+1$ vecteurs du Simplexe.
Etape 2. **Réfléchir**. Générer les points réfléchis (voir figure 1.2) :

$$x^r = \bar{x} + \rho(\bar{x} - x^{n+1}),$$

Où $\bar{x} = \frac{1}{n}\sum_{i=0}^{n-1} x_i$. Ensuite, evaluer $f(x^r)$. Si $f(x^1) \leq f(x^r) < f(x^n)$ alors on accepte le point réfléchi x^r et on termine l'iteration.
Etape 3. **Elargir**. Si $f(x^1) > f(x^r)$ alors on calcule le point d'expansion (voir figure 1.3) :

$$x^e = \bar{x} + \chi(x^r - \bar{x}),$$

et on évalue $f(x^e)$. Si $f(x^e) < f(x^r)$ alors on accepte x^e et on termine l'iteration ; sinon on accepte x^r et on termine l'iteration.
Etape 4. **Contracter**. Si $f(x^r) \leq f(x^n)$, réaliser une contraction entre $\bar{x}$ et le meilleur entre x^{n+1} et x^r. (voir figure 1.4)

1. **Contraction externe.** Si $f(x^n) \leq f(x^r) < f(x^{n+1})$, cad, x^r est meilleur strictement que x^{n+1}, alors réaliser une contraction externe ; calculer :

   $$x^c = \bar{x} + \gamma(x^r - \bar{x}),$$

   et évaluer $f(x^c)$. Si $f(x^c) \leq f(x^r)$ alors on accepte x^c et on termine l'iteration ; sinon aller à l'Etape 5 (Rétrécir).
2. **Contraction interne.** Si $f(x^r) \geq f(x^{n+1})$, alors réaliser une contraction interne ; calculer :

   $$x^{cc} = \bar{x} + \gamma(\bar{x} - x^{n+1}),$$

   et évaluer $f(x^{cc})$. Si $f(x^{cc}) < f(x^{n+1})$ alors on accepte x^{cc} et on termine l'iteration ; sinon aller à l'Etape 5 (Rétrécir).

Etape 5. **Rétrécir**(voir figure 1.5). Calculer les n points

$$v^i = x^1 + \sigma(x^i - x^1),$$

et calculer $f(v^i)$, pour $i = 2, ..., n+1$. Les vecteurs du simplexe à la nouvelle iteration sont $x^1, v^2, ..., v^{n+1}$.

1.1.5 Méthodes stochastiques

Ce sont des méthodes itératives où l'optimisation est guidée partiellement ou totalement par un processus stochastique. Lorsqu'on veut résoudre un problème difficile, dès que sa dimension est grande, les méthodes d'exploration traditionnelles, déterministes ou énumératives peuvent avoir un temps de calcul déraisonnable, et on a alors recours aux méthodes stochastiques. Cependant, elles présentent l'inconvénient de ne pas garantir la convergence que d'une manière asymptotique (Wallace et Kall 1995, Prekopa 1995, Birge et Louveaux 1997). Parmi les différentes méthodes stochastiques d'optimisation globale, nous allons uniquement nous intéresser aux heuristiques modernes. Le mot heuristique vient du grec heurein "découvrir" et qualifie tout ce qui sert à la découverte, à l'invention et à la recherche. Une heuristique peut être conçue pour résoudre un type de problème donné, ou bien être conçue comme une méthode générale, qui peut être adaptée à divers problèmes d'optimisation, dans ce cas on parle de méta heuristique. Les méta-heuristiques sont à l'origine dédiées aux problèmes combinatoires, où les paramètres ne peuvent prendre que des valeurs discrètes. Pour l'optimisation d'une fonction continue,

ces méthodes d'optimisation peuvent être adaptées moyennant des transformations plus au moins aisées, en inventant une nouvelle topologie. Chaque paramètre doit être discrétisé de façon individuelle. La difficulté majeure réside dans la détermination de la taille optimale du pas de discrétisation et de sa direction. Les heuristiques comportent souvent plusieurs paramètres contrôlant les différents opérateurs et l'influence du ou des processus stochastiques. L'efficacité d'une heuristique dépend du choix de ses paramètres de contrôle. Ce réglage est complexe, surtout quand le nombre de paramètres est élevé et quand la plage de variation de chacun de ces paramètres est étendue. Les différents paramètres sont généralement corrélés, ce qui rend encore plus difficile leur réglage. Les métaheuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage, ou en mêlant ces notions de façon plus étroite. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint. Enfin pour un jeu de paramètres de contrôle donnés, l'aspect stochastique fait que les résultats varient d'une exécution à l'autre. Les principales métaheuristiques modernes les plus utilisées, suivant l'ordre chrnologique d'apparition, sont : les algorithmes génétiques(1975), la méthode du recuit simulé (1983), La méthode de recherche tabou(1986), l'algorithme de colonie de fourmis (1990) et l'algorithme essaim de particules (1995) (Bianchi et al. 2009).

1.1.5.1 Les algorithmes génétiques

1.1.5.1.1 Description L'algorithme génétique (GA) est une technique d'optimisation et de recherche basée sur les principes de la génétique et de la sélection naturelle. La méthode a été, à l'origine développée, par John Hollande (1975) et popularisée par David Goldberg (Goldberg 1998). GA est apprécié par sa flexibilité, malgré le coût élevé de calcul. Par exemple, GA peut résoudre des problèmes d'optimisation avec variables continues ou discrètes et il est bien adapté pour le calcul parallèle (Haupt et Haupt 1998). Plusieurs variétés de GA n'exigent pas d'information sur la dérivé et peuvent traiter des fonctions objectif discontinues. Cependant, le traitement des problèmes avec contraintes et le coût élevé de calcul peuvent limiter leur utilisation.
GA permet à une population composée d'individus d'évoluer de génération en génération (c-à-d, par l'application successive des opérateurs de croisement, de mutation et de selection) à un état qui maximise la qualité de la population (c.-à-d., réduit au minimum les valeurs de la fonction objectif)(Haupt et Haupt 1998, Goldberg 1998). Ce processus d'évolution continue jusqu'à ce que les critères d'arrêt soient atteints. Dans la formulation pionnière, chaque élément de l'espace de travail est codé en binaire, appelé chromosome. Les étapes de croisement et de mutation sont conçues dans une optique d'imitation de la nature : elles consistent en une combinaison et une transformation aléatoire du chromosome. Dans l'étape de la selection, on compare les nouveaux chromosomes produits et on favorise l'élimination de ceux ayant les mauvaises fitnesses (par exemple, nous pouvons utiliser la méthode de la "loterie biaisé" qui donne une plus grande probabilité de survie à la meilleure fitnesse). Ensuite, beaucoup d'approches alternatives ont été introduites. Par exemple, en traitant des fonctions à valeurs réelles, nous pouvons considérer une codification par des nombres réelles en utilisant un vecteur de points flottant au lieu de la codification binaire des chromosomes. Dans ce cas, l'opérateur de croisement peut être construit en empruntant le concept de la combinaison linéaire des vecteurs de l'ensemble convexe. L'opérateur de mutation peut consister en une modification déterministe, aléatoire ou hybride de l'élément, comme, par exemple, l'utilisation d'une descente par gradient (Janikow et Michalewicz 1991). Dans notre travail, nous employons la méthode de Nelder-Mead pour la construction d'un opérateur de mutation.

1.1.5.1.2 Algorithme

Algorithme 2 : Algorithme génétique

Etape 1 (**Initialisation**) :
- Générer N population / chromosomes en utilisant une distribution aléatoire normale dans S,
- Niter=0,

Etape 2 (**Evaluation**) : Évaluer le coût de chaque point /chromosome
Etape 3 (**Test d'arrêt**) : Si la condition d'arrêt = vrai alors Fin
Etape 4 (**Opération génétiques**) :
- ***Selection*** : Sélectionner $m \leq N$ parents parmi la population,
- ***Croisement*** : Créer m nouveaux points (offsprings), à partir des parents selectionnés précedement,
- ***Mutation*** : Muter les offsprings produits dans l'étape de croisement avec la probabilité P_m,

Etape 5 (**Remplacement**) : remplacer les m mauvaises chromosomes de la population avec les offsprings générés précédemment,
Etape 6 (**Technique locale**) : créer en utilisant la procedure de la technique local un point d'essai $\bar{x}$. Si $f(\bar{x}) \leq f(x_h)$ où x_h est le plus mauvais point en cours. alors on remplace x_h par $\bar{x}$,
Etape 7 (**Incrémentaient**) :
- ***Affecter*** : niter = niter+1,
- ***Aller à l'***Etape 2,

1.1.5.2 La méthode du recuit simulé

S. Kirkpatrick et ses collègues, spécialistes en physique statistique, (Kirkpatrick et al. 1983) (et indépendamment V. Cerny (Černỳ 1985)) ont proposé de traiter les problèmes rencontrés dans leurs travaux en prenant comme point de départ la technique expérimentale du *recuit* utilisée par les métallurgistes pour obtenir un état de solide *bien ordonnée*, d'énergie minimale (en évitant les structures *métastables*, caractéristiques des minima locaux de l'énergie). Cette technique consiste à chauffer le matériau à haute température, puis d'abaisser cette température lentement. Le procédé de recuit simulé transpose le processus de recuit à la recherche de la solution d'un problème d'optimisation : la fonction objective du problème, similaire à l'énergie d'un matériau, est alors réduit au minimum, à l'aide de l'introduction d'une température fictive, qui est , dans ce cas, un simple paramètre contrôlable de l'algorithme. L'algorithme est basé sur deux résultats de la physique statistique. D'une part, lorsque l'équilibre thermodynamique est atteint à une température donnée, la probabilité pour un système physique d'avoir une énergie E donnée, est proportionnelle au facteur de Boltzmann : $e^{-\frac{E}{k_B T}}$, où k_B représente la constante de Boltzmann. Ensuite, la répartition des états d'énergie est la distribution de Boltzmann à la température considérée. En outre, pour simuler l'évolution d'un système physique vers son équilibre thermodynamique à une température donnée, l'algorithme de Metropolis (Metropolis et al. 1953) peut être utilisé : sur la base d'une configuration donnée, le système est soumis à une modification élémentaire, si cette transformation provoque une diminution de la fonction objectif (ou "énergie") du système, elle est acceptée, au contraire, si elle provoque une augmentation δE de la fonction objectif, il est également admis, mais avec une probabilité $e^{-\frac{\delta E}{T}}$ (dans la pratique, cette condition est réalisée de la manière suivante : un nombre réel est tirés au hasard compris entre 0 et 1, et la configuration provoquant une dégradation δE dans la fonction objectif est acceptée, si le nombre aléatoire tiré est inférieur ou égal à $e^{-\frac{\delta E}{T}}$). En observant cette règle plusieurs fois, une séquence de configurations est générée, ce qui constitue une chaîne de Markov. Avec ce formalisme en place, il est possible de montrer que, lorsque la chaîne est de longueur in-

finie (suffisamment longue), le système peut atteindre l'équilibre thermodynamique à la température considérée : en d'autres termes, cela nous amène à une distribution de Boltzmann des états d'énergie à cette température. D'où le rôle confié à la température à l'aide de la règle de Metropolis. A haute température, $e^{-\frac{\delta E}{T}}$ est proche de 1, donc la majorité des mouvements sont acceptées et l'algorithme devient équivalent à une marche aléatoire simple dans l'espace des configurations. A basse température, $e^{-\frac{\delta E}{T}}$ est proche de 0, donc la majorité des mouvements croissants de l'énergie est refusée. Ainsi l'algorithme nous rappelle une amélioration itérative classique. À une température intermédiaire, l'algorithme autorise par intermittence les transformations qui dégradent la fonction objectif : d'où il laisse un champ pour le système pour être tiré à partir d'un minimum local. Une fois l'équilibre thermodynamique est atteint à une température donnée, la température est abaissée "légèrement", et une nouvelle chaîne de Markov est mise en oeuvre à ce stade de la nouvelle température. En comparant les distributions successives de Boltzmann obtenues à la fin des différents paliers de température, une augmentation progressive de la masse des configurations de basse énergie peuvent être observés : lorsque la température tend vers zéro, l'algorithme converge vers un minimum absolu de l'énergie. Dans la pratique, le processus est terminé lorsque le système est "durcie" (cela signifie que soit la température atteint la valeur finale ou il n'a y a pas d'amélioration de la fonction objectif)(Dréo et al. 2006). La méthode du recuit simulé a l'avantage d'être flexible par rapport aux évolutions du problème et facile à mettre en oeuvre. Il a donné d'excellents résultats pour un certain nombre de problèmes, généralement de grande taille. cependant, cette méthode a des inconvénients qui se situent d'une part dans le grand nombre de paramètres mis en jeu, et les "bons ajustements" à faire, comme le contrôle de la diminution de la température. En outre, le temps de calcul peut devenir très important (Szu et Hartley 1987).

1.1.5.3 La méthode de Recherche Tabou (RT)

La méthode tabou a été formalisée en 1986 par Glover (1986). Sa caractéristique principale est basée sur l'utilisation de mécanismes inspirés de la mémoire humaine. La méthode tabou prend, de ce point de vue, un chemin en face de celle du recuit simulé, qui n'utilise pas de mémoire du tout, et est donc incapable de tirer les leçons du passé. D'autre part, la modélisation de la mémoire présente plusieurs degrés de liberté, qui s'oppose à une analyse mathématique rigoureuse de la méthode tabou (Glover 1986, Gendreau et Centre for Research on Transportation Montréal(2002).
Le principe de la méthode tabou est simple : comme le recuit simulé, la méthode tabou fonctionne en même temps avec une seule "configuration actuelle", qui est mis à jour au cours des itérations successives. À chaque itération, le mécanisme de passage d'une configuration, appelée s, à la suivante, appelée t, comprend deux étapes :

- On construit l'ensemble des voisins de s, c'est à dire l'ensemble des configurations accessibles en un seul mouvement élémentaire de s (si cet ensemble est trop vaste, on applique une technique de réduction de sa taille : par exemple, on utilise une liste des candidats, ou un extrait au hasard un sous-ensemble des voisins de taille fixe) ; soit $V(s)$ l'ensemble (ou sous-ensemble) de ces voisins ;
- On évalue la fonction objectif pour chaque configuration appartenant à $V(s)$. La configuration t, qui succède à s dans la série des solutions obtenues par le procédé tabu, est la configuration de $V(s)$ dans laquelle f prend la valeur minimale. Notons que cette configuration est adoptée, même si elle est pire que s, c'est à dire si $f(t) > f(s)$: En raison de cette caractéristique de la méthode tabou permet d'éviter les minima locaux.

Tout comme mentionné, il y a un risque important de revenir à une configuration déjà visitée. Pour éviter ce phénomène, il faut mettre à jour la liste des mouvements interdits, la «liste tabou» : cette liste -qui a donné son nom à la méthode- contient des mouvements

$m(t \to s)$, qui sont à l'opposé des dernières mouvements $m(s \to t)$ effectuées. L'algorithme modélise ainsi une forme rudimentaire de la mémoire, la mémoire à court terme des solutions visitées récemment. Deux mécanismes supplémentaires, nommés intensification et diversification, sont souvent mis en oeuvre pour équiper l'algorithme avec une mémoire à long terme. Ce processus n'exploite pas davantage la proximité temporelle des événements particuliers, mais plutôt la fréquence de leur apparition, sur une plus longue période. L'intensification consiste à chercher plus loin dans l'exploration de certaines zones de l'espace des solutions, identifiées comme particulierement prometteuses. La diversification est au contraire la reorientation périodique de la recherche d'un optimum vers les zones, rarement visités jusqu'à présent. Pour certains problèmes d'optimisation, la méthode tabou a donné d'excellents résultats, d'ailleurs, dans sa forme de base, le procédé comprend moins de paramètres d'ajustement que le recuit simulé, ce qui le rend plus facile à utiliser. Cependant, les divers mécanismes supplémentaires, comme l'intensification et la diversification, apportent une complexité remarquable(Dréo et al. 2006).

1.1.5.4 Algorithme de colonie de fourmis

Cette approche, proposée par Colorni, Dorigo et Maniezzo (Colorni et al. 1991), s'attache à simuler la capacité collective à résoudre certains problèmes, comme on l'observe dans une colonie de fourmis, dont les membres sont équipées individuellement avec des facultés très limitées. Leur succès soulève de nombreuses questions. En particulier, les entomologistes ont analysé la collaboration qui s'établit entre les fourmis dans la recherche de nourriture en dehors de la fourmilière. Il est remarquable que les fourmis suivent toujours le même chemin, et ce chemin est le plus court possible. Ce contrôle est le résultat d'un mode de communication indirecte, via l'environnement : le "stigmergie". Chaque fourmi dépose, le long de son chemin, une substance chimique, appelée "phéromone". Tous les membres de la colonie perçoivent cette substance et se dirige préférentiellement vers les zones les plus «odorants». Il en résulte en particulier dans une faculté collective pour trouver rapidement le chemin le plus court, si celui-ci est bloqué par hasard par un obstacle. Bien que ce comportement a été pris comme point de départ pour modéliser l'algorithme, Colorni et al. (1991) ont proposé un nouvel algorithme pour la solution du problème du voyageur de commerce. Depuis ce travail de recherche, la méthode a été étendu à de nombreux problèmes d'optimisation combinatoires ou continus. Les algorithmes de colonies de fourmis ont plusieurs caractéristiques intéressantes ; dont en particulier le parallélisme intrinsèque élevée, la flexibilité (une colonie de fourmis est capable de s'adapter aux modifications de l'environnement), la robustesse (une colonie est prête à maintenir son activité, même si certaines personnes n'y parviennent pas), la décentralisation (une colonie n'obéit pas à une autorité centralisée) et l'auto-organisation (une colonie se trouve une solution, qui n'est pas connu à l'avance). Cette méthode semble particulièrement utile pour les problèmes d'évolution dynamique, qui exigent une forte tolérance aux pannes. A ce stade de développement de ces algorithmes, la transposition à chaque problème d'optimisation n'est cependant pas anodine : elle doit faire l'objet d'un traitement spécifique, qui peut être difficile.

1.1.5.5 Algorithme d'essaim de particules

L'optimisation par essaim de particules (Particle Swarm Optimization : PSO) est l'un des derniers techniques métaheuristiques développées par Kennedy et Eberhart (1995). Dans cet algorithme, les solutions candidates d'une population, appelées des particules, coexistent et évoluent simultanément en se basant sur le partage des connaissances avec les particules voisines. Alors qu'il volait à travers l'espace de recherche, chaque particule génère une solution utilisant son vecteur vitesse. Chaque particule modifie sa vitesse pour trouver une meilleure solution (position) en appliquant sa propre expérience de vol (c-à-d

mémoire ayant la meilleure position trouvée dans les vols précédents) et l'expérience de particules voisines (c-à-d meilleure solution trouvée de la population).
A chaque itération k, la position x_i^k de la particule i est mise à jour par une vitesse v_i^{k+1}, qui dépend de trois facteurs : la vitesse courante v_i^k, les vecteurs de différence pondérés $(p_i^k - x_i^k)$ et $(p_g^k - x_i^k)$. Plus précisément, l'ensemble est mis à jour pour la prochaine itération en utilisant :

$$x_i^{k+1} = x_i^k + v_i^{k+1},$$

avec :

$$v_i^{k+1} = v_i^k + r_1 c_1 (p_i^k - x_i^k) + r_2 c_2 (p_g^k - x_i^k).$$

où x_i^k représente la position actuelle de la particule i dans l'espace des solutions et k indique le numéro de l'itération en cours ; p_i^t est la meilleure position trouvée pour la particule i jusqu'à l'itération k et représente la contribution cognitive de la vitesse de recherche v_i^k. Chaque composante de v_i^k peut être choisie dans un intervalle $[-v_{max}, v_{max}]$ pour contrôler l'itinérance excessive de particules hors de l'espace de recherche ; p_g^k est la meilleure position trouvée parmi toutes les particules dans l'essaim jusqu'à itération k. Les paramètres r_1 et r_2 sont des ,nombres uniformément réparties dans $[0,1]$ et c_1 et c_2, connus sous le nom de paramètres cognitifs et sociaux, respectivement, sont généralement pris égaux à $c_1 = c_2 = 2$ (Clerc 2006).
Dans l'algorithme PSO sans voisinage, toutes les particules sont voisines les unes des autres. Ainsi, la position de la meilleure particule globale de l'essaim est utilisée dans l'expression de l'équation de mise à jour de la vitesse. La convergence dans ce cas est rapide, car toutes les particules sont attirées simultanément à la meilleure région de l'espace de recherche. Toutefois, si l'optimum global n'est pas proche de la meilleure particule, il peut être impossible de l'essaim d'explorer d'autres domaines, ce qui signifie que l'essaim peut être piégé dans un optimum local. Avec la notion de voisinages(voir algorithme 3, seul un nombre spécifique de particules (nombre de voisins) peuvent influer sur la vitesse d'une particule donnée. Les voisinages de l'essaim convergent lentement mais cela permet d'avoir plus de chance de localiser l'optimum global (Clerc 2006).

Algorithme 3 : Algorithme des essaims particulaires avec voisinage

```
for chaque particule do
    Initialisation aléatoire de la particule
end for
repeat
    for chaque voisinage do
        for chaque particule du voisinage do
            Calculer la valeur de la fonction (fitness)
            if fitness < pbest (meilleure valeur rencontrée pour la particule) then
                pbest = fitness
            end if
        end for
        gbest = meilleur pbest de toutes les particules
        for chaque particule do do
            calcul de la vitesse de la particule par l'équation de la vitesse
            mise à jour de la position de la particule x_i^{k+1} = x_i^k + v_i^{k+1}
        end for
    end for
until un des critères de convergence est atteint
```

1.1.6 Optimisation avec Contraintes

Pour la résolution des problèmes d'optimisation avec contraintes, deux approches de base sont généralement utilisées. La première consiste en une projection de la solution en cours sur l'ensemble des solutions réalisables (méthode du gradient projeté). Une deuxième approche commune pour faire face aux problèmes d'optimisation sous contrainte est d'introduire un terme de pénalité dans la fonction objectif pour sanctionner les violations de contraintes (Avriel 2003). L'introduction du terme de pénalité permet de transformer un problème d'optimisation sous contrainte en un problème sans contraintes, tel que celui donné par l'équation :

$$\text{minimize } F_{r,q} = f(x) + r\varphi_q(x)$$

avec pénalité :

$$\varphi_q(x) = \sum_{j=1}^{m} \max\{0;\ g_j(x)\}^q + \sum_{i=1}^{p} \mid h_i(x) \mid^q$$

où $r > 0$ est un paramètre de pénalité positif et q l'exposant de pénalité. Si $q = 1$ on peut prouver que la fonction de pénalité est exacte, c'est à dire pour une valeur suffisamment grande de $r > 0$ tout minimum global de la fonction pénalisé est un minimum global du problème initial. Cependant, la fonction de pénalité exacte n'est pas dérivable en tout point. Pour cette raison, les valeurs $q > 1$, qui garantissent la différentiabilité de la nouvelle fonction objectif, sont également utilisés. Dans ce cas, les solutions dans des conditions modérées convergent vers un minimum global lorsque $r \longrightarrow \infty$ (Avriel 2003, Luenberger 1973). Si r est trop petit, une solution non réalisable peut ne pas être suffisamment pénalisée. Si r est trop grand, une solution réalisable est très susceptible de se produire, mais pourrait être de très mauvaise qualité. Une grande valeur de r décourage l'exploration des régions non faisables même dans les premiers stades de la recherche. Ceci est particulièrement inefficace pour des problèmes où les régions réalisables dans l'espace de recherche sont disjointes. Dans ce cas, il est difficile de se déplacer d'une région à l'autre à moins qu'ils soient très proches les unes des autres. L'exploration raisonnable des régions non faisables peut agir comme ponts reliant deux ou plusieurs régions possibles. La question cruciale ici est dans quelles mesures l'exploration de la région non faisables (c-à-d pour quelle valeur de r) doit être considéré comme raisonnable.

1.2 Optimisation Multi-objectif

1.2.1 Introduction

Les problèmes d'optimisation rencontrés en pratique sont rarement mono objectif ; plusieurs paramètres contradictoires à satisfaire sont présents. L'optimisation multiobjectif est disponible depuis environ trois décennies, et son application dans les problèmes du monde réel ne cesse d'augmenter. Son but est de chercher à optimiser plusieurs composantes de la fonction objectif. La solution du problème n'est pas un vecteur unique, mais un ensemble de solutions connu comme l'ensemble de solutions Pareto optimales. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur une composante du vecteur sans dégradation d'au moins une autre composante. La détermination de l'ensemble Pareto optimale n'est qu'une première phase dans la résolution pratique des problèmes d'optimisation multiobjectif, qui nécessite dans un deuxième temps le choix d'une solution à partir de cet ensemble suivant des préférences choisies par le décideur. Le choix d'une solution par rapport à une autre nécessite la connaissance du problème et des nombreux facteurs liés au problème. Ainsi, une solution choisie par un décideur peut ne pas être acceptable pour un autre. Le choix d'une solution peut aussi être remis en cause dans un environnement dynamique. Il est donc utile

d'avoir plusieurs alternatives dans le choix d'une solution Pareto optimale. Pour résoudre un problème d'optimisation multi-objectif(POM), trois comportements liant le décideur vis-à-vis du solveur peuvent se présenter :

- **Intervention à priori :** Le décideur combine les différentes solutions suivant une certaine fonction d'utilité, et le problème est transformé en un problème d'optimisation mono-objectif. Dans ce cas le décideur est censé connaître le poids de chaque objectif, cependant dans la plupart des cas, la fonction coût n'est pas connue à priori et les différentes composantes du vecteur coût sont exprimées dans des unités différentes.
- **Intervention à posteriori :** Le décideur choisit une solution parmi les solutions de l'ensemble Pareto optimale fourni par le solveur. Il doit pouvoir explorer l'ensemble des solutions en fonction de ses préférences.
- **Intervention interactive :** Il y a coopération progressive entre le décideur et le solveur. A partir des résultats fournis par le solveur pendant la résolution du problème, le décideur fournit des préférences qui seront prises en compte par le solveur. Ce processus est réitéré plusieurs fois, ce qui permettra au décideur de disposer d'une connaissance approfondie pour retenir une solution acceptable.

1.2.2 Concepts de base et terminologie

1.2.2.1 Définition du POM

Un problème d'optimisation multiobjectif (POM) peut être définit comme suit :

$$\left\{ \begin{array}{c} Min\ f(x) = (f_1(x), f_2(x), ..., f_m(x)) \\ sous\ la\ contrainte \\ x \in C \end{array} \right\}$$

ou :
m : Nombre de fonctions objectif, $m \geq 2$.
$x = (x_1, x_2, ..., x_n)$ est le vecteur représentant les m variables de décision.
C : l'ensemble des solutions réalisables associé à des contraintes d'égalité, d'inégalité et des bornes explicites.$C = \{\ x \in \mathbb{R}^n\ /\ h(x) = 0,\ \ g(x) \leq 0\ \ et\ \ x^L \leq x \leq x^U\ \}$
f : Vecteur des critères à optimiser.
$Y = f(C)$: est l'ensemble des points réalisables dans l'espace des critères (espace des objectifs)(Voir figure 1.6).
En optimisation mono-objectif, l'ensemble des solutions réalisables Y est totalement ordonné selon une fonction d'objectif f : soient deux solutions a, b de C soit $f(a) \leq f(b)$ soit $f(b) > f(a)$. Le but est de trouver la solution (ou les solutions) qui donne la valeur minimale à f. Cependant, lorsque plusieurs objectifs sont concernés, la situation change. l'ensemble C est, en général, non totalement ordonné, mais partiellement ordonné. Ceci est illustré dans la figure 1.7. La solution représentée par le point B est meilleure que la solution représentée par le point C : elle fournit une meilleure performance à un coût plus faible. De même pour la solution C qui est meilleure que la solution D à coûts égaux.
Dans le but d'exprimer cette situation mathématiquement, les relations $=, <= et\ <$ sont étendues aux vecteurs d'objectif par analogie au cas des objectifs uniques :

1.2.2.2 Relations

Soient deux vecteurs d'objectifs u et v,

$$u\ =\ v\ \Leftrightarrow\ \forall i \in [1,n]\ :\ u_i = v_i$$

$$u\ \leq\ v\ \Leftrightarrow\ \forall i \in [1,n]\ :\ u_i \leq v_i$$

$$u\ <\ v\ \Leftrightarrow\ \forall i \in [1,n]\ :\ u_i < v_i$$

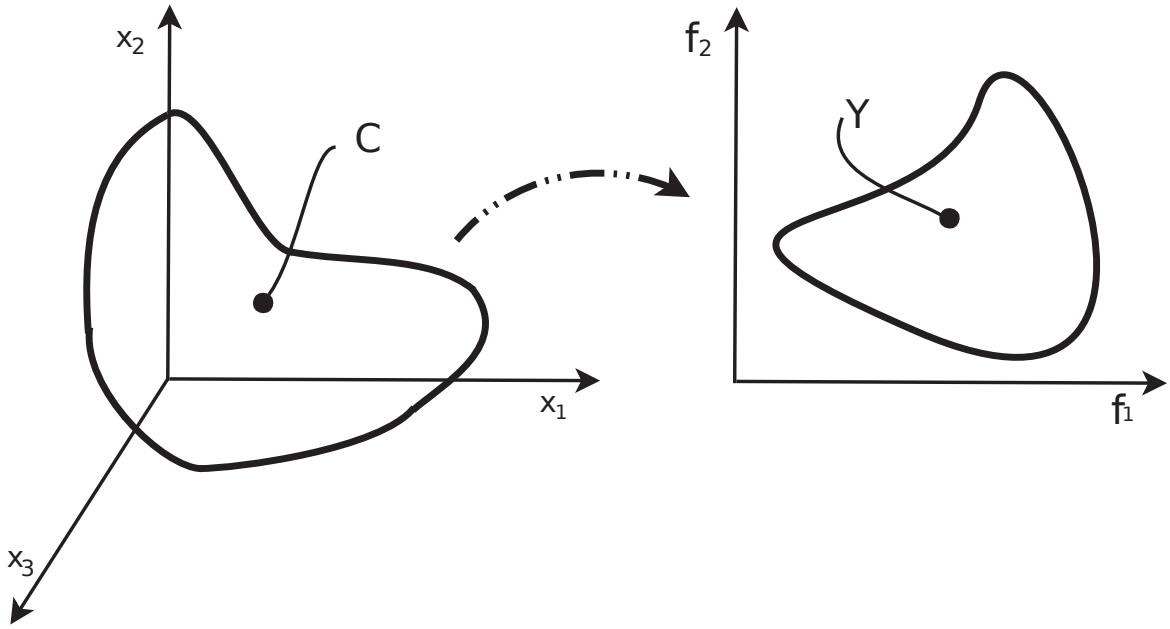


FIGURE 1.6 – *Concepts de base : Espace des variables de décision (à gauche) et espace des objectifs(à droite)*

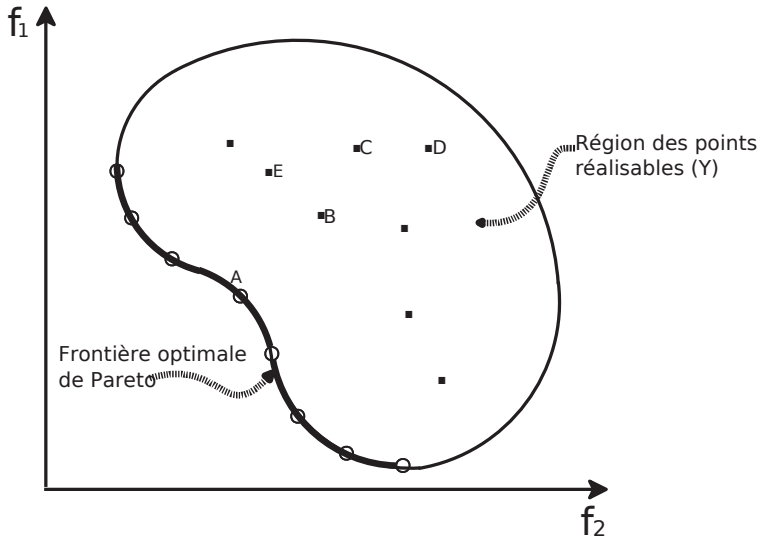


FIGURE 1.7 – *Illustration de l'optimalité de Pareto dans l'espace des objectifs*

Les relations >= et >sont définies de façon similaire.
En utilisant ces notions, il s'avère que $B < C$, $C < D$, et, par conséquent $B < D$. Cependant, lorsque l'on compare les solutions B et E, aucune n'est inférieure à l'autre, puisque $B \not< E$ et $E \not< B$. Bien que la solution associée à B soit moins coûteuse, elle fournit

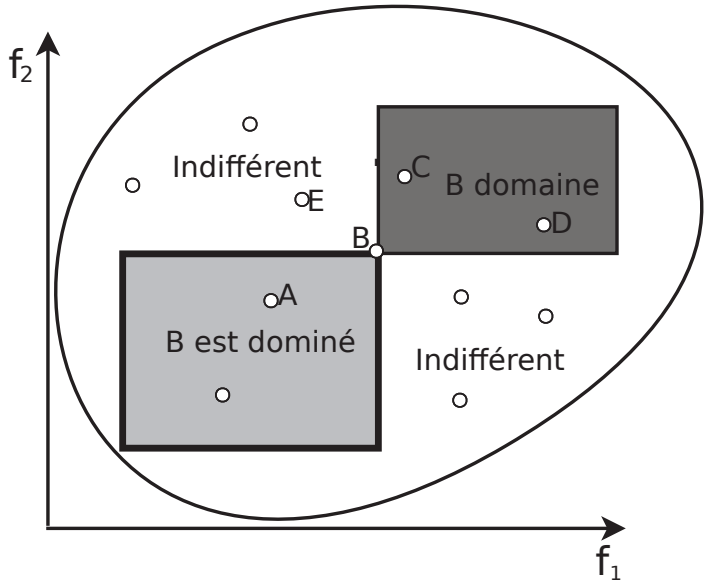


FIGURE 1.8 – *Illustration des relations possibles entre les solutions*

une plus faible performance que la solution représentée par E. Donc, deux vecteurs de décision a et b peuvent avoir trois types de relations en optimisation multi-objectif (par rapport à l'opérateur) : $f(a) \leq f(b), f(b) \leq f(a)$ *ou* $f(a) \nleq f(b) \wedge f(b) \nleq f(a)$

Les symboles et termes suivants sont utilisés dans le but de différencier ces trois situations.

1.2.2.3 Domination de Pareto

Soient deux vecteurs de décision a et b :

$$\text{a} \prec \text{b (a domine b)} \Leftrightarrow \text{f(a)} < \text{f(b)}$$

$$\text{a} \prec= \text{b (a domine faiblement b)} \Leftrightarrow \text{f(a)} \leq \text{f(b)}$$

$$\text{a} \sim \text{b (a est indifférent pour b)} \Leftrightarrow \text{f(a)} \nleq \text{f(b)} \wedge \text{f(b)} \nleq \text{f(a)}$$

Dans la figure 1.8, le rectangle gris clair entoure la région dans l'espace des objectifs qui est dominée par le vecteur de décision représenté par B. Le rectangle gris foncé contient les vecteurs d'objectifs qui correspondent aux vecteurs de décision dominant la solution associée à B. Toutes les solutions dont le vecteur de décision correspondant est ni dans le rectangle clair, ni dans le rectangle foncé, sont indifférentes pour la solution représentée par B.
Basé sur le concept de domination de Pareto, les critères d'optimalité pour les POM peuvent être introduits. Dans les figures 1.7 et 1.8, A domine les points B, C, D et E : son vecteur de décision a n'est dominé par aucun autre vecteur de décision. Cela signifie que a est optimal dans le sens où il ne peut être amélioré sur aucun objectif sans causer la dégradation d'au moins un autre objectif. De telles solutions sont appelées solutions optimales de Pareto ; parfois le terme de solutions non-inférieures est utilisé.

1.2.2.4 Optimalité de Pareto

Un vecteur de décision x de C est dit non dominé par rapport à un ensemble A de C ssi :

$$\nexists a \in A \ : \ a \prec x$$

Dans la figure 1.7, les points blancs représentent les solutions optimales de Pareto. Elles sont indifférentes les une des autres. Cela constitue la principale différence avec les problèmes à objectif unique (POU) : il n'y a pas une seule solution optimale mais plutôt un ensemble de compromis optimaux. Aucun d'eux ne peut être identifié comme étant meilleur que les autres sans qu'une information de préférence ne soit incluse.

L'ensemble des solutions optimales de Pareto est appelé ensemble optimal de Pareto ; les vecteurs d'objectifs correspondants constituent la frontière (ou front) de Pareto ou plus généralement la surface de Pareto.

1.2.2.5 Ensemble optimal local et global

Considérons un ensemble de vecteurs de décision A de C.

1. L'ensemble A est dénommé ensemble optimal local de Pareto ssi :

$$\forall a \in A : \nexists x \in C : x \prec a \wedge \|x-a\| < \varepsilon \wedge \|f(x)-f(a)\| < \delta$$

   où $\|.\|$ est une métrique de distance et $\varepsilon > 0$, $\delta > 0$.
2. L'ensemble A est dénommé ensemble optimal global de Pareto ssi :

$$\forall a \in A : \nexists x \in C : x \prec a$$

La différence entre optima locaux et globaux est représentée dans la figure 1.9. La ligne en pointillés constitue la frontière optimale globale de Pareto, tandis que la ligne continue représente une frontière optimale locale de Pareto. Les vecteurs de décision associés à la dernière sont localement non dominés bien que non Pareto-optimaux, parce que la solution relative au point A domine chacun d'eux. Finalement, notons qu'un ensemble optimal global de Pareto ne contient pas nécessairement toutes les solutions optimales de Pareto et que tout ensemble optimal global de Pareto est aussi un ensemble optimal local de Pareto.

1.2.3 Classification des méthodes

Les approches pour la résolution des problèmes d'optimisation multiobjectif (POM) peuvent être classées en trois catégories.

- **Méthodes Agrégées :** Ces approches basées sur la transformation du problème multiobjectif en un problème uniobjectif. Cette classe d'approches comprend par exemple les méthodes basées sur l'agrégation qui combine entre les différentes fonctions objectif f_i en une seule fonction objectif f. Ces approches nécessitent d'avoir une bonne connaissance du problème.
- **Approches non Pareto :** Elles ne transforment pas les POM en un problème uni objectif, mais elles utilisent des opérations de recherche qui traitent séparément les différents objectifs.
- **Approches Pareto :** Elles utilisent directement la notion d'optimalité Pareto dans leur processus de recherche. Le processus de sélection des solutions générées est basé sur la notion de non dominance.

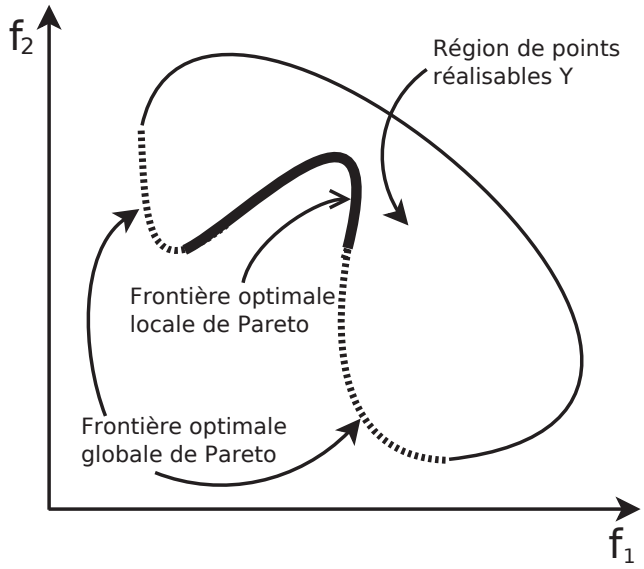


FIGURE 1.9 – *Illustration des ensembles de solutions localement et globalement optimales dans l'espace des objectifs.*

1.2.4 Méthodes Agrégées

Plusieurs méthodes traditionnelles transforment le POM en un problème uni objectif (PUO) , elles reposent sur l'optimisation d'une fonction d'utilité $U = U(f_1, f_2, ..., f_n)$. Parmi ces méthodes, on trouve principalement les méthodes d'agrégation, les méthodes ε-contrainte, et les méthodes de programmation par but (goal programming).

1.2.4.1 Méthodes d'Agrégation

Ces méthodes sont parmi les premières méthodes utilisées pour la génération de solutions Pareto optimale. Le principe de base consiste à transformer le POM en un problème PUO, POM_λ en additionnant tous les objectif après avoir affecter à chacun d'eux un coefficient de poids. Ces coefficients représentent l'importance relative que le décideur attribue à l'objectif, et on obtient une seule fonction objectif f telle que :

$$f(x) = \sum_{i=1}^{n} \lambda_i f_i(x)$$

où les poids vérifient : $\lambda_i \in [0..1]$ et $\sum_1^n \lambda_i = 1$; Différents poids fournissent différentes solutions supportées. La même solution peut être générée en utilisant des poids différents. Les résultats obtenus dans la résolution du problème dépendent fortement des paramètres choisis en fonction des préférences associées aux objectifs. Ainsi, une approche généralement utilisée consiste à résoudre le problème POM_λ avec différentes valeurs de λ. La figure 1.10 illustre le fonctionnement de la méthode d'agrégation linéaire. Fixer un vecteur poids revient à trouver un hyper-plan dans l'espace objectif (une ligne pour un problème bi-critère) avec une orientation fixée. La solution Pareto optimale est le point où l'hyper-plan possède une tangente commune avec l'espace réalisable (point x dans la figure 1.10).
Si les fonctions objectif sont exprimées en des unités de mesure différents, on peut transformer l'équation précédente sous la forme :

$$f(x) = \sum c_i \lambda_i f_i(x)$$

où les c_i sont les constantes qui mettent à la même échelle les différents fonctions objectif, généralement les constantes c_i sont initialisées à $\frac{1}{f_i^*}$, où les f_i^* est la solution optimale associée à la fonction objectif f_i(à déterminer au préalable). L'inconvenient majeur de cette méthode est qu'elle ne garantie pas d'obtenir la solution Pareto pour les espaces non convexes (Marler et Arora 2004).

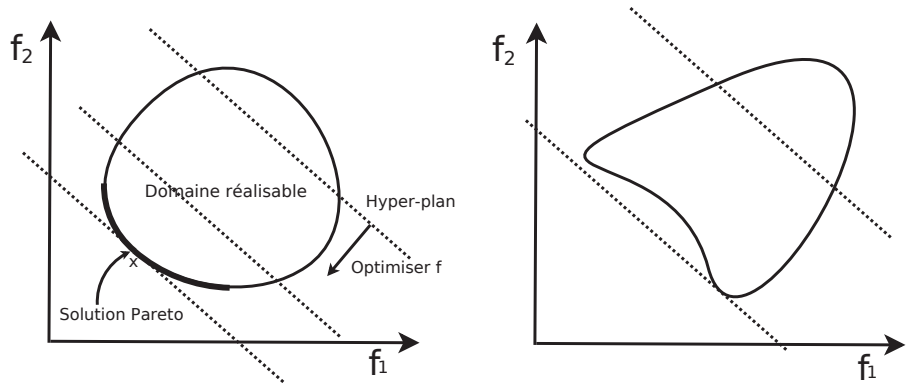


FIGURE 1.10 – *La méthode d'agrégation*

1.2.4.2 Méthode ε-contrainte

Une autre technique, dite méthode ε-contrainte, consiste à transformer les (n-1) objectif en contraintes. L'objectif restant, qui peut être choisi arbitrairement, est la fonction d'objectif du problème d'optimisation mono-objectif qui en résulte, en d'autre termes dans cette approche, le problème consiste à optimiser une fonction f_k sujette à des contraintes sur les autres fonctions :

$$(POM_k(\varepsilon)) \left\{ \begin{array}{c} Min\, f_k(x) \\ sous\ les\ contraintes \\ x \in C \\ et\ \ f_i(x) \leq \varepsilon_i,\, i = 1,..,n,\, i \neq k \end{array} \right\}$$

Différentes valeurs de ε_i peuvent être données pour pouvoir générer différentes solutions Pareto Optimales. La connaissance à priori des intervalles appropriés pour les valeurs ε_i est requise pour tous les objectifs. Pour pouvoir définir les valeurs adéquates pour ε_i, le vecteur idéal doit être calculé pour déterminer les bornes inférieurs $\varepsilon_i \geq f_i(x^*)$, $\forall i = 1,2,...k-1,k+1,...n$. La représentation géométrique suivante (voir figure 1.11 montre une représentation à deux dimensions de la méthode ε-contrainte pour un problème à deux objectif.

$$\left\{ \begin{array}{c} Min\ f_1(x) \\ sous\ lescontraintes \\ x \in C \\ et\ f_2(x) \leq \varepsilon_2 \end{array} \right\}$$

Grâce à cette méthode, on peut résoudre des problèmes convexes et non convexes. L'inconvénient est que l'on doit connaître à priori les intervalles pour les valeurs de ε_i pour toutes les fonctions objectif(Bérubé et al. 2009).

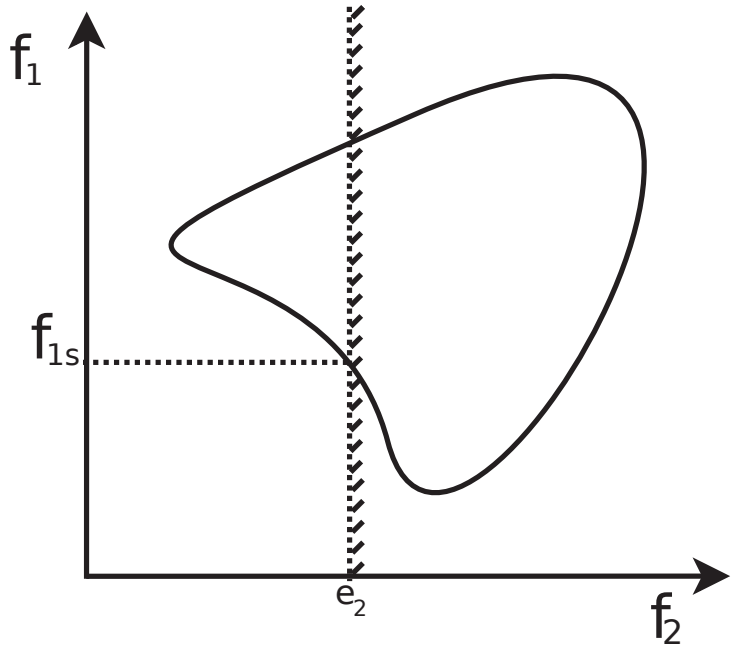


FIGURE 1.11 – *Représentation géométrique de la méthode ε-Contrainte*

1.2.4.3 Programmation par but

Dans cette méthode, le décideur doit définir les buts ou références qu'il désire atteindre pour chaque objectif. Ces valeurs sont introduites dans la formulation du problème qui se transforme en un problème uni-objectif. La fonction coût peut être rédigée sous la forme suivante :

$$POM_{(\lambda,z)} \left\{ \begin{array}{c} \min\left(\sum_{i=1}^{n} \lambda_i (f_i(x) - z_i)^p\right)^{\frac{1}{p}} \\ sous\ les\ contraintes \\ x \in C \end{array} \right\}$$

avec $p \in [1, \infty[$, généralement p=2 et z est la valeur de référence choisie par le décideur ; dans ce cas, on cherche à minimiser les dérivations par rapports aux buts. Dans le cas ou $p = \infty$, l'équation revient à une fonction min-max (Talbi 1999). Le choix du vecteur but joue un rôle important dans la résolution du problème, un mauvais choix du vecteur de référence peut aboutir à une solution qui n'est pas Pareto optimale. Les approches basées sur la programmation par but aboutissent à une solution non dominée si le but est choisi dans le domaine réalisable. Cependant, le décideur est chargé de choisir le vecteur but et les poids associés à chaque objectif, ce qui est une tâche difficile sans connaître la structure de l'espace de recherche. Si le domaine réalisable n'est pas facile à approcher, la méthode peut être inefficace. Une autre approche dite " Goal atainment " est basée sur la spécification des préférences par l'intermédiaire d'un but à atteindre. Dans cette approche, le décideur choisit un vecteur de poids w et les buts désirés z pour tous les objectifs(Rosenthal 1985).

1.2.4.4 Discussion sur les méthodes d'agrégation

L'agrégation de multiples objectifs en un seul critère d'optimisation possède l'avantage que les stratégies classiques d'optimisation à objectif unique peuvent être appliquées sans plus de modifications. Cependant, cela requière une grande connaissance du domaine qui n'est souvent pas disponible. Effectuer la recherche avant la prise de décision supprime cet inconvénient, mais exclut l'articulation des préférences par le preneur de décision qui doit réduire la complexité de l'espace de recherche.

1.2.5 Méthodes Non Agrégées et Non Pareto

En général, les méthodes dites non agrégées et non Pareto possèdent un processus de recherche qui traite séparément les objectifs. Parmi ces méthodes on trouve principalement les méthodes VEGA et la méthode lexicographique.

1.2.5.1 Méthode VEGA

La méthode VEGA(Vector Evaluated Genetic Algorithm), proposé par David Schaffer (Schaffer 1985), est une extension d'un algorithme génétique avec la seule différence la manière dont s'effectue la sélection. Le principe est le suivant (voir figure 1.12) : Si nous avons n objectifs et une population de m individus, une sélection de m/n individus est effectuée pour chaque objectif. Ainsi n sous-populations vont être créées, chacune d'entre elles contenant les m/n meilleurs individus pour un objectif particulier. Les n sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille m. Le processus se termine par l'application des opérateurs génétiques de modification (croisement et mutation).

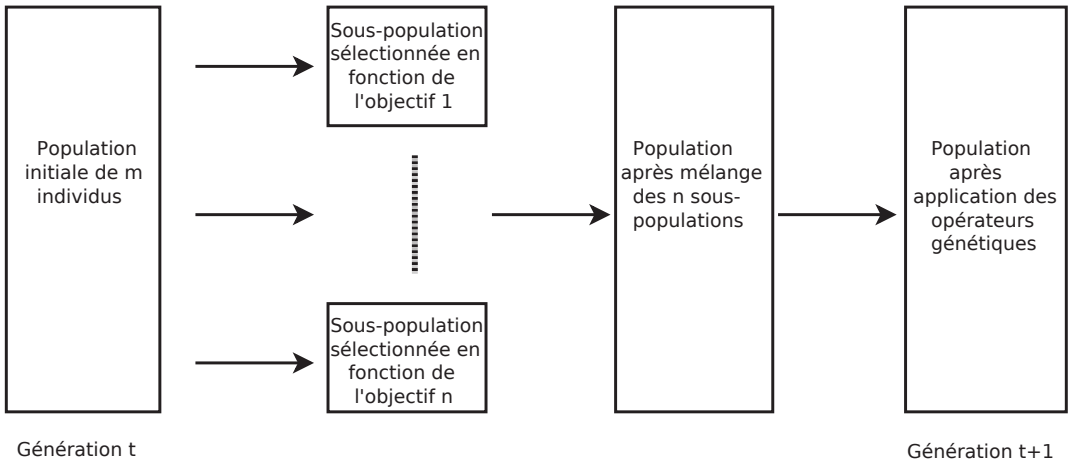


FIGURE 1.12 – *Principe de fonctionnement de VEGA*

Les sous populations utilisées par cette méthode génèrent des résultats favorables à un objectif particulier. En effet, comme la méthode de sélection ne tient compte que d'un seul objectif, elle privilégie les individus qui obtiennent une bonne performance pour cet objectif ; et donc ces individus ne seront sélectionnés que lorsqu'on effectuera la sélection sur cet objectif. Malgré que la méthode VEGA soit efficace et facile à mettre en oeuvre, son inconvenient majeure est que si la sélection proportionnelle est utilisée, la fusion de tous les sous-populations correspond à la moyenne des composantes de remise en

forme associées à chacun des objectifs. En d'autres termes, dans ces conditions, VEGA se comporte comme une approche d'agrégation et donc, elle est soumise aux mêmes inconvénients de ces techniques (Coello et al. 2007).

1.2.5.2 La Méthode lexicographique

Dans cette méthode, proposé par Fourman (1985), les fonctions objectif sont d'abord rangés par ordre d'importance. Ensuite, on optimise dans cet ordre les fonctions objectif les unes après les autres. La formulation mathématique du problème est la suivante : Soient n fonctions objectif $f_i,\ avec\ i = 1,..n$, les fonctions objectif sont rangées suivant le choix du décideur dans cet ordre : $f_1, f_2, f_3, ..., f_n$.
On résout le problème suivant :

$$\left\{ \begin{array}{c} Min\, f_1(x) \\ sous\ les\ contraintes \\ x \in C \end{array} \right\}$$

soit x_1^* la meilleure solution trouvée avec $f_1^* = f_1(x_1^*), f_1^*$ devient alors une nouvelle contrainte, et l'expression du nouveau problème est donc :

$$\left\{ \begin{array}{c} Min\, f_2(x) \\ sous\ les\ contraintes \\ x \in C \\ et\ f_1(x) = f_1^* \end{array} \right\}$$

soit x_2^* la meilleure solution trouvée avec $f_2^* = f_2(x_2^*)$, l'expression du problème sera le suivant :

$$\left\{ \begin{array}{c} Min\, f_i(x) \\ sous\ les\ contraintes \\ x \in C \\ et\ f_1(x) = f_1^*, f_2(x) = f_2^*, ..., f_{i-1}(x) = f_{i-1}^* \end{array} \right\}$$

La procédure est répétée jusqu'à ce que tous les fonctions objectif soient traités. La solution obtenue à la dernière étape sera la solution du problème. Une autre variante de la méthode lexicographique consiste à choisir aléatoirement la fonction objectif devant être prise en compte. Cette façon de procéder équivaut à une somme pondérée dans laquelle un poids correspond à la probabilité que la fonction objectif associée soit sélectionnée. L'inconvénient essentiel de cette méthode est l'importance donnée aux objectifs classés en premier. La solution f_i^* trouvée pour l'objectif le plus important va influencer l'algorithme et le diriger vers une zone restreinte de l'espace faisable(Ehrgott et Gandibleux 2000).

1.2.6 Méthodes Pareto

L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg en 1989 (Goldberg 1989) pour résoudre les problèmes multiobjectif. Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée d'objectifs contradictoires. L'utilisation d'une sélection basée sur la notion de dominance de Pareto va faire converger la population vers un ensemble de solutions efficaces. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais il apporte une aide précieuse au décideur. Plusieurs techniques utilisant ce concept sont proposées dans la littérature.

1.2.6.1 Méthode MOGA(Multiple Objective Genetic Algorithm)

La méthode MOGA, Proposée par Fonesca et Fleming (Da Fonseca 1995) est une variante de la technique de Goldberg, dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent. On utilise une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant le même rang. Considérons, par exemple, un individu x_i à la génération t, dominé par $p_i(t)$ individus de la génération en cours, il se voit attribuer le rang rang(x_i, t) = 1 +p_i(t).
Tous les individus non dominés sont affectés du rang 1, tandis que les dominés sont pénalisés en fonction de la densité de la population de la région correspondante de la surface de compromis. L'évaluation de la fitness de chaque individu se fait de la façon suivante :

1. Tri de la population selon le rang.
2. Attribuer la fitness des individus par interpolation du meilleur (rang 1) au pire (rang n $\leq N$) dans le sens proposé par David E. Goldberg (Goldberg 1989) selon une certaine fonction, généralement linéaire.
3. faire la moyenne des fitnesses des individus ayant le même rang, de sorte que chacun d'entre eux soit échantillonné à la même vitesse. Ceci permet de maintenir la fitness global de la population, tout en maintenant constante la pression de sélection appropriée.

Le principal avantage de MOGA est qu'il est efficace et relativement facile à mettre en oeuvre. La principale critique envers MOGA a été qu'il effectue le partage de l'espace des objectifs, ce qui implique que deux vecteurs différents avec les mêmes valeurs de la fonction objectif ne peuvent exister simultanément dans la population. Sa principale faiblesse est que, comme toutes les autres techniques de classement de Pareto, sa performance dépend fortement d'une sélection appropriée du facteur de partage(sharing). Cependant, il est important de noter que Fonseca et Fleming (Fonseca et al. 1993) ont développé une bonne méthodologie pour calculer cette valeur.

1.2.6.2 Méthode NPGA(Niched Pareto Genetic Algorithm)

Dans cette méthode proposé par Jeffrey Horn et ses collègues (rey Horn et al. 1993, Horn et al. 1994) deux individus choisis au hasard sont comparés à un sous-ensemble de l'ensemble de la population (en général, environ 10% de la population). Si l'un d'eux est dominé (par les individus choisis au hasard dans la population) et l'autre non, alors il est positionné dans la population suivante. Dans les autres cas une fonction de sharing est appliquée pour sélectionner l'individu. Le réglage de la taille de la sous population permet d'exercer une pression variable sur la population et ainsi d'augmenter ou de diminuer la convergence de l'algorithme. Le choix de cette taille a un effet majeur sur la vitesse de convergence, une valeur de l'ordre de 10% de N(Nombre d'individus de la population) donne une bonne distribution des individus. Cette valeur apporte un bon compromis entre élitisme et représentativité statistique de la population. Elle permet non seulement d'exercer une pression suffisante sur la sélection des individus, mais également d'éviter une convergence non désirée due à une sélection effectuée sur un échantillon non représentatif de la population. Cette méthode nécessite la maîtrise de deux paramètres : paramètres de sharing et la taille de sous population. Les solutions trouvées sont de bonne précision et cette approche est plus rapide que l'approche précédente car le sharing n'est appliqué que sur une portion de la population.

1.2.6.3 Méthode SPEA(Strength Pareto Evolutionary Algorithm)

L'algorithme SPEA a été introduit par Eckart et Lothar Thiele Zitzler (Zitzler et Thiele 1999). Cette approche a été conçue comme un moyen d'intégrer différents algorithmes évolutionnaires d'optimisation multi-objectif. SPEA utilise une archive externe contenant des solutions non dominées précédemment trouvées (ce qu'on appelle le jeu externe non dominé)(Voir Algorithme 4). A chaque génération, les individus non dominés sont copiés dans cette archive. Pour chaque individu dans cette série externe, une valeur de résistance est calculée. Cette force est similaire à la valeur de classement de MOGA (Fonseca et al. 1993), car elle est proportionnelle au nombre de solutions à un certain individu qui domine. Dans la SPEA, l'aptitude de chaque membre de la population actuelle est calculée en fonction des points forts de toutes les solutions externes non dominées qui la dominent. Le processus d'affectation de remise en forme de SPEA considère à la fois la proximité du front de Pareto et la répartition homogène de solutions. Ainsi, au lieu d'utiliser des créneaux en fonction de la distance, la dominance de Pareto est utilisée pour s'assurer que les solutions sont bien répartis le long du front de Pareto. Bien que cette approche ne nécessite pas un rayon de niche, son efficacité repose sur la taille de l'ensemble non dominé. En fait, du fait que le poste externe non dominée participe au processus de sélection de la SPEA, si sa taille devient trop importante, elle peut réduire la pression de sélection, ce qui ralentit la recherche. Pour cette raison, les auteurs ont décidé d'adopter une technique qui élague le contenu de la consigne externe non dominée afin que sa taille reste inférieure à un certain seuil. L'approche adoptée pour cette cause était une technique de clustering appelée méthode de liaison moyenne(average linkage method) (Morse 1980). Une version révisée de la SPEA (appelée SPEA2)(Voir algorithme 5) a été introduite avec trois différences principales par rapport à son prédécesseur (Zitzler et al. 2001, Giannakoglou et de Métodos Numéricos en Ingeniería 2002) (Coello et al. 2007) :

1. il intègre une stratégie qui tient compte pour chaque individu le nombre d'individus qui le dominent et le nombre d'individus qu'il domine ;
2. il utilise une technique d'estimation de la densité du voisin le plus proche qui permet de guider la recherche de manière plus efficace,
3. il a une méthode de troncature d'archivage améliorée qui garantit la préservation de solutions aux limites.

Algorithme 4 : Algorithme SPEA

Initialiser la population P_0
Créer l'archive externe vide $\bar{P} = \emptyset$
repeat
  Calcul de la valeur d'adaptation pour tous les individus de $P \bigcup \bar{P}$
  Selection dans $P_t \bigcup \bar{P}$ en fonction de la valeur d'adaptation
  Opération de croisement
  Opération de mutation
  Mise à jour de $\bar{P}$ à partir des individus non dominés de P
until critère d'arrêt réalisé

Algorithme 5 : Algorithme SPEA2

Initialiser la population P_0
Créer l'archive externe vide $\bar{P} = \emptyset$
repeat
  Calcul de la valeur de chaque individu de $P et \bar{P}$
  Copie des individus non dominés de $P et \bar{P}$ dans $\bar{P}$
  if La taille de $\bar{P}$ dépasse la limite **then**
    Réduire $\bar{P}$ par l'opérateur de troncature
  else
    Compléter avec les individus dominés de P
  end if
  Selection par tournoi binaire avec remplacement pour remplir la population intermédiaire
  Appliquer les opérateurs de croisement et de mutation à la population intermédiaire
until critère d'arrêt réalisé

1.2.6.4 Méthode PAES(Pareto Archived Evolution Strategy)

La méthode PAES, conçue et mise en oeuvre par D. Knowles et Joshua David W. Corne (Knowles et Corne 2000) est une approche assez simple, comme indiqué dans l'algorithme 6. Son principe n'est pas basée sur une population car elle n'utilise qu'un seul individu à la fois pour la recherche des solutions. Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales. Cette archive est utilisée comme un ensemble de références à laquelle chaque individu muté est comparé. PAES utilise également une nouvelle approche pour maintenir la diversité, qui consiste en une procédure de regroupement qui divise l'espace objectif de manière récursive. En outre, la procédure a une complexité de calcul plus faible que les méthodes traditionnelles de niching.

Algorithme 7 : Fonction test(c,m,archive)

if l'archive n'est pas plein **then**
  Ajout de m à l'archive.
else if m est dans une région moins encombrée qu'une solution x de l'archive **then**
  Ajout de m et suppression d'un membre de la zone la plus encombrée de l'archive.
end if
if m est dans une région moins encombrée que c **then**
  m est acceptée comme solution courante.
else
  On conserve c comme solution courante.
end if

Algorithme 6 : Algorithme PAES

Génération aléatoire d'une solution *c* et ajout de *c* à l'archive
repeat
  Production d'une solution *m* par mutation de *c* et évaluation de *m*
  if *c* domine *m* **then**
    On écarte *m*
  else if *m* domine *c* **then**
    On remplace *c* par *m* et ajout de *m* à l'archive
  else if *m* est dominé par un membre de l'archive **then**
    On écarte *m*
  else
    On applique une fonction de test(*c*, *m*, archive) qui détermine la nouvelle solution courante
  end if
until critère d'arrêt réalisé

1.2.6.5 Méthode NSGA(Non dominated Sorting Genetic Algorithm)

N. Srinivas and Kalyanmoy Deb (Srinivas et Deb 1994) ont proposé une autre variante de l'approche de Goldberg appelée «Non dominated Sorting Genetic Algorithm» (NSGA). Le pseudo-code de cette méthode est donnée dans l'algorithme 8. Cet algorithme est basé sur plusieurs couches de classifications des individus. Afin de maintenir la diversité de la population, ces individus classées sont partagés avec leurs valeurs nominales de fitness. Ensuite, ce groupe d'individus classés est ignoré et une autre couche d'individus non dominées est considérée. Le processus se poursuit jusqu'à ce que tous les individus de la population sont classés. La NSGA a connu un succès pendant plusieurs années (Weile et al. 1996), bien que plusieurs études comparatives de l'époque (Van Veldhuizen 1999) a indiqué qu'il a été devancé par MOGA et NPGA. NSGA est également un algorithme inefficace en raison de la façon dont il classe les individus. Deb et al. (2000; 2002) ont proposé une version améliorée de l'algorithme NSGA, appelée NSGA-II. Elle introduit la notion de rang de non-domination pour évaluer la force de chaque individu. Des procédures d'élitisme, visant à conserver les meilleures solutions, de génération en génération, contribuent à accélérer la vitesse de convergence et les performances générales des algorithmes génétiques. La diversité est également garantie à l'aide d'une méthode de "crowding" ne nécessitant pas de paramètre supplémentaire.

Algorithme 8 : Pseudo code de NSGA

Génération de la population initiale
repeat
  while Population n'est pas classée **do**
    Recherche des individus non dominés
    Calcul fitness factice (artificielle)
    Sharing
  end while
  Sélection
  Croisement
  mutation
until Critère d'arrêt réalisé

1.2.6.6 Méthode NBI(Normal Boundary Intersection)

1.2.6.6.1 Introduction : En réponse à la défaillance de la méthode de la somme pondérée, Das et Dennis (1998) ont proposé la méthode Normal Boundary Intersection (NBI) pour la génération d'un ensemble uniformément réparti de solutions Pareto optimales pour les problèmes d'optimisation multiobjectif non linéaires. L'approche par la méthode des sommes pondérées a un inconvénient majeur de ne pas pouvoir en général générer un ensemble étendu de solution Pareto optimales même si on utilise des coefficients de pondération répartis uniformément. L'approche NBI est facilement implémentable pour des problèmes multiobjectif pour un nombre d'objectif supérieur à 2. Elle a l'avantage d'être applicable pour des objectifs non-commensurables (effet d'echelle).

1.2.6.6.2 Notations et définitions Soit x_i^* la solution du problème de minimisation de la fonction $f_i(x)$ $i = 1,...,n\, pour\, x \in C$.
Soit $F_i^* = f(x_i^*),\ soit\ \phi$ la matrice d'ordre (n,n) et dont la $i^e me$ colonne est $F_i^* - f^*$, avec $f^* = (f_1^*, f_2^*, f_3^*, ..., f_n^*)^T$: Vecteur des solutions optimales de chaque objectif. ϕ s'écrit sous la forme suivante :

$$\phi = \begin{pmatrix} 0 & f_1(x_2^*) - f_1(x_1^*) & . & . & . & f_1(x_n^*) - f_1(x_1^*) \\ f_2(x_1^*) - f_2(x_2^*) & 0 & . & . & . & f_2(x_n^*) - f_2(x_2^*) \\ . & & & & & . \\ . & & & . & & . \\ . & & & & . & . \\ f_n(x_1^*) - f_n(x_n^*) & & & & & 0 \end{pmatrix}$$

Dans ces conditions, l'ensemble des point de R^n, combinaison convexe de $F_i^* - f^*$, s'écrit : $\left\{ \phi\beta \ : \beta \in R^n, \sum_{i=1}^{n} \beta_i = 1 ; \beta_i \geq 0 \right\}$ est appelé Enveloppe Convexe des Individus Minimum, ECIM (en anglais Convex Hull Individual Minima CHIM)(voir figure 1.13.
Soit $f = \{ f(x)/x \in C \}$ l'ensemble des solutions réalisables de $\mathbb{F}$(espace des objectifs réalisables), et notons par Γ la frontière de $\mathbb{F}$. Désormais, on adoptera un changement de variables $f(x) - f^*$.

1.2.6.6.3 Principe de base de la méthode NBI : NBI est une technique qui cherche la portion d'espace qui contient les point Pareto optimals. L'idée algébrique derrière cette approche est une idée simple et évidente : le point d'intersection entre la frontière Γ et la normal pointant vers l'origine émanant de tout point de la ECIM est un point de la portion de Γ qui contient les points Pareto optimales. Cela revient à chercher la distance maximale entre un point du ECIM et la frontière Γ en pointant vers l'origine. Les points obtenus peuvent être ou non des points Pareto optimal. La figure 1.14 montre les solutions obtenues (a, b, c, d, e, f, g, h) correspondant aux points équidistants (A, B, C, D, E, F, G, H) du ECIM (ligne AH). Il est clair que les points d, e, et f sont non Pareto optimal malgré qu'ils ont été trouvés par la méthode NBI.

Dans le cas de situations non convexes, même les points Pareto optimal qui ne peuvent pas être obtenus par la méthode traditionnelle de la somme pondérée peuvent être trouvés par cette méthode, c'est le cas du point x dans la figure 1.14. Pour un coordonné barycentrique β, $\phi\beta$ représente un point du CHIM. Notons par $\vec{n}$ le vecteur unitaire normal au CHIM pointant vers l'origine. Dans ces conditions $\phi\beta + t\vec{n}$, $t \in \Re$ représente l'ensemble des points sur cette nomal. Le point d'intersection de la normal $\vec{n}$ et de la limite Γ la plus proche de l'origine est la solution global du nouveau POM

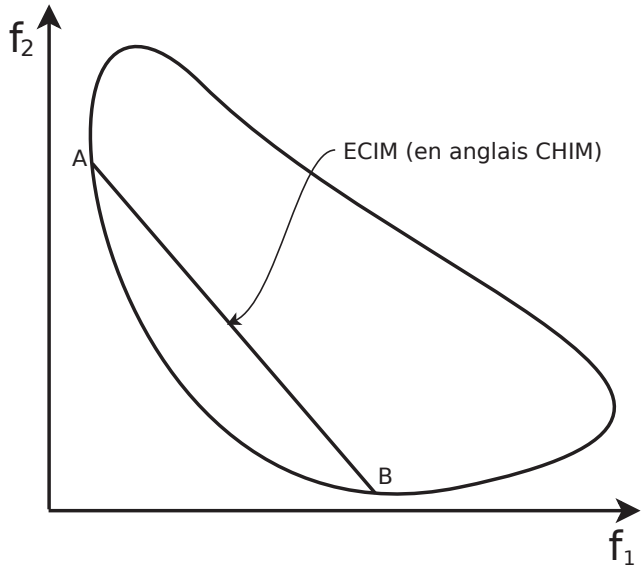


FIGURE 1.13 – *Principe de base de la NBI : CHIM*

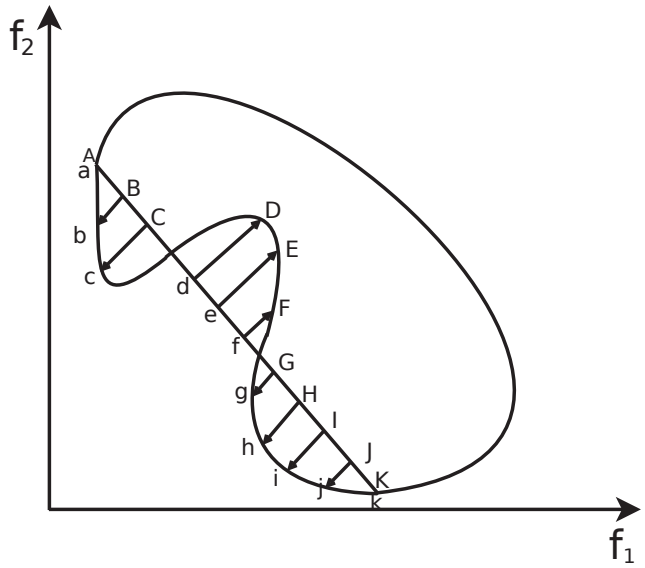


FIGURE 1.14 – *Principe de base de la NBI*

suivant :

$$POM \left\{ \begin{array}{c} \max\limits_{x,t} t \\ Sous\ les\ contraintes \\ \phi\beta + t\,\vec{n} = f(x) \\ et\ x \in C \end{array} \right\}$$

Il faut noter que si l'origine n'est pas prise en f^*, la première contrainte doit s'écrire sous la forme $\phi\beta + t\,\vec{n} = f(x) - f^*$. On obtient pour un problème de minimisation une autre

formulation noté NBI(β)

$$NBI(\beta) \ : \ \begin{pmatrix} \underset{x,t}{Min} - t \\ s/c \ \phi\beta + t\,\vec{n} = f(x) - f^* \\ et \ x \in C \end{pmatrix}$$

L'idée est de résoudre NBI(β) pour différentes valeurs de β et trouver les différents points de la limite Γ et ainsi construire efficacement une approximation de la frontière efficiente. Le vecteur $\vec{n}$ peut être choisi quasi-normal avec des composantes négatives (pointe vers l'origine), les résultats obtenus seront les mêmes. Dans la pratique, on choisit une direction quasi-normale comme combinaison linéaire des colonnes de ϕ, multiplié par -1 pour assurer la direction vers l'origine :

$$\vec{n} = -\phi\, e \ , \ e = (1,1,...,n)^T$$

Ce choix offre l'avantage d'avoir des points NBI, pour un β donné, complètement indépendants de l'échelle des fonctions objectif.

1.2.6.6.4 Génération du vecteur β : Soit $\delta > 0$ un pas uniforme entre deux valeurs consécutifs de β_i, $i = 1,...,n$.
Pour simplifier, on prend p= $\frac{1}{\delta}$ un entier donné, les valeurs possibles qui peuvent être prises par β_1 *sont* $[0, \delta, 2\delta, ..., 1]$.
Pour β_i donné, on définie $m_1 = \frac{\beta_1}{\delta}$, alors les valeurs possibles de β_2 sont $[0, \delta, 2\delta, ..., k_2\delta]$ *avec* $k_2 = \frac{1-\beta_1}{\delta} = \frac{1-m_1\delta}{\delta} = p - m_1$.
On définit de même $m_2 = \frac{\beta_2}{\delta}$, alors les valeurs possibles de β_3 sont $[0, \delta, 2\delta, ..., k_3\delta]$ *avec* $k_3 = \frac{1-\beta_1-\beta_2}{\delta} = \frac{1-m_1\delta-m_2\delta}{\delta} = p - m_1 - m_2$.
Et par récurrence, pour $\beta_j, j = 2,...,n-1$, il prend les valeurs parmi $[0, \delta, 2\delta, ..., k_j\delta]$ *avec* $k_j = p - \sum_{i=1}^{j-1} m_i$.
Finalement : $\beta_n = 1 - \sum_{i=1}^{n-1} \beta_i$. Prenons l'exemple pour n=4 et $\delta = 0.2$, le schéma de la figure 1.15 présente une partie de l'arborescence de β.

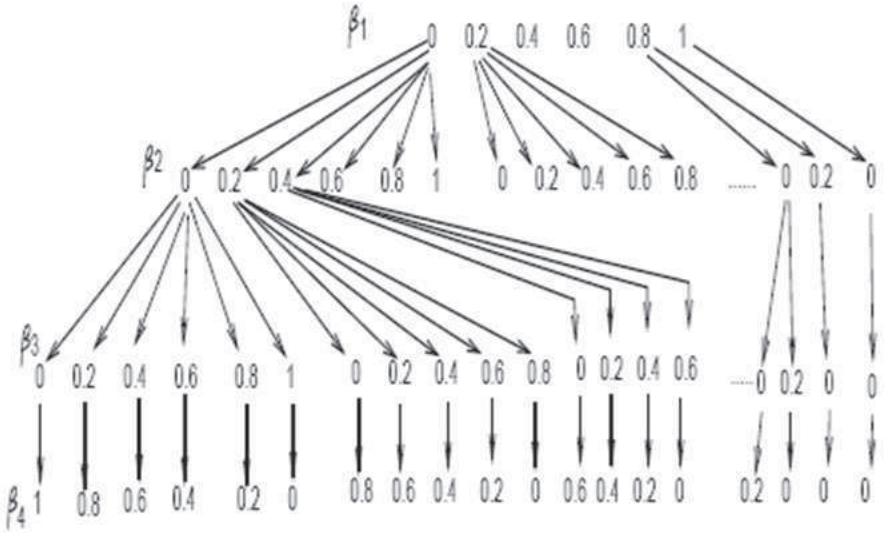


FIGURE 1.15 – *Génération du vecteur beta*

1.2.6.6.5 Avantages de la méthode NBI : Parmi les avantages de cette méthode, on peut citer :

1. elle génère un ensemble uniforme de points Pareto représentatifs de l'ensemble des points 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 ;
2. la méthode est applicable pour des problèmes de plus de deux fonctions objectif ;
3. elle n'exige pas un calcul du hessien exact comme c'est le cas pour d'autres méthodes ;
4. chaque point trouvé sert de point de départ pour la recherche du point suivant ;
5. le problème d'échelle des fonctions objectif n'est pas posé ;
6. les techniques de traitement parallèle peuvent être utilisées dans l'implémentation de la méthode.

1.2.6.7 Méthode NNC(Normalized Normal Constraint)

La méthode de la contrainte normale normalisée (NNC) est une amélioration de la méthode NBI avec contraintes d'inégalité. Elle utilise une procedure de normalisation des fonctions objectif et un filtre de Pareto pour éliminer les points de non Pareto. Le procédé peut fournir un ensemble de points Pareto optimales régulièrement espacés dans l'espace de solutions. Sa performance est indépendante des échelles des fonctions objectif (Messac et al. 2003). La méthode NNC a été mise au point par Messac et Ismail-Yahaya pour améliorer la méthode originale NC qui ne génèrait pas uniquement les solutions Pareto (Messac et al. 2003). Elle est relativement facile à mettre en oeuvre, et peut générer une distribution uniforme de points de Pareto le long de la frontière de Pareto. Elle est insensible à l'ampleur des fonctions objectif en normalisant celles-ci, et elle est valable pour un nombre arbitraire de fonctions objectif. La NNC, comme la méthode NBI, fonctionne en transformant le problème d'optimisation multi-objectif en un ensemble de sous-problèmes d'optimisation non linéaire. Chaque sous-problème fait l'objet d'une réduction de l'espace réalisable en introduisant une contrainte d'inégalité supplémentaire. Soit $F^* = (f_1^*, f_2^*, ..., f_\ell^*)^T$ le point idéal du problème d'optimisation multi-objectif avec ℓ fonctions objectif et n variables. Le minimum individuel des fonctions objectif atteint en x_i^* pour tout $i = 1, ..., \ell$. Le points $F(x_i^*)$ pour tout $i = 1, 2, ..., \ell$ sont appelés les points d'ancrage(anchor point)(voir figure **4.3**. La méthode NNC utilise les valeurs normalisées des fonction objectif, notés $\bar{f}$ pour faire face à des échelles hétérogènes des fonctions. Il peut être obtenu en utilisant les points utopie (f_i^u) et les points pseudo-nadir (f_i^{Np}) comme suit :

$$\bar{f}_i = \frac{f_i - f_i^u}{f_i^{Np} - f_i^u}, \quad pour \;\; i \in \{1, ..., n\}$$

où

$$f_i^u = f_i(x_i^*) \;\; and \;\; f_i^{Np} = max\{f_i(x_1^*), ..., f_i(x_n^*)\}$$

On note par V_k (pour $k = 1, 2, ..., \ell - 1$) le vecteur de l'hyper-plan utopique générés par des points d'ancrage ($V_k = f_n^* - f_k^*$), en faisant varier systématiquement $\beta \;\; (= (\beta_1, ..., \beta_\ell)^T | . \sum_{i=1}^{\ell} \beta_i = 1)$ notons Z_β le point sur hyper-plan idéal corespondent au vecteur β.
Le sous-problème NNC, notée NNC_β, pour β donné, est comme suit :

$$\left\{ \begin{array}{rl} & \min_{(x)} \quad \bar{f}_\ell(x) \\ \text{sujet à} & g_i(\mathbf{x}) \leq 0 \;,\; i = 1, ..., m, \\ & h_j(\mathbf{x}) = 0 \;,\; j = 1, ..., p, \\ & V_k^T(\bar{F} - Z_\beta) \leq 0 \;\; pour \;\; 1 \leq k \leq \ell - 1 \end{array} \right. \tag{1.2}$$

La procédure peut être représentée dans la figure 4.3 pour le cas bidimensionnel. En raison de la production de solutions non Pareto, une procédure de filtrage est utilisée pour les éliminer.

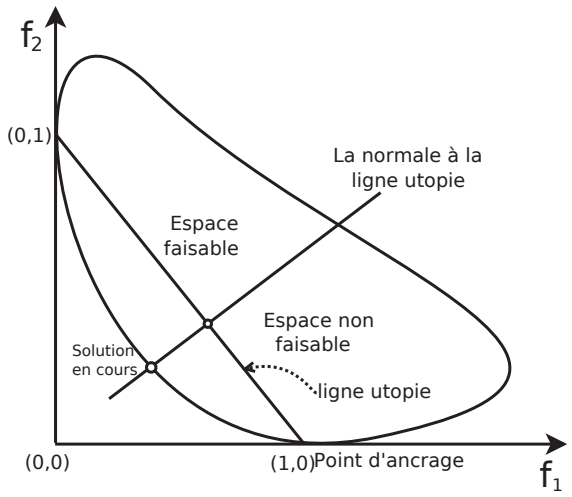


FIGURE 1.16 – *La méthode NNC pour le cas de problème à deux dimensions.*

1.3 CONCLUSION

Dans ce chapitre, nous avons passé en revue plusieurs méthodes d'optimisation mono-objectif et multi-objectif en accordant une importance particulière aux méthodes utilisées dans nos travaux de recherche ; notamment la méthode de Nelder Mead, l'algorithme génétique, la méthode d'optimisation par essaim des particules, ainsi que les méthodes d'optimisation multi-objectif NBI, NNC, SPEA, NSGA. Pour les autres méthodes, une description sommaire a été faite en faisant référence à la documentation pour tout besoin d'informations ou de détails supplémentaires.

CHAPITRE 2

REPRESENTATION DE SOLUTION : CONTRIBUTION À L'OPTIMISATION GLOBALE

SOMMAIRE

3.1 INTRODUCTION 57
3.2 CONCEPTION D'UN PILIER 57
3.3 CONCEPTION DE LA STRUCTURE DE TREILLIS 61
3.4 EFFICACITÉ ET EFFICIENCE DE RFNM 65
3.4.1 Influence de la fonction de Pincus 65
3.4.2 Comparaison avec GA, PSO et NSGA II 66
3.5 CONCLUSION 68

DANS ce chapitre, nous proposons une nouvelle approche fondée sur une formule de représentation pour résoudre les problèmes d'optimisation globale. On considère le problème de minimisation d'une fonction donnée : $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ sur un ensemble S non vide, régulier et fermé. Quand f atteint un minimum global en un point $x^* \in S$, nous avons :

$$x^* = \lim_{\lambda \to +\infty} \frac{E\left(X\, g\left(\lambda, f(X)\right)\right)}{E\left(g\left(\lambda, f(X)\right)\right)}$$

pour une variable aléatoire X convenable et une fonction $g : \mathbb{R}^2 \longrightarrow \mathbb{R}$ bien définie.

Nous proposons d'utiliser la formule de représentation (RF) pour générer numériquement une population initiale.

Pour obtenir un résultat plus précis, la formule de représentation est couplée avec les algorithmes suivants :

- l'algorithme génétique classique(GA). On obtient un nouveau algorithme appelé (RFGA),
- l'algorithme génétique, avec utilisation de l'algorithme de Nelder Mead, au niveau de la mutation (GANM). On obtient un nouveau algorithme qu'on appelle (RFGANM),

- l'algorithme de Nelder Mead. On obtient un nouveau algorithm, qu'on appelle (RFNM).

Pour évaluer les performances des six algorithmes obtenus (RF, GA, RFGA, GANM, RFGANM,RFNM), on propose de les faire tester sur 21 fonctions tests (benchmark) avec une analyse complète de l'effet de plusieurs paramètres sur les différentes méthodes.
Le chapitre est organisé comme suit : Dans la section 2.2, la représentation de solution est introduite. La section 2.3 décrit l'hybridation de cette représentation avec l'algorithme génétique et la méthode du simplexe. Enfin, les résultats numériques pour les fonctions d'essai, présentées dans la section 2.4, sont exposés dans la section 2.5, suivise d'une conclusion à la section 2.6.

La principale contribution en rapport avec ce chapitre est :

H. Zidani, R. Ellaia, and J.E. Souza de Cursi. **"A Hybrid Simplex Search For Global Optimization With Representation Formula And Genetic Algorithm"**, *soumis à Applied Mathematics and Computation, (ISSN :* 0096 – 3003*), 2012.*

2.1 INTRODUCTION

Dans le cadre de la résolution des problèmes de l'ingénierie, plusieurs algorithmes d'optimisation ont été proposés, examinés et analysés dans les dernières décennies. Cependant, l'optimisation dans le domaine de l'ingénierie demeure un champ actif de recherches, puisque beaucoup de problèmes réels d'optimisation restent très complexes en réalité et difficiles à résoudre par les algorithmes existants.

La littérature existante présente des efforts intensifs de recherches pour résoudre certaines difficultés, pour lesquels seulement des réponses partielles ont été obtenues. Parmi ces derniers, nous pouvons citer : la manipulation de la non convexité - particulièrement quand les contraintes d'optimisation sont impliqués, manipulation de fonctions ou contraintes avec des évaluations incomplètes ou incorrectes, augmentation du nombre de variables d'optimisation au delà de ceux des conceptions réalistes dans la pratique, traitement des fonctions non régulières (discontinues ou non différentiables) et l'identification des points de départ commodes pour les méthodes itératives (Floudas et al. 2009) (Weise 2009) (Floudas et Gounaris 2008).

Nous observons que les difficultés liées au sujet de la non-convexité et du choix des points de départ sont liés : les méthodes efficaces pour l'optimisation des fonctions régulières sont souvent déterministes et utilisent des gradients, mais dépendent fortement du point initial - elles peuvent être emprisonnés par des minimum locaux si un choix initial non commode est employé. Alternativement, les méthodes basés sur l'exploration de l'espace des variables de conception impliquent, habituellement, un aspect stochastique - ainsi, une croissance significative en coût de traitement - et sont moins dépendantes du choix initial, mais les améliorations de leur performance nécessitent la combinaison avec des méthodes déterministes qui peuvent présenter une dépendance à l'égard du choix initial. Cette dernière approche tend au recours à des méthodes hybrides mettant en jeu les deux approches et essaie de tirer bénéfice du meilleur de chaque méthode. C'est pour cette raison que la littérature traitant les méthodes stochastiques/déterministes hybrides s'est développée depuis la fin du XXeme siècle (Lee et El-Sharkawi 2008)(Chelouah et Siarry 2003)(Fan et Zahara 2007)(Yen 1998)(Chelouah et Siarry 2005)(Fan et al. 2004)(Renders et Flasse 1996)(Fan et al. 2006).

Ces algorithmes hybrides fonctionnent mieux si le point initial appartient à une zone d'attraction proche de l'optimum. Ceci montre l'importance de l'estimation du point initial pour les algorithmes d'optimisation (Ivorra et al. 2011). Par conséquent, nous propo-

sons, dans ce chapitre, d'employer une formule de représentation pour fournir une valeur initiale commode de la solution.

Soit S un domaine régulier, borné et fermé de l'espace euclidian $\mathbb{R}^n$, et soit f une fonction continue définie sur S et qui prend ses valeurs dans $\mathbb{R}$. Un problème d'optimisation sans contrainte peut être formulé, d'une manière générale, comme suit :

$$x^* = Arg \min_{x \in S} f(x) \ , \tag{2.1}$$

Dans la littérature, les formules de représentation ont été présentées afin de caractériser d'une manière explicite les solutions du problème 2.1. Généralement ces représentations supposent que S contient un seul point optimal $\mathbf{x}^*$ (mais beaucoup de minimum locaux peuvent exister dans S). Par exemple, Pincus (1968; 1970) a proposé la formule de représentation suivante :

$$x^\star = \lim_{\lambda \to +\infty} \frac{\int_S x\, e^{-\lambda f(x)} dx}{\int_S e^{-\lambda f(x)} dx} \ .$$

Une extension aux minimum multiples a été introduit par Charnes et Wolfe (1989) et une approche basé sur la bornitude d'une séquence d'intégrales a été proposé par Falk Falk (1973). Plus récemment, la représentation originale proposé par Pincus a été reformulé par Souza de Cursi (Souza de Cursi et al. 2007) comme suit :
Soit X une variable aléatoire prenant ses valeurs sur S et soit une fonction $g : \mathbb{R}^2 \longrightarrow \mathbb{R}$. Si ces éléments sont convenablement choisis, on a :

$$x^* = \lim_{\lambda \to +\infty} \frac{E\left(X\, g\left(\lambda, f(X)\right)\right)}{E\left(g\left(\lambda, f(X)\right)\right)} \tag{2.2}$$

La formulation de Pincus correspond à $g(\lambda, s) = e^{-\lambda s}$, ce qui est un choix commode. Les propriétés de X et de g sont détaillés, par exemple, dans (Bez et Gon¸calves 2004), (Bez et al. 2010), (Bez et al. 2005) et (Souza de Cursi et El Hami 2011). Une extension aux dimensions infinies peut être trouvée dans (Souza de Cursi et El Hami 2011) (Souza de Cursi et al. 2007).

Dans ce chapitre, nous proposons l'utilisation de la formule de représentation donnée par Eq.(2.2) hybridée avec l'algorithme de Nelder Mead et l'algorithme génétique, pour l'optimisation globale de fonctions multimodales.

Les algorithmes génétiques et de Nelder Mead sont bien connus pour leur popularité et facilité d'utilisation et n'ont pas besoin de dérivation de la fonction objectif. Cependant, malgré qu'elle soit une méthode globale, un inconvénient principal de la première méthode est sa convergence prématurée au minimum local ayant pour conséquence une faible précision tandis que le second est très sensible au choix des points initiaux et n'assure pas d'atteindre l'optimum global. Par conséquent, il serait intéressant d'étudier la possibilité d'hybrider les deux méthodes en utilisant la formule de Pincus pour produire les points de départ.

2.2 FORMULE DE REPRÉSENTATION

2.2.1 Introduction

Les formules de représentation sont des méthodes qui caractérisent de manière explicite le point optimum à partir du calcul intégral. Un résultat utilisant la représentation intégrale a été donné par Falk (1973). Supposons que la fonction f soit strictement positive sur S, pour $n \in \mathbb{N}$, on définit la suite suivante :

$$r_n(x^*) = \int_S \left[\frac{f(x)}{f(x^*)}\right]^n dx. \tag{I.5}$$

Alors x^* est un minimum global de f sur S si et seulement si la suite $\{r_n(x^*)\}_{n\in\mathbb{N}}$ est bornée.

Une caractérisation du point minimal a été proposée par Pincus (1968). Si x^* est l'unique minimum global de la fonction f supposée continue et partout finie, alors

$$x^* = \lim_{\lambda\to\infty} \frac{\int_S xe^{-\lambda f(x)}dx}{\int_S e^{-\lambda f(x)}dx}. \qquad (I.4)$$

Une caractérisation similaire du point minimum global d'une fonction f supposée partout négative sur un ensemble S a été donnée par Ivanov (1985). Si x^* est l'unique minimum de f sur S, alors on a :

$$x^* = \lim_{\gamma\to\infty} \frac{\int_S x \mid f(x) \mid^{\gamma} dx}{\int_S \mid f(x) \mid^{\gamma} dx}. \qquad (I.3)$$

Bien que ces caractérisations semblent trés séduisantes du point de vue théorique, numériquement il est très difficile de décrire des algorithmes et d'implémenter des codes, en se servant de ces représentations.

2.2.2 Hypothèses et notations

On considére un ensemble $S \subset \mathbb{R}^n$, fermé et borné, (les résultats peuvent s'étendre pour le cas ou S n'est pas borné), On ne suppose pas S convexe.
Soit $f : \mathbb{R}^n \to \mathbb{R}$ une application continue, On ne suppose pas f convexe. on note :

$$x^* = Arg\min_S f,$$

cad,

$$x^* \in S, \quad \forall x \in S, \quad f(x^*) \leq f(x),$$

On suppose que x^* est unique, c-à-d, que la région S contient un seul minimum global. Soit $\varepsilon > 0$, un nombre réel destiné à tendre vers zéro. On note B^*_ε la boule ouverte de centre x^* et de rayon ε :

$$B^*_\varepsilon = \{x \in S \ \mid \ |x - x^*| < \varepsilon\}$$

et $S^*_\varepsilon = S - B^*_\varepsilon$ son complémentaire par rapport à S :

$$S^*_\varepsilon = \{x \in S \ \mid \ |x - x^*| \geq \varepsilon\}$$

Dans ce qui suit, on note : $\chi^*_\varepsilon(x) = \begin{cases} 1 & x \in B^*_\varepsilon \\ 0 & x \notin B^*_\varepsilon \end{cases}$; $\psi^*_\varepsilon(x) = \begin{cases} 1 & x \in S^*_\varepsilon \\ 0 & x \notin S^*_\varepsilon \end{cases}$
On a : $\forall x \in S \quad : \quad \chi^*_\varepsilon(x) + \psi^*_\varepsilon(x) = 1.$
Soit P une probabilité sur S. On suppose que :

$$\forall \eta > 0 \ : \ P(B^*_\eta) > 0.$$

Pour la variable aléatoire $Y, E(Y)$ dénote son espérance mathématique :

$$E(Y) = \int_S Y(x)P(dx).$$

Soit $\lambda > 0$ un nombre réel destiné à tendre vers l'infini et $g : \mathbb{R}^2 \to \mathbb{R}$ une application continue telle que :$g \geq 0$
Compte tenu des hypothèses précédentes, on a :

$$0 \leq E(g(\lambda, f(x))) < \infty.$$

On suppose, de plus, qu'il existe un nombre réel $\varepsilon_0 > 0$ et deux fonctions $h_1, h_2 : \mathbb{R}^2 \to \mathbb{R}$ telles que :

$$\forall \varepsilon \in (0, \varepsilon_0) : E(g(\lambda, f(x))) \geq h_1(\lambda, \varepsilon) > 0 \quad ; \quad E(\psi_\varepsilon^*(x) g(\lambda, f(x))) \leq h_2(\lambda, \varepsilon)$$

et

$$\forall \varepsilon \in (0, \varepsilon_0) : \frac{h_2(\lambda, \varepsilon)}{h_1(\lambda, \varepsilon)} \underset{\lambda \to \infty}{\to} 0.$$

2.2.3 Le résultat central

On va établir le résultat central suivant :

Theorem 2.2.1 *Sous les hypothèses précédentes, on a :*

$$x^* = \lim_{\lambda \to \infty} \frac{E(xg(\lambda, f(x)))}{E(g(\lambda, f(x)))}$$

.

Preuve :

1. Nous avons :
$$\frac{E(xg(\lambda, f(x)))}{E(g(\lambda, f(x)))} = \frac{E(x^* g(\lambda, f(x)))}{E(g(\lambda, f(x)))} + \frac{E((x - x^*) g(\lambda, f(x)))}{E(g(\lambda, f(x)))}$$
Ainsi
$$\frac{E(xg(\lambda, f(x)))}{E(g(\lambda, f(x)))} = x^* + \frac{E((x - x^*) g(\lambda, f(x)))}{E(g(\lambda, f(x)))}$$
2. Soit $\varepsilon \in (0, \varepsilon_0)$. Puisque,
$$E((x - x^*) g(\lambda, f(x))) = E((x - x^*)(\chi_\varepsilon^*(x) + \psi_\varepsilon^*(x)) g(\lambda, f(x))),$$
nous avons :
$$E((x - x^*) g(\lambda, f(x))) = E((x - x^*) \chi_\varepsilon^*(x) g(\lambda, f(x)) + E((x - x^*) \psi_\varepsilon^*(x) g(\lambda, f(x)))$$
3. Or, d'une part :
$$|E((x - x^*) \chi_\varepsilon^*(x) g(\lambda, f(x)))| \leq E(|x - x^*| \chi_\varepsilon^*(x) g(\lambda, f(x))) \leq \varepsilon M$$
4. Et d'autre part :
$$E(\psi_\varepsilon^*(x) g(\lambda, f(x))) \leq h_2(\lambda, \varepsilon),$$
de façon que :
$$|E((x - x^*) g(\lambda, f(x)))| \leq \varepsilon E(\chi_\varepsilon^*(x) g(\lambda, f(x))) + h_2(\lambda, \varepsilon).$$
5. Ainsi,
$$\left|\frac{E((x - x^*) g(\lambda, f(x)))}{E(g(\lambda, f(x)))}\right| \leq \frac{\varepsilon E(\chi_\varepsilon^*(x) g(\lambda, f(x))) + h_2(\lambda, \varepsilon)}{E(g(\lambda, f(x)))} = \frac{\varepsilon E(\chi_\varepsilon^*(x) g(\lambda, f(x)))}{E(g(\lambda, f(x)))} + \frac{h_2(\lambda, \varepsilon)}{E(g(\lambda, f(x)))}.$$
Puisque :
$$E(\chi_\varepsilon^*(x) g(\lambda, f(x))) \leq 1$$
et
$$\frac{h_2(\lambda, \varepsilon)}{E(g(\lambda, f(x)))} \leq \frac{h_2(\lambda, \varepsilon)}{h_1(\lambda, \varepsilon)},$$
il vient :
$$\left|\frac{E((x - x^*) g(\lambda, f(x)))}{E(g(\lambda, f(x)))}\right| \leq \varepsilon + \frac{h_2(\lambda, \varepsilon)}{h_1(\lambda, \varepsilon)}.$$

6. Ainsi,

$$\forall \varepsilon \in (0,\varepsilon_0) : \lim_{\lambda\to+\infty} |\frac{E((x-x^*)g(\lambda,f(x)))}{E(g(\lambda,f(x)))}| \leq \varepsilon + \lim_{\lambda\to+\infty} \frac{h_2(\lambda,\varepsilon)}{h_1(\lambda,\varepsilon)} = \varepsilon.$$

Donc,

$$\forall \varepsilon \in (0,\varepsilon_0) : \lim_{\lambda\to+\infty} |\frac{E((x-x^*)g(\lambda,f(x)))}{E(g(\lambda,f(x)))}| \leq \varepsilon.$$

7. Il en résulte que :

$$\lim_{\lambda\to+\infty} |\frac{E((x-x^*)g(\lambda,f(x)))}{E(g(\lambda,f(x)))}| = 0$$

et par conséquent,

$$\lim_{\lambda\to+\infty} \frac{E((x-x^*)g(\lambda,f(x)))}{E(g(\lambda,f(x)))} = 0$$

8. Ainsi,

$$\lim_{\lambda\to+\infty} \frac{E(xg(\lambda,f(x)))}{E(g(\lambda,f(x)))} = x^* + \lim_{\lambda\to+\infty} \frac{E((x-x^*)g(\lambda,f(x)))}{E(g(\lambda,f(x)))} = x^*$$

et on a le résultat énoncé.

2.2.4 Choix de la fonction

Le choix de la fonction g peut être orienté par le résultat suivant, applicable aux situations où x^* est le seul point de minimum global dans S, c-à-d, où la région S ne contient qu'un seul point de minimum global.

Proposition 2.2.2 *Soit $g : \mathbb{R}^2 \to \mathbb{R}$, une application continue telle que $g > 0$ et $\xi \to g(\lambda,\xi)$ est strictement décroissante pour tout $\lambda > 0$. Si x^* est le seul point de minimum global dans S, alors les hypothèses sur h_1 et h_2 sont vérifiées.*

Preuve :

1. La décroissance de g entraîne que :

$$\forall x \in S : g(\lambda,f(x)) \leq g(\lambda,f(x^*)),$$

2. Pour ε_0 suffisamment petit et $\varepsilon \in (0,\varepsilon_0)$:
$\exists\ \theta = \theta(\varepsilon) > 0$ tel que $\max_{S_\varepsilon^*} g(\lambda,f(x)) \leq g(\lambda,f(x^*))\exp(-\lambda\theta)$.
En effet, si :

$$\max_{S_\varepsilon^*} g(\lambda,f(x)) = g(\lambda,f(x^*))$$

alors, il existe une suite $\{x_n\}_{n\geq 0} \subset S_\varepsilon^*$ telle que :

$$g(\lambda,f(x_n)) \underset{n\to+\infty}{\to} g(\lambda,f(x^*)).$$

Puisque S_ε^* est fermé et borné, cette suite admet un point d'accumulation $\overline{x} \in S_\varepsilon^*$ tel que $g(\lambda,f(\overline{x})) = g(\lambda,f(x^*))$ Puisque $\xi \to g(\lambda,\xi)$ est strictement décroissante, nous avons $f(\overline{x}) = f(x^*)$, ce qui est en contradiction avec l'unicité de x^*.

3. Ainsi :

$$\forall x \in S_\varepsilon^* : g(\lambda,f(x)) \leq g(\lambda,f(x^*))\exp(-\lambda\theta)$$

et nous avons :

$$E(\psi_\varepsilon^*(x)g(\lambda,f(x))) \leq g(\lambda,f(x^*))\exp(-\lambda\theta)E(\psi_\varepsilon^*(x)).$$

On pose :

$$h_2(\lambda,\varepsilon) = exp(-\lambda\theta(\varepsilon))exp(-\lambda f(x^*))P(S_\varepsilon^*).$$

Alors :

$$E(\psi_\varepsilon^*(x)g(\lambda,f(x))) \leq h_2(\lambda,\varepsilon).$$

4. Soit $\delta > 0$ un nombre réel tel que $\delta < \theta$. La continuité de f entraîne l'existance de $\eta = \eta(\delta) > 0$ tel que :

$$\max_{B_\eta^*} \; g(\lambda,f(x)) \geq g(\lambda,f(x^*))\exp(-\lambda\delta).$$

Ainsi :

$$\forall x \in B_\eta^* : g(\lambda,f(x)) \geq g(\lambda,f(x^*))\exp(-\lambda\delta)$$

et

$$E(g(\lambda,f(x))) \geq E(\chi_\eta^*(x)g(\lambda,f(x))) \geq g(\lambda,f(x^*))\exp(-\lambda\delta)E(\chi_\eta^*(x)).$$

Donc, en posant :

$$h_1(\lambda,\varepsilon) = g(\lambda,f(x^*))\exp(-\lambda\delta)P(B_\eta^*),$$

il vient :

$$E(g(\lambda,f(x))) \geq h_1(\lambda,\varepsilon) > 0$$

5. Enfin,

$$\forall\varepsilon \in (0,\varepsilon_0) \quad : \quad \frac{h_2(\lambda,\varepsilon)}{h_1(\lambda,\varepsilon)} = \frac{g(\lambda,f(x^*))\exp(-\lambda\theta)P(S_\varepsilon^*)}{g(\lambda,f(x^*))\exp(-\lambda\delta)P(B_\eta^*)} = \frac{P(S_\varepsilon^*)}{P(B_\eta^*)}\exp(-\lambda(\theta-\delta)) \underset{\lambda\to+\infty}{\to} 0.$$

Remark 2.2.3 *Un choix possible est $g(\lambda,f(x)) = \exp(-\lambda f(x))$. ce choix a été suggéré par Pincus (1968). La preuve peut être adapté à ce choix de la façon suivante :*

1. Pour ε_0 suffisamment petit et $\varepsilon \in (0,\varepsilon_0)$:
$\exists \;\; \theta = \theta(\varepsilon) > 0$ tel que $\min\limits_{S_\varepsilon^*} \; |f(x) - f(x^*)| \geq \theta > 0.$
En effet, si :

$$\min_{S_\varepsilon^*} \; |f(x) - f(x^*)| = 0$$

alors, il existe une suite $\{x_n\}_{n\geq 0} \subset S_\varepsilon^*$ telle que :

$$f(x) \underset{n\to+\infty}{\to} f(x^*).$$

Puisque S_ε^* est fermé et borné, cette suite admet un point d'accumulation $\overline{x} \in S_\varepsilon^*$ tel que $f(\overline{x})) = f(x^*)$, ce qui est en contradiction avec l'unicité de x^*.

2. Donc :

$$\forall x \in S_\varepsilon^* : \exp(-\lambda(f(x) - f(x^*))) = \exp(-\lambda|f(x) - f(x^*)|) \leq exp(-\lambda\theta)$$

et nous avons :

$$\exp(-\lambda,f(x)) = \exp(-\lambda f(x))\exp(-\lambda(f(x) - f(x^*))) \leq \exp(-\lambda\theta)\exp(-\lambda f(x^*)).$$

Par conséquent :

$$E(\psi_\varepsilon^*(x)g(\lambda,f(x))) \leq \exp(-\lambda\theta)\exp(-\lambda f(x^*))E(\psi_\varepsilon^*(x)).$$

On pose :

$$h_2(\lambda,\varepsilon) = exp(-\lambda\theta(\varepsilon))exp(-\lambda f(x^*))P(S_\varepsilon^*).$$

Alors :

$$E(\psi_\varepsilon^*(x)g(\lambda,f(x))) \leq h_2(\lambda,\varepsilon).$$

3. Soit $\delta > 0$ un nombre réel tel que $\delta < \theta$. La continuité de f entraîne l'existance de $\eta = \eta(\delta) > 0$ tel que :

$$\max_{B_\eta^*} |f(x) - f(x^*)| \leq \delta.$$

Ainsi :

$$\forall x \in B_\eta^* : \exp(-\lambda(f(x) - f(x^*))) = \exp(-\lambda|f(x) - f(x^*)|) \geq \exp(-\lambda\delta)$$

et

$$\exp(-\lambda f(x)) = \exp(-\lambda f(x^*))\exp(-\lambda(f(x) - f(x^*))) \geq \exp(-\lambda\delta)\exp(-\lambda f(x^*)).$$

Ainsi,

$$E(g(\lambda, f(x))) \geq E(\chi_\eta^*(\lambda, f(x))) \geq \exp(-\lambda\delta)\exp(-\lambda f(x^*))E(\psi_\eta^*(x)).$$

en posant :

$$h_1(\lambda, \varepsilon) = \exp(-\lambda f(x^*))\exp(-\lambda\delta)P(B_\eta^*),$$

il vient :

$$E(g(\lambda, f(x))) \geq h_1(\lambda, \varepsilon) > 0$$

4. Enfin,

$$\forall \varepsilon \in (0, \varepsilon_0) \quad : \quad \frac{h_2(\lambda, \varepsilon)}{h_1(\lambda, \varepsilon)} = \frac{\exp(-\lambda f(x^*))\exp(-\lambda\theta)P(S_\varepsilon^*)}{\exp(-\lambda f(x^*))\exp(-\lambda\delta)P(B_\eta^*)} = \frac{P(S_\varepsilon^*)}{P(B_\eta^*)}\exp(-\lambda(\theta - \delta)) \underset{\lambda \to +\infty}{\to} 0.$$

2.3 Hybridation de la méthode du simplexe avec la formule de représentation et l'algorithme génétique

2.3.1 Introduction

Des méthodes hybrides ont été présentées pour garder la flexibilité des méthodes stochastiques et l'efficacité des algorithmes déterministes. La méthode hybride que nous proposons pour résoudre des problèmes d'optimisation globale est un couplage de la formule de représentation proposée par Pincus (1968), l'algorithme de Nelder Mead(Simplexe) et l'algorithme génétique. Dans un premier temps, la formule de représentation est employée pour trouver la région contenant la solution globale, en se basant sur la génération d'un nombre fini d'échantillons de variables aléatoires mis en jeu dans l'expression pour avoir une approximation des limites. Par exemple, nous pouvons choisir λ assez grand et produire des échantillons en utilisant les générateurs de nombre aléatoire standard.
La génération de ces points peut être faite en utilisant une distribution normale ou gaussienne. Dans le cas de la distribution gaussienne, quand un point d'essai produit se trouve en dehors de S, on le projette sur les bornes afin d'obtenir un point admissible.
Afin d'obtenir des résultats plus précis, il est commode de réaliser une amélioration des résultats par les algorithmes suivants :

- l'algorithme génétique classique (GA). On obtient un nouveau algorithme dit (RFGA)
- l'algorithme génétique en utilisant Nelder Mead au niveau de la mutation (GANM). On obtient un algorithme hybride qu'on nomme (RFGANM)
- l'algorithme Nelder Mead. On obtient un nouveau algorithme nommé (RFNM)

2.3.2 La Formule de Representation

Comme précédemment signalé, si f atteint son minimum global en un point x^* sur S, nous avons

$$x^* = \lim_{\lambda \to +\infty} \frac{E\left(X\, g\left(\lambda, f(X)\right)\right)}{E\left(g\left(\lambda, f(X)\right)\right)}, \tag{2.3}$$

où $g : \mathbb{R}^2 \longrightarrow \mathbb{R}$ est une fonction continue et strictement positive, $g(\lambda, s)$ est strictement décroissante pour tout $s \in f(S)$ et $\lambda > 0$, alors que X est une variable aléatoire donnée (voir, par exemple, Souza de Cursi et al. (2007)). Ces conditions sont remplies, par exemple, quand X est uniformément distribuée ou gaussienne et $g(\lambda, s) = e^{-\lambda s}$ (ce qui correspond au choix classique de Pincus). Nous employons ces choix particuliers dans la suite du chapitre.

Cette formule peut être interprétée comme étant une moyenne pondérée des points $x \in S$, où le poids diminue quand $f(x)$ augmente. La limite $g(\lambda, f(x))$ dans la formule de représentation disparaît pour les points ayant des valeurs plus grandes que $f(x^*)$ quand $\lambda \to +\infty$.

Bien que ce résultat mathématique semble être attirant, il est difficile de mettre en application des codes utilisant cette représentation. Dans notre travail, nous proposons d'employer une approximation numérique basée sur des échantillons finis des variables aléatoires impliquées dans l'expression et une approximation de la limite.

Dans la pratique, il suffit de prendre λ assez grande afin de représenter la limite $\lambda \to \infty$. Afin d'empêcher un dépassement de capacité, λ devrait être augmentée graduellement jusqu'à la valeur désirée et il peut être commode d'utiliser des fonctions f positive, en faisant un changement de variable (par exemple, en ajoutant une constante à f). Un échantillon fini de X est produit, selon une loi de probabilité P - ceci consiste simplement à générer N points admissibles $(x_1, x_2, ..., x_N) \in S$ - et des estimations des moyennes sont utilisées pour avoir une approximation de la moyenne exacte, ce que mène à :

$$x^* \simeq x_c^* = \frac{\sum_{i=1}^{N} x_i \;\; g\left(\lambda, f(x_i)\right)}{\sum_{i=1}^{N} g\left(\lambda, f(x_i)\right)}$$

2.3.3 Algorithme :

Algorithme 9 : Algorithmes hybrides

Etape 1 (**Initialisation**) :
a) Choisir Nb : taille de la population pour l'algorithme génétique,
b) Choisir λ, $Tol = 1e-3$, $\delta = 1$
c) Choisir $\alpha \in [0,1]$,
d) Choisir $\beta \in [1,2]$,
Etape 2 (**Generation des échantillons**) : Générer Nb échantillons de N éléments pour la variable aléatoire X (par exemple, utiliser une distribution uniforme dans S ou une restriction à S de la distribution gaussienne),
Etape 3 (**Approximation de la limite**) :(Nb fois) Calculer X_0 en utilisant Eq.(5.7)
repeat
  $\lambda = \beta * \lambda$
  Calculer Xe par Eq.(5.7)
  if Overflow **then**
    $\lambda = \alpha * \lambda$
    $\beta = \beta * \alpha$
  else
    $\delta = norm(X_e - X_0)$
    $X_e = X_0$
  end if
until $\delta > Tol$ Ou dépacement de capacité
Etape 4a (**Algorithme Génétique**) : utiliser les résultats comme points initiaux de la population,
Etape 4b (**Algorithme de Nelder Mead**) : Faire une recherche locale pour chaque individu,
Etape 4c (**Algorithme Génétique avec Nelder Mead**) : utiliser Nelder Mead au niveau de la mutation,

2.4 Fonctions test

Pour démontrer l'efficacité et la robustesse des algorithmes hybrides proposés, 21 fonctions test de différents niveaux de complexité et multimodalité ont été choisies de la littérature d'optimisation globale (Floudas 1999), (Gaviano et al. 2003), (Deep 2007), (Xinchao 2010), (Lee 2005), (Wang et al. 2005), (Huang et al. 2007), (Horst et al. 1995), (Liberti et Maculan 2006), (Shelokar et al. 2007), (Kao 2008), (Yuan et al. 2008).
Pour chaque fonction test, on réalise cent exécutions et on calcule le taux de succés, pour chacun des six algorithmes. Les fonctions test utilisées sont les suivantes (classées par ordre alphabétique) :

1. Bohachevsky 1 **BO1**
   $f(x) = x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1) - 0.4cos(4\pi x_2) + 0.7$,
   avec $x \in [-10,10]^2$. La valeur du minimum global est 0.0.
2. Bohachevsky 2 **BO2**
   $f(x) = x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1)cos(4\pi x_2) + 0.3$,
   avec $x \in [-10,10]^2$. La valeur du minimum global est 0.0.
3. Branin function **BR**
   $f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cos(x_1) + 10$,
   avec $x \in [-5,10][0,15]$. La valeur du minimum global est 0.397887.
4. Camel function **CA**
   $f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$,
   avec $x \in [-5,5]^2$. La valeur du minimum global est -1.0316.

5. Cosine mixture function **CO**
$f(x) = \sum_{i=1}^{n} x_i^2 - 0.1 \sum_{i=1}^{n} cos(5\pi x_i)$,
avec $x \in [-5, 10]^n$. La valeur du minimum global est -0.4, dans notre experience on a fixé $n = 4$.
6. DeJoung function **DE**
$f(x) = x_1^2 + x_2^2 + x_3^2$,
avec $x \in [-5.12, 5.12]^3$. La valeur du minimum global est 0.0.
7. Goldstein and price function **GO**
$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$,
avec $x \in [-2, 2]^2$. Le minimum global est obtenue au point $(0, -1)$ avec valeur de la fonction $f^* = 3.0$
8. Griewank function **GR**
$f(x) = \frac{1}{200} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$,
avec $x \in [-600, 600]^n$. Le minimum global est obtenue au point $x_i = 0$ avec valeur de la fonction $f^* = 0.0$
9. Hansen function **HN**
$f(x) = \sum_{i=1}^{5} icos[(i-1)x_1 + i] \sum_{j=1}^{5} jcos[(j+1)x_2 + j]$,
avec $x \in [-10, 10]^2$. La valeur du minimum global de la focntion est -176.541793.
10. Hartman 3 function **HR3**
$f(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$,
avec $x \in [0, 1]^3$, et
$a = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}, c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$ *and*
$p = \begin{pmatrix} 0.3689 & 0.117 & 0.26736 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{pmatrix}$ La valeur du minimum global est -3.862782.
11. Hartman 6 function **HR6**
$f(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$,
avec $x \in [0, 1]^6$, et
$a = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$,
$c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$ *and* $p = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix}$
La valeur du minimum global est -3.322368.
12. Rastrigin function **RA**
$f(x) = \sum_{i=1}^{n-1} (x_i^2 - 10cos(2\pi x_i) + 10)$,

avec $x \in [-5.12, 5.12]^2$. Le minimum global est situé au point 0.0 avec valeur de la fonction objectif égale à 0.0.

13. Rosenbrock function **RO**
$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$,
avec $x \in [-10, 10]^n$. Le minimum global est situé au point $x_i = 1.0$ avec valeur de la fonction objectif égale à 0.0. Dans notre cas $n = 10$.

14. Shekel 5 **SK5**
$f(x) = \sum_{i=1}^{5} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$,
avec $x \in [0, 10]^4$, et $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{pmatrix}, c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \end{pmatrix}$. La valeur du minimum global est -10.107749.

15. Shekel 7 **SK7**
$f(x) = \sum_{i=1}^{7} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$,
avec $x \in [0, 10]^4$, et $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \end{pmatrix}$,
$c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}$. La valeur du minimum global est -10.342378.

16. Shekel 10 **SK10**
$f(x) = \sum_{i=1}^{10} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$,
avec $x \in [0, 10]^4$, et
$a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix}, c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}$.
La valeur du minimum global est -10.536410.

17. Shubert 1 function **SH1**
$f(x) = \sum_{j=1}^{5} jcos((j+1)x + j)$,

avec $x \in [-10, 10]$, La valeur du minimum global est -12.871. dans notre experience on a fixé $n = 1$.

18. Shubert 2 function **SH2**
$f(x) = (\sum_{j=1}^{5} jcos((j+1)x_1 + j))(\sum_{j=1}^{5} jcos((j+1)x_2 + j))$,
avec $x \in [-10, 10]^2$, La valeur du minimum global est -186.730909. dans notre experience on a fixé $n = 2$.
19. Shubert 3 function **SH3**
$f(x) = \prod_{i=1}^{n}(\sum_{j=1}^{5} jcos((j+1)x_i + j))$,
avec $x \in [-10, 10]^3$, La valeur du minimum global est -2709.07. dans notre experience on a fixé $n = 3$.
20. Shubert 4 function **SH4**
$f(x) = \prod_{i=1}^{n}(\sum_{j=1}^{5} jcos((j+1)x_i + j))$,
avec $x \in [-10, 10]^4$, La valeur du minimum global est -39303.17. dans notre experience on a fixé $n = 4$.
21. Wolfe nondifferentiable function **WO** (Zowe 1985)

$$\begin{cases} f(x) = f_1(x), & \text{if } x_1 > |x_2|, \\ f(x) = f_2(x), & \text{if } 0 < x_1 \leq |x_2|, \\ f(x) = f_3(x), & \text{if } x_1 \leq 0, \\ f_1(x) = 5\sqrt{9x_1^2 + 16x_2^2}, & \\ f_2(x) = 9x_1 + 16|x_2|, & \\ f_3(x) = 9x_1 + 16|x_2| - x_1^9, & \end{cases}$$

avec $x \in [-100, 100]^2$, La valeur du minimum global est -8.00. dans notre experience on a fixé $n = 2$.

2.5 Résultats Numériques

Dans cette section nous nous intéressons à l'étude de l'efficacité des six algorithmes, c.-à-d. la formule de représentation (RF), l'algorithme génétique classique (GA), la formule de représentation avec GA (RFGA), l'algorithme génétique utilisant l'algorithme de Nelder Mead à l'étape de mutation (GANM), la formule de représentation avec GA et Nelder Mead (RFGANM), et la formule de représentation avec Nelder Mead (RFNM).

Une série d'expériences ont été réalisées pour faire un nombre d'analyses des performances de ces algorithmes. Pour éviter d'attribuer les résultats d'optimisation à un choix particulier des conditions initiales, et pour procéder à des comparaisons équitables, nous avons réalisé, pour chaque test, 100 exécutions, à partir de divers points choisis d'une façon aléatoire dans le domaine du hyper plan de recherche.
Les paramètres utilisés dans l'algorithme génétique sont : Taille de population : de 2 à 50, le taux de mutation est égale à 0.2, la méthode de selection est 'rank weighting', les critères d'arrêt sont le nombre maximal d'itérations (fixé à 2000 pour GA) et le nombre maximum d'itérations sans amélioration de la solution, et il est fixé à 1000 pour GA.
En ce qui concerne NM, nous avons adopté les paramètres standard recommandés par les auteurs, les critères d'arrêt utilisés sont : Nombre maximal d'évaluation de la fonction, $maxfun = 50000$, et le nombre maximal d'itération, $maxiter = 10000$, tolérance de variation de la fonction, $tolf = 10^{-5}$ et tolérance sur la variation de x est $xtolx = 10^{-6}$. Les

algorithmes proposés sont programmés sous Matlab 7.0, et sous le système d'exploitation Windows Xp, duo core, Intel 2.0 GHz avec 2 Go de RAM.

Des expérimentations sur les influences de différents paramètres ont été réalisés : Influence du choix de la distribution de probabilité, influence du choix de la fonction de Pincus, influence de la taille de population pour GA, influence de la dimension de l'échantillon pour la formule de représentation et enfin une comparaison de l'efficacité et l'efficiente de RFNM avec les résultats constatés pour d'autres méthodes et algorithmes de la littérature. Les abréviations suivantes sont utilisées pour alléger le texte :
TestF : Fonction d'essai ; Dim : Dimension de la variable ; PopGA : Taille de la population pour l'algorithme génétique ; SS : Dimension de l'échantillon utilisé dans la formule de représentation ; SR : Taux de réussite ; SD : Écart type ; CPUT : Temps CPU (en secondes) ; NEvalF : Nombre d'évaluations de la fonction.
Les résultats rapportés sont en termes de taux de réussite (SR), de l'écart type, du temps CPU et de la moyenne du nombre d'évaluations de la fonction. Le terme SR est le nombre d'essais réussis c-à-d quand l'algorithme produit une solution avec la précision demandée, où "la précision demandée" est une valeur maximum donnée qui est calculée comme la valeur absolue de la différence entre la solution trouvée et l'optimum global, divisé par la solution globale (quand elle est non nulle). La précision choisie pour tous les essais réalisés est de 10^{-4}.

2.5.1 Influence du choix de la distribution de probabilité

Lors de nos expériences, une probabilité définie sur le domaine $S = [a,b]$ est utilisée pour générer des échantillons finis de variables aléatoires impliquées dans la formule de représentation. L'échantillon peut être produit, soit en employant un générateur de nombre aléatoire suivant la loi uniforme ou une distribution gaussienne, $N(\mu,\sigma)$, et si un échantillon produit par la dernière méthode se trouve en dehors de S, il est remplacé par un autre produit par une distribution uniforme. Les valeurs de μ et de σ sont choisies ainsi : $\mu = \frac{a+b}{2}$ et $\sigma = \frac{b-a}{5}$.
Les tests réalisés sur le choix de la distribution de probabilité ont montrés l'absence d'effet majeur sur la qualité de solution produite. Dans les tests suivants, une distribution gaussienne $N(\mu,\sigma)$ est utilisée.

2.5.2 Influence de la fonction g

Dans la formule de représentation, l'expression de Pincus correspond à $g(\lambda,s) = e^{-\lambda s}$, qui est un choix convenable. Dans nos expériences, quatre autres fonctions de descente (continues et strictement décroissantes) ont été employées pour résoudre les fonctions tests avec les algorithmes proposés :

$$g_1(\lambda,s) = \frac{1}{\lambda\ s^3}$$

$$g_2(\lambda,s) = \frac{1}{\lambda\ ln(s)}$$

$$g_3(\lambda,s) = \frac{e^{-\lambda}}{s^3}$$

$$g_4(\lambda,s) = 10^{(-\lambda\ s)}$$

Les essais réalisés ont montrés que le choix de la fonction n'a pas d'effet significatif sur la qualité de la solution, seule une légère différence en terme de temps d'exécution a été observée.

2.5.3 Influence de la taille de la population pour GA

Pour examiner l'effet de la taille de la population aussi bien sur la qualité de la solution obtenue que sur le coût de calcul pour acquérir la solution optimale, utilisant GA, GANM, RFGA et RFGANM, huit tailles (2, 4, 6, 8, 12, 16, 20 et 50) sont examinées et les résultats expérimentaux sont rapportés dans la table 2.1 pour les méthodes GA et GANM et dans la table 2.2 pour RFGA et RFGANM. La dimension de l'échantillon utilisé pour la RF dépend de la dimension de la fonction test : $SS = 60$ pour $Dim = 1$ ou 2, $SS = 100$ pour $Dim = 3$, $SS = 300$ pour $Dim = 4$ ou 5, et $SS = 500$ pour $Dim = 6$ ou 10.
Une première constatation, est qu'en augmentant la taille de la population, les algorithmes ont besoin de plus de temps d'exécution (plus de nombre d'évaluations de la fonction) et tendent à trouver une meilleure solution. Ceci se manifeste par des valeurs faibles de l'écart type et des taux de succès (SR) plus grands. Dans le cas d'un problème plus complexe (cas de la fonction de Rosenbrock) GA et le RFGA n'arrivent pas à trouver la bonne solution pour tous les essais ($SR = 0\%$) même pour $PopGA = 200$. Une légère amélioration est obtenue par la formule de représentation (RFGA). on observe une détérioration de la solution obtenue par GA pour GANM (excepté pour SK10, BO2, GR et RO). Les résultats obtenus par RFGANM sont les meilleurs en terme de taux de succès ($SR = 100\%$ pour $PopGA = 50$) avec un nombre supplémentaire d'évaluation de fonction.

2.5.4 Influence de la taille de l'échantillon pour RF

Pour étudier l'effet de la taille de l'échantillon utilisé dans la formule de représentation sur la qualité des algorithmes hybrides, nous avons choisi six valeurs pour la taille de l'échantillon ($SS = 60, 100, 500,\ 1000, 2000\ et\ 5000$) et nous avons fixé la valeur de la taille de la population pour l'algorithme génétique sur la valeur de 6. Les résultats expérimentaux sont reportés dans le tableau 2.3. On observe que le taux de succès (SR) et le nombre d'évaluation de la fonction augmentent avec la taille de l'échantillon. La méthode RF seule n'arrive pas à retrouver l'optimum dans la plupart des tests ($SR = 0\%$ dans la plupart des cas) sauf pour le cas de la fonction HR3 ($SR = 100\%\ pour\ SS = 5000$), CA ($SR = 97\%\ pour\ SS = 5000$), SH1 ($SR = 100\%\ pour\ SS = 60$) et SH2 ($SR = 100\%\ pour\ SS = 5000$) avec une précision de 10^{-3}.

Concernant l'algorithme RFGANM, les résultats sont améliorés ($SR = 100\%$) pour des échantillons $SS \geq 1000$ sauf pour SH4, SK10 et GR. Un taux de succès de 100% est obtenu pour RFNM même pour des tailles d'échantillon de $SS = 60$.

2.5.5 Comparison entre méthodes

Les résultats présentés dans les tableaux 2.4 et 2.5 sont basés sur $PopGA = 12$ pour les algorithmes GA et GANM, et $PopGA = 6$ pour les autres. La taille d'échantillon choisie pour les fonctions test sont choisies en fonction de la dimension du problème : $SS = 60$ pour $Dim = 1/2$, 100 pour $Dim = 3$, 500 pour $Dim = 4/5$, 1000 pour $Dim = 6$ et 2000 pour $Dim = 10$.

Les tableaux 2.4 et 2.5 résument les résultats (c-à-d SR, SD, CPUT et NEvalF) obtenus pour les 100 éxecutions des six algorithmes, pour les 21 fonctions benchmark.
D'après les résultats dressés sur ces tableaux, on constate que :

- Les taux de succès pour GA sont généralement modestes (4% à 70% et SR=0% pour GR et RO), sauf pour le cas de SH1 et HR3 (80% et 90% respectivement).
- Les résultats sont améliorés pour GANM, pour lequel le taux de succès est 100%

TABLE 2.1 – *Influence de la taille de la population pour GA et GANM*

			GA				GANM			
TestF	Dim	PopGA	SR	SD	CPUT	NEvalF	SR	SD	CPUT	NEvalF
SH4	4	6	8%	8,94E-02	0,31	13 801	0%	9,91E-02	7,13	102 135
SH4	4	12	60%	2,13E-02	0,35	23 613	4%	2,68E-01	14,88	213 782
SH4	4	20	94%	2,13E-02	0,43	39 860	19%	3,16E-01	25,05	360 290
SH4	4	50	100%	5,77E-06	0,72	98 557	77%	1,11E-01	74,01	1 060 096
SK10	4	6	1%	3,20E-01	0,76	13 775	57%	3,07E-01	33,33	105 539
SK10	4	12	27%	3,52E-01	0,80	23 812	83%	2,08E-01	68,77	218 926
SK10	4	20	33%	3,33E-01	0,88	39 762	94%	1,22E-01	120,73	381 753
SK10	4	50	61%	3,31E-01	1,12	95 898	100%	5,64E-08	341,22	1 079 693
BO2	2	6	7%	1,08E-01	0,22	12 444	96%	4,30E-02	0,21	4 304
BO2	2	12	24%	1,07E-01	0,23	20 562	100%	5,72E-11	0,29	6 359
BO2	2	20	53%	1,06E-01	0,25	31 919	100%	2,68E-11	0,48	10 824
BO2	2	50	85%	7,61E-02	0,37	72 035	100%	3,47E-12	1,24	28 147
CA	2	6	70%	3,80E-04	0,23	12 662	96%	1,56E-01	1,66	31 178
CA	2	12	93%	1,00E-04	0,24	20 489	98%	1,11E-01	3,39	64 233
CA	2	20	100%	1,39E-05	0,28	33 515	100%	3,72E-08	5,74	109 435
CA	2	50	100%	6,06E-06	0,43	77 729	100%	3,72E-08	15,85	303 313
CO	4	6	39%	2,04E-04	0,26	13 639	7%	4,22E-01	1,72	31 747
CO	4	12	100%	1,16E-05	0,30	23 818	29%	2,46E-01	1,87	35 774
CO	4	20	100%	1,22E-06	0,35	39 336	65%	2,02E-01	1,91	38 478
CO	4	50	100%	1,09E-06	0,56	93 199	100%	6,88E-08	2,41	54 340
GR	5	6	0%	2,55E-01	0,28	13 776	71%	3,56E-01	12,19	183 428
GR	5	12	0%	1,87E-01	0,33	23 998	93%	4,17E-02	24,97	376 071
GR	5	20	1%	1,75E-01	0,39	39 884	99%	1,46E-02	42,64	642 752
GR	5	50	44%	1,18E-01	0,67	102 050	100%	1,95E-13	113,94	1 716 703
HN	2	6	46%	7,93E-02	0,24	12 721	23%	3,39E-01	2,33	39 949
HN	2	12	81%	5,06E-02	0,27	21 712	52%	2,62E-01	4,69	80 671
HN	2	20	94%	3,85E-02	0,32	36 018	76%	1,29E-01	8,29	142 888
HN	2	50	100%	3,19E-06	0,47	82 607	99%	1,76E-02	22,37	384 416
RA	2	6	6%	1,62E-02	0,22	12 522	11%	4,42E+00	1,28	24 635
RA	2	12	27%	5,05E-03	0,22	19 169	51%	1,24E+00	1,35	26 337
RA	2	20	64%	3,62E-04	0,26	32 278	67%	6,11E-01	1,57	31 274
RA	2	50	93%	1,11E-04	0,34	67 520	98%	1,40E-01	1,61	35 270
RO	10	6	0%	4,17E+01	0,29	14 006	93%	1,02E+00	22,17	338 628
RO	10	12	0%	3,45E+01	0,33	24 012	100%	7,88E-12	44,78	683 973
RO	10	20	0%	3,37E+01	0,40	40 020	100%	5,49E-12	75,83	1 161 545
RO	10	50	0%	2,92E+01	0,69	102 050	100%	2,99E-12	200,54	3 071 508
WO	2	6	0%	1,17E-01	0,22	12 685	100%	0,00E+00	1,00	19 462
WO	2	12	1%	4,63E-02	0,23	20 512	100%	0,00E+00	2,04	39 624
WO	2	20	6%	2,59E-02	0,25	31 105	100%	0,00E+00	3,55	69 076
WO	2	50	43%	1,07E-02	0,37	70 768	100%	0,00E+00	9,42	183 422

pour 7 fonctions test, et $SR \geq 90\%$ pour 6 functions test . Ces résultats sont similaires à ceux obtenus dans Chelouah et Siarry (2003).

- La représentation de solution RF a échoué de converger vers l'optimum global pour presque tous les tests, sauf pour SH1 où $SR = 67\%$($SR = 100\%$ pour une précision de 10^{-3}). On note que le nombre d'évaluation de fonction a considérablement augmenté.
- La précision pour l'algorithme RFGA est plus faible que GANM, avec un nombre d'évaluation de la fonction plus important. Un succès total obtenu pour seulement 5 fonctions ($SR \geq 99\%$). Le nombre d'évaluation de la function est similaire à celui de l'algorithme RFGA.
- Compte tenu de l'efficacité et l'efficiente des algorithmes dans l'ensemble, l'approche de RFGANM reste la plus compétitive par rapport à RFNM que les autres méthodes. En effet, le SR est de 100% pour 12 fonctions test, et $SR \geq 82\%$ pour 18

TABLE 2.2 – *Influence de la taille de la population pour RFGA et RFGANM*

			RFGA				RFGANM			
TestF	Dim	PopGA	SR	SD	CPUT	NEvalF	SR	SD	CPUT	NEvalF
SH4	4	6	10%	8,53E-02	0,42	22 838	63%	1,89E-01	7,31	112 254
SH4	4	12	50%	5,08E-02	0,57	41 613	90%	6,42E-02	14,82	228 329
SH4	4	20	89%	2,13E-02	0,80	69 809	98%	3,00E-02	25,57	391 947
SH4	4	50	100%	7,11E-06	1,63	172 398	100%	5,05E-08	70,56	1 072 064
SK10	4	6	2%	2,92E-01	0,92	22 737	86%	1,81E-01	33,77	115 454
SK10	4	12	45%	2,64E-01	1,12	41 626	98%	7,16E-02	68,23	234 259
SK10	4	20	84%	1,95E-01	1,41	69 439	100%	5,64E-08	117,89	401 052
SK10	4	50	94%	1,22E-01	2,45	170 297	100%	5,64E-08	342,74	1 155 194
BO2	2	6	8%	1,08E-01	0,23	14 341	77%	1,05E-01	0,47	10 879
BO2	2	12	34%	9,73E-02	0,26	24 310	98%	3,07E-02	0,36	10 864
BO2	2	20	62%	8,04E-02	0,29	38 068	100%	3,02E-11	0,50	16 495
BO2	2	50	92%	3,74E-02	0,43	82 245	100%	4,28E-12	1,27	41 913
CA	2	6	67%	5,21E-04	0,24	14 199	100%	3,72E-08	1,66	32 662
CA	2	12	96%	4,03E-05	0,27	23 890	100%	3,72E-08	3,35	66 514
CA	2	20	99%	2,33E-05	0,31	38 013	100%	3,72E-08	5,72	114 142
CA	2	50	100%	4,52E-06	0,49	84 487	100%	3,72E-08	15,76	314 430
CO	4	6	48%	3,01E-04	0,32	22 680	98%	5,20E-02	0,31	14 498
CO	4	12	100%	1,52E-05	0,41	41 790	100%	6,88E-08	0,58	28 515
CO	4	20	100%	1,36E-06	0,56	69 246	100%	6,88E-08	0,97	47 799
CO	4	50	100%	4,89E-07	1,07	170 920	100%	6,88E-08	2,52	122 363
GR	5	6	0%	2,44E-01	0,36	22 823	38%	3,22E-01	12,13	190 443
GR	5	12	0%	1,76E-01	0,47	42 012	74%	9,71E-02	25,04	392 890
GR	5	20	0%	1,62E-01	0,64	70 011	88%	5,05E-02	42,92	673 123
GR	5	50	46%	1,24E-01	1,27	177 050	100%	2,13E-13	114,24	1 787 308
HN	2	6	67%	1,76E-02	0,27	14 654	99%	1,76E-02	2,33	41 571
HN	2	12	97%	4,53E-05	0,32	25 478	100%	6,37E-08	4,69	83 688
HN	2	20	100%	1,31E-05	0,39	40 485	100%	6,37E-08	8,08	144 040
HN	2	50	100%	1,68E-06	0,65	93 454	100%	6,37E-08	22,25	393 771
RA	2	6	11%	8,52E-03	0,23	14 268	74%	4,39E-01	0,37	8 954
RA	2	12	26%	1,43E-03	0,25	23 131	95%	2,18E-01	0,39	11 430
RA	2	20	53%	5,02E-04	0,29	36 968	98%	1,40E-01	0,56	17 506
RA	2	50	95%	7,22E-05	0,44	83 305	100%	1,41E-11	1,28	42 039
RO	10	6	0%	1,27E+00	0,43	28 985	100%	2,28E-11	21,59	342 395
RO	10	12	0%	1,97E+00	0,60	54 012	100%	8,42E-12	43,90	696 293
RO	10	20	0%	9,84E-01	0,85	90 020	100%	4,57E-12	74,82	1 188 783
RO	10	50	0%	6,08E-01	1,80	227 050	100%	2,36E-12	197,37	3 130 515
WO	2	6	0%	1,47E-01	0,24	14 214	100%	0,00E+00	1,01	21 029
WO	2	12	0%	6,09E-02	0,27	23 663	100%	0,00E+00	2,05	42 744
WO	2	20	6%	2,96E-02	0,30	35 050	100%	0,00E+00	3,52	73 376
WO	2	50	46%	6,06E-03	0,50	82 856	100%	0,00E+00	9,55	198 171

fonctions ($SR = 100\%$ pour $PopGA = 50$, voir tableau 2.2 pour tous les fonctions test). Le nombre d'évaluations de fonction est similaire à celui de RFGA.

– Concernant l'algorithme RFNM, les résultats expérimentaux obtenus pour les 21 fonctions test montrent une grande efficacité, avec 100% de taux de succès pour tous les exemples traités, avec un nombre moindre d'évaluations de fonction par rapport à RFGANM et RFGA.

2.5.6 Comparison avec d'autres algorithmes

Dans cette section, l'experimentation a pour but de comparer les performances de l'algorithme RFNM par rapport à cinq méthodes d'optimisation globale listées ci après (Tableau 2.6). Pour pouvoir assurer des résultats comparables, tous les conditions de d'essais sont les mêmes que celle des algorithme à comparer (100 exécutions, et la précision utilisée est fixée à 10^{-4}).

TABLE 2.3 – *Influence de la taille d'échantillon utilisé dans RF*

TestF	Dim	RF Size	RF SR	RF Neval	RFGA SR	RFGA Neval	RFGANM SR	RFGANM Neval	RFNM SR	RFNM Neval
SH4	4	60	0%	1 800	7%	15 460	38%	103 489	43%	7 981
SH4	4	100	0%	3 000	9%	16 892	34%	104 957	47%	9 137
SH4	4	500	0%	15 000	8%	28 885	74%	117 804	100%	21 010
SH4	4	1000	0%	30 000	11%	43 838	84%	133 282	100%	36 010
SH4	4	5000	0%	150 000	10%	163 800	97%	252 327	100%	155 899
SK10	4	60	0%	1 800	0%	15 470	78%	108 783	100%	8 002
SK10	4	100	0%	3 000	2%	16 669	72%	110 233	100%	9 209
SK10	4	500	0%	15 000	4%	28 783	91%	121 826	100%	21 198
SK10	4	1000	0%	30 000	1%	43 725	95%	136 186	100%	36 202
SK10	4	5000	0%	150 000	1%	163 758	99%	255 924	100%	155 946
BO2	2	60	0%	1 800	10%	14 320	83%	9 338	100%	4 421
BO2	2	100	0%	3 000	9%	15 477	78%	11 953	100%	5 532
BO2	2	500	0%	15 000	14%	26 791	92%	20 139	100%	17 332
BO2	2	1000	0%	30 000	15%	41 376	100%	32 914	100%	32 275
BO2	2	5000	0%	150 000	17%	160 866	100%	152 901	100%	152 160
CA	2	60	1%	1 800	68%	13 957	100%	32 760	100%	4 338
CA	2	100	1%	3 000	71%	15 106	100%	33 982	100%	5 565
CA	2	500	1%	15 000	71%	27 101	100%	45 952	100%	17 481
CA	2	1000	7%	30 000	65%	41 226	100%	60 842	100%	32 464
CA	2	5000	11%	150 000	73%	160 307	100%	180 812	100%	152 405
CO	4	60	0%	1 800	58%	15 421	92%	8 142	96%	8 990
CO	4	100	0%	3 000	50%	16 752	93%	9 167	96%	10 219
CO	4	500	0%	15 000	43%	28 507	100%	20 130	100%	22 540
CO	4	1000	0%	30 000	54%	43 704	100%	35 123	100%	37 627
CO	4	5000	0%	150 000	52%	163 426	100%	155 101	100%	157 744
GR	5	60	0%	1 800	0%	15 685	43%	183 996	97%	27 227
GR	5	100	0%	3 000	0%	16 866	32%	185 066	100%	28 212
GR	5	500	0%	15 000	0%	28 872	49%	196 576	100%	40 228
GR	5	1000	0%	30 000	0%	43 887	39%	210 601	100%	54 615
GR	5	5000	0%	150 000	0%	163 962	33%	330 770	100%	172 940
HN	2	60	1%	1 800	63%	14 338	100%	41 534	100%	4 008
HN	2	100	2%	3 000	63%	15 387	100%	42 604	100%	5 252
HN	2	500	1%	15 000	57%	27 103	100%	54 571	100%	17 107
HN	2	1000	4%	30 000	71%	42 027	100%	69 684	100%	32 130
HN	2	5000	14%	150 000	72%	161 323	100%	189 857	100%	152 137
RA	2	60	0%	1 800	7%	13 856	64%	10 726	93%	4 054
RA	2	100	0%	3 000	13%	15 105	79%	9 321	100%	5 256
RA	2	500	0%	15 000	8%	26 663	96%	18 566	100%	17 306
RA	2	1000	0%	30 000	7%	41 441	100%	32 914	100%	32 312
RA	2	5000	1%	150 000	5%	160 487	100%	152 868	100%	152 329
RO	10	60	0%	1 800	0%	15 806	99%	326 193	96%	133 508
RO	10	100	0%	3 000	0%	16 979	99%	328 651	100%	133 634
RO	10	500	0%	15 000	0%	29 006	100%	341 868	100%	146 056
RO	10	1000	0%	30 000	0%	43 985	99%	354 962	100%	161 398
RO	10	5000	0%	150 000	0%	163 858	100%	477 806	100%	281 391
WO	2	60	0%	1 800	0%	14 399	100%	21 077	100%	6 467
WO	2	100	0%	3 000	0%	15 090	100%	22 260	100%	7 567
WO	2	500	0%	15 000	0%	27 260	100%	34 235	100%	19 385
WO	2	1000	0%	30 000	0%	41 178	100%	49 244	100%	34 352
WO	2	5000	0%	150 000	0%	160 978	100%	169 121	100%	154 137

Dans les tests précédents, nous avons utilisé les mêmes paramètres pour toutes les fonctions test. Les résultats ont montré que l'algorithme RFNM est robuste (SR = 100%). Pour comparer RFNM avec les algorithmes de la liste ci-dessus, les paramètres sont ajustés pour chaque situation (fonction).
le tableau 2.7 montre que RFNM est plus performant que les autres algorithmes, pour trois fonctions test (BR, SK10, RO) avec un taux de succès de 100%, et un nombre supplémentaire d'évaluation de fonction dans d'autres cas.

2.6 CONCLUSION

Dans ce chapitre, nous avons proposé une nouvelle approche fondée sur une formule de représentation pour résoudre des problèmes d'optimisation globale. Des expérimen-

TABLE 2.4 – *Comparaison entre méthodes : GA, GANM et RF*

			GA				GANM				RF			
Num	FTest	Dim	SR	SD	CPUT	NEvalF	SR	SD	CPUT	NEvalF	SR	SD	CPUT	NEvalF
1	BO1	2	52%	5,09E-04	0,22	19 653	100%	4,97E-11	0,29	6 422	0%	2,66E-01	0,01	1 800
2	BO2	2	24%	1,07E-01	0,23	20 562	100%	5,72E-11	0,29	6 359	0%	1,74E-01	0,01	1 800
3	BR	2	50%	6,22E-04	0,25	22 138	100%	5,11E-08	4,49	86 543	0%	6,27E-02	0,01	1 800
4	CA	2	93%	1,00E-04	0,24	20 489	98%	1,11E-01	3,39	64 233	1%	4,65E-02	0,01	1 800
5	CO	4	100%	1,16E-05	0,30	23 818	29%	2,46E-01	1,87	35 774	0%	1,05E-01	0,10	15 000
6	DE	3	99%	2,30E-05	0,27	23 214	100%	2,76E-12	0,36	8 530	0%	1,02E-01	0,02	3 000
7	GO	2	62%	2,23E-04	0,33	22 190	98%	1,27E+00	2,60	39 874	0%	3,79E-01	0,01	1 800
8	GR	5	0%	1,87E-01	0,33	23 998	93%	4,17E-02	24,97	376 071	0%	2,74E+01	0,14	15 000
9	HN	2	81%	5,06E-02	0,27	21 712	52%	2,62E-01	4,69	80 671	1%	5,52E-02	0,02	1 800
10	HR3	3	100%	1,58E-06	0,31	22 829	100%	6,43E-08	9,22	131 457	0%	4,22E-03	0,02	3 000
11	HR6	6	66%	1,71E-02	0,33	24 012	98%	5,05E-03	32,29	397 117	0%	2,02E-02	0,27	30 000
12	RA	2	27%	5,05E-03	0,22	19 169	51%	1,24E+00	1,35	26 337	0%	6,81E-01	0,01	1 800
13	RO	10	0%	3,45E+01	0,33	24 012	100%	7,88E-12	44,78	683 973	0%	3,69E+00	2,69	60 000
14	SH1	1	99%	1,97E-05	0,16	17 519	96%	6,66E-02	1,99	41 092	67%	1,38E-04	0,03	1 800
15	SH2	2	95%	3,81E-05	0,26	20 779	62%	2,69E-01	4,77	81 145	0%	5,03E-02	0,02	1 800
16	SH3	3	69%	5,46E-02	0,32	23 017	25%	3,28E-01	9,75	149 699	0%	1,41E-01	0,05	3 000
17	SH4	4	60%	2,13E-02	0,35	23 613	4%	2,68E-01	14,88	213 782	0%	1,40E-01	0,15	15 000
18	SK10	4	27%	3,52E-01	0,80	23 812	83%	2,08E-01	68,77	218 926	0%	1,05E-01	0,20	15 000
19	SK5	4	25%	3,39E-01	0,54	23 623	96%	1,12E-01	40,24	217 139	0%	8,64E-02	0,23	15 000
20	SK7	4	21%	3,29E-01	0,65	23 726	90%	1,58E-01	51,66	217 461	0%	1,21E-01	0,23	15 000
21	WO	2	1%	4,63E-02	0,23	20 512	100%	0,00E+00	2,04	39 624	0%	3,14E+00	0,02	1 800

TABLE 2.5 – *Comparaison entre méthodes : RFGA, RFGANM et RFNM*

			RFGA				RFGANM				RFNM			
Num	FTest	Dim	SR	SD	CPUT	NEvalF	SR	SD	CPUT	NEvalF	SR	SD	CPUT	NEvalF
1	BO1	2	19%	6,64E-03	0,23	13 842	82%	1,65E-01	0,29	7 460	99%	4,13E-02	0,15	4 349
2	BO2	2	10%	1,09E-01	0,23	14 320	83%	8,88E-02	0,39	9 338	100%	4,49E-11	0,15	4 421
3	BR	2	27%	2,63E-03	0,22	13 463	100%	5,11E-08	2,19	43 712	100%	5,11E-08	0,14	4 202
4	CA	2	68%	2,99E-04	0,23	13 957	100%	3,72E-08	1,65	32 760	100%	3,72E-08	0,15	4 338
5	CO	4	43%	2,20E-04	0,36	28 507	100%	6,88E-08	0,33	20 130	100%	6,88E-08	0,53	22 540
6	DE	3	78%	5,48E-05	0,25	16 260	100%	4,99E-12	0,19	7 073	100%	2,25E-12	0,29	8 271
7	GO	2	25%	1,11E-03	0,32	14 649	100%	0,00E+00	1,27	20 873	100%	0,00E+00	0,17	4 212
8	GR	5	0%	2,41E-01	0,43	28 872	49%	2,04E-01	12,12	196 576	90%	5,05E-02	1,77	40 228
9	HN	2	63%	1,76E-02	0,27	14 338	100%	6,37E-08	2,33	41 534	100%	6,37E-08	0,16	4 008
10	HR3	3	100%	1,34E-05	0,30	16 290	100%	6,43E-08	4,37	65 699	100%	6,43E-08	0,33	7 570
11	HR6	6	35%	1,74E-02	0,58	43 874	73%	1,60E-02	16,81	231 152	100%	2,00E-08	1,76	48 337
12	RA	2	7%	9,79E-03	0,22	13 856	64%	5,03E-01	0,46	10 726	93%	2,55E-01	0,13	4 054
13	RO	10	0%	8,02E-01	2,98	73 994	99%	3,99E-01	23,75	386 100	98%	2,41E-03	11,24	190 569
14	SH1	1	88%	9,12E-05	0,15	9 942	100%	4,91E-08	1,00	21 925	100%	4,91E-08	0,08	2 756
15	SH2	2	51%	4,18E-04	0,27	14 308	100%	3,48E-08	2,29	40 642	100%	3,48E-08	0,16	4 047
16	SH3	3	36%	3,66E-02	0,33	16 512	93%	7,48E-02	4,72	75 502	99%	3,38E-02	0,29	6 771
17	SH4	4	8%	7,18E-02	0,46	28 885	74%	1,38E-01	7,27	117 804	83%	1,05E-01	0,56	21 010
18	SK10	4	4%	2,57E-01	0,96	28 783	91%	1,51E-01	34,07	121 826	100%	5,64E-08	2,16	21 198
19	SK5	4	3%	2,57E-01	0,74	28 858	83%	1,88E-01	20,33	122 398	100%	0,00E+00	1,37	21 129
20	SK7	4	2%	2,72E-01	0,84	28 693	91%	1,51E-01	25,71	121 771	100%	7,07E-08	1,70	21 206
21	WO	2	0%	1,17E-01	0,24	14 399	100%	0,00E+00	1,01	21 077	100%	0,00E+00	0,26	6 467

TABLE 2.6 – *Comparison avec d'autre méthodes, liste des méthodes*

Méthodes	Références
Representation de solution avec Nelder Mead (RFNM)	Cette thèse
Enhanced Continuous Tabu Search (ECTS)	Chelouah et Siarry (2005) Georgievaa et Jordanov (2010)
Staged Continuous Tabu Search (SCTS)	Zheng (2005) Georgievaa et Jordanov (2010)
Continuous Hybrid Algorithm (CHA)	Chelouah et Siarry (2003) Georgievaa et Jordanov (2010)
Differential Evolution (DE)	Price et al. (2005) Georgievaa et Jordanov (2010)
LPτNM Algorithm (LPτNM)	Georgievaa et Jordanov (2010)

TABLE 2.7 – *Comparison avec d'autres méthodes en terme de taux de succès et nombre d'évaluations de fonction*

TestF(Dim)	LPτNM NEvalF(SR)	SCTS NEvalF(SR)	CHA NEvalF(SR)	DE NEvalF(SR)	RFNM NEvalF(SR)
SH2(2)	303(85%)	370(100%)	345(100%)	4498(95%)	677(100%)
GO(2)	182(100%)	231(100%)	259(100%)	595(100%)	286(100%)
BR(2)	247(100%)	245(100%)	295(100%)	807(100%)	165(100%)
HR3(3)	292(100%)	548(100%)	492(100%)	679(100%)	312(100%)
SK10(4)	1079(96%)	898(75%)	635(85%)	-(-%)	1680(100%)
SK7(4)	837(100%)	910(80%)	620(85%)	3064(100%)	1059(100%)
SK5(4)	839(100%)	825(75%)	698(85%)	3920(100%)	874(100%)
HR6(6)	1552(100%)	1520(100%)	930(100%)	-(-%)	2551(100%)
RO(10)	9188(88%)	15720(85%)	14532(83%)	54134(100%)	42572(100%)

tation ont été réalisés pour l'optimisation de fonctions non linéaires, non différentiables et multimodaux utilisant cette formule de représentation, hybridée avec l'algorithme génétique et l'algorithme de Nelder Mead, et ont montré que RFNM est plus robuste que les autres méthodes. le RFGANM demeure robuste pour des tailles de population plus important, avec un nombre d'évaluations de fonctions plus grand.
Des expérimentations approfondies concernant l'effet de différents paramètres ont été réalisées : l'influence du choix de la distribution de probabilité, l'influence de la fonction de Pincus, l'influence de la taille de la population pour l'algorithme génétique, et l'influence de la taille de l'échantillon pour la formule de représentation RF. Une comparaison avec d'autres algorithmes proposés dans la littérature montre qu'en général RFNM est plus efficace. Des recherches plus approfondies consistent à inclure d'autres algorithmes stochastiques tels que l'algorithme essaim de particules avec la formule de représentation.

CHAPITRE 3

APPLICATIONS : PROBLÈMES DU PILIER DE PONT ET D'UNE STRUCTURE EN TREILLIS

SOMMAIRE

4.1 INTRODUCTION 70
4.2 GÉNÉRALITÉS SUR LA CONCEPTION MÉCANIQUE LORS DE L'EXAMEN DE LA FIABILITÉ DES STRUCTURES 71
4.3 EQUATIONS DU PROBLÈME DE VIBRATION MÉCANIQUE 72
4.4 FORMULATION DU PROBLÈME D'OPTIMISATION SOUS CONTRAINTE FIABILISTE 73
4.5 PROPAGATION DES INCERTITUDES POUR LE SYSTÈME MÉCANIQUE 74
4.6 AMÉLIORATION DE LA MÉTHODE NNC 75
4.6.1 Introduction 75
4.6.2 Formulation et amélioration de la méthode NNC 75
4.7 APPLICATIONS 78
4.7.1 Premier Problème d'optimisation 78
4.7.2 Le deuxième problème d'optimisation 79
4.8 COMPARAISON AVEC D'AUTRES MÉTHODES D'OPTIMISATION MULTIOBJECTIF 81
4.9 CONCLUSION 85

DANS ce chapitre, deux problèmes linéaires avec une charge aléatoire gaussienne sont transformés en problèmes d'optimisation multi-objectif. Le premier problème est la conception d'une géométrie de pilier à l'égard d'un processus de chargement aléatoire de compression. Le deuxième problème est la conception d'une structure en treillis par rapport à un processus de chargement aléatoire vertical pour plusieurs bandes de fréquences. l'algorithme RFNM, motivé par la formule de représentation hybridée avec l'algorithme de Nelder-Mead, est proposé pour résoudre les deux problèmes d'optimisation multi-objectif. Pour générer la courbe de Pareto, la méthode NBI (Normal Boundary Intersection) est utilisée pour transformer le problème multi-objectif sans contraintes en une série

de problèmes mono-objectif avec contraintes. Selon la bande de fréquence d'excitation, on peut avoir comme courbe de Pareto un seul point, une courbe convexe de Pareto, ou une courbe discontinue de Pareto. Ce cas est confronté pour la première fois dans la littérature. Une comparaison des performances de RFNM couplé avec NBI, avec l'algorithme génétique(GA), l'algorithme essaim de particule(PSO), et NSGA II est réalisée afin de tester l'efficacité et l'efficience de notre algorithme.

Le chapitre est organisé comme suit : la section 3.2 présente une description de la conception de pilier de pont, tandis que la section 3.3 est consacrée au problème de conception de structure en treillis. l'étude de l'efficacité de la méthode RFNM est mis en évidence dans la section 3.4, suivie de quelques remarques finales dans la section 3.5.

Les principales contributions en rapport avec ce chapitre sont :

H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. **Multi-Objective Optimization By A New Hybridized Method : Applications To Random Mechanical Systems**, *Engineering optimization*, (ISSN 0305 – 215X (Print), 1029 – 0273 (Online)), DOI : 10.1080/0305215*X*.2012.713355.

H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. **Multi-Objective Optimization Of The Dynamic Of A Bridge Pillar And A Truss Structure Subjected To Random Load By A New Hybridized Method**, *Mecánica Computacional*, (ISSN 1666 – 6070), v. 30, p. 3339 – 3355, 2011.

3.1 Introduction

L'optimisation des problèmes d'ingénierie est très complexe et assez difficile pour l'approcher. Ils sont en général multi-objectif et ont des paramètres ou charges qui se comportent d'une manière aléatoire. Pour les problèmes multi-objectif, la solution peut être présentée comme un front de Pareto et l'ingénieur de design doit faire un compromis entre les différents critères d'optimisation.

Deux problèmes avec charges aléatoires sont traités. A cause de la normalité de la charge et de la linéarité des problèmes, chacun d'eux sera transformé en un problème multi-objectif pour les moments de déplacement, d'où l'élimination de l'aspect aléatoire du problème. Dans notre travail, on considère deux moments seulement : la moyenne et l'écart type du déplacement. Le premier problème est un problème standard, et le résultat de l'optimisation est une courbe convexe ; le deuxième problème est plus intéressant car le front de Pareto dépend de la bande de fréquence choisie pour la charge. Pour des valeurs spécifiques de la bande de fréquence, la courbe de Pareto est non convexe et discontinue.

Pour construire la courbe de Pareto du problème d'optimisation multi-objectif, la stratégie adoptée est d'abord de transformer le problème en une séquence de problèmes mono-objectif en utilisant la méthodologie du NBI. Quelques nouvelles contraintes sont introduites dans les nouveaux problèmes et la méthode des pénalités est utilisée pour situer le problème dans un cadre qui permet l'utilisation de l'algorithme RFNM pour construire la courbe de Pareto.

3.2 Conception d'un pilier

Le premier problème considéré est la conception d'un pilier pour pont constitué de deux parties (voir figure 3.1). La première partie est construite en béton, la seconde est en acier. Les deux sections S_1 et S_2 de ces éléments structuraux doivent être obtenues par la

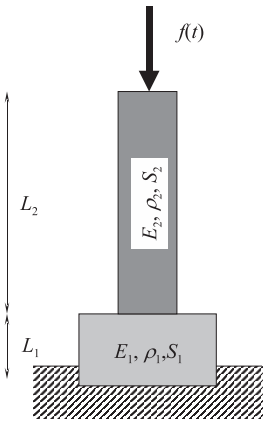


FIGURE 3.1 – *Schéma et dimension du pilier*

procédure d'optimisation. Toutefois, elles doivent être supérieures à des valeurs données pour que le pilier ne se déforme pas de manière irréversible et le matériau reste dans le domaine élastique. Une autre contrainte est fixée pour la longueur minimale du pilier qui doit être supérieur ou égal à 1 m afin de s'assurer de la faisabilité de la fondation de la partie en béton du pilier. Dans cette étude, seul le mouvement axial de la colonne est considéré, en supposant que le fond correspondant à $x = 0$ est fixe tandis que la partie pour $x = L$ est soumise à une moyenne plus une charge en fluctuation ou en mouvement.

Le module de Young et la densité sont désignés par E_1 et ρ_1 pour le béton et E_2 et ρ_2 pour l'acier. La géométrie de la partie en béton du pilier a une longueur L_1 avec une section transversale constante S_1, alors que la partie en acier a une aire constante S_2, avec une longueur L_2 tels que $L = L_1 + L_2$. Pour l'application numérique, on a choisi $E_1 = 3.6 \times 10^{10}$ Pa, $E_2 = 2.1 \times 10^{11}$ Pa, $\rho_1 = 2500$ kg/m ^{3}, $\rho_2 = 7800$ kg/m ^{3} et $L = 15m$.

D'après la littérature de la mécanique des milieux continus , lorsque l'amortissement, la charge de gravité et les effets transversaux ne sont pas considérés, les équations à vérifier pour le pilier (Géradin et Rixen 1996) sont les suivantes :

$$\begin{cases} -E_1 \frac{\partial^2 u}{\partial x^2} = \omega^2 \rho_1 u(x,\omega) & \text{pour} 0 \leq x \leq L_1 \\ -E_2 \frac{\partial^2 u}{\partial x^2} = \omega^2 \rho_2 u(x,\omega) & \text{pour} L_1 \leq x \leq L \end{cases} \tag{3.1}$$

En tenant compte des conditions aux limites et de compatibilité, on a :

$$\begin{cases} u(x,\omega) = 0 & \text{pour} x = 0 \\ -E_2 S_2 \frac{\partial u}{\partial x} = f(\omega) & \text{pour} x = L \\ -E_1 S_1 \frac{\partial u}{\partial x} = E_2 S_2 \frac{\partial u}{\partial x} & \text{pour} x = L_1 \end{cases} \tag{3.2}$$

où $u(x,\omega)$ et $f(\omega)$ représentent respectivement le champ de déplacement longitudinal déterministe et la charge au niveau de la pointe du pilier dans le domaine fréquentiel, ω étant la fréquence circulaire. En plus, le champ u doit être partout continu dans le pilier, en particulier à l'interface entre les deux parties (qui constitue une deuxième condition de compatibilité). En introduisant les vitesses d'onde $c_1 = \sqrt{\frac{E_1}{\rho_1}}$ et $c_2 = \sqrt{\frac{E_2}{\rho_2}}$, la solution est recherchée sous la forme :

$$u(x,\omega) = \begin{cases} C_1 \cos\left(\frac{\omega x}{c_1}\right) + D_1 \sin\left(\frac{\omega x}{c_1}\right) & \text{pour} \quad 0 \leq x \leq L_1 \\ C_2 \cos\left(\frac{\omega x}{c_2}\right) + D_2 \sin\left(\frac{\omega x}{c_2}\right) & \text{pour} \quad L_1 \leq x \leq L \end{cases}$$

où C_i et D_i sont des constantes à déterminer à partir des quatres limites et conditions de compatibilité. Toutefois, seuls les déplacements u à l'extrémité $x = L$ du pilier nous intéressent. On introduit les variables $\alpha_i = \omega \frac{L_i}{c_i}$ et $\beta_i = \frac{E_i S_i}{L_i} \alpha_i$ pour $i = \{1,2\}$, on obtient

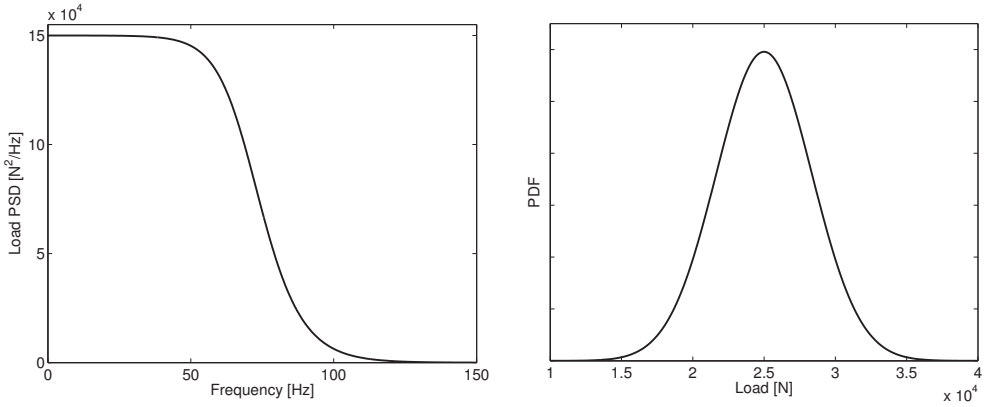


FIGURE 3.2 – *PSD et PDF de la charge appliquée au pilier*

après quelques manipulations :

$$u(x,\omega) = \begin{cases} \frac{\sin\left(\alpha_1 \frac{x}{L_1}\right)}{\sin(\alpha_1)} u(x = L_1, \omega) & \text{pour} \quad 0 \leq x \leq L_1 \\ \frac{\sin\left(\alpha_2 \frac{L-x}{L-L_1}\right)}{\sin(\alpha_2)} u(x = L_1, \omega) + \frac{\sin\left(\alpha_2 \frac{L_1-x}{L_1-L}\right)}{\sin(\alpha_2)} u(x = L, \omega) & \text{pour} \quad L_1 \leq x \leq L \end{cases}$$

avec :

$$u(x = L_1, \omega) = \left(\frac{\beta_1}{\tan(\alpha_1)\tan(\alpha_2)} - \beta_2\right)^{-1} \frac{f(\omega)}{\sin(\alpha_2)}$$

et :

$$u(x = L, \omega) = h(\omega) \times f(\omega)$$

Où $h(\omega)$ est la fonction de réponse en fréquence de l'extrémité du pilier à $x = L$ et est telle que :

$$h(\omega) = \frac{1}{\beta_2}\left(\frac{\beta_1}{\tan(\alpha_1)\tan(\alpha_2)} - \beta_2\right)^{-1}\left(\frac{\beta_1}{\tan(\alpha_1)} + \frac{\beta_2}{\tan(\alpha_2)}\right)$$

Toutefois, étant donné que les structures réelles sont amorties, il a été ajouté un léger amortissement dans ce modèle en remplaçant les constantes du vitesse réel du son c_i par les vitesses complexes des sons $\tilde{c}_i$ telles que $\tilde{c}_i = c_i(1 + j\eta_i)$ avec les facteurs d'amortissement η_i ayant la valeur arbitraire $\eta_i = 0.07$.

Dans ce problème, il est intéressant de concevoir la géométrie du pilier par rapport à un processus de charge de compression aléatoire $F(t)$, où t désigne le temps. Cette charge est censée être gaussienne, ergodique et suivre un processus aléatoire stationnaire (gaussian ergodic and stationary random process) (Lin 1967, Wijker 2009). Ainsi, il pourrait être décrit par sa moyenne $\overline{F} = \mathrm{E}[F(t)]$ et sa fonction de covariance $\Sigma_{FF}(\tau) = \mathrm{E}\left[\left(F(t) - \overline{F}\right)\left(F(t+\tau) - \overline{F}\right)\right]$ ou sa fonction de densité spectrale de puissance (PSD) $\Phi_{FF}(\omega) = \int_{-\infty}^{+\infty} \Sigma_{FF}(\tau) \mathrm{e}^{-j\omega\tau} \mathrm{d}\tau$. Par conséquent, sa fonction de densité de probabilité (PDF) est donnée par :

$$p(f) = \frac{1}{\sqrt{2\pi\sigma_{F0}^2}} \exp\left(-\frac{\left(f - \overline{F}\right)^2}{2\sigma_{F0}^2}\right)$$

où $\sigma_{F0}^2 = \Sigma_{FF}(0) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \Phi_{FF}(\omega)\,\mathrm{d}\omega$ est nommé le moment spectrale zéro ou la charge de variance (the variance load). Pour le présent exemple, la charge a une moyenne $\overline{F} =$ 25000 N en raison de la chaussée et d'un PSD qui peut être modélisé par une fonction logistique généralisée (Gillespie 1992) : $\Phi_{FF}(\omega) = a + \frac{k-a}{\left(1+q\exp\left(-b\left(\frac{\omega}{2\pi}-m\right)\right)\right)^{\frac{1}{\nu}}}$ avec $a = 0$, $k = 150{,}000$, $b = -0.15$, $q = k$, $m = 150$ et $\nu = 1.4$. Ce PSD est tracée dans la figure 3.2.

Du fait qu'on a considéré une structure linéaire, la réponse aléatoire est aussi gaussienne pour une charge gaussienne. Il suffit donc de caractériser cette réponse par sa moyenne $\bar{U}$ et la variance σ_{U0}^2. On a :

$$\bar{U} = h(\omega = 0) \times \overline{F}$$

ou :

$$\bar{U} = \left(\frac{L_1}{E_1 S_1} + \frac{L_2}{E_2 S_2} \right) \times \overline{F}$$

et $\sigma_{U0}^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \Phi_{UU}(\omega) \, \mathrm{d}\omega$ avec :

$$\Phi_{UU}(\omega) = h(\omega) \Phi_{FF}(\omega) h^{\mathrm{H}}(\omega)$$

où $\bullet^{\mathrm{H}}$ désigne l'opérateur Hermitien.

Pour la formulation du problème d'optimisation, la masse totale de la colonne donnée par :

$$m = \rho_1 S_1 L_1 + \rho_2 S_2 L_2$$

est fixée à une valeur $\hat{m}$. Cette limitation de la masse est due à la charge admissible qui est la capacité que peut supporter le sol et est donnée par une étude expérimentale géotechnique. Bien que cette étude expérimentale n'est pas faite, il serait intéressant de trouver la solution de conception optimale pour des valeurs arbitraires de plusieurs $\hat{m}$. En outre, il existe une limite inférieure pour les deux zones S_1 et S_2 pour assurer que le pilier reste dans le domaine élastique. Ces limitations doivent être évaluées à partir de la contrainte admissible qui est fonction de la limite d'élasticité en compression des matériaux considérés et les marges de sécurité utilisées dans les calculs. Dans notre cas d'étude, ceci est assuré si les deux surfaces sont supérieures à $0,0002\ m^2$. Dans les recherches d'ingénierie classiques, une stratégie d'optimisation mono-objectif consiste généralement à trouver des paramètres qui minimisent le déplacement moyen majoré d'une valeur donnée de l'écart type (trois fois l'écart-type est habituellement choisi pour une variable aléatoire gaussienne). Dans cette étude, on cherche à trouver le front de Pareto en résolvant un problème d'optimisation à deux fonctions objectif, la première étant la moyenne et la deuxième étant l'écart type. Ceci permettra de choisir la meilleure conception en fonction de contraintes techniques supplémentaires, tels que le meilleur compromis entre la contrainte moyenne qui doit être prise en compte par les matériaux et un niveau prescrit des vibrations, sachant que c'est un niveau acceptable pour le confort humain. Enfin, le problème d'optimisation peut être formulé sous la forme suivante :

$$\begin{cases} \min F(S_1, S_2, L_1) & = & \left(\bar{U}(S_1, S_2, L_1), \sigma_{U0}(S_1, S_2, L_1)\right)^T \\ \text{sujet} & & S_i \geq 0.0002,\ i = 1, 2, \\ & & 1 \leq L_1 \leq 15, \\ & & \rho_1 S_1 L_1 + \rho_2 S_2 (L - L_1) - \hat{m} = 0 \end{cases}$$

où $\hat{m}$ est un paramètre à choisir. Pour toute optimisation avec la masse totale fixée $\hat{m}$, les limitations de surface n'ont pas d'influence puisque l'optimum est atteint pour $L_1 = 1$ m, ce qui signifie que le pilier doit être réalisé avec de l'acier seulement(à l'exception de la fondation en béton imposée). Ceci a un sens physique si on considère la densité de chaque matériau et la limitation imposée sur la masse. Toutefois, le meilleur déplacement moyen et le meilleur écart-type sont obtenus pour différentes valeurs des variables de conception S_1 et S_2. Par exemple, lorsque la masse totale est fixée à $\hat{m} = 1{,}000$ kg, le meilleur déplacement moyen, qui a une valeur de $0.219mm$, est atteint pour $S_1 = 0,036$ m ^{2} et $S_2 = 0.0083$ m ^{2}, tandis que le meilleur écart-type, qui a une valeur de 0.0513 mm, est obtenue pour $S_1 = 0.056$ m^2 et $S_2 = 0.0079$ m^2. Lorsque la masse totale est fixée à $\hat{m} = 1.500$ kg, le meilleure déplacement moyen, qui a une valeur de 0.146 mm, est obtenu pour $S_1 = 0.053$ m^2 et $S_2 = 0.0125$ m^2, alors que le meilleur écart type de déplacement,

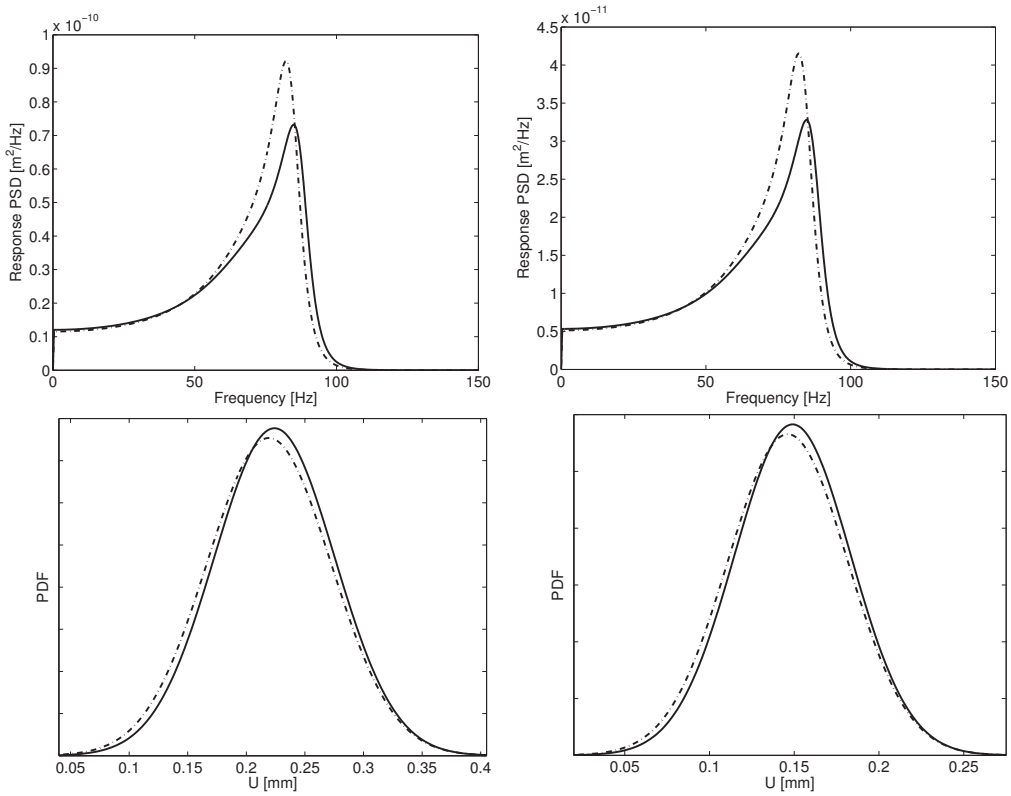


FIGURE 3.3 – *PSD et PDF pour 1,000 kg (à gauche) et 1,500 kg (à droite) du pilier*

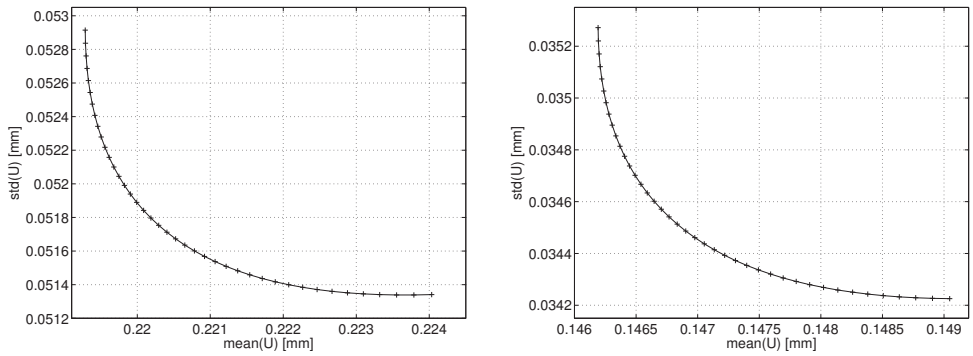


FIGURE 3.4 – *Front de Pareto pour 1,000 kg (à gauche) et 1,500 kg (à droite) du pilier*

qui a une valeur de 0.0342 mm, est atteint pour $S_1 = 0.082$ m^2 et $S_2 = 0.0118$ m^2. Comme prévu, toutes les fonctions objectif ont des valeurs plus faibles lorsqu'on considère une masse plus grande. Le PDF du déplacement pour ces deux cas de masses sont présentés sur la figure 3.3. Deux courbes sont tracées pour chaque situation, une (pour le meilleur écart-type) avec une ligne épaisse et l'autre (pour la meilleure moyenne) avec une ligne en pointillés pour montrer chaque optimum. Les fronts de Pareto pour ces deux cas de masse sont présentés sur la figure 3.4. Ces graphiques aident le décideur à choisir le meilleur compromis entre la moyenne et l'écart type de l'extrémité du pilier.

3.3 CONCEPTION DE LA STRUCTURE DE TREILLIS

Comme seconde application, une treilli à deux barres est considérée (voir figure 3.5). Les deux barres sont en acier. La barre horizontale a une aire constante S_1, tandis que l'autre barre a une aire constante S_2. Ces valeurs doivent être calculées en résolvant un problème d'optimisation multiobjectif sous des contraintes arbitraires sur les sections.

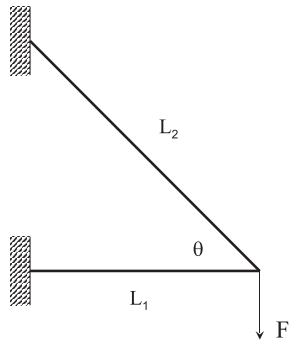


FIGURE 3.5 – *Schéma et dimensions du treillis à deux bars*

Dans cette étude, les extrémités gauches des deux barres sont fixées, tandis que les extrémités droites sont soumis à une moyenne plus une charge fluctuante.

On note par E et ρ les propriétés du matériau déterministes (module d'Young et la densité) de l'acier. La barres horizontale a une longueur L_1, tandis que l'autre barre a une longueur L_2 tel que $L_2 = \frac{L_1}{\cos(\theta)}$ où $\theta = \frac{\pi}{4}$ rd est l'angle entre les deux barres. Pour l'application numérique, il a été choisi $E = 2.1 \times 10^{11}$ Pa, $\rho = 7800$ kg/m^3 et $L_1 = 1$ m. les valeurs limites pour les sections sont définies de telle sorte que $2.2 < S_1 < 20$ et $5 < S_2 < 10$ si elles sont exprimées en mm ^{2}.

Chaque tige est modélisée en utilisant les éléments de la barre classique, impliquant une approximation affine du champ de déplacement sur chaque élément, voir par exemple Felippa (2004).
De l'assemblage des deux éléments finis et lorsque les conditions aux limites sont prises en compte, l'équilibre global est obtenu :

$$\left(\mathbf{K} - \omega^2 \mathbf{M}\right) \mathbf{u}(\omega) = \mathbf{F}(\omega) \tag{3.3}$$

ayant deux degrés de libertés $\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$ avec u et v sont le déplacement horizontal et vertical (respectivement) et avec la matrice de rigidité $\mathbf{K}$ et la matrice des masses $\mathbf{M}$ telles que :

$$\mathbf{K} = \begin{bmatrix} k_1 + k_2 \cos^2(\theta) & k_2 \cos(\theta)\sin(\theta) \\ k_2 \cos(\theta)\sin(\theta) & k_2 \sin^2(\theta) \end{bmatrix} \qquad \text{et} \qquad \mathbf{M} = m \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

où $k_1 = ES_1L_1$, $k_2 = ES_2L_2$ et $m = \frac{\rho}{3}(S_1L_1 + S_2L_2)$. Dans cette équation, le vecteur de charge agissant uniquement dans le sens vertical, et elle est telle que $\mathbf{F}(\omega) = \begin{bmatrix} 0 \\ f(\omega) \end{bmatrix}$.
Le système (3.3) peut être inversée pour provoquer le déplacement vertical, qui est le déplacement qui nous intéresse :

$$v(\omega) = h(\omega) \times f(\omega)$$

où $h(\omega)$ est la fonction de réponse en fréquence du treillis et est telle que :

$$h(\omega) = \left(\frac{k_2 \left(k_1 - \omega^2 m\right) \sin^2(\theta)}{k_1 - \omega^2 m + k_2 \cos^2(\theta)} - \omega^2 m \right)^{-1}$$

Notez que cette fonction de réponse peut avoir des pôles et des valeurs nulles pour certaines valeurs des paramètres de conception, conduisant à des fréquences de résonance et d'anti-résonance. Dans ce deuxième problème, il est intéressant de concevoir le trellis en tenant compte du processus de la charge verticale $F(t)$, où t désigne le temps. Cette charge est censée être gaussienne, suivre un processus aléatoire stationnaire et ergodique.

Ainsi, il pourrait être décrit par sa moyenne $\overline{F} = 10,000$ N et son PSD, qui peut être modélisé par une fonction constante sur une plage de fréquences limitée :

$$\Phi_{FF}(f) = \begin{cases} 2000 & \text{si } f_1 \leq f \leq f_2 \\ 0 & \text{sinon} \end{cases}$$

Trois bandes de fréquences, (f_1, f_2), seront pris en considération : $(0,200)$, $(100,300)$, $(600,800)$. Puisqu'on a considéré une structure linéaire, la réponse aléatoire est aussi gaussienne pour une charge gaussienne. Il suffit donc de caractériser cette réponse par sa moyenne μ_V et l'écart-type σ_V.

On a :

$$\mu_V = h(\omega = 0) \times \overline{F}$$

ou :

$$\mu_V = \left(\frac{1}{k_1 \tan^2(\theta)} + \frac{1}{k_2 \sin^2(\theta)} \right) \times \overline{F}$$

et $\sigma_V^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \Phi_{VV}(\omega)\, \mathrm{d}\omega$ avec :

$$\Phi_{VV}(\omega) = h(\omega)\, \Phi_{FF}(\omega)\, h^{\mathrm{H}}(\omega)$$

Dans ce problème, il est encore intéressant de trouver le front de Pareto corespndant à une optimisation bi-objectif, à savoir la moyenne et l'écart-type. Le problème d'optimisation est ainsi formulé sous la forme suivante :

$$\begin{cases} \min F(S_1, S_2) & = & (\mu_V(S_1, S_2), \sigma_V(S_1, S_2))^{\mathrm{T}} \\ \text{sujet} & & 22 \leq S_1 \leq 200 \\ & & 50 \leq S_2 \leq 100 \end{cases}$$

Notons que pour ce problème μ_V est une fonction convexe alors que σ_V ne l'est pas. Voir les figures3.7, 3.8 et 3.9 pour une courbe de contours de ces deux fonctions dans les bandes de fréquences $0-200$, $100-300$ et $600-800$ Hz. L'optimum de la moyenne, dans les trois cas, se situe aux limites des variables de conception, correspondant aux valeurs maximales ; c'est le point a dans les figures 3.7 et 3.8 ,3.9. C'est une conception très logique puisque plus la barre est lourde plus la rigidité du treillis est plus élevée, ce qui réduit par conséquent son déplacement global. D'autre part, la fonction de variance a deux minimum, situés à la limite du domaine. L'un d'eux, le point a, est un minimum local correspondant à la conception $\{S_1, S_2\} = \{200, 100\}$, l'autre étant un minimum global, c'est le point c, correspondant à la conception $\{S_1, S_2\} = \{22, 100\}$. En outre, notons que cet optimum global se trouve dans une vallée profonde. L'interprétation physique d'un tel comportement provient de la présence ou non de la fréquence d'anti-résonance dans la fréquence excitée. Afin de mieux comprendre cette situation, le PSD complet sur une large plage de fréquences est présenté sur la figure 3.6, indiquant l'emplacement des fréquences de résonance et d'anti-résonance pour les deux conceptions optimales.

Le front de Pareto de ce treillis à deux barres pour la bande de fréquence $(600,800)$ Hz est présenté sur la figure 3.10. Remarquons que la courbe de Pareto dans la figure de droite présente un saut, ce qui peut être assimilé à une discontinuité ou un front de Pareto local, tandis que la figure de gauche présente un front lisse régulier de Pareto. A partir du point situé à gauche, $a = (1.58, 0.605)$, qui est un point de Pareto, le point de Pareto suivant de la courbe est celui immédiatement en dessous du premier point de Pareto. Les points intermédiaires ne sont pas des points de Pareto. À partir du point a, qui correspond à $(S_1, S_2) = (200, 100)$, le point suivant dans la figure 3.10, $b = (2.73, 0.65)$ correspond à $(S_1, S_2) = (200, 54)$ et la courbe correspond à $S_1 = 200$. La discontinuité se produit et le point $b' = b$, qui correspond maintenant à $(S_1, S_2) = (34.4, 100)$ et commence une autre courbe jusqu'au point $c = (3.51, 0.472)$ qui correspond à $(S_1, S_2) = (22, 100)$ et cette

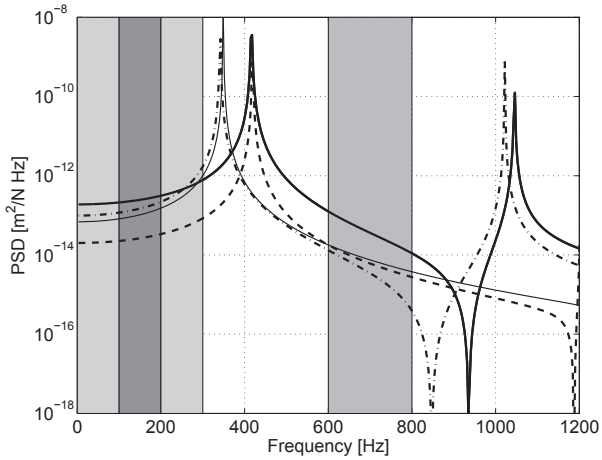


FIGURE 3.6 – *PSD du treillis à deux barres pour quatre extremum ; Ligne continue épaisse : $S_1 = 22\ mm^2$ et $S_2 = 50\ mm^2$; traits interrompus : $S_1 = 22\ mm^2$ et $S_2 = 100\ mm^2$; ligne continue mince : $S_1 = 200\ mm^2$ et $S_2 = 50\ mm^2$; ligne en pointillés : $S_1 = 200\ mm^2$ et $S_2 = 100\ mm^2$*

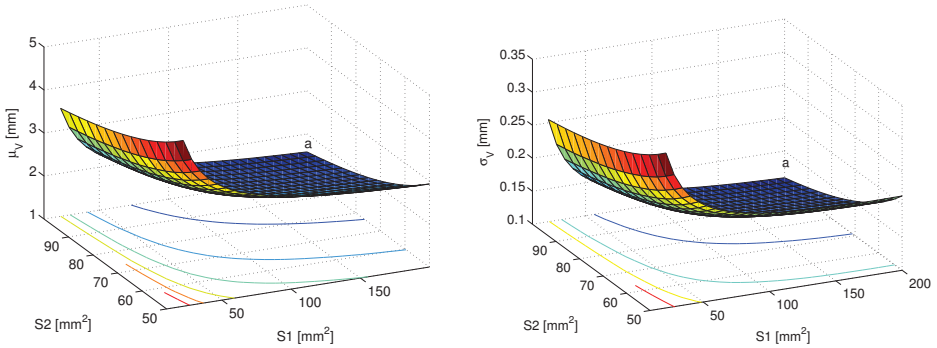


FIGURE 3.7 – *Surfaces de réponses du treillis à deux barres pour la moyenne (gauche) et l'écart type (droite) pour la bande de fréquence [0-200] Hz*

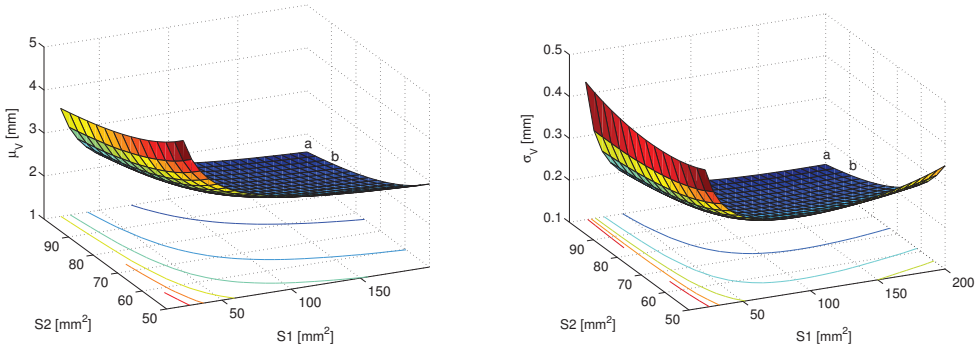


FIGURE 3.8 – *Surfaces de réponses du treillis à deux barres pour la moyenne (gauche) et l'écart type (droite) pour la bande de fréquence [100-300] Hz*

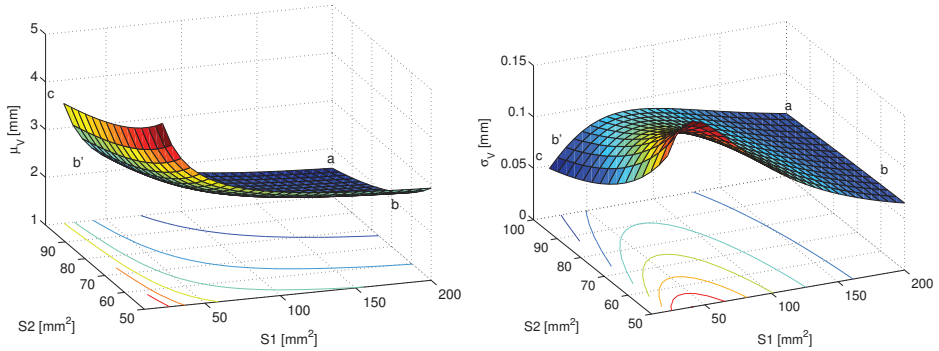


FIGURE 3.9 – *Surfaces de réponses du treillis à deux barres pour la moyenne (gauche) et l'écart type (droite) pour la bande de fréquence [600-800] Hz*

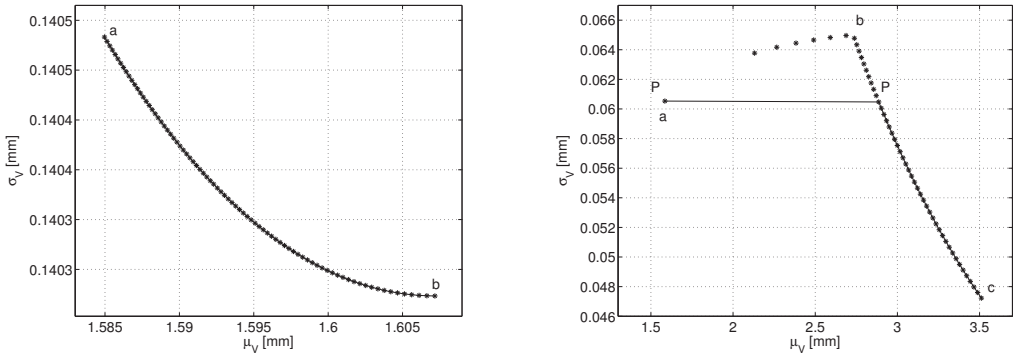


FIGURE 3.10 – *Front de Pareto pour le treillis à deux barres ; μ_V représente la moyenne de V et σ_V représente son écart type.*

courbe à $S_2 = 100$. Comme on peut le voir la discontinuité dans la courbe de Pareto correspond à une discontinuité dans les variables de conception. Ceci est clairement visible sur la figure 3.9. Cette courbe de Pareto permet de choisir le meilleur compromis entre la moyenne et l'écart type du déplacement vertical du treillis.

3.4 EFFICACITÉ ET EFFICIENCE DE RFNM

3.4.1 Influence de la fonction de Pincus

Dans la formule de représentation, l'expression Pincus correspond à $g(\lambda, s) = e^{-\lambda s}$, ce qui est un choix convenable. Dans notre expérience, quatre autres fonctions de descente (continues et strictement décroissantes) ont été utilisées pour résoudre le problème de conception de la structure de treillis, en utilisant le RFNM, NBI et la méthode de pénalité :

$$g_1(\lambda, s) = \frac{1}{\lambda\ s^3}$$

$$g_2(\lambda, s) = \frac{1}{\lambda\ ln(s)}$$

$$g_3(\lambda, s) = \frac{e^{-\lambda}}{s^3}$$

$$g_4(\lambda, s) = 10^{(-\lambda\ s)}$$

Les tests montrent que le choix de la fonction n'a pas d'effet significatif sur la qualité de la solution et de la performance des algorithmes utilisés. De faibles différences en termes de temps d'exécution ont été observées.

3.4.2 Comparaison avec GA, PSO et NSGA II

Dans cette section, l'efficience et l'efficacité de RFNM seront comparés avec l'algorithme génétique GA standard et l'algorithme PSO pour une population aléatoire initiale ainsi que pour une population générée par la formule de représentation (RFGA et RFPSO respectivement). D'autre part, elles seront comparées à GA en utilisant l'algorithme de Nelder Mead (NM) à l'étape de mutation (RFGANM). Dans tous les cas, les méthodes NBI et la pénalité sont utilisées pour la génération du front de Pareto et l'élimination des contraintes engendrées. Par ailleurs, une comparaison avec la méthode multi-objectif NSGA II est présentée. Les paramètres utilisés dans l'algorithme génétique sont : taille de la population, 20 et 40, le taux de mutation est fixé à 0.2 ; la méthode de sélection est la pondération du rang (rank weighting) ; les critères d'arrêt sont l'itération maximale (fixée à 5000 pour l'AG) et le nombre maximum d'itérations sans amélioration de la solution est fixé à 300 pour l'AG. En ce qui concerne NM, les paramètres standard recommandées par les auteurs ont été adoptées, les critères d'arrêt utilisées sont : nombre maximum d'évaluations de la fonction, $maxfun = 5000$, maximum d'itération, $maxiter = 1000$, la tolérance sur la variation de la fonction valeur $Tolf = 10^{-6}$, et la tolérance sur la variation des variables x est $tolx = 10^{-6}$. Pour PSO, les paramètres suivants sont adoptés : taille de la population, 40 et 60, les critères d'arrêt sont le nombre d'itérations maximal (fixé à 600), et le nombre maximum d'itérations en continue sans amélioration de la solution est fixé à 100, et le nombre de voisins est pris égale à 11. En ce qui concerne la taille de l'échantillon pour la formule de représentation, il est fixé à 200. Tous ces algorithmes seront appliqués pour générer 41 points de front de Pareto pour le problème de conception du treillis en utilisant la méthode NBI (avec $e = (1, 0.3)^T$) au lieu de la valeur standard) et la méthode de pénalité. L'algorithme NSGA II est utilisé avec les paramètres suivants : la probabilité de croisement $= 0.8$; fraction de Pareto $= 0.35$; indice de distribution de mutation $= 100$, l'indice de distribution de croisement $= 20$. Les algorithmes proposés sont programmées sous Matlab 7.0 pour travailler sous Windows XP avec un processeur Intel Core 2 Duo à 2.0 GHz et 2 Go de RAM. Les résultats présentés sont en termes de taux de succès (SR), le temps moyen CPU (en secondes) et la moyenne de l'erreur, notée Err. Le terme SR est le pourcentage de réussite de trouver un point de Pareto c-à-d quand l'algorithme obtient une solution avec la précision requise, où la *précision requise* est une valeur maximale donnée calculée comme étant la différence absolue entre la solution trouvée et l'optimum global divisé par la solution globale (quand il est non nul). La précision choisie est fixée à 10^{-4}. L'erreur moyenne est Err :

$$Err = \frac{1}{2}\left(\frac{1}{41}\left(\sum_{i=1}^{41}\frac{|\mu_i - \mu_i^*|}{\mu_i^*} + \sum_{i=1}^{41}\frac{|\sigma_i - \sigma_i^*|}{\sigma_i^*}\right)\right)$$

où μ_i et σ_i sont la moyenne et l'écart type pour le i^{th} point Pareto et μ_i^* et σ_i^* sont les meilleurs points Pareto connus au même point.

Les résultats de la comparaison sont exposés sur les figure 3.11, 3.12, 3.13(a) et dans le tableau 3.1. Au cours des simulations, il a été constaté que la taille de la population a un effet significatif sur la performance de l'AG, RFGA, PSO et RFPSO (SR est plus grand et Err est plus petite). L'utilisation de la formule de représentation afin de générer la population initiale n'a amélioré que la méthode PSO. On peut voir que PSO est plus performant que le GA, et la convergence pour GA, RFGA, PSO et RFPSO est lente. Le résultat obtenu par la méthode GANM est meilleur par rapport à GA, RFGA, PSO et RFPSO ($SR = 90\%$), pour un temps d'exécution plus grand. Comme pour la plupart des méthodes d'optimisation locale, le choix de la population initiale a une importance majeure sur les résultats obtenus par la méthode NM. En effet, l'erreur moyenne obtenue pour NM est la plus grande. Le NSGAII parvient à générer la frontière de Pareto pour une population supérieure à 300, mais les points obtenus ne sont pas uniformément réparties (voir figure 3.13 (b)). Les résultats montrent que la méthode RFNM est une alternative

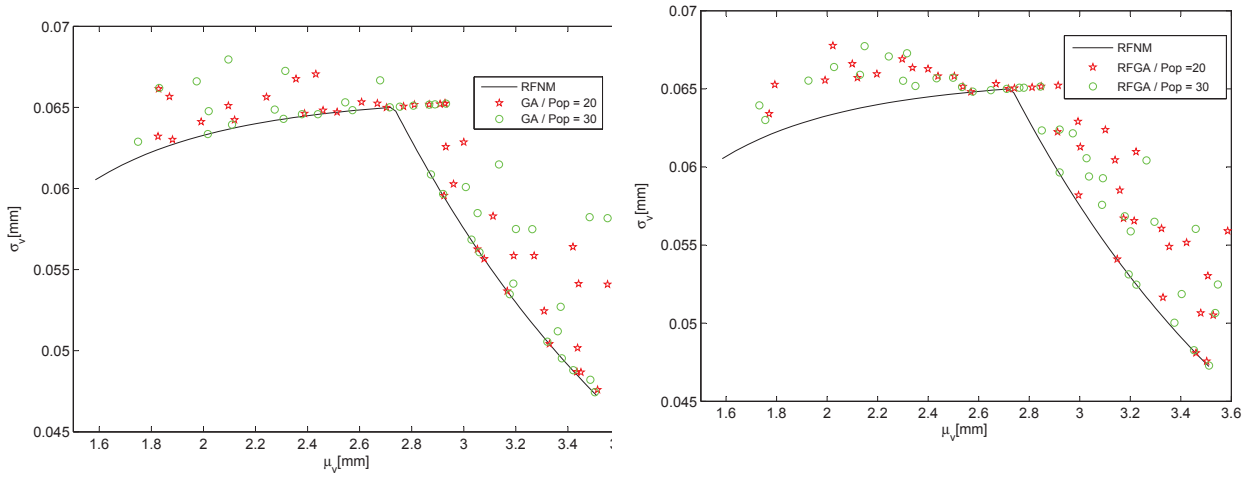


FIGURE 3.11 – *Front de Pareto pour treillis à deux barres, en utilisant GA(a) et RFGA(b).*

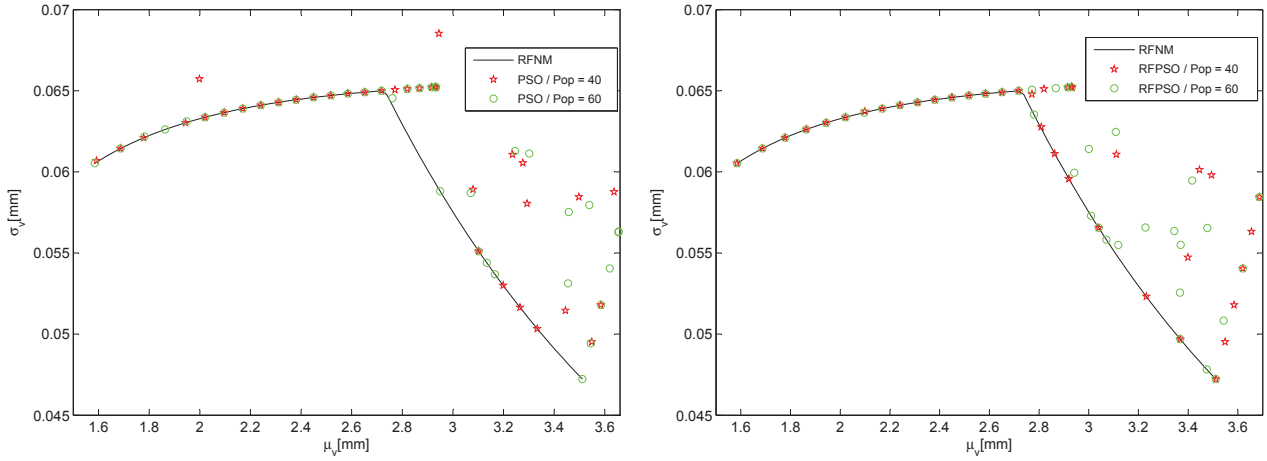
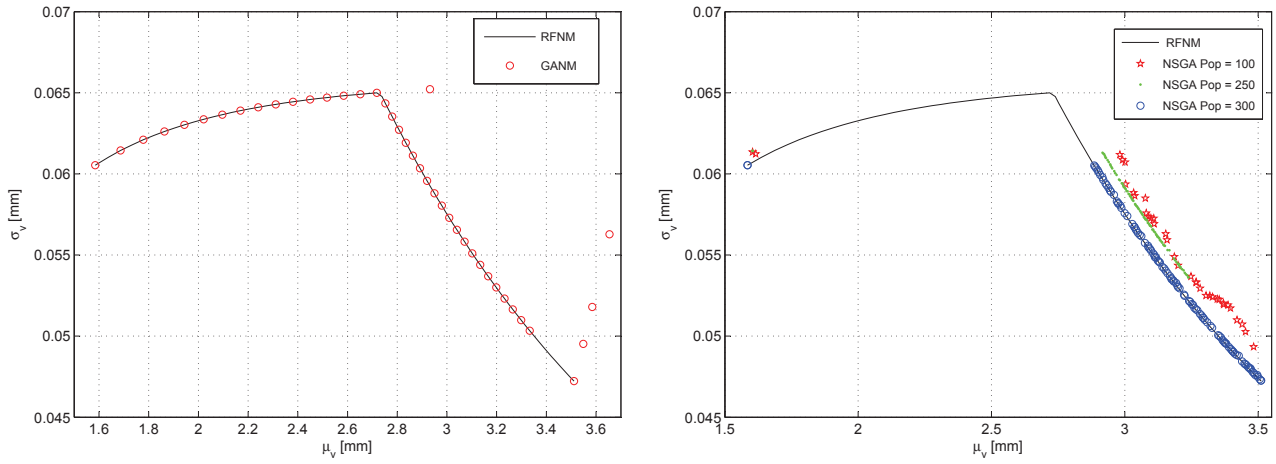


FIGURE 3.12 – *Front de Pareto pour treillis à deux barres, en utilisant PSO(a) et RFPSO(b).*

TABLE 3.1 – *Comparaison entre les méthodes*

Algorithm	Pop	SR	Err	TCPU
GA	20	21.95%	2.80 %	110
GA	30	36.59%	2.76 %	160
RFGA	20	14.63%	3.66 %	92
RFGA	30	21.95%	3.15 %	155
GANM	30	90.24%	2.13 %	640
PSO	40	46.34%	4.13 %	160
PSO	60	53.66%	3.08 %	246
RFPSO	40	56.10%	3.26 %	213
RFPSO	60	56.10%	2.17%	329
NM	–	46.34%	10.55%	1.32
RFNM	–	100%	0.00 %	12.34



(a) (b)

FIGURE 3.13 – *Front de Pareto pour treillis à deux barres, en utilisant GANM(a) et NSGA II(b).*

viable par rapport aux autres métaheuristiques pour résoudre ce problème ($SR = 100\%$ et $Err = 0\%$) avec une convergence plus rapide et précise par rapport à GA et PSO.

3.5 CONCLUSION

Une méthodologie pour transformer un problème linéaire avec charge gaussienne en un problème multi-objectif a été proposée et appliquée à deux exemples de conception mécanique. Pour résoudre le problème d'optimisation qui en résulte, une nouvelle hybridation, l'algorithme RFNM, a été proposée. L'approche est basée sur la combinaison d'une formule modifiée de représentation de solution de Pincus et l'algorithme de Nelder-Mead. L'avantage principal de cette hybridation est la grande fiabilité et la stabilité de l'algorithme. Nous avons utilisé la méthode NBI pour transformer le problème à deux fonctions objectif en une séries de problèmes mono-objectif et l'algorithme RFNM pour les résoudre et construire la courbe de Pareto. Les résultats ont montré que RFNM était en mesure de localiser l'optimum global avec une grande précision et sa simplicité pour l'application aux problèmes généraux de conception. Le problème du treillis à deux barres a une caractéristique intéressante : en fonction de la bande de fréquence de la charge, la forme de la courbe de Pareto varie d'un seul point pour la bande de fréquence $(0, 100)$, à une courbe de Pareto discontinu pour la bande $(600, 800)$. Pour l'exemple traité, RFNM est plus précis, fiable et efficace pour localiser l'optimum global que les autres alternatives analysées. Avec les tests réalisés jusqu'à présent, ce résultat semble être vrai, cependant, d'autres tests sont prévus, pour des problèmes plus complexes, et les résultats sont publiés dans le prochain chapitre.

CHAPITRE 4

NNC AMÉLIORÉ POUR LA CONCEPTION D'UN DISPOSITIF FIABLE DE SUSPENSION DYNAMIQUE AVEC RIGIDITÉ DYNAMIQUE ALÉATOIRE ET UNIFORME

SOMMAIRE

5.1 INTRODUCTION 87
5.2 NOTATION 88
5.3 MESURES FINIES DANS UN ESPACE DE HILBERT À DIMENSION INFINIE 89
5.4 INTÉGRATION AVEC UNE MESURE SUR UN ESPACE DE HILBERT 89
5.5 HYPOTHÈSES 91
5.6 UNE REPRESENTATION DE L'OPTIMUM GLOBAL 91
5.7 APPROCHE NUMÉRIQUE 95
5.8 RÉSULTATS EXPÉRIMENTAUX 97
5.8.1 Influence de la fonction de Pincus 98
5.8.2 Problèmes élémentaires 98
5.8.3 Problème du Brachistochrone 100
5.8.4 Problème Exponentiel 100
5.8.5 Problème de la chaînette 102
5.8.6 Géodésiques dans un cylindre 102
5.9 CONCLUSION 103

Ce chapitre traite de la conception d'un dispositif de suspension, modélisé en tant qu'un système ressort-masse-amortisseur. L'amplitude d'un système nominal est contrainte de satisfaire certaines limites dans une bande de fréquence donnée et la conception doit être faite comme une optimisation sous contrainte fiabiliste. Pour se concentrer sur l'idée principale, seule la raideur du système sera considéré comme aléatoire. La raideur est caractérisé par une variable aléatoire uniforme, et sa moyenne et son écart-type sont les paramètres d'optimisation. On cherche à minimiser la raideur

moyenne et à maximiser l'écart-type, alors que la réponse en amplitude doit être comprise dans le domaine acceptable de vibration prescrit. Pour générer le front de Pareto, une amélioration de la méthode Normalized Normal Constraint(INNC) est proposée en utilisant RFNM. Les résultats montrent qu'une courbe discontinue de Pareto est obtenue pour un certain choix de la contrainte. Par conséquent, dans cet exemple simple, on montre que des situations difficiles peuvent se produire pendant la conception de systèmes dynamiques lors de la prescription de la coque de réponse d'amplitude. Malgré la simplicité de l'exemple traité ici, choisi pour mettre en évidence les idées principales sans distraction, la stratégie proposée ici peut être généralisée pour les cas plus complexes et de donner des résultats valables, capables d'aider les concepteurs à choisir le meilleur compromis entre la moyenne et l'écart type dans le domaine de la conception fiabiliste. Une comparaison des résultats obtenus en termes de temps d'exécution et nombre d'évaluations de fonctions objectif avec d'autres méthodes d'optimisation multi-objectif telles que NBI, NNC, NSGA II et SPEA2 a été réalisée.

Ce chapitre est organisé comme suit : la section 4.2 présente des généralités sur la conception mécanique lors de l'examen de la fiabilité des structures, tandis que la section 4.3 est consacrée aux équations du problème de vibration mécanique. La formulation du problème d'optimisation sous contrainte fiabiliste est mis en évidence dans la section 4.4. Dans la section 4.5 la notion de propagation des incertitudes pour le système mécanique est introduite, ensuite une amélioration de la méthode NNC est proposée dans la section 4.6 suivie d'applications dans la section 4.7 et une comparaison avec d'autres méthodes d'optimisation multiobjectif dans la section 4.8, enfin une conclusion et perspectives sont présentées dans la section 4.9.

Les principales contributions en rapport avec ce chapitre est :

H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. **Reliability-Based Design Of A Dynamical Suspension With Random Stiffness Under An Imposed Amplitude Response Hull**, *Mecánica Computacional*, (ISSN 1666 – 6070), Vol XXXI, p. 2765 – 2781 (2012).

H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. **Improved Normalized Normal Constraint Multiobjective Optimization Method for the Reliable Design of a Dynamical Suspension Device**, *Congress on Numerical Methods in Engineering 2013*, Bilbao, Spain, 25-28 June, 2013.

4.1 Introduction

On considère la conception d'une suspension, modélisée par un système ressort-masse-amortisseur, ayant un comportement linéaire, conduisant à un système dynamique simple à un degré de liberté (simple-degree-of-freedom) (SDOF). Le paramètre de conception qui nous interesse est la raideur du ressort, tandis que la contrainte de conception consiste en la prescription d'une courbe donnant la coque de réponse d'amplitude acceptable pour les réponses du système sur une gamme de fréquence donnée. Et ceci, dans la plage de fréquence donnée, la fonction de réponse en fréquence du système pour chaque valeur de la rigidité doit être à l'intérieur de la région prescrite.

Dans notre problème, on considère des incertitudes dans le système, les variables de conception traitées sont des variables aléatoires, dans note cas, la raideur du système. Plus précisément, nous avons choisi d'étudier le cas particulier d'une distribution limitée pour la raideur, ce qui nous amène à envisager une distribution uniforme, une conséquence

du principe de l'entropie maximale(Maximum Entropie Principle). Ensuite, la raideur peut être caractérisée par sa moyenne et son écart type, qui deviennent les paramètres de conception. En outre, puisque la conception est une optimisation fiabiliste, il faut raisonner en termes de probabilité d'acceptation pour les contraintes de conception. Ainsi, la probabilité acceptable de défaillance spécifiée par le concepteur est un paramètre supplémentaire du problème.

Pour faire la distinction entre les nombreuses solutions de conception, les fonctions objectif doivent être définies, afin de construire le problème d'optimisation sous ses formes standards. Deux fonctions objectif sont choisies : On cherche une suspension ayant la plus faible raideur moyenne, afin de maintenir bas le coût du matériel, et le plus grand écart-type, afin de maintenir bas les coûts de fabrication. Ainsi, la moyenne et l'écart type de la raideur aléatoire sont les objectifs de conception.

4.2 Généralités sur la conception mécanique lors de l'examen de la fiabilité des structures

Quand une structure est chargée, il se déforme et se développe des contraintes internes. Cette déformation et contraintes internes doivent être situés dans certaines limites qui caractérisent la résistance du matériau. Et ceci dans les limites de l'intégrité et le bon fonctionnement de la structure. En dehors des limites, on dit que la structure est en défaillance. La frontière entre ces deux situations est appelé l'état limite (limit-state). Un exemple simple de l'état limite ultime est la contrainte de Von Mises s, comparée à une résistance acceptable r du matériau de structure. Des exemples de l'état limite d'utilisation peut être une déviation maximale ou une vibration excessive qui ne dépassent pas un seuil de confort humain. En plus, en considérant qu'une structure complète exige l'analyse non seulement d'un seul point spacial, mais de tous ses champs pour assurer sa conception. Cela nous amène à définir un vecteur d'états limites lors de l'adoption d'une description spatiale numérique associée à un maillage de la partie mécanique pour un champ. Ainsi, la tâche d'ingénierie consiste généralement à trouver la conception nominale décrite par l'ensemble des paramètres $\mathbf{x} \in \mathbb{R}^n$ qui optimisent un vecteur de fonctions objectif $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^q$, sous réserve d'un vecteur de critères de défaillance $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$. Une formulation typique peut s'écrire ainsi :

$$\min_{\mathbf{x}} (\mathbf{f}(\mathbf{x}))$$
$$\mathbf{g_j}(\mathbf{x}) = \mathbf{r_j}(\mathbf{x}) - \mathbf{s_j}(\mathbf{x}) > \mathbf{0} \qquad j = 1, ..., m$$

Où $\mathbf{s}$ est la charge et $\mathbf{r}$ est la résistance, m est le nombre de points de contrôle (spatiale) sur la structure. Cependant, la conception de la solution qui satisfait cette formulation ne prend pas en compte le caractère incertain, ce qui implique la possibilité d'avoir des réponses structurelles indésirables en leurs présences. Ainsi, pour gérer les incertitudes sur la structure ou les charges, il est préférable de travailler en termes de fiabilité. Dans ce cas, nous introduisons l'ensemble des variables aléatoires $\mathbf{Y}$ qui doit être considéré dans la conception de la structure, et la structure sera considérée comme fiable si la probabilité de défaillance de l'état-limite est supérieure à une valeur prescrite. Les fonctions d'état limite $\mathbf{G}$ et la probabilité de défaillance $\mathbf{P}_f$ sont définis comme suit :

$$\mathbf{G}(\mathbf{x}, \mathbf{Y}) = \mathbf{R}(\mathbf{x}, \mathbf{Y}) - \mathbf{S}(\mathbf{x}, \mathbf{Y})$$
$$\mathbf{P}_f = \text{Prob}[\mathbf{G}(\mathbf{x}, \mathbf{Y}) < \mathbf{0}]$$

Les deux fonctions $\mathbf{R}$ et $\mathbf{S}$ sont maintenant des fonctions de conception nominale $\mathbf{x}$ et la variable aléatoire $\mathbf{Y}$. La région de défaillance est délimitée par $\mathbf{G} < \mathbf{0}$, alors que $\mathbf{G} = \mathbf{0}$

et $\mathbf{G} > \mathbf{0}$ indiquent la surface de défaillance et la région de sécurité, respectivement. La probabilité de défaillance est :

$$\text{Prob}\left[G_j\left(\mathbf{x},\mathbf{Y}\right) < \mathbf{0}\right] = \int \ldots \int_{G_j(\mathbf{x},\mathbf{Y}) \leq \mathbf{0}} p_j\left(\mathbf{Y}\right) \mathrm{d}\mathbf{Y}$$

où p_j désigne la fonction de densité de probabilité commune de la fonction de l'état limite j lors de l'examen de toutes les variables aléatoires pertinentes $\mathbf{Y}$. Notons que la probabilité de non-défaillance $\mathbf{P}_r$ est la suivante :

$$\mathbf{P}_r = 1 - \mathbf{P}_f$$

4.3 EQUATIONS DU PROBLÈME DE VIBRATION MÉCANIQUE

Les vibrations peuvent souvent conduire à des résultats indésirables, tels que l'inconfort ou de la fatigue des passagers d'une voiture dont la suspension est à concevoir. La défaillance structurale et mécanique peut souvent être causée par des vibrations soutenues. Dans cette étude, nous nous intéressons à la conception d'un ressort semblable au dispositif de suspension d'une voiture ou d'une moto. Pour limiter le risque de vibrations, des spécifications sont définies pour la quantité de vibrations qu'un appareil peut supporter. Ainsi, au cours de la conception, il est intéressant de régler les paramètres physiques du système de telle sorte que la réponse aux vibrations présente le niveau de crête spécifié proposé par le cahier des charges. En règle générale, une coque de niveaux de pics acceptable de vibration sont établies dans le domaine fréquentiel et sont exprimés en termes d'accélérations, mais elle peut être également exprimée en termes des déplacements de crêtes sans difficulté.
Ce dispositif est modélisé dans notre cas par un system linéaire simple masse-ressort-amortisseur fixé à une extrémité et soumis à un déplacement harmonique imposé z à l'autre extrémité (voir la figure 4.1). Dans le domaine fréquentiel, le déplacement de ce degré de liberté simple (SDOF),u, est donnée par Lin (1967) :

$$\left(k - m\left(2\pi f\right)^2 + \mathrm{j}2\pi f c\right) u\left(f\right) = q\left(f\right) \tag{4.1}$$

$$\text{avec :} \quad q\left(f\right) = m\left(2\pi f\right)^2 z\left(f\right) \tag{4.2}$$

où f est la fréquence. Dans cette équation, k, m et c sont la raideur, la masse et les paramètres du système d'amortissement (respectivement), et le déplacement $u\left(f\right)$ est obtenu par la résolution de l'équation (4.1), ce qui conduit à :

$$u\left(f\right) = \frac{q\left(f\right)}{k - m\left(2\pi f\right)^2 + \mathrm{j}2\pi f c} \tag{4.3}$$

Contrairement à un problème mécanique générale qui est continu dans la dimension de l'espace (comme il est décrit dans la section précédente), ce système simple a un seul degré de liberté spatiale. Pour étudier ce système, il est intéressant d'introduire la fonction de réponse en fréquence (FRF) $h\left(f\right)$ donnée par unité de force, conduisant à :

$$h\left(f\right) = \frac{1}{k - m\left(2\pi f\right)^2 + \mathrm{j}2\pi f c} \tag{4.4}$$

qui a l'amplitude :

$$\left|h\left(f\right)\right| = \frac{1}{\sqrt{\left(k - m\left(2\pi f\right)^2\right)^2 + \left(2\pi f c\right)^2}} \tag{4.5}$$

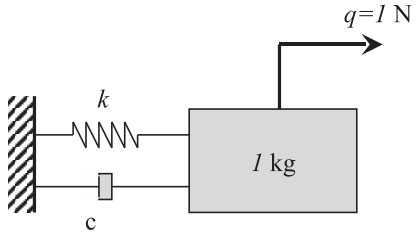


FIGURE 4.1 – *SDOF system*

Ce système devient stochastique lorsque le paramètre de raideur ou les charges (ou les deux) ne sont plus déterministes. Dans la suite, les variables aléatoires seront notées en lettre majuscule pour représenter la variable déterministe ; K et U dans ce cas pour la raideur et l'amplitude de réponse. La formulation adoptée dans ce travail est basée sur des amplitudes de crête de la réponse du système.
Considérant le dispositif a une raideur aléatoire, l'amplitude de la réponse du système est une variable aléatoire, ce qui conduit à étudier :

$$U(f) = \frac{1}{\sqrt{\left(K - m(2\pi f)^2\right)^2 + (2\pi f c)^2}}$$

pour une fonction unité de force. En plus, nous avons choisi de prendre une unité de masse pour les applications numériques. Dans ce travail, la loi de probabilité de la variable aléatoire K est choisi dans le principe de l'entropie maximale (Kapur et Kesavan 1992, Rubinstein et Kroese 2008). Ce principe annonce que la probabilité uniforme maximise l'entropie dans le cas du domaine borné pour une variable aléatoire. Ainsi, lorsque K suit une distribution uniforme, $K \sim U(\mu_k, \sigma_k)$, sa fonction de densité de probabilité (PDF) est donnée par la valeur constante $p_K(k) = \frac{1}{2\sqrt{3}\sigma_k}$ sur $[\mu_k - \sqrt{3}\sigma_k, \mu_k + \sqrt{3}\sigma_k]$.

4.4 FORMULATION DU PROBLÈME D'OPTIMISATION SOUS CONTRAINTE FIABILISTE

A partir de la description du problème dans la section précédente, l'ensemble des variables aléatoires est $\mathbf{Y} = \{K, U(f)\}$ et l'ensemble des paramètres de conception est $\mathbf{x} = \{\mu_k, \sigma_k\}$ qui représentent les deux paramètres de la distribution affectés à la raideur du ressort. Notons que μ_k doit être strictement positif, tandis que σ_k est non-négatif. D'un point de vue conception, un objectif intéressant est de concevoir des ressorts avec la plus faible raideur qui correspond à un coût moindre, mais aussi autoriser une grande dispersion sur la conception nominale lors de la fabrication de ressorts (mois d'exigence sur la qualité). Ainsi, le problème d'optimisation multi-objectif qui en découle est celui qui minimise la moyenne de la raideur du ressort μ_k et simultanément maximise son écart-type σ_k, ce qui réduit le coût total de fabrication. De plus, l'amplitude de la réponse en vibration doit respecter la limite pour le niveau de crête proposée par le cahier des charges, au moins pour une valeur de probabilité de défaillance choisi $P_{f\,\text{admissible}}$. Ceci définit l'état limite comme suit :

$$f \in [0, f_{\max}]$$

où $u_{\max}(f)$ est la borne de la limite d'amplitude crête donnée. Mais pour traiter numériquement cet état limite continue, on le remplace par un ensemble d'états limite définies à m fréquences fixes :

$$j \in [1, ..., m]$$

qui sont collectées dans le vecteur $\mathbf{G}$ des états limites, comme cela est fait pour gérer les états limites de l'espace des champs mécaniques lors de la considération des contraintes

statiques ou des déformations des solides continues. Par conséquent, le problème d'optimisation avec contraintes fiabilistes a la forme suivante :

$$\begin{cases} \min \mathscr{F}(\mu_k, \sigma_k) & = & (\mu_k, -\sigma_k)^{\mathrm{T}} \\ \text{sujet} & & P_f = \mathrm{Prob}\left[u_{\max}(f_j) - U(\mu_k, \sigma_k, K; f_j) < 0\right] \leq P_{f\,\text{admissible}}, \\ & & \qquad\qquad\qquad\qquad\qquad\qquad \text{pour } j = 1, \ldots, m \\ & & \mu_k > 0,\, \sigma_k \geq 0 \\ & & K \sim U(\mu_k, \sigma_k). \end{cases}$$

4.5 Propagation des incertitudes pour le système mécanique

Dans notre problème, les pics limite $u_{\max}(f_j)$ sont considérés comme déterministes pour chaque fréquence fixe f_j. Ainsi, nous pouvons écrire la probabilité de non-défaillance comme :

$$\mathrm{Prob}\left[U(\mu_k, \sigma_k, K; f_j) \leq u_{\max}(f_j)\right] = P_U(u_{\max}; f_j)$$

où P_U désigne la fonction de distribution cumulative de la réponse en amplitude du système U à la fréquence fixe f_j, donc :

$$\mathrm{Prob}\left[U(\mu_k, \sigma_k, K; f_j) \leq u_{\max}(f_j)\right] = \int_0^{u_{\max}(f_j)} p_U(u; f_j)\,\mathrm{d}u = \int_{u_{\inf}(f_j)}^{u_{\max}(f_j)} p_U(u; f_j)\,\mathrm{d}u$$

$P_U(u; f_j)$ est la fonction de densité de probabilité (PDF) du système et $u_{\inf}(f_j) = \inf\left\{u : p_U(u; f_j) > 0\right\}$ est l'extrémité gauche du support de P_U. Dans la réalité, la fonction de distribution est liée au système des variables aléatoires. Par conséquent, compte tenu d'un ensemble $\mathbf{Z}$ des variables aléatoires du système et la fonction de réponse en fréquence du système h, nous avons :

$$P_U(u; f) = \int \ldots \int_{U(f) \leq u} p_{\mathbf{Z}} \mathrm{d}\mathbf{Z}$$

où $p_{\mathbf{Z}}$ est le PDF conjointe des variables $\mathbf{Z}$. Pour le problème considéré, nous avons une seule variable aléatoire K. Lorsque K suit une distribution uniforme, sa PDF est donnée par la valeur constante $p_k(k) = \frac{1}{2\sqrt{3}\sigma_k}$ sur $[k_{\inf} = \overline{k} - \sqrt{3}\sigma_k, k_{\sup} = \overline{k} + \sqrt{3}\sigma_k]$. Comme il n'y a qu'une seule variable aléatoire, la formule précédente est simplifiée, permettant l'évaluation analytique de la réponse en amplitude du système (Zwillinger et Kokoska 2000) :

$$p_U(u) = \frac{1}{\left|\frac{\mathrm{d}U}{\mathrm{d}K}\right|_{k_1}} p_K(k_1) + \cdots + \frac{1}{\left|\frac{\mathrm{d}U}{\mathrm{d}K}\right|_{k_n}} p_K(k_n) \tag{4.6}$$

utilisant $p_U(u) = \frac{\mathrm{d}P_u}{\mathrm{d}u}$. Dans cette expression, k_j pour $j = 1, \ldots, n$ désigne les racines de l'équation algébrique $u(k, f) = u$, pour f fixe (à noter que $j = 1$ pour une fonction bijective). Cela produit (Pagnacco et al. 2011) :

$$p_U(u, f) = \begin{cases} \frac{1}{u\sqrt{u - (2\pi f c u)^2}} \times p_K(k) & \text{si} \quad u_{\inf}(f) \leq u(f) \leq u_{\sup}(f) \\ 0 & \text{sinon} \end{cases}$$

où $u_{\sup}(f)$ est l'enveloppe supérieure de la réponse du système qui est donnée par :

$$u_{\sup}(f) = \begin{cases} \frac{1}{\sqrt{\left(k_{\inf} - m(2\pi f)^2\right)^2 + 4\pi^2 c^2 f^2}} & \text{pour } f^2 \in \left[0, \frac{k_{\inf}}{m}\right] \\ \frac{1}{2\pi f c} & \text{pour } f^2 \in \left[\frac{k_{\inf}}{m}, \frac{k_{\sup}}{m}\right] \\ \frac{1}{\sqrt{\left(k_{\sup} - m(2\pi f)^2\right)^2 + 4\pi^2 c^2 f^2}} & \text{pour } f^2 \in \left[\frac{k_{\sup}}{m}, +\infty\right[\end{cases}$$

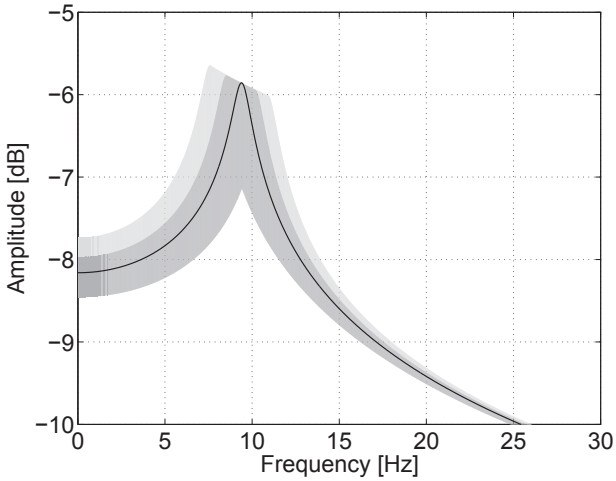


FIGURE 4.2 – *Amplitude de la réponse en fréquence du système à SDOF ; en trait fin la réponse du système nominal ; zone grise pour la dispersion d'amplitude totale ; gris clair est pour une probabilité plus grande que* 75%.

et $u_{\inf}(f)$ est l'enveloppe inférieure de la réponse du système, donnée par :

$$u_{\inf}(f) = \begin{cases} \frac{1}{\sqrt{\left(k_{\inf} - m(2\pi f)^2\right)^2 + 4\pi^2 c^2 f^2}} & \text{pour } f^2 \in [0, \mu_k] \\ \frac{1}{\sqrt{\left(k_{\sup} - m(2\pi f)^2\right)^2 + 4\pi^2 c^2 f^2}} & \text{pour } f^2 \in [\mu_k, +\infty[\end{cases}$$

Pour illustrer ces résultats, nous avons choisi un système SDOF ayant une moyenne de $3500N$ et un écart type de $700N$. Dans la figure 4.2, l'amplitude nominale en fonction de la fréquence est tracée ainsi que la dispersion totale, et un niveau de probabilité de 75%.

4.6 AMÉLIORATION DE LA MÉTHODE NNC

4.6.1 Introduction

La méthode de la contrainte normale normalisé (NNC) a été mise au point par Messac et Ismail-Yahaya pour améliorer sa version originale qui produit aussi des points non Pareto (Messac et al. 2003). Elle est relativement facile à mettre en oeuvre, et peut générer une distribution uniforme de points de Pareto le long de la frontière de Pareto. Elle est insensible aux dimensions des fonctions objectif grace à la normalisation de celles-ci, et elle est valable pour un nombre arbitraire de fonctions objectif. La méthode NNC, comme la méthode NBI, fonctionne en transformant le problème d'optimisation multi-objectif en un ensemble de sous-problèmes non linéaires. Chaque sous-problème fait l'objet d'une réduction supplémentaire de l'espace de recherche.

4.6.2 Formulation et amélioration de la méthode NNC

Soit $F^* = (f_1^*, f_2^*, ..., f_\ell^*)^T$ le point utopique du problème d'optimisation multi-objectif pour ℓ fonctions objectif et n variables. Soit x_i^* la variable correspondant à l'objectif de la fonction f_i pour $i = 1, ..., \ell$. Le point $F(x_i^*)$ pour tout $i = 1, 2, ..., \ell$ sont appelés les points d'ancrage. La méthode NNC utilise les valeurs normalisées du point de la fonction objectif, notée barre de F pour faire face au problème d'échelle des fonctions. Ceci peut être obtenu en utilisant les points utopie (f_i^u) et les points pseudo-nadir (f_i^{NP}) comme suit :

$$\bar{f}_i = \frac{f_i - f_i^u}{f_i^{Np} - f_i^u}, \quad pour i \in \{1, ..., n\}$$

où

$$f_i^u = f_i(x_i^*) \quad et \;\; f_i^{Np} = max\{f_i(x_1^*), ..., f_i(x_n^*)\}$$

Notons par V_k (pour $k = 1, 2, ..., \ell - 1$) les vecteurs de l'utopie hyper-plan générés par chacun des points d'ancrage, en faisant varier systématiquement β $(= (\beta_1, ..., \beta_\ell)^T | \sum_{i=1}^{\ell} \beta_i = 1)$. On note par Z_β le point sur hyper plan idéal corespondent au vecteur β.
Le sous-problème NNC, notée NNC_β, pour une valeur donnée de β, est formulé comme suit :

$$\begin{cases} \min\limits_{x} \quad \bar{f}_\ell(x) \\ \text{sujet à} \quad g_i(\mathbf{x}) \leq 0 \;,\; i = 1, ..., m, \\ \qquad h_j(\mathbf{x}) = 0 \;,\; j = 1, ..., p, \\ \quad V_k^T(\bar{F} - Z_\beta) \leq 0 \;\; for \;\; 1 \leq k \leq \ell - 1 \end{cases} \tag{4.7}$$

La procédure de résolution par cette méthode est schématisée dans la figure 4.3 pour le cas bi-dimensionnel. Dans le cas de génération de solutions non Pareto, une procédure de filtrage est utilisée dans le but de les éliminer. Parmi les inconvénients de la méthode

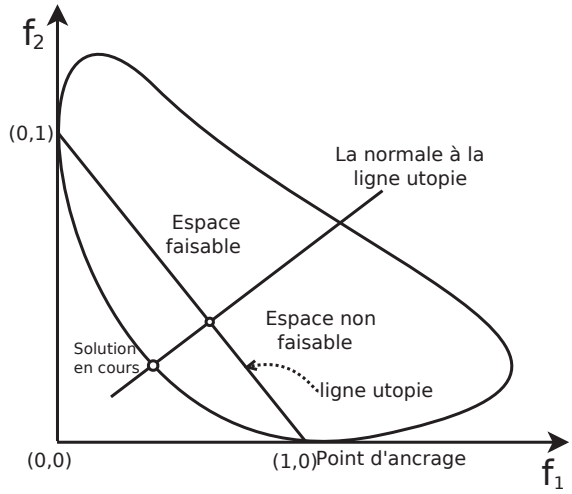


FIGURE 4.3 – *Méthode de la normalisation de la contrainte normal (NNC) pour le cas bi-dimensionnel.*

NNC est qu'elle peut être inefficace à cause de la génération de solutions redondantes (Shukla et Deb 2007). Cette situation se produit dans le cas des problèmes non convexes. Dans l'exemple donné dans Utyuzhnikov et al. (2009), soixante six points sur le plan utopie donnent seulement vingt quatre solutions Pareto.
Pour pallier à cet inconvénient, nous proposons une amélioration de la méthode de NNC (dite INNC) en effectuant, pour chaque itération, une projection de la solution trouvée sur l'hyperplan et en la comparant à la valeur de Z_β. Celle-ci est mise à jour si elle est différente de la projection. Pour illustrer ce principe, prenons un exemple de problème bi-objectif, non convexe, correspondant au 2me problème qu'on traitera dans la suite de ce chapitre : Le graphe 4.4 représente l'évolution de la recherche par la méthode NNC standard : la recherche se fait pour tous les points (41pts), y compris les points 16 à 18,

avec un pas constant. Pour INNC, la recherche se fait d'une manière intelligente : le pas à utiliser dépend de la projection du point en cours sur la ligne utopie. Ainsi, les point 15 à 28 ont été sautés (voir figure 4.5.

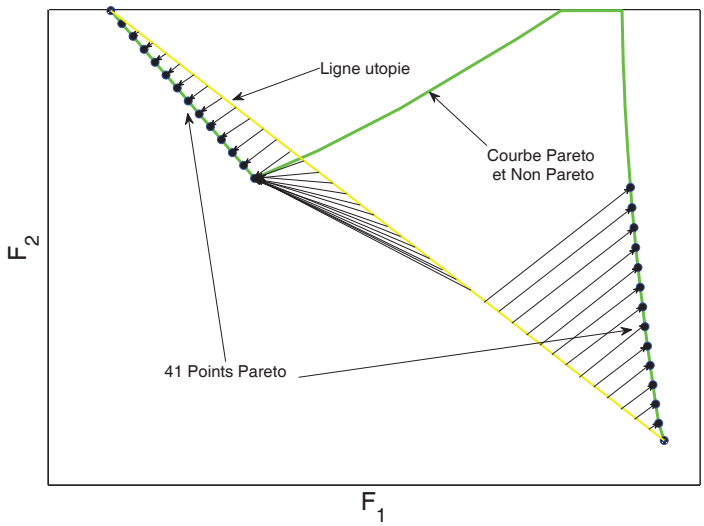


FIGURE 4.4 – *Evolution de la méthode NNC, point par point, pour le cas de problème non convex à deux dimensions.*

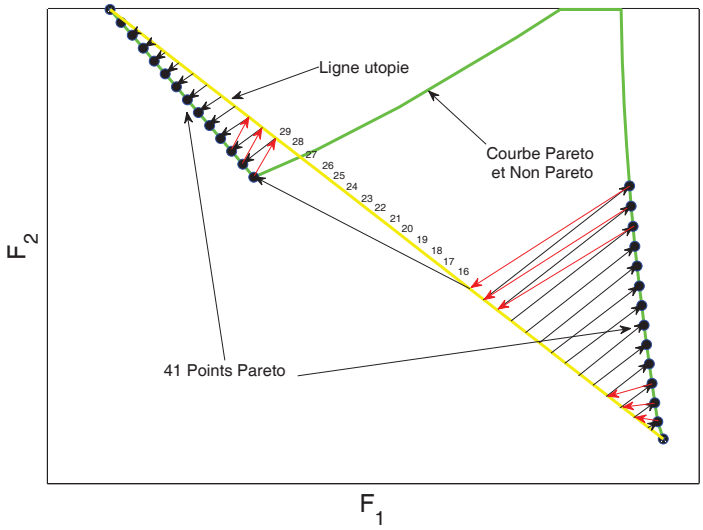


FIGURE 4.5 – *Evolution de la méthode INNC(NNC améliorée), point par point,pour le cas de problème à deux dimensions.*

Dans la section 4.8, Les performances de INNC seront comparés à celles de la méthode NNC et à d'autres méthodes d'optimisation multi-objectif bien connues, à savoir le NBI, SPEA2 et NSGAII, et ce pour le deuxième problème traité. L'algorithm RFNM est utilisé pour résoudre les sous problèmes générés par les méthodes INNC, NNC, et NBI. La méthode de la pénalité est appelée pour faire face aux contraintes générés par ces méthodes.

Points	A	B	C	D	E	F
Fréquence [Hz]	0	3	9	13	16	29
Amplitude [dB]	-8	-5.5	-5.5	-6.45	-6.75	-10

TABLE 4.1 – *Coordonnées des points qui définissent la coque de réponse d'amplitude de crête pour le premier problème*

Points	G	H	I	J
$\bar{k}$ [N/m]	2981	4296	5957	7142
σ_k [N/m]	0	1518	765	3325
Points	G'	H'	I'	J'
$\bar{k}$ [N/m]	2981	4292	5983	7194
σ_k [N/m]	0	757	378	1644

TABLE 4.2 – *Coordonnées des points characteristiques dans le front de Pareto (à gauche pour $P_{f1} = 25\,\%$ et à droite pour $P_{f2} = 0$) pour le premier problème.*

4.7 APPLICATIONS

Dans cette section, deux applications sont présentés en considérant deux coques de réponse d'amplitude de crête(amplitude peak response hull) distincts, ce qui conduit à des solutions différentes au niveau du front de Pareto.

4.7.1 Premier Problème d'optimisation

Le premier problème d'optimisation est défini à partir de la coque de réponse d'amplitude de crête qui correspond aux segments de lignes délimités par les points A à F donnée dans le tableau 4.1. La figure 4.6 montre ces coques de réponse d'amplitude de crête disponibles, ainsi qu'une amplitude FRF correspondant à un système SDOF déterministe qui appartient à la zone autorisée, car il est entièrement sous la courbe délimitée par la réponse de crête maximum.

Pour cette première application, deux situations distinctes sont étudiées qui correspondent à deux probabilités de défaillance : $P_{f1} = 25\,\%$ et $P_{f2} = 0$. Cela conduit à deux fronts de Pareto présentés dans la figure 4.7. Bien que leurs valeurs sont différentes, elles ont des formes similaires, ce qui peut être décrit par quatre points angulaires, à savoir G, H, I et J pour P_{f1} et G', H', J' et I' pour P_{f2} dans la figure 4.7 et dans le tableau 4.2. L'interprétation au sujet d'une solution optimale et l'explication du comportement mécanique associé est commune pour les deux fronts de Pareto. Par conséquent, il n'est pas nécessaire de distinguer ces deux situations dans les explications suivantes, et nous discuterons seulement le front de Pareto correspondant à P_{f1}. Pour interpréter les résultats, nous allons parcourir le front de Pareto du point G, situé à gauche, jusqu'au point I situé à droite. Le point G correspond à la moyenne minimale de raideur permise par la conception c-à-d qu'il n'existe pas de solution de conception qui peut avoir une raideur moyenne plus faible sans dépasser la coque réponse maximale(peak response hull) tout en respectant les contraintes sur les paramètres (variables de conception positives). A ce stade de conception, l'écart type de la raideur est également minime, en prenant la valeur nulle. Telle est la situation limite où un système aléatoire devient déterministe. Donc, il n'y a pas d'incertitude acceptable à ce point de conception. Par conséquent, nous considérons dans cette situation extrême que la contrainte de probabilité de défaillance est respectée pour toute valeur cible. L'amplitude de réponse du système mécanique (mechanical system response amplitude) permet de mieux comprendre la position du point de conception G. En fait, l'amplitude de réponse en fréquence représenté sur la figure 4.6

est précisément celle du système mécanique correspondant à ce point de conception G. Nous pouvons observer sur cette figure 4.6 que l'amplitude de FRF croise les coques de réponse d'amplitude de crête à la fréquence nulle [1]. Il ressort clairement de cette figure qui donne un écart type non nul conduit à une dispersion autour de la réponse en amplitude nominale, ce qui est impossible car il n'y a pas de marges entre l'amplitude de la réponse nominale et la coque pic d'amplitude de réponse à cette fréquence(nulle).
En outre, on peut voir que la diminution de cette valeur de raideur moyenne optimale augmenterait l'amplitude FRF sur toutes les fréquences, c'est également impossible pour la même raison. C'est ce qui explique mécaniquement quelle est cette solution et comment ou pourquoi elle existe. Au contraire, l'augmentation de la raideur moyenne en résulte une diminution de la réponse en amplitude du système nominal à toutes les fréquences, y compris la fréquence nulle. Cela permet au système nominal d'aller loin de la zone interdite délimitée par les coques de réponse d'amplitude de crête, ce qui permet désormais un caractère aléatoire dans la réponse. Ainsi, l'augmentation de la raideur moyenne permet la possibilité d'augmenter l'écart type du raideur, à condition que la probabilité de défaillance ne soit pas dépassée. Cependant, il faut garder à l'esprit que l'augmentation de la raideur moyenne augmente également la fréquence de résonance du système. De même, les solutions de conception optimale se déplacent le long du front de Pareto du point G au point H (voir figure 4.7). Toutefois, le point de conception H est une situation limite où il devient impossible de poursuivre l'augmentation de l'écart type en augmentant la raideur moyenne. A ce stade de conception (H), la dispersion autorisée du réponse du système (dans le sens où il respecte la contrainte sur la probabilité de défaillance) est à la fois limitée à deux fréquences distinctes, à savoir 0 Hz et 12 Hz, comme il est indiqué sur la figure 4.8. Ainsi, l'augmentation de la raideur moyenne nécessite dès maintenant la réduction de l'écart type de la raideur afin de respecter la contrainte sur la probabilité de défaillance. En suivant ces nouvelles règles, on peut aller du point de conception H au point de conception I situé sur le front de Pareto (à noter que ces solutions de conception n'appartiennent pas strictement au front de Pareto, puisque les deux fonctions objectif sont détériorées lorsqu'on parcourt ce chemin).

Puis, revenant sur le point de conception I du front de Pareto, la raideur moyenne obtenue a conduit à une baisse suffisante de la réponse globale en amplitude, permettant à la réponse en amplitude maximale de passer sous le segment D-E de réponse de la coque pic d'amplitude. Ce qui permet maintenant une possibilité d'augmentation de l'écart type lorsque l'on augmente la raideur moyenne, allant du point de conception I au point J du front de Pareto. Ce dernier point de conception du front de Pareto correspond à celui qui peut avoir le plus grand écart-type, qui est le meilleur point de conception lorsque l'on considère seulement cet objectif.

4.7.2 Le deuxième problème d'optimisation

Le deuxième problème d'optimisation défini à partir de la coque de réponse de crête d'amplitude qui correspond au segment délimité par les points A" à F" donnés dans le tableau 4.3. Seule la probabilité de défaillance nulle est considérée, conduisant au front de Pareto de la figure 4.9. La comparaison des tableaux 4.1 et 4.3 montre que seul le point D" diffère du point D entre le premier et le deuxième problème d'optimisation. En conséquence, la première partie du front de Pareto pour ce second problème est commune avec celui du premier problème, comme on peut le voir dans la figure 4.9. Cependant, il y a une différence significative lorsqu'on s'approche des points $I_1^{"}$ et $I_2^{"}$, car il n'y a pas de solution pour le problème d'optimisation $I_1^{"}$ et $I_2^{"}$. Cette situation conduit à un front de Pareto discontinu et se produit car il n'y a aucune possibilité pour l'amplitude FRF d'avoir une résonance de pic sous le point D" de l'amplitude coque réponse maximale(amplitude

1. La fréquence nulle correspond à un chargement statique

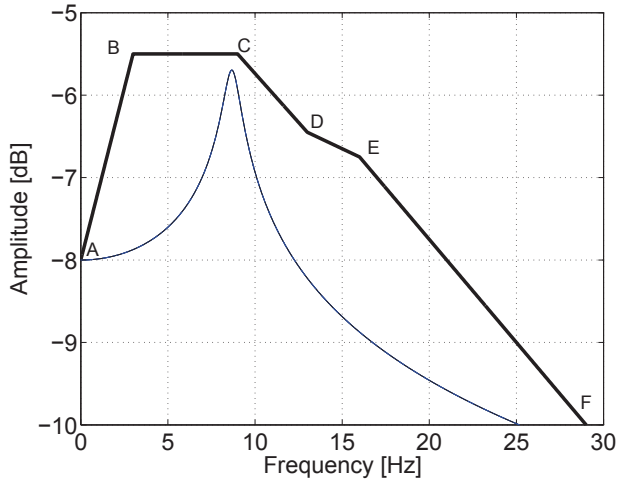


FIGURE 4.6 – *Amplitude de réponse en fréquence pour un système SDOF déterministe et l'amplitude coque réponse maximale pour le premier problème*

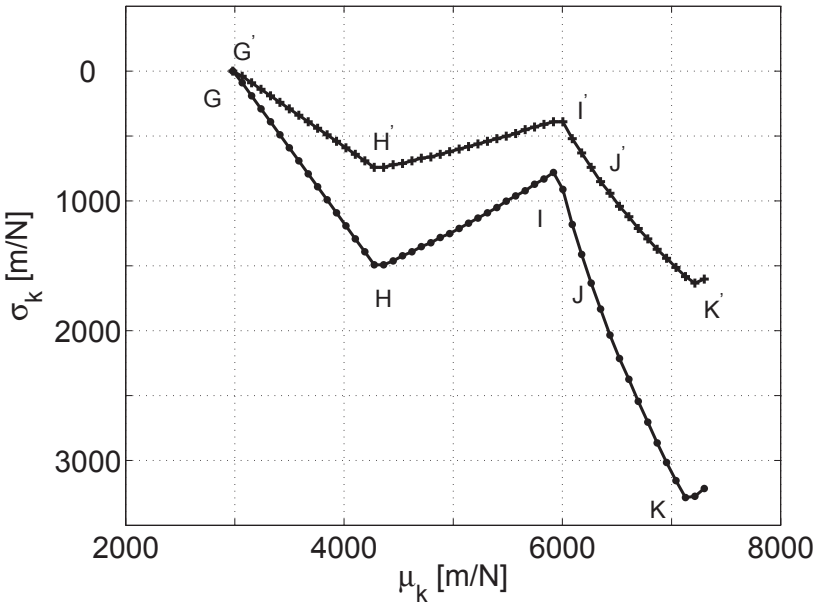


FIGURE 4.7 – *front de Pareto pour le premier problème ; • : est pour* $P_{f1} = 25\,\%$*,* **+** *: pour* $P_{f2} = 0\%$*.*

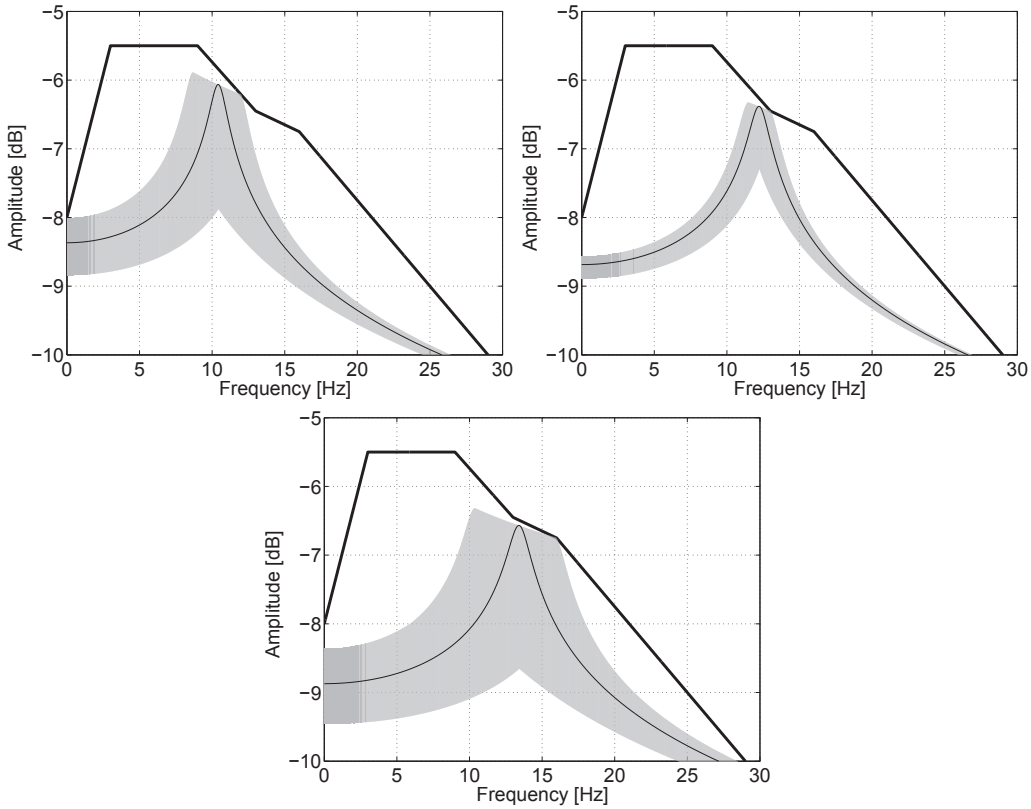


FIGURE 4.8 – *Amplitude de réponse du système SDOF aux points de conception H (haut, gauche), I (haut, droite), J (bas). La zone grise montre la dispersion qui respect la contrainte $P_f \leq P_{f1}$; le trait fin indique la réponse du système nominal.*

Points	A"	B"	C"	D"	E"	F"
Frequency [Hz]	0	3	9	13	16	29
Amplitude [dB]	-8	-5.5	-5.5	-6.52	-6.75	-10

TABLE 4.3 – *Coordonnées des points qui définissent* ***the amplitude peak response hull*** *pour le deuxième problème d'optimisation*

peak response hull). En fait, même pour un écart type nul, le niveau de réponse du système SDOF est trop élevé dans cette gamme de fréquence, lorsque la raideur moyenne prend des valeurs entre 6434 et 6902 N/m.

4.8 COMPARAISON AVEC D'AUTRES MÉTHODES D'OPTIMISATION MULTIOBJECTIF

Dans cette section, la méthode INNC (NNC amélioré) sera comparée à des méthodes très utilisées en optimisation multiobjectif dans le domaine de l'ingénierie. Pour cette comparaison sont utilisées, les algorithmes suivants : Normalized Normal Constraint (NNC), Normal Boundary Intersection (NBI), Strength Pareto Evolutionary Algorithm

Points	G"	H"	$I_1^{"}$	$I_2^{"}$	J"
$\bar{k}$ [N/m]	2981	4084	6434	6902	7208
σ_k [N/m]	0	637.2	0	0	1639

TABLE 4.4 – *Coordonnées des points characteristiques du front de Pareto pour le second problème d'optimisation.*

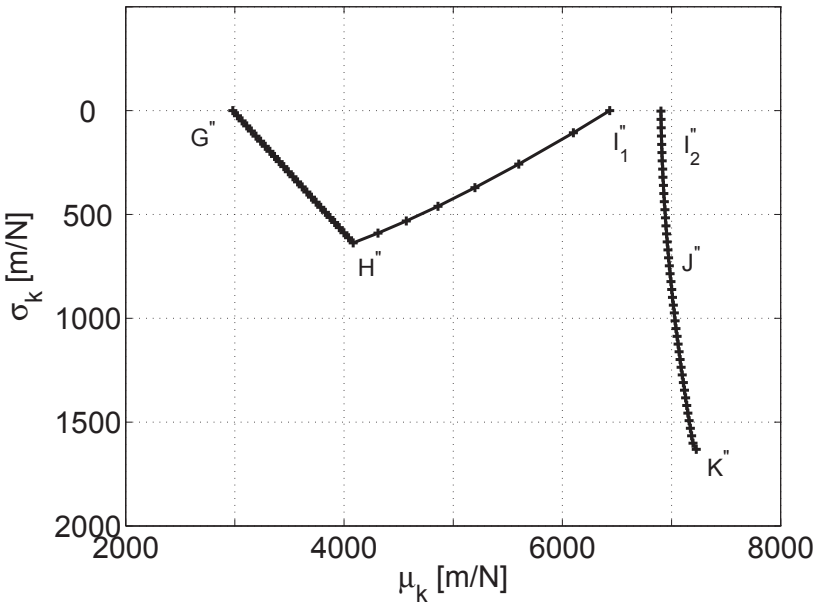


FIGURE 4.9 – *Front de Pareto pour le second problème.*

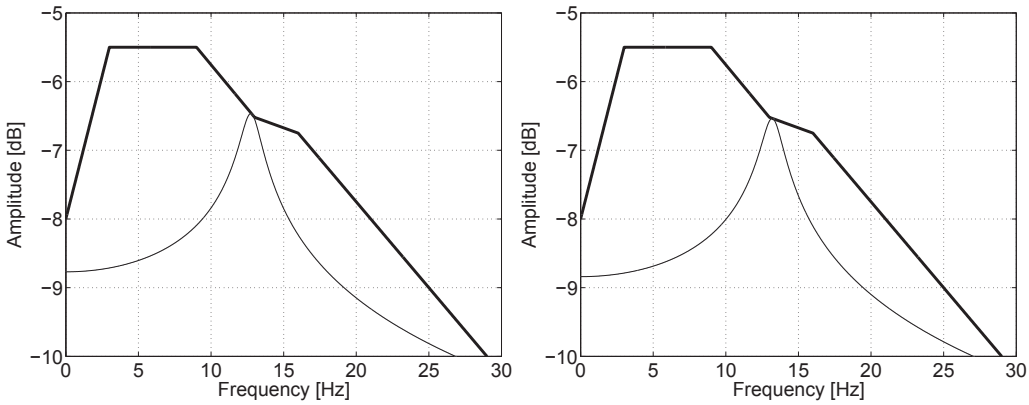


FIGURE 4.10 – *Graphes du* ***system response amplitude*** *de SDOF aux points de conception* $I_1^{"}$ *(à gauche) et* $I_2^{"}$ *(à droite).*

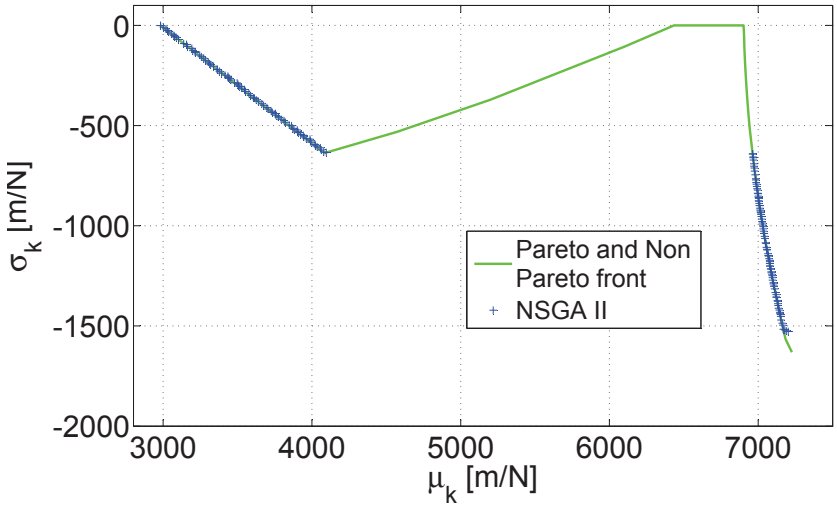


FIGURE 4.11 – *Front de Pareto généré par la méthode NSGA II*

	Methods				
	INNC	NNC	NBI	SPEA2	NSGA II
CPUT(s)	30.30	48.23	64.89	51.59	40.22
NevF	39 984	59 342	124 443	50 000	57 501

TABLE 4.5 – *Comparison entres le méthodes INNC, NNC, NBI, SPEA2 et NSGA II*

2 (SPEA2), et Nondominated Sorting Genetic Algorithm (NSGA II). La comparaison est faite en utilisant ces méthodes pour générer le front de Pareto pour le deuxième problème. Pour INNC, NNC et NBI le nombre de points du front de Pareto est fixé à 41 et notre méthode hybride RFNM est utilisée pour résoudre les sous-problèmes générés par ces méthodes. En ce qui concerne NM, nous avons adopté les paramètres standard recommandées par les auteurs, les critères d'arrêt utilisées sont : nombre maximum d'évaluations de la fonction, $maxfun = 100$, nombre maximum d'itérations, $maxiter = 100$; tolérance sur l'amélioration de la la fonction $Tolf = 10^{-3}$, et la tolérance sur la variation relative de x est $tolx = 10^{-3}$. En ce qui concerne la taille de l'échantillon pour la formule de représentation, il est fixé à 50. Les paramètres utilisés dans SPEA2 sont les suivants : taille de la population, 100 ; le nombre maximum de générations est fixé à 50, le nombre d'individus pour le tournoi(Number of individuals for tournament) est de 2 ; la graine pour le générateur aléatoire(seed for random generator) est 1 ; la variable de probabilité de mutation et la variable de la probabilité d'échange (swap probability) sont respectivement 1 et 0.5 L'algorithme NSGA II est utilisé avec les paramètres suivants : la probabilité de croisement = 0.8 ; fraction Pareto = 0.35 ; Indice de distribution de mutation (Mutation Distribution Index) = 100, indice de distribution de croisement (Crossover Distribution Indexs) = 20. Les algorithmes proposés sont programmées sous Matlab et fonctionnent sous Windows XP avec processeur Intel Core 2 Duo à 2.0 GHz et 2 Go de RAM.
Les résultats présentés sont en termes de temps CPU (CPUT en secondes) et le nombre d'évaluations de la fonction (NevF).
Les résultats de la comparaison sont exposées dans les figures 4.11, 4.12, 4.13, 4.14, et dans le tableau 4.5.

On observe que la méthode SPEA2 parvient à générer le front de Pareto, avec une

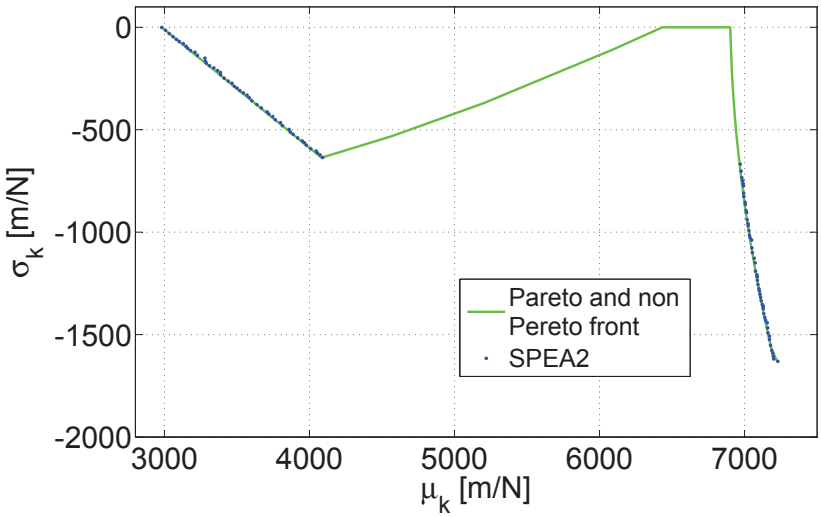


FIGURE 4.12 – *Front de Pareto généré par la méthode SPEA2*

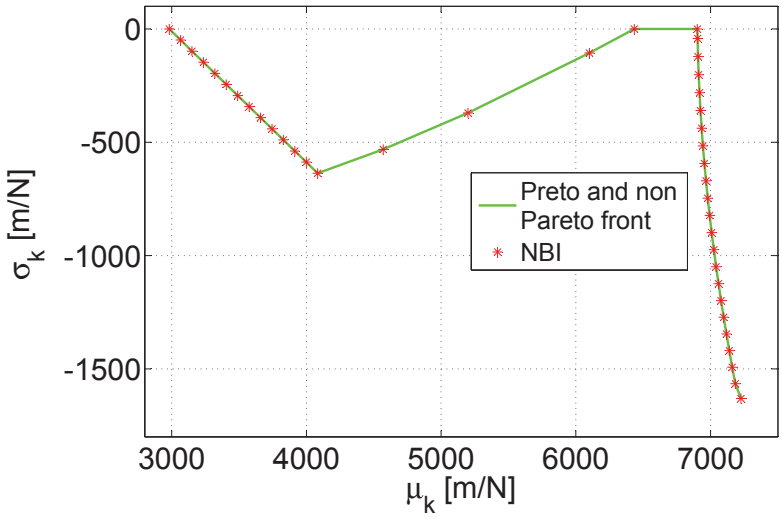


FIGURE 4.13 – *Front de Pareto généré par la méthode NBI*

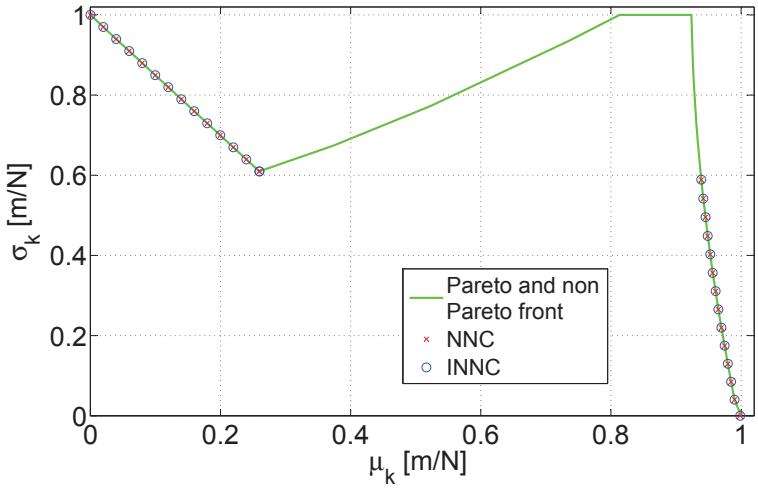


FIGURE 4.14 – *Front de Pareto généré par les méthodes NNC et INNC*

faible précision (voir la figure 4.11) et un temps CPU plus élevé que celui relatif à la méthode NSGAII qui ne parvient pas à générer les point de Pareto pour μ_k située au alentour de 7250 $[m/N]$ (voir figure4.12). Pour ces deux méthodes les points de Pareto ne sont pas uniformément réparties. Les résultats montrent aussi que la méthode NBI est précise mais il est coûteuse en terme de temps CPU en comparaison avec les autres méthodes, cela est dû à des contraintes d'égalité qui consomme beaucoup de temps CPU. On note aussi qu'une amélioration significative est obtenue par l'utilisation de INNC, en effet le front de Pareto généré est aussi précis que la méthode NNC avec une convergence plus rapide (temps CPU plus bas avec un nombre d'évaluations de la fonction plus faible).

4.9 Conclusion

Un dispositif de suspension est considéré dans ce chapitre. Il est modélisé par un système dynamique linéaire SDOF impliquant une rigidité aléatoire uniforme. La conception de la raideur est réalisée en respectant une coque de réponse amplitude maximale, compte tenu de la probabilité de défaillance. Les deux fonctions objectifs découlent de l'intérêt pour la réalisation d'un dispositif de suspension ayant la plus faible rigidité et le plus grand écart type. Pour générer le front de Pareto, pour ce problème convexe on a amélioré la méthode NNC (INNC) pour obtenir la courbe de Pareto en utilisant notre algorithme hybride RFNM. Les résultats montrent qu'une courbe discontinue de Pareto peut être obtenu pour des valeurs spécifiques de la coque réponse en amplitude crête. Par conséquent, il est montré à travers cet exemple relativement simple (SDOF) que des situations difficiles peuvent se produire pour la conception des systèmes mécaniques dynamiques lors de l'examen d'amplitude coque réponses, même pour cette situation simple ; mais la stratégie proposée ici donne des résultats valables, capables d'aider les concepteurs à choisir le meilleur compromis entre la moyenne et l'écart type pour ce genre de problèmes. Une comparaison de NNC amélioré (INNC), en utilisant RFNM, avec des algorithmes d'optimisation multiobjectif connus a démontré son efficacité et son efficience.

CHAPITRE 5

APPROXIMATION NUMÉRIQUE DE LA SOLUTION EN OPTIMISATION EN DIMENSION INFINIE EN UTILISANT LA FORMULE DE REPRESENTATION

SOMMAIRE

6.1 INTRODUCTION 105
6.2 REPRESENTATION DE SOLUTION POUR L'OPTIMISATION MULTI-OBJECTIF(RSMO) 106
6.2.1 Principe de base 106
6.2.2 Formulation 106
6.2.3 Algorithme 109
6.3 VALIDATION EXPÉRIMENTALE 109
6.3.1 Exemple illustratif : problème Zhang 110
6.3.2 Comparaison avec d'autres méthodes multi-objectif 112
6.3.2.1 Problème Deb (1999) 113
6.3.2.2 Problème de Zitzler ZDT1 116
6.4 CONCLUSION 118

DANS ce chapitre, nous proposons une extension de la formule de représentation en dimension infinie. Pour cela, on considère un problème d'optimisation non convexe, impliquant une fonctionnelle régulier J, sur un sous-ensemble fermé et borné S d'un espace de Hilbert séparable V. Aucune hypothèse de convexité n'est introduite. Les solutions sont représentées à l'aide d'une formule close impliquant des moyens de variables aléatoires convenables, analogue à Pincus (1968). La représentation nécessite l'utilisation d'une méthode numérique basée sur la génération d'échantillons en vue d'estimer ces moyens. Trois stratégies pour la mise en oeuvre sont examinés, avec l'originalité qu'ils n'impliquent pas une approximation en dimension finie à priori de la solution et considère une base hilbertienne ou une famille dense dénombrable de V. Les

résultats peuvent être améliorés sur un sous-espace de dimension finie par une procédure d'optimisation, afin d'obtenir des solutions de meilleure qualité. Des exemples numériques impliquant des situations classiques issues du calcul des variations sont présentées et ont montré l'efficacité de la démarche de calcul.
Ce chapitre est organisé comme suit : la section 5.2 présente les notations utilisées, tandis que la section 5.3 est consacrée aux mesures finies dans un espace de Hilbert à dimension infinie, suivie de l'intégration avec une mesure sur un espace de Hilbert dans la section5.4. Les hypothèses présentées dans la section 5.5 servent de base pour la représentation de solution en dimension infinie traitée dans la section 5.6. Une approche numérique dans la section 5.7 suivie de présentation des résultats numériques dans la section 5.8 suivie d'une conclusion et perspectives dans la section 5.9.

La principale contribution en rapport avec ce chapitre est :

H. Zidani, J.E. Souza de Cursi, and R. Ellaia. **Numerical Approximation Of The Solution In Ifinite Dimensional Global Optimization Using A Representation Formula**, *soumis au 'Journal of Global optimization'*, (ISSN : 0925 – 5001 (print version), 1573 – 2916 (online)).

5.1 Introduction

La théorie des processus stochastiques fournit des outils pour la représentation des solutions de certains problèmes mathématiques en tant que moyen de variables aléatoires convenablement choisies. Une telle représentation est généralement désignée comme étant une représentation de Feynman-Kac de la solution. Dans le cadre des problèmes aux limites, les représentations de Feynman-Kac ont été introduits et des algorithmes ont été développés pour des situations linéaires et non linéaires (Souza de Cursi 1994, Milshtein 1995, Morillon 1997, Bossy et Talay 1997, Souza de Cursi 2002). Dans l'optimisation à dimension finie, un résultat pionnier (Pincus 1968) a proposé une représentation de la solution global d'optimisation d'une fonction continue. Ce résultat a été récemment étendue à une représentation numérique plus intéressante avec les algorithmes numériques (Bez et al. 2005, Koscianski et de Cursi 2005, Souza de Cursi 2003; 2005). Dans ce travail, on considère la généralisation pour un espace de Hilbert quelconque, éventuellement pour des problèmes à dimensions infinies, ce qui s'applique à des problèmes issus du calcul des variations ou de contrôle, utilisant des fonctions comme inconnues. Notre problème modèle est la détermination d'un élément u de l'espace fonctionnel V qui soit le minimum global d'une fonctionnelle $J : V \longrightarrow \mathbb{R}$ sur un ensemble borné $S \subset V$:

$$u = \arg\min_{S} J, \;\; i.e., u \in S \, , \; J(u) \; \leq \; J(v) \; , \; \forall \, v \; \in \; S \, . \tag{5.1}$$

J, et S ne sont pas supposés convexes, mais on suppose que u existe et est déterminée de façon unique, *c-à-d*, que la région de S contient exactement un minimum global (mais il peut contenir de nombreux minima locaux). Ensuite, une représentation de la valeur de $\ell(u)$ peut être obtenue, pour toute application linéaire continue ℓ ayant ses valeurs dans un espace de dimension finie : ainsi, elle caractérise l'action sur u de tout élément ℓ du dual topologique de V. Ainsi, la représentation caractérise entièrement u. En outre, elle peut être utilisée afin de représenter explicitement les projections de u sur n'importe quel sous-espace de dimension finie de V : le théorème conduit également à une représentation de la solution u sur une base arbitraire de Schauder de V.

5.2 NOTATION

Soit $\mathbb{N}$, l'ensemble des entiers naturels et $\mathbb{N}^*$ l'ensemble des entiers strictement positifs. $\mathbb{R}$ représente l'ensemble des nombres réels $]-\infty,+\infty[$ et $\mathbb{R}_e$ est l'ensemble des nombres réels étendus $\mathbb{R}\cup\{-\infty,+\infty\}$. $(a,b)=\{x\in\mathbb{R}:a<x<b\}$ un interval ouvert de nombre réels. Pour un entier $k\in\mathbb{N}^*$, $\mathbb{R}^k=\{x=(x_1,...,x_k):x_n\in\mathbb{R},1\leq n\leq k\}$ désigne l'ensemble des k-uplets de nombres réels. une analogue notation est utilisée pour $\mathbb{R}_e^k$. On note $|\bullet|_p$ la p-norme standard sur $\mathbb{R}^k$. L'ensemble de tous les rectangles ouverts de $\mathbb{R}^\infty$ est notée $\mathfrak{R}_k$. On utilise la notation $\mathbb{R}^\infty=\{x=(x_1,x_2,x_3,...):x_n\in\mathbb{R},\forall\, n\in\mathbb{N}^*\}$ pour l'espace de toutes les suites de nombres réels. Une notation analogue est utilisée pour $\mathbb{R}_e^\infty$. L'ensemble de tous les rectangles ouverts de $\mathbb{R}^\infty$ est noté $\mathfrak{R}_\infty$. On note $\mathbb{R}_0^\infty$ le sous-ensemble de $\mathbb{R}^\infty$ formée par des suites contenant un nombre fini non nul d'éléments. $\mathbb{R}_0^\infty$ peut être identifié à $\Pi=\bigcup_{k\in\mathbb{N}^*}\left((\mathbb{N}^*)^K\times\mathbb{R}^k\right)$: l'application $\pi:x=(x_1,x_2,x_3,...)\longrightarrow(n_1,...,n_k,x_{n_1},...,x_{n_k})\in\Pi$ - où les éléments non nulle de x sont $x_{n_1},...,x_{n_k}$ - est une bijection entre $\mathbb{R}_0^\infty$ et Π.
Dans la suite, nous utilisons l'ensemble $\mathfrak{P}=\bigcup_{k\in\mathbb{N}^*}\left((\mathbb{N}^*)^k\times\mathfrak{R}_k\right)$. On considère un *espace de Hilbert séparable réel* $(V,(\bullet,\bullet))$. $\|\bullet\|$ est la norme sur V. Puisque notre objectif est l'application de la théorie aux problèmes de calcul des variations, notre modèle pour l'espace V est un espace fonctionnel tel que, par exemple, celui de Sobolev : les éléments de V sont supposés prendre leurs valeurs dans $\mathbb{R}^d$ pour un certain $d\in\mathbb{N}^*$. Le théorème principal s'étend à des espaces de Banach tels que $[L^p(\Omega)]^d$. *Pour alléger les notations,* $(\bullet,\bullet)$ *et* $\|\bullet\|$ *sont utilisés dans la suite pour les produits scalaires et normes relatives à d'autres espaces. Aucune confusion n'est induite par cette simplification. En cas de nécessité, l'espace sera spécifié en écrivant* $\|\bullet\|_V$.

Il existe une *famille orthonormale dénombrable* $\Phi=\{\varphi_n\}_{n\in\mathbb{N}^*}\subset V$ tel que toute $v\in V$ peut être représentée par une série convergente unique $v=\sum_{n=1}^{+\infty}v_n\varphi_n$. Ainsi, chaque $v\in V$ est entièrement caractérisé par la suite de nombres réels $v=(v_1,v_2,v_3,...)\in\mathbb{R}^\infty$, qui est déterminée d'une façon unique et V peut être identifié à un sous-ensemble $\mathscr{V}$ de $\mathbb{R}^\infty$, défini par $\mathscr{V}=\{a=(a_1,a_2,a_3,...):\|a\|<\infty\}$; $\|a\|=\left(\sum_{n=1}^{+\infty}a_n^2\right)^{1/2}$.

Soit $(x,y)=\sum_{n=1}^{+\infty}x_ny_n$ et $V\ni v=\sum_{n=0}^{+\infty}V_n\varphi_n\longrightarrow^I v=(v_1,v_2,v_3,...)\in\mathscr{V}$.
Comme $(I(V),I(w))=(v,w)$, I est une bijection isométrique entre V et $\mathscr{V}$ et $(\mathscr{V},(\bullet,\bullet))$ est un espace de Hilbert. Cette identification conduit à la construction de la base topologique et des mesures sur V en utilisant celles définies sur $\mathscr{V}$ (voir la section 5.3). Plus précisément, on doit utiliser la base topologique $\mathfrak{B}_\infty=\mathfrak{R}_\infty\cap\mathscr{V}$ formée par les rectangles ouverts de $\mathscr{V}$. La version de dimension finie $\mathfrak{B}_k$ de $\mathfrak{B}_\infty$ associée à $\mathbb{R}^k$ est simplement $\mathfrak{B}_k=\mathfrak{R}_k$. Pour des raisons pratiques, et des besoins d'approximations numériques, on considère également une *famille dénombrable* $\Psi=\{\psi_n\}_{n\in\mathbb{N}}\subset V$ telle que $\mathscr{D}=\overline{[\psi]}$, c-à-d le sous-espace vectoriel engendré de ψ est *dense* dans V. $\mathscr{D}$ peut être identifié à un sous-ensemble $\mathscr{V}_0=\{a\in\mathscr{V}:a\in\mathbb{R}_0^\infty\}\subset\mathscr{V}$. Ainsi, $I\circ\pi$ est une bijection entre $\mathscr{D}$ et $\mathfrak{V}_0=\pi(\mathscr{V}_0)$. L'introduction de la famille Ψ conduit à une extension des résultats à des espaces de Banach séparables. Soit $(W,(\bullet,\bullet))$ est une autre espace de Hilbert. On note par $\mathscr{L}(V,W)$ l'ensemble de toutes les applications linéaires continues $\ell:V\longrightarrow W$.
Lorsque $W=\mathbb{R}^d$, il existe un nombre réel $M_p\in\mathbb{R}$ tel que $\forall v\in V:|\ell(v)|_p\leq M_p\|v\|$. Pour un sous-ensemble $A\subset V$, on note $\mathscr{P}(A)$ l'ensemble des parties de A, *c-à-d*, l'ensemble de tous les sous-ensembles de A.

5.3 MESURES FINIES DANS UN ESPACE DE HILBERT À DIMENSION INFINIE

La façon la plus courante pour la manipulation de mesures finies sur les espaces de Banach ou Hilbert est l'utilisation de *mesures cylindriques* (Schwartz 1969). *Ici, nous allons adopter une approche différente, qui est l'une des originalités de ce travail* : pour chaque $k \in \mathbb{N}^*$, nous allons introduire une suite sommable de nombres réels positifs :

$$p = (p_1, p_2, p_3, ...) \in \mathbb{R}^\infty \; ; \; p_n \geq 0, \forall n \in \mathbb{N}^* \; ; \sum_{n=1}^{+\infty} p_n = p \geq 0 \tag{5.2}$$

et k suites de nombres réels strictement positifs et k suites de densités sur $\mathbb{R}$ telles que, pour $k, i, n \in \mathbb{N}^*$:

$$\begin{aligned} \mathbf{q}_k &= \left(q_{k,1}, ..., q_{k,k}\right), \mathbf{q}_{k,i} = \left(q_{k,i,1}, q_{k,i,2}, q_{k,i,3}, ...\right), \; q_{k,i,n} \geq 0 \; , \\ \mathbf{f}_k &= \left(f_{k,1}, ..., f_{k,k}\right), \mathbf{f}_{k,i} = \left(f_{k,i,1}, f_{k,i,2}, f_{k,i,3}, ...\right), \; f_{k,i,n} \geq 0 \; , \end{aligned}$$

$$\sum_{n=1}^{+\infty} q_{k,i,n} \quad = \quad Q_{k,i} \leq 1; \int_{\mathbb{R}} f_{k,i,n} \leq 1.$$

Soit $m \in \mathbb{R}_e^k$ et $M \in \mathbb{R}_e^k$, $\mathbf{n} = (n_1, ..., n_k)$;
$\mathscr{R}_k(m, M) = \left\{x \in \mathbb{R}^k : \; m_n < x_n < M_n, 1 \leq n \leq k\right\} = \prod_{n=1}^{k} (m_n, M_n)$. On définie une mesure finie η sur $\mathfrak{P}$ par : $\eta(\mathbf{n} \times \mathscr{R}_k(m, M)) = p_k \prod_{i=1}^{k} q_{k,i,n_i} \left(\int_{m_i}^{M_i} f_{k,i,n_i}(x_{n_i}) \, dx_{n_i} \right)$.
Ainsi, $\mu = \eta \circ I$ estt une mesure finie sur V, tandis que $\mu = \eta \circ \pi \circ I$ est une mesure finie sur $\mathscr{D}$.

Une mesure alternative peut être définie si $f = (f_1, f_2, f_3, ...)$ est une suite bornée de fonctions intégrables positives sur $\mathbb{R}$ (i. e.,$\forall \; n \in \mathbb{N}$: $f_n \geq 0$ sur $\mathbb{R}$ et $\sup^{n \in \mathbb{N}} \int_{\mathbb{R}} f_n < +\infty$) et

$$\sum_{n=1}^{+\infty} q_{k,i,n} = Q_{k,i} \; , \; \sum_{i=1}^{k} Q_{k,i} = Q_k \leq \Lambda \in \mathbb{R},$$

où Λ est indépendant de k. Donc $\eta(\mathbf{n} \times \mathscr{R}_k(m, M)) = p_k \sum_{i=1}^{k} q_{k,i,n_i} \left(\int_{m_i}^{M_i} f_{k,i,n_i}(x_{n_i}) \, dx_{n_i} \right)$ définit une mesure finie sur Π et $\nu = \eta \circ \pi$ est une mesure finie sur $\mathbb{R}_0^\infty$. Par analogie, $\mu = \nu \circ I$ est une mesure finie sur V et $\mu = \eta \circ \pi \circ I$ est une mesure finie sur $\mathscr{D}$.

5.4 INTÉGRATION AVEC UNE MESURE SUR UN ESPACE DE HILBERT

Comme nous l'avons remarqué, notre objectif est l'application de la théorie à des problèmes de calcul des variations : ainsi, V et W sont des espaces fonctionnel réels tels que, par exemple, $[L^p(\Omega)]^n$ ou l'espace de Sobolev standard, et ses élements prennent leurs vleurs sur $\mathbb{R}^k$ pour des $k > 0$.

La théorie présentée ci-dessous concerne la valeur moyenne $E(U)$ sur $S \subset V$ d'une variable aléatoire U définie sur V, c-à-d, la valeur moyenne sur S d'une fonction $U : V \longrightarrow W$, associant à chaque $v \in V$ un élement $U(v) \in W$. On a

$$E(U) = \int_S U d\mu \Big/ \mu(S)$$

L'évaluation de $\int_S U(v)\,d\mu$ implique une intégration non standard, puisque μ est définie sur un espace de dimension infinie. Pour une définition formelle, nous nous sommes d'abord limités à la situation où les éléments de W sont des nombres réels ou des fonctions réelles (*c-à-d*, $W = \mathbb{R}$ ou $W = L^p(\Omega)$).

Soit $\mathscr{M}$ l'ensemeble des *fonctions mesurables*, qui sont les fonctions qui transforment les ensembles mesurables en ensembles mesurables. On considére l'ensemble des *images des fonctions finies mesurbles* : $\mathscr{F} = \{\ f \in \mathscr{M} : f(V)$ est une fonction finie $\}$.
For $F \in \mathscr{F}$, On définie $R^*(f,S) = \{\alpha \in \mathbb{R} : \alpha \neq 0 \text{ et } \alpha \in f(S)\}$. L'ensemble des *fonctions élementaires* est :

$$\mathscr{E}(\mu,S) = \left\{e \in \mathscr{F} : \mu\left(e^{-1}(\{y\})\ \right) < \infty, \forall y \in R^*(e,S)\right\}$$

Si μ est une mesure *finie*, On a $\mathscr{E}(\mu,S) = \mathscr{F}$. pour $e \in \mathscr{E}(\mu,S)$ tel que $R^*(e,S) = \{\alpha_1,...,\alpha_n\}$, l'intégral est définie comme suit :

$$\int_S e d\mu = \sum_{i=1}^{n} \alpha_i \mu(A_i)\,;\, A_i = e^{-1}(\{\alpha_i\}) \text{ pour } 1 \leq i \leq n.$$

Cette définition est étendue à l'ensemble des fonctions mesurables positives $\mathscr{M}_+ = \{m \in \mathscr{M} : m \geq 0\}$ comme suit : soit $U \in \mathscr{M}_+$. On considére les fonctions positives élémentaires majorés par U : $\mathscr{E}_+(\mu,S,U) = \{e \in \mathscr{E}(\mu,S)\ : 0 \leq e \leq U\}$ et on définie

$$\int_S U d\mu = \sup\left\{\int_S e : e \in \mathscr{E}_+(\mu,S,U)\right\}.$$

Finalement, $U \in \mathscr{M}$ arbitraire, est décomposé en $U = U^+ - U^-$, $U^+(x) = \max\{U(x)\,,\,0\}$, $U^-(x) = \max\{-U(x)\,,\,0\}$ et on définit :

$$\int_S U d\mu = \int_S U^+ d\mu - \int_S U^- d\mu.$$

Le membre de droite est bien défini, puisque les deux U^+ et U^- sont des éléments de $\mathscr{M}_+$. La définition donne l'inégalité classique : $|\int_S U d\mu| \leq \int_S |U|\,d\mu$, ce qui correspond à une forme particulière de l'inégalité de Jensen : pour une fonction *convexe* $g : \mathbb{R} \longrightarrow \mathbb{R}$ telle que $g(U)$ est mesurable, on a $g(E(U)) \leq E(g(U))$. L'extension à la situation où les éléments de W sont des vecteurs réels ou fonctions réelles à valeurs vectorielles (*c-à-d*, $W = \mathbb{R}^d$ ou $W = [L^p(\Omega)]^d$) est réalisée en considérant les composantes de U : on a $U = (U_1,...,U_d)$ et on définit

$$\int_S U d\mu = \left(\int_S U_1 d\mu,...,\int_S U_d d\mu\right).$$

Soit $g : \mathbb{R}^d \longrightarrow \mathbb{R}$ une fonction continue *convexe* telle que $g(U)$ est mesurable. Puisque g est la borne supérieure ponctuelle de tous ses minorants affines (Tiel 1984), on a l'inégalité de Jensen :

$$g\left(\int_S U d\mu\right) \leq \int_S g(U)\,d\mu. \tag{5.3}$$

Il en suit que :

Proposition 5.4.1 $|E(U)|_p \leq E\left(|U|_p\right)$, *pour tout* $1 \leq p \leq \infty$ *tel que* $|U|_p$ *est mesurable.*

Démonstration. Appliquer (5.3) à $g(\xi) = |\xi|_p$. □

Comme d'habitude, l'intégrale de $U : V \longrightarrow W$ sur $S \subset V$ peut être approchée par :

$$\int_S U d\mu \approx \sum_{i=1}^{n} U(x_i)\,\mu(A_i)\ ;\ \bigcup_{i=1}^{n} A_i = S; x_i \in A_i \text{ pour } 1 \leq i \leq n \tag{5.4}$$

et correspond à une limite pour $n \longrightarrow +\infty$, avec le diamètre maximal des sous-ensembles A_i tendant vers zéro. Dans nos calculs numériques, on a *n'a pas* utilisé l'approximation (5.4), mais une *approximation par Montecarlo* : nous générons un échantillon $U_1, \ldots U_{nr}$ de nr variantes de la distribution de μ et nous avons fait une approximation

$$E(U) \approx \frac{1}{nr}\sum_{i=1}^{nr} U_i. \tag{5.5}$$

Cette approche est plus pratique pour la situation considérée : avec les distributions standards, tels que, par exemple, les gaussiennes ou de poisson, des méthodes efficaces de géneration existent dans la littérature.

5.5 Hypothèses

Soit $S \subset V$ un ensemble non vide, fermé et borné. Alors, il existe une constante $\alpha \in \mathbb{R}$ telle que $\|v\| \leq \alpha,\ \forall\, v \in S$.
Soit $J : V \longrightarrow \mathbb{R}$ une fonctionnelle continu. On *ne suppose* ni la differentiabilité of J ni sa convexité, mais on suppose qu'il existe une constante $\beta \in \mathbb{R}$ telle que $|J(v)| \leq \beta,\ \forall\, v \in S$. Cette supposition est vérifiée, par exemple, si J est borné inférieurement et coercitive. On note par B_ε^* la partie de S comprise dans la boule ouverte de centre u et de rayon ε et $S_\varepsilon^* = S - B_\varepsilon^*$ son complément dans S. On suppose qu'il existe $\varepsilon_0 > 0$ tel que $\mu(B_\varepsilon^*) > 0, \forall\, \varepsilon \in (0,\ \varepsilon_0)$.
Dans la suite, on note par χ_ε^* la fonction caractéristique de B_ε^* et par ψ_ε^* la fonction caractéristique de S_ε^*. On a $\chi_\varepsilon^*(v) + \psi_\varepsilon^*(v) = 1$, $\forall\, v \in S$. Soit $\lambda > 0$ un nombre réel assez grand (dans la suite, on prendra $\lambda \longrightarrow +\infty$) et $g : \mathbb{R}^2 \longrightarrow \mathbb{R}$ une fonction continue telle que $g \geq 0$. On suppose aussi qu'il existe un nombre réel $\varepsilon_0 > 0$ et deux fonctions $h_1,\ h_2 : \mathbb{R}^2 \longrightarrow \mathbb{R}$ telles que , $\forall\, \varepsilon \in (0,\ \varepsilon_0)$:

$$E(g(\lambda,\ J(v))) \ \geq\ h_1(\lambda,\ \varepsilon) \ >\ 0 \ ;\ E(\psi_\varepsilon^*(v) g(\lambda,\ J(v))) \ \leq\ h_2(\lambda,\ \varepsilon) \tag{5.6}$$

$$\forall\, \varepsilon \ \in\ (0,\ \varepsilon_0) \ :\ \frac{h_2(\lambda,\ \varepsilon)}{h_1(\lambda,\ \varepsilon)} \xrightarrow[\lambda \longrightarrow +\infty]{} 0\ . \tag{5.7}$$

5.6 Une representation de l'optimum global

Notre résultat principal est le théorème suivant :

Theorem 5.6.1 *On suppose que (5.6) et (5.7) soient satisfaites. Si V_d est une sous espace de dimension finie de V et $\P_d : V \longrightarrow V_d$ est une projection orthogonale sur V_d. Alors*

$$\frac{E(P_d(v)\, g(\lambda, J(v)))}{E(g(\lambda, J(v)))} \xrightarrow[\lambda \longrightarrow +\infty]{} P_d(u)\,,\ \text{fortement in } V. \tag{5.8}$$

De plus, pour tout $\ell \in \mathscr{L}(V, \mathbb{R})$,

$$\frac{E(\ell(P_d(v))\, g(\lambda, J(v)))}{E(g(\lambda, J(v)))} \xrightarrow[\lambda \longrightarrow +\infty]{d \longrightarrow +\infty} \ell(u)\ ;$$

$$\frac{E(P_d(v)\, g(\lambda, J(v)))}{E(g(\lambda, J(v)))} \xrightarrow[\lambda \longrightarrow +\infty]{d \longrightarrow +\infty} u,\ \text{faiblement dans } V$$

Corollary 5.6.2 *Supposons, de plus, que le théorème de Fubini-Tonnelli s'applique. Alors*

$$E((v, \varphi_n)\, g(\lambda, J(v))) = (E(v g(\lambda, J(v)), \varphi_n)\,, \forall n \in \mathbb{N}^*$$

et

$$\frac{E(v g(\lambda, J(v)))}{E(g(\lambda, J(v)))} \xrightarrow[\lambda \longrightarrow +\infty]{} u,\ \text{faiblement in } V. \tag{5.9}$$

Le théorème est une conséquence de la proposition suivante :

Proposition 5.6.3 *Supposons que (5.6) et (5.7) soient verifiées. Soit $\ell \in \mathscr{L}\left(V,\mathbb{R}^d\right)$, $d \in \mathbb{N}^*$. Alors*

$$\ell(u) = \lim_{\lambda \longrightarrow +\infty} \frac{E\left(\ell(v)\, g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} \quad .$$

Démonstration. de la proposition :

1- On a

$$\frac{E\left(\ell(v)\, g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} = \frac{E\left(\ell(u)\, g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} + \frac{E\left(\ell(v-u) g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} \; .$$

Donc,

$$\frac{E\left(\ell(v)\, g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} = \ell(u) + \frac{E\left(\ell(v-u) g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} \quad . \tag{5.10}$$

2- Soit $\varepsilon \in (0,\ \varepsilon_0)$. On a

$$E\left(\ell(v-u) g(\lambda, J(v))\right) = E\left(\ell(v-u)(\chi_\varepsilon^*(v) + \psi_\varepsilon^*(v)) g(\lambda,\ J(v))\right) \quad .$$

ainsi,

$$\begin{aligned} E\left(\ell(v-u) g(\lambda, J(v))\right) &= E\left(\ell(v-u)\chi_\varepsilon^*(v) g(\lambda, J(v))\right) \\ &\quad + E\left(\ell(v-u)\psi_\varepsilon^*(v) g(\lambda, J(v))\right) \quad . \end{aligned}$$

3- Puisque $\ell \in \mathscr{L}\left(V,\mathbb{R}^d\right)$, il existe une constante $M_p \in \mathbb{R}$ telle que, pour $v \in B_\varepsilon^*$,

$$|\ell(v-u)|_p \leq M_p \|v-u\| \leq M_p \varepsilon \tag{5.11}$$

Alors que, pour $v \in S_\varepsilon^*$,

$$|\ell(v-u)|_p \leq M_p \|v-u\| \leq M_p \left(\|v\| + \|u\|\right) \leq 2M_p \alpha \tag{5.12}$$

4- D'une part, de l'inégalité de Jensen (5.4.1),

$$\left|E\left(\ell(v-u)\chi_\varepsilon^*(v) g(\lambda, J(v))\right)\right|_p \leq E\left(|\ell(v-u)|_p\, \chi_\varepsilon^*(v) g(\lambda,\ J(v))\right).$$

et on a, de (5.11),

$$E\left(|\ell(v-u)|_p\, \chi_\varepsilon^*(v) g(\lambda,\ J(v))\right) \leq M_p \varepsilon E\left(\chi_\varepsilon^*(v) g(\lambda,\ J(v))\right).$$

En revanche, l'inéqualité de Jensen (5.4.1) montre aussi que,

$$\left|E\left(\ell(v-u)\psi_\varepsilon^*(v) g(\lambda, J(v))\right)\right|_p \leq E\left(|\ell(v-u)|_p\, \psi_\varepsilon^*(v) g(\lambda,\ J(v))\right).$$

et on a, de (5.12) et (5.6),

$$E\left(|\ell(v-u)|_p\, \psi_\varepsilon^*(v) g(\lambda,\ J(v))\right) \;\leq\; 2M_p \alpha h_2(\lambda,\ \varepsilon) \quad .$$

5- Donc,

$$\left|E\left(\ell(v-u) g(\lambda, J(v))\right)\right|_p \leq \varepsilon M_p E\left(\chi_\varepsilon^*(v) g(\lambda, J(v))\right) + 2M_p \alpha h_2(\lambda,\ \varepsilon) \quad .$$

et

$$\begin{aligned} \left|\frac{E\left(\ell(v-u)\chi_\varepsilon^*(v) g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)}\right|_p &\leq \frac{\varepsilon M_p E\left(\chi_\varepsilon^*(v) g(\lambda, J(v))\right) + 2M_p \alpha h_2(\lambda,\ \varepsilon)}{E\left(g(\lambda, J(v))\right)} \\ &= \frac{\varepsilon M_p E\left(\chi_\varepsilon^*(v) g(\lambda, J(v))\right)}{E\left(g(\lambda, J(v))\right)} + \frac{2M_p \alpha h_2(\lambda,\ \varepsilon)}{E\left(g(\lambda, J(v))\right)} \; . \end{aligned}$$

6- Puisque
$$\frac{E\left(\chi_\varepsilon^*(v)g(\lambda,\,J(v))\right)}{E\left(g(\lambda,\,J(v))\right)} \leq 1$$
et
$$\frac{h_2(\lambda,\,\varepsilon)}{E\left(g(\lambda,\,J(v))\right)} \leq \frac{h_2(\lambda,\,\varepsilon)}{h_1(\lambda,\,\varepsilon)}\ ,$$
on a
$$\left|\frac{E\left(\ell(v-u)g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)}\right|_p \leq \left(\varepsilon+2\alpha\frac{h_2(\lambda,\,\varepsilon)}{h_1(\lambda,\,\varepsilon)}\right)M_p$$

7- Cette inégalité combinée avec Eq. (5.7) conduit à
$$\forall\,\varepsilon\,\in\,(0,\varepsilon_0)\ :\ \lim_{\lambda\longrightarrow+\infty}\left|\frac{E\left(\ell(v-u)g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)}\right|_p \leq \varepsilon M_p\ .$$
Donc
$$\lim_{\lambda\longrightarrow+\infty}\left|\frac{E\left(\ell(v-u)g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)}\right|_p = 0$$
et on a
$$\lim_{\lambda\longrightarrow+\infty}\frac{E\left(\ell(v-u)g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)} = 0\ , \tag{5.13}$$

8- En prenant la limite pour $\lambda \longrightarrow +\infty$ dans Eq.(5.10) et utilisant Eq.(5.13), on a le résultat.

□

Démonstration. du théorème :

1- Soit $\{\phi_1,...,\phi_d\}$ une base orthonormale de V_d et $\ell : V \longrightarrow \mathbb{R}^d$ est donné par $\ell(v) = ((v,\phi_1),...,(v,\phi_n))$. ℓ est linéaire et $|\ell(v)|_2 = \|P_d(v)\| \leq \|v\|$. Donc, $\ell \in \mathscr{L}\left(V,\mathbb{R}^d\right)$ et, de la proposition 5.6.3 :
$$\lim_{\lambda\longrightarrow+\infty}\frac{E\left((v,\phi_n)\,g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)} = (u,\phi_n)\ .$$
Ainsi,
$$\frac{E\left(P_d(v)\,g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)} = \sum_{n=0}^{d}\frac{E\left((v,\phi_n)\,g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)}\phi_n \xrightarrow[\lambda\longrightarrow+\infty]{} \sum_{n=0}^{d}(u,\phi_n)\,\phi_n$$
fortement dans V et nous avons le premier résultat.

2- Soit $\ell \in \mathscr{L}(V,\mathbb{R})$, $\Phi = \{\varphi_n\}_{n\in\mathbb{N}^*} \subset V$ une base orthonormale de Schauder, $V_d = [\{\varphi_1,...,\varphi_d\}]$, $P_d(v) = \sum_{n=1}^{d}(u,\varphi_n)\,\varphi_n$ la projection orthogonale de v sur V_d. Puisque ℓ est linéaire et continue, on a : $|\ell(v)| \leq M_1\|v\|$ et $|\ell(P_d(v))| \leq M_1\|P_d(v)\| \leq M_1\|v\|$. Puisque S est bornée, il existe une constante $\alpha \in \mathbb{R}$ telle que $\|v\| \leq \alpha$, $\forall v \in S$ et on a $|\ell(P_d(v))| \leq M_1\alpha$, $\forall\, v \in S$. On pose :
$$L(d,\lambda) = \frac{E\left(\ell(P_d(v))\,g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)}$$
De (5.4.1)on a :
$$|L(d,\lambda)| \leq \frac{E\left(|\ell(P_d(v))|\,g(\lambda,J(v))\right)}{E\left(g(\lambda,J(v))\right)} \leq M_1\alpha.$$

3- Soit $L = \limsup L(d,\lambda)$. Alors,

$$\forall \varepsilon > 0\text{: } \lambda \geq \lambda_0(\varepsilon) \text{ et } d \geq d_0(\varepsilon) \Longrightarrow \sup L(d,\lambda) - \varepsilon \leq L \leq \sup L(d,\lambda)$$

Puisque $L(d,\lambda) \longrightarrow \ell(P_d(u))$ pour $\lambda \longrightarrow +\infty$:

$$\forall \eta > 0\text{: } \lambda \geq \lambda_1(\eta,d) \Longrightarrow \ell(P_d(u)) - \eta \leq L(d,\lambda) \leq \ell(P_d(u)) + \eta.$$

Ainsi, $\lambda \geq \lambda_1(\eta,d) \Longrightarrow \ell(P_d(u)) - \eta \leq \sup L(d,\lambda) \leq \ell(P_d(u)) + \eta$ et

$$d \geq d_0(\varepsilon) \text{ et } \lambda \geq \lambda_1(\eta,d) \Longrightarrow \ell(P_d(u)) - \eta - \varepsilon \leq L \leq \ell(P_d(u)) + \eta.$$

Pour $\eta \longrightarrow 0$, on obtient $d \geq d_0(\varepsilon) \Longrightarrow \ell(P_d(u)) - \varepsilon \leq L \leq \ell(P_d(u))$. En outre, puisque $P_d(u) \longrightarrow u$ fortement dans V, $\ell(P_d(u)) \longrightarrow \ell(u)$: on a $\ell(u) - \varepsilon \leq L \leq \ell(u)$. Donc, en prenant $\varepsilon \longrightarrow 0 : L = \ell(u)$.

4- De la même manière, on obtient $\liminf L(d,\lambda) = \ell(u)$. Ainsi, $\liminf L(d,\lambda) = \limsup L(d,\lambda) = \ell(u)$. Donc $\ell(u) = \lim L(d,\lambda)$.

5- Soit

$$u(d,\lambda) = \frac{E(P_d(v)\, g(\lambda,J(v)))}{E(g(\lambda,J(v)))} = \sum_{n=1}^{k} a_n(\lambda)\,\varphi_n;\ a_n(\lambda) = \frac{E((v,\varphi_n)\, g(\lambda,J(v)))}{E(g(\lambda,J(v)))}.$$

et $\ell \in \mathscr{L}(V,\mathbb{R})$. On a $\ell(u(d,\lambda)) = \sum_{n=1}^{k} a_n(\lambda)\,\ell(\varphi_n) = L(d,\lambda)$. Ainsi, comme précédemment établi, $\ell(u(d,\lambda)) = L(d,\lambda) \longrightarrow \ell(u)$ pour $\lambda \longrightarrow +\infty, d \longrightarrow +\infty$. So, $u(d,\lambda) \rightharpoonup u$, faiblement dans V.

□

Démonstration. du corollaire : Si $E((v,\varphi_n)\, g(\lambda,J(v))) = (E(vg(\lambda,J(v)),\varphi_n)$, alors

$$a_n(\lambda) = \frac{(E(v),\varphi_n)\, g(\lambda,J(v))}{E(g(\lambda,J(v)))} \Longrightarrow u(d,\lambda) = \frac{P_d(E(vg(\lambda,J(v)))}{E(g(\lambda,J(v)))}.$$

En prenant $d \longrightarrow +\infty$, on a :

$$u_\lambda = \sum_{n=1}^{+\infty} a_n(\lambda)\,\varphi_n = \frac{E(vg(\lambda,J(v)))}{E(g(\lambda,J(v)))}.$$

Puisque $u_\lambda \rightharpoonup u$, faiblement dans V, pour $\lambda \longrightarrow +\infty$, on a le résultat. □

Remark 5.6.4 *Situation de la dimension finie*

Comme observé précédemment, notre objectif principal est l'application de la théorie à des situations de dimension infinie. Cependant, il est intéressant de noter que les résultats s'étendent à la détermination de minima globaux de fonctionnelles de dimension finie, c-à-d, à des situations où V est un espace de dimension finie, tel que, par exemple, $V = \mathbb{R}^k$. Dans ce cas, μ est une mesure finie standard sur $\mathbb{R}^k$ (par exemple, la mesure de probabilité associée à une distribution standard sur $\mathbb{R}^k$) et l'évaluation de $\int_S U(v)\,d\mu$ devient une intégration standard sur $\mathbb{R}^k$. En outre, la convergence de toutes les projections sur des sous-espaces de dimension finie (cf. théorème 5.6.1) implique la convergence forte sur $\mathbb{R}^k$: on a

$$\frac{E(vg(\lambda,J(v)))}{E(g(\lambda,J(v)))} \xrightarrow[\lambda \longrightarrow +\infty]{} u \text{ dans } V. \tag{5.14}$$

5.7 APPROCHE NUMÉRIQUE

Soit $V = H^1(a,b)$, et une fonctionnelle de la forme :

$$J(v) = \int_a^b f(t,v,v')dt \quad (a,b) \in \mathbb{R}^2.$$

La résolution d'un problème d'optimisation à dimension infinie, faisant intervenir des fonctions comme inconnues, a pour but de trouver un élément v^* de l'espace fonctionnel V réalisant le minimum global du fonctionnel $J : V \to \mathbb{R}$ sur un ensemble borné $S \subset V$:

$$v^* = \min_{v \in S} J(v), \quad i.e., v^* \in S, J(v^*) \leq J(v), \forall v \in S.$$

On considère aussi, trois familles de base d'éléments non orthogonales :
F1 : La famille des polynômes : $\varphi_i(t) = t^{i-1}, \quad pour\ i \geq 1.$
F2 : La famille trigonométrique : $\varphi_1(t) = 1,\ \varphi_{2i+1}(t) = \frac{1}{\sqrt{2}} sin((i)t),$
$\varphi_{2i+2}(t) = \frac{1}{\sqrt{2}} cos((i)t) \quad pour\ i \geq 1.$
F3 : La famille des éléments finis : formée par la réunion dénombrable pour $k \geq 1$ d'élements finis de forme $P1$ ayant des noeuds sur $a, a+h, a+2h, ..., b$, $h = (b-a)/k$,

Ainsi $v \subset S$ peut être formulé come : $v(t) = \sum_{i=1}^{k} u_i \varphi_{n_i}(t),\ i,k,n_i \in \mathbb{N}^*$ et $u_i \in \mathbb{R}$.

On propose d'utiliser une approximation numérique basée sur des échantillons finis de variables aléatoires intervenant dans l'expression et une approximation de la limite. Une mise en oeuvre numérique peut être effectuée en prenant une grande valeur fixe de λ pour représenter la limite $\lambda \to +\infty$. Afin d'éviter un débordement(oveflow), λ doit être augmentée progressivement jusqu'à la valeur désirée et il peut être pratique d'utiliser des fonctions positives pour J (par exemple, en ajoutant une constante à la fonction originale J). Un échantillon fini de V est généré, selon une probabilité P -cela consiste simplement à générer N points admissibles $(v_1, v_2, ..., v_n) \in S$- et des estimations des moyens sont utilisées pour approcher les moyens exactes, ce qui conduit à :

$$v^* \simeq v_a^* = \frac{\sum_{i=1}^{N} v_i \ \ g(\lambda, J(v_i))}{\sum_{i=1}^{N} g(\lambda, J(v_i))}$$

Les mêmes résultats peuvent être obtenus si l'on remplace les points d'échantillonnage générés v_i par un échantillon des coefficients u_i, avec moins de variables (en général), selon une famille de bases donnée. Dans ce cas, on obtient la moyenne du vecteur des coefficients comme suit :

$$u^* \simeq u_a^* = \frac{\sum_{i=1}^{N} u_i \ \ g(\lambda, J(v_i))}{\sum_{i=1}^{N} g(\lambda, J(v_i))}, \quad avec \quad v_i = \sum_{i=1}^{k} u_i \varphi_{n_i}(t) \tag{5.15}$$

On considère également l'amélioration des approximations, obtenues par l'équation, en utilisant l'algorithme de Nelder Mead (NM) pour les petites dimensions et l'algorithme essaim de particules (PSO) pour les grandes dimensions (voir la section 1.1.5.5). La fonction optimale peut être écrite comme : $v_a^*(t) = \sum_{i=1}^{k} u_a^* \varphi_{n_i}(t)$. Pour mettre en oeuvre cette méthode, nous avons besoin de spécifier la famille des fonctions de base et de choisir un ensemble de coefficients pour obtenir l'estimation de v. Il est intéressant de noter

que cette procedure n'implique pas des approximations en dimension fini de la solution u, puisque les valeurs de k et n sont arbitraires, cette méthode fonctionne sur tout l'espace $\mathbb{R}_0^\infty$. Par conséquent, nous avons élaboré trois scénarios de l'approximation numérique basés sur ces considérations. Dans le premier scénario, notée $S1$, nous considérons un nombre fixe $k \in \mathbb{N}^*$, on pose $n = (1,2,...,k)^T$, et on génère un élément $u \in \mathbb{R}^k$ comme suit :

Algorithme 10 : Scénario S1

Require: φ famille de base, $k, N \in \mathbb{N}^*, c \in \mathbb{R}$
Ensure: *Approximation de la solution en utilisant la première strategie*
for i=1 to N **do**
$n \leftarrow (1,2,...,k)^T$
Générer u pour une distribution uniforme dans $[-c,c]$
Calculer $v_i = \sum_{i=1}^{k} u_i \varphi_{n_i}$
end for
Approcher la solution avec l'équation 5.15
End

Dans le deuxième scénario, noté S2, On consideère un nombre donné $k \in \mathbb{N}$ et on génère un élément de $(\mathbb{N}^*)^k \times \mathbb{R}^k$ comme suit :

Algorithme 11 : Scénario S2

Require: φ Famille de base, $(k, N, n_{max}) \in (\mathbb{N}^*)^3, (\theta, c) \in \mathbb{R}^2$,
Ensure: *Approximation de la solution en utilisant la deuxième stratégie*
for i=1 to N **do**
Générer n par k échantillons à partir de $\mathbb{N}$ en utilisant la distribution de Poisson $P(\theta)$ ou la distribution uniforme dans $\{1,...,n_{max}\}$.
Générer u à partir de la distribution uniforme dans $[-c,c]$
Calculer $v_i = \sum_{i=1}^{k} u_i \varphi_i$
end for
Approcher la solution par l'équation 5.15
End

Dans la troisième strategie, notée S3, on procède suivant l'algorithme qui suit :

Algorithme 12 : Scénario S3

Require: φ famille de base, $(k, k_{max}, N, k_0, n_{max} \in (\mathbb{N}^*)^5, (\theta, \sigma, c) \in \mathbb{R}^3$,
Ensure: *Approximation de la solution en utilisant la troisième strategie*
for i=1 to N **do**
Générer k comme k_{max} échantillons de $\mathbb{N}$ en utilisant la distribution de Poisson $P(\sigma)$ ou une distribution uniforme dans $\{1,...,k_{max}\}$.
Générer n comme k échantillons de $\mathbb{N}$ en utilisant la distribution de Poisson $P(\theta)$ ou une distribution uniforme dans $\{1,...,n_{max}\}$.
Générer u à partir d'une distribution uniforme dans $[-c,c]$
Calculer $v_i = \sum_{i=1}^{k} u_i \varphi_i$
end for
Approcher la solution avec l'équation 5.15
End

On remarque que les scénarion $S2$ et $S3$ travaillent sur un espace fonctionnel de dimension infinie. Pour $S2$, seule la longueur de l'élément v_i est fixée : il est formé par k éléments de $\Phi = \{\varphi_n\}_{n \in \mathbb{N}^*}$. Les valeurs de n et u couvrent tout l'espace $(\mathbb{N}^*)^k \times \mathbb{R}^k$, et par

conséquent, v couvre toute la base ϕ en tenant compte des combinaisons linéaires de k éléments de ϕ. Pour $S3$, la longueur de l'élément v_i n'est pas fixée et v couvre l'ensemble des combinaisons linéaires finies d'éléments de ϕ.
Toutes les intégrales sont numériquement évaluée en utilisant une approximation quadratique basé sur plusieurs points.

5.8 Résultats Expérimentaux

Dans cette section, nous nous intéressons à plusieurs problèmes classiques d'optimisation relevant du domanine du calcul variationnel dans le but de valider noter approche. Compte tenu de la complexité et de la multitude de stratégies, de bases, ainsi que des différents paramètres qui entrent en jeu dans les algorithmes de résolution, nous avons développé une interface informatique avec une base de données des problèmes test, pour nous faciliter la tâche d'expérimentation. Comme illustré dans la figure 5.1, l'interface proposée est composée de cinq zones : (1) pour sélectionner le problème test, (2) pour choisir la stratégie adoptée et ses paramètres, (3) pour définir la base, (4) pour sélectionner l'algorithme à utiliser pour l'amélioration de l'approximation et (5) d'un affichage des résultats sous forme de courbes pour faire la comparaison avec la solution théorique.

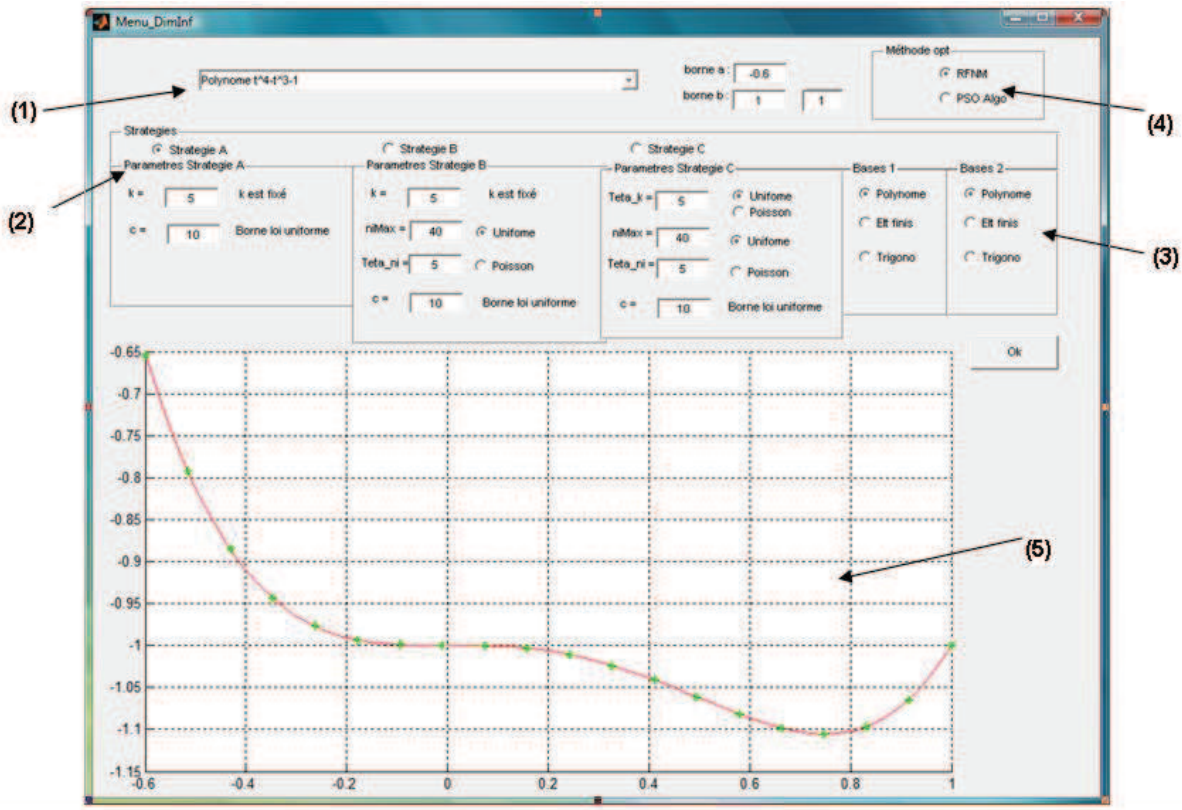


FIGURE 5.1 – *Interface informatique des tests.*

La taille de l'échantillon utilisé pour la formule de représentation dépend de la dimension des variables (200 pour dim ≤ 3 ; 500 Pour dim $4-5$ et 1200 pour $dim \geq 6$). En ce qui concerne NM, nous avons adopté les paramètres standard recommandées par les auteurs, les critères d'arrêt utilisés sont : Nombre maximal d'évaluation de la fonction, $maxfun = 5000$, le nombre maximal d'itération est fixé à $maxiter = 2500$, la tolérance sur la variation relative de la fonction est de $Tolf = 10^{-8}$, et la tolérance sur

sur la variation relative des variables x est $tolx = 10^{-8}$. Pour PSO, nous avons adopté les paramètres suivants : taille de la population = 30, les critères d'arrêt sont : nombre maximal des itérations (réglé sur 300), et le nombre maximum d'itérations continues sans amélioration de la solution est fixé à 10 ; et la longueur d'un voisinage est fixé à 11. Pour faire face aux contraintes impliquant les problèmes de test, nous avons utilisé la méthode de pénalité pour transformer le problème d'optimisation sous contrainte en un problème sans contraintes (voir la section 1.1.6). Les algorithmes proposés sont programmées sous Matlab 7.0, en utilisant le système d'exploitation Windows XP avec un processeur Intel Core Duo 2, 2.0 GHz et 1 Go de RAM.
Les résultats présentés sont exprimés en termes de l'erreur relative (Err) et de temps CPU (CPUT en secondes). L'erreur relative Err est la différence absolue entre la solution trouvée et l'optimum global théorique divisé celle ci (quand il est non nul). Les abréviations suivantes sont introduites pour alléger le texte :
Jopt : valeur de J obtenue par le processus d'optimisation ; U : pour une distribution uniforme ; P : pour distribution de Poisson ; Err_{min} : Erreur relative minimale ; Err_{moy} : Erreur relative moyenne.
En raison de la limitation de l'espace, seules quelques expériences sont présentés, choisis pour illustrer des aspects importants.

5.8.1 Influence de la fonction de Pincus

Dans la formule de représentation, l'expression Pincus correspond à $g(\lambda, s) = e^{-\lambda} s$, ce qui est un choix pratique. Dans ces expériences, quatre autres fonctions de descente (continues et strictement décroissantes) ont été utilisées pour résoudre les problèmes test ci-après (paragraphe 5.8.2) en utilisant notre approche :

$$g_1(\lambda, s) = \frac{1}{\lambda\ s^3}$$

$$g_2(\lambda, s) = \frac{1}{\lambda\ ln(s)}$$

$$g_3(\lambda, s) = \frac{e^{-\lambda}}{s^3}$$

$$g_4(\lambda, s) = 10^{(-\lambda\ s)}$$

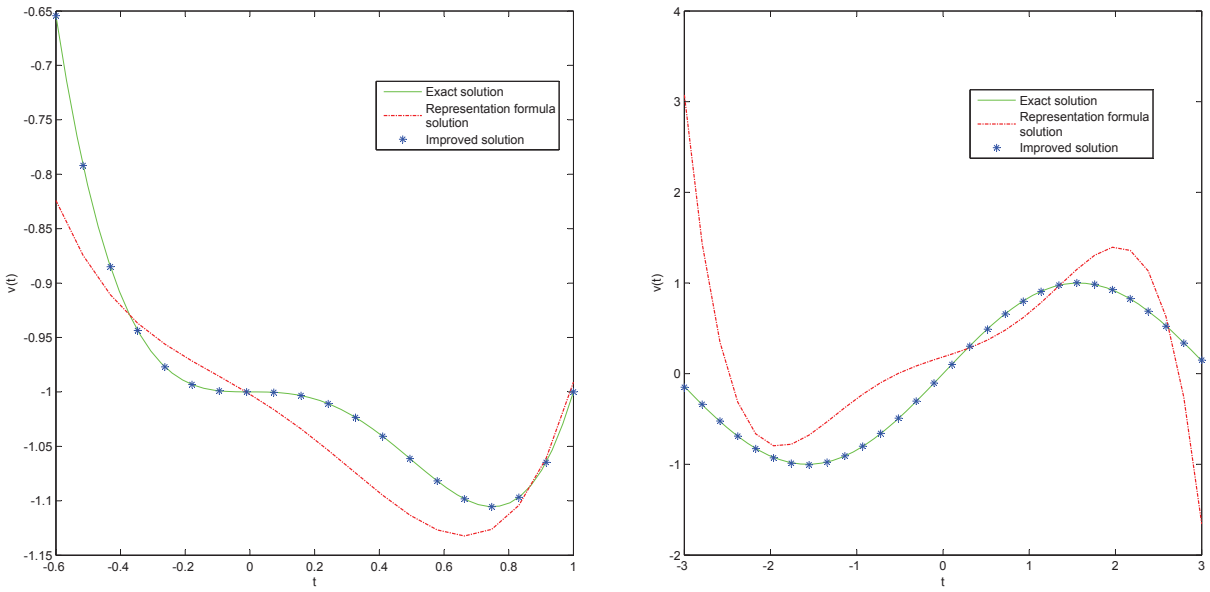
Ces tests ont montré que le choix de la fonction n'a pas d'effet significatif sur la qualité de la solution et de la performance des algorithmes utilisés , seulement une faible différence en terme de temps d'exécution a été observée.

5.8.2 Problèmes élémentaires

$$J(v) = \int_a^b (v(t) - u_0(t))^2 dt$$

La solution triviale de ces problèmes est $u(t) = u_0(t)$. Le test est fait pour deux cas de figure :

1. Cas de polynôme : $u_0(t) = -1 - t^3 + t^4, a = -0.6, b = 1$.
2. Cas de sinus : $u_0(t) = sin(t), a = -3, b = 3$.



(a) (b)

FIGURE 5.2 – *Résultat obtenu par la formule de représentation seule, Résultat amélioré et la solution exacte pour le problème du polynôme (a) et pour le problème du sinus (b) pour l'essai N°5, tableau 5.1 et essai N°3, tableau 5.2 respectivement.*

TABLE 5.1 – *Problème test 1 : Polynôme, pour la stratégie S1*

N°	Base	k	c	Jopt	CPUT
1	Polynom	2	5	4,02E-03	1,70
2	Polynom	3	5	1,93E-03	1,71
3	Polynom	4	5	1,54E-03	3,56
4	Polynom	5	5	1,35E-19	4,46
5	Trigo	2	5	3,95E-03	1,02
6	Trigo	3	5	3,32E-03	1,67
7	Trigo	4	170	2,31E-03	5,75
8	Trigo	5	8000	1,94E-03	8,18
9	Trigo	10	8000	1,75E-03	38,09
10	FE	4	5	1,85E-03	3,36
11	FE	6	5	3,35E-04	7,90
12	FE	8	5	1,01E-04	56,85
13	FE	10	5	1,85E-05	432,30
14	FE	20	5	6,30E-20	944,43

Pour ce problème, nous avons adopté la stratégie S1. Les résultats sont exposés dans le tableau 5.1 pour le problème du polynôme et dans le tableau 5.2 pour le problème sinusoïdale. On observe, comme prévu, que la qualité des résultats s'améliore lorsque k augmente pour une base donnée. La Figure 5.2 représente les résultats des tests obtenus par la formule de représentation seule, par son amélioration en utilisant l'algorithme NM ou PSO, et la solution exacte correspondant à l'essai $N°5$, du tableau 5.1 et l'essai $N°3$ du tableau 5.2. Nous observons qu'une amélioration significative est obtenue par l'utilisation de procédures d'optimisation à partir du résultat de base fournie par la formule de représentation seule. Dans le cas polynomial, comme prévu, les résultats de base polynomiale sont les meilleurs. De même pour le cas sinusoïdale en utilisant la base trigonométrique. La base d'éléments finis donne de bons résultats pour les grandes

TABLE 5.2 – *Problem test 1 : Sinus, pour la stratégie S1*

N°	Basis	k	c	Jopt	CPUT
1	Polynom	3	5	9,89E-01	3,64
2	Polynom	4	5	2,04E-02	5,98
3	Polynom	6	5	1,07E-04	122,82
4	Polynom	8	1	4,80E-08	244,09
5	Trigo	2	5	6,89E-01	1,59
6	Trigo	3	5	1,36E-17	1,95
7	Trigo	4	5	9,40E-19	3,35
8	FE	4	5	7,91E-02	9,17
9	FE	6	5	1,45E-02	28,33
10	FE	8	5	2,51E-03	44,77
11	FE	10	5	3,98E-06	915,34
12	FE	20	5	2,68E-33	2377,26

valeurs de k avec un temps CPU plus important.

5.8.3 Problème du Brachistochrone

Le problème de la trajectoire la plus rapide entre deux points (le problème brachistochrone) fut soumis par Jean Bernoulli à ses contemporains en 1696. Une courbe brachistochrone, ou la courbe de descente la plus rapide, est la trajectoire la plus rapide permettant à une masse ponctuelle soumise uniquement à la gravité et sans vitesse initiale d'aller d'un point A sans vitesse initiale à un point B, sous l'action de la pesanteur (constante) et en l'absence de frottement(Pedregal 2003).

$$J(v) = \int_a^b \frac{\sqrt{1+v'^2}}{\sqrt{2gv}} dt.$$

La solution du problème du brachistochrone est donnée sous forme paramètrique comme suit :
$t = A(\theta - sin(\theta)$, et $u = A(1 - cos(\theta))$. On prend $A = 1$, $a = 0$ et $b = 4.7568$.

Pour ce problème, nous avons adopté la stratégie S2. Nous exposons certains des résultats obtenus dans le tableau 5.3. Les résultats montrent que pour des valeurs de θ ou n_{nmax} inférieur à k, l'optimum n'est pas atteint dans la plupart des cas. En général, nous avons observé que la distribution de Poisson est plus lente que la loi uniforme. La base des éléments finis fournit de meilleurs résultats pour des valeur élevées de k. La figure5.3 montre les résultats fournis par la formule de représentation, son amélioration par NM ou PSO et la solution exacte, et ce pour le test $N^\circ 12$, tableau 5.3. Comme dans l'exemple précédent, une amélioration significative est obtenue en utilisant des procédures d'optimisation du point de départ fourni par la formule de représentation.

5.8.4 Problème Exponentiel

$$J(v) = \int_a^b (v^2 + v'^2) dt, \quad v(0) = 1, v(1) = 0,$$

pour a=0 et b=1, la solution est $u(t) = Ae^t + Be^{-t}, A = \frac{-1}{(e^2-1)}$ et $B = \frac{e^2}{(e^2-1)}$

Pour ce problème, nous avons adopté la stratégies $S3$ et nous avons utilisé la distribution uniforme pour la génération de la valeur de k. Nous observons que la base

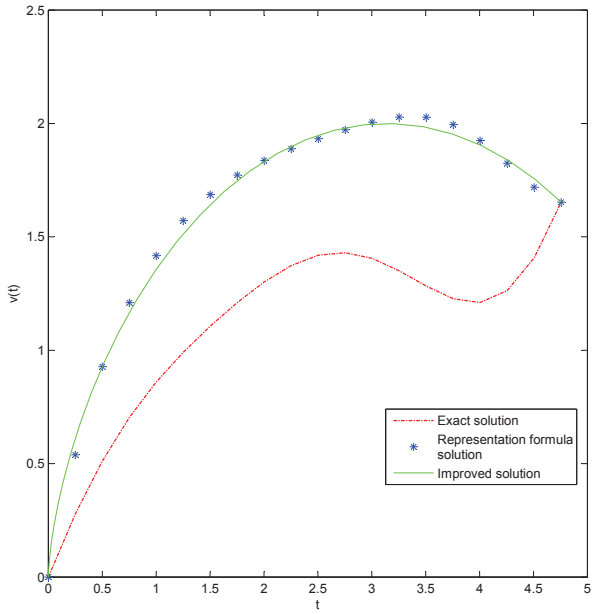


FIGURE 5.3 – *Solution par le formule de représentation, améliorée et la solution exacte pour le problème du brachistochrone pour le test N°12, Tableau 5.3.*

TABLE 5.3 – *Problème test 2 : Brachistochrone, pour la stratégie S2*

$N°$	Base	k	c	Distribution	n_{max}/θ	Err	CPUT
1	Polynom	5	320	U	3	1.61%	169,10
2	Polynom	5	320	U	5	0.70%	216,85
3	Polynom	5	320	P	5	0.51%	226,85
4	Polynom	5	320	P	7	0.51%	273,98
5	Polynom	7	320	U	3	1.61%	259,45
6	Polynom	7	5000	U	5	3.34%	281,94
7	Polynom	7	5000	P	5	70.23%	639,72
8	Polynom	7	5000	P	7	0.35%	2822,00
9	Trigo	5	21	U	3	1.00%	201,22
10	Trigo	5	21	U	5	0.83%	248,85
11	Trigo	5	21	P	3	1.07%	845,18
12	Trigo	5	21	P	4	0.84%	948,48
13	Trigo	7	15	U	5	0.64%	1054,49
14	Trigo	7	14	U	7	0.72%	1743,05
15	Trigo	7	14	P	5	0.78%	1823,93
16	Trigo	7	14	P	6	0.80%	1130,53
17	FE	6	14	U	4	6.36%	381,49
18	FE	6	2	U	6	3.65%	551,10
19	FE	6	2	P	4	13.74%	276,11
20	FE	6	2	P	6	8.70%	650,47
21	FE	10	10	U	8	3.65%	2331,64
22	FE	10	10	U	10	0.19%	943,42
23	FE	10	10	P	8	7.92%	1259,61
24	FE	10	10	P	10	0.19%	3751,12

polynômial et celle des éléments finis fournissent une bonne approximation de l'optimum spécialement pour $k = 4$. la base trigonométrique n'est pas adaptée à ce problème, et ne parvient pas à atteindre l'optimum (voir le tableau 5.4). La raison principale est que la base trigonométriques, d'une part, nécessite des valeurs de k assez grandes, et d'autre part, les coefficients associés doivent aussi être assez grands, ce qui rend le problème

TABLE 5.4 – *Problème test 3 : Problème exponential, pour la stratégie S3*

$N°$	Base	c	Distribution(k)	k_{max}	Distribution(n)	n_{max}/θ	Err	CPUT
1	Polynom	5	U	2	U	5	3.26%	31,17
2	Polynom	5	U	2	P	2	3.26%	2,64
3	Polynom	5	U	4	U	5	0.25%	64,08
4	Polynom	5	U	4	P	3	0.24%	69,69
5	Trigo	2	U	3	U	3	281.42%	29,10
6	Trigo	2	U	3	P	2	281.67%	149,36
7	Trigo	2	U	4	U	3	114.75%	35,79
8	Trigo	10	U	4	P	3	114.75%	128,80
9	FE	10	U	2	U	3	3.26%	17,37
10	FE	2	U	2	P	2	3.26%	45,63
11	FE	10	U	4	U	3	0.01%	348,26
12	FE	2	U	4	P	3	0.01%	440,51

plus difficile à résoudre avec cette base.

5.8.5 Problème de la chaînette

Le problème de la caténaire (the catenary), également connu sous le nom de chaînette, la détermination de la forme prise, sous l'action d'un champ de pesanteur uniforme, par un fil pesant flexible, homogène, inextensible et suspendu entre deux points. L'expression mathématique d'une courbe caténaire est donnée par(Pedregal 2003) :

$$v(x) = \beta + \gamma\, cosh(\frac{x}{\gamma} + \alpha).$$

α, β, et γ doivent être choisis de telle manière que la courbe caténaire passe par deux positions extrêmes désignées et la longueur totale de la chaînette est la valeur correcte. La formulation du problème de calcul des variations du problème de la chaînette est :

$$\min_v J(v) = \min_v \int_a^b v(t)\sqrt{1+v'^2}dt,$$

sous les contraintes : $v(a) = V_a, v(b) = v_b$ et $\int_a^b \sqrt{1+v'^2}dt = \ell$.
Dans nos expériences, nous avons utilisé : $\alpha = -0.5, \quad \beta = -cosh(0.5), \quad \gamma = 1, \quad a = 0 \;\; et \;\; b = 1$.
Pour ce problème, nous avons adopté la stratégies $S3$ et utilisé la distribution de Poisson pour k. On observe (voir le tableau 5.5) que la base polynomiale fournit la meilleure solution, tandis que la base des éléments finis fournit une bonne approximation de l'optimum spécialement pour $k = 8$, mais à un coût de calcul plus élevé. La base trigonométrique ne parvient pas à atteindre l'optimum global.

5.8.6 Géodésiques dans un cylindre

Une courbe, tracée sur une surface, s'appelle une géodésique si, quelque soient deux point de cette courbe, le chemin le plus court est celui qui les joint tout en restant sur la même surface et en suivant cette courbe(Pedregal 2003).
Soit C le cylindre formé par les points (x,y,z) tels que $x^2+y^2 = 1$, et soit P et Q deux points distincts de C. Nous cherchons le plus court chemin C allant de P à Q. Sans perte de généralité, on peut supposer que $P = (1,0,0)$. On pose le problème en coordonnées cylindriques (r,θ,z) avec $r = 1, \quad -\pi \leq \theta \leq \pi$, définissant C. Soit $Q = (1,\theta_0,z_0)$,. on peut représenter une courbe arbitraire joignant $(1,0,0)$ et $(1,\theta_0,z_0)$ sous la forme :

TABLE 5.5 – *Problème test 4 : Catenary, pour la stratégie S3*

N°	Base	c	Distrib.(k)	σ	Distrib.(n)	n_{max}/θ	Err	CPUT
1	Polynom	2	P	3	U	3	1.43%	21,28
2	Polynom	2	P	3	P	4	1.42%	69,07
3	Polynom	2	P	4	U	3	0.55%	43,06
4	Polynom	2	P	4	P	5	0.13%	38,96
5	Trigo	2	P	4	U	5	84.89%	126,42
6	Trigo	2	P	4	P	5	84.27%	228,87
7	Trigo	2	P	5	U	6	80.12%	428,87
8	Trigo	2	P	5	P	6	80.67%	493,29
9	FE	2	P	5	U	6	7.73%	1557,07
10	FE	2	P	5	P	6	7.63%	1731,58
11	FE	2	P	8	U	10	3.27%	5557,07
12	FE	2	P	8	P	10	3.32%	5320,00

$\sigma(\theta) = (cos(\theta), sin(\theta), z(\theta)), 0 \leq \theta \leq \theta_0), \theta \in [0, \theta_0]$. Telle courbe est entièrement déterminée par la fonction $z(\theta), z(0) = 0, z(\theta_0) = z_0$. Le coût fonctionnelle à minimiser est la longueur de σ :

$$J(z) = \int_0^{\theta_0} \sqrt{1 + z'(\theta)^2}.$$

La fonction linéaire $u(\theta) = \frac{z_0}{\theta_0}\theta$ représente la géodésique unique reliant ces deux points. Cette fonction linéaire en coordonnées cylindriques correspond à un arc de spirale au-dessus de C

Dans nos expériences, nous avons utilisé $\theta_0 = z_0 = 1$.

Afin de comparer les différentes stratégies, nous avons effectué le test, pour le problème géodésique, 10 fois pour chaque stratégie, et pour différentes bases. nous avons choisi $k = 3$, $c = 5$, et nous avons utilisé la distribution uniforme. Les résultats exposés dans le tableau 5.6 montrent que la base polynômiale réalise les meilleurs résultats (erreurs moindre et moins de temps CPU), et la stratégie $S1$ est la plus rapide tandis que la stratégie $S3$ est la plus lente. Nous constatons, aussi, que pour la base polynômiale, les stratégie $S3$ et $S2$ donnent de meilleurs résultats comparés à la stratégie $S1$, tandis que pour les autres bases, la stratégie $S1$ est meilleure.

TABLE 5.6 – *Problème test 5 : Géodésique, pour les strategies S1, S2 et S3*

N°	Base	Strategy	k	c	Distr(k)	σ	Distr.(n)	n_{max}	Err_{Min}	Err_{Avg}	CPUT
1	Polynom	S1	3	5	-	-	-	-	0.00%	2.35%	3,60
2	Polynom	S2	3	5	-	-	U	4	0.00%	0.51%	5,23
3	Polynom	S3	3	5	U	4	U	4	0.00%	0.39%	5,66
4	Trigo	S1	3	5	-	-	-	-	34.76%	34.76%	8,42
5	Trigo	S2	3	5	-	-	U	4	34.76%	98.80%	9,11
6	Trigo	S3	3	5	U	4	U	4	34.76%	80.39%	10,37
7	FE	S1	3	5	-	-	-	-	0.54%	0.58%	10,31
8	FE	S2	3	5	-	-	U	4	2.80%	9.95%	11,12
9	FE	S3	3	5	U	4	U	4	0.54%	5.08%	12,03

5.9 CONCLUSION

Nous avons établi une représentation de la solution u réalisant le minimum global d'une fonctionnelle J, sur un sous-ensemble S fermé, borné de l'espace de Hilbert séparable V. On a supposé que u est unique sur S, mais aucune hypothèse de convexité n'est impliquée sur S, ou sur J. La représentation fait intervenir une mesure finie μ sur V et une fonction g décroissante et régulière quelconque : des exemples constructifs de μ et

g ont été donnés. La représentation suggère une méthode numérique simple pour le rapprochement de u, ce qui présente l'originalité de ne pas introduire ni une approximation en dimension finie à priori de u ni la résolution d'équations algébriques linéaires ou non. Cependant, il fait intervenir un nombre important d'évaluations de J. Cette méthode a été testée sur quelques problèmes classiques du calcul des variations, en utilisant différentes stratégies et bases et a conduit à de bons points de départ pour une amélioration ultérieure par l'optimisation avec méthodes standards . Des améliorations par Nelder Mead ou optimisation par essaim de particules sur un sous-espace à dimension finie ont été testés et la procédure globale a montré que l'approche est efficace.

CHAPITRE 6

REPRESENTATION DE SOLUTION POUR L'OPTIMISATION MULTI-OBJECTIF : GÉNÉRATION SUFFISANTE DE LA COURBE DE PARETO

DANS ce chapitre, une nouvelle approche, appelée (RSMO), pour générer un nombre suffisant de points pour représenter la solution optimale le long de la frontière de Pareto est présentée. Cette approche peut être considérée comme une formule de représentation de solution du problème d'optimisation multi-objectif. La méthode peut traiter à la fois les problèmes convexes et non convexes. Dans une première étape, nous nous intéresserons aux problèmes d'optimisation multi-objectif convexe, et nous illustrons la méthode et les idées sous-jacentes à travers plusieurs exemples. L'extension aux problèmes non convexe fera l'objet d'autres travaux.

Le chapitre est organisée comme suit : La représentation de solution en optimisation multiobjectif est présentée dans la section 6.2, suivie de la validation expérimentation par un exemple illustaratif et comparaison avec d'autres méthodes dans la section 6.3. la section 6.4 donne une conclusion et perspectives du chapitre.

La principale publication en rapport avec le chapitre :

H. Zidani, R. Ellaia, and J.E. Souza de Cursi. **Representation of Solution for Multiobjective Optimization : RSMO for Generating a Suffisant Pareto Front**, *soumis au 'Computational & Applied Mathematics'*, (ISSN : 0101 – 8205 (print version), 1807 – 0302 (online)).

6.1 INTRODUCTION

Les problèmes d'optimisation multicritères sont très fréquents dans le processus d'optimisation pour le domaine de l'ingénierie. Ils sont parmi les problèmes les plus difficiles d'optimisation, qui se posent dans de nombreuses applications du monde réel. Il existe une multitude de méthodes pour résoudre ces problèmes (Ehrgott et Wiecek 2005, Gould 2006, Van Veldhuizen 1999).La plupart d'entre elles utilisent des processus ité-

ratifs pour générer un ensemble de points se rapprochant de la courbe de Pareto en une seule execution (méthode d'optimisation multi-objectif évolutionnaires telle que NSGA II (voir la section 1.2.6.5) ou en transformant les problèmes multi-objectif en une série de problèmes d'optimisation simples (méthode de la somme pondérée, ε-contrainte, NBI, NNC, ...(voir chapitre 1)). Un ensemble de point optimal au sens de Pareto est considéré comme la solution mathématique à un problème multi-objectif. Dans le cas continu, le nombre de solutions Pareto optimales est généralement infini. Ce n'est que dans des cas relativement simples que l'ensemble Pareto optimal peut être déterminé analytiquement. En général, on est satisfait d'obtenir "assez" de points Pareto optimales pour couvrir l'ensemble de la courbe Pareto dans l'espace des solutions. Ce sous-ensemble obtenu pourrait être considéré comme un ensemble représentatif de la courbe de Pareto et sa qualité peut être jugée par exemple par sa capacité à couvrir cet ensemble d'une façon uniforme. Cependant, dans certains problèmes, le coût de production d'un seul optimum de Pareto peut devenir si grand que le concepteur ne peut se permettre que quelques solutions Pareto optimales. La méthode d'optimisation multi-objectif (RSMO) proposée a l'avantage de générer suffisamment de points pour représenter la courbe de Pareto en commençant par les points les plus représentatifs.

6.2 Representation de Solution pour l'optimisation Multi-Objectif(RSMO)

6.2.1 Principe de base

Dans la littérature, les formules de représentation ont été présentées afin de caractériser d'une manière explicite les solutions du problème d'optimisation mono-objectif(voir chapitre 2). A notre connaissance, aucune formule de représentation de solution pour les problèmes d'optimisation multi-objectif n'a été proposée à ce jour. La méthode "Représentation de Solution pour l'optimisation Multi-Objectif" ($RSMO$), qu'on propose dans le cadre de nos travaux de recherche, est une formule de représentation de solution, puisqu'elle suggère d'écrire d'une manière explicite la solution du problème d'optimisatuion multi-objectif. C'est une technique de recherche de la portion d'espace qui contient les points Pareto optimal. Le principe de base de RSMO est fondé sur une observation simple et évidente : Pour un problème d'optimlisation bi-objectif, la surface limitée par la courbe de Pareto et les deux axes $f_1 = f(x_2^*)$ et $f_1 = f(x_1^*)$ constitue la surface minimale de l'ensemble des courbes joignant les points idéaux A et B (voir graphe 6.1).

6.2.2 Formulation

Soit le problème à deux fonctions objectif à valeurs réelles définit comme suit :

$$\left\{ \begin{array}{c} \min\limits_{x} \; f(x) = (f_1(x), f_2(x))^T \\ \textit{sous la contrainte} \\ x \in C \end{array} \right\} \tag{6.1}$$

avec :
f : vecteur des critères à optimiser.
$x \in \mathbb{R}^n$ est le vecteur représentant les variables de décision.
C : l'ensemble des solutions réalisables associé à des contraintes d'égalité, d'inégalité et des bornes explicites. $C = \{ \, x \in \mathbb{R}^n \; / \; h(x) = 0, \;\; g(x) \leq 0 \;\; et \;\; x^L \leq x \leq x^U \, \}$
$S = f(C) = \{(f_1(x), f_2(x)), x \in C\}$: est l'ensemble des points réalisables dans l'espace des critères (espace des objectifs)(voir graphe 6.1).

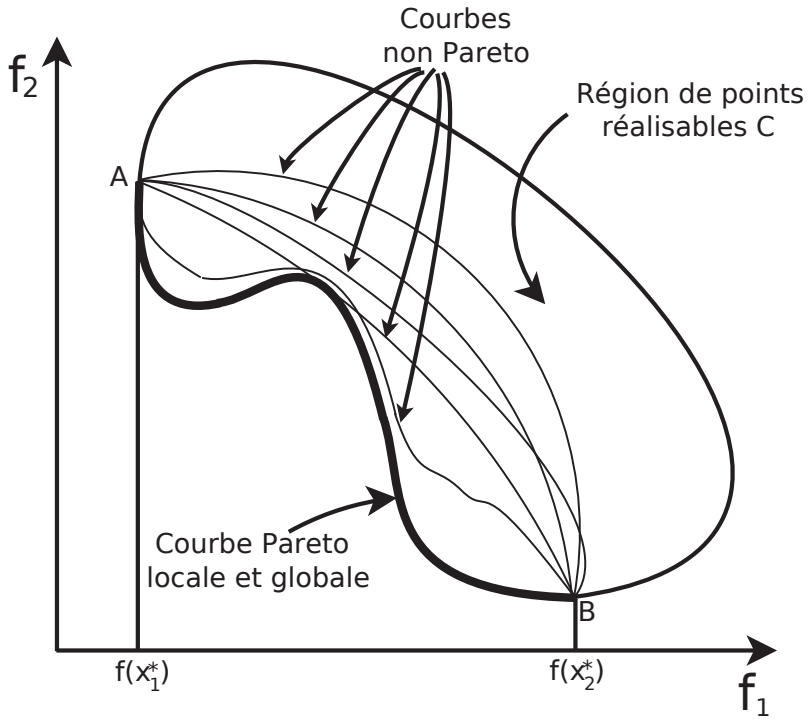


FIGURE 6.1 – *Méthode RSMO : Principe de base*

Soit $H^1(a,b)$ un espace de Hilbert, avec $(a,b) \in \mathbb{R}^2$ et une fonctionnelle de la forme :

$$J(v) = \int_a^b v(t)dt, \qquad v \in H^1(a,b)$$

et soit $E = \{(t, v(t)), t \in [a,b]\}$,
avec $a = f_1(x_1^*)$, $\quad b = f_1(x_2^*)$ (x_1^* et x_2^* sont les solutions optimales pour les fonctions f_1 et f_2 respectivement

La solution de ce problème peut être écrit sous forme d'un problème de calcul variationnel :

$$\left\{ \begin{array}{c} \min\limits_v \ J(v) = \min \ \displaystyle\int_a^b v(t)dt \\ \textit{sous les contraintes} \\ v(a) = f_2(x_1^*), \\ v(b) = f_2(x_2^*), \ \textit{et} \\ E \subset S \end{array} \right\} \tag{6.2}$$

De la même manière, pour le problème d'optimisation à trois fonctions objectif suivant :

$$\left\{ \begin{array}{c} \min\limits_x \ f(x) = (f_1(x), f_2(x), f_3(x))^T \\ \textit{sous la contrainte} \\ x \in C \end{array} \right\} \tag{6.3}$$

la solution est la surface de Pareto qui passe par les trois points $f(x_1^*)$, $f(x_2^*)$ et $f(x_3^*)$ et qui minimise le volume situé entre cette surface et sa projection sur le plan (f_1, f_2).
Le nouveau problème de calcul variationnel à résoudre est :

$$\left\{ \begin{array}{c} \min\limits_v \ J(v) = \min\limits_v \ \displaystyle\int_{a_1}^{b_1}\int_{a_2}^{b_2} v(s,t)dsdt \\ \textit{sous les contraintes} \\ v(a_1, c_2) = f_3(x_1^*), \\ v(c_1, a_2) = f_3(x_2^*), et \\ v(a_1, c_2) = f_3(x_3^*), \\ E \subset S \end{array} \right\} \tag{6.4}$$

avec : $E = \{(t, s, v(t)), t \in [a_1, b_1]$ et $s \in [a_2, b_2]\}$,
$S = f(C)$, $a_1 = f_1(x_1^*)$, $a_2 = f_2(x_2^*)$, $b_1 = f_1(x_3^*)$, $b_2 = f_2(x_3^*)$, $c_1 = f_1(x_2^*)$,
$c_2 = f_2(x_1^*)$ (x_1^*, x_2^* et x_3^* sont les variables optimales pour les fonctions f_1, f_2 et f_3 respectivement.
Pour un problème multi-objectif à m fonctions($m \geq 2$), la représentation de solution peut s'écrire :

$$\left\{ \begin{array}{c} \min\limits_v \ J(v) = \min\limits_v \ \displaystyle\int_{a_1}^{b_1}\int_{a_2}^{b_2} \cdots \int_{a_{m-1}}^{b_{m-1}} v(t_1, t_2, \ldots, t_{m-1})dt_1, dt_2, \ldots, dt_{m-1} \\ \textit{sous les contraintes} \\ v(f_1(x_i^*), f_2(x_i^*), \ldots, f_{m-1}(x_i^*)) = f_m(x_i^*), \ \textit{pour } i = 1 \ \ m, \ \textit{et} \\ E \subset S \end{array} \right\} \tag{6.5}$$

avec : $S = f(C)$, et
$E = \{(t_1, t_2, t_{m-1}, \ldots, t_{m-1}, v(t_1, t_2, \ldots, t_{m-1})), \ t_i \in [f_i(x_i^*), f_i(x_m^*)\}$,
x_i^* est la variable optimale pour la fonction f_i, pour $i = 1 \ \ m$.

6.2.3 Algorithme

Comme mentionné précédement, pour illustrer cette méthode nous nous limiterons aux cas de problèmes convexes, à travers plusieurs fonctions test à deux objectifs. La résolution du problème 6.2 peut se faire de plusieurs manières, soit en construisant la courbe de Pareto point par point ou en générant un ensemble de points en une seule fois. Dans l'algorithme 13, on propose une procédure itérative qui se base sur le calcul de la surface (intégrale) par la méthode des trapèzes, et pour chaque itération, on cherche le meilleur point qui peut représenter le mieux la courbe Pareto, en tenant compte des points identifiés auparavent.

Algorithme 13 : Algorithme RSMO

Require: Np : Nombre maximum de points Pareto désiré, $TolJ$: Erreur relative de la solution
Ensure: *Front de Pareto suffisant*
$t = (f_1(x_1^*), f_1(x_2^*))$
$v = (f_2(x_1^*), f_2(x_2^*))$
$J = \frac{1}{2}.(t(2) - t(1)).(v(1) + v(2))$
$N = 0$
repeat
$N = N + 1$
Construire la fonction J(x) (voir algorithme 14)
Minimiser $J(x)$ par un algorithme global tel que RFNM, on obtient x_N^* et J_N^*
$t \leftarrow [t, f_1(x_N^*)]$
$v \leftarrow [v, f_2(x_N^*)]$
$\delta_{Err} = |\frac{J_N^* - J}{J_N^*}|$
$J = J_N^*$
until $\delta_{Err} \leq TolJ$ ou $N = Np$
End

Algorithme 14 : Fonction $J(x)$

Require: t et v en cours, la variable x
Ensure: *Fonction Approximation de la surface par la méthode des trapèzes*
$t^p \leftarrow [t, f_1(x)]$
$v^p \leftarrow [v, f_2(x)]$
Trier t^p et v^p
$J(x) = \frac{1}{2}.\sum_{i=1}^{N+1}(t_{i+1}^p - t_i^p)(v_i^p + v_{i+1}^p)$
End

6.3 Validation expérimentale

Pour démontrer l'efficacité et la validité de cette nouvelle méthode, trois problèmes multi-objectifs convexes types ont été choisis à partir de la littérature de l'optimisation globale.
Le premier problème, à une seule variable, servira pour bien illustrer la méthode RSMO, et montrer le déroulement de la construction du front de Pareto. Pour les autres problèmes tests, une comparaison avec des méthodes très utilisées en optimisation multiobjectif dans le domaine de l'ingénierie sera considérée. Pour cette comparaison sont utilisées les algorithmes suivants : Normalized Normal Constraint (NNC), Normal Boundary Intersection (NBI), et Nondominated Sorting Genetic Algorithm (NSGA II). La comparaison est faite en utilisant ces méthodes pour générer le front de Pareto pour le deuxième et le troisième problème.

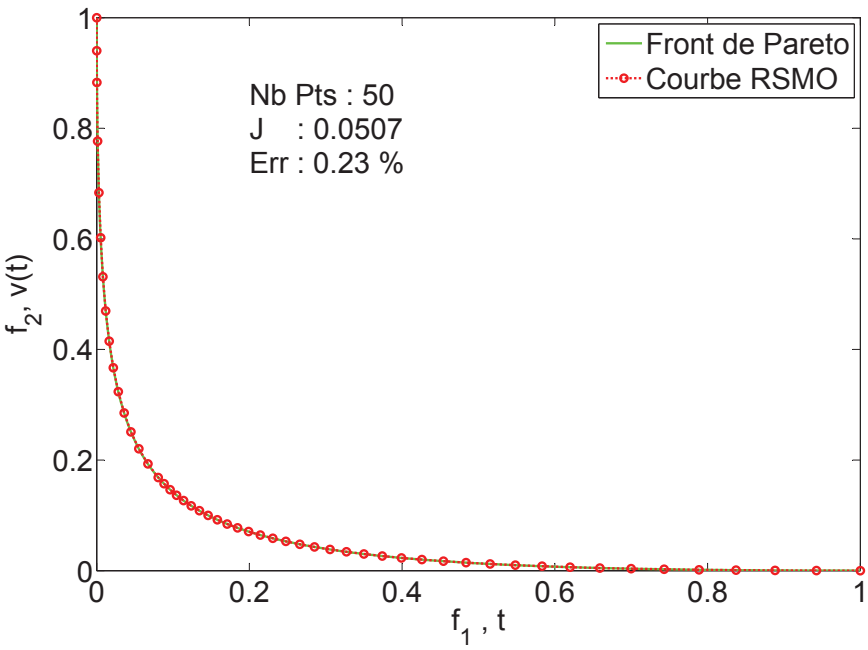


FIGURE 6.2 – *Courbe de Pareto Pour le problème 1 (Zhang) avec RSMO)*

6.3.1 Exemple illustratif : problème Zhang

Pour illustrer le fonctionnement de cette méthode, un problème bi-objectif à une seule variable est considéré (Zhang et Gao 2006). Sa formulation est la suivante :

$$\begin{cases} \min \quad F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T, avec: \\ \qquad f_1(\mathbf{x}) = cosh(\mathbf{x}) \\ \qquad f_2(\mathbf{x}) = \mathbf{x}^2 - 12\mathbf{x} + 35 \\ \text{sujet à} \qquad 0 \leq \mathbf{x} \leq 10. \end{cases} \tag{6.6}$$

La résolution de ce problème, en se basant sur l'algorithme 13, et en utilisant notre algorithm RFNM, avec une erreur relative de $10^{-4}\%$ conduit à la courbe de Pareto de la figure 6.2. On observe que les points générés sont plutôt concentrés sur la courbure du front de Pareto, situé entre $t = 0.07$ et 0.40. En faisant accroître le nombre de points de la courbe de Pareto, on obtient des approximations de cette courbe, avec des erreur relatives de plus en plus faibles. Les graphes 6.3, 6.4, 6.5 et 6.6 sont obtenus pour un objectif de $1, 3, 5, 10, 20, 25, 30$ et 50 points.
On observe que la qualité de la courbe Pareto s'améliore en augmentant le nombre de points ; une approximation acceptable de la courbe de Pareto est obtenue dés que le nombre de points dépasse 7 ; et qu'un nombre de 30 points est suffisant pour représenter convenablement le front de Pareto (avec une erreur relative de 0.52%). La courbe 6.7 qui représente l'évolution de l'erreur relative en fonction du nombre de points cherchés, montre que cette erreur décroit rapidement entre 1 et 10 points, et à partir de 60 points cette erreur tend vers zéro ; l'erreur relatif étant $Err = \frac{J-J^*}{J^*} * 100$, avec J^* est la valeur exacte(pour un nombre N assez grand). Les résultats obtenus, pour ce premier exemple, montrent que la méthode RSMO est efficace pour la génération de suffisement de points pour représenter la courbe de Pareto.

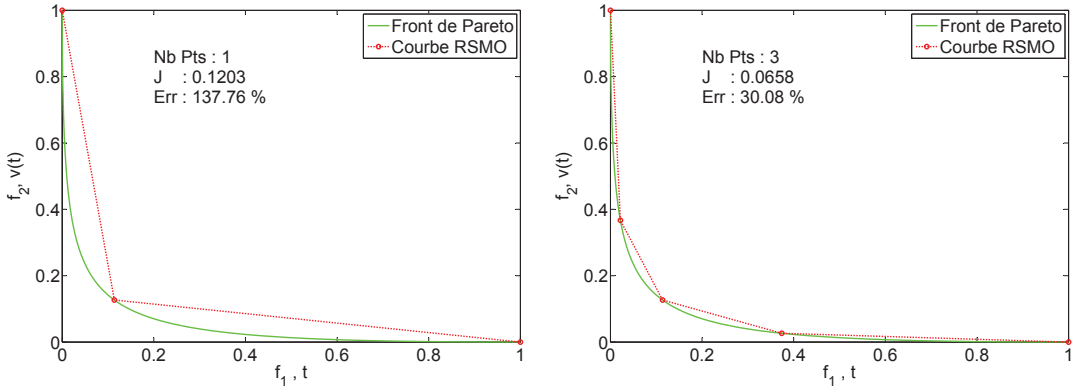


FIGURE 6.3 – *Courbe de Pareto pour 3 et 5 points*

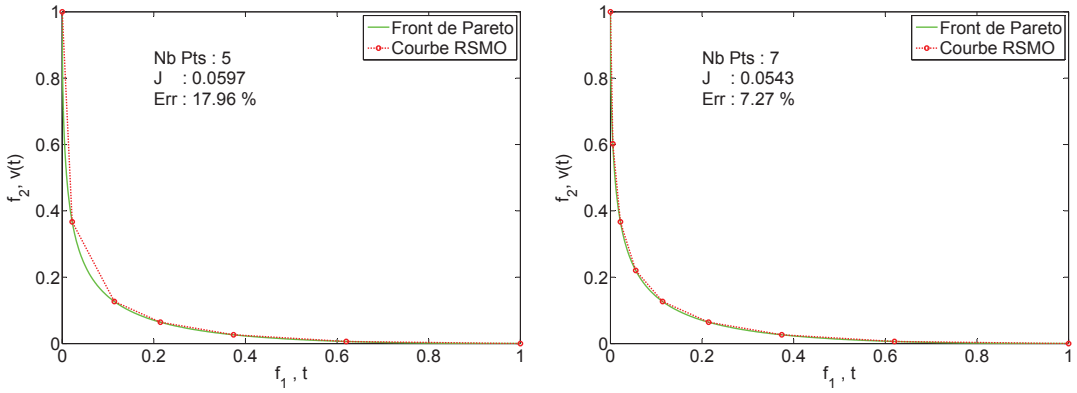


FIGURE 6.4 – *Courbe de Pareto pour 5 et 7 points*

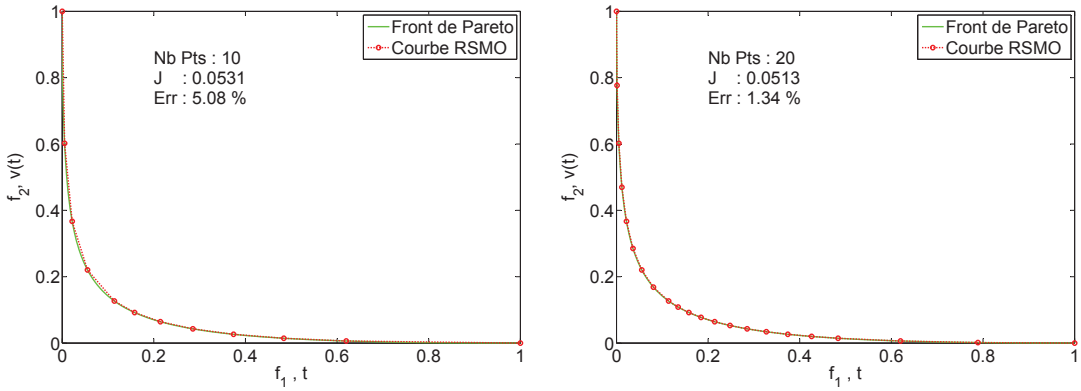


FIGURE 6.5 – *Courbe de Pareto pour 10 et 20 points*

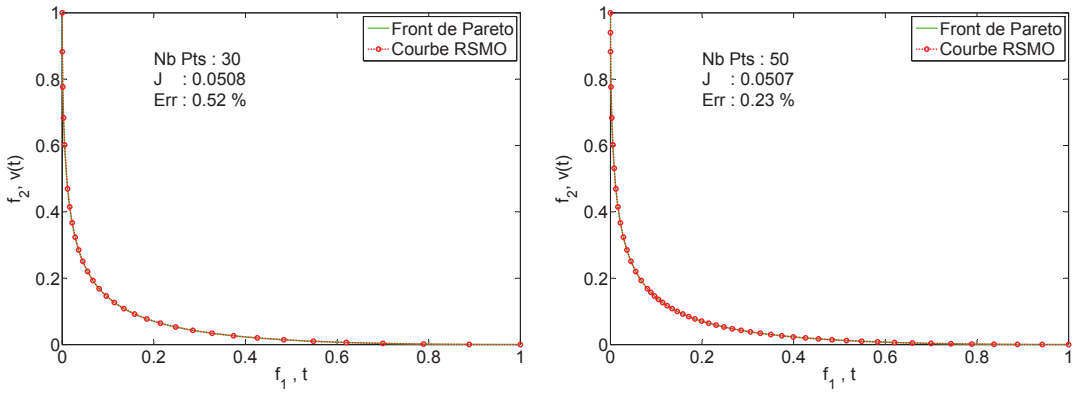


FIGURE 6.6 – *Courbe de Pareto pour 30 et 50 points*

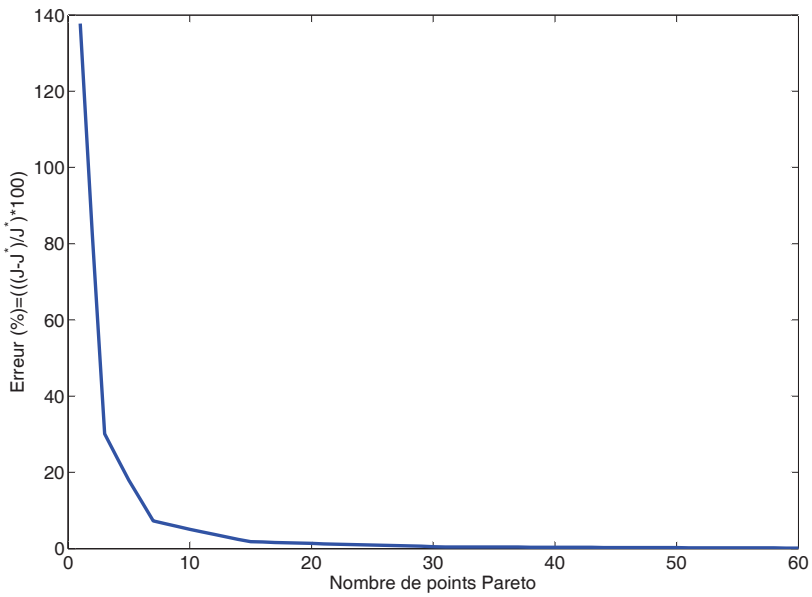


FIGURE 6.7 – *Evolution de l'érreur relative en fonction du nombre de points de Pareto générés par RSMO*

6.3.2 Comparaison avec d'autres méthodes multi-objectif

Dans cette section, on propose de faire une comparaison de la méthode RSMO avec des méthodes très utilisées en optimisation multiobjectif dans le domaine de l'ingénierie, en l'occurrence les méthodes NNC, NBI et NSGA II :

1. **Normal Boundary Intersection (NBI) :** Cette stratégie relativement récente est une méthode déterministe. Elle a un certain nombre d'avantages par rapport aux autres méthodes existantes, y compris une garantie de générer des points répartis sur la courbe de Pareto (Das et Dennis 1998). Elle utilise une paramétrisation géométrique intuitive pour produire une répartition équilibrée de points sur la surface de Pareto, ce qui donne une image exacte de la surface totale. Elle est capable de traiter les problèmes où la surface de Pareto est discontinue ou non lisse. Malheureusement, un point généré par NBI peut ne pas être un point de Pareto si la frontière de l'espace des objectifs contenant les points de Pareto est non convexe (ce qui arrive rarement dans les problèmes posés par les applications réelles). Une formulation mathématique détaillée de la méthode est présentée dans la section 1.2.6.6.
2. **Normalized Normal Constraint (NC) :** La méthode 'Normalized Normal Constraint' (NNC) est une amélioration de la méthode NBI avec des contraintes d'inégalité. Elle utilise des fonctions objectives normalisées et un filtre de Pareto pour éliminer les points non Pareto. Le procédé peut fournir un ensemble de points régulièrement espacés de Pareto optimales dans l'espace des objectifs. Sa performance est indépendante de l'échelles des objectifs. Pour plus de détail sur la formulation de cette méthode, le lecteur peut se référer à la section 1.2.6.7.
3. **Non-dominated Sorting Genetic Algorithm-II (NSGA-II) :** NSGA-II est un algorithme génétique basé sur le principe de la non-domination qui pallie les inconvénients de NSGA tels que le manque d'élitisme et de la nécessité d'un choix à priori de la valeur du paramètre de partage optimal. Elle introduit la notion de rang de non-domination pour évaluer la force de chaque individu. Des procédures

d'élitisme, visant à conserver les meilleures solutions, de génération en génération, contribuent à accélérer la vitesse de convergence et les performances générales des algorithmes génétiques. La diversité est également garantie à l'aide d'une méthode de "crowding" ne nécessitant pas de paramètre supplémentaire. Pour plus de détails sur la méthode, voir la section 1.2.6.5.

6.3.2.1 Problème Deb (1999)

C'est un problème bi-objectif à deux variables (Deb 1999, Lu et Yen 2003) :

$$\begin{cases} \min \quad F(\mathbf{x}) = (f_1(\mathbf{x_1},\mathbf{x_2}), f_2(\mathbf{x_1},\mathbf{x_2}))^T, \qquad avec: \\ \qquad f_1(\mathbf{x_1},\mathbf{x_2}) = sin(\frac{\pi}{2}\mathbf{x_1}) \\ \qquad f_2(\mathbf{x_1},\mathbf{x_2}) = \frac{(1-e^{-\frac{(\mathbf{x_2}-0,1)^2}{0,0001}})+(1-0,5e^{-\frac{(\mathbf{x_2}-0,8)^2}{0,8}})}{arctan(100\mathbf{x_1})} \\ \qquad \text{sujet à} \quad 0 \leq \mathbf{x_i} \leq 1,\ i = 1,2 \end{cases} \tag{6.7}$$

- **Paremètres** :
  Pour NNC et NBI le nombre de points du front de Pareto est fixé à 21 et notre méthode hybride RFNM est utilisée pour résoudre les sous-problèmes générés par ces méthodes. En ce qui concerne NM, nous avons adopté les paramètres standard recommandées par les auteurs, les critères d'arrêt utilisées sont : nombre maximum d'évaluations de la fonction, $maxfun = 6000$, nombre maximum d'itérations, $maxiter = 6000$; tolérance sur l'amélioration de la la fonction $Tolf = 10^{-4}$, et la tolérance sur la variation relative de x est $tolx = 10^{-4}$. En ce qui concerne la taille de l'échantillon pour la formule de représentation, il est fixé à une valeur inférieure à 50. L'algorithme NSGA II est utilisé avec les paramètres suivants : la probabilité de croisement $= 0.8$; fraction Pareto $= 0.45$; Taille de la population $= 1000$, indice de distribution de croisement (Crossover Distribution Indexs) $= 20$. La méthode de pénalité est utilisée pour palier aux contraintes générées par les méthodes NBI et NNC. Les algorithmes proposés sont programmées sous Matlab et fonctionnent sous Windows XP avec processeur Intel Core 2 Duo à 2.0 GHz et 2 Go de RAM. Les résultats présentés sont en termes de temps CPU (CPUT en secondes) et le nombre d'évaluations de la fonction objectif (NevF).

- **Résultats :**
  Les résultats de la comparaison sont exposées dans les figures 6.8, 6.9, 6.10, 6.11, et dans le tableau 6.1.
  On observe que la méthode NSGA II parvient à générer le front de Pareto, sauf pour la partie $f_2 > 6.0$ (voir 6.10) avec un temps CPU et un nombre d'évaluation de la fonction plus élevé que ceux relatifs aux méthodes NBI et NNC ; les points de Pareto ne sont pas uniformément réparties. Les résultats montrent aussi que les méthodes NBI et NNC génèrent un front de Pareto uniformément réparti. La méthode NBI est plus lente et coûteuse en terme d'évaluation de fonction par rapport à la méthode NNC, cela est dû aux contraintes d'égalité qui consomment beaucoup de temps CPU. La qualité du front de Pareto dans la région correspondant à f_1 comprise entre 0.01 et 0.1 n'est pas bonne, puisque les 3 points générés ne suffisent pas pour représenter correctement cette partie (voir figures 6.8 et 6.9).
  On note aussi qu'une amélioration très importante de la qualité de la courbe de Pareto est obtenue par l'utilisation de RSMO. En effet les 17 points qui représentent le front de Pareto sont répartis d'une façon optimale pour couvrir d'une facon efficace cette courbe avec une convergence plus rapide (temps CPU le plus bas et un nombre d'évaluations de la fonction le plus faible) que les autres méthodes.

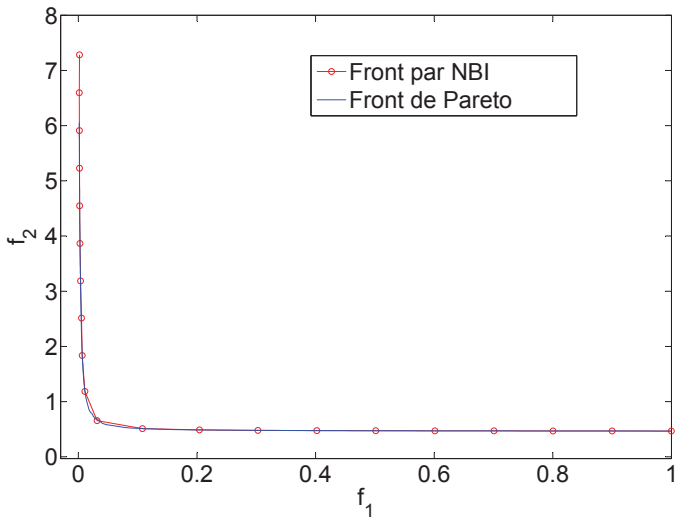


FIGURE 6.8 – *Front de Pareto généré par la méthode NBI pour le problème Deb*

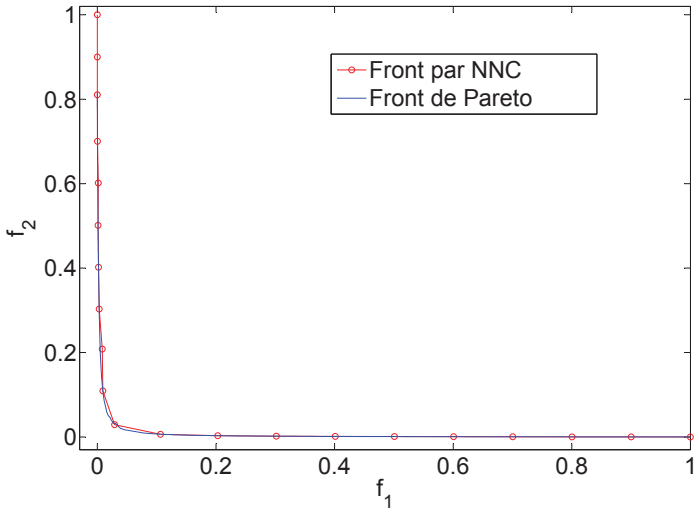


FIGURE 6.9 – *Front de Pareto généré par la méthode NNC(fonction normalisée) pour le problème Deb*

	Méthodes			
	NBI	NNC	NSGA II	RSMO
CPUT(s)	15.90	14.98	17.55	9.21
NevF	45 648	36 817	54 001	15 216

TABLE 6.1 – *Comparaison entres les méthodes NBI, NNC, NSGA II, et RSMO pour le problème Deb*

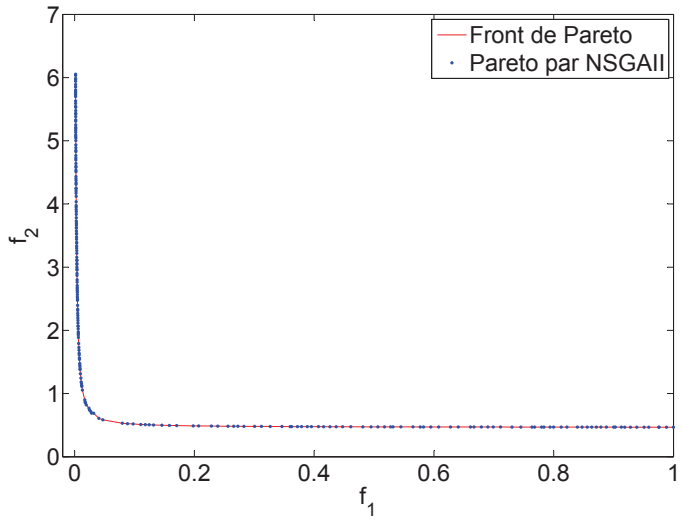


FIGURE 6.10 – *Front de Pareto généré par la méthode NSGAII pour le problème Deb*

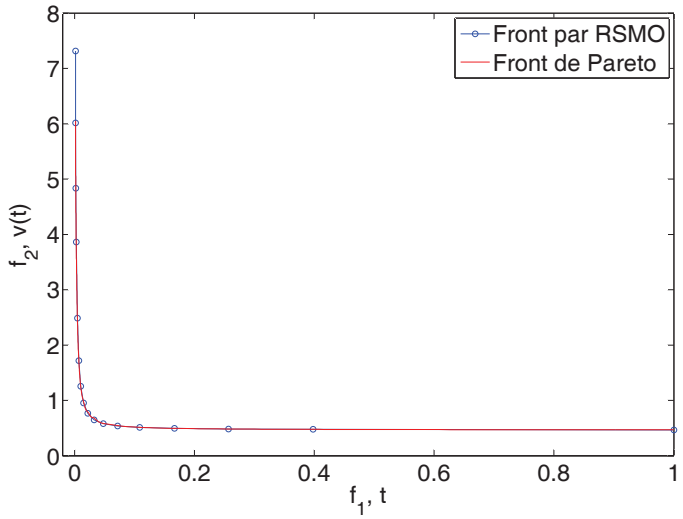


FIGURE 6.11 – *Front de Pareto généré par la méthode RSMO pour le problème Deb*

	Méthodes			
	NBI	NNC	NSGA II	RSMO
CPUT(s)	19.95	18.28	22.26	9.79
NevF	82 299	64 614	103 001	48 564

TABLE 6.2 – *Comparison entres les méthodes NBI, NNC, NSGA II, et RSMO pour le problème ZDT1*

6.3.2.2 Problème de Zitzler ZDT1

Sa formulation est la suivante :

$$\begin{cases} \min \quad F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T, \ \ avec: \\ \qquad f_1(\mathbf{x}) = \mathbf{x_1} \\ \qquad f_2(\mathbf{x}) = g(\mathbf{x}).(1-\sqrt{\frac{\mathbf{x_1}}{g(\mathbf{x})}}) \\ \qquad g(\mathbf{x}) = 1+9.\frac{(\sum_{i=2}^{n}\mathbf{x_i})}{(n-1)} \\ \quad \text{sujet à} \quad 0 \leq \mathbf{x_i} \leq 15. \end{cases} \tag{6.8}$$

- **Paramètres** :
  Pour NNC et NBI le nombre de points du front de Pareto est fixé à 21 et notre méthode hybride RFNM est utilisée pour résoudre les sous-problèmes générés par ces méthodes. En ce qui concerne NM, nous avons adopté les paramètres standard recommandées par les auteurs, les critères d'arrêt utilisées sont : nombre maximum d'évaluations de la fonction, $maxfun = 6000$, nombre maximum d'itérations, $maxiter = 6000$; tolérance sur l'amélioration de la la fonction $Tolf = 10^{-4}$, et la tolérance sur la variation relative de x est $tolx = 10^{-4}$. En ce qui concerne la taille de l'échantillon pour la formule de représentation, il est fixé à une valeur inférieure à 250. L'algorithme NSGA II est utilisé avec les paramètres suivants : la probabilité de croisement $= 0.8$; fraction Pareto $= 0.45$; Taille de la population $= 500$, indice de distribution de croisement (Crossover Distribution Indexs) $= 20$. La méthode de pénalité est utilisée pour palier aux contraintes générées par les méthodes NBI et NNC. Les algorithmes proposés sont programmées sous Matlab et fonctionnent sous Windows XP avec processeur Intel Core 2 Duo à 2.0 GHz et 2 Go de RAM. Les résultats présentés sont en termes de temps CPU (CPUT en secondes) et le nombre d'évaluations de la fonction objectif (NevF).
- **Résultats :**
  Les résultats de la comparaison sont exposées dans les figures 6.12, 6.13, 6.14, et dans le tableau 6.2.
  On observe que la méthode NSGA II génére un front de Pareto situé au dessous du front de pareto optimal (voir figure 6.13, avec un temps CPU et un nombre d'évaluation de la fonction plus élevés que ceux relatifs aux méthodes NBI et NNC ; les points de Pareto ne sont pas uniformément réparties. Les résultats montrent aussi que les méthodes NBI et NNC génèrent un front de Pareto uniformément réparti. La méthode NBI est plus lente et coûteuse en terme d'évaluation de fonction par rapport à la méthode NNC, cela est dû aux contraintes d'égalité qui consomment beaucoup de temps CPU.
  La méthode RSMO a fourni une courbe de Pareto avec 11 points seuelemnt avec une répartition optimale sur cette courbe et avec une convergence plus rapide (temps CPU le plus bas et un nombre d'évaluations de la fonction le plus faible) que les autres méthodes.

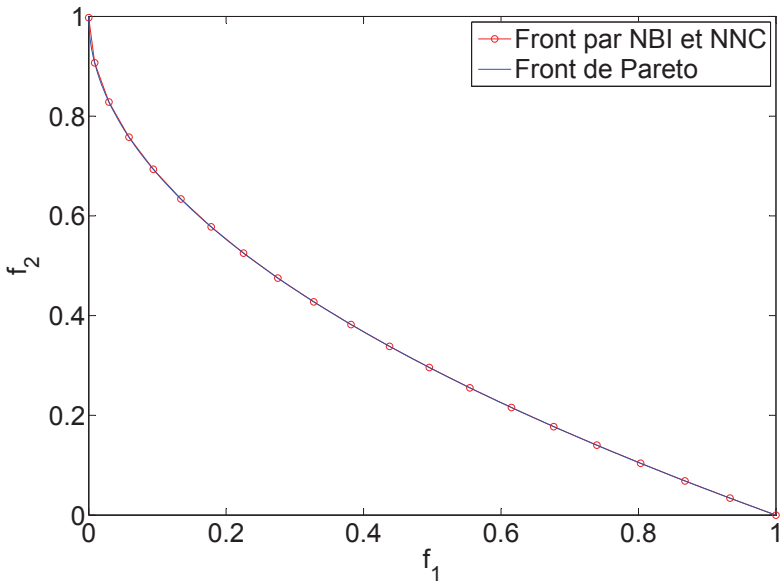


FIGURE 6.12 – *Front de Pareto généré par les méthode NBI et NNC pour le problème ZDT1*

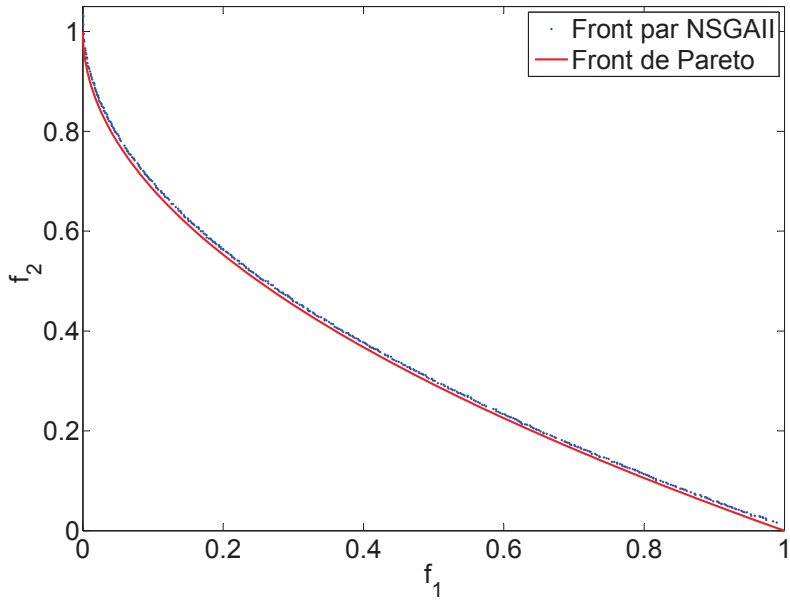


FIGURE 6.13 – *Front de Pareto généré par la méthode NSGAII pour le problème ZDT1*

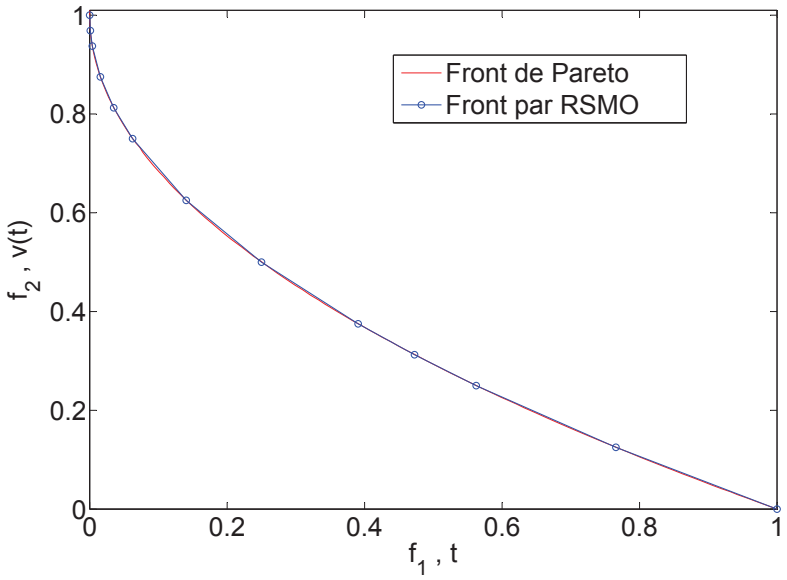


FIGURE 6.14 – *Front de Pareto généré par la méthode RSMO pour le problème ZDT1*

6.4 CONCLUSION

Une nouvelle méthode dite RSMO est présentée. Elle permet de résoudre les problèmes d'optimisation multi-objectif en générant un nombre suffisant de points du front de Pareto. Une implémentation de cette méthode, pour des problèmes convexes, a été proposée. Des tests sur quelques problèmes de la litérature ont été réalisés, et ont démontré l'efficacité et l'efficience de cette approche. Des comparaisons avec des méthodes bien connues dans le domaine de l'optimisation multi-objective ont montré que RSMO est plus efficace et plus efficiente que les autres approches. Des extensions aux problèmes non convexes et des applications dans le domaine de l'ingénierie, ainsi que des propositions de nouvelles implémentations de cette méthodes feront l'objet de nos prochains travaux.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Les résutats des travaux réalisés dans le cadre de cette thèse constituent des contributions à l'optimisation globale, à l'application à l'optimisation de plusieurs structures mécaniques et à l'optimisation multi-objectif :

CONTRIBUTION À L'OPTIMISATION GLOBALE

Dans ce contexte, une nouvelle approche fondée sur une formule de représentation pour résoudre des problèmes d'optimisation globale a été adoptée. Des expérimentations ont été réalisées pour l'optimisation de fonctions non linéaires, non différentiables et multimodaux utilisant cette formule de représentation, hybridée avec l'algorithme génétique et l'algorithm de Nelder Mead, ont montré que RFNM est plus robuste que les autres méthodes. le RFGANM demeure robuste pour des tailles de population plus grande, avec un nombre d'évaluation de fonction plus important. Des expérimentations approfondies concernant l'effet de différents paramètres ont été réalisées : l'influence du choix de la distribution de probabilité, l'influence de la fonction de Pincus, l'influence de la taille de la population pour l'algorithme génétique, et l'influence de la taille de l'échantillon pour la formule de représentation RF. Des comparaisons avec d'autres algorithmes proposés dans la littérature ont montrés que RFNM est en général plus efficace.
D'autre part, Nous avons établi une représentation de la solution u réalisant le minimum global d'une fonctionnelle J sur un sous-ensemble S fermé, borné de l'espace de Hilbert séparable V. On a supposé que u est unique sur S, mais aucune hypothèse de convexité n'est impliquée sur S, ou sur J. La représentation suggère une méthode numérique simple pour approcher u, ce qui présente l'originalité de ne pas introduire ni une approximation en dimension finie à priori de u ni la résolution d'équations algébriques linéaires ou non. Cependant, il fait intervenir un nombre important d'évaluations de J. Cette méthode a été testée sur quelques problèmes classiques du calcul des variations, en utilisant différentes stratégies et bases et a conduit à de bons points de départ pour une amélioration ultérieure par l'optimisation avec des méthodes standards . Des améliorations par Nelder Mead ou par optimisation par essaim de particules sur un sous-espace à dimension finie ont été testées et la procédure globale a montré que l'approche est efficace.

Les principales publications liées à ces contributions sont :

• H. Zidani, R. Ellaia, and J.E. Souza de Cursi. "A Hybrid Simplex Search For Global Optimization With Representation Formula And Genetic Algorithm" , *soumis à 'Applied Mathematics and Computation'*, (ISSN : 0096 – 3003).

• H. Zidani, J.E. Souza de Cursi, and R. Ellaia. Numerical Approximation Of The Solution In Ifinite Dimensional Global Optimization Using A Representation Formula", *soumis au 'Journal of Global optimization*, (ISSN : 0925 – 5001 (print version), 1573 – 2916 (online)).

CONTRIBUTION À L'OPTIMISATION DE STRUCTURES MÉCANIQUES

Une méthodologie pour transformer un problème linéaire avec charge gaussienne en un problème multi-objectif a été proposée et appliquée à deux exemples de conception mécanique. Pour résoudre le problème d'optimisation qui en résulte, une nouvelle hybridation, l'algorithme RFNM, a été proposée. L'avantage principal de cette hybridation est la grande fiabilité et la stabilité de l'algorithme. Nous avons utilisé la méthode NBI pour transformer le problème à deux fonctions objectif en une série de problèmes mono-objectif résolus par l'algorithme RFNM afin de construire la courbe de Pareto. Les résultats ont montré que RFNM était en mesure de localiser l'optimum global avec une grande précision et sa simplicité pour l'application aux problèmes généraux de conception. Le problème du treillis à deux barres étudié a une caractéristique intéressante : en fonction de la bande de fréquence de la charge, la forme de la courbe de Pareto varie d'un seul point pour la bande de fréquence $(0, 100)$, à une courbe de Pareto discontinu pour la bande $(600, 800)$. Pour les deux exemples traités, RFNM est plus précis, fiable et efficace pour localiser l'optimum global que les autres alternatives analysées.

D'autres part, un dispositif de suspension modélisé par un système dynamique linéaire SDOF impliquant une rigidité aléatoire uniforme a été considéré. La conception du raideur est réalisée en respectant une coque de réponse d'amplitude maximale, compte tenu de la probabilité de défaillance. Les deux fonctions objectifs découlent de l'intérêt pour la réalisation d'un dispositif de suspension ayant la plus faible rigidité et le plus grand écart type. Pour générer le front de Pareto, pour ce problème convexe on a amélioré la méthode NNC pour obtenir la courbe de Pareto en utilisant notre algorithme hybride RFNM. Les résultats montrent qu'une courbe discontinue de Pareto peut être obtenue pour des valeurs spécifiques de la coque de réponse d'amplitude crête. Par conséquent, il est montré à travers cet exemple relativement simple (SDOF) que des situations difficiles peuvent se produire pour la conception des systèmes mécaniques dynamiques lors de l'examen de la coque de réponse d'amplitude, même pour cette situation simple. La stratégie proposée ici donne des résultats valables, capables d'aider les concepteurs à choisir le meilleur compromis entre la moyenne et l'écart type pour ce genre de problèmes. Une comparaison de NNC amélioré (INNC), en utilisant RFNM, avec des algorithmes d'optimisation multiobjectif connus a démontré son efficacité et son efficience.

Les principales publications en rapport avec cette contribution sont :

• H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. Multi-Objective Optimization By A New Hybridized Method : Applications To Random Mechanical Systems", *Engineering optimization*, (ISSN 0305 – 215X (Print), 1029 – 0273 (Online)), DOI : $10.1080/0305215X.2012.713355$.

• H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. Multi-Objective Optimization Of The Dynamic Of A Bridge Pillar And A Truss Structure Subjected To Random Load By A New Hybridized Method, *Mecánica Computacional*,

(ISSN 1666 – 6070), v. 30, p. 3339 – 3355, 2011.

• H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. Reliability-Based Design Of A Dynamical Suspension With Random Stiffness Under An Imposed Amplitude Response Hull, *'Mecánica Computacional*, (ISSN 1666 – 6070), Vol XXXI, p. 2765 – 2781 (2012).

Contribution à l'optimisation multi-objectif

Une nouvelle méthode dite RSMO est présentée. Elle permet de résoudre les problèmes d'optimisation multi-objectif en générant un nombre suffisant de points du front de Pareto. Une implémentation de cette méthode, pour des problèmes convexes, a été proposée. Des tests sur quelques problèmes de la litérature ont été réalisés, et ont démontré l'efficacité et l'efficience de cette approche. Des comparaisons avec des méthodes bien connues dans le domaine de l'optimisation multi-objective ont montré que RSMO est plus efficace et plus efficiente que les autres approches.
D'autre part, une amélioration de la méthode NNC pour les problèmes d'optimisation multi-objectif nonconvexes a été proposée et testée sur des problèmes de mécaniques, avec des comparaisons avec des méthodes bien connues. Les résultats ont montré que cette modification apporte une amélioration considérable de la méthode NNC.

Les principales publications liées à cette contribution sont :

• H. Zidani, R. Sampaio, R. Ellaia, E. Pagnacco and J.E. Souza de Cursi. Improved Normalized Normal Constraint Multiobjective Optimization Method for the Reliable Design of a Dynamical Suspension Device", *en cours de soumission* (2013).

• H. Zidani, R. Ellaia, and J.E. Souza de Cursi. **Representation of Solution for Multiobjective Optimization : RSMO for Generating a Suffisant Pareto Front**, *soumis au 'Computational & Applied Mathematics'*, (ISSN : 0101 – 8205 (print version), 1807 – 0302 (online)).

Perspectives

Les travaux menés dans le cadre de cette thèse ouvrent de large perspectives de recherches, notamment :

- L'utilisation d'autres formules de représentation telles que celle de Falk ou Ivanov, et faire une étude comparative entre elle,
- L'hybridation des formules de représentation avec d'autres algorithmes locaux ou globaux, tel que le PSO, colonie de fourmis, recherche tabu, ...etc,
- L'application de la formule de représentation en dimension infinie sur des problèmes de calcul des variation dans le domaine de l'ingénierie,
- Prévoir un autre scénario pour le cas de dimension infinie, avec un choix aléatoire de la base,
- L'étude théorique détaillée de la méthode RSMO,
- La mise en ouevre de RSMO pour des problèmes multi-objectif non convexe,
- La recherche d'autre algorithms d'implementation de RSMO,
- L'hybridation de RSMO avec des algorithmes évolutionnaires d'optimisation multi-objectif,

- L'implementation du parallélisme pour les formules de représentations,
- L'implementation du parallélisme pour RSMO,
- L'application de RSMO pour des problèmes de l'ingénierie.

BIBLIOGRAPHIE

Avriel Mordecai. *Nonlinear programming : analysis and methods*. Courier Dover Publications, 2003. (Cité page 17.)

Bérubé Jean-François, Gendreau Michel, et Potvin Jean-Yves. An exact ε-constraint method for bi-objective combinatorial optimization problems : Application to the traveling salesman problem with profits. *European Journal of Operational Research*, 194 (1) :39–50, 2009. (Cité page 23.)

Bez Edson Tadeu, Gonçalves Mirian Buss, et Souza de Cursi José Eduardo. A procedure of global optimization and its application to estimate parameters in interaction spatial models. *International Journal for Simulation and Multidisciplinary Design Optimization*, 4(2) :85–100, 2010. (Cité page 38.)

Bez Edson Tadeu, Souza de Cursi JE, et Gonçalves Mirian Buss. A hybrid method for continuous global optimization involving the representation of the solution. Dans *6th World Congress on Structural and Multidisciplinary Optimization–WCSMO6*, 2005. (Cité pages 38 et 87.)

Bez Souza de Cursi J.E., E.T. et Gon¸calves M.B. Um procedimento numérico para a otimização global baseado em uma representação da solução. *Tema - Tend. Mat. Apl. Comput.*, 5(2) :185–194, 2004. (Cité page 38.)

Bianchi Leonora, Dorigo Marco, Gambardella Luca Maria, et Gutjahr Walter J. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing : an international journal*, 8(2) :239–287, 2009. (Cité page 12.)

Birge John R et Louveaux Francois. *Introduction to stochastic programming*. Springer Verlag, 1997. (Cité page 11.)

Bossy Mireille et Talay Denis. A stochastic particle method for the mckean-vlasov and the burgers equation. *Mathematics of Computation of the American Mathematical Society*, 66(217) :157–192, 1997. (Cité page 87.)

Černỳ VLADIMÍR. Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1) : 41–51, 1985. (Cité page 13.)

Charnes A. et Wolfe M. Extended pincus theorems and convergence of simulated annealing. *International journal of systems science*, 20(8) :1521–1533, 1989. (Cité page 38.)

Chelouah R. et Siarry P. Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. *European Journal of Operational Research*, 148(2) :335–348, 2003. (Cité pages 37, 51 et 54.)

Chelouah Rachid et Siarry Patrick. A hybrid method combining continuous tabu search and nelder–mead simplex algorithms for the global optimization of multiminima functions. *European Journal of Operational Research*, 161(3) :636–654, 2005. (Cité pages 37 et 54.)

Chong Edwin K. P. et Zak Stanislaw H. *Engineering optimization : theory and practice*. Wiley-lnterscience Publication, 2011. (Cité page 8.)

Clerc Maurice. *Particle swarm optimization*, volume 243. Iste London, 2006. (Cité page 16.)

Coello Carlos A Coello, Lamont Gary B, et Veldhuizen David A Van. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007. (Cité pages 26 et 28.)

Colorni Alberto, Dorigo Marco, Maniezzo Vittorio, et others . Distributed optimization by ant colonies. Dans *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142. Paris, France, 1991. (Cité page 15.)

Da Fonseca Carlos Manuel Mira. *Multiobjective genetic algorithms with application to control engineering problems*. Citeseer, 1995. (Cité page 27.)

Das I. et Dennis J.E. Normal-boundary intersection : a new method for generating pareto optimal point in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3) :631–657, 1998. (Cité page 112.)

Deb Kalyanmoy. Multi-objective genetic algorithms : Problem difficulties and construction of test problems. *Evolutionary computation*, 7(3) :205–230, 1999. (Cité pages viii, 86 et 113.)

Deb Kalyanmoy, Agrawal Samir, Pratap Amrit, et Meyarivan Tanaka. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization : Nsga-ii. *Lecture notes in computer science*, 1917 :849–858, 2000. (Cité page 30.)

Deb Kalyanmoy, Pratap Amrit, Agarwal Sameer, et Meyarivan TAMT. A fast and elitist multiobjective genetic algorithm : Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2) :182–197, 2002. (Cité page 30.)

Deep K., M. Thakur. A new mutation operator for real coded genetic algorithms. *Applied Mathematics and Computation*, 193 :211–230, 2007. (Cité page 45.)

Dennis John E et Schnabel Robert B. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16. Society for Industrial and Applied Mathematics, 1987. (Cité page 9.)

Dréo Johann, Petrowski Alain, Siarry Patrick, et Taillard Eric. *Metaheuristics for hard optimization*. Springer, 2006. (Cité pages 14 et 15.)

Ehrgott M. et Wiecek MM. *Mutiobjective Programming, In Multiple Criteria Decision Analisys : State of the art surveys*, volume 78. Volume 78, Springer New York, 2005. (Cité page 105.)

Ehrgott Matthias et Gandibleux Xavier. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum*, 22(4) :425–460, 2000. (Cité page 26.)

Falk J.E. Condition for global optimization in nonlinear programming. *Operations Research*, 21 :337–340, 1973. (Cité page 38.)

Fan Shu-Kai S, Liang Yun-chia, et Zahara Erwie. Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. *Engineering Optimization*, 36(4) :401–418, 2004. (Cité page 37.)

Fan Shu-Kai S, Liang Yun-Chia, et Zahara Erwie. A genetic algorithm and a particle swarm optimizer hybridized with nelder–mead simplex search. *Computers & Industrial Engineering*, 50(4) :401–425, 2006. (Cité page 37.)

Fan Shu-Kai S et Zahara Erwie. A hybrid simplex search and particle swarm optimization for unconstrained optimization. *European Journal of Operational Research*, 181(2) : 527–548, 2007. (Cité page 37.)

Felippa Carlos A. Introduction to finite element methods. *University of Colorado, Boulder, http ://www. colorado. edu/engineering/CAS/courses. d/IFEM. d*, 2004. (Cité page 62.)

Floudas , Christodoulos , Pardalos , et Panos . *Encyclopedia of Optimization*. Springer, 2009. (Cité page 37.)

Floudas C.A. et Gounaris C.E. A review of recent advances in global optimization. *Journal of Global Optimization*, 2008. (Cité page 37.)

Floudas C.A. et al. *Handbook of Test Problems in Local and Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1999. (Cité page 45.)

Fonseca Carlos M, Fleming Peter J, et others . Genetic algorithms for multiobjective optimization : Formulation, discussion and generalization. Dans *Proceedings of the fifth international conference on genetic algorithms*, volume 423, pages 416–423. San Mateo, California, 1993. (Cité pages 27 et 28.)

Fourman Michael P. Compaction of symbolic layout using genetic algorithms. Dans *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 141–153. L. Erlbaum Associates Inc., 1985. (Cité page 26.)

Gaviano Marco, Kvasov Dmitri E, Lera Daniela, et Sergeyev Yaroslav D. Algorithm 829 : Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Transactions on Mathematical Software (TOMS)*, 29(4) :469–480, 2003. (Cité page 45.)

Gendreau Michel et Centre for Research on Transportation (Montréal Québec). *An introduction to tabu search*. Springer, 2002. (Cité page 14.)

Georgievaa A. et Jordanov I. A hybridmeta-heuristic for global optimisation using low-discrepancy sequences of points. *Computers & Operations Research*, 37(3) :456–469, 2010. (Cité page 54.)

Géradin M. et Rixen D. *Théorie des Vibrations - Applications à la Dynamique des Structures*. Ed. Masson, 1996. (Cité page 58.)

Giannakoglou K.C. et de Métodos Numéricos en Ingeniería Centro Internacional. *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems : Proceedings of the EUROGEN2001Conference, Athens, Greece, September 19-21, 2001*. Theory and engineering applications of computational methods. International Center for Numerical Methods in Engineering (CIMNE), 2002.

ISBN 9788489925977. URL http://books.google.co.ma/books?id=ctr4AAAACAAJ. (Cité page 28.)

Gillespie T.D. *Fundamentals of vehicle dynamics*. Society of Automotive Engineers, 1992. (Cité page 59.)

Glover Fred. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5) :533–549, 1986. (Cité page 14.)

Goldberg David E. Genetic algorithms in search, optimization, and machine learning. 1989. (Cité pages 26 et 27.)

Goldberg D.E. *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Wesley, New York, 1998. (Cité page 12.)

Gould Nicholas. An introduction to algorithms for continuous optimization, 2006. (Cité page 105.)

Haupt R.L. et Haupt S.E. *Practical Genetic Algorithms*. Wiley/Interscience, New York, 1998. (Cité page 12.)

Horn Jeffrey, Nafpliotis Nicholas, et Goldberg David E. A niched pareto genetic algorithm for multiobjective optimization. Dans *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 82–87. IEEE, 1994. (Cité page 27.)

Horst R., Pardalos P. M., et Romeijn H. E. Handbook of global optimization. *Springer Publisher, Kluwer*, 1995. (Cité page 45.)

Huang Z., Miao X., et Wang P. A revised cut-peak function method for box constrained continuous global optimization. *Applied Mathematics and Computation*, 194 :224–233, 2007. (Cité page 45.)

Ivanov Girlin S.K. & Liudvichenko, V.V. Problems and results of global search in case of continuous functions. Dans *In Fiodorov (ed.) Models and Methods of Global Optimization, Vol 13*, 1985. (Cité page 39.)

Ivorra B., Mohammadi B., Ramos A.M., et Redont P. Optimizing initial guesses to improve global minimization. *Journal Of Global Optimization*, 2011. (Cité page 37.)

Janikow Cezary Z et Michalewicz Zbigniew. An experimental comparison of binary and floating point representations in genetic algorithms. Dans *Proceedings of the fourth international conference on genetic algorithms*, volume 31, page 36. San Diego, CA, 1991. (Cité page 12.)

Kao Y., E. Zahara. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8 :849–857, 2008. (Cité page 45.)

Kapur J.N. et Kesavan H. K. *Entropy optimization principles with applications*. London, 1992. (Cité page 73.)

Kelley C.T. *Iterative Methods for Linear and Nonlinear Equations*. SIAM Publication, Philadelphia, 1995. (Cité page 9.)

Kelley C.T. *Iterative Methods for Optimization*. SIAM Publication, Philadelphia, 1999. (Cité page 9.)

Kelley C.T. *Solving Nonlinear Equations with Newton's Method*. SIAM Publication, Philadelphia, 2003. (Cité page 9.)

Kirkpatrick Scott, Jr. D. Gelatt, et Vecchi Mario P. Optimization by simulated annealing. *science*, 220(4598) :671–680, 1983. (Cité page 13.)

Knowles Joshua D et Corne David W. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2) :149–172, 2000. (Cité page 29.)

Kolda Tamara G, Lewis Robert Michael, et Torczon Virginia. Optimization by direct search : New perspectives on some classical and modern methods. *SIAM review*, 45 (3) :385–482, 2003. (Cité page 9.)

Koscianski A et de Cursi JE Souza. Physically constrained neural networks and regularization of inverse problems. Dans *Proceedings of the 6th. World Congress on Structural and Multidisciplinary Optimisation, Rio de Janeiro, Brazil*, 2005. (Cité page 87.)

Lee K. S., Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization : harmony search theory and practice. *Comput. Methods Appl. Mech. Engrg.*, 194 :3902–3933, 2005. (Cité page 45.)

Lee Kwang Y et El-Sharkawi Mohamed A. *Modern heuristic optimization techniques : theory and applications to power systems*, volume 39. Wiley-IEEE press, 2008. (Cité page 37.)

Liberti Leo et Maculan Nelson. *Global Optimization : From Theory to Implementation*, volume 84. Springer Science+ Business Media, 2006. (Cité page 45.)

Lin Y.K. *Probabilistic theory of structural dynamics*. McGraw-Hill, Inc., N.Y., 1967. (Cité pages 59 et 72.)

Lu Haiming et Yen Gary G. Rank-density-based multiobjective genetic algorithm and benchmark test function study. *Evolutionary Computation, IEEE Transactions on*, 7 (4) :325–343, 2003. (Cité page 113.)

Luenberger D.G. *Introduction to Linear and Nonlinear Programming*. Stanford, 1973. (Cité page 17.)

Marler R Timothy et Arora Jasbir S. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6) :369–395, 2004. (Cité page 23.)

Messac Achille, Ismail-Yahaya Amir, et Mattson Christopher A. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2) :86–98, 2003. (Cité pages 34 et 75.)

Metropolis Nicholas, Rosenbluth Arianna W, Rosenbluth Marshall N, Teller Augusta H, et Teller Edward. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21 :1087, 1953. (Cité page 13.)

Milshtein GN. The solving of boundary value problems by numerical integration of stochastic equations. *Mathematics and computers in simulation*, 38(1) :77–85, 1995. (Cité page 87.)

Morillon Jean-Paul. Numerical solutions of linear mixed boundary value problems using stochastic representations. *International journal for numerical methods in engineering*, 40(3) :387–405, 1997. (Cité page 87.)

Morse Joel N. Reducing the size of the nondominated set : Pruning by clustering. *Computers & Operations Research*, 7(1) :55–66, 1980. (Cité page 28.)

Nelder J.A. et Mead R. A simplex method for function minimization. *Computer Journal*, 7 :308–313, 1965. (Cité page 9.)

Pagnacco E., Sampaio R., et Souza de Cursi E. Frequency response functions of random linear mechanical systems and propagation of uncertainties. *Mecanica Computacional*, XXX :3357–3380, 2011. (Cité page 74.)

Pedregal Pablo. *Introduction to optimization*, volume 46. Springer, 2003. (Cité pages 100 et 102.)

Pincus M. A closed formula solution of certain programming problems. *Operations Research*, 16(3) :690–694, 1968. (Cité pages 38, 39, 42, 43, 86 et 87.)

Pincus M. A monte carlo method for the approximate solution of certain types of constrained optimization problems. *Operation Research*, 18 :1225, 1970. (Cité page 38.)

Powell MJD. Direct search algorithms for optimization calculations. *Acta Numerica*, pages 287–336, 1998. (Cité page 9.)

Prekopa A. Stochastic programming. *Kluwer Academic Publishers, Dordrecht*, 1995. (Cité page 11.)

Price K., Storn R., et Lampinen J. *Differential evolutionary practical approach to global optimisation*. Springer, Berlin, Germany, 2005. (Cité page 54.)

Rao Singiresu S. *Engineering optimization : theory and practice*. Wiley, 2009. (Cité pages 6 et 7.)

Renders J. M. et Flasse S. P. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, 26(2) :243–258, 1996. (Cité page 37.)

rey Horn Je, Nafpliotis Nicholas, et Goldberg David E. Multiobjective optimization using the niched pareto genetic algorithm. *IlliGAL report*, (93005) :61801–2296, 1993. (Cité page 27.)

Rosenthal Richard E. Concepts, theory, and techniques principles of multiobjective optimization*. *Decision Sciences*, 16(2) :133–152, 1985. (Cité page 24.)

Rubinstein R. Y. et Kroese D. P. *Simulation and the Monte Carlo method*. New Jersey, USA, 2008. (Cité page 73.)

Schaffer J David. Multiple objective optimization with vector evaluated genetic algorithms. Dans *Proceedings of the 1st international Conference on Genetic Algorithms*, pages 93–100. L. Erlbaum Associates Inc., 1985. (Cité page 25.)

Schwartz L. Extension du théorème de sazonov-minlos. Dans *Séminaire de Probabilités III Université de Strasbourg*, pages 1–23. Springer, 1969. (Cité page 89.)

Shelokar P. S., Siarry P., Jayaraman V.K., et Kulkarni B.D. Particle swarm and ant colony algorithms hybridized for improved continuous optimization. *Applied Mathematics and Computation*, 188 :129–142, 2007. (Cité page 45.)

Shukla Pradyumn Kumar et Deb Kalyanmoy. On finding multiple pareto-optimal solutions using classical and evolutionary generating methods. *European Journal of Operational Research*, 181(3) :1630–1652, 2007. (Cité page 76.)

Souza de Cursi J, Tarocco E, de Souza Neto EA, et Novotny AA. Representation of solutions in variational calculus. *E. Tarocco*, 2007. (Cité pages 38 et 44.)

Souza de Cursi J. E. A feynman-kac method for the determination of the stefan's free boundary. Dans *Proceedings of the International Congress on Inverse Problems in Engineering*. Angra dos Reis, Brazil, 2002. (Cité page 87.)

Souza de Cursi J.Ẽ. A numerical method for the representation of the solution. *TEMA*, 5 (2) :185–194, 2005. (Cité page 87.)

Souza de Cursi JE. Numerical methods for linear boundary value problems based on feyman–kac representations. *Mathematics and computers in simulation*, 36(1) :1–16, 1994. (Cité page 87.)

Souza de Cursi JE. Representation and numerical determination of the global optimizer of a continuous function on a bounded domain. *Nonconvex Optimization and its Applications*, 74 :517–540, 2003. (Cité page 87.)

Souza de Cursi J.E. et El Hami A. Representation of solutions in continuous optimization. *to appear in International Journal for Simulation and Multidisciplinary Design Optimization*, 2011. (Cité page 38.)

Spendley W, Hext GR, et Himsworth FR. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4) :441–461, 1962. (Cité page 9.)

Srinivas Nidamarthi et Deb Kalyanmoy. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3) :221–248, 1994. (Cité page 30.)

Szu Harold et Hartley Ralph. Fast simulated annealing. *Physics letters A*, 122(3) :157–162, 1987. (Cité page 14.)

Talbi El-ghazali. Métaheuristiques pour l'optimisation combinatoire multi-objectifs : Etat de l'art. *Rapport CNET (France Telecom) Octobre*, 1999. (Cité page 24.)

Tiel Jan van. Convex analysis : an introductory text. *Royal Neterland Meteorological Institute*, 1984. (Cité page 90.)

Utyuzhnikov SV, Fantini Paolo, et Guenov Marin D. A method for generating a well-distributed pareto set in nonlinear multiobjective optimization. *Journal of Computational and Applied Mathematics*, 223(2) :820–841, 2009. (Cité page 76.)

Van Veldhuizen David A. Multiobjective evolutionary algorithms : classifications, analyses, and new innovations. Rapport technique, DTIC Document, 1999. (Cité pages 30 et 105.)

Wallace Stein W et Kall P. Stochastic programming, 1995. (Cité page 11.)

Wang L., Chen K., et Ong Y.S. Advances in natural computation. *Part III, Springer Science & Business Publisher, Changsha, China*, 2005. (Cité page 45.)

Weile Daniel S, Michielssen Eric, et Goldberg David E. Multiobjective synthesis of electromagnetic devices using nondominated sorting genetic algorithms. Dans *Antennas and Propagation Society International Symposium, 1996. AP-S. Digest*, volume 1, pages 592–595. IEEE, 1996. (Cité page 30.)

Weise T. et al. Why is optimization difficult ? Dans *Nature-Inspired Algorithms for Optimisation*, pages 1–50, 2009. (Cité page 37.)

Wijker Jaap. *Random Vibrations in Spacecraft Structures Design : Theory and Applications*, volume 165. Springer, 2009. (Cité page 59.)

Wright Margaret H. Direct search methods : Once scorned, now respectable. *Pitman Research Notes In Mathematics Series*, pages 191–208, 1996. (Cité page 9.)

Xinchao Z. A perturbed particle swarm algorithm for numerical optimization. *Applied Soft Computing*, 10 :119–124, 2010. (Cité page 45.)

Yen J. et al. A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, 28(2) :173–191, 1998. (Cité page 37.)

Yuan Q., He Z., et Leng H. A hybrid genetic algorithm for a class of global optimization problems with box constraints. *Applied Mathematics and Computation*, 197 :924–929, 2008. (Cité page 45.)

Zhang WH et Gao T. A min–max method with adaptive weightings for uniformly spaced pareto optimum points. *Computers & structures*, 84(28) :1760–1769, 2006. (Cité page 110.)

Zheng R. et al. A staged continuous tabu search algorithm anor the global optimisation and its applications to the design of fibre bragg gratings. *Computational Optimisation and Applications*, 30(3) :319–335, 2005. (Cité page 54.)

Zitzler Eckart, Laumanns Marco, Thiele Lothar, Zitzler Eckart, Zitzler Eckart, Thiele Lothar, et Thiele Lothar. Spea2 : Improving the strength pareto evolutionary algorithm, 2001. (Cité page 28.)

Zitzler Eckart et Thiele Lothar. Multiobjective evolutionary algorithms : A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4) :257–271, 1999. (Cité page 28.)

Zowe J. Nondifferentiable optimization. *Computational Mathematical Programming*, 15 :323–355, 1985. (Cité page 48.)

Zwillinger D. et Kokoska S. *Standard probability and statistics tables and formulae*. New York, USA, 2000. (Cité page 74.)

ANNEXE

FONCTIONS TESTS MATHÉMATIQUES

1. Fonction de Cosine mixture

$$f(x) = \sum_{i=1}^{d} x_i^2 - 0.1 \sum_{i=1}^{d} cos(5\pi x_i)$$

avec

$$-1 \leq x_i \leq 1$$

Cette fonction possède un minimum $f(x) = -0.1\ d$ en $x = (0,0,...,0)^T$. Avec d les dimensions du probblème

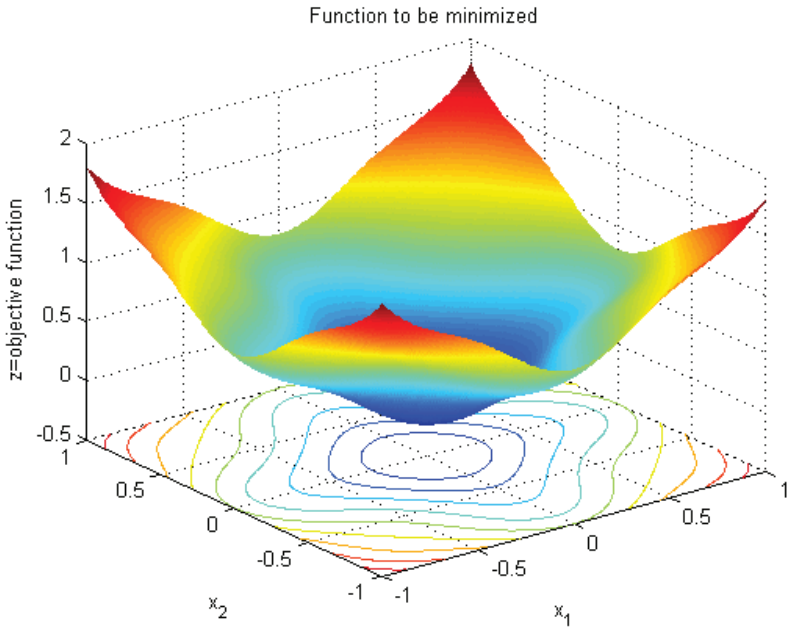


FIGURE 15 – *Représentation graphique de Cosine mixture pour d=2*

2. Fonction de Rosenbrock

$$f(x) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$$

avec

$$-30 \leq x_i \leq 30$$

Cette fonction possède un minimum $f(x) = 0$ en $x = (1,1,...,1)^T$. La difficulté pour la recherche de ce point est liée au fait qu'il se trouve à l'intersection de deux vallées de pentes très faibles (voir figure 16).

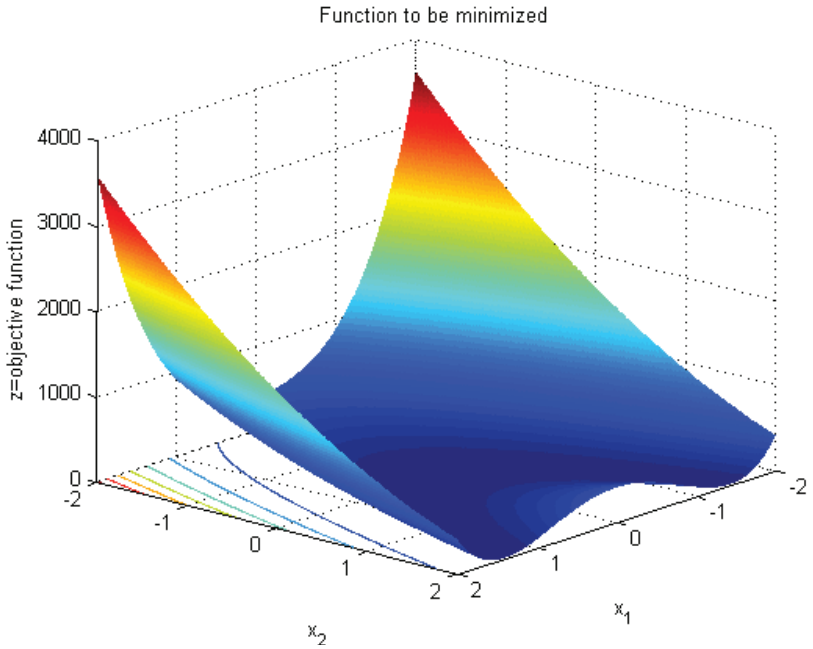


FIGURE 16 – *Représentation graphique de Rosenbrock pour d=2*

3. Fonction de Goldstein and Price

$$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30+$$
$$(2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

avec

$$-5 \leq x_i \leq 5$$

Cette fonction possède un minimum global $f(x) = 3$ en $x = (0, -1)^T$.

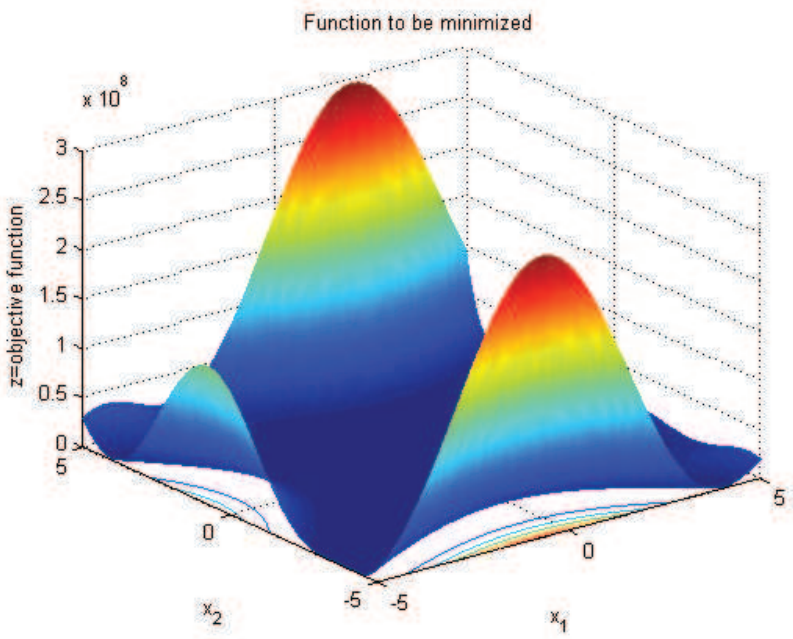


FIGURE 17 – *Représentation graphique de Goldstein and Price pour d=2*

4. Fonction de Camel

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

avec

$$-10 \leq x_i \leq 10$$

Cette fonction possède un minimum $f = -1.0316285$ en $x = (\pm 0.089842, \mp 0.712656)^T$. Les deux figures 18 et 19 représentent le graphe ainsi que le contours de la fonction de camel

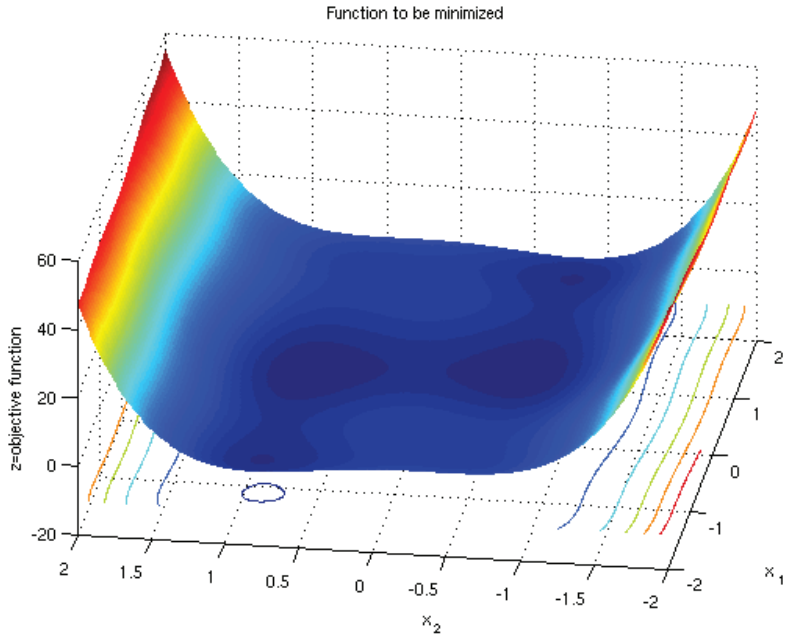


FIGURE 18 – *Représentation graphique de Camel pour d=2*

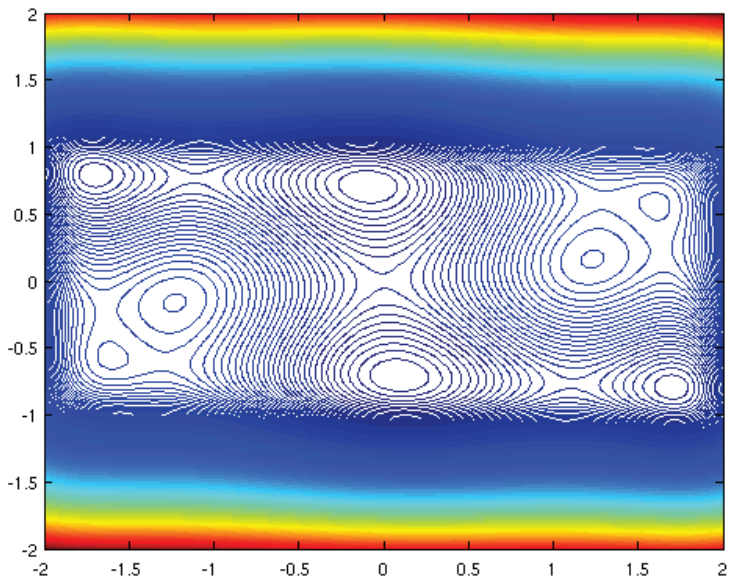


FIGURE 19 – *Représentation graphique de Camel pour d=2*

5. Fonction de Shubert

$$f(x) = \prod_{i=1}^{d} \sum_{j=1}^{5} j cos[(j+1)x_i + j]$$

avec

$$-10 \leq x_i \leq 10$$

Cette fonction possède un minimum global $f = -186.730909$ en $x = (-1.42513, -0.80032)^T$ (voir figure 20 pour d=2).

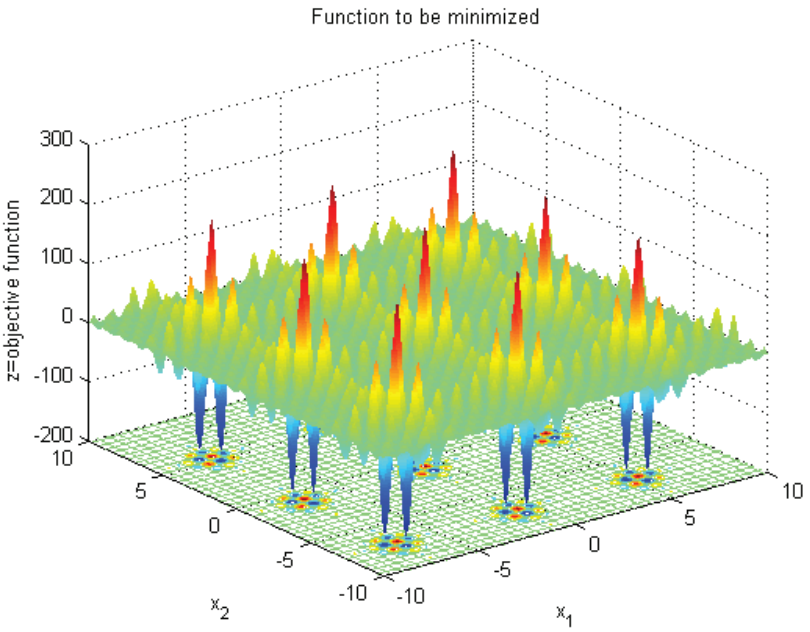


FIGURE 20 – *Représentation graphique de Shubert pour d=2*

6. Fonction de Shekel

$$f_{SK5}(x) = -\sum_{i=1}^{m} \frac{1}{(x-a_i)^T(x-a_i)+c_i} \quad x \in \mathbb{R}^d \ \ et \ \ m = 5$$

$$f_{SK7}(x) = -\sum_{i=1}^{m} \frac{1}{(x-a_i)^T(x-a_i)+c_i} \quad x \in \mathbb{R}^d \ \ et \ \ m = 7$$

$$f(x)_{SK10} = -\sum_{i=1}^{m} \frac{1}{(x-a_i)^T(x-a_i)+c_i} \quad x \in \mathbb{R}^d \ \ et \ \ m = 10$$

Les coefficients de la fonction de Shekel a_{ij} , c_i sont :

i/j	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.4
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

TABLE 3 – *four-dimensional Shekel function*

Quelques informations sur la fonction de Shekel sont données par la table 4

Fonction	l'optimum global	$f_{théorique}$	domaine de recherche
$f_{SK5}(x)$	$(4,4,4,4)$	-10.1532	$[0,10]^{d=4}$
$f_{SK7}(x)$	$(4,4,4,4)$	-10.40294	$[0,10]^{d=4}$
$f_{SK10}(x)$	$(4,4,4,4)$	-10.53641	$[0,10]^{d=4}$

TABLE 4 – *informations basiques sur la fonction Shekel*

7. Fonction de Rastrigin

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

avec

$$-5.12 \leq x_i \leq 5.12$$

Cette fonction possède un minimum global $f = 0$ en $x = (0,0)^T$.

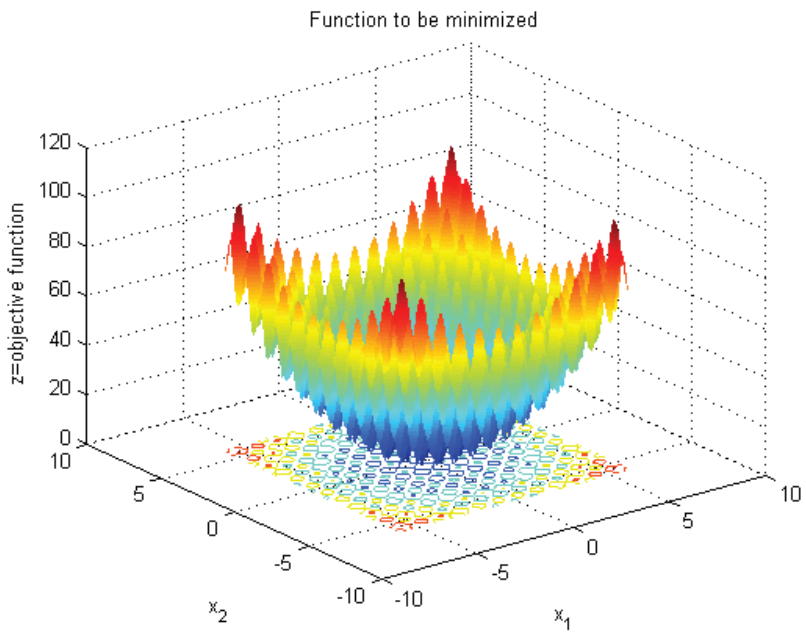


FIGURE 21 – *Représentation graphique de Rastrigin pour d=2*

8. Fonction de Griewank

$$f(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

avec

$$-600 \leq x_i \leq 600$$

Cette fonction possède un minimum global $f = 0$ en $x = (0,0)^T$.

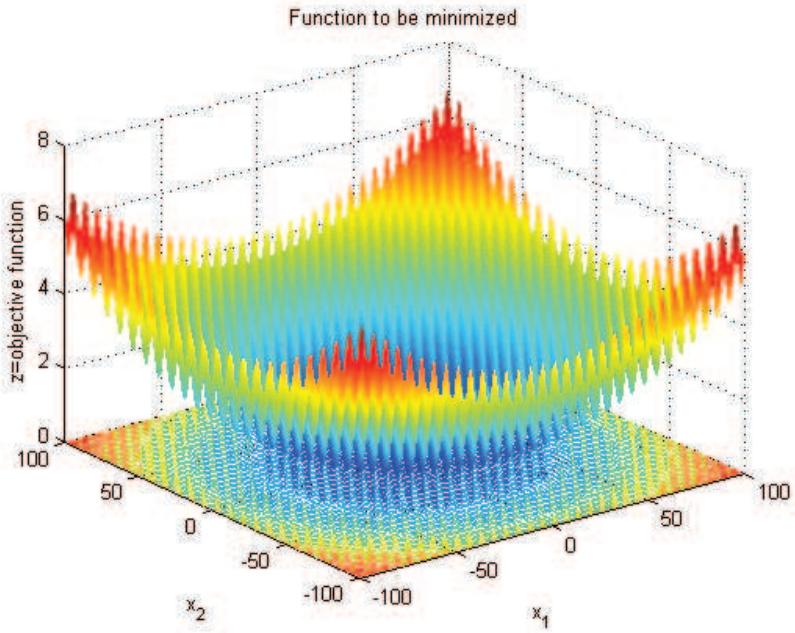


FIGURE 22 – *Représentation graphique de Griewank pour d=2*