UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis

Mention : Automatique, Traitement du Signal et des Images

présentée et soutenue par

Hoël LANGOUËT

OPTIMISATION SANS DÉRIVÉES SOUS CONTRAINTES

deux applications industrielles en ingénierie de réservoir et en calibration des moteurs

Thèse dirigée par Éric THIERRY

soutenue le 28 Juin 2011

Jury :

M. L. DUMAS	Professeur, UVSQ	Examinateur
M. S. GRATTON	Professeur, CERFACS	Rapporteur
M. M. MASMOUDI	Professeur, CNRS Toulouse	Rapporteur
Mme. A. SARTENAER	Professeur, FUNDP, Belgique	Examinateur
M. T. SCHAAF	Docteur, Ingénieur Réservoir, GDF SUEZ E&P	Examinateur
Mme. D. SINOQUET	Docteur, Ingénieur de Recherche IFPEN	Encadrant
M. É. THIERRY	CNRS UNSA	Directeur de thèse

« Commence par faire le nécessaire, puis fais ce qu'il est possible de faire et tu réaliseras l'impossible sans t'en apercevoir. »

François d'Assise

Remerciements

Au terme de ce travail, je voudrais remercier les personnes qui, de près ou de loin, ont contribué à sa réalisation.

J'adresse un grand Merci à Delphine Sinoquet, de IFP Energies nouvelles pour son encadrement, ses qualités humaines et bien plus encore ; en particulier pour tous ces moments inoubliables passés ensemble, la marguarita du lac Michigan, la copa de cava de la Sagrada Familia et bien sûr les pisco Sour du désert d'Atacama...

Je remercie Eric Thierry, mon directeur de thèse, d'avoir permis le bon déroulement de celle-ci.

Je remercie Serge Gratton et Mohamed Masmoudi, qui m'ont fait l'honneur d'être les rapporteurs de mes travaux de thèse, ainsi qu'Annick Sartenaer, Laurent Dumas et Thomas Schaaf pour leur participation au jury. Je les remercie d'avoir consacré du temps à la lecture de mon manuscrit.

Je remercie Fréderic Delbos, mon co-bureau pour l'ambiance qui régnait dans le bureau IS/112 et pour m'avoir initié au Tennis-calendar. Je le remercie tout autant d'avoir répondu à toutes les questions que je me posais et reposais sur l'optimisation.

L'application en ingénierie de réservoir traitée n'aurait pu l'être sans la réactivité et l'aide précieuse apportées par Sébastien Da Veiga.

Je remercie également tout le département des moteurs pour les explications en calibration que j'ai reçues par l'intermédiaire de Delphine.

Je remercie Raphaël Gayno mon premier co-bureau de m'avoir accompagné dans mes premiers mois de thèse et pour tous les bugs corrigés ensemble.

Je remercie Quang Huy Tran de m'avoir fait réfléchir à l'éventualité de faire une thèse et de m'avoir donné goût à la recherche au travers d'un stage intéressant à IFP Energies nouvelles.

Je remercie les départements de mathématiques appliquées et informatique, et tout particulièrement ses doctorants et stagiaires pour tous ces longs repas, ces sorties (resto, ski, café Leffe, barbecue, Porquerolles, merci de m'avoir légué "justice" alors qu'elle était déjà morte...), et pour tous les délires improvisés ou non... Merci à Elodie, Greg, Long, Philippe, Laura, Thibaut, Jeff, Lhoussine, Samuel, Christophe, Mikolaj, Longmin, Ivan, Cindy, Florian, Xavier, Aurélie, Mohamed, Eugenio, Carole, Nathaliya, les Thomas, Sebastien, Antoine, Simon, Florent et Mathieu. Je remercie également les secrétaires de R11

pour leur gentillesse et leur aide.

Je voudrais également remercier tous les doctorants de l'IFP avec lesquels j'ai parcouru le monde entier, aussi bien pour des formations doctorales que pour des congrès ou des voyages ADIFP aux US et en Russie : Céline, Samir, Marie, Nadège, Zyed, Amine, Monsef, Laure, Thomas, Vincent, Marc, Arnaud, Régis, Lun, Frank, Simone, Marius...

J'ai une pensée particulière pour tout le département réservoir, notamment ses doctorants, pour leur accueil très sympathique et pour m'avoir permis d'approfondir mes connaissances en réservoir.

Je remercie toute mon équipe de Water polo du SCNCR de m'avoir sorti l'optimisation de la tête tous les soirs de la semaine à l'entraînement et pour leur amitié.

Je remercie mes amis de toujours Cyril, Jo, Ana, Glo et Marie.

Je terminerai en exprimant ma plus sincère reconnaissance à mes parents, mes petites sœurs, Énora et Maéva, et à mes grands parents pour tout le soutien qu'ils m'ont apporté.

Enfin, je souhaite remercier Louloute pour ses emails d'amour "perso". Promis je ne dirai rien.

Résumé : L'optimisation intervient dans de nombreuses applications IFPEN, notamment dans l'estimation de paramètres de modèles numériques à partir de données en géosciences ou en calibration des moteurs. Dans ces applications, on cherche à minimiser une fonction complexe, coûteuse à estimer, et dont les dérivées ne sont pas toujours disponibles. A ces difficultés s'ajoutent la prise en compte de contraintes non linéaires et parfois l'aspect multi-objectifs. Au cours de cette thèse, nous avons développé la méthode SQA (Sequential Quadradic Approximation), une extension de la méthode d'optimisation sans dérivées de M.J.D. Powell pour la prise en compte de contraintes à dérivées connues ou non. Cette méthode est basée sur la résolution de problèmes d'optimisation simplifiés basés sur des modèles quadratiques interpolant la fonction et les contraintes sans dérivées, construits à partir d'un nombre limité d'évaluations de celles-ci. Si la résolution de ce sous-problème ne permet pas une progression pour l'optimisation originale, de nouvelles simulations sont réalisées pour tenter d'améliorer les modèles. Les résultats de SQA sur différents benchmarks montrent son efficacité pour l'optimisation sans dérivées sous contraintes. Enfin, SQA a été appliqué avec succès à deux applications industrielles en ingénierie de réservoir et en calibration des moteurs. Une autre problématique majeure en optimisation étudiée dans cette thèse est la minimisation multi-objectifs sous contraintes. La méthode évolutionnaire Multi-Objective Covariance Matrix Adaptation, adaptée à la prise en compte des contraintes, s'est révélée très performante dans l'obtention de compromis pour la calibration des moteurs.

Mots clés : optimisation sans dérivées, optimisation sous contraintes, méthode de région de confiance, optimisation multi-objectifs, algorithme évolutionnaire, caractérisation de réservoir, calage historique de données sismiques et de production, calibration moteur.

Abstract : Optimization takes place in many IFPEN applications : for instance estimation of parameters of numerical models from experimental data in geosciences or for engine calibration. These optimization problems consist in minimizing a complex function, expensive to estimate, and for which derivatives are often not available. Moreover, additional difficulties arise with the introduction of nonlinear constraints and even several objectives to be minimized. In this thesis, we developed the SQA method (Sequential Quadratic Approximation), an extension of a derivative-free optimization method proposed by M.J.D Powell for optimization with constraints with known or unknown derivatives. This method consists in solving optimization sub-problems based on local quadratic interpolation models of objective function and derivative-free constraints built with a limited number of function evaluations. If the solution of this sub-problem does not progress toward a solution of the original problem, new simulations are performed to try to improve the model quality. Numerical results on benchmarks show that SQA is an effective method for constrained derivative-free optimization. Finally, SQA has been tested with success on two industrial applications in reservoir engineering and in engine calibration. An other problem studied in this thesis is multi-objective minimization under constraints. The MO-CMA-ES method (Multi-Objective - Covariance Matrix Adaptation - Evolution Strategy) adapted to take into account constraints has been successful to determine different compromises for an engine calibration application.

Keywords : Derivative-free optimization, constrained optimization, trust region method, multi-objvective optimization, evolutionary algorithm, reservoir characterization, history matching, engine calibration.

Table des matières

Remerciements **3**

Résumé **5**

Abstract **6**

Table des matières **11**

Introduction **19**

I Optimisation sans dérivées **22**

1 Etat de l'art **23**

1 Recherche directe 24
1.1 Méthode du simplexe de Nelder-Mead 24
1.2 Recherche directe directionnelle 26
1.2.1 Pattern Search 26
1.2.2 Méthode de recherche multi-directionnelle (MDS) 28
1.2.3 Méthodes de directions conjuguées 28
2 Métaheuristiques ou méthodes d'échantillonnage 29
2.1 Algorithme évolutionnaire 29
2.2 Recuit simulé 31
2.3 Autres métaheuristiques 31
3 Optimisation à l'aide de modèles de substitution 31
3.1 Modèles des gradients de la fonction objectif 32
3.1.1 Approximations des gradients pour les méthodes classiques 32
3.1.2 Méthode de recherche linéaire basée sur les dérivées provenant d'un simplexe 33
3.2 Modèles de la fonction objectif 35
3.2.1 Optimisation à l'aide de surfaces de réponses globales . . . 37

3.2.2 Approche par région de confiance 40
4 Méthodes hybrides 41
5 Discussion 42

2 Optimisation sans dérivées : méthodes de région de confiance 45
1 Optimisation avec dérivées dans une région de confiance 47
2 Modèles de la fonction objectif 48
2.1 Modèle linéaire 49
2.2 Modèle quadratique 50
2.2.1 Modèle quadratique complet basé sur les polynômes de Lagrange 51
2.2.2 Polynômes fondamentaux de Newton 53
2.2.3 Modèle quadratique sous déterminé 54
2.3 Modèles à base de "Radial Basis Functions" cubiques 58
3 Mise à jour de l'échantillon unisolvant 60
4 Méthode d'optimisation sans dérivées basée sur la méthode de région de confiance 62

3 Sequential Quadratic Approximation (SQA) 66
1 Principe général et prise en compte des contraintes 67
2 Description de la méthode SQA 68
2.1 Modèle d'interpolation sous déterminé 70
2.2 Description de l'algorithme 71
2.2.1 Initialisation 72
2.2.2 Minimisation du modèle 74
2.2.3 Amélioration du modèle 78
2.2.4 Evolution de ρ 80
2.2.5 Paramètres par défaut 81
2.3 Convergence de la méthode sans contraintes 82
3 Evolution des paramètres de l'algorithme sur un cas simple 82
3.1 Exemple sous contrainte avec dérivées 84
3.2 Exemple sous contrainte sans dérivées 87
4 Améliorations de la méthode et perspectives 92
4.1 Prise en compte de contraintes d'égalité 92
4.2 Prise en compte du bruit 92
4.3 Fonction objectif de type moindres carrés 92
4.4 Mise à jour des poids de pénalisation dans la fonction de mérite . . 98

4 Résultats 100
1 Benchmarks 100
1.1 Benchmark avec contraintes de bornes (Moré et Wild) 101
1.1.1 Problèmes lisses 103
1.1.2 Problèmes bruités 107

1.1.3 Problèmes non différentiables 110
1.1.4 Conclusions 113
1.2 Benchmark CUTEr sous contraintes avec dérivées 114
1.3 MOPTA08 116
2 Applications 121
2.1 Applications réservoir 121
2.1.1 Problème d'inversion de données de production et de données sismiques : problème de calage historique (ou History Matching) 121
2.1.2 Calcul des scénarios extrêmes 135
2.2 Application en calibration des moteurs 142
3 Conclusions sur les résultats numériques obtenus avec la méthode SQA . . 164

II Optimisation Multi-Objectifs sans dérivées 165

5 Optimisation multi-objectifs 166
1 Définitions 166
2 Approches de résolution 168
2.1 Approches scalaires 169
2.1.1 Méthodes d'agrégations 169
2.1.2 Méthode avec vecteur cible 169
2.1.3 Méthode de ϵ-contraintes 170
2.2 Approches Pareto 170
2.3 Approches non scalaires et non Pareto 170

6 MO-CMA-ES 172
1 Sélection multi-objectifs 172
1.1 Premier critère de sélection : tri par non dominance 173
1.2 Second critère de sélection 173
1.2.1 Distance d'encombrement 173
1.2.2 Mesure de l'hypervolume 174
2 Croisement et mutation 176
2.1 Equations de base 177
2.2 Estimation des paramètres caractéristiques de la distribution 178
2.2.1 Adaptation de l'écart-type global $\sigma_k^{(g)}$: 179
2.2.2 Adaptation de la matrice de covariance $C_k^{(g)}$: 180
3 L'algorithme MO-CMA-ES 182
4 Cadre des tests 184
4.1 Nombre d'exécutions et critère d'arrêt 184
4.2 Evaluation de la performance d'un algorithme d'optimisation multi-objectifs 184

7 Application en calibration des moteurs **187**
1 Formulation du problème de calibration 187
1.1 La mise au point des moteurs 188
1.1.1 Les objectifs 188
1.1.2 Les moyens d'action 190
1.1.3 Le problème d'optimisation des cartographies 192
2 Prise en compte des contraintes dans MOCMAES 195
3 Application sur un cas réel de calibration de moteur 196
3.1 Présentation des résultats 196
3.2 Reformulation du problème 197
3.3 Optimisation aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar 199
3.4 Optimisation aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar avec robustesse 202
3.5 Comparaison entre les résultats du MO-CMA-ES et une méthode d'agrégation 204
4 Conclusions 206

Conclusions générales **208**

Annexes **211**

A Profils de performances et profils de données **212**

B Méthode d'optimisation classique **214**
Méthode de Quasi-Newton 214

C Optimisation avec contraintes : résultats généraux **215**
1 Problème d'optimisation sous contraintes 215
1.1 Formulation du problème 215
1.2 Conditions nécessaires d'optimalité du premier ordre de Karush-Kuhn-Tucker 216
2 Méthodes de pénalisation 218
3 Méthode de programmation quadratique successive 220
4 Pénalisation exacte et globalisation par recherche linéaire de la méthode SQP 223

D Optimisation globale par Krigeage **225**
1 Modèle de Krigeage 225
2 Algorithme d'optimisation par Krigeage 226

E Initialisation de la méthode SQA **228**

F Résultats obtenus sur le benchmark Moré et Wild sous contraintes de bornes **231**
1 Problèmes lisses 231
2 Problèmes bruités 237
3 Problèmes non différentiables 239

G Distribution normale **241**
1 Distribution normale à une dimension 241
2 Distribution normale à N dimensions 241

Bibliographie **255**

Table des figures

1.1 Représentation du simplexe de Nelder-Mead y^0, y^1, y^2 sont tels que $f(y^0) \leq f(y^1) \leq f(y^2)$ 24

1.2 Dix premières itérations de coordinate search (pattern search sans l'étape de recherche, ni d'ordre sur le choix des points dans la base $\mathbb{D} = \mathbb{D}_{\oplus} = [e_1 \cdots e_n - e_1 \cdots - e_n]$) 27

1.3 Méthode de recherche multi-directionnelle à base d'un simplexe 28

1.4 Principe général d'un algorithme évolutionnaire 30

1.5 Krigeage avec ajout de point par espérance de gain 38

3.1 Fonction utilisée en exemple 83

3.2 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour la minimisation (3.52) sous contrainte avec dérivées avec SQA pour un nombre de points d'interpolation de 5 $(2n+1)$ 84

3.3 Evolution du modèle et de la région de confiance de rayon Δ et de la région minimale ρ lors des premières itérations de SQA avec la contrainte considérée comme une contrainte avec dérivées (correspondant aux 11 premières évaluations) 85

3.4 Evolution du modèle et la région de confiance de rayon Δ et de la région minimale ρ lors des itérations de SQA avec la contrainte considérée comme une contrainte avec dérivées (correspondant aux évaluations 12 à 14) . . . 86

3.5 Evolution du rayon de la région de confiance Δ et du rayon minimal ρ au cours des itérations 87

3.6 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour la minimisation (3.52) sous contrainte sans dérivées avec SQA pour un nombre de points d'interpolation de 5 $(2n+1)$ 88

3.7 Evolution du modèle et de la région de confiance de rayon Δ et de la région minimale ρ lors des premières itérations de SQA avec la contrainte considérée comme une contrainte sans dérivées (correspondant aux 11 premières évaluations) 89

3.8 Evolution du modèle et de la région de confiance de rayon Δ et de la région minimale ρ lors des itérations de SQA avec la contrainte considérée comme une contrainte avec dérivées (correspondant aux évaluations 12 à 17) . . . 90
3.9 Evolution du rayon de la région de confiance Δ et du rayon minimal ρ au cours des itérations 91
3.10 Evolution de la valeur de la fonction objectif en fonction du nombre de simulations pour la minimisation de la fonction de moindres carrés (3.60) sans dérivées avec différentes méthodes d'optimisation 95
3.11 Evolution de la valeur de la fonction objectif en fonction du nombre de simulations pour la minimisation de la fonction de moindres carrés (3.61) sans dérivées avec différentes méthodes d'optimisation 96
3.12 Evolution de la valeur de la fonction objectif en fonction du nombre de simulations pour la minimisation de la fonction de moindres carrés (3.62) sans dérivées avec différentes méthodes d'optimisation 97

4.1 Profils de données pour les problèmes lisses du benchmark sous contraintes de bornes More et Wild de taille inférieure à 7 pour différentes précisions τ 103
4.2 Profils de performances pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ 104
4.3 Profils de données pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 105
4.4 Profils de performances pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 106
4.5 Profils de données pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ 107
4.6 Profils de performances pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ 108
4.7 Profils de données pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 109
4.8 Profils de performances pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 109
4.9 Profils de données pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ 110
4.10 Profils de performances pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ 111
4.11 Profils de données pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 112

4.12 Profils de performances pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 113
4.13 Profils de données pour les problèmes du benchmark CUTEr sous contraintes de taille inférieure à 13 pour différentes précisions τ 115
4.14 Profils de performances pour les problèmes du benchmark CUTEr sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . 116
4.15 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation par une méthode SQP classique (iSIGHT-FD) avec une approximation des dérivées par différences finies . . 117
4.16 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation par une méthode SQP classique (Harwell VF13) avec une approximation des dérivées par différences finies . 118
4.17 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation avec une stratégie d'évolution . 119
4.18 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation avec la méthode COBYLA d'approximation linéaire locale 120
4.19 Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation avec la méthode SQA pour un nombre de points d'interpolation de 249 $(2n+1)$ 121
4.20 Fonction objectif en fonction du nombre d'évaluations, avec SQA et SQA Least Square pour deux nombres de points d'interpolation, puis avec SQPAL avec méthode GN 134
4.21 Fonction objectif en fonction du nombre d'évaluations, avec SQA et SQA Least Square pour deux nombres de points d'interpolation, puis avec SQPAL avec méthode BFGS ou avec méthode GN 135
4.22 Etape 1 du problème de prévision : points rouge – données observées ; ligne bleue – données simulées (de T_0 à T_2) 137
4.23 Etape 2 du problème de prévision : scénarios optimiste / pessimiste (lignes vertes) estimant les bornes supérieures et inférieures de la prévision 138
4.24 Production cumulée d'huile : les points bleus représentent les données observées pendant 10 ans, la ligne pointillée noire représente la prévision de production calculée à partir de la solution du calage historique, les autres lignes montrent max / min de la prévision de production pour différentes valeurs de $\varepsilon = 5$, 10, 20, 30 (précision du calage historique) 139
4.25 Erreur relative sur la prévision de production cumulée d'huile 140
4.26 Cette figure est similaire à la figure 4.25 excepté pour les lignes pointillées marron qui correspondent à la prévision de la production sans contrainte d'inégalité non linéaire max / min 141

5.1 Exemples de dominance de Pareto et de non dominance pour une minimisation bi-objectifs . . . 167
5.2 Exemple d'un front de Pareto pour l'optimisation de deux objectifs 168

6.1 Distance d'encombrement en dimension 2 . . . 174
6.2 Mesure de l'hypervolume en dimension 2 . . . 176
6.3 Indicateur de l'hypervolume en dimension 2. Nous n'avons ni $A \rhd B$, ni $B \rhd A$, en revanche l'indicateur de l'hypervolume indique $\mathcal{I}_{S,A} = 0.41$ et $\mathcal{I}_{S,B} = 0.42$ . . . 185

7.1 Exemple d'un cycle de conduite (cycle européen) . . . 189
7.2 Représentation du cycle de conduite européen discrétisé dans le plan régime / charge . . . 190
7.3 Exemple de cartographie associée au réglage de la pression d'injection dans un moteur diesel . . . 191
7.4 Schéma de banc moteur . . . 192
7.5 Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 1 bar . . . 200
7.6 Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 5 bar . . . 201
7.7 Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 1 bar avec l'ajout de la contrainte de robustesse . . . 203
7.8 Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 5 bar avec l'ajout de la contrainte de robustesse . . . 204
7.9 Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 1 bar avec l'ajout de la contrainte de robustesse . . . 205
7.10 Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 5 bar avec l'ajout de la contrainte de robustesse . . . 206

F.1 Profils de données pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 232
F.2 Profils de performances pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 233
F.3 Profils de données pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 234
F.4 Profils de performances pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 235
F.5 Profils de données pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 236
F.6 Profils de performances pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 237
F.7 Profils de données pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ . . . 238

F.8 Profils de performances pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 238
F.9 Profils de données pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 239
F.10 Profils de performances pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ 240

Liste des tableaux

1.1 Tableau récapitulatif des différents solveurs cités 43

4.1 Tableau récapitulatif des différents solveurs 102
4.2 SQPAL vs. SQA sur le problème de maximisation (4.5)–(4.6) 141
4.3 SQPAL vs. SQA sur le problème de minimisation (4.5)–(4.6) 142

6.1 Tableau des valeurs par défaut des paramètres du MO-CMA-ES (N : nombre de variables d'optimisation) 183

7.1 Normes de l'Union européenne pour les émissions des moteurs diesel lourds, g/kWh (fumée en m^{-1}) 189
7.2 Résultats de l'optimisation aux points de fonctionnement 1 750-1 et 1 750-5 201
7.3 Résultats de l'optimisation aux points de fonctionnement 1 750-1 et 1 750-5 avec robustesse 202

Liste des algorithmes

1.1 Méthode de Nelder-Mead 25
1.2 Méthode de recherche directe directionnelle 26
1.3 Algorithme du gradient conjugué non linéaire 32
1.4 Méthode de Quasi-Newton avec approximation de la Hessienne 32
1.5 Implicit Filtering method 34
1.6 Optimisation à l'aide de surface de réponse 37
1.7 Modèles locaux valables dans une région de confiance 40
1.8 Méthode de recherche directe directionnelle hybride 42
2.1 Algorithme de région de confiance classique 48
2.2 Améliorer l'unisolvance à l'aide des polynômes de Lagrange 61
2.3 Etape critique 64
2.4 Méthode d'optimisation avec région de confiance basée sur un modèle quadratique 65
3.1 SQA (Sequential Quadratic Approximation) 71
6.1 Algorithme $\lambda_{MO} \times (\lambda + 1)$-MO-CMA-ES pour $\lambda = 1$ 182
C.1 Algorithme SQP local 222

Introduction

Au cours des XVIIe et XVIIIe siècles, les Mathématiques ont connu un essor sans précédent avec la naissance du calcul infinitésimal et l'analyse algébrique. La communauté scientifique a alors commencé à chercher à représenter le monde réel au travers de formalismes mathématiques. La *fonction* devient un objet d'étude à part entière et se définit comme une combinaison d'opérations à partir de variables et d'éventuels paramètres constants. L'introduction de fonctions toujours plus complexes pour décrire les processus et les prévoir a conduit naturellement au développement de l'optimisation. Cette branche des mathématiques a pour but de rechercher des valeurs de paramètres qui minimisent/maximisent une ou plusieurs fonctions objectif. On appelle fonction objectif, une fonction servant de critère pour déterminer la meilleure solution à un problème d'optimisation. On la dénommera de façon générique par f dans cette introduction. Les méthodes d'optimisation initiales basées sur l'évaluation directe des valeurs de f ont été grandement améliorées par l'utilisation des dérivées de différents ordres de f. En effet ces fonctions sont porteuses d'informations très utiles quand on cherche à optimiser f. Ainsi il est généralement connu que, dans le cas de fonctions continues et différentiables, la recherche d'un minimum local passe par la résolution de l'équation dérivée première de f égale à zéro. Cependant, il existe de nombreux cas où les gradients de f ne sont pas disponibles ou sont inutilisables ; pourtant, l'optimisation de f reste un but à atteindre. La fonction objectif peut également être appelée boîte noire quand aucune information autre que la valeur en un point de celle-ci n'est accessible. De plus, l'évaluation de cette fonction objectif, au travers de mesures ou de simulations numériques est coûteuse. Ce coût, de nature financière, temporelle, humaine ou une combinaison des trois, implique un crédit d'évaluations limité lors de l'optimisation.

C'est pourquoi une nouvelle classe de méthodes a vu le jour, celle de l'optimisation sans dérivées. De nombreux domaines d'application sont en demande constante de nouvelles méthodes d'optimisation sans dérivées du fait principalement de la complexité toujours croissante des modélisations mathématiques et des moyens de calcul toujours plus puissants. De plus, la formulation des problèmes d'optimisation provenant de l'ingénierie nécessite la prise en compte de nombreuses contraintes. Celles-ci peuvent être formulées, mais également être calculées par la même boîte noire que l'évaluation de la fonction objectif, qui ne fournit pas les dérivées, nommées par la suite contraintes sans dérivées, com-

plexifiant encore la résolution des problèmes. Par conséquent, l'optimisation sans dérivées ainsi que la prise en compte de contraintes sont actuellement des domaines de recherche en plein essor et constituent des enjeux importants en ingénierie.

L'optimisation sans dérivées sous contraintes est le sujet principal de ce mémoire de thèse. Les domaines d'application sont très nombreux et variés, allant du calibrage des pales d'un hélicoptère afin de minimiser les vibrations [18], jusqu'à la résistance des véhicules au crash [172]. Il serait évidemment illusoire de croire que l'optimisation sans dérivées puisse résoudre tous les problèmes actuels d'optimisation. Ces méthodes ont également leurs limites, notamment lorsque le nombre de variables devient très important (typiquement supérieur à une centaine) ou que la fonction possède de nombreux minima locaux. Il est donc important de garder à l'esprit qu'à partir du moment où les gradients de f sont disponibles et/ou que les fonctions sont lisses et peu bruitées, il n'est pas conseillé d'utiliser les méthodes d'optimisation sans dérivées qui s'avèreront plus coûteuses en temps de calcul. Ces méthodes sont donc à privilégier pour des cas plus complexes où les coûts de calcul des gradients deviennent prohibitifs par exemple et où le nombre de paramètres restent inférieur à cent.

Ce mémoire de thèse s'articule autour du développement et de l'amélioration de méthodes d'optimisation sans dérivées.

Il est construit en deux parties distinctes intitulées respectivement optimisation sans dérivées (**partie I**) et optimisation multi-objectifs sans dérivées (**partie II**).

La **partie I** comporte quatre chapitres. Le **chapitre 1** présente un état de l'art exhaustif des différentes méthodes d'optimisation permettant de minimiser une fonction coûteuse à évaluer dont les dérivées ne sont pas disponibles. Le **chapitre 2** s'intéresse plus particulièrement à une méthode de région de confiance utilisant un modèle simple de la fonction objectif local afin de diriger l'optimisation. Au cours du **chapitre 3**, nous proposons une méthode d'optimisation de ce type permettant la prise en compte de contraintes, extension de la méthode proposée par Powell [139, 140]. Nous appellerons cette méthode SQA (Sequential Quadratic Approximation) dans la suite de ce mémoire. Enfin le **chapitre 4** présente les résultats obtenus avec cette méthode sur des benchmarks d'optimisation sous contraintes. Une application de calage d'historique et une approche de calcul d'incertiture de production sont abordées. Puis le principe de la calibration des moteurs est décrit en détails ainsi que les problèmes d'optimisation associés. Enfin, les résultats obtenus sur ces applications dans les domaines de l'ingénierie de réservoir et des moteurs sont exposés. Ces résultats sont comparés avec les résultats obtenus par d'autres méthodes, décrites brièvement dans l'état de l'art du chapitre 1.

La **partie II** est organisée en trois chapitres. Le **chapitre 5** pose les différents problèmes inhérents à l'optimisation multi-objectifs ainsi que les différentes approches existantes. Le **chapitre 6** est centré sur l'algorithme MO-CMA-ES (Multi-Objective - Covariance Matrix Adaptation - Evolution Strategy), un algorithme évolutionnaire multi-objectifs, proposé par Hansen [85]. Enfin le **chapitre 7** de la partie II présente la for-

mulation multi-objectifs du problème de calibration des moteurs. La méthode de prise en compte des contraintes employée sur cette application moteur est décrite, ainsi que les résultats obtenus en optimisation multi-objectifs. Les résultats sont ensuite comparés à ceux obtenus avec une méthode mono-objectif approchant le problème par une fonction d'agrégation linéaire des objectifs visés.

Sept annexes ont été ajoutées à la fin du manuscrit. Dans l'**annexe A** les notions de profils de performance et de profils de données sont présentées. La méthode de mise à jour Quasi-Newton fait l'objet de l'**annexe B**. Dans l'**annexe C** les résultats généraux de l'optimisation avec contrainte sont présentés. L'optimisation globale par krigeage est décrite dans l'**annexe D**. Un descriptif exhaustif de l'initialisation de la méthode SQA est présenté en **annexe E**. Les profils de performance et les profils de données obtenus sur le benchmark, proposé par Moré et Wild [121], sous contraintes de bornes de taille inférieure à 13 sont analysés et comparés dans l'**annexe F**. Enfin des rappels sur la distribution normale à une et à N dimension(s) sont présentés en **annexe G**.

Première partie

Optimisation sans dérivées

Chapitre 1

Etat de l'art

Ce chapitre fournit un aperçu des méthodes d'optimisation sans dérivées (DFO : "Derivative Free Optimization" [34, 135, 141]) pour résoudre

$$\min_{x \in \mathbb{R}^n} f(x). \tag{1.1}$$

Dans le cadre d'une optimisation sans dérivées, on suppose que les gradients de la fonction f ne sont pas disponibles (trop coûteux à calculer ou non calculables). De plus, le calcul de f résultant d'une mesure physique, chimique ou économique, d'une simulation, de la résolution d'un système d'équations, d'une équation aux dérivées partielles, peut être coûteux.

Ainsi, ces méthodes d'optimisation sans dérivées utilisent exclusivement les évaluations de la fonction objectif lors de leur déroulement. Ce sont de bonnes alternatives aux méthodes classiques d'optimisation substituant les gradients de la fonction objectif par des approximations par différences finies, par exemple, qui nécessitent le choix difficile en pratique d'une valeur de la perturbation à appliquer.

Nous présentons tout d'abord les méthodes de recherche directe : la méthode du Simplexe de Nelder-Mead et la méthode de Pattern Search qui comparent toutes les deux les valeurs de la fonction aux points évalués pour se déplacer suivant des critères géométriques.

Ensuite les méthodes d'échantillonnage ou métaheuristiques (incluant les algorithmes évolutionnaires) sont présentées, ceux-ci utilisent la notion de sélection naturelle introduite au $XIX^{\text{ème}}$ siècle par Darwin et l'appliquent à une population de solutions potentielles au problème de minimisation.

Puis, l'optimisation à l'aide de modèles de substitution de la fonction objectif est présentée. Cette idée semble intéressante puisqu'elle remplace l'optimisation de la fonction objectif par l'optimisation moins coûteuse d'un modèle de substitution.

Enfin, les méthodes hybrides sont également exposées. Ces méthodes associent généralement deux méthodes différentes pour résoudre ce problème d'optimisation.

Nous concluons ce chapitre par une discussion sur les avantages et inconvénients des méthodes décrites.

1 Recherche directe

Les méthodes de recherche dites "directe" sont des méthodes de DFO qui échantillonnent la fonction objectif en un nombre fini de points à chaque itération et décident des évaluations suivantes à partir de ces évaluations sans aucune approximation des dérivées ni aucun modèle de la fonction. Ces méthodes peuvent être classées en deux catégories : les méthodes basées sur un simplexe et des opérations sur ce simplexe (réflexion, expansion ou contraction) comme la méthode de Nelder-Mead [124] et les méthodes de recherche directe guidées par un ensemble de directions.

1.1 Méthode du simplexe de Nelder-Mead

La première méthode du simplexe a été proposée par Spendley et al en 1962 [161]. En 1965, la méthode de minimisation de Nelder-Mead [124] ou méthode du simplexe est basée sur la comparaison des valeurs de la fonction aux $n+1$ sommets d'un simplexe général $Y = \{y^0, y^1, ..., y^n\} : f(y^0) \leq f(y^1) \leq ... \leq f(y^n)$. Le simplexe se modifie à travers des opérations de réflexion, d'expansion ou de contraction (cette dernière peut être intérieure ou extérieure), en remplaçant le sommet associé à la plus grande valeur de la fonction coût y^n par un autre point de la droite passant par y^n et $y^c : y = y^c + \delta(y^c - y^n), \delta \in \mathbb{R}$, où $y^c = \sum_{i=0}^{n-1} y^i/n$ est le barycentre des n meilleurs points. Les valeurs de δ indiquent le type de l'itération. Les choix standards pour ces coefficients sont : $\delta^r = 1$ pour la réflexion, $\delta^e = 2$ pour l'expansion, $\delta^{ci} = -1/2$ et $\delta^{ce} = 1/2$ pour les contractions intérieures et extérieures. Nous représentons ces différentes situations sur la figure 1.1 (cas $n = 2$).

Si aucun des points évalués ne réduit la fonction objectif, le simplexe est réduit en y^0, tous les sommets de Y excepté y^0 sont remplacés par $y^0 + \gamma^s(y^i - y^0), i = 1, ..., n$, $\gamma^s = 1/2$ (voir figure 1.1).

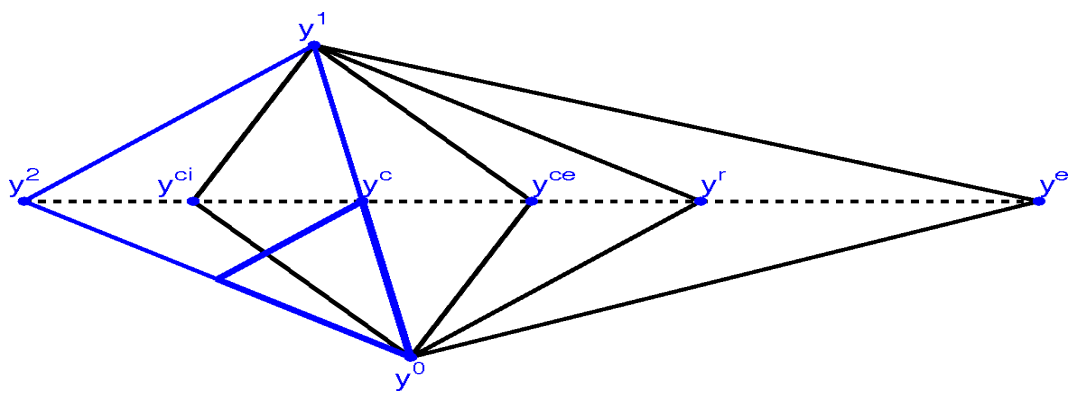


FIGURE 1.1 – Représentation du simplexe de Nelder-Mead y^0, y^1, y^2 sont tels que $f(y^0) \leq f(y^1) \leq f(y^2)$

Algorithme 1.1 Méthode de Nelder-Mead

- **Initialisation** : Choix d'un simplexe de sommets $Y_0 = \{y_0^0, y_0^1, ..., y_0^n\}$. Evaluer f en ces points. Choisir les constantes : $0 < \gamma^s < 1$, $-1 < \delta^{ic} < 0 < \delta^{oc} < \delta^r < \delta^e$.
- Tant que le diamètre $diam(Y)$ du simplexe Y n'atteint pas Δ_{tol} :
  (1) $Y = Y_k$
  (2) Ranger les $n+1$ sommets $Y = \{y^0, y^1, ..., y^n\}$ tel que $f^0 = f(y^0) \leq f^1 = f(y^1) \leq ... \leq f^n = f(y^n)$.
  (3) **Réflexion** : Réfléchir le plus mauvais sommet y^n sur le barycentre $y^c = \sum_{i=0}^{n-1} y_i/n$ des sommets restants : $y^r = y^c + \delta^r(y^c - y^n)$. Evaluer $f^r = f(y^r)$. Si $f^0 \leq f^r < f^{n-1}$, le point réfléchi y^r remplace y^n et l'itération est terminée : $Y_{k+1} = \{y^0, y^1, ..., y^{n-1}, y^r\}$.
  (4) **Expansion** : Si $f^r < f^0$, le point d'expansion $y^e = y^c + \delta^e(y^c - y^n)$ est évalué $f^r = f(y^r)$. Si $f^e \leq f^r$, le point d'expansion y^e remplace y^n et l'itération est terminée : $Y_{k+1} = \{y^0, y^1, ..., y^{n-1}, y^e\}$. Autrement, le point réfléchi y^r remplace y^n et l'itération est terminée : $Y_{k+1} = \{y^0, y^1, ..., y^{n-1}, y^r\}$.
  (5) **Contraction** : Si $f^r \geq f^{n-1}$, une contraction est effectuée entre y^r ou y^n suivant la plus petite valeur de $f(y^r)$ et $f(y^n)$.
    (a) **Contraction extérieure** : Si $f^r < f^n$, le point de contraction extérieure $y^{ce} = y^c + \delta^{ce}(y^c - y^n)$ est évalué $f^{ce} = f(y^{ce})$. Si $f^{ce} \leq f^r$, le point de contraction extérieure y^{ce} remplace y^n et l'itération est terminée : $Y_{k+1} = \{y^0, y^1, ..., y^{n-1}, y^{ce}\}$. Autrement une réduction du simplexe est effectuée.
    (b) **Contraction intérieure** : Si $f^r \geq f^n$, le point de contraction intérieure $y^{ci} = y^c + \delta^{ci}(y^c - y^n)$ est évalué $f^{ci} = f(y^{ci})$. Si $f^{ci} < f^n$, le point de contraction intérieure y^{ci} remplace y^n et l'itération est terminée : $Y_{k+1} = \{y^0, y^1, ..., y^{n-1}, y^{ci}\}$. Autrement une réduction du simplexe est effectuée.
  (6) **Réduction** : Evaluer f aux n points $y^0 = \gamma^s(y^i - y^0), i = 1, ..., n$, et remplacer $y^1, ..., y^n$ par ces nouveaux points et l'itération est terminée : $Y_{k+1} = y^0, y^1, ..., y^{n-1}, y^e$. Autrement, le point réfléchi y^r remplace y^n et l'itération est terminée : $Y_{k+1} = \{y^0 + \gamma^s(y^i - y^0), i = 0, ..., n\}$.

La méthode s'arrête lorsque le diamètre du simplexe Y de sommets $y^0, y^1, ..., y^n$: $diam(Y) = max_{0 \leq i < j \leq n} \|y^i - y^j\|$ est réduit jusqu'à Δ_{tol}.

Cette méthode est très simple à implémenter, ce qui constitue son principal avantage. Les inconvénients de cette méthode sont la prise en compte de contraintes non linéaires qui n'est pas forcément évidente ainsi que le nombre élevé d'évaluations de la fonction objectif nécessaires lors de chaque itération. En effet, 1, 2 ou $n + 2$ évaluations par itération sont effectuées, cette méthode est donc efficace lorsque la dimension de l'espace des variables est peu élevée. De plus, le choix du simplexe initial n'est pas évident et le résultat de l'optimisation en dépend fortement.

1.2 Recherche directe directionnelle

Les méthodes de recherche directe directionnelle utilisent un ensemble de directions avec des propriétés appropriées pour guider l'optimisation.

1.2.1 Pattern Search

Algorithme 1.2 Méthode de recherche directe directionnelle

- **Initialisation** : Choix de $x_0, \alpha_0 > 0, 0 < \beta_1 \leq \beta_2 < 1$ et $\gamma \geq 1$. Soit $\mathbb{D}$ un ensemble formant une base positive.
- Tant que le diamètre $\alpha_k \geq \alpha_{tol}$:
  (1) **Etape de recherche** : Essayer de calculer un point qui vérifie $f(x) < f(x_k)$ en évaluant la fonction objectif en un nombre fini de points. Si un point est trouvé, $x_{k+1} = x$, l'itération et l'étape de recherche sont réussies et l'étape de sondage est passée.
  (2) **Etape de sondage** : Choisir une base positive $\mathbb{D}_k$ à partir de l'ensemble $\mathbb{D}$. Ranger l'ensemble des points du sondage $P_k = \{x_k + \alpha_k d / d \in \mathbb{D}_k\}$. Démarrer les évaluations de f aux points de sondage suivant l'ordre choisi. Si un point sondé $x_k + \alpha_k d_k$ vérifie $f(x_k + \alpha_k d_k) < f(x_k)$, le sondage est stoppé, $x_{k+1} = x_k + \alpha_k d_k$, l'itération et l'étape de sondage sont réussies. Sinon, l'itération (et l'étape de sondage) échoue et $x_{k+1} = x_k$.
  (3) **Mise à jour du maillage** : Si l'itération est réussie, le pas du maillage est maintenu ou augmenté : $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$. Autrement, réduire le pas du maillage : $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

Les méthodes de type Pattern Search ont été introduites pour la recherche directe par Hooke et Jeeves [90]. Les papiers de Torczon [162, 163] sont une référence pour ces méthodes. Nous décrivons maintenant une méthode de recherche directe directionnelle basée sur la méthode de "pattern-search" de [3]. A chaque itération, x_k est le point courant et α_k la valeur courante pour le pas d'échantillonnage ou le pas du maillage. L'objectif de l'itération k est de déterminer l'itération suivante x_{k+1} telle que $f(x_{k+1}) < f(x_k)$. Celle-ci s'effectue en deux phases : la phase de recherche ou "search step" et la phase de sondage ou "poll step". L'étape de recherche est optionnelle, elle consiste à évaluer la fonction objectif en un nombre fini de points. Ces points sont choisis arbitrairement mais un modèle de substitution de f (voir section 4 de ce chapitre) pourrait être utilisé afin d'améliorer l'efficacité de la méthode de recherche directe. Cette étape est réussie si un nouveau point x_{k+1} vérifie $f(x_{k+1}) < f(x_k)$. L'étape de sondage est effectuée seulement si l'étape de recherche a échoué. Elle consiste en une recherche locale autour de l'itération courante : la fonction f est évaluée aux points de l'ensemble $P_k = \{x_k + \alpha_k d / d \in \mathbb{D}\}$ (base positive $\mathbb{D}$ qui permet de définir les directions de recherche de la méthode) afin de trouver un point de P_k qui réduit la valeur de la fonction objectif f. P_k est appelé l'ensemble des points du sondage et $\mathbb{D}$ est l'ensemble des directions du sondage. Cette étape de sondage

est réussie lorsqu'un nouveau point $x_{k+1} \in P_k$ vérifie $f(x_{k+1}) < f(x_k)$. Si l'étape ne parvient pas à produire un point dans P_k où la fonction objectif est inférieure à $f(x_k)$, l'itération k est infructueuse, alors le pas du maillage α_k est réduit. Lorsque l'itération est réussie (une des deux étapes fait décroître la fonction objectif f), le pas de la grille α_k est inchangé ou augmenté. Cette classe de méthode est décrite dans l'algorithme 1.2.

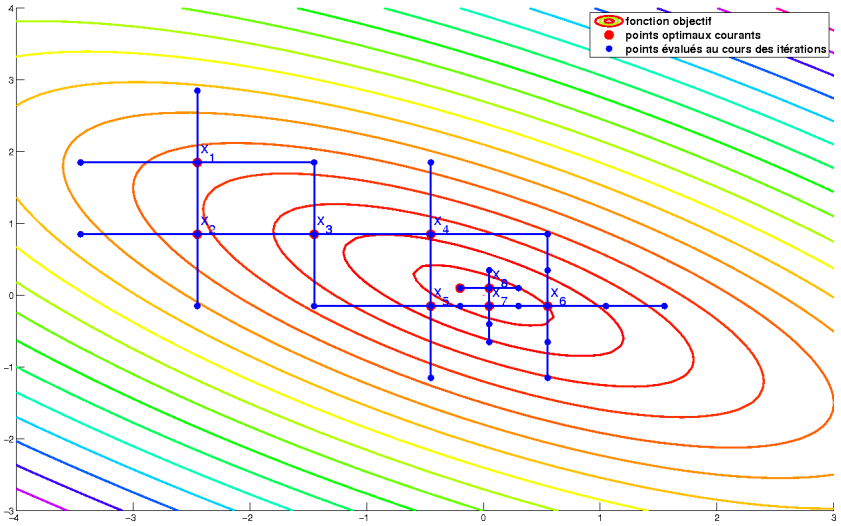


FIGURE 1.2 – Dix premières itérations de coordinate search (pattern search sans l'étape de recherche, ni d'ordre sur le choix des points dans la base $\mathbb{D} = \mathbb{D}_\oplus = [e_1 \cdots e_n - e_1 \cdots - e_n]$)

Cette méthode est une généralisation de la méthode de "coordinate search" ou "compass search" de [21, 20] qui utilise la base positive maximale $\mathbb{D} = \mathbb{D}_\oplus = [e_1 \cdots e_n - e_1 \cdots - e_n]$ sans l'étape de recherche (avec les paramètres de l'algorithme 1.2 suivants : $\beta_1 = \beta_2 = 1/2$ et $\gamma = 1$ ou 2). La recherche est effectuée le long d'une grille de pas α_k, le pas est réduit lorsqu'il ne permet pas de trouver un point pour lequel la fonction est inférieure au minimum courant (voir la figure 1.2). En effet, pour chaque α_k les points évalués sont restreints aux points de la grille régulière de pas α_k autour de x_k. Ainsi, le nombre de points de la grille évalués est fini dans une région bornée de $\mathbb{R}^n$. Les conditions pour réduire α_k dépendent de l'ensemble généré $\mathbb{D}_{\oplus_k}$ à l'itération courante k. Soit x^k le meilleur point de l'itération k, c'est-à-dire le point évalué de la grille courante qui correspond à la plus petite valeur de fonction objectif. Si un point de l'ensemble $\mathbb{D}_\oplus$ vérifie $f(x_{\mathbb{D}_{\oplus_k}}) < f(x_k)$ alors $x_{k+1} = x_{\mathbb{D}_\oplus}$, sinon, $x_{k+1} = x_k$ et α_k est réduit de moitié pour la prochaine itération.

Une mine d'informations sur les méthodes "pattern search", dont un développement permettant de prendre en compte des contraintes, se trouve dans l'article de Kolda, Lewis et Torczon [108] (2003).

Les méthodes de pattern search ont l'avantage de pouvoir être parallélisées facilement, mais sont coûteuses en nombre d'évaluations de la fonction objectif.

1.2.2 Méthode de recherche multi-directionnelle (MDS)

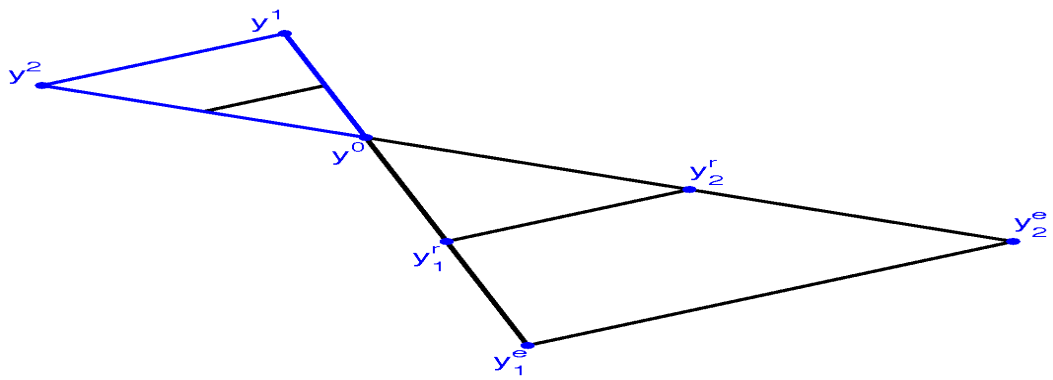


FIGURE 1.3 – Méthode de recherche multi-directionnelle à base d'un simplexe

La méthode de recherche multi-directionnelle [47] est une méthode de recherche directionnelle qui utilise un simplexe. Comme pour la méthode de Nelder-Mead, MDS démarre avec un simplexe de $n+1$ sommets $Y = \{y^0, y^1, ..., y^n\}$. Chaque itération est centrée sur le sommet de simplexe y^0 qui correspond à la valeur de la fonction objectif la plus faible (au contraire de Nelder-Mead qui se focalise sur le sommet qui correspond à la valeur de fonction objectif la plus grande). Ensuite un simplexe est construit par une symétrie centrale des $y^i, i = 1, n$ par rapport à y^0 (voir figure 1.3). Si une meilleure valeur de la fonction objectif est obtenue, une extension du simplexe est effectuée dans la même direction que le simplexe pivoté et l'itération suivante recommencera à partir du meilleur simplexe pivoté ou étendu. Par contre si la meilleure valeur de fonction objectif obtenue par le simplexe pivoté n'est pas inférieure à $f(y^0)$, une étape de réduction du simplexe est effectuée comme pour Nelder-Mead et l'itération suivante recommencera à partir du simplexe réduit.

Cette méthode, contrairement à la méthode du simplexe de Nelder-Mead, explore plusieurs directions, permettant une recherche moins précise dans la direction de descente, ce qui peut être intéressant étant donné le peu d'informations recueillies sur la fonction objectif à minimiser. Une démonstration de la convergence globale de cette méthode est disponible dans [162, 34, 33].

1.2.3 Méthodes de directions conjuguées

La recherche directe ne se limite pas à des méthodes directionnelles basées sur un ensemble fixé de directions. En réalité, certaines des premières méthodes directes de recherche directionnelle modifient les directions de recherche à la fin de chaque itération en combinant un ensemble de directions préalablement calculées. Ces méthodes avec des ensembles adaptatifs de directions de recherche ont été proposées par Rosenbrock [150] et Powell [129, 135] qui utilisent des directions conjuguées. Celles-ci tentent d'accélérer

la recherche en construisant des directions à partir d'informations sur la courbure de la fonction objectif recueillies au cours de la recherche. Une approche récente a été poursuivie par Frimannslund et Steihaug [69] par rotation de l'ensemble de directions basée sur les informations de courbure extraites des valeurs de la fonction.

Cette méthode est l'une des premières à utiliser l'information de courbure obtenue au cours des itérations.

2 Métaheuristiques ou méthodes d'échantillonnage

Les méthodes de recherche par échantillonnage pour l'optimisation, aussi appelées métaheuristiques, ou optimisation probabiliste, consistent à échantillonner l'espace des solutions admissibles. Elles sont souvent inspirées par des systèmes naturels provenant de la physique (cas du recuit simulé), de la biologie (cas des algorithmes génétiques inspirés de la théorie de l'évolution) ou encore de l'éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires). Les métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin de trouver une approximation de la meilleure solution.

2.1 Algorithme évolutionnaire

Contrairement aux méthodes présentées dans les sections précédentes, les algorithmes évolutionnaires sont des algorithmes d'optimisation globale. Ce sont des processus itératifs faisant évoluer un ensemble de points de façon à explorer efficacement l'espace de recherche. Ceci est un avantage par rapport aux algorithmes « classiques » d'optimisation, méthodes de gradient, méthodes de Gauss-Newton, méthodes de quasi-Newton (ex : BFGS), qui sont des algorithmes de recherche locale pouvant converger vers un optimum local. Ainsi lorsque le nombre de minima locaux est grand, les algorithmes évolutionnaires sont plus efficaces qu'une méthode multi-start (plusieurs optimisations avec différentes initialisations) à base de méthodes locales. D'autre part, contrairement aux méthodes quasi-newtoniennes, les algorithmes évolutionnaires ne nécessitent pas d'hypothèse de régularité sur la fonction objectif.

Les algorithmes évolutionnaires s'inspirent des mécanismes de sélection naturelle introduite par Darwin ([38] théorie de l'évolution). Précisément, les individus les mieux adaptés à l'environnement ont tendance à vivre assez longtemps et à se reproduire, alors que les plus faibles ont tendance à disparaître.

Par analogie avec l'évolution naturelle, un algorithme évolutionnaire est un processus itératif faisant évoluer un ensemble de solutions candidates, appelé « population », composé d'« individus » représentant des solutions possibles du problème d'optimisation. Au cours de chaque itération, appelée « génération », la performance des individus appelés « parents » est évaluée à l'aide de la fonction objectif. Les meilleurs individus sont

sélectionnés pour être « géniteurs », puis subissent des croisements et des mutations afin de diversifier la population. Les individus créés (« enfants ») forment une nouvelle population pour la génération suivante après remplacement des anciens individus par les nouveaux. Le processus se poursuit de génération en génération jusqu'à ce qu'un critère d'arrêt soit satisfait (ex. : nombre maximal de générations ou d'évaluations de la fonction coût).

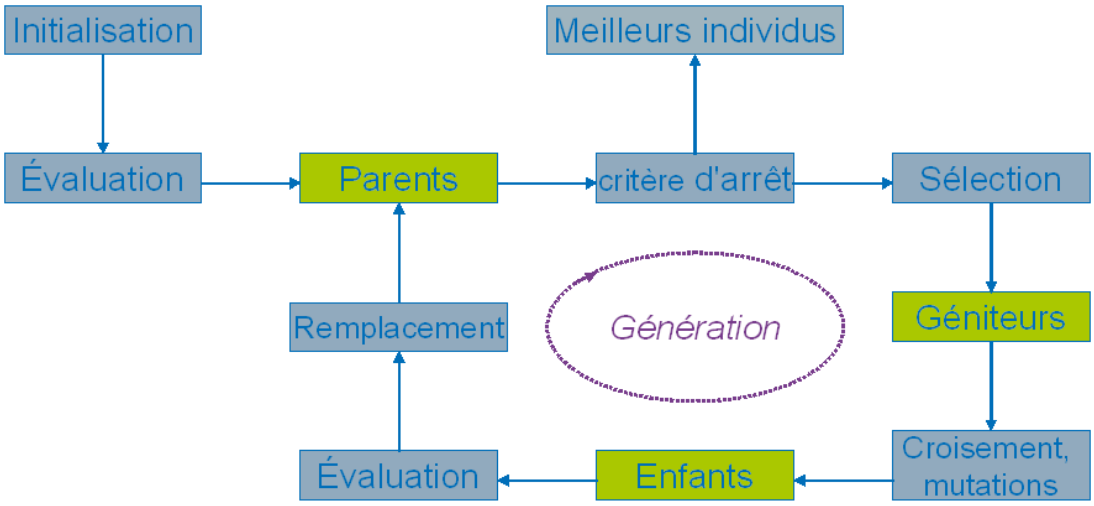


FIGURE 1.4 – Principe général d'un algorithme évolutionnaire

Un algorithme évolutionnaire repose sur l'application des trois opérateurs de sélection, de croisement, et de mutation sur une population d'individus (voir figure 1.4). Ces méthodes ont été appliquées pour résoudre de nombreux problèmes réels, mais sont toutefois coûteuses en nombre d'évaluations de la fonction objectif.

Les algorithmes évolutionnaires sont composés de quatre grandes familles d'algorithmes [175, 96] : les stratégies d'évolutions, proposées par I. Rechenberg [144], la programmation évolutionnaire conçue par Fogel et al, la programmation génétique et les algorithmes génétiques [89, 75].

La stratégie d'évolution de Hansen [87, 86, 88, 36], le CMA-ES (Covariance Matrix Adaptation Evolution Strategy), est un algorithme d'optimisation globale très performant notamment dans le cas de fonctions non linéaires à paramètres continus pouvant présenter de nombreux optima locaux. De plus, elle ne nécessite pas d'hypothèse de régularité sur la fonction objectif à optimiser. Une version multi-objectifs de cet algorithme existe également, celle-ci appelée MO-CMA-ES [85] (Multi-Objective - Covariance Matrix Adaptation - Evolution Strategy) permet d'obtenir l'intégralité du front de Pareto, et se révèle particulièrement efficace pour l'optimisation sous contraintes [109, 111, 112]. Cette méthode sera décrite en détail dans le chapitre 6.

2.2 Recuit simulé

Le recuit simulé (simulated annealing [105, 135]) est une méthode d'optimisation globale inspirée d'un processus utilisé en métallurgie. La méthode d'optimisation considère chaque point x de l'espace de recherche comme l'état d'un système physique et la fonction objectif $f(x)$ comme l'énergie de cet état. L'objectif est de mettre le système d'un état arbitraire à un état à énergie la plus faible possible. L'algorithme génère une suite de solutions (x_n) comme suit : soit x_n la solution à l'itération n. Une nouvelle solution y_n est générée aléatoirement dans un voisinage de x_n. Le critère de choix du nouveau point y_n est la règle de Boltzmann définie comme suit :

$$x_{n+1} = \begin{cases} y_n & \text{si } f(y_n) \leq f(x_n), \\ y_n & \text{si } f(y_n) > f(x_n) \text{ avec la probabilité } e^{\frac{f(x_n)-f(y_n)}{t_n}}, \\ x_n & \text{autrement,} \end{cases} \tag{1.2}$$

où t_n est la température : un paramètre positif qui tend vers zéro. L'objectif de l'échantillonnage par le critère de choix de Boltzmann de la nouvelle solution y_n est d'éviter de rester coincé dans un optimum local. En pratique, la suite t_n doit être une suite décroissante permettant que la probabilité d'accepter une mauvaise solution décroisse au cours des itérations. Les résultats de convergence requièrent que cette suite décroisse lentement vers zéro pour échapper aux minima locaux. L'inconvénient majeur de ce type de méthode est le réglage du paramètre de température t_n.

2.3 Autres métaheuristiques

De nombreuses autres métaheuristiques peuvent être utilisées pour l'optimisation sans dérivées de fonction continue comme les algorithmes d'optimisation par essaims particulaires ([104] particle swarm) et les algorithmes de colonies de fourmis ([25, 53] ant colony optimization) implémentés tout d'abord pour l'optimisation combinatoire et adaptés ensuite pour l'optimisation continue [15].

Ces classes de méthodes permettent d'effectuer une optimisation globale de la fonction objectif, et peuvent facilement être parallélisées. En revanche, le nombre d'évaluations nécessaires au fonctionnement de ces méthodes étant élevés, ces méthodes deviennent vite problématiques pour des problèmes de taille moyenne.

3 Optimisation à l'aide de modèles de substitution

L'utilisation de modèles de substitution du gradient de la fonction objectif ou/et de la fonction objectif elle-même permet de tirer un maximum d'informations sur la fonction objectif à partir des évaluations précédemment effectuées. Nous répartissons donc l'optimisation à base de modèle de substitution en deux classes : les méthodes utilisant des modèles du gradient de la fonction objectif et les méthodes utilisant un modèle de celle-ci.

3.1 Modèles des gradients de la fonction objectif

Concernant l'optimisation avec utilisation des gradients de la fonction objectif, il est important de rappeler que dans le cadre de l'optimisation de fonction coûteuses en provenance d'un modèle boîte noire, la différentiation automatique ne peut pas être utilisée car le code de calcul de la fonction objectif n'est pas disponible. Néanmoins, si cette méthode peut être utilisée, nous notons qu'elle peut rencontrer des difficultés lorsque la fonction objectif est bruitée.

3.1.1 Approximations des gradients pour les méthodes classiques

La construction de modèles de substitution des dérivées premières de la fonction objectif permet d'utiliser des méthodes classiques d'optimisation [125, 17] pour les problèmes d'optimisation sans dérivées : méthode du gradient conjugué non linéaire et les méthodes de quasi-Newton (algorithme 1.3 et 1.4 ci-dessous).

Algorithme 1.3 Algorithme du gradient conjugué non linéaire

- **Initialisation** : Soit x_0. Evaluation de $f(x_0)$ et de $\nabla f_0 = \nabla f(x_0)$. Soit $p_0 = -\nabla f_0$ et $k = 0$.
- Tant que $\nabla f_k \neq 0$
  (1) Soit $x_{k+1} = x_k + \alpha_k p_k$, avec α_k calculé à partir d'une procédure de recherche linéaire afin de satisfaire les conditions de Wolfe.
  (2) Evaluer ∇f_{k+1}
  (3) $\beta_{k+1} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$
  (4) $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$
  (5) $k = k + 1$

Algorithme 1.4 Méthode de Quasi-Newton avec approximation de la Hessienne

- **Initialisation** : Soit x_0, $\epsilon > 0$ la tolérance pour la convergence, H_0 l'approximation de l'inverse de la Hessienne. $k = 0$.
- Tant que $\|\nabla f_k\| > \epsilon$
  (1) Calculer la direction de recherche $p_k = -H_k \nabla f_k$.
  (2) Soit $x_{k+1} = x_k + \alpha_k p_k$, où α_k est calculé à partir d'une procédure de recherche linéaire afin de satisfaire les conditions de Wolfe.
  (3) Définir $s_k = x_{k+1} - x_k$ et $y_k = \nabla f_{k+1} - \nabla f_k$
  (4) Mise à jour de H_{k+1} par des formules de type DFP, BFGS (voir [125] et annexe B) ou SR1.
  (5) $k = k + 1$

Une approximation du gradient $\nabla f(x)$ peut être obtenue à partir de l'évaluation de la fonction objectif f en $(n+1)$ points. Nous introduisons deux formules : les différences finies

avant et les différences finies centrées. Supposons que nous voulions calculer le gradient $\nabla f(x)$, la formule de différence finie avant approchant la dérivée partielle $\partial f/\partial x_j$ est

$$\frac{\partial f}{\partial x_j} = \frac{f(x + h_j e_j) - f(x)}{h_j}, \tag{1.3}$$

et celle par différence finie centrée, requérant $2n$ points, est

$$\frac{\partial f}{\partial x_j} = \frac{f(x + h_j e_j) - f(x - h_j e_j)}{2h_j}, \tag{1.4}$$

où h_j est le pas de différence finie et e_j est le j ième vecteur unité. Le gradient peut être calculé en appliquant la formule pour $i = 1, 2, ..., n$.

La principale difficulté de l'approximation par différence finie est le choix du pas de différence finie h_j. Pour obtenir l'approximation des dérivées la plus précise possible, le pas de différence finie doit être choisi le plus petit possible. Toutefois, nous remarquons qu'une petite erreur ϵ dans le numérateur des approximations 1.3 ou 1.4 provoque une erreur ϵ/h_j sur les dérivées. Ainsi, un petit pas de différence finie peut mener à une erreur d'approximation conséquente et h_j doit être choisi attentivement en fonction de la précision du calcul de $f(x)$. Dennis et Schnabel [46] suggèrent certains choix de pas de différence finie. Pour les applications, l'erreur de calcul de la fonction objectif peut être relativement importante, il n'est donc pas approprié de choisir le pas de différence finie h_j trop petit.

Il existe d'autres approximations des dérivées premières que les différences finies : les évaluations précédentes proches du point courant complétées par certaines évaluations choisies peuvent être utilisées pour construire, par interpolation ou régression [34], un modèle linéaire local. Le gradient de ce modèle sera ensuite considéré comme l'approximation du gradient utilisé dans les méthodes classiques d'optimisation différentiable. Cette technique permet d'obtenir l'approximation du gradient de la fonction objectif à partir d'un nombre d'évaluations de la fonction objectif moins important que celui nécessaire pour des différences finies classiques.

Ces méthodes d'optimisation différentiable utilisant des gradients approchés sont assez difficiles à mettre en œuvre en pratique. En effet, le choix du pas de différences finies, qu'il soit fixe ou adaptatif, est assez difficile à effectuer en raison du bruit numérique de la réponse du simulateur. Dans le cas de gradient de modèle réutilisant certaines évaluations, le choix des points à utiliser n'est pas évident. Cependant, ces méthodes ont permis de résoudre certaines applications en ingénierie de réservoir (chapitre 4 ou [44, 159]).

3.1.2 Méthode de recherche linéaire basée sur les dérivées provenant d'un simplexe

La méthode "Implicit-filtering" de Kelley [183, 103] peut être vue comme une méthode de recherche linéaire basée sur les dérivées provenant d'un simplexe. Cette méthode, décrite dans l'algorithme 1.5, est assez efficace pour traiter les fonctions objectifs bruitées.

Ainsi, à partir des évaluations de la fonction objectif pour un ensemble de $n+1$ points, où $Y = \{y_k^0, y_k^1, ..., y_k^n\}$ est unisolvant ("poised", voir la définition 2.2 dans la section 2.1 du chapitre 2) pour l'interpolation linéaire, un gradient provenant de ce simplexe basé en y_k^0 est calculé. Le gradient est alors donné par : $\nabla_s f(x_k) = L_k^{-1}\delta f(Y_k)$ où $L_k = [y_k^1 - y_k^0 \cdots y_k^n - y_k^0]^T$ et

$$\delta f(Y_k) = \begin{bmatrix} f(y_k^1) - f(y_k^0) \\ \vdots \\ f(y_k^n) - f(y_k^0) \end{bmatrix}.$$

Nous définissons également $\Delta_k = \max_{1\leq i\leq n} \|y_k^i - y_k^0\|$.

A chaque itération de recherche linéaire, la condition d'Armijo sur la réduction minimale de la fonction doit être vérifiée pour sélectionner le nouveau point : $f(x_k + \alpha d_k) - f(x_k) \leq \nu\alpha\nabla_s f(x_k)^T d_k$.

Algorithme 1.5 Implicit Filtering method

- **Initialisation** : Choix de $\beta, \nu \in (0, 1)$. Choix d'un point initial x_0 et l'approximation de la Hessienne initiale H_0. Sélectionner $j_{max} \in \mathbb{N}$. Soit $\mathbb{D}$ un ensemble formant une base positive (voir la définition 2.1 dans la section 2.1 du chapitre 2).
- Tant que le diamètre $\Delta_k \geq \Delta_{tol}$:
  (1) **Calcul du gradient du simplexe** : Calculer le gradient du simplexe $\nabla_s f(x_k)$ tel que $\Delta_k \leq \|\nabla_s f(x_k)\|$ (voir [34]). Calculer $d_k = -H_k^{-1}\nabla_s f(x_k)$.
  (2) **Recherche linéaire** : Pour $j = 0, 1, 2, ..., j_{max}$
    (a) Fixer $\alpha = \beta^j$ et évaluer $f(x_k + \alpha d_k)$.
    (b) Si la condition de réduction suffisante $f(x_k + \alpha d_k) - f(x_k) \leq \nu\alpha\nabla_s f(x_k)^T d_k$ est vérifiée pour α, la recherche linéaire est terminée avec $\alpha_k = \alpha$.
  (3) **Nouveau point** : Si la recherche linéaire est réussie, alors $x_{k+1} = x_k + \alpha_k d_k$.
  (4) **Mise à jour de la Hessienne** : Si la recherche linéaire échoue, fixer $H_{k+1} = H_0$ et passer à l'itération suivante. Autrement mettre à jour H_{k+1} à partir de H_k en utilisant une mise à jour quasi-Newton (BFGS par exemple, voir l'annexe B) basée sur $x_{k+1} - x_k$ et $\nabla_s f(x_{k+1}) - \nabla_s f(x_k)$ (voir annexe B).

En pratique, lorsqu'une recherche linéaire échoue, il peut être nécessaire d'effectuer certains ajustements pour le bon usage de cet algorithme. Par exemple, nous pouvons recalculer les points échantillonnés (afin de vérifier $\Delta_k \leq \mu\|\nabla_s f(x_k)\|$ pour tout $1 > \mu > 0$ [34]).

Cette méthode peut être assimilée à une méthode "classique" de quasi-Newton substituant des gradients de la fonction objectif par une approximation de ceux-ci. La recherche linéaire peut échouer en raison d'une mauvaise approximation des gradients. Mais l'heuristique permettant le calcul du gradient du simplexe approchant celui de la fonction objectif permet un très bon comportement de cette méthode pour les fonctions bruitées.

3.2 Modèles de la fonction objectif

L'optimisation à l'aide de modèles de substitution ("surrogate model") de la fonction objectif est un thème de recherche en pleine expansion. Celle-ci permet de déterminer un point minimisant la fonction objectif à partir d'une approximation de cette fonction. Cette idée semble intéressante dans le sens où elle remplace une évaluation de la fonction objectif par une évaluation moins coûteuse du modèle de substitution.

Depuis plus d'un demi-siècle (Box et Wilson, 1951 [21]), la solution généralement retenue pour diminuer le nombre des évaluations de la fonction objectif est la construction d'un modèle simplifié de celle-ci dont l'évaluation est peu coûteuse. Ce modèle, construit à partir d'un nombre limité d'évaluations choisies avec soin (on parle alors de plan d'expériences), va guider la recherche d'un minimum local ou global et du ou des minimiseur(s) associé(s). Pour éviter la confusion entre ce modèle et la fonction objectif d'autres terminologies ont été proposées. On parle ainsi de surfaces de réponse, de méta-modèles, d'émulateurs, de modèles boîte noire, de modélisations comportementales, ou encore de modèles de substitution ("'surrogate model"). C'est cette appellation que nous utiliserons par la suite.

Serafini [158] et Sondergaard [160] suggèrent de classer les modèles de substitution en deux catégories : les modèles physiques et les modèles fonctionnels.

Les modèles physiques sont construits à partir d'une simplification physique ou numérique de la fonction objectif, formant un modèle simple et peu coûteux $c(x)$. Il sont fondés sur la connaissance du système physique ou des phénomènes déjà modélisés. Il existe des méthodes permettant de construire un modèle physique à partir d'informations sur la fonction objectif :

- les méthodes de correction de réponse appelées "β-correlation Method" [82] et "general response correction" [160] construisent un modèle physique composé d'une fonction de correction et d'un modèle peu coûteux,
- la méthode multipoint [164] utilise une combinaison de modèles peu coûteux, les paramètres de cette combinaison sont déterminés par régression de moindres carrés aux points de la fonction objectif pré-échantillonnés,
- la méthode de "space-mapping" [6, 7, 8, 160] effectue une transformation de l'espace des paramètres de manière à ce que le modèle peu coûteux $c(x)$ montre le même comportement que la fonction objectif f.

Ainsi, dans les deux applications présentées dans le paragraphe 2 du chapitre 4, on pourrait construire des modèles simplifiés des simulateurs. Ceci fait l'objet de recherche en cours, mais ne sera pas abordé dans cette thèse.

Les modèles fonctionnels sont construits sans connaissance particulière du système physique. Ce sont des représentations algébriques du comportement de la fonction objectif basées sur les points pour lesquels la valeur de la fonction est connue. Nous verrons dans la suite qu'une grande diversité de ces modèles de substitution fonctionnels est disponible dans la littérature. Les méthodes les plus utilisées sont les modèles de régression paramétrique ou non-paramétrique, ainsi que les modèles simples polynomiaux locaux.

Dans la suite de ce manuscrit, les modèles fonctionnels seront utilisés : en effet l'objectif

de cette thèse est l'optimisation de modèle boîte noire sans aucun a priori sur la fonction objectif.

Nous exposons désormais les méthodes de résolution du problème d'optimisation 1.1 utilisant un modèle de substitution de la fonction objectif. Nous présentons tout d'abord l'approche utilisant des modèles de surface de réponse globale puis ensuite l'approche par région de confiance. Cette dernière considère le modèle de substitution valide uniquement localement dans une région de confiance.

Un large panel de modèles de substitution différents tels que les modèles de régression paramétrique et non-paramétrique ou les modèles locaux peuvent être utilisés. Les modèles nécessitent la définition suivante.

Définition 3.1 *Les modèles de régression sont basés sur des fonctions de base qui sont ajustées à des échantillons de données. Un modèle de régression, représenté par la fonction $m(x, \beta)$ (où β est le paramètre servant à décrire le modèle), s'ajuste à des échantillons de données $(x_i, y_i), i = 1, ..., p$ de la fonction coûteuse f. Les paramètres β peuvent être estimés en résolvant le problème de moindres carrés*

$$\min_{\hat{\beta}} \sum (y_i - m(x_i, \hat{\beta}))^2. \tag{1.5}$$

- **Modèle de régression paramétrique :** ce type de modèle est définit de manière explicite et le nombre p de fonction de base est fixé. Ce modèle peut prendre plusieurs formes dont le modèle de régression linéaire où $m(x, \beta) = \sum_{k=1}^{p} \beta_k m_k(x)$ est linéaire en β. Dans ce cas $m(x, \beta) = \beta^T v(x)$, où $v : \mathbb{R}^n \mapsto \mathbb{R}^u$, est un vecteur avec u fonctions de base : $v_j(x), j = 1, ..., u$.
  Les fonctions de base les plus utilisées pour une régression de moindres carrés linéaire sont les modèles polynomiaux du premier et du second ordre.
- **Modèle de régression non paramétrique :** cette approche ne nécessite aucune hypothèse a priori sur la forme du modèle utilisé. La forme la plus utilisée et la plus efficace de modèle de régression non paramétrique est la régression avec méthode par lissage à noyau (Kernel regression).
  Les méthodes de ce type les plus utilisées sont les RBF (Radial Basis Function), le krigeage, les SVM (Support Vector Machine), les splines, les ondelettes [165] et certains réseaux de neurones.
- **Modèle local :** cette classe de modèle est basée sur l'information obtenue au voisinage d'un point, usuellement l'itération courante x_k d'une procédure d'optimisation. Les modèles de Taylor appartiennent à cette classe. Toutefois les modèles de Taylor d'ordre faible ne sont valables comme approximation de f que dans une petite région limitée autour de x_k.
  Le modèle de Taylor du premier ordre ou du second ordre pourrait être utilisé dans une région autour de x_k si les dérivées de f étaient disponibles (ex : méthode de Newton). Dans le cas contraire, un modèle m_k de la fonction objectif peut être déterminé par un schéma d'interpolation multivariée utilisant uniquement les valeurs

de la fonction objectif en plusieurs points d'interpolation distincts. Les valeurs de fonction objectif disponibles $f(x_i)$ qui correspondent aux x_i, ($i < k$ évaluations) les plus proches de x_k sont donc utilisées pour interpoler la fonction objectif. Ce modèle ainsi créé peut être un polynôme linéaire ou un polynôme quadratique valide dans un voisinage de l'itération courante appelé région de confiance.

3.2.1 Optimisation à l'aide de surfaces de réponses globales

Ces méthodes reposent sur une procédure séquentielle d'expérimentation du domaine de définition de la fonction dans le but d'optimiser la fonction objectif. Elles nécessitent la construction d'un plan d'expérience, de méthodes de modélisation par régression ou interpolation (Box et Wilson [21] et Myers et Montgomery [123]) et d'un critère d'échantillonage permettant l'optimisation de la fonction objectif à chaque itération. En effet, le principe des méthodes d'optimisation utilisant un modèle global de la fonction objectif est d'évaluer cette fonction objectif itérativement en un ou plusieurs points obtenus à l'aide du critère d'échantillonnage appliqué au modèle de substitution courant de la fonction objectif.

Algorithme 1.6 Optimisation à l'aide de surface de réponse

- **Initialisation** : itération $i = 1$. Sélectionner l'ensemble fini $S_1 = \{x_1, x_2, ..., x_k\}$ de k points initiaux répartis dans le domaine de définition de la fonction objectif. Evaluer la fonction objectif aux points initiaux de S_1 et sélectionner f_{min} la meilleure valeur et x_{min} le point correspondant.
- Répéter tant que le critère d'arrêt n'est pas atteint.
  (1) **Créer ou mettre à jour le modèle de substitution** : la surface de réponse m_i est construite en utilisant les points S_i.
  (2) Recherche du point ou des nouveaux points à évaluer en fonction du critère d'échantillonnage.
  (3) Ajout de ce(s) point(s) au plan d'expérience courant et évaluation de la fonction objectif en ces points. Mise à jour de f_{min} et x_{min}.

x_{min} est le meilleur point calculé.

L'initialisation de l'algorithme 1.6, c'est-à-dire la sélection de k points bien répartis dans l'espace, peut être réalisée à partir d'un plan d'expérience de type Hypercube Latin [117]. Le critère d'arrêt utilisé est en général un nombre d'évaluation de la fonction objectif maximum.

Un modèle de substitution, construit à partir d'évaluations de f disponibles, peut être utilisé pour construire un critère quantifiant l'intérêt d'une évaluation potentielle. Plus précisément, c'est le résultat de l'optimisation de ce critère (que l'on nommera critère d'échantillonnage), dépendant des résultats des évaluations précédentes, qui détermine le prochain point de l'espace de recherche où l'évaluation doit être effectuée. On remplace donc un problème d'optimisation coûteux par une série de problèmes d'optimisation moins

coûteux. Pour cette méthode, le critère d'échantillonnage est classiquement le minimiseur de la surface de réponse globale.

Nous présentons maintenant différentes surfaces de réponses utilisés classiquement pour l'optimisation sans dérivées. Ces méthodes ont des propriétés propres permettant l'utilisation de différents critères d'échantillonnages.

3.2.1.1 Krigeage Le krigeage (présenté en annexe D) est la méthode de base la plus populaire pour l'optimisation globale basée sur des méthodes de surfaces de réponses. En effet, le krigeage est basé sur une approche probabiliste qui permet d'estimer les incertitudes sur le modèle de substitution de la fonction objectif. Le krigeage (Krige, 1951) est une méthode (ou plutôt un ensemble de méthodes) de prédiction reposant sur un modèle probabiliste (généralement gaussien) qui peut être utilisé pour interpoler des données. Le terme krigeage (kriging) provient du nom de famille de l'ingénieur minier sud africain Daniel Gerhardus Krige à l'origine de la technique, mais la formalisation mathématique revient à George Matheron en 1963. Depuis les années 60, le krigeage, qui n'est au départ qu'une prédiction linéaire, a été largement développé et utilisé en géostatistiques (dans des espaces 3D).

Ces modèles possèdent plusieurs avantages qui les rendent particulièrement adaptés à l'optimisation globale de fonctions coûteuses. Tout d'abord, ils permettent la mise en place d'un cadre bayésien qui facilite le traitement des incertitudes en entrée et en sortie du système ainsi que la prise en compte d'a priori sur la fonction inconnue. Enfin, ils permettent l'estimation de l'erreur d'approximation à partir de laquelle des critères d'échantillonnage pertinents pour l'optimisation globale (Jones, 1998 [100]) peuvent être construits.

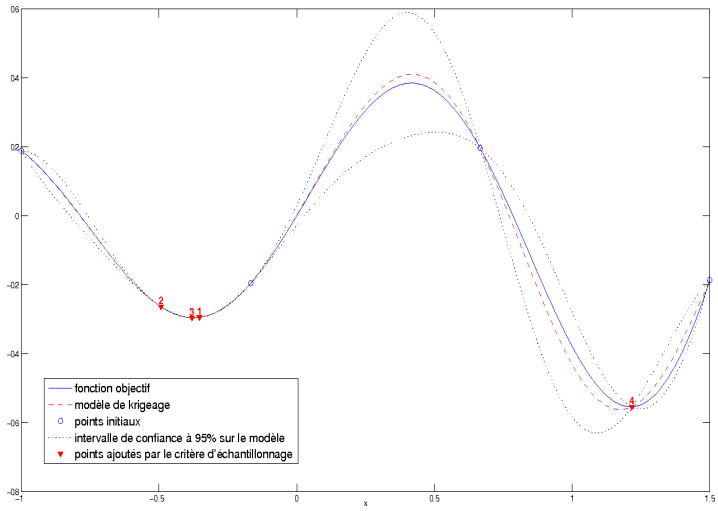


FIGURE 1.5 – Krigeage avec ajout de point par espérance de gain

Le critère d'échantillonnage classiquement utilisé est la maximisation de l'espérance

de gain (Expected Improvement [156, 155, 100, 99]) (notons qu'il en existe de nombreux autres, voir l'annexe D). Ce critère permet de réaliser un compromis entre exploration et recherche locale. L'algorithme EGO (Efficient Global Optimization), basé sur l'algorithme 1.6, proposé par Jones et al.([100] 1998), choisit comme nouveau point d'évaluation un maximiseur d'espérance de gain, et met à jour les paramètres du modèle de krigeage après chaque nouvelle évaluation. Notons qu'une recherche globale du point d'espérance de gain maximale est effectuée à chaque itération, ce qui est un problème d'optimisation difficile en grande dimension. La figure 1.5 représente la construction d'un modèle de krigeage avec ajout de point par espérance de gain pour l'optimisation sans dérivées d'une fonction.

Cette approche a permis de résoudre de nombreux problèmes, tels que l'optimisation de la forme d'un conduit d'admission ainsi que l'optimisation du contrôle de la direction assistée électrique [172]. Mais également de résoudre des problèmes de conception de pale d'un rotor d'hélicoptère [18], des problèmes de conception de plateforme de production de véhicule, ainsi qu'un problème d'effort de portée [18]. Plus récemment, une application en ingénierie de réservoir a été traitée par Guimaraes [52].

Une adaptation de cette méthode a également été effectuée par Huang [91] pour une meilleure prise en compte du bruit de la fonction objectif. Knowles [106] a quand à lui adapté la méthode de Jones pour l'optimisation multi-objectifs.

3.2.1.2 Autres surfaces de réponses D'autres surfaces de réponses ont été utilisées pour l'optimisation sans dérivées comme par exemple les Radial Basis Functions (RBF) [92, 93, 16]. Gutmann [80, 81] utilise également des RBF comme surface de réponse, avec un critère d'échantillonnage consistant à sélectionner le minimum global de la surface de réponse courante ce qui peut faire converger la méthode vers un minimum local. Regis [146, 147, 148, 149] recherche un compromis entre recherche locale et recherche globale en pondérant la minimisation globale de la surface de réponse par la distance entre les points précédemment évalués. De cette façon, les bassins avec de petites valeurs de fonction ayant déjà été visités de nombreuses fois ne sont pas visités aussi souvent que les zones très peu visitées. Cela assure un équilibre entre l'amélioration de la valeur minimale courante de la fonction et les incertitudes sur le modèle. Ces deux méthodes permettent également de prendre en compte des contraintes non linéaires, celles-ci sont ainsi respectées lors de la maximisation/minimisation du critère d'échantillonnage. Käck [102] propose une nouvelle technique de prise en compte des contraintes que Rudholm [152] puis Jakobsson et al [94, 95] ont adaptée pour l'optimisation multi-objectifs.

Les réseaux de neurones ont également été utilisés pour l'optimisation sans dérivées [170], avec une application dans l'industrie automobile sur l'optimisation de la forme de pare-brise.

D'après les résultats de la thèse [22], on peut envisager d'utiliser les Sparse Grid dans le cadre de l'optimisation. Ceci fait l'étude de travaux en cours à IFPEN par E. Echagüe.

L'optimisation à partir de surfaces de réponses globales permet d'approcher le minimum global de la fonction objectif, mais ces méthodes sont difficiles à utiliser en grande dimension du fait d'une part de l'optimisation du critère d'échantillonnage mais également

de la construction de la surface de réponse elle-même. Le krigeage, de sa formation probabiliste permettant d'estimer les incertitudes sur le modèle de substitution de la fonction objectif, semble être une très bonne alternative pour les problèmes en faible dimension.

Une autre difficulté de ces méthodes est la prise en compte de contraintes autres que des contraintes de borne.

3.2.2 Approche par région de confiance

Les méthodes classiques de région de confiance font partie des algorithmes les plus étudiés pour les problèmes de programmation non linéaire. Ces méthodes ont de nombreux avantages : le fait qu'elles soient basées sur des modèles quadratiques les rendent particulièrement attractives pour traiter les informations sur la courbure de la fonction. Leur robustesse est en partie liée à l'effet de régularisation obtenu lors de la minimisation de modèles quadratiques dans une région de confiance [29].

Algorithme 1.7 Modèles locaux valables dans une région de confiance

- **Initialisation** : soit $0 < \nu_0 \leq \nu_1$ et $0 < \gamma_0 < 1 < \gamma_1$. Choix d'une classe de modèle et d'un algorithme d'amélioration de modèle. Choix du point de départ x_0 et du rayon de la région de confiance initial Δ_0 et $\Delta_{min} < \Delta_{max}$ le rayon minimal et maximal de la région de confiance. $k = 0$. Calcul de $f(x_0)$ et construction du modèle initial m_0.
- Tant que $\Delta \geq \Delta_{min}$:

  (1) **Minimisation du modèle** : calculer le pas d_k qui réduit suffisamment le modèle m_k et tel que $x_k + d_k \in B(x_k; \Delta_k)$ (la boule de rayon Δ_k et de centre x_k).

  (2) **Approbation du nouveau point** : évaluer $f(x_k + d_k)$ et

  $$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}.$$

  Si $\rho_k \geq \nu_1$, $x_{k+1} = x_k + d_k$ et mise à jour du modèle m_{k+1}, sinon $x_{k+1} = x_k$.

  (3) **Amélioration du modèle** : si $\rho_k < \nu_1$, le modèle est amélioré à l'aide de nouvelles évaluations de la fonction objectif. La meilleure de ces évaluations est notée $\hat{x}_k$. Calculer

  $$\hat{\rho}_k = \frac{f(x_k) - f(\hat{x}_k)}{m_k(x_k) - m_k(\hat{x}_k)}.$$

  Si $\hat{\rho}_k \geq \nu_0$, $x_{k+1} = \hat{x}_k$ et mise à jour du modèle m_{k+1}, sinon $x_{k+1} = x_k$.

  (4) **Mise à jour du rayon de la région de confiance Δ_k** :

  $$\Delta_{k+1} \in \begin{cases} [\gamma_1 \Delta_k, \Delta_{max}), & \text{si } \rho_k \geq \nu_1, \\ \Delta_k, & \text{si } \rho_k \in [\nu_0, \nu_1), \\ \gamma_0 \Delta_k, & \text{si } \rho_k < \nu_0. \end{cases}$$

  $k = k + 1$.

Ces méthodes classiques ont été adaptées pour l'optimisation sans dérivées. Elles main-

tiennent un modèle simple (linéaire, quadratique ou RBF) basé uniquement sur les valeurs de la fonction objectif calculées aux points échantillonnés. Les modèles correspondants peuvent être construits à partir de polynômes d'interpolation ou de régression ou par un autre procédé d'approximation. Celui-ci est utilisé au cours des itérations pour diriger l'optimisation vers un minimum local dans une région de confiance. Pour ce faire, une minimisation de ce modèle est effectuée sous une contrainte de région de confiance. Le point ainsi sélectionné comme étant le possible minimum local de la fonction est évalué et le rapport

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}$$

est calculé. Ce paramètre nous indique le degré d'adéquation du modèle à la fonction objectif, il est utilisé pour mettre à jour le rayon de la région de confiance. Lorsque la précision du modèle approchant la fonction objectif est insuffisante, une étape d'amélioration du modèle est nécessaire. L'algorithme 1.7 représente simplement cette approche qui sera plus longuement détaillée dans les chapitres 2 et 3.

Ces méthodes ont été utilisées sur des problèmes de quelques centaines de paramètres [139, 140]. Certaines adaptations ont permis de prendre en compte des contraintes [133, 31, 32, 29, 167, 169, 168, 142].

4 Méthodes hybrides

De nombreuses méthodes d'optimisation sans dérivées hybrides sont apparues ces dernières années.

Ainsi Booker et al [18] introduisent l'utilisation d'un modèle de substitution dans le contexte d'un algorithme de recherche directe directionnelle (voir le paragraphe 1.2.1 de ce chapitre). L'idée est basée sur l'étape de recherche de l'algorithme 1.2. De la même manière, Conn et al [30], dans un algorithme de recherche directe, tire parti plus particulièrement d'un modèle quadratique pour l'optimisation sous contraintes sans dérivées. Le modèle de substitution $m(x)$ de la fonction $f(x)$ peut être construit et recalibré au cours des itérations de la méthode. Nous décrivons cette méthode dans l'algorithme 1.8, en donnant deux alternatives possibles pour la prise en compte du modèle quadratique dans l'étape de recherche. L'étape de sondage peut elle aussi bénéficier du modèle de substitution en ordonnant les évaluations de la fonction objectif f dans $P_k = \{x_k + \alpha_k d / d \in \mathbb{D}_k\}$ suivant les valeurs du modèle de substitution $m(x)$.

Une autre adaptation de la méthode de recherche directe directionnelle (voir le paragraphe 1.2.1 de ce chapitre) utilisant cette fois-ci une métaheuristique des essaims particulaires dans l'étape de recherche de l'algorithme 1.2 est présentée dans [171].

La méthode des colonies de fourmis a a été associée à une recherche locale de Nelder-Mead pour l'optimisation en variables continues [54], une application en génie biomédical a ainsi été traitée.

Algorithme 1.8 Méthode de recherche directe directionnelle hybride

- **Initialisation** : choix de $x_0, \alpha_0 > 0, 0 < \beta_1 \leq \beta_2 < 1$ et $\gamma \geq 1$. Soit $\mathbb{D}$ un ensemble formant une base positive.
- Tant que le diamètre $\alpha_k \geq \alpha_{tol}$:
  (1) **Etape de recherche** : essayer de calculer un point x qui vérifie $f(x) < f(x_k)$ en évaluant la fonction objectif en un nombre fini de points par l'un des procédés suivants :
    (a) Minimiser le modèle $m(x)$ dans un ensemble faisable. Soit y_k la solution de la minimisation. Evaluer $f(y_k)$.
    (b) Evaluer $m(.)$ aux points de l'ensemble $Y_k = \{y_k^1, ..., y_k^p\}$. Ordonner Y_k tel que $m(y_k^1) \leq ... \leq m(y_k^p)$. Evaluer f aux points de Y_k dans l'ordre.

  Si un point x est trouvé, $x_{k+1} = x$, l'itération et l'étape de recherche sont réussies et l'étape de sondage est passée.
  (2) **Etape de sondage** : choisir une base positive $\mathbb{D}_k$ à partir de l'ensemble $\mathbb{D}$. Ranger l'ensemble des points du sondage $P_k = \{x_k + \alpha_k d / d \in \mathbb{D}_k\}$. Démarrer les évaluations de f aux points de sondage suivant l'ordre choisi. Si un point sondé $x_k + \alpha_k d_k$ vérifie $f(x_k + \alpha_k d_k) < f(x_k)$, le sondage est stoppé, $x_{k+1} = x_k + \alpha_k d_k$, l'itération et l'étape de sondage sont réussies. Sinon, l'itération (et l'étape de sondage) échouent et $x_{k+1} = x_k$.
  (3) **Mise à jour du maillage** : si l'itération est réussie, le pas du maillage est maintenu ou augmenté : $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$. Autrement, réduire le pas du maillage : $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

De la même manière, une utilisation des essaims particulaires associée à la méthode de Nelder-Mead pour l'optimisation globale d'une fonction non linéaire est présentée dans [107, 59, 184]. Dans [59], cette méthode hybride a été comparée avec une autre méthode hybride couplant un algorithme génétique avec la méthode de Nelder-Mead avec des résultats similaires.

Des méthodes hybrides liant les avantages des algorithmes évolutionnaires et de modèles de substitution ont également été utilisés pour résoudre des problèmes industriels (dans les domaines médicales ou de l'automobiles avec les RBF [55, 56]). Une seconde méthode hybride permettant d'alterner entre recherche globale et recherche locale à l'aide d'un algorithme évolutionnaire global (algorithme génétique, stratégie d'évolution,...) et d'une méthode classique locale (direction de plus grande descente ou BFGS) est également employée dans [55].

5 Discussion

Le tableau 1.1 récapitule différents solveurs d'optimisation sans dérivées classés suivant les différents types de méthodes présentées précédemment. Nous incitons le lecteur à

visiter le site "NEOS tree" (`http://neos-server.org/neos/`) pour une liste de solveurs d'optimisation sans dérivées.

Table 1.1 – Tableau récapitulatif des différents solveurs cités

Classes	Noms	Commentaires
Recherche directe	NMSMAX	Méthode de Nelder-Mead
	APPSPACK	Pattern Search parallèle
	NOMAD	Pattern Search
Métaheuristiques	CMAES	Stratégie d'évolution
Optim. à base de modèle de ∇f	SQPAL	SQP avec gradients calculés par différences finies et approximation BFGS de la Hessienne
	Implicit Filtering	
Optim. à base de modèle de f	EGO	krigeage avec ajout de point par espérance de gain sans contraintes
	COBYLA	modèle linéaire local valable dans une région de confiance avec contraintes "derivative free"
	DFO	modèle quadratique local valable dans une région de confiance avec contraintes "derivative based" et "derivative free"
	UOBYQA	modèle quadratique complet local valable dans une région de confiance sans contraintes
	CONDOR	modèle quadratique complet local valable dans une région de confiance avec contraintes "derivative based"
	WEDGE	modèle linéaire ou quadratique complet local valable dans une région de confiance sans contrainte
	NEWUOA	modèle quadratique local valable dans une région de confiance sans contraintes
	BOBYQA	modèle quadratique local valable dans une région de confiance avec contraintes de bornes
	BOOSTERS	modèle RBF local valable dans une région de confiance sans contraintes
	ORBIT	modèle RBF local valable dans une région de confiance sans contraintes
	SQA	modèle quadratique local valable dans une région de confiance avec contraintes "derivative based" et "derivative free"

Les méthodes directes, telles que la méthode du simplexe de Nelder-Mead, la méthode de pattern search et la méthode de recherche multi-directionnelle n'utilisent pas toutes les informations obtenues lors des itérations précédentes. L'évaluation de la fonction objectif étant coûteuse, il semble logique d'utiliser au maximum toutes les informations connues sur cette fonction objectif. En revanche, la parallélisation de ces méthodes est naturelle. La méthode des directions conjuguées utilise l'information obtenue lors des évaluations successives de la fonction objectif, mais la construction d'un modèle de substitution de la fonction et en particulier l'évolution de sa courbure nous paraît être une meilleure alternative.

Les métaheuristiques permettent l'optimisation globale de la fonction objectif, et peuvent facilement être parallélisées. Cependant, le nombre d'évaluations élevé nécessaire à leur fonctionnement devient vite problématique lorsque la dimension du problème augmente.

Les modèles de substitution des gradients de la fonction objectif ou de la fonction objectif elle-même permettent d'utiliser au maximum les renseignements disponibles sur la fonction objectif : l'utilisation de modèles de substitution des gradients de la fonction permet l'utilisation de méthodes classiques d'optimisation différentiables. Les méthodes utilisant des gradients approchés par différences finies sont difficiles à paramétrer, le choix du pas de différences finies est très sensible au bruit sur la fonction objectif. Les méthodes de type Implicit Filtering, approchant le gradient à l'aide des évaluations précédentes, se comportent mieux vis-à-vis du bruit (effet régularisant) et sont moins coûteuses en évaluation de fonction, cependant l'approximation du gradient est moins précise et cela peut entraîner un échec de la recherche linéaire.

L'optimisation à l'aide de modèles de substitution de la fonction objectif permet de déterminer un point minimisant la fonction à partir de ce modèle de substitution. Cette idée semble intéressante dans le sens où elle remplace une évaluation de la fonction objectif par une évaluation moins coûteuse du modèle de substitution. Cette méthode permet de tirer un maximum d'informations sur la fonction au cours de chaque itération. L'optimisation à partir de surfaces de réponses globales permet d'obtenir le minimum global de la fonction objectif, mais ces méthodes sont difficiles à utiliser en grande dimension du fait d'une part de l'optimisation du critère d'échantillonnage mais également de la construction de la surface de réponse elle-même. L'optimisation locale à partir de modèles simples de la fonction objectif permet d'utiliser toutes les informations connues sur la fonction objectif, ces méthodes fonctionnent sur des problèmes d'une centaine de paramètres, et la prise en compte de contraintes sur la fonction objectif semble a priori réalisable.

Dans la suite de ce mémoire, nous nous intéresserons à des modèles de substitution polynômiaux qui approchent la fonction objectif dans une région de confiance c'est à dire un voisinage convenable du minimum courant. Ces modèles sont particulièrement bien adaptés à l'optimisation locale d'une fonction coûteuse à évaluer sous contrainte pour des problèmes de dimension avoisinant une centaine de paramètres.

Chapitre 2

Optimisation sans dérivées : méthodes de région de confiance

Winfield (1969 [181, 182]) propose une première approche utilisant les valeurs de la fonction objectif $f(y^i)$ disponibles afin de construire un modèle quadratique interpolant. Ce modèle sera supposé valide dans un voisinage du point courant qui est décrit comme une région de confiance dont le rayon est ajusté au cours des itérations. Le modèle est minimisé dans cette région de confiance et donne un point qui réduit probablement la valeur de la fonction. Lorsque de nombreuses valeurs de la fonction objectif sont disponibles, l'ensemble des points définissant le modèle d'interpolation est mis à jour afin de ne contenir que les points les plus proches du point courant. Powell (1994 [133]) réutilise cette idée et propose une méthode (nommée COBYLA) permettant l'optimisation d'une fonction sous contraintes utilisant un modèle d'interpolation linéaire de la fonction objectif et des contraintes. En développant cette idée, Powell (1994 [134, 136]) propose une méthode similaire à celle de Winfield permettant l'optimisation sans contrainte utilisant un modèle d'interpolation quadratique de la fonction objectif construite à l'aide de polynômes fondamentaux de Lagrange. La disparité entre ces deux méthodes vient de la mise à jour de l'ensemble des points d'interpolation qui préserve certaines propriétés géométriques [1]. Pour ce faire, une relation entre les polynômes fondamentaux de Lagrange associés au problème d'interpolation (permettant de construire le modèle d'interpolation) et les coordonnées des points d'interpolation dans la base des polynômes est utilisée.

Une variante de ces méthodes d'interpolation quadratique utilisant les polynômes fondamentaux de Newton a été présentée par Conn et Toint [35]. Un résultat de convergence globale vers des points critiques du premier ordre pour les méthodes de ce type a été donné par Conn, Scheinberg et Toint [31, 29], ainsi qu'une description des techniques pour faire respecter des propriétés géométriques à l'ensemble des points d'interpolation. Ces tech-

1. Ces propriétés sont nécessaires pour garantir l'existence et l'unicité du modèle interpolant. En effet, en dimension deux, six points alignés ou sur un cercle ne permettent pas de définir un polynôme d'interpolation quadratique complet.

niques sont basées sur l'exploitation de la théorie d'interpolation de Newton. Conn, Gould et Toint [29] et Conn, Scheinberg et Vicente [34] présentent une extension de ce résultat à une convergence du second ordre.

La méthode de région de confiance basée sur un modèle d'interpolation quadratique de la fonction objectif, utilisant des polynômes fondamentaux de Newton, de Scheinberg (appelée DFO [31, 32, 29]) permet la prise en compte de contraintes à l'aide d'une méthode d'optimisation non linéaire de type SQP par exemple (sequential quadratic programming, voir annexe C). Powell propose UOBYQA (2002 [137]) une méthode d'optimisation sans contrainte par région de confiance basée sur la théorie qu'il avait exposé peu après Winfield. Cette méthode a été adaptée par Vanden Berghen ([167, 169, 168]) afin de prendre en compte des contraintes également au moyen d'un SQP dans le code CONDOR. Marazzi et Nocedal développent WEDGE [116] une méthode de région de confiance utilisant soit des modèles linéaires, soit des modèles quadratiques. Colson [26, 27] développe quant à lui une technique de filtre permettant d'éviter l'utilisation d'une fonction de mérite et la gestion de poids de pénalisation pour la prise en compte de contraintes. Powell propose une méthode d'optimisation sans contrainte par région de confiance basée sur un modèle d'interpolation quadratique sous déterminé (2006 NEWUOA [139, 140]) : en effet celui-ci est construit à partir d'un nombre de points d'interpolation $m \in [n+2, \frac{1}{2}(n+1)(n+2)]$, n étant le nombre de paramètres du problème d'optimisation. Les degrés de liberté du modèle d'interpolation restants sont obtenus en utilisant la méthode de Broyden symétrique pour mettre à jour la Hessienne du modèle quadratique qui consiste à minimiser l'évolution de la courbure du modèle au cours des itérations. Puis Powell suggère de prendre en compte les contraintes de bornes (2009 BOBYQA [142]) dans une méthode similaire à NEWUOA. Une procédure de gradient conjugué tronqué permet la prise en compte des contraintes lors du choix de chaque nouvelle évaluation. Enfin Powell [143] présente la convergence de ses méthodes. Gratton et al [78] proposent une technique d'"active set" permettant une prise en compte des contraintes de bornes efficaces.

Les approximations quadratiques de la fonction objectif sont très utilisées pour accélérer la convergence des algorithmes d'optimisation sans dérivées, la courbure de la fonction objectif étant ainsi prise en compte. Ces méthodes ont été évaluées en pratique sur des problèmes de conception de pale d'un rotor d'hélicoptère [18], ainsi que sur un problème de conception de turbine [1, 166].

Alexandrov et al puis Conn et al [2, 29, 34] démontrent que l'utilisation d'un modèle linéaire ou quadratique n'est pas l'unique choix pour le modèle. D'ailleurs, R. Oeuvray [126] utilise un modèle constitué de RBF (Radial Basis Functions) valable dans la région de confiance ; celui-ci est construit à partir d'un nombre de points d'interpolation fixé au démarrage de la méthode. La méthode d'optimisation sans contrainte d'Oeuvray, BOOSTERS, a également permis de traiter une application en imagerie médicale, et est adaptée pour prendre en compte une information sur le gradient lorsque celle-ci est disponible. Wild [178, 177] propose une méthode par région de confiance pour l'optimisation sans contrainte : ORBIT. Celle-ci permet d'utiliser un modèle RBF ou un modèle quadratique sous déterminé. L'avantage de cette méthode est qu'elle ne maintient pas un

nombre de points d'interpolation fixe au cours des itérations ; celui-ci dépend du nombre de points disponibles suffisamment proches de l'optimum courant. Wild [179] démontre la convergence globale vers un point critique du premier ordre de cette méthode. Celle-ci a permis de résoudre deux problèmes environnementaux : la calibration d'un modèle de simulation de bassin et l'optimisation de la biorestauration des eaux souterraines.

Ces méthodes de région de confiance pour l'optimisation sans dérivées maintiennent un modèle (linéaire, quadratique ou de RBF) qui est basé seulement sur les évaluations de la fonction objectif calculées aux points échantillonnés. Les modèles correspondants peuvent être construits au moyen de polynômes d'interpolation ou de régression ou par une autre technique d'approximation. Il n'est pas requis que les modèles soient des fonctions polynomiales tant qu'une réduction peut-être extraite du sous problème de région de confiance. Pour ce faire, le modèle doit avoir un comportement local uniforme (parfois après un nombre fini de modifications de l'ensemble échantillonné) similaire à ce qu'on pourrait observer avec des modèles de Taylor quand on a accès aux dérivées.

Nous présentons tout d'abord la méthode classique d'optimisation dans une région de confiance utilisée pour l'optimisation différentiable qui a inspiré les méthodes d'optimisation par région de confiance. Ensuite les différents modèles utilisés par ces méthodes sont présentés avec leurs propriétés. Le processus permettant de maintenir la qualité des modèles nécessaire au cours des itérations de la méthode d'optimisation sans dérivées est décrit par la suite. Puis nous terminons par présenter l'algorithme de cette méthode d'optimisation sans dérivées basée sur la méthode de région de confiance.

1 Optimisation avec dérivées dans une région de confiance

Nous sommes intéressés, ici, par la minimisation d'une fonction objectif sans contrainte :

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.1}$$

où $f(x)$ est une fonction à valeurs dans $\mathbb{R}$.

A chaque itération k, nous définissons le modèle quadratique, $Q_k(x)$, qui approche la fonction objectif dans un voisinage convenable du point courant x_k, appelé région de confiance.

$$Q_k(x_k + d) = Q_k(x_k) + \langle g_k, d \rangle + \tfrac{1}{2}\langle d, H_k d \rangle$$

où $Q_k(x_k) = f(x_k)$, $g_k = \nabla_x f(x_k)$ et H_k est une approximation symétrique de $\nabla_{xx} f(x_k)$. La région de confiance est l'ensemble des points tels que : $\mathcal{B}_k = \{x \in \mathbb{R}^n, \|x - x_k\|_k \leq \Delta_k\}$, où Δ_k est le rayon de la région de confiance et $\|\cdot\|_k$ est la norme à l'itération k à définir (généralement les normes ℓ_1, ℓ_2, ℓ_∞ ou une norme variant avec les itérations, par exemple la norme associée à la matrice Hessienne de la fonction objectif).

Nous cherchons ensuite le pas d'essai d_k qui donne le point $x_k + d_k$ afin de réduire le modèle tout en vérifiant la contrainte : $\|d_k\|_k \leq \Delta_k$. Ce choix de d_k est alors le résultat

d'un problème de minimisation du modèle quadratique sous contrainte. Après avoir calculé le pas d'essai, la fonction objectif est évaluée en $x_k + d_k$ et comparée à la valeur prédite par le modèle en ce point : $Q_k(x_k + d_k)$. Si une réduction, prédite par le modèle, a été réalisée par la fonction objectif, le point d'essai est accepté pour la prochaine itération et le rayon de la région de confiance est augmenté ou reste inchangé. Si la réduction du modèle s'avère être une mauvaise prédiction du comportement de la fonction objectif, le point d'essai est rejeté et la région de confiance est rétrécie afin de fournir une meilleure prédiction dans une région plus petite.

L'algorithme 2.1 proposé par Conn, Gould et Toint [29] est décrit ci-dessous :

Algorithme 2.1 Algorithme de région de confiance classique

(1) **Initialisation** : le point initial x_0 et le rayon initial de la région de confiance Δ_0 sont donnés. Les constantes η_1, η_2, γ_1 et γ_2 sont aussi données et satisfont : $0 < \eta_1 \leq \eta_2 < 1$ et $0 < \gamma_1 \leq \gamma_2 < 1$. Calcul de $f(x_0)$ et $k = 0$.

(2) Itérations à région de confiance fixée

(a) Définition du modèle : choix de la norme $\|\cdot\|_k$ et définition du modèle Q_k dans $\mathcal{B}_k$.

(b) Calcul du pas d_k qui minimise le modèle Q_k et tel que $x_k + d_k \in \mathcal{B}_k$.

(c) Validation du point d'essai : évaluation de $f(x_k + d_k)$ et de
$R_k = \frac{f(x_k)-f(x_k+d_k)}{Q_k(x_k)-Q_k(x_k+d_k)}$.
Si $R_k \geq \eta_1 \Rightarrow x_{k+1} = x_k + d_k$, sinon $x_{k+1} = x_k$

(d) Mise à jour du rayon de la taille de région de confiance :

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty), & \text{si } R_k \geq \eta_2, \\ [\gamma_2\Delta_k, \Delta_k], & \text{si } R_k \in [\eta_1, \eta_2), \\ [\gamma_1\Delta_k, \gamma_2\Delta_k], & \text{si } R_k < \eta_1. \end{cases}$$

$k = k + 1$ et on retourne au début de l'étape (2).

Le critère d'arrêt de l'algorithme 2.1 peut-être une mesure sur le gradient de la fonction objectif en x_k, $\|\nabla_x f(x_k)\|$. Un aspect important des méthodes de région de confiance est la forme de la région de confiance elle-même, qui est déterminée par la norme utilisée dans l'étape (2)(a) (typiquement la norme associée à la matrice Hessienne de la fonction objectif). Idéalement, cette forme de région de confiance devrait refléter la région où le modèle approche au mieux la fonction objectif qui correspond à la région dans laquelle la fraction $R(d) = \frac{f(x_k)-f(x_k+d)}{Q_k(x_k)-Q_k(x_k+d)}$ est proche de 1.

2 Modèles de la fonction objectif

Des modèles d'interpolation ou de régression polynomiaux d'ordre élevé ou des modèles RBF quelconques peuvent être utilisés par ces méthodes. Mais en pratique, les trois différents types de modèles utilisés dans la littérature sont les modèles d'interpolation linéaire, quadratique et les modèles RBF cubiques.

En général, les modèles d'interpolation utilisés pour l'optimisation sans dérivées font partie de l'espace $\mathcal{P}_n^d$ des polynômes de degré inférieur ou égal à d dans $\mathbb{R}^n$.

Nous présentons quelques propriétés de ces modèles pour l'interpolation. Notons que des propriétés équivalentes existent dans le cas de la régression (voir [28]).

2.1 Modèle linéaire

L'intérêt envers les bases positives fut récemment relancé par l'introduction et l'analyse de classes d'algorithmes de recherche directe en optimisation. Celles-ci permettent également la construction de modèle linéaire pour les méthodes de région de confiance [133].

Définition 2.1 *Une* ***base positive*** *est un ensemble de vecteurs minimal dont les combinaisons linéaires positives engendrent l'espace* $\mathbb{R}^n$ *en entier.*

On peut aisément démontrer que la cardinalité de chaque base positive est bornée inférieurement par $n+1$ (nombre de termes d'un polynôme de degré 1 complet) et supérieurement par $2n$.

Soit $Y = \{y^0, y^1, ..., y^n\} \in \mathbb{R}^n$. Soit $m(x)$, un polynôme de degré $d = 1$ interpolant la fonction objectif f aux points de Y, c'est à dire satisfaisant les conditions d'interpolation

$$m(y^i) = f(y^i),\ i = 0, ..., n. \tag{2.2}$$

Nous pouvons écrire $m(x) = \alpha_0 + \alpha_1 x_1 + ... + \alpha_n x_n$, en utilisant la base de l'espace $\mathcal{P}_n^1$ des polynômes de degré 1 : $\bar{\phi} = 1, x_1, ..., x_n$. Nous pouvons réécrire 2.2 comme

$$\begin{bmatrix} 1 & y_1^0 & \cdots & y_n^0 \\ 1 & y_1^0 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^n & \cdots & y_n^n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^n) \end{bmatrix}.$$

On note M, la matrice de ce système linéaire :

$$M = M(\bar{\phi}, Y) = \begin{bmatrix} 1 & y_1^0 & \cdots & y_n^0 \\ 1 & y_1^0 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^n & \cdots & y_n^n \end{bmatrix}. \tag{2.3}$$

Définition 2.2 *L'ensemble* $Y = \{y^0, y^1, ..., y^n\}$*, est* ***unisolvant*** *("poised")* ***pour l'interpolation linéaire dans*** $\mathbb{R}^n$ *si la matrice correspondante* $M(\bar{\phi}, Y)$ *est inversible.*

Soit $B(y^0; \Delta(Y))$ la plus petite boule fermée centrée en y^0 de rayon $\Delta = \Delta(Y) = \max_{1 \le i \le p} \|y^i - y^0\|$ contenant Y.

Hypothèses 2.3 *Supposons que* $Y = \{y^0, y^1, ..., y^n\} \subset \mathbb{R}^n$ *est un ensemble unisolvant de points au sens de l'interpolation linéaire, contenu dans la boule* $B(y^0; \Delta(Y))$ *de rayon* $\Delta = \Delta(Y)$ *(le diamètre de la plus petite boule* $B(Y)$ *contenant* Y*). Et la fonction* f *est continûment différentiable dans un domaine ouvert* Ω *contenant* $B(y^0; \Delta)$ *et* ∇f *est Lipschitz continu dans* Ω *avec la constante* $\nu > 0$*.*

Par la méthode d'élimination de Gauss-Jordan, on peut réduire la matrice $M = M(\bar{\phi}, Y)$ à

$$\begin{bmatrix} 1 & y_0^\top \\ 0 & L \end{bmatrix}, \tag{2.4}$$

avec

$$L = [y^1 - y^0 \cdots y^n - y^0]^\top.$$

Théorème 2.4 *Supposons l'hypothèse 2.3. Le gradient du modèle d'interpolation linéaire satisfait, pour tout point* $y \in B(y^0; \Delta)$*, une erreur de la forme*

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\Delta,$$

où $\kappa_{eg} = \nu(1 + n^{\frac{1}{2}}\|\hat{L}^{-1}\|/2)$ *et* $\hat{L} = L/\Delta$*.*

Théorème 2.5 *Sous l'hypothèse 2.3, le modèle d'interpolation linéaire satisfait, pour tout point* $y \in B(y^0; \Delta)$*, une erreur de la forme*

$$|f(y) - m(y)| \leq \kappa_{ef}\Delta^2,$$

où $\kappa_{ef} = \kappa_{eg} + \nu/2$*.*

Nous remarquons que l'erreur d'interpolation d'un modèle linéaire complet dépend du rayon de la région de confiance, ainsi que de la constante de Lipschitz.

2.2 Modèle quadratique

Il est également possible d'utiliser des modèles plus complexes, dans l'espace $\mathcal{P}_n^2$ des polynômes de degré inférieur ou égal à 2, ou polynômes quadratiques. L'approximation locale de la fonction objectif $f(x)$ s'écrit alors :

$$Q_k(x_k + d) = Q_k(x_k) + \langle g_k, d\rangle + \tfrac{1}{2}\langle d, H_k d\rangle.$$

Pour construire ces modèles, nous pouvons par exemple choisir la base naturelle donnée par les polynômes de Lagrange ou par les polynômes de Newton. Il est également possible de construire des modèles quadratiques sous déterminés.

2.2.1 Modèle quadratique complet basé sur les polynômes de Lagrange

Soient les points d'interpolation $Y = \{y^0, y^1, ..., y^p\}$, où le polynôme $m(x)$ interpole la fonction $f(x)$ (avec ϕ_j une base de $\mathcal{P}_n^2$ de dimension $p_1 = p + 1 = (n+1)(n+2)/2$) :

$$m(x) = \sum_{j=0}^{p} \alpha_j \phi_j(y^i) = f(y^i),\ i = 0, ..., p. \tag{2.5}$$

On obtient le système linéaire suivant :

$$M(\phi, Y)\alpha_\phi = f(Y),$$

$$\text{où } M(\phi, Y) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_p(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_p(y^1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_p(y^p) \end{bmatrix},$$

$$\alpha_\phi = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix}, \text{ et } f(Y) = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^p) \end{bmatrix},$$

De manière analogue à l'interpolation linéaire, la définition d'un ensemble unisolvant est la suivante :

Définition 2.6 *L'ensemble $Y = \{y^0, y^1, ..., y^p\}$ est* ***unisolvant pour l'interpolation quadratique dans*** *$\mathbb{R}^n$ si la matrice correspondante $M(\phi, Y)$ est inversible pour la base correspondante ϕ dans $\mathcal{P}_n^2$.*

L'unisolvance mesure le conditionnement de la matrice des conditions d'interpolation $M(\phi, Y)$.

Lemme 2.7 *Soit une fonction $f : \mathbb{R}^n \to \mathbb{R}$ et un ensemble unisolvant $Y \in \mathbb{R}^n$, le polynôme d'interpolation $m(x)$,défini en 2.5, existe et est unique.*

Définition 2.8 *Soit un ensemble de points d'interpolation $Y = \{y^0, y^1, ..., y^p\}$, une base de $p_1 = p+1 = \frac{1}{2}(n+1)(n+2)$ polynômes $\ell_j(x)$, $j = 0, ..., p$, (dans l'espace des polynômes de degré inférieur ou égal à 2 dans $\mathbb{R}^n$: $\mathcal{P}_n^2$) est appelée une* ***base de polynômes de Lagrange*** *si*

$$\ell_j(y^i) = \delta_{ij} = \begin{cases} 1 & si \quad i = j \\ 0 & si \quad i \neq j. \end{cases} \tag{2.6}$$

Lemme 2.9 *Pour toute fonction $f : \mathbb{R}^n \to \mathbb{R}$ et tout ensemble unisolvant*

$$Y = \{y^0, y^1, ..., y^p\} \subset \mathbb{R}^n,$$

l'unique polynôme de $\mathcal{P}_n^2$, $m(x)$, qui interpole $f(x)$ en Y s'exprime de la façon suivante :

$$m(x) = \sum_{i=0}^{p} f(y^i)\ell_i(x), \tag{2.7}$$

où $\ell_i(x), i = 0, ..., p$ est la base de polynômes de Lagrange pour Y.

Définition 2.10 *Soit $\Lambda > 0$ et l'ensemble $B \in \mathbb{R}^n$. $\phi = \{\phi_0(x), \phi_1(x), ..., \phi_p(x)\}$ une base dans $\mathcal{P}_n^2$. L'ensemble unisolvant $Y = \{y^0, y^1, ..., y^p\}$ est dit* ***Λ-unisolvant dans B au sens de l'interpolation*** *si et seulement si*

- *pour la base des polynômes de Lagrange associée à Y*

$$\Lambda \geq \max_{0\leq i\leq p} \max_{x\in B} |\ell_i(x)|,$$

  ou de manière équivalente
- *pour tout $x \in B$, il existe $\lambda(x) \in \mathbb{R}^{p_1}$ tel que*

$$\sum_{i=0}^{p} \lambda_i(x)\phi(y^i) = \phi(x) \;\; \text{avec } \|\lambda(x)\|_\infty \leq \Lambda,$$

  c'est-à-dire que M peut être replacé dans la base canonique de manière unique :

$$M(\phi, Y)^\top \lambda(x) = \phi(x).$$

Notons que cette définition n'implique pas que l'ensemble de point Y soit contenue dans l'ensemble B dans lequel les valeurs absolues du polynôme de Lagrange sont maximisées. Toutefois, comme nous allons le voir, pour garantir la validité de l'algorithme 2.2 permettant de construire un ensemble Λ-unisolvant, les points sont générés dans B.

De manière analogue aux hypothèses 2.3 pour l'interpolation linéaire, on pose :

Hypothèses 2.11 *Supposons que $Y = \{y^0, y^1, ..., y^p\} \subset \mathbb{R}^n$, avec $p_1 = p + 1 = (n + 1)(n+2)/2$, est un ensemble de points échantillonnés unisolvant au sens de l'interpolation quadratique contenu dans $B(y^0; \Delta(Y))$ de rayon $\Delta = \Delta(Y)$. Et nous supposons que la fonction f est deux fois continûment différentiable dans un domaine ouvert Ω contenant $B(y^0; \Delta)$ et $\nabla^2 f$ est Lipschitz continu dans Ω avec la constante $\nu_2 > 0$.*

La mesure de conditionnement de la matrice des conditions d'interpolation $M(\phi, Y)$ peut être utilisé pour évaluer l'unisolvance de l'ensemble Y. Cette mesure peut être grande suivant le choix de la base ϕ et la normalisation des Y. Soit $\hat{Y} = \{0, \hat{y}^1, ..., \hat{y}^p\} \subset \mathcal{B}(0, 1)$ et $\max_{1\leq i\leq p} \|\hat{y}^i - \hat{y}^0\| = \max_{1\leq i\leq p} \|\hat{y}^i\| = 1$, et la base naturelle $\bar{\phi}$ suivante :

$$\bar{\phi} = \{1, x_1, x_2, ..., x_n, x_1^2/2, x_1x_2, ..., x_{n-1}x_n, x_n^2/2\}. \tag{2.8}$$

La matrice $\hat{M}$ est la matrice $M(\phi, \hat{Y})$ avec le choix de la base naturelle $\bar{\phi}$.

Théorème 2.12 *Supposons l'hypothèse 2.11. Alors, pour tout point $y \in B(y^0; \Delta(Y))$, nous obtenons*

- *l'erreur entre la Hessienne du modèle d'interpolation quadratique et la Hessienne de la fonction :*
$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\Delta,$$
- *l'erreur entre le gradient du modèle d'interpolation quadratique et le gradient de la fonction :*
$$|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\Delta^2,$$
- *l'erreur entre le modèle d'interpolation quadratique et la fonction :*
$$|f(y) - m(y)| \leq \kappa_{ef}\Delta^3,$$

où κ_{eh}, κ_{eg} et κ_{ef} sont donnés par :

$$\kappa_{eh} = 3\sqrt{2}p^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|/2,$$

$$\kappa_{eg} = 3(1+\sqrt{2})p^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|/2,$$

$$\kappa_{ef} = (6+9\sqrt{2})p^{\frac{1}{2}}\nu_2\|\hat{Q}^{-1}\|/4 + \nu_2/6.$$

La matrice $\hat{Q}$ est la matrice formée des dernières $p = (n+1)(n+2)/2-1$ lignes et colonnes de la matrice $\hat{M} = M(\bar{\phi}, \hat{Y})$, avec $\hat{Y} = Y/\Delta \subset \mathcal{B}(0,1)$ l'ensemble des points normalisés.

De plus, si l'ensemble $\hat{Y}$ est Λ-unisolvant dans la boule unité $\mathcal{B}(0,1)$ centrée en 0, le résultat suivant est démontré dans [28] :

Théorème 2.13
$$\|\hat{Q}^{-1}\| \leq \|\hat{M}^{-1}\| \leq \theta p_1^{\frac{1}{2}}\Lambda, \tag{2.9}$$

avec $\theta > 0$ dépendant de n, mais indépendant de $\hat{Y}$ et Λ.

Donc l'erreur d'interpolation est faible lorsque $\|\hat{Q}^{-1}\|$ est petit et donc lorsque la mesure d'unisolvance Λ est minimisée.

Nous remarquons que l'erreur d'interpolation d'un modèle quadratique complet dépend du rayon de la région de confiance et de la constante de Lipschitz (d'ordre 2 pour le quadratique), comme pour le modèle linéaire, mais également de la constante d'unisolvance Λ. Ainsi, la qualité du modèle de la fonction objectif est liée à l'unisolvance de l'ensemble des points d'interpolation échantillonnés.

2.2.2 Polynômes fondamentaux de Newton

Dans le cas des polynômes quadratiques, nous pouvons aussi choisir la base naturelle donnée par (2.8).

La différence avec les polynômes de Lagrange réside dans le fait que les points d'interpolation et les polynômes fondamentaux de Newton sont groupés en blocs. On organise ainsi les points de l'ensemble d'interpolation Y en $d+1 = 3$ blocs.

$$Y^{[\ell]} = \{y_1^{[\ell]}, ..., y_{|Y^{[\ell]}|}^{[\ell]}\} \quad (\ell = 0, ..., d),$$

avec le ℓ-ième bloc contenant : $|Y^{[\ell]}| = C^{\ell}_{\ell+n-1}$ points (où C^k_n est un coefficient binomial).

A chaque point $y_i^{[\ell]} \in Y^{[\ell]}$ correspond un seul polynôme fondamental de Newton $N_i^{[\ell]}$, de degré ℓ satisfaisant les conditions :

$$N_i^{[\ell]}(y_j^{[m]}) = \delta_{ij}\delta_{\ell m}, \text{ pour tout } y_j^{[m]} \in Y^{[m]} \text{ avec } m \leq \ell. \tag{2.10}$$

Le polynôme d'interpolation $m(x)$ est alors :

$$m(x) = \sum_{y_i^{[\ell]} \in Y} y_i^{[\ell]}(Y, f) N_i^{[\ell]}(x), \tag{2.11}$$

où les coefficients $y_i^{[\ell]}(Y, f)$ sont des différences finies généralisées appliquées à f.

Des propriétés équivalentes sur l'"unisolvance" de l'échantillon de points existent dans le cadre de l'utilisation de polynômes de Newton fondamentaux. Une borne sur l'erreur d'interpolation peut également être obtenue pour les polynômes de Newton fondamentaux [29, 28, 167]. Un avantage de ces polynômes sur les polynômes de Lagrange est le fait que les polynômes associés avec les blocs d'ordre faible sont indépendants des blocs d'ordre élevé. Ainsi, lorsque l'ensemble des points d'interpolation Y ne contient pas suffisamment de points pour construire un modèle quadratique complet, un bloc de points échantillonnés pour l'interpolation linéaire peut être utilisé comme base linéaire.

Cependant, le nombre d'évaluations de fonction requis pour obtenir un ensemble de points unisolvant pour l'interpolation quadratique n'est pas très attirant pour une grande partie des problèmes. Pour cette raison, l'utilisation de modèle quadratique sous déterminé ou de modèle à base de "Radial Basis Functions" permet de prendre en compte une information sur la courbure de la fonction objectif pour un coût sensiblement plus élevé que pour l'utilisation de modèle linéaire, mais cependant moindre que pour l'utilisation de modèle quadratique complet. Notons que les polynômes de Lagrange ou de Newton pourraient également être utilisés pour $d > 2$, pour un modèle plus précis de la fonction, mais la construction de modèles plus précis demanderait un coût encore plus important.

2.2.3 Modèle quadratique sous déterminé

Un modèle est sous déterminé lorsque le nombre de points d'interpolation p dans Y est inférieur aux nombres d'éléments dans la base des polynômes ϕ (dans le cas quadratique $q_1 = q + 1 = (n+1)(n+2)/2$). Dans ce cas, la matrice $M(\phi, Y)$ définissant les conditions d'interpolation a plus de colonnes que de lignes et le polynôme d'interpolation défini par

$$m(y^i) = \sum_{k=0}^{q} \alpha_k \phi_k(y^i) = f(y^i),\ i = 0, ..., p, \tag{2.12}$$

n'est pas unique.

Ce modèle est intéressant pour l'optimisation sans dérivées dans la mesure où la simulation de la fonction objectif est coûteuse à calculer. Le moyen de conserver un faible coût de construction pour l'ensemble des points d'interpolation est d'utiliser des modèles

linéaires ; toutefois, la convergence est lente car aucune information sur la courbure de la fonction objectif n'est exploitée. L'utilisation de modèles d'interpolation quadratique sous déterminés permet de remédier à ce problème, cela nécessite moins de points qu'un modèle d'interpolation quadratique complet, mais plus de points qu'un modèle d'interpolation linéaire. Ces points supplémentaires sont utilisés pour prendre en compte une information sur la courbure de la fonction. De plus, cette approche peut avec une certaine connaissance de la fonction objectif permettre de prendre en compte la structure creuse du gradient et de la Hessienne de la fonction objectif [27].

On définit l'ensemble des polynômes de Lagrange pour l'interpolation quadratique avec la base naturelle $\bar{\phi}$ (voir l'équation 2.8) et la matrice correspondante $M(\bar{\phi}, Y)$:

Définition 2.14 *Soit un ensemble de points d'interpolation $Y = \{y^0, y^1, ..., y^p\}$, avec $p < q$ (où $q_1 = q + 1 = (n+1)(n+2)/2$ est la dimension de $\mathcal{P}_n^2$), un ensemble de $p_1 = p + 1$ polynômes $\ell_j(x) = \sum_{i=0}^{q} (\alpha_j)_i \bar{\phi}_i(x)$, $j = 0, ..., p$, est appelé un* ***ensemble de polynômes de Lagrange de norme minimale pour la base*** *$\bar{\phi}$ (voir 2.8) si il est solution de norme minimale de*

$$\ell_j(y^i) = \delta_{ij} = \begin{cases} 1 & si \quad i = j \\ 0 & si \quad i \neq j. \end{cases} \tag{2.13}$$

Théorème 2.15 *L'ensemble des polynômes de Lagrange de norme minimale existe et est unique si la matrice $M(\bar{\phi}, Y)$ a ses lignes linéairement indépendantes. Dans ce cas, l'ensemble Y est dit unisolvant (remarque : $M(\bar{\phi}, Y)\alpha_j = e_{j+1}$ $j = 0, ..., p$, où e_{j+1} est le $j+1$ème colomne de la matrice identité d'ordre $q+1$).*

Lemme 2.16 *Pour toute fonction $f : \mathbb{R}^n \to \mathbb{R}$ et tout ensemble unisolvant*

$$Y = \{y^0, y^1, ..., y^p\} \subset \mathbb{R}^n,$$

le polynôme d'interpolation de norme minimale $m(x)$ (en termes de base $\bar{\phi}$) qui interpole $f(x)$ en Y s'exprime de la façon suivante :

$$m(x) = \sum_{i=0}^{p} f(y^i)\ell_i(x),$$

où $\{\ell_i(x),\ i = 0, ..., p\}$ est l'ensemble des polynômes de Lagrange de norme minimale pour Y.

Une définition de Λ-unisolvance valable dans le cadre de modèles d'interpolation quadratique sous déterminés est la suivante :

Définition 2.17 *Soit Λ et l'ensemble $B \in \mathbb{R}^n$ et $\bar{\phi}$ la base naturelle de monomes de $\mathcal{P}_n^2$. L'ensemble unisolvant $Y = \{y^0 = y^1, ..., y^p\}$ ($p < q$, où $q_1 = q+1$ est la dimension de $\mathcal{P}_n^2$) est dit* ***Λ-unisolvant dans B au sens de la norme minimale*** *si et seulement si*

- *pour l'ensemble des polynômes de Lagrange de norme minimale associé à Y*

$$\Lambda \geq \max_{0\leq i\leq p} \max_{x\in B} |\ell_i(x)|,$$

ou de manière équivalente

- *pour tout $x \in B$, il existe $\lambda(x) \in \mathbb{R}^{p_1}$ tel que*

$$\sum_{i=0}^{p} \lambda_i(x)\bar{\phi}(y^i) = \bar{\phi}(x) \;\; \textit{avec} \; \|\lambda(x)\|_\infty \leq \Lambda.$$

Il est important de noter pour la suite qu'un ensemble de $p_1 \geq n+1$ points bien placés pour l'interpolation quadratique de norme minimale n'est pas toujours unisolvant pour l'interpolation linéaire. Typiquement, pour l'utilisation de modèles d'interpolation sous déterminé, il est préferable de construire des modèles linéaires précis et de les enrichir avec de l'information sur la courbure, en espérant que la précision du modèle sera meilleure que celle du modèle linéaire.

Hypothèses 2.18 *Supposons que $Y = \{y^0, y^1, ..., y^p\} \subset \mathbb{R}^n$ soit un ensemble de points échantillonnés unisolvant au sens de l'interpolation linéaire (ou régression si $p > n$) contenu dans $B(y^0; \Delta(Y))$ de rayon $\Delta = \Delta(Y)$. Et nous supposons que la fonction f est continûment différentiable dans un domaine ouvert Ω contenant $B(y^0; \Delta)$ et ∇f est Lipschitz continu dans Ω avec la constante $\nu > 0$.*

Il est important de choisir un ensemble de points qui soit unisolvant pour l'interpolation linéaire, ce qui permet d'avoir l'erreur entre le modèle et la fonction du théorème suivant. On définit

$$\hat{L} = \frac{1}{\Delta} L = \frac{1}{\Delta}[y^1 - y^0 \cdots y^p - y^0]$$

et $\hat{L}^\dagger = (\hat{L}^\top \hat{L})^{-1}\hat{L}^\top$. Ces bornes sont indiquées en fonction du conditionnement de cette version normalisée de L.

Théorème 2.19 *Soit l'hypothèse 2.18. Alors, pour tout point $y \in B(y^0; \Delta(Y))$, nous avons*

- *l'erreur entre le gradient du modèle d'interpolation quadratique sous déterminé et le gradient de la fonction est :*

$$\|\nabla f(y) - \nabla m(y)\| \leq \frac{5\sqrt{p}}{2}\|\hat{L}^\dagger\|(\nu + \|H\|)\Delta,$$

- *l'erreur entre le modèle d'interpolation quadratique sous déterminé et la fonction est :*

$$|f(y) - m(y)| \leq \frac{5\sqrt{p}}{2}\|\hat{L}^\dagger\|(\nu + \|H\|)\Delta^2 + \frac{1}{2}(\nu + \|H\|)\Delta^2,$$

où H est la Hessienne du modèle.

Ainsi, nous notons qu'il est intéressant de construire des modèles pour lesquelles la norme de la Hessienne reste faible. Par conséquent, nous nous intéressons au modèle de norme de Frobenius minimale que l'on décrit dans la suite.

Nous considérons maintenant le cas de l'interpolation quadratique sous déterminée en séparant la base $\bar{\phi}$ en deux parties : la partie linéaire $\bar{\phi}_L = \{1, x_1, ..., x_n\}$ et la partie quadratique $\bar{\phi}_Q = \{x_1^2/2, x_, x_2, ..., x_n^2/2\}$. Le modèle d'interpolation s'écrit alors :

$$m(x) = \alpha_L^\top \bar{\phi}_L(x) + \alpha_Q^\top \bar{\phi}_Q(x),$$

où α_L et α_Q sont les parties appropriées du vecteur α (parties linéaire et quadratique).

La solution de norme de Frobenius minimale est alors la solution du problème d'optimisation en α_L et α_Q suivant :

$$\begin{array}{ll} \min & \frac{1}{2}\|\alpha_Q\|^2 \\ \text{s.t.} & M(\bar{\phi}_L, Y)\alpha_L + M(\bar{\phi}_Q, Y)\alpha_Q = f(Y). \end{array} \tag{2.14}$$

Cette technique est appelée ainsi car minimiser la norme de α_Q est équivalent à minimiser la norme de Frobenius de la Hessienne de $m(x)$:

$$\|A\|_F = \{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^2\}^{1/2}, \quad A \in \mathbb{R}^{n\times n}. \tag{2.15}$$

La condition pour l'existence et l'unicité du modèle de norme de Frobenius minimale est que la matrice suivante soit inversible :

$$F(\bar{\phi}, Y) = \begin{bmatrix} M(\bar{\phi}_Q, Y)M(\bar{\phi}_Q, Y)^\top & M(\bar{\phi}_L, Y) \\ M(\bar{\phi}_L, Y)^\top & 0 \end{bmatrix}. \tag{2.16}$$

L'ensemble Y est alors unisolvant dans le sens de la norme de Frobenius minimale, ce qui implique l'unisolvance au sens de l'interpolation linéaire et de l'interpolation quadratique sous déterminée dans le sens de la norme minimale.

Définition 2.20 *Soit un ensemble de points d'interpolation $Y = \{y^0, y^1, ..., y^p\}$, avec $n < p < q$, ($q+1$ est la dimension de $\mathcal{P}_n^2$), l'ensemble de $p_1 = p+1$ polynômes $\ell_i(x) = \sum_{j=0}^{q}(\alpha_i)_j\bar{\phi}_j(x)$, $i = 0, ..., p$, est appelé un* ***ensemble de polynômes de Lagrange de norme de Frobenius minimale pour la base*** *$\bar{\phi}$ si le $i$ème polynôme de Lagrange est une solution de (2.14) avec la fonction f remplacée par la fonction caractéristique pour y^i, pour tout $i = 0, ..., p$.*

Définition 2.21 *Soit $\Lambda > 0$ et l'ensemble $B \in \mathbb{R}^n$. Soit $\bar{\phi}$ la base naturelle des monomes de $\mathcal{P}_n^2$. L'ensemble unisolvant $Y = \{y^0, y^1, ..., y^p\}$ ($p < q$, où $q_1 = q+1 = (n+1)(n+2)/2$ est la dimension de $\mathcal{P}_n^2$) est dit* ***Λ-unisolvant dans B au sens de la norme de Frobenius minimale*** *si et seulement si :*

- *pour l'ensemble des polynômes de Lagrange de norme minimale associé à Y*

$$\Lambda \geq \max_{0\leq i\leq p}\max_{x\in B}|\ell_i(x)|,$$

ou de manière équivalente

- *pour tout $x \in B$, il existe $\lambda(x) \in \mathbb{R}^{p_1}$ tel que*

$$\begin{array}{ll} \min & \|M(\bar{\phi}_Q, Y^\top \lambda(x) - \bar{\phi}_Q(x)\|^2 \\ s.t. & M(\bar{\phi}_L, Y)^\top \lambda(x) = \bar{\phi}_L(x). \end{array},$$

$$\textit{avec } \|\lambda(x)\|_\infty \leq \Lambda.$$

Théorème 2.22 *Supposons l'hypothèse 2.18. Supposons que Y est Λ-unisolvant dans le sens de la norme de Frobenius minimale. Soit la borne supérieure Δ_{max} sur Δ, alors la Hessienne H du modèle de norme Frobenius minimale vérifie :*

$$\|H\| \leq \frac{4p_1\sqrt{q_1}\nu\Lambda}{c(\Delta_{max})},$$

où $c(\Delta_{max}) = \min\{1, 1/\Delta_{max}, 1/\Delta_{max}^2\}$.

Nous remarquons que l'erreur d'interpolation d'un modèle quadratique sous déterminé dépend comme pour le modèle quadratique à base de polynômes de Lagrange du rayon de la région de confiance et de la constante de Lipschitz, mais également de la constante d'unisolvance Λ. Notons comme pour un modèle quadratique complet que la qualité du modèle de la fonction objectif est liée à l'unisolvance de l'ensemble des points d'interpolation échantillonnés. En revanche, la construction de ce modèle ne nécessite pas autant d'évaluations de la fonction objectif que la construction d'un modèle quadratique complet. Ces modèles ont été utilisés dans l'optimisation sans dérivées par Powell [138, 142, 140, 142] et Wild [176, 177], ainsi que dans la méthode SQA présentée dans le chapitre 3.

2.3 Modèles à base de "Radial Basis Functions" cubiques

Un modèle de substitution RBF peut être utilisé pour interpoler la fonction f en $Y = \{y^0, y^1, ..., y^p\}$

$$m(x) = \sum_{i=0}^{p} \lambda_i \phi(\|x - y^i\|), \tag{2.17}$$

où $\phi(r) = r^3 : \mathbb{R}_+ \to \mathbb{R}$ et $\lambda_0, ..., \lambda_p \in \mathbb{R}$.

Un modèle de la forme suivante est classiquement utilisé [126, 178, 177] :

$$m(x) = \sum_{i=0}^{p} \lambda_i \phi(\|x - y^i\|) + \sum_{j=0}^{q} \gamma_j p_j(x), \tag{2.18}$$

où $p_j \in \mathcal{P}_n^{d-1}$, $j = 0, ..., q$ sont les fonctions de base pour les polynômes (permettant d'inclure des polynômes de degré $d-1$) et $\gamma_0, ..., \gamma_q \in \mathbb{R}$ avec

$$\sum_{i=0}^{p} \lambda_i p_j(y^i) = 0,\ j = 0, ..., q.$$

On obtient alors les conditions d'interpolation $m(y^i) = f(y^i)$, $i = 0, ..., p$ équivalentes à :

$$\begin{bmatrix} \phi & P \\ P^\top & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} f(Y) \\ 0 \end{bmatrix}, \tag{2.19}$$

où $\phi_{ij} = \phi(\|y^i - y^j\|)$ et $P_{ij} = p_j(y^i)$ pour $i, j \in \{0, ..., p\}$, et $f(Y)$ est le vecteur de valeurs $f(y^0), ..., f(y^p)$.

Définition 2.23 *ϕ est **définie positive conditionnelle d'ordre** d lorsque*

$$\sum_{i,j=0}^{p} \phi(\|y^i - y^j\|)\lambda_i\lambda_j$$

est positive pour tous points distincts $y^0, ..., y^p$ et $\lambda \neq 0$ satisfaisant

$$\sum_{i=0}^{p} \lambda_i p_j(y^i) = 0,\ j = 0, ..., q,$$

où le p représente la base de $\mathcal{P}_n^{d-1}$.

Les approches de [126, 178, 177] pour l'optimisation sans dérivées utilisent des modèles RBF cubiques et des polynômes linéaires :

$$m(x) = \sum_{i=0}^{p} \lambda_i \|x - y^i\|^3 + c + g^\top x. \tag{2.20}$$

Dans l'exemple ci-dessus, $p \in \mathcal{P}_1^n$ est linéaire et ϕ est définie positive conditionnelle d'ordre 2. Il existe alors un unique modèle interpolant en Y de la forme (2.20) unisolvant pour l'interpolation linéaire [177]. Dans ce cas précis, $n + 1$ points d'interpolation permettent de construire le polynôme d'interpolation linéaire, et les points d'interpolation restants permettent de construire le terme RBF. Wild, Regis et Shoemaker [178, 177] ont étudié le conditionnement de la matrice du système (2.19) pour obtenir une borne sur l'erreur d'interpolation. Oeuvray [126] et Wild, Regis et Shoemaker [178] utilisent ces modèles dans le cadre de l'optimisation sans dérivées. Ainsi, pour chacun des modèles utilisés par les méthodes d'optimisation sans dérivées basées sur la méthode de région de confiance, l'erreur d'interpolation du modèle peut être exprimée. Celle-ci dépend de la constante d'unisolvance de l'ensemble des points d'interpolation permettant de construire le modèle.

Certains modèles d'interpolation permettent de prendre en compte une information sur la courbure de la fonction objectif pour un coût sensiblement plus élevé que pour la construction d'un modèle linéaire, mais ne nécessitent pas autant d'évaluations que la construction d'un modèle quadratique complet. Ces modèles semblent être de bon choix pour l'optimisation sans dérivées basée sur la méthode de région de confiance.

3 Mise à jour de l'échantillon unisolvant pour l'optimisation sans dérivées

La méthode d'optimisation sans dérivées présentée dans la section suivante nécessite l'utilisation de modèle totalement linéaire. Dans cette section, la définition de la classe des modèles totalement linéaire est présentée, muni d'un algorithme permettant l'amélioration du modèle. Les modèles quadratiques sous déterminés construient à partir d'un ensemble Λ-unisolvant font partis de cette classe de modèle.

Nous souhaitons donc modifier l'ensemble Y des points d'interpolation pour améliorer la qualité du modèle. Nous savons que celle-ci est liée à l'unisolvance de l'ensemble des points d'interpolation. Un algorithme permettant d'améliorer (et maintenir) la constante d'unisolvance pour l'ensemble des points d'interpolation sera présenté. Rappelons que la définition de Λ-unisolvance (définitions 2.10 ou 2.17) requiert que le maximum des valeurs absolues des polynômes de Lagrange associés aux points d'interpolation soit inférieur à Λ.

Hypothèses 3.1 *Supposons qu'un ensemble $S \subset \mathbb{R}^n$ et un rayon Δ_{max} soient donnés. Supposons que f est continûment différentiable avec un gradient Lipschitz continu dans un domaine ouvert approprié contenant le voisinage Δ_{max} de l'ensemble S : $\cup_{x \in S} B(x; \Delta_{max})$.*

Définition 3.2 *Soit la fonction $f : \mathbb{R}^n \to \mathbb{R}$, qui satisfait l'hypothèse 3.1. Un ensemble de modèles $\mathcal{M} = \{m : \mathbb{R}^n \to \mathbb{R}, m \in \mathcal{C}^1\}$ est appelé* ***une classe de modèles totalement linéaires*** *si :*

(1) Il existe des constantes positives κ_{ef}, κ_{eg} et ν_1^m telles que pour tout $x \in S$ et $\Delta \in (0, \Delta_{max}]$, il existe un modèle $m(x+s)$ dans $\mathcal{M}$, avec un gradient Lipschitz continu ainsi qu'une constante de Lipschitz bornée par ν_1^m tel que :
- *l'erreur entre le gradient du modèle et le gradient de la fonction vérifie*

$$\|\nabla f(x+s) - \nabla m(x+s)\| \leq \kappa_{eg}\Delta, \ \forall s \in \mathcal{B}(0; \Delta), \tag{2.21}$$

- *l'erreur entre le modèle et la fonction vérifie*

$$|f(x+s) - m(x+s)| \leq \kappa_{ef}\Delta^2, \ \forall s \in \mathcal{B}(0; \Delta), \tag{2.22}$$

Ainsi le modèle m est dit totalement linéaire dans $\mathcal{B}(x; \Delta)$.

(2) Pour cette classe $\mathcal{M}$ il existe un algorithme permettant l'amélioration du modèle qui en un nombre d'itérations fini (uniformément borné) peut :
- *permettre d'établir si un modèle $m \in \mathcal{M}$ est totalement linéaire dans $\mathcal{B}(x; \Delta)$ (on parle de modèle certifié totalement linéaire),*
- *ou trouver un modèle $\tilde{m} \in \mathcal{M}$ qui est totalement linéaire dans $\mathcal{B}(x; \Delta)$.*

Pour $\Lambda > 1$, nous remarquons qu'un modèle d'interpolation polynomial linéaire construit à partir d'un ensemble Λ-unisolvant est un élément de la classe des modèles totalement linéaires, pour qui les constantes κ_{ef} et κ_{eg} dépendent seulement de Λ, du cardinal

de l'ensemble échantillonné, p_1, de la dimension de l'espace des polynômes, q_1, et de la constante de Lipschitz de ∇f (voir les définitions et théorèmes 2.2, 2.4 et 2.5). Les modèles de norme de Frobenius minimale basés sur un ensemble Λ-unisolvant, au sens de la norme de Frobenius minimale, sont totalement linéaires avec κ_{ef} et κ_{eg} dépendant seulement de Λ, p_1, q_1 et de la constante de Lipschitz de ∇f (voir les définitions et théorèmes 2.17 et 2.19). Une définition équivalente de modèle totalement quadratique dans le cadre de l'approximation quadratique est donnée dans [28].

Le maximum des valeurs absolues des polynômes de Lagrange $\ell(x)$ dans la région d'intérêt définit la Λ-unisolvance de l'ensemble d'interpolation dans la région (voir les définitions 2.10 et 2.17). Il est alors naturel de contrôler la valeur de $\ell(x)$ dans le but de maintenir l'unisolvance de l'ensemble d'interpolation.

Il est nécessaire d'avoir une procédure qui permet de vérifier si un ensemble de points d'interpolation Y est Λ-unisolvant dans une boule B. Dans le cas où l'ensemble n'est pas Λ-unisolvant, cette procédure effectue une étape d'amélioration qui permet en un nombre fini d'étapes de garantir que l'ensemble Y devienne Λ-unisolvant. Pour vérifier si y est Λ-unisolvant, les maximums de la valeur absolue des polynômes de Lagrange dans B sont calculés. Si la valeur absolue maximale est inférieure à Λ, l'ensemble Y est Λ-unisolvant. Dans le cas contraire, le point d'interpolation correspondant au polynôme de Lagrange de la valeur absolue maximale est remplacé par le point qui maximise ce polynôme de Lagrange dans la boule. L'algorithme permettant l'amélioration du modèle de la définition de la classe de modèles totalement linéaires est alors l'algorithme 2.2 pour les modèles construient à base de polynômes de Lagrange.

Algorithme 2.2 Améliorer l'unisolvance à l'aide des polynômes de Lagrange

Initialisation : Choisir une constante $\Lambda > 1$. Supposons que l'ensemble des points d'interpolation Y_0 est unisolvant (de cardinal p_1). Calculer les polynômes de Lagrange $\ell_i(x)$, $i = 0, ..., p$ associés à Y_0. **Pour k=1,2,...**

(1) Estimer

$$\Lambda_{k-1} = \max_{0 \leq i \leq p} \max_{x \in B} |\ell_i(x)|.$$

Λ_{k-1} peut être calculé exactement en maximisant la valeur absolue de tous les polynômes de Lagrange dans B.

(2) Si $\Lambda_{k-1} > \Lambda$, alors on détermine $i_k \in \{0, ..., p\}$, un indice pour lequel

$$\max_{x \in B} |\ell_{i_k}(x)| > \Lambda,$$

et $y_*^{i_k} \in B$ un point qui maximise $\ell_{i_k}(x)$ dans B (approché ou exact suivant la première étape). Mettre à jour Y_k en effectuant l'échange de points suivant : Y_k devient $Y_{k-1} \cup \{y_*^{i_k}\} \setminus \{y^{i_k}\}$.
Autrement (i.e., $\Lambda_{k-1} \leq \Lambda$), Y_{k-1} est Λ-unisolvant, l'algorithme s'arrête.

(3) Mise à jour de tous les coefficients des polynômes de Lagrange.

Ainsi, la mise à jour des polynômes de Lagrange, lors du remplacement du point

y^k par le point y_*^k dans l'ensemble des points d'interpolation Y, consiste à effectuer les changements suivants (si $\ell_k(y_*^k) \neq 0$) :

$$\ell_k^*(x) = \ell_k(x)/\ell_k(y_*^k), \tag{2.23}$$

$$\ell_j^*(x) = \ell_j(x) - \ell_j(y_*^k)\ell_k^*(x),\ \forall j \in \{0, ..., p\}, j \neq k, \tag{2.24}$$

où $\ell_j^*(x)$ sont les nouveaux polynômes de Lagrange correspondant à $Y \cup \{y_*^k\} \setminus \{y^k\}$.

D'après les thèorèmes 2.9 et 2.12, puis 2.19 et 2.22, plus la valeur de Λ est faible, plus le modèle est considéré comme un bon modèle (lorsqu'un des points d'interpolation est considéré comme le centre de la région de confiance B, $\Lambda > 1$). Le polynôme de Lagrange correspondant au point sélectionné par la maximisation des polynômes de Lagrange dans la boule subit ensuite les mises à jour 2.23 et 2.24, ce qui peut améliorer l'unisolvance.

Théorème 3.3 *Pour tout $\Lambda > 1$ donné, une boule fermée B, et une base polynomiale ϕ, l'algorithme 2.2 se termine avec un ensemble Y Λ-unisolvant après au plus $N = N(\Lambda, \phi)$ itérations, où N est une constante qui dépend de Λ et ϕ.*

Dans le cas de modèles de norme de Frobenius minimale, l'ensemble de polynômes de Lagrange est, de la même manière que pour les modèles quadratiques complets, calculé en résolvant un système d'équations linéaires. Lorsque la matrice de ce système 2.16 est inversible, la mise à jour de l'ensemble d'interpolation pour génerer un ensemble unisolvant est plus complexe que pour la construction d'un modèle quadratique complet. Une mise à jour efficace de l'ensemble des points d'interpolation (équivalente à l'algorithme 2.2) permettant de maintenir la Λ-unisolvance au sens de la norme de Frobenius minimale est suggérée dans [138]. Notons que l'étape de mise à jour des polynômes de Lagrange de l'algorithme 2.2 est adaptée pour prendre en compte les polynômes de norme de Frobenius minimale. Le théorème suivant équivalent au théorème (3.3) montre que dans le cas de polynômes de norme de Frobenius minimale, le nombre d'itérations de l'algorithme 2.2 est également fini.

Théorème 3.4 *Pour tout $\Lambda > 1$ donné, une boule fermée B, et une base polynomiale ϕ, l'algorithme 2.2 pour les polynômes de Lagrange de norme de Frobenius minimale termine avec un ensemble Y Λ-unisolvant après au plus $N = N(\Lambda, \phi)$ itérations, où N est une constante qui dépend de Λ et ϕ.*

Les démonstrations des théorèmes sont présentent dans [34].

4 Méthode d'optimisation sans dérivées basée sur la méthode de région de confiance

Après avoir présenté la construction de modèle quadratique à partir de polynômes de Lagrange et une procédure permettant d'en controler la qualité par modification

de l'échantillon de points d'interpolation, la méthode d'optimisation sans dérivées est présentée.

Considérons le modèle quadratique

$$m_k(x_k + s) = m_k(x_k) + s^\top g_k + \tfrac{1}{2} s^\top H_k s. \tag{2.25}$$

Dans le cadre de l'optimisation sans dérivées, nous utilisons des modèles où $H_k \neq \nabla^2 f(x_k)$, $g_k \neq \nabla f(x_k)$ (et même parfois $m_k(x_k) \neq f(x_k)$ pour la régression).

A chaque itération, le modèle $m_k(x_k + s)$ approche la fonction f dans un voisinage de x_k, la région de confiance $\mathcal{B}(x_k, \Delta_k) = \{x \in \mathbb{R}^n, \|x - x_k\| \leq \Delta_k\}$, avec Δ_k le rayon de la région de confiance et $\|\cdot\|$ une norme (souvent euclidienne). Le sous problème d'optimisation dans la région de confiance que nous considérons est le suivant :

$$\min_{x \in \mathcal{B}(0,\Delta_k)} m_k(x_k + s). \tag{2.26}$$

Le pas de Cauchy

Définition 4.1 *Le **pas de Cauchy** s_k^C est le pas vers le minimum du modèle dans la direction de plus forte pente dans la région de confiance. Si nous définissons*

$$t_k^C = \arg\min_{t \geq 0 : x_k - t g_k \in \mathcal{B}(x_k;\Delta_k)} m_k(x_k - t g_k),$$

le pas de Cauchy est donné par :

$$s_k^C = -t_k^C g_k. \tag{2.27}$$

Théorème 4.2 *Considérons le modèle 2.25 et le pas de Cauchy 2.27. Alors*

$$m_k(x_k) - m_k(x_k + s_k^C) \geq \tfrac{1}{2}\|g_k\| \min\{\frac{\|g_k\|}{\|H_k\|}, \Delta_k\}, \tag{2.28}$$

où nous supposons $\|g_k\|/\|H_k\| = +\infty$ lorsque $H_k = 0$.

En réalité, il n'est pas nécessaire de trouver le pas de Cauchy pour obtenir la convergence vers un point critique du premier ordre.

La mise à jour du rayon de la région de confiance

Supposons que l'itération courante est k, l'itéré courant x_k, et que le candidat pour l'itération suivante est $x_k + s_k$. Soit ν_0, ν_1 et γ satisfaisant $\gamma \in (0,1)$ et $0 \leq \nu_0 \leq \nu_1 < 1$ (avec $\nu_1 \neq 0$).

(1) $\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$

(2) Test de réduction suffisante : si $\rho_k \geq \nu_1$, le nouveau point $x_k + s_k$ est accepté, sinon le point candidat est rejeté.

(3) Test de réduction simple : si $\rho_k \geq \nu_0$, le nouveau point $x_k + s_k$ est accepté, mais la région de confiance est réduite si $\rho_k < \nu_1$, sinon le point candidat est rejeté.

(4) Mise à jour de la région de confiance : $\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty), & \text{si } \rho_k \geq \nu_1, \\ \{\gamma \Delta_k\}, & \text{si } \rho_k < \nu_1. \end{cases}$

L'idée importante, qui assure la convergence, est que plus le rayon de la région de confiance devient petit, plus le modèle est bon [34].

Description de la méthode

L'algorithme d'optimisation sans dérivées sans contraintes est présenté dans l'algorithme 2.4. Notons que le modèle m_k et le rayon de région de confiance Δ_k sont fixé après l'étape critique. L'itération termine par définir le modèle m_k^C et le rayon de région de confiance Δ_k^C courant pour la prochaine itération, qui peuvent être modifié par l'étape critique.

La procédure de l'étape critique de l'algorithme 2.4 est décrite dans l'algorithme 2.3.

Algorithme 2.3 Etape critique

Cet algorithme est appliqué seulement si $\|g_k^C\| \leq \epsilon_c$ et au moins une des propositions suivantes est vérifiée : le modèle m_k^C n'est pas certifié totalement linéaire dans $\mathcal{B}(x_k; \Delta_k^C)$ ou $\Delta_k^C > \mu\|g_k^C\|$. La constante $\omega \in (0,1)$ est choisie lors de l'initialisation de l'algorithme 2.4.

(1) **Initialisation** : $i = 0$ et $m_k^{(0)} = m_k^C$.

(2) **Répéter jusqu'à** $\tilde{\Delta}_k \leq \mu\|g_k^{(i)}\|$: $i = i+1$. Utiliser l'algorithme 2.2 d'amélioration de modèle pour améliorer le précédent modèle $m_k^{(i-1)}$ jusqu'à ce qu'il soit totalement linéaire dans $\mathcal{B}(x_k; \omega^{i-1}\Delta_k^C)$ (cela peut être fait en un nombre fini uniformément borné d'étapes déterminé par le choix, à l'étape d'initialisation de l'algorithme 2.4, de l'algorithme d'amélioration de modèle). Le nouveau modèle est noté $m_k^{(i)}$. $\tilde{\Delta}_k = \omega^{i-1}\Delta_k^C$ et $\tilde{m}_k = m_k^{(i)}$.

L'algorithme 2.3 reprend quelques idées de l'Implicit Filtering (voir section 3.1.2 du chapitre 1) pour améliorer le modèle courant.

Eléments pour démontrer la convergence vers un point critique de premier ordre

La convergence de la méthode présentée dans l'algorithme 2.3 présentant l'étape critique est démontrée dans [34]. Celle-ci se termine après un nombre finis de pas d'amélioration si $\|\nabla f(x_k)\| \neq 0$. Si $\|\nabla f(x_k)\| = 0$, l'étape critique de l'algorithme 2.3 boucle jusqu'au critère d'arrêt.

Hypothèses 4.3 *Supposons que f a une borne inférieure $L(x_0)$; alors il existe une constante κ_* telle que, pour tout $x \in L(x_0)$, $f(x) \geq \kappa_*$.*

Hypothèses 4.4 *Il existe une constante $\kappa_{bhm} > 0$ telle que, pour tout x_k généré par l'algorithme 2.4*

$$\|H_k\| \leq \kappa_{bhm}. \tag{2.29}$$

A partir des trois hypothèses (3.1), (4.3) et (4.4) et de la convergence de Δ_k vers 0, la convergence de l'algorithme 2.4 vers un point critique du premier ordre est démontrée dans [34].

Algorithme 2.4 Méthode d'optimisation avec région de confiance basée sur un modèle quadratique

(1) **Initialisation** : choix d'une classe de modèles totalement linéaires $\mathcal{M}$ et d'un algorithme permettant l'amélioration du modèle. Le point initial x_0 et le rayon maximal de la région de confiance Δ_{max} sont donnés. Nous supposons que le modèle initial $m_0^C(x_0+s)$ et le rayon de la région de confiance $\Delta_0^C \in (0, \Delta_{max}]$ sont donnés. Les constantes η_0, η_1, γ, γ_{inc}, ϵ_c, β, μ et ω sont données et satisfont : $0 \leq \eta_0 \leq \eta_1 < 1$ ($\eta_1 \neq 0$), $0 < \gamma_1 < 1 < \gamma_{inc}$, $\epsilon_c > 0$, $\mu > \beta > 0$ et $\omega \in (0,1)$. $k = 0$.

(2) **Etape critique** : si $\|g_k^C\| > \epsilon_c$, alors $m_k = m_k^C$ et $\Delta_k = \Delta_k^C$. Si $\|g_k^C\| \leq \epsilon_c$, alors l'algorithme 2.2 d'amélioration de modèle est utilisé pour vérifier si le modèle m_k^C est totalement linéaire dans $\mathcal{B}(x_k; \Delta_k^C)$. Si une des conditions suivantes est vérifiée :
- le modèle m_k^C n'est pas certifié totalement linéaire dans $\mathcal{B}(x_k; \Delta_k^C)$,
- $\Delta_k^C > \mu\|g_k^C\|$,

alors l'algorithme 2.3 est appliqué pour construire un modèle $\tilde{m}_k(x_k+s)$ totalement linéaire (pour les constantes κ_{ef}, κ_{eg} et ν_1^m) dans la boule $\mathcal{B}(x_k; \tilde{\Delta}_k)$, pour tout $\tilde{\Delta}_k \in (0, \mu\|\tilde{g}_k\|]$. Dans ce cas,

$$m_k = \tilde{m}_k \text{ et } \Delta_k = \min\{\max\{\tilde{\Delta}_k, \beta\|\tilde{g}_k\|\}, \Delta_k^C\}.$$

Autrement, fixer $m_k = m_k^C$ et $\Delta_k = \Delta_k^C$.

(3) **Minimisation du modèle** : calculer le pas s_k qui réduit suffisamment le modèle m_k tel que $x_k + s_k \in \mathcal{B}(x_k; \Delta_k)$.

(4) **Validation du point d'essai** : calculer $f(x_k+s_k)$ et définir

$$\rho_k = \frac{f(x_k) - f(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}.$$

Si $\rho_k \geq \eta_1$ ou si $\rho_k \geq \eta_0$ et le modèle est totalement linéaire dans $\mathcal{B}(x_k, \Delta_k)$, alors $x_{k+1} = x_k + s_k$ et le modèle $m_{k+1}^C(x_{k+1}+s)$ est mis à jour en incluant le nouvel itéré x_{k+1} dans l'ensemble échantillonné. Autrement le modèle et l'itération restent inchangés ($m_{k+1}^C = m_k$ et $x_{k+1} = x_k$).

(5) **Amélioration du modèle** : si $\rho_k < \eta_1$, utiliser l'algorithme 2.2 d'amélioration du modèle pour
- tenter de certifier que le modèle m_k est totalement linéaire dans $\mathcal{B}(x_k; \Delta_k)$,
- si nous ne pouvons pas certifier que le modèle m_k est totalement linéaire, une ou plusieurs étapes d'amélioration sont effectuées.

m_{k+1}^C est le nouveau modèle.

(6) **Mise à jour du rayon de la région de confiance** :

$$\Delta_{k+1}^C \in \begin{cases} [\Delta_k, \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}], & \text{si } \rho_k \geq \eta_1, \\ \{\gamma\Delta_k\}, & \text{si } \rho_k < \eta_1 \text{ et } m_k \text{ est totalement linéaire}, \\ \{\Delta_k\}, & \text{si } \rho_k < \eta_1 \\ & \text{et } m_k \text{ n'est pas certifé totalement linéaire}. \end{cases}$$

$k = k+1$ et on retourne au début de l'étape (2).

Chapitre 3

Sequential Quadratic Approximation (SQA)

Ce chapitre décrit la méthode SQA (Sequential Quadratic Approximation), basée sur la méthode d'optimisation sans contrainte NEWUOA de Powell [139, 140]. Cette méthode utilise un modèle quadratique interpolant la fonction objectif dans une région de confiance. Elle a été développée au cours de cette thèse afin de résoudre le problème d'optimisation sans dérivées sous contraintes suivant :

$$\min_{x \in \mathbb{R}^n} f(x), \tag{3.1}$$

sous les contraintes

$$\begin{cases} C_{DB}(x) \leq 0, \\ C_{DF}(x) \leq 0, \end{cases} \tag{3.2}$$

où C_{DB} représente les contraintes "derivative based" (appelées également "lower-level constraints" dans [51]), contraintes dont les dérivées sont connues et C_{DF} représente les contraintes "derivative free" (appelées également "upper-level constraints" dans [51]), contraintes dont les dérivées ne sont pas connues ; ces contraintes sont également appelées contraintes "boîtes noires" ou contraintes "difficiles" ($C_{DB} : \mathbb{R}^n \to \mathbb{R}^{n_{DB}}$, $C_{DF} : \mathbb{R}^n \to \mathbb{R}^{n_{DF}}$). Il est à noter que ces deux types de contraintes peuvent être linéaires ou non linéaires.

Les approximations quadratiques de la fonction objectif sont très utilisées pour accélérer la convergence des algorithmes d'optimisation sans dérivées, la courbure de la fonction objectif étant ainsi prise en compte. Ces méthodes ont été évaluées en pratique sur des problèmes de conception de pale d'un rotor d'hélicoptère [18], ainsi que sur un problème de conception d'agitateur [166].

Ce chapitre est organisé de la façon suivante. Nous présentons tout d'abord le principe général de la méthode d'optimisation sous contraintes par modèle quadratique approché (sous déterminé ou complet) dans une région de confiance, SQA, ainsi que son exten-

sion pour prendre en compte des contraintes. Puis, deux exemples de l'évolution des paramètres de l'algorithme sous contraintes sont représentés. Enfin, certaines perspectives d'amélioration de notre méthode seront proposées.

1 Principe général et prise en compte des contraintes

La méthode SQA est une méthode permettant l'optimisation de fonction sans dérivées sous contraintes. Elle consiste à transformer la résolution du problème d'optimisation (3.1) en la résolution d'une suite de problèmes d'optimisation locale (voir le chapitre 2). Cette méthode fait évoluer de manière séquentielle un modèle de la fonction objectif et des contraintes (construit à partir des évaluations précédentes) valable dans une région de confiance de rayon Δ autour du point optimal courant. Cette région de confiance évolue en fonction de l'adéquation du modèle à la fonction objectif. Une seconde région évolue au cours des itérations, celle-ci a pour but de maintenir assez de distance entre les points d'interpolation ; elle agit comme un zoom sur le point optimal courant, et ne fait donc que diminuer au cours des itérations de la même manière que la méthode NEWUOA de Powell [139].

La prise en compte des contraintes dans les méthodes d'interpolation par région de confiance est relativement naturelle pour les contraintes avec dérivées (définissant Ω l'ensemble des x admissibles pour les contraintes avec dérivées). L'idée est de générer des points uniquement dans l'intersection de Ω avec la région de confiance $\mathcal{B}(x_k; \Delta_k)$ (en considérant que le point de départ appartient au domaine admissible). Suivant la forme du domaine admissible Ω, le problème de minimisation du modèle de la fonction dans la région de confiance (voir la section 4 du chapitre 2) peut devenir difficile. L'optimisation des polynômes de Lagrange nécessaire pour maintenir un ensemble de points Λ-unisolvant (voir la section 3 du chapitre 2) est également un défi en présence de contraintes. Lorsque l'ensemble Ω est défini par des contraintes linéaires ou des contraintes de bornes, ce problème est plus facile à résoudre.

La prise en compte de contraintes sans dérivées disponibles doit être traitée de manière différente que celle de contraintes avec dérivées : des modèles de substitution sont construits de façon analogue à la fonction objectif. Notons que ces contraintes ne sont pas forcément vérifiées à chaque itération de la méthode. En revanche, elles n'auront pas d'effet sur l'unisolvance, qui est calculée sans considérer les contraintes sans dérivées. La simple idée de combiner la fonction objectif avec les contraintes sans dérivées en une fonction de mérite permet d'appliquer directement la théorie de convergence sans contrainte. Cette approche intuitive est souvent utilisée lorsque l'utilisateur a de bonnes intuitions pour combiner la fonction f et les contraintes C_{DF} en une fonction de pénalité. Une approche plus adaptée serait de construire des modèles quadratiques du Lagrangien (cela nécessite des modèles quadratiques des contraintes sans dérivées) permettant de se placer dans le cadre d'un algorithme SQP. La difficulté résidant dans la mise à jour des multiplicateurs de Lagrange. Nous avons choisi de minimiser les modèles de la fonction objectif dans la région de confiance et sous les modèles des contraintes. Les modèles des contraintes

peuvent être linéaires, conduisant à un sous problème de minimisation facile, ou bien quadratiques si un modèle quadratique est calculé pour f. Dans ce cas, la résolution du sous problème dans la région de confiance est alors plus difficile, mais la représentation des contraintes devient plus précise, ce qui est plus intéressant lorsque les simulations sont coûteuses.

Le principal avantage de construire un modèle quadratique de chacune des contraintes sans dérivées est que ces modèles sont généralement disponibles à moindre coût. C'est typiquement le cas pour les applications où les contraintes et la fonction objectif sont évaluées simultanément ("black-box"). Ainsi, dans le cas d'un modèle de f interpolant en un ensemble $Y\{x_0, x_1, ..., x_m\} \subset \mathbb{R}^n$ Λ-unisolvant, nous disposons de l'ensemble des polynômes de Lagrange pour Y et des valeurs des fonctions de contraintes en tous points de Y, $C_{DF}(x_i)$, alors les modèles quadratiques des contraintes peuvent être calculés en utilisant la propriété fondamentale des polynômes de Lagrange décrite dans le lemme (2.9) ou (2.16) du chapitre 2. Le coût additionnel de l'utilisation des modèles quadratiques des contraintes peut venir de la difficulté à résoudre le sous problème d'optimisation dans la région de confiance. En effet, en fonction de la forme des contraintes et du coût des évaluations de la fonction, ce coût peut devenir prohibitif. Ainsi, le sous problème peut contenir des contraintes incompatibles lors d'une itération, ce sous problème d'optimisation dans la région de confiance n'a alors pas de solution. Dans ce cas, un modèle de fonction de pénalisation peut être optimisé ou une méthode de filtre peut être utilisée [31, 32].

Une extension de UOBYQA (de Powell [137]) à la prise en compte de contraintes avec dérivées fait l'objet de plusieurs publications [167, 169, 168]. Le code DFO de Scheinberg [31, 32] permet également de prendre en compte les contraintes avec dérivées, mais également d'inclure les modèles quadratiques des contraintes sans dérivées aux sous problèmes d'optimisation dans la région de confiance. Les contraintes peuvent aussi tout simplement être prises en compte dans une fonction de mérite qui sera minimisée à la place de f. Powell (avec COBYLA dans [133]) modélise la fonction et les contraintes sans dérivées par des modèles d'interpolation linéaires. Colson, dans sa thèse [26], applique à sa méthode d'optimisation basée sur l'interpolation dans une région de confiance une méthode de filtre (tirée de l'algorithme de filtre couplé à un SQP suggéré dans [66]) pour éviter d'utiliser une fonction de mérite et de choisir les paramètres de pénalisation.

2 Description de la méthode SQA : méthode d'optimisation sans dérivée sous contraintes

Nous présentons notre méthode d'optimisation permettant la prise en compte de contraintes, SQA, basée sur la méthode NEWUOA de Powell [139, 140] Le traitement des contraintes dans ces méthodes reste une difficulté, des méthodes de pénalisation peu efficaces étant souvent utilisées.

Dans notre algorithme, nous faisons la différence entre deux types de contraintes avec

dérivées :

- les contraintes dites dures ("hard constraints") que l'on doit nécessairement satisfaire pour éviter par exemple un arrêt du simulateur (ces contraintes sont supposées linéaires) et
- les contraintes souples ("soft constraints") que l'on peut enfreindre.

La méthode permet donc de trouver la valeur minimale d'une fonction

$$\min_{x \in \mathbb{R}^n} f(x), \tag{3.3}$$

sous les contraintes

$$\begin{cases} C_{DB_h}(x) \leq 0, \\ C_{DB_s}(x) \leq 0, \\ C_{DF}(x) \leq 0, \end{cases} \tag{3.4}$$

où C_{DB_h} (respectivement C_{DB_s}) représente les contraintes avec dérivées dures (respectivement souples) et C_{DF} représente les contraintes sans dérivées ($C_{DB_h} : \mathbb{R}^n \to \mathbb{R}^{n_{DB_h}}$, $C_{DB_s} : \mathbb{R}^n \to \mathbb{R}^{n_{DB_s}}$, $C_{DF} : \mathbb{R}^n \to \mathbb{R}^{n_{DF}}$).

A chaque itération, un modèle quadratique Q est défini, approchant la fonction objectif f dans une région de confiance et des modèles quadratiques des contraintes C_{DF}. Chaque modèle quadratique a $\frac{1}{2}(n+1)(n+2)$ paramètres indépendants définis par autant d'évaluations de la fonction ; faire ces évaluations est trop coûteux en particulier pour des applications où la dimension du problème n est élevée. Notre méthode essaie de construire un modèle quadratique incomplet convenable à partir de moins de données. Le modèle $Q(x), x \in \mathbb{R}^n$, satisfait alors seulement m conditions d'interpolation lors de chaque itération. En effet, pour toute fonction f ou C_{DF} et tout ensemble $Y = \{x_1, x_2, ..., x_m\} \subset \mathbb{R}^n$ unisolvant, les polynômes Q et $\tilde{C}_{DF}$ interpolant respectivement f et C_{DF} en Y s'expriment :

$$Q(x) = \sum_{i=1}^{m} f(x_i)\ell_i(x), \ \tilde{C}_{DF}(x) = \sum_{i=1}^{m} C_{DF}(x_i)\ell_i(x), \tag{3.5}$$

où $\{\ell_i(x), i = 1, ..., m\}$ est l'ensemble des polynômes de Lagrange utilisant la technique de norme minimale pour Y (voir [34]). On a :

$$Q(x_i) = f(x_i), \quad i = 1, ..., m, \tag{3.6}$$

où le paramètre m est choisi initialement tel que $m \in [n+2, \frac{1}{2}(n+1)(n+2)]$. On prendra $m \geq n+2$ afin de toujours fournir au moins quelques conditions sur la matrice des dérivées secondes $\nabla^2 Q$ et $m \leq \frac{1}{2}(n+1)(n+2)$ car, en général, dans le cas contraire, aucun modèle quadratique Q ne peut satisfaire toutes les conditions (3.6). Les résultats numériques présentés par Powell sans contraintes [139] sont particulièrement intéressants dans le cas $m = 2n+1$ pour NEWUOA. Powell suggère d'utiliser $m = n+6$ points d'interpolation pour BOBYQA sur les cas traités dans [142] (sur des exemples de 20 à 200 paramètres).

De plus, pour chacune des contraintes sans dérivées, un modèle quadratique $\tilde{C}_{DF}$ interpolant est construit de la même manière que pour la fonction objectif dans la méthode

SQA. Ceux-ci sont construits à partir des mêmes points d'interpolation que ceux permettant de construire le modèle quadratique Q de la fonction f.

2.1 Modèle d'interpolation sous déterminé

L'efficacité de cette méthode tient dans la technique (suggérée par la méthode de Broyden symétrique [46] lorsque la dérivée première de f est disponible) de mise à jour de $\nabla^2 Q$. Celle-ci a été réutilisée par Powell dans [139] dans le cadre de l'optimisation sans dérivées. Soit à une itération donnée Q_{old} le modèle de l'itération précédente, et soit Q_{new} le nouveau modèle de l'itération courante qui doit satisfaire certaines conditions compatibles aux deux itérations et perdre certaines libertés sur les paramètres de Q_{new} pour satisfaire une nouvelle condition d'interpolation. Cette technique consiste à minimiser $\|\nabla^2 Q_{new} - \nabla^2 Q_{old}\|_F$, où l'indice $\cdot_F$ représente la norme de Frobenius (2.15).

L'utilisation de la norme de Frobenius de cette façon est équivalente à celle proposée dans la section 2.2.3 du chapitre 2 : le modèle est solution de

$$\begin{array}{ll} \min & \frac{1}{2}\|\alpha_Q - \alpha_Q^{old}\|^2 \\ \text{s.t.} & M(\bar{\phi}_L, Y)\alpha_L + M(\bar{\phi}_Q, Y)\alpha_Q = f(Y), \end{array} \tag{3.7}$$

où α_L et α_Q sont des parties appropriées de vecteur α (partie linéaire et quadratique) et $M(\bar{\phi}, Y)$ correspond à la matrice du système linéaire d'interpolation. Plutôt que de résoudre (3.7) directement, on peut résoudre un problème déplacé $\alpha_Q^{dif} = \alpha_Q - \alpha_Q^{old}$ et calculer α_Q comme $\alpha_Q^{dif} + \alpha_Q^{old}$ (de même pour α_L). Le problème déplacé devient alors :

$$\begin{array}{ll} \min & \frac{1}{2}\|\alpha_Q^{dif}\|^2 \\ \text{s.t.} & M(\bar{\phi}_L, Y)\alpha_L^{dif} + M(\bar{\phi}_Q, Y)\alpha_Q^{dif} = f^{dif}(Y), \end{array} \tag{3.8}$$

où $f^{dif}(Y) = f(Y) - m^{old}(Y)$. Le problème (3.8) est alors équivalent à (2.14).

Les conditions sur le nouveau modèle Q_{new} sont les équations d'interpolation (3.6). Ainsi $\nabla^2 Q_{new}$ est définie de manière unique, et Q_{new} lui aussi est unique, car le choix des points x_i exclut la possibilité qu'un polynôme linéaire différent de zéro $p(x), x \in \mathbb{R}^n$ vérifierait $p(x_i) = 0$, $i = 1, 2, ..., m$.

L'avantage de cette technique de mise à jour peut être expliqué en considérant le cas où la fonction objectif f est quadratique. Guidé par le modèle Q_{old} au départ de l'itération courante, un nouveau vecteur de variables $x_{new} = x_{opt} + d$ est choisi, où x_{opt} est tel que $f(x_{opt})$ est la valeur minimale de f calculée jusqu'à présent. Si l'erreur $|f(x_{new}) - Q_{old}(x_{new})|$ est relativement petite, alors le modèle a bien prédit la nouvelle valeur de f en ce point, même si l'erreur d'approximation $\|\nabla^2 Q - \nabla^2 f\|$ est importante. A l'inverse, si l'erreur $|f(x_{new}) - Q_{old}(x_{new})|$ est relativement grande, alors, en satisfaisant $Q_{new}(x_{new}) = f(x_{new})$, la technique de mise à jour devrait améliorer la précision du modèle de façon significative. Comme l'explique Powell [139], cette alternative fournit une excellente convergence des vecteurs des paramètres générés par l'algorithme, bien que l'erreur sur la dérivée seconde $\|\nabla^2 Q - \nabla^2 f\|_F$ soit grande pour tout Q.

Le nouveau modèle Q_{new} provient de Q_{old}, celui-ci est défini par un système de $(m + n + 1)^2$ équations linéaires. Soit W la matrice de ce système. La matrice inverse $H = W^{-1}$

est mise à jour à chaque itération, permettant une mise à jour séquentielle du modèle très rapide.

2.2 Description de l'algorithme

L'algorithme nécessite un vecteur initial $x_0 \in \mathbb{R}^n$, le choix du nombre m de conditions d'interpolation (3.6), les valeurs du rayon initial et du rayon final de la région de confiance appelées ρ_{beg} et ρ_{end}, ainsi que le nombre maximal d'évaluations de la fonction.

m est un entier fixé tel que :

$$n + 2 \leq m \leq \tfrac{1}{2}(n+1)(n+2). \tag{3.9}$$

Le choix $m = 2n + 1$ présenté dans [139] est particulièrement efficace, il constitue un bon compromis entre richesse du modèle et coût en nombre d'évaluations de la fonction objectif. Les points d'interpolation initiaux $x_i, i = 1, 2, ..., m$, vérifient la propriété suivante $\|x_i - x_0\|_\infty = \rho_{beg}$.

Algorithme 3.1 SQA (Sequential Quadratic Approximation)

(1) **Initialisation** : sélection des m points d'interpolation initiaux. Soit x_{opt} le point initial où la fonction de mérite $\varphi = f + \sum_{i=1}^{n_{DF}} \lambda_i C_{DF_{(i)\#}}$ [1] est minimale pour les m points évalués. Construction du premier modèle quadratique $Q \approx f$ de la fonction objectif f, des modèles quadratiques $\tilde{C}_{DF}$ des contraintes sans dérivées C_{DF} et du modèle résultant $\tilde{\varphi}$ de la fonction de mérite φ. $\rho = \Delta = \rho_{beg}$ (**voir section 2.2.1 de ce chapitre**)

(2) Résout le problème pour une précision ρ sur x

(a) (i) **Minimisation du modèle** : soit $d_k = arg\min_d Q(x_{opt} + d)$

$$\text{s.t.} \begin{cases} \|d\| \leq \Delta, \\ C_{DB}(x_{opt} + d) \leq 0, \\ \tilde{C}_{DF}(x_{opt} + d) \leq 0. \end{cases} \quad (\textbf{voir section 2.2.2 de ce chapitre})$$

(ii) Si $\|d_k\| < \frac{1}{2}\rho$: Δ est réduit d'un facteur 10 ou à sa borne inférieure ρ.

Si lors des trois dernières évaluations de la fonction objectif $j = k-1, k-2, k-3$, $\|d_j\| \leq \rho$ et l'écart du modèle à la fonction est tel que $|Q(x_{opt} + d_k) - f(x_{opt} + d_k)| \leq \frac{1}{8}\rho^2 CRVMIN$ ($CRVMIN$ est une estimation de la courbure minimale du modèle) $\rightarrow$ (2)(c) car le modèle est suffisamment proche de la fonction (**voir section 2.2.4 de ce chapitre**).

Sinon $\rightarrow$ (2)(b) car il est nécessaire de s'assurer de la validité du modèle avant d'effectuer de petits pas.

1. [1] Notons $C_\# = \max\{C, 0\}$.

(iii) Calcul de $f(x_{opt}+d_k)$, $C_{DF}(x_{opt}+d_k)$, $\varphi(x_{opt}+d_k) = f(x_{opt}+d_j) + \sum_{i=1}^{n_{DF}} \lambda_i C_{DF_{(i)}\#}(x_{opt}+d_j)$ et mise à jour du rayon Δ de la région de confiance à partir de $R = \frac{\varphi(x_{opt})-\varphi(x_{opt}+d_k)}{\tilde{\varphi}(x_{opt})-\tilde{\varphi}(x_{opt}+d_k)}$. Mise à jour des points d'interpolation : $x_{opt} = x_{opt}+d_k$ si $\varphi(x_{opt}+d_k) < \varphi(x_{opt})$ ou un simple ajout à l'ensemble des points d'interpolation si $x_{opt}+d_k$ améliore le modèle. Mise à jour du modèle Q et des modèles $\tilde{C}_{DF}$ pour interpoler la fonction f et les contraintes sans dérivées C_{DF} en $x_{opt}+d_k$.

(iv) Si $R > 0.1$ (décroissance satisfaisante) $\rightarrow$ (2)(a)(i) sinon continuer.

(b) Test de la validité des modèles Q et $\tilde{C}_{DF}$. Le modèle est considéré valide si tous les points d'interpolation courants x_i, $i = 1, ..., m$ sont assez proches du point optimal x_{opt}, c'est-à-dire si la distance euclidienne $\|x_i - x_{opt}\| < 2\Delta$ pour tout x_i, $i = 1, ..., m$.

(i) Si le modèle n'est pas valide :
Amélioration de la qualité des modèles Q et $\tilde{C}_{DF}$:
$\max_{d\in\mathbb{R}^n} |\ell_t(x_{opt}+d)|$
s.t. $\begin{cases} \|d\| \leq \overline{\Delta}, \\ C_{DB}(x_{opt}+d) \leq 0. \end{cases}$ (**voir section 2.2.3 de ce chapitre**)
calcul de $f(x_{opt}+d_k)$ et $C_{DF}(x_{opt}+d_k)$, l_t est le polynôme de Lagrange associé au plus mauvais point (plus éloigné) de l'ensemble d'interpolation qui est supprimé et remplacé par le nouveau point évalué $x_{new} = x_{opt}+d_k$ (voir section 2.2.3 de ce chapitre). Ainsi la précision de Q et de $\tilde{C}_{DF}$ après leur mise à jour est supérieure $\rightarrow$ (2)(a)(i).

(ii) Si les modèles sont valides : Si $\|d\| > \rho$ ou $\Delta > \rho$ ou $R > 0 \rightarrow$ (2)(a)(i) sinon continuer.

(c) Si $\rho > \rho_{end}$:
Evolution de ρ : réduction de ρ et de Δ (zoom sur x) (**voir section 2.2.4 de ce chapitre**) $\rightarrow$ (2)(a) sinon continuer.

(3) $\rho = \rho_{end}$ et $f(x_{opt}+d_k)$ et $C_{DF}(x_{opt}+d_k)$ sont calculés si cela n'a pas déjà été effectué ($\|d_k\| < \frac{1}{2}\rho$).

Chaque étape de l'algorithme 3.1 est explicitée dans la suite de ce chapitre. Notons que dans notre version, les poids de pénalisation utilisés λ_i, $i = 1, ..., n_{DF}$ sont constants au cours des itérations. Leurs valeurs par défaut est $\lambda_i = 100$ (ceci suppose une normalisation adéquate des contraintes sans dérivées et de la fonction objectif). De possibles mises à jour séquentielles de ces paramètres sont présentées dans la section 4.4 de ce chapitre.

2.2.1 Initialisation

Le paramètre ρ_{beg} et le point de départ x_0 peuvent être modifiés afin que le point initial vérifie chacune des contraintes avec dérivées et que les $m-1$ points initiaux vérifient les

contraintes dures (contraintes à vérifier absolument pour éviter un éventuel échec de la simulation). Si une faible réduction de ρ_{beg} permet à tous les points initiaux de vérifier les contraintes dures, celle-ci est effectuée ; dans le cas contraire, le point de départ x_0 est déplacé pour permettre aux m points initiaux de vérifier les contraintes dures, tout en vérifiant lui-même toutes les contraintes avec dérivées (les détails de l'initialisation sont dans l'annexe E). Le paramètre ρ_{beg} doit être choisi de manière à ce que les valeurs de la fonction objectif f calculées aux points d'interpolation initiaux fournissent de l'information utile sur le comportement de cette fonction objectif au voisinage de x_0. Le paramètre ρ_{end}, qui satisfait $\rho_{end} \leq \rho_{beg}$, est de l'ordre de grandeur de la précision requise sur x à la convergence.

Lors de l'étape (1) de l'algorithme, le modèle quadratique de f à la première itération est de la forme :

$$Q(x_0 + d) = Q(x_0) + d^T \nabla Q(x_0) + \tfrac{1}{2} d^T \nabla^2 Q d, \quad d \in \mathbb{R}^n, \tag{3.10}$$

où x_0 est le vecteur initial.

Les m points initiaux $x_i, i = 1, 2, ..., m$ sont choisis de manière bien précise autour du point x_0 (voir annexe E) et permettent de définir le polynôme initial afin de vérifier les m conditions d'interpolation (3.6).

Le paramètre ρ est la borne inférieure du rayon de la région de confiance Δ, il appartient à $[\rho_{end}, \rho_{beg}]$. La valeur de Δ est mise à jour au cours des itérations, alors que le but de ρ est de maintenir assez de distance entre les points d'interpolation x_i, $i = 1, 2, ..., m$, afin de restreindre les erreurs sur Q qui proviennent des conditions d'interpolation (3.6) lorsque le calcul de f est entâché d'erreur. C'est pourquoi, de manière analogue à la réduction du pas des méthodes de simplexe de Nelder-Mead et de Pattern Search (comme l'explique Powell dans [141]), ρ est diminué seulement lorsque la contrainte $\Delta \geq \rho$ semble empêcher de poursuivre la minimisation de la fonction objectif. Dans ce cas, ρ est divisé par un facteur dix, excepté dans le cas $\rho = \rho_{end}$ où les itérations se terminent.

La matrice initiale permettant de résoudre le système linéaire d'interpolation $H = W^{-1}$ est initialisée (W est le système linéaire d'équations permettant la mise à jour du modèle quadratique). Q est mis à jour à l'aide de Q_{old} : $Q_{new} = Q_{old} + D$, où la fonction quadratique D est construite de sorte que $\|\nabla^2 D\|_F^2$ soit minimale sous les contraintes

$$D(x_i) = f(x_i) - Q_{old}(x_i), \quad i = 1, 2, ..., m. \tag{3.11}$$

Ces contraintes sont équivalentes à $Q_{new}(x_i) = f(x_i), i = 1, 2, ..., m$. Le calcul de D est un problème de programmation quadratique sous contraintes linéaires. Soit $\lambda_j, j = 1, 2, ..., m$, les multiplicateurs de Lagrange vérifient les conditions KKT :

$$\sum_{j=1}^{m} \lambda_j = 0 \quad et \quad \sum_{j=1}^{m} \lambda_j (x_j - x_0) = 0, \tag{3.12}$$

et la matrice des dérivées secondes de D prend la forme

$$\nabla^2 D = \sum_{j=1}^{m} \lambda_j x_j x_j^T = \sum_{j=1}^{m} \lambda_j (x_j - x_0)(x_j - x_0)^T. \tag{3.13}$$

En utilisant cette forme de $\nabla^2 D$, on obtient

$$D(x) = c + (x - x_0)^T g + \tfrac{1}{2} \sum_{j=1}^{m} \lambda_j \{(x - x_0)^T (x_j - x_0)\}^2, \quad x \in \mathbb{R}^n, \tag{3.14}$$

où $c \in \mathbb{R}, g \in \mathbb{R}^n$ et $\lambda \in \mathbb{R}^m$ restent à déterminer. Les conditions (3.11) et (3.12) donnent le système d'équations linéaires

$$\left(\begin{array}{c|c} A & X^T \\ \hline X & 0 \end{array}\right) \left(\begin{array}{c} \lambda \\ \hline c \\ g \end{array}\right) = \left(\begin{array}{c} r \\ \hline 0 \end{array}\right) \begin{array}{l} \updownarrow m \\ \updownarrow n+1 \end{array} \tag{3.15}$$

où A contient les éléments

$$A_{ij} = \tfrac{1}{2}\{(x_i - x_0)^T (x_j - x_0)\}^2, \quad 1 \leq i, j \leq m, \tag{3.16}$$

X est la matrice $(n+1) \times m$

$$X = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 - x_0 & x_2 - x_0 & \cdots & x_m - x_0 \end{pmatrix} \tag{3.17}$$

et r a les composantes $f(x_i) - Q_{old}(x_i)$, $i = 1, 2, ..., m$. Par conséquent, nous définissons les matrices W et H :

$$W = \left(\begin{array}{c|c} A & X^T \\ \hline X & 0 \end{array}\right) \quad \text{et} \quad H = W^{-1} = \left(\begin{array}{c|c} \Omega & \Xi^T \\ \hline \Xi & \Upsilon \end{array}\right) \tag{3.18}$$

Ainsi W est déterminé à partir des vecteurs $x_i, i = 1, 2, ..., m$, mais les éléments de Ξ et Υ sont requis ainsi que la factorisation de Ω. La position des points d'interpolation initiaux fournit des formules pour l'initialisation de tous ces termes (voir [139]).

Les modèles de chacune des contraintes sans dérivées $\tilde{C}_{DF}$ sont initialisés à partir des évaluations de celles-ci aux mêmes points initiaux que pour le modèle de la fonction objectif Q. Ceux-ci sont donc construits à partir du même système d'équations linéaires (3.15) pour chaque contrainte $i = 1, ..., n_{DF}$, où cette fois-ci, r a les composantes $C_{DF_{(i)}}(x_j) - \tilde{C}_{DF_{(i)}}(x_j)$, $i = 1, 2, ..., m$ ($\tilde{C}_{DF_{(i)}}(x_j)$ est le modèle de l'itération courante avant sa mise à jour).

2.2.2 Minimisation du modèle

Lors des phases (2)(a), l'algorithme détermine un pas d_k solution du sous problème :

$$\begin{array}{l} \min\limits_{d \in \mathbb{R}^n} Q(x_{opt} + d) \\ \text{s.t.} \left\{ \begin{array}{l} \|d\| \leq \Delta, \\ C_{DB}(x_{opt} + d) \leq 0, \\ \tilde{C}_{DF}(x_{opt} + d) \leq 0, \end{array} \right. \end{array} \tag{3.19}$$

dont la solution, si elle existe, est obtenue à l'aide d'une méthode d'optimisation SQP (Sequential Quadratic Programming [159] ou annexe C). La méthode SQP s'est révélée, en effet, comme une des méthodes les plus efficaces pour la résolution des problèmes non linéaires sous contraintes. Celle-ci repose sur une reformulation itérative du problème en problème de programmation quadratique (QP), au moyen d'une approximation quadratique du Lagrangien et d'une linéarisation des contraintes

$$\mathcal{L}(x,\lambda) = f(x) + \lambda_{DB}^T C_{DB}(x) + \lambda_{DF}^T \tilde{C}_{DF}(x), \tag{3.20}$$

où λ_{DB} et λ_{DF} sont les vecteurs des multiplicateurs de Lagrange associés aux contraintes d'inégalité avec dérivées et respectivement des modèles des contraintes sans dérivées. Le problème QP résultant à chaque itération est ensuite résolu. Un des points forts de la méthode SQP réside dans le fait que la progression vers la solution optimale est effectuée à partir de points intermédiaires « faisables »mais aussi « non faisables », proches du domaine des contraintes. Ainsi, contrairement à de nombreuses méthodes qui vérifient les contraintes à chaque itération, la méthode SQP n'impose le respect des contraintes que pour la solution finale. Cette caractéristique, associée à la technique BFGS (Broyden-Fletcher-Goldfarb-Shanno) d'approximation de l'inverse de la matrice Hessienne, conduit à faire de la méthode SQP utilisée, une méthode rapide.

Dans notre cas, la fonction objectif est déjà un modèle quadratique Q et sa Hessienne est disponible, nous utilisons donc cette matrice dans la méthode SQP. La résolution de ce problème peut être difficile car le polynôme quadratique n'est pas toujours convexe, c'est-à-dire que la matrice Hessienne du Lagrangien n'est pas forcément définie positive. Pour surmonter cette difficulté, une modification M_k est choisie à chaque itération afin d'assurer une Hessienne du Lagrangien $\nabla^2 Q(x_k) + \lambda_{DB}\nabla^2 C_{DB}(x_k) + \lambda_{DF}\nabla^2 \tilde{C}_{DF}(x_k) + M_k$ semi-définie positive au cours des itérations du SQP. Nous sommes intéressés par une modification de la Hessienne du Lagrangien à moindre coût. La stratégie "idéale" selon Nocedal [125] consiste à utiliser la décomposition en valeurs propres de la matrice $\nabla^2 Q(x_k) + \lambda_{DB}\nabla^2 C_{DB}(x_k) + \lambda_{DF}\nabla^2 \tilde{C}_{DF}(x_k)$ et à annuler les valeurs propres inférieures à 0. Cette stratégie est coûteuse mais est tout de même moins coûteuse en général que l'évaluation de notre fonction objectif f. Soient $(\kappa_1, ..., \kappa_n)$ les valeurs propres de la diagonale de Λ. Nous pouvons donc écrire $\nabla^2 Q(x_k)+\lambda_{DB}\nabla^2 C_{DB}(x_k)+\lambda_{DF}\nabla^2 \tilde{C}_{DF}(x_k) = L\Lambda L^T$, avec L la matrice des vecteurs propres. La modification de la Hessienne du Lagrangien est alors : $L\Lambda_{M_k}L^T$, où les éléments $(|\min(\kappa_1, 0)|, ..., |\min(\kappa_n, 0)|)$ sont sur la diagonale de Λ_{M_k}, la Hessienne du Lagrangien modifiée est alors semi-définie positive.

Ce problème de minimisation du modèle quadratique de f dans la région de confiance sous contrainte peut ne pas avoir de solution en raison de l'approximation des contraintes sans dérivées $\tilde{C}_{DF}$, dans le cas où des contraintes seraient incompatibles (contraintes modélisées $\tilde{C}_{DF}$ et contraintes avec dérivées C_{DB}). Dans ce cas, le problème de minimisation précédent est alors remplacé par le problème suivant :

$$\begin{aligned} &\min_{d\in\mathbb{R}^n} \tilde{\varphi}(x_{opt} + d) \\ &\text{s.t.} \begin{cases} \|d\| \leq \Delta, \\ C_{DB}(x_{opt} + d) \leq 0, \end{cases} \end{aligned} \tag{3.21}$$

où $\tilde{\varphi}(x) = Q(x) + \sum_{i=1}^{n_{DF}} \lambda_i \tilde{C}_{DF_{(i)}\#}(x)$. Les poids de pénalisation utilisés λ_i, $i = 1, ..., n_{DF}$ sont constants au cours des itérations. Leurs valeurs par défaut sont $\lambda_i = 100$. De possibles mises à jour séquentielles de ces paramètres sont présentées dans la section 4.4 de ce chapitre.

Si d_k la solution du sous problème 3.19 ou 3.21 est telle que $\|d_k\| < \Delta$, alors Q a une courbure positive dans toutes les directions, $CRVMIN$ est alors fixé à la courbure minimale (non nulle, car $\kappa_i > 0$ pour tout $i \in [1, n]$ et Q est convexe), c'est-à-dire à la valeur propre minimale de la Hessienne du Lagrangien :

$$CRVMIN = \min\{\kappa_1, ..., \kappa_n\}. \tag{3.22}$$

Dans le cas contraire, $CRVMIN$ est fixé à zéro. Cette courbure minimale est utilisée si $\|d\| < \rho/2$ (voir algorithme 3.1), ce qui est explicité par la suite. Si à l'inverse, $\|d_k\| \geq \rho/2$, l'étape (2)(a)(iii) suit l'étape (2)(a)(ii), et Δ est mis à jour selon le rapport de l'amélioration observée de la fonction objectif et de l'amélioration prédite par l'approximation quadratique

$$R = \frac{\varphi(x_{opt}) - \varphi(x_{opt} + d)}{\tilde{\varphi}(x_{opt}) - \tilde{\varphi}(x_{opt} + d)} \tag{3.23}$$

et selon la norme euclidienne du pas d_k. Soient Δ_{old} et Δ_{new} respectivement les valeurs de l'itération précédente et de l'itération courante de Δ. Les valeurs possibles de Δ_{new} sont $\frac{1}{2}\|d_k\|$, $\|d_k\|$ et $2\|d_k\|$ respectivement dans les cas $R \leq 0.1$, $0.1 < R \leq 0.7$ et $R > 0.7$, mais si $R > 0.1$ une réduction trop importante de Δ serait trop restrictive à la prochaine itération. De plus, on doit toujours avoir $\Delta \geq \rho$ et nous préferons éviter que le rayon de la région de confiance Δ soit proche de ρ. Δ_{new} est donc fixé à

$$\Delta_{new} = \begin{cases} \rho, & \Delta_{int} \leq 1.5\rho, \\ \Delta_{int}, & \Delta_{int} > 1.5\rho, \end{cases} \tag{3.24}$$

où Δ_{int} est la valeur intermédiaire

$$\Delta_{int} = \begin{cases} \frac{1}{2}\|d_k\|, & R \leq 0.1, \\ \max\{\|d_k\|, \frac{1}{2}\Delta_{old}\}, & 0.1 < R \leq 0.7, \\ \max\{2\|d_k\|, \frac{1}{2}\Delta_{old}\}, & R > 0.7. \end{cases} \tag{3.25}$$

Les points d'interpolation associés au prochain modèle quadratique sont également sélectionnés : en pratique un des points courants x_i, $i = 1, 2, ..., m$, est remplacé par $x_{opt} + d_k$ et tous les autres sont retenus. Dans ce cas, l'indice du point d'interpolation que l'on supprime, $MOVE$, est fixé à l'indice du point d'interpolation remplacé de la façon suivante :
soit $\mathcal{T}$ l'ensemble $\{1, 2, ..., m\}$, excepté l'entier opt dans le cas $f(x_{opt} + d_k) \geq f(x_{opt})$, afin que le meilleur point jusqu'alors x_{opt} ne puisse pas être supprimé de l'ensemble des points d'interpolation. $MOVE$ est fixé par l'algorithme à zéro ou à l'entier $t^* \in \mathcal{T}$ qui satisfait l'équation :

$$wg_{t^*}|\sigma_{t^*}| = \max\{wg_t|\sigma_t| : t \in \mathcal{T}\}, \tag{3.26}$$

où

$$\sigma_t = (e_t^T H e_t)(\tfrac{1}{2}\|x^+ - x_0\|^4 - w^T H w) + (e_t^T H w)^2, \quad t \in \mathcal{T} \tag{3.27}$$

est une valeur intervenant au dénominateur lors de la mise à jour du modèle d'interpolation (cf Powell pour la mise à jour du modèle [139]), avec $w \in \mathbb{R}^{m+n+1}$ tel que,

$$\begin{array}{l} w_i = \frac{1}{2}(x_i - x_0)^T (x^+ - x_0)^2, \quad i = 1, 2, ..., m \\ w_{m+1} = 1 \text{ et } w_{i+m+1} = (x^+ - x_0)_i, \quad i = 1, 2, ..., n. \end{array} \tag{3.28}$$

Choisir $MOVE$ tel que $|\sigma_t|$ soit aussi grand que possible n'est pas avantageux pour la validité du modèle, toutefois σ est pondéré par wg_t qui prend en compte la distance euclidienne entre le point optimal courant x_{opt} et les points d'interpolation. Le point enlevé est alors un compromis entre le plus éloigné et celui qui a une forte contribution au modèle, ce point est alors un point éloigné pouvant faussé le modèle.wg_t permet alors de supprimer les points d'interpolation les plus éloignés du point optimal et prend la valeur

$$wg_t = \max[1, \{\|x_t - x^*\| / \max[0.1\Delta, \rho]\}^6], \quad t \in \mathcal{T}, \tag{3.29}$$

où x^* est le point sélectionné précédemment, soit x_{opt}, soit $x_{opt} + d_k$. $MOVE = 0$ préserve les anciens points d'interpolation, c'est pourquoi, $MOVE$ est fixé à zéro à la place de t^* si et seulement si

$$\varphi(x_{opt} + d_k) \geq \varphi(x_{opt}) \text{ et } wg_{t^*}|\sigma_{t^*}| \leq 1.$$

En effet, une petite valeur de $wg_{t^*}|\sigma_{t^*}|$ signifie que tous les points d'interpolation sont assez proches du point optimal car nous avons $|\sigma_t|$ de l'ordre de $\mathcal{O}(1)$ si $\|x_t - x_{opt}\| \leq \Delta$ et de l'ordre de $\mathcal{O}(\Delta^4/\|x_t - x_{opt}\|^4)$ si $\|x_t - x_{opt}\| > \Delta$ [138, 139].

Lors de cette étape, les mises à jour des modèles quadratiques Q et $\tilde{C}_{DF}$ sont effectuées lorsque l'ensemble des points d'interpolation est modifié. Les nouveaux points sont alors

$$\left\{ \begin{array}{l} x_t^+ = x_{opt} + d = x^+, \\ x_i^+ = x_i \quad i \in \{1, 2, ..., m\} \backslash \{t\}, \end{array} \right. \tag{3.30}$$

où t est noté à la place de $MOVE$. La nouvelle évaluation est donc ajoutée à l'ensemble des points d'interpolation pour remplacer le point le moins intéressant lorsque celle-ci minimise la fonction objectif ou également lorsqu'elle améliore le modèle. L'écart $D = Q_{new} - Q_{old}$ satisfait des conditions équivalentes à (3.11) pour les nouveaux points, alors que Q_{old} interpole f aux anciens points d'interpolation. Donc D est la fonction quadratique qui minimise $\|\nabla^2 D\|_F$ sous les contraintes suivantes

$$D(x_i^+) = \left\{ \begin{array}{ll} f(x^+) - Q_{old}(x^+), & \text{si } i = t \\ 0, & \text{sinon} \end{array} \right\} = \{f(x^+) - Q_{old}(x^+)\}\delta_{it}, \quad i = 1, 2, ..., m. \tag{3.31}$$

Soit W^+ et H^+ les matrices

$$W^+ = \left(\begin{array}{c|c} A^+ & (X^+)^T \\ \hline X^+ & 0 \end{array} \right) \quad \text{et} \quad H^+ = (W^+)^{-1} = \left(\begin{array}{c|c} \Omega^+ & (\Xi^+)^T \\ \hline \Xi^+ & \Upsilon^+ \end{array} \right), \tag{3.32}$$

où A^+ et X^+ sont définis en remplaçant les anciens points d'interpolation par les nouveaux dans les équations (3.16) et (3.17). La nouvelle fonction D est déduite du système (3.15) et des conditions (3.31)

$$D(x) = c^+ + (x - x_0)^T g^+ + \tfrac{1}{2} \sum_{j=1}^{m} \lambda_j^+ \{(x - x_0)^T (x_j^+ - x_0)\}^2, \quad x \in \mathbb{R}^n, \tag{3.33}$$

les paramètres sont les composantes du vecteur

$$\begin{pmatrix} \lambda^+ \\ \hline c^+ \\ g^+ \end{pmatrix} = \{f(x^+) - Q_{old}(x^+)\} H^+ e_t, \tag{3.34}$$

où e_t est maintenant dans $\mathbb{R}^{m+n+1}$. L'algorithme utilise donc (3.33) et (3.34) pour générer la fonction D par la formule de mise à jour

$$Q_{new}(x) = Q_{old}(x) + D(x), \quad x \in \mathbb{R}^n. \tag{3.35}$$

La mise à jour de chacun des modèles des contraintes sans dérivées est basée sur la même idée que l'équation (3.34) adaptée à chacune des contraintes :

$$\begin{pmatrix} \lambda^+ \\ \hline c^+ \\ g^+ \end{pmatrix} = \{C_{DF_{(i)}}(x^+) - \tilde{C}_{DF_{(i)}}(x^+)\} H^+ e_t, \tag{3.36}$$

où $\tilde{C}_{DF_{(i)}}$ est le modèle avant sa mise à jour. Les explications des mises à jour de H^+, de la Hessienne $\nabla^2 Q_{new} = \nabla^2 Q_{old} + \nabla^2 D$ et du gradient $\nabla Q_{new}(x_0) = \nabla Q_{old}(x_0) + g^+$ sont détaillées dans [139]. En effet, pour éviter d'inverser la matrice W lors de chaque itération, la matrice H est simplement stockée et mise à jour en fonction des paramètres α, β, τ et σ

$$\begin{aligned} &\alpha = e_t^T H e_t, \quad \beta = \tfrac{1}{2}\|x^+ - x_0\|^4 - w^T H w, \\ &\tau = e_t^T H w, \quad \text{et } \sigma = \alpha\beta + \tau^2 (\text{voir l'équation (3.27)}), \end{aligned} \tag{3.37}$$

où w provient de l'équation (3.28). La mise à jour de H dépend de l'inverse de σ. Des difficultés peuvent donc survenir en présence d'éléments trop grands de H^+ dans le cas où $|\sigma_t|$ est trop petit [138]. La phase (2)(a)(iv) est suivie de la phase (2)(a)(i) et une autre itération de région de confiance est effectuée si le rapport de l'amélioration effective sur l'amélioration prédite R (3.23) est suffisamment important.

2.2.3 Amélioration du modèle

Lorsque la réduction prédite par la fonction de mérite des modèles $\tilde{\varphi}(x_{opt}) - \tilde{\varphi}(x_{opt} + d_k)$ s'avère être une mauvaise prédiction du comportement de la fonction de mérite $\varphi(x_{opt}) - \varphi(x_{opt} + d_k)$, la phase (2)(a)(iv) est suivie de la phase (2)(b). En pratique, les positions des points x_i pour les équations d'interpolation (3.6) ne suffisent pas à maintenir un "bon" modèle quadratique au sens de l'unisolvance, surtout lorsque les itérations de région de

confiance ont fait augmenter les distances $\|x_i - x_{opt}\|, i = 1, 2, ..., m$ à tel point qu'elles soient supérieures à Δ.
La phase (2)(b) permettra alors d'identifier le point d'interpolation courant x_{MOVE} qui est le plus éloigné de x_{opt}. Si $\|x_{MOVE} - x_{opt}\| \geq 2\Delta$, alors les modèles Q et $\tilde{C}_{DF}$ peuvent être améliorés en remplaçant les conditions d'interpolation $Q(x_{MOVE}) = f(x_{MOVE})$ et $\tilde{C}_{DF}(x_{MOVE}) = C_{DF}(x_{MOVE})$ respectivement par $Q(x_{opt}+d_k) = f(x_{opt}+d_k)$ et $\tilde{C}_{DF}(x_{opt}+d_k) = C_{DF}(x_{opt} + d_k)$, pour un pas d_k satisfaisant $\|d_k\| \leq \Delta$. Le choix de d_k est fait lors de la phase 2(b)(i) dans le but d'aider le conditionnement du système linéaire qui permet le calcul de Q.
Le choix de d_k dépend de la position x_i des points d'interpolation courants, mais il dépend aussi des erreurs qui surviennent dans la matrice H lors de la mise à jour du modèle à chaque itération. Le t-ième polynôme de Lagrange des points d'interpolation courants ($t = MOVE$ pour garder les notations précédentes), polynôme de Lagrange associé au point d'interpolation le plus éloigné, est important, car il s'agit du polynôme permettant de maintenir la qualité du modèle. Il s'agit du polynôme quadratique $\ell_t(x)$, $x \in \mathbb{R}^n$ qui satisfait les conditions de Lagrange

$$\ell_t(x_i) = \delta_{it}, \quad i = 1, 2, ..., m, \tag{3.38}$$

(les degrés de liberté du polynôme quadratique sont complétés par la minimisation de la norme de Frobenius $\|\nabla^2 \ell_t\|_F$ lorsque $m < (n+1)(n+2)/2$). ℓ_t est la fonction

$$\ell_t(x) = c + (x - x_0)^T g + \tfrac{1}{2} \sum_{k=1}^{m} \lambda_k \{(x - x_0)^T (x_k - x_0)\}^2, \quad x \in \mathbb{R}^n, \tag{3.39}$$

où c, g et λ_k, $k = 1, 2, ..., m$, sont définis par le système d'équations suivant avec $e_t \in \mathbb{R}^{m+n+1}$ $(1 \leq t \leq m)$

$$\left(\begin{array}{c|c} A & X^T \\ \hline X & 0 \end{array}\right) \left(\begin{array}{c} \lambda \\ \hline c \\ g \end{array}\right) = e_t. \tag{3.40}$$

Ces polynômes permettent de définir le modèle de la fonction et des contraintes sans dérivées :

$$Q(x) = \sum_{i=1}^{m} f(x_i)\ell_i(x), \; \tilde{C}_{DF} = \sum_{i=1}^{m} C_{DF}(x_i)\ell_i(x). \tag{3.41}$$

Soit $w(x) \in \mathbb{R}^{m+n+1}$

$$\begin{array}{l} w(x)_k = \frac{1}{2}(x - x_0)^T (x_k - x_0)^2, \quad k = 1, 2, ..., m \\ w(x)_{m+1} = 1 \text{ et } w(x)_{i+m+1} = (x - x_0)_i, \quad i = 1, 2, ..., n \end{array} \tag{3.42}$$

d'où l'expression (3.39) qui peut être ré-écrite sous la forme

$$\ell_t(x) = \sum_{k=1}^{m} \lambda_k w(x)_k + c w(x)_{m+1} + \sum_{i=1}^{n} g_i w(x)_{i+m+1} = (He_t)^T w(x). \tag{3.43}$$

Ainsi, après avoir mis à jour H, à partir du changement des points d'interpolation (3.30), l'expression (3.37) de τ donne

$$\tau = e_t^T H w = e_t^T H w(x^+) = (He_t)^T w(x_{opt} + d) = \ell_t(x_{opt} + d). \tag{3.44}$$

Afin d'avoir $|\sigma| = |\alpha\beta + \tau^2|$ assez grand pour la formule de mise à jour de H, nous allons maximiser $|\tau|$ (en théorie α et β sont positifs [138]). D'où

$$\begin{aligned} &\max_{d \in \mathbb{R}^n} |\ell_t(x_{opt} + d)| \\ &\text{s.t.} \begin{cases} \|d\| \leq \overline{\Delta}, \\ C_{DB}(x_{opt} + d) \leq 0, \end{cases} \end{aligned} \tag{3.45}$$

où $\overline{\Delta} > 0$ est déterminé de la manière suivante [139] :

$$\overline{\Delta} = \max[\min\{0.1DIST, 0.5\Delta\}, \rho], \tag{3.46}$$

où $DIST = \|x_t - x_{opt}\|$, pour conserver la distance maximale entre les points d'interpolation et x_{opt} proche de ρ.
Cette opération consistant à maximiser le polynôme de Lagrange associé au point le plus éloigné permet de maximiser la contribution du point supprimé, et donc d'augmenter l'information prise en compte par le modèle. Dans l'algorithme SQA, la sélection de t est basée sur la distance entre le point d'interpolation t et le centre de la région de confiance plutôt que d'effectuer l'optimisation globale de tous les polynômes de Lagrange et de sélectionner l'indice du polynôme pour lequel la valeur absolue est la plus grande (voir la section 3 du chapitre 2). Cela permet d'éviter de nombreuses optimisations globales des polynômes de Lagrange et une certaine flexibilité dans la sélection du point à remplacer. La première optimisation $\max |\tau|$ est effectuée sous contrainte à l'aide de la méthode d'optimisation SQP, utilisée pour résoudre le sous problème de minimisation du modèle quadratique de SQA. Enfin, après avoir calculé la valeur de la fonction $f(x_{opt} + d_k)$, une mise à jour du modèle Q est effectuée. Puis l'algorithme retourne toujours à la phase (2)(a)(i) pour minimiser le nouveau modèle quadratique.

2.2.4 Evolution de ρ

Si, contrairement au cas de la section précédente, les positions des points courants $x_i, i = 1, 2, ..., m$, vérifient

$$\|x_i - x_{opt}\| < 2\Delta, i = 1, 2, ..., m, \tag{3.47}$$

c'est-à-dire que le modèle est considéré valide, alors la phase (2)(b)(ii) suit le test de validité. Cette étape détermine si la région minimale ρ actuelle nécessite d'être réduite. Celle-ci ne l'est pas si et seulement si une ou plusieurs des conditions $\|d_k\| > \rho$, $\Delta > \rho$ ou $R > 0$ sont vérifiées. Une autre itération de région de confiance est exécutée avec la même région minimale ρ dans les deux premiers cas, car ρ n'a pas restreint le choix le

plus récent de d. Dans le troisième cas, $R > 0$ implique $\varphi(x_{opt} + d_k) < \varphi(x_{opt})$, l'ancien ρ est retenu lorsqu'une réduction stricte de la fonction de mérite a été obtenue.

Finalement, nous considérons le cas où le pas d_k obtenu lors de la minimisation du modèle quadratique de l'étape (2)(a)(i) satisfait $\|d_k\| < \frac{1}{2}\rho$. Dans ce cas, $f(x_{opt} + d_k)$ et $C_{DF}(x_{opt} + d_k)$ ne sont pas évalués car les différences $f(x_{opt}) - f(x_{opt} + d_k)$ et $C_{DF}(x_{opt}) - C_{DF}(x_{opt} + d_k)$ peuvent donner de trompeuses informations sur le comportement de f et C_{DF} dans la région de confiance de rayon Δ, $\|d_k\|$ étant trop petit. Une grande réduction de Δ est alors effectuée tout en maintenant $\Delta \geq \rho$

$$\Delta = \begin{cases} \max(\Delta/10, \rho), & \text{si } \Delta > 1.5\rho, \\ \rho, & \text{si } \Delta \leq 1.5\rho. \end{cases} \tag{3.48}$$

Dans ce cas, le rapport de l'amélioration effective sur l'amélioration prédite est fixé à $R = -1$. On passe alors soit :

- à la phase (2)(b). Comme précédemment, on choisit entre remplacer le point d'interpolation x_{MOVE} pour améliorer le modèle, exécuter une itération de région de confiance avec le nouveau Δ, ou exécuter l'étape (2)(c) car l'évolution de la région de confiance avec le ρ courant est achevée.
- à la phase (2)(c). En effet, il n'est pas nécessaire d'améliorer le modèle quadratique Q si au moins trois nouvelles valeurs de f ont été calculées pour la région minimale ρ courante, et si

$$\|d^{(j)}\| \leq \rho \quad \text{et} \quad |Q_j(x_{opt}^{(j)} + d^{(j)}) - f(x_{opt}^{(j)} + d^{(j)})| \leq \frac{1}{8}\rho^2 CRVMIN, \quad j \in \mathcal{J}, \tag{3.49}$$

  où Q_j, $d^{(j)}$ et $x_{opt}^{(j)}$ sont Q, d et x_{opt} de la phase (2)(a)(iii) de la j-ième itération, $CRVMIN$ est calculé à la génération courante, et $\mathcal{J}$ représente les itérations des trois derniers passages de l'algorithme dans la phase (2)(a)(iii) avant l'itération courante. Enfin ρ peut être modifié bien que quelques distances $\|x_i - x_{opt}\|$, $i = 1, 2, ..., m$, soient supérieures à 2ρ.

Dans la phase (2)(c), ρ est mis à jour si $\rho > \rho_{end}$:

$$\rho_{new} = \begin{cases} \rho_{end}, & \rho_{old} \leq 16\rho_{end}, \\ (\rho_{old}\rho_{end})^{1/2}, & 16\rho_{end} < \rho_{old} \leq 250\rho_{end}, \\ 0.1\rho_{old}, & \rho_{old} > 250\rho_{end}. \end{cases} \tag{3.50}$$

En revanche, si $\rho = \rho_{end}$, la phase (3) calcule $f(x_{opt} + d)$ et $C_{DF}(x_{opt} + d)$ si cela n'a pas été effectué et l'algorithme est arrêté. En effet, le critère de convergence de cet algorithme est basé sur le rayon minimal de la région de confiance minimale.

2.2.5 Paramètres par défaut

Il existe des paramètres par défaut utilisés dans cette méthode : les rayons permettant de maintenir l'écart entre les points à l'initialisation $\rho_{beg} = 1$, ou celui utilisé comme critère d'arrêt $\rho_{end} = 10^{-6}$. Mais aussi le nombre constant de points d'interpolation $m = 2n + 1$. Pour les contraintes, le poids de pénalisation est fixe au cours des itérations, celui-ci est fixé à 100.

2.3 Convergence de la méthode sans contraintes

Cette section présente rapidement les grandes différences entre la méthode SQA et la méthode présentée dans le chapitre 2.

Pour prouver la convergence de la méthode de Powell, NEWUOA, c'est à dire SQA sans contraintes, certaines modifications ont été proposées par Conn et al [34]. La méthode SQA identifie en plus du rayon de la région de confiance, au contraire des algorithmes tels que présentés dans le chapitre 2, un rayon minmal de la région de confiance. Le rayon de la région de confiance et le rayon de l'ensemble des points d'interpolation sont dissociés et évoluent chacun de leur côté. Le rayon de l'ensemble des points d'interpolation converge vers 0 alors que la valeur du rayon de la région de confiance peut diverger de celle du rayon de l'ensemble d'interpolation. Cela permet d'effectuer des pas assez grands pour une convergence éventuellement plus rapide. Le plus petit des deux rayons force les points d'interpolation à être suffisament éloignés pour éviter l'influence du bruit sur les valeurs de f. L'étape de mise à jour de ces rayons, de région de confiance et de région minmale, est logiquement plus ardue que celle du chapitre 2.

L'algorithme nécessite l'ajout d'une étape critique (comme l'algorithme 2.4) qui permet d'assurer d'avoir un modèle totalement linéaire à chaque itération. De manière analogue pour la méthode de Powell ou notre méthode, la géométrie est améliorée lorsque le pas d_k est plus petit que Δ_k. L'algorithme requiert également la modification de l'étape d'amélioration de modèle permettant d'assurer un modèle totalement linéaire en un nombre fini d'étapes.

Nous invitons le lecteur à lire l'article de Powell [143] qui présente une preuve de la convergence de ses méthodes sans contraintes. A notre connaissance, aucun résultat de convergence sur des méthodes proches de SQA sur des problèmes sans contraintes n'a été démontré.

3 Exemple de l'évolution des paramètres de l'algorithme sur un cas simple

Soit la fonction objectif suivante, extraite de la thèse de Vanden Berghen [167] représentée sur la figure 3.1.

$$f(x_1, x_2) = -10x_1^2 + 10x_2^2 + 4\sin(x_1x_2) - 2x_1 + x_1^4 \tag{3.51}$$

Les deux minima de cette fonction sont atteints en $x = \begin{pmatrix} -2.210220 \\ 0.329748 \end{pmatrix}$ avec $f(x) = -22.143$ et en $x = \begin{pmatrix} 2.306630 \\ -0.332309 \end{pmatrix}$ avec $f(x) = -31.181$.

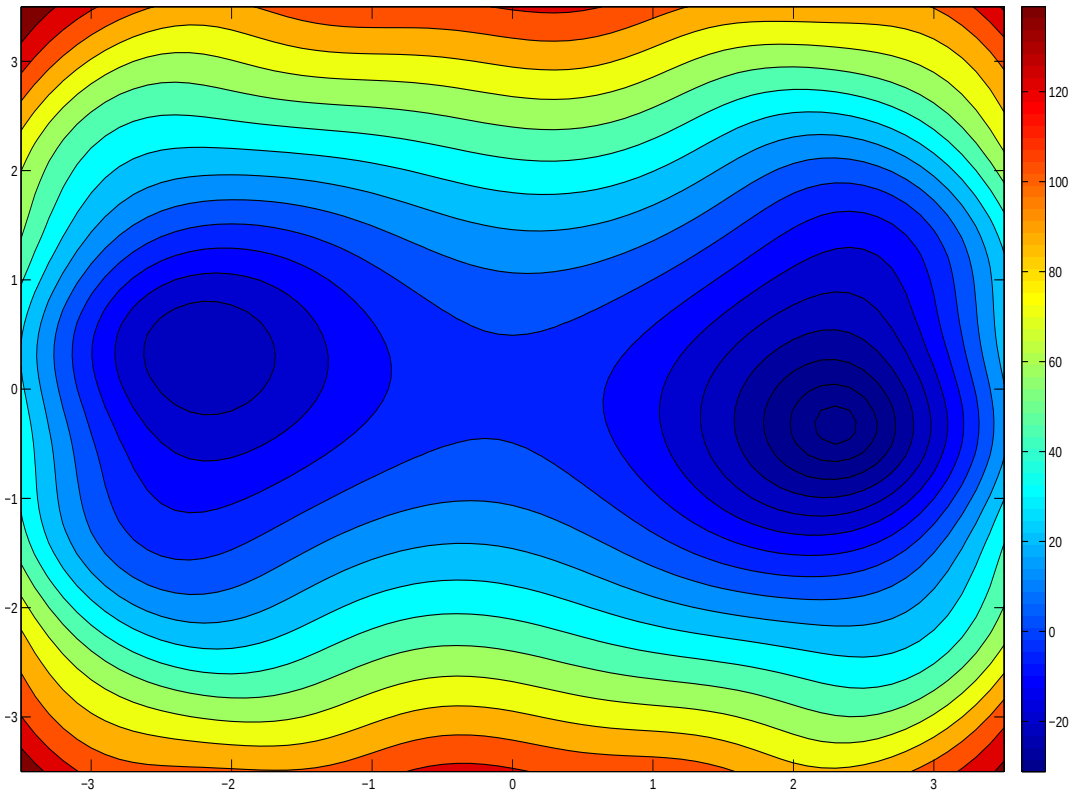


FIGURE 3.1 – Fonction utilisée en exemple

Nous nous intéressons à la minimisation de cette fonction (représentée par les lignes de niveaux noires sur les figures 3.3, 3.4, 3.7 et 3.8) avec une contrainte non linéaire (représentée par la courbe noire, le domaine admissible étant la zone blanche sur les figures 3.3, 3.4, 3.7 et 3.8) :

$$\begin{array}{l} \min\limits_{x_1,x_2\in\mathbb{R}} f(x_1,x_2) \\ \text{s.t.} \left\{ \begin{array}{l} x_1, x_2 \in [-4.3, 4.3], \\ 0 \leq -(x_1+0.6)^3 + \frac{\sin(x_1)}{100} + \frac{1}{2} - x_2, \end{array} \right. \end{array} \tag{3.52}$$

Cette contrainte est tout d'abord prise en compte comme une contrainte avec dérivées puis comme une contrainte sans dérivées dans deux optimisations différentes. Les figures 3.3, 3.4, 3.7 et 3.8 illustrent l'évolution des rayons des deux régions (minimale ρ et de confiance Δ), des points d'interpolation, ainsi que l'évolution du modèle quadratique interpolant la fonction objectif (3.51) au cours des itérations de deux optimisations.

Le point de départ de la minimisation de la fonction objectif est : $x = \begin{pmatrix} 1.7 \\ -3.3 \end{pmatrix}$.

3.1 Exemple sous contrainte avec dérivées

La contrainte non linéaire est traitée comme une contrainte avec dérivées dans cette section.

La figure 3.2 montre la décroissance de la fonction objectif au cours des itérations. Nous distinguons les évaluations de minimisation du modèle (croix rouge) de celles d'amélioration du modèle (croix verte). Nous remarquons qu'une suite d'amélioration du modèle et de réduction de région minimale ρ de la région de confiance est nécessaire pour s'assurer de la convergence de la méthode. Lorsqu'on se trouve dans le voisinage de la solution, de nombreuses étapes d'amélioration du modèle sont nécessaires afin de s'assurer de la convergence.

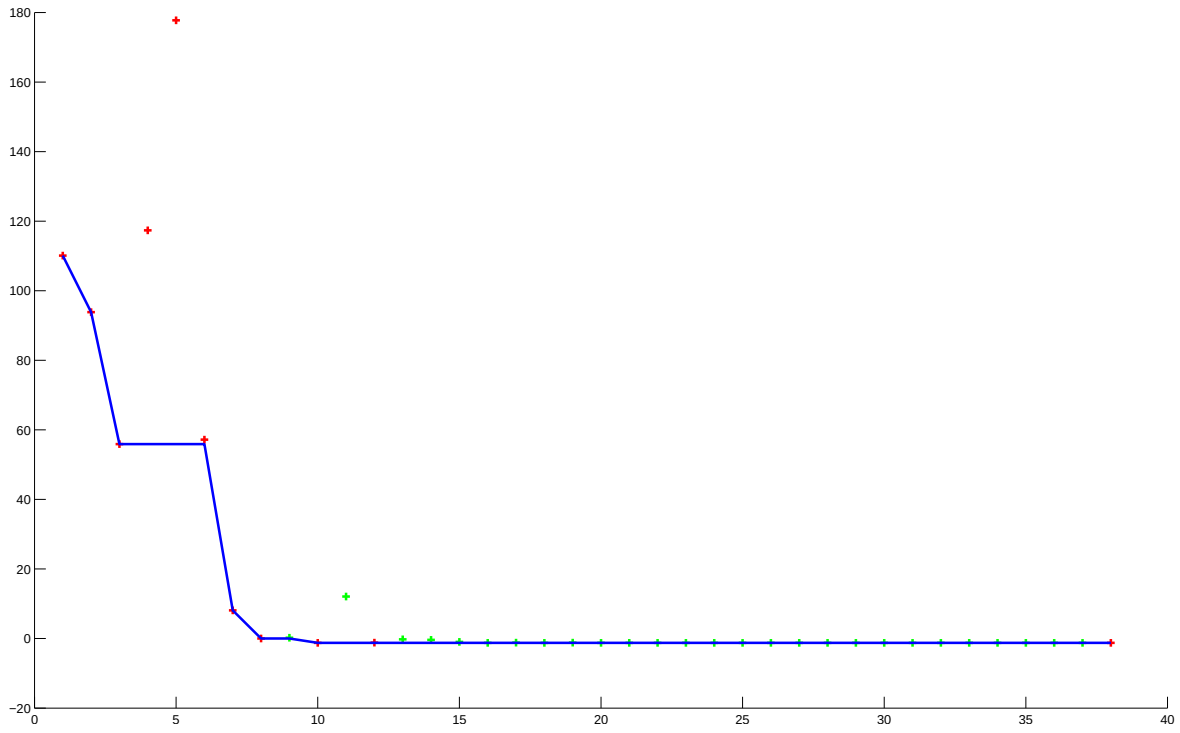


FIGURE 3.2 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour la minimisation (3.52) sous contrainte avec dérivées avec SQA pour un nombre de points d'interpolation de 5 $(2n+1)$

Sur chacune des figures 3.3 et 3.4 sont illustrés les points d'interpolation de l'itération courante (étoiles bleues) ainsi que le modèle quadratique, représenté par des lignes de niveau en couleur, dans la région de confiance de rayon Δ (cercle bleu). Le plus petit cercle en pointillé correspont à celui de rayon ρ (résolution minimale demandée à l'itération courante).

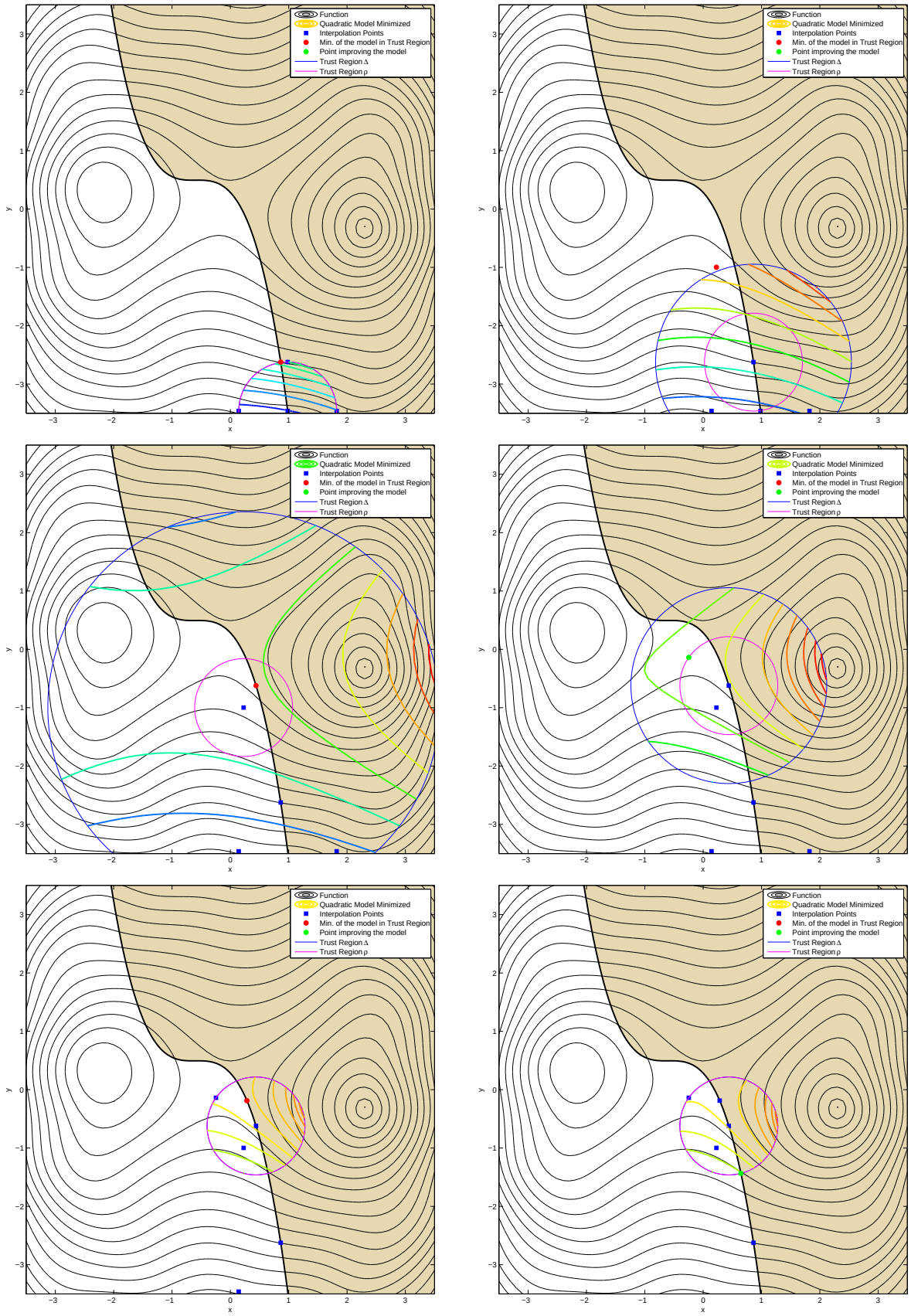


FIGURE 3.3 – Evolution du modèle et de la région de confiance de rayon Δ et de la région minimale ρ lors des premières itérations de SQA avec la contrainte considérée comme une contrainte avec dérivées (correspondant aux 11 premières évaluations)

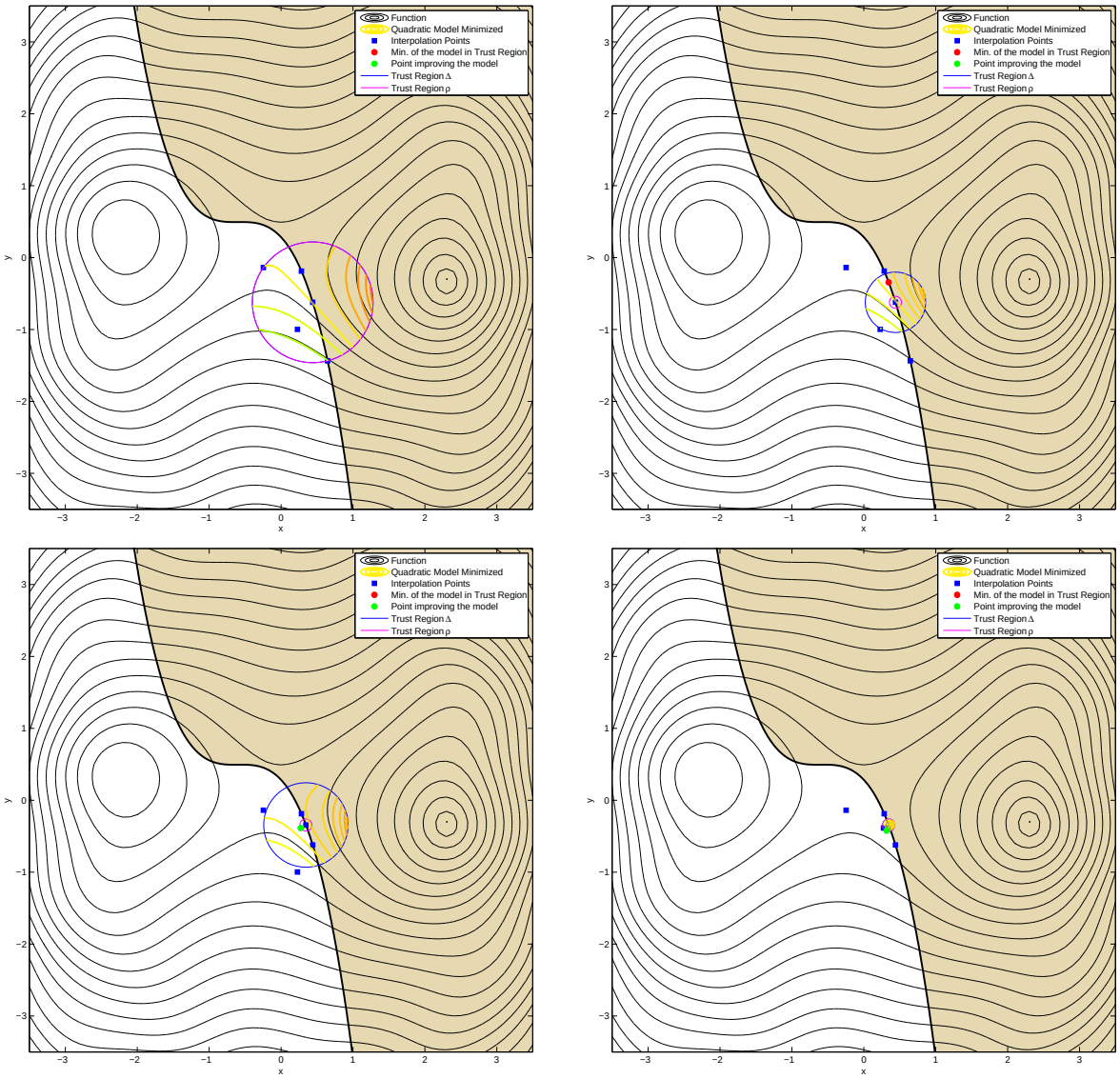


FIGURE 3.4 – Evolution du modèle et la région de confiance de rayon Δ et de la région minimale ρ lors des itérations de SQA avec la contrainte considérée comme une contrainte avec dérivées (correspondant aux évaluations 12 à 14)

Remarquons que le point de départ de l'optimisation a été modifié afin de vérifier la contrainte avec dérivées. En revanche, les autres évaluations initiales (autour du point de départ de l'optimisation, voir l'annexe E) permettant de construire le premier modèle quadratique de la fonction objectif ne vérifient pas tous la contrainte.

Le minimum du modèle quadratique dans la région de confiance obtenu à l'étape de minimisation du modèle de la section 2.2.2 de ce chapitre (étoile rouge) est évalué si sa distance au point optimal de l'itération courante (centre de la région de confiance) est supérieure à $\frac{1}{2}\rho$, soit la moitié du rayon ρ du cercle représenté en pointillés de couleur violette sur les figures, maintenant une distance minimale entre les points d'interpolation afin de limiter les erreurs du modèle lorsque la fonction est bruité et de contrôler la vitesse de convergence du modèle. Si le minimum du modèle dans la région de confiance est trop

proche du centre de la région de confiance (point optimal courant) ou si le rapport de l'amélioration effective sur l'amélioration prédite R (3.23) est inférieur à un seuil, un point permettant d'enrichir le modèle obtenu après l'étape d'amélioration du modèle de la section 2.2.3 de ce chapitre (étoile verte) peut être évalué si le test de validité du modèle le permet. Entre chaque itéré le rayon de la région de confiance Δ et le rayon minimal ρ sont mis à jour (voir la section 2.2.3 de ce chapitre).

Nous remarquons que la contrainte avec dérivées est vérifiée au cours de toutes les itérations de la méthode (après l'initialisation). La contrainte étant prise en compte explicitement dans la résolution du sous-problème de l'étape (2)(a)(i) de l'algorithme 3.1, ce qui ne sera pas le cas pour une contrainte sans dérivées.

La figure 3.5 illustre l'évolution du rayon de la région de confiance Δ et du rayon ρ maintenant l'écart entre les points d'interpolation. Le rayon de la région de confiance peut augmenter si le rapport de l'amélioration effective sur l'amélioration prédite R (3.23) annonce un bon modèle. Le cercle de rayon ρ peut parfois être confondu avec la région de confiance de rayon Δ et ne peut que décroître.

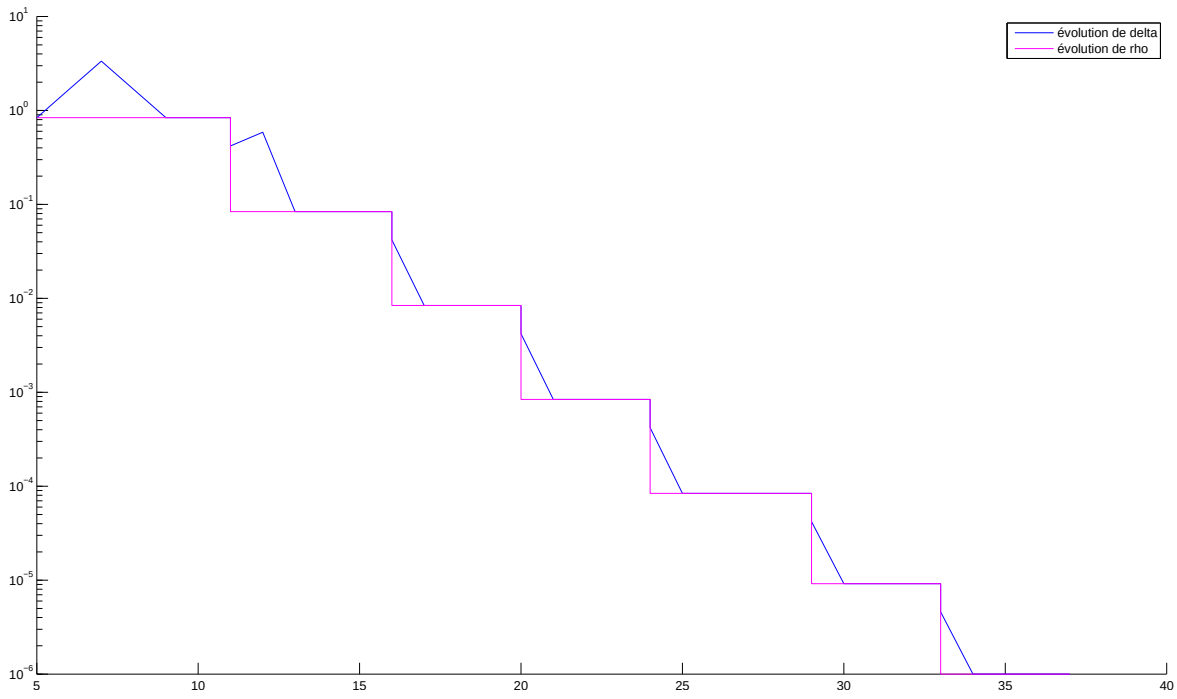


FIGURE 3.5 – Evolution du rayon de la région de confiance Δ et du rayon minimal ρ au cours des itérations

3.2 Exemple sous contrainte sans dérivées

La contrainte non linéaire est maintenant traitée comme une contrainte sans dérivées dans cette section.

La figure 3.6 montre la décroissance de la fonction objectif au cours des itérations. Nous distinguons les évaluations de minimisation du modèle (croix rouge) de celles d'amélioration du modèle (croix verte). Nous remarquons, comme pour l'optimisation précédente (section 3.1), qu'une suite d'amélioration de modèle et de réduction de région minimale ρ de la région de confiance est nécessaire pour s'assurer de la convergence de la méthode.

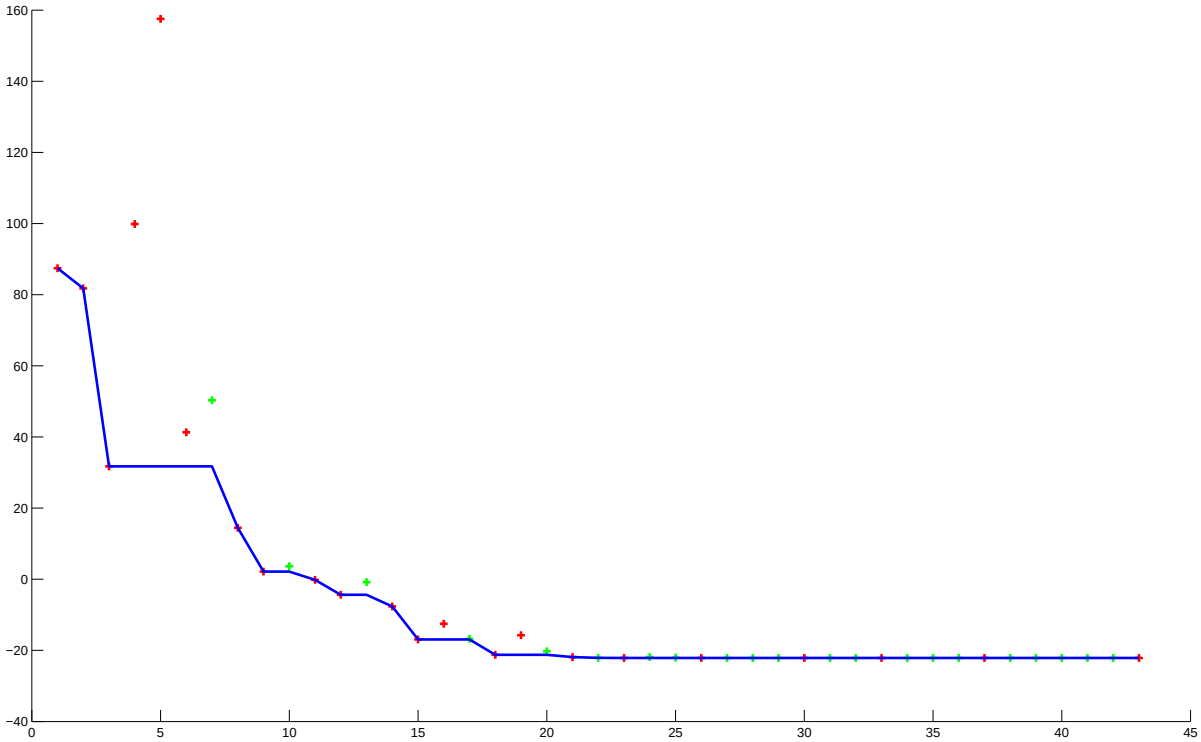


FIGURE 3.6 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour la minimisation (3.52) sous contrainte sans dérivées avec SQA pour un nombre de points d'interpolation de 5 $(2n+1)$

Les figures 3.7 et 3.8 sont illustrées, comme dans la section précédente, les points d'interpolation de l'itération courante (étoiles bleues) ainsi que le modèle quadratique, représenté par des lignes de niveau en couleur, dans la région de confiance de rayon Δ (cercle bleu). Le plus petit cercle en pointillé correspont à celui de rayon ρ (résolution minimale demandée à l'itération courante). La borne du modèle de la contrainte sans dérivées est également représentée sur ces figures par la courbe rouge.

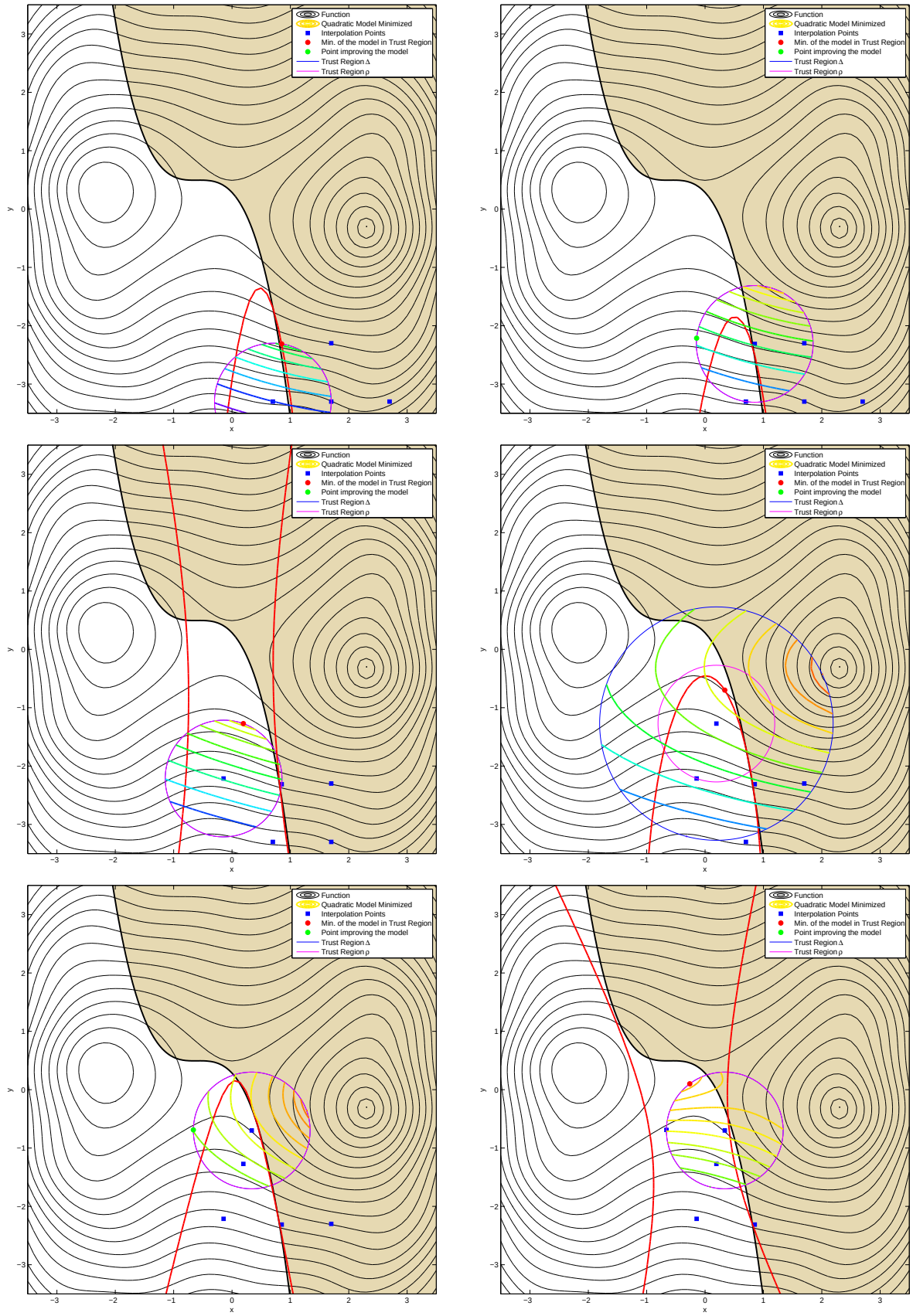


FIGURE 3.7 – Evolution du modèle et de la région de confiance de rayon Δ et de la région minimale ρ lors des premières itérations de SQA avec la contrainte considérée comme une contrainte sans dérivées (correspondant aux 11 premières évaluations)

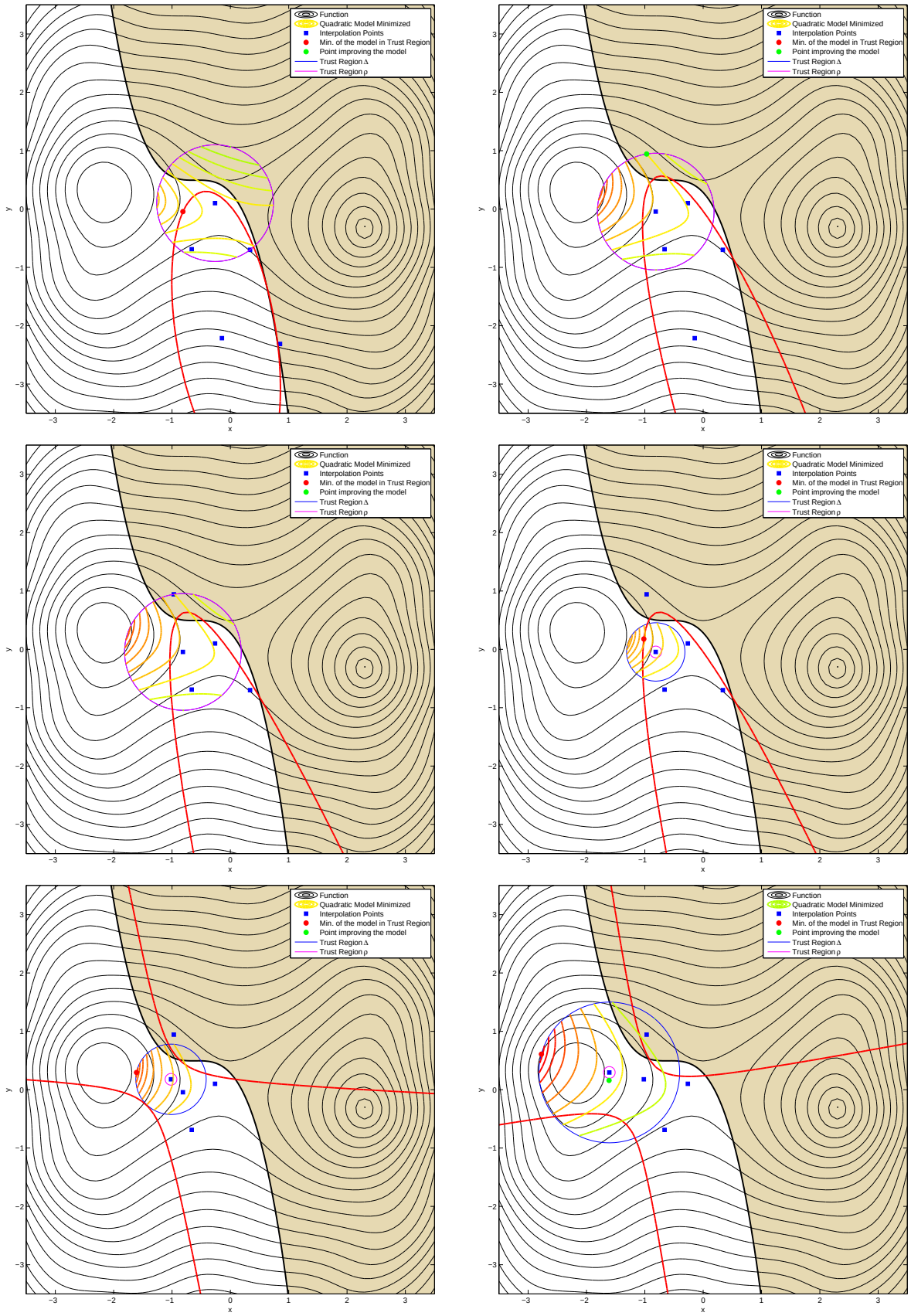


FIGURE 3.8 – Evolution du modèle et de la région de confiance de rayon Δ et de la région minimale ρ lors des itérations de SQA avec la contrainte considérée comme une contrainte avec dérivées (correspondant aux évaluations 12 à 17)

La contrainte sans dérivées est mise à jour à chaque itération à l'aide des mêmes points d'interpolation que ceux permettant de construire le modèle quadratique de la fonction objectif. Notons que la contrainte sans dérivées n'est pas toujours vérifiée au cours des itérations. En effet, seul le modèle de celle-ci est pris en compte lors de la minimisation du modèle de la fonction objectif (ou de la fonction de mérite dans certain cas), et le modèle de celle-ci n'est pas pris en compte lorsqu'une amélioration du modèle est effectuée. La solution trouvée est différente de celle de la section 3.1 précédente. Le problème précédent est résolu plus précisément, la solution trouvée est le minimum local le plus proche du point de départ. Lorsque la dérivée de la contrainte n'est pas connu, la convergence est moins rapide et a permis de voir la courbure conduisant à un autre minimum.

La figure 3.9 illustre l'évolution du rayon de la région de confiance Δ et du rayon ρ maintenant l'écart entre les points d'interpolation. L'évolution est légérement différente de celle des résultats pour l'optimisation sous contrainte avec dérivées précédents. Certaines étapes de l'algorithme à ρ fixé sont plus coûteuses en évaluations (le nombre total d'évaluations est légèrement plus important que pour l'optimisation sous contrainte avec dérivées).

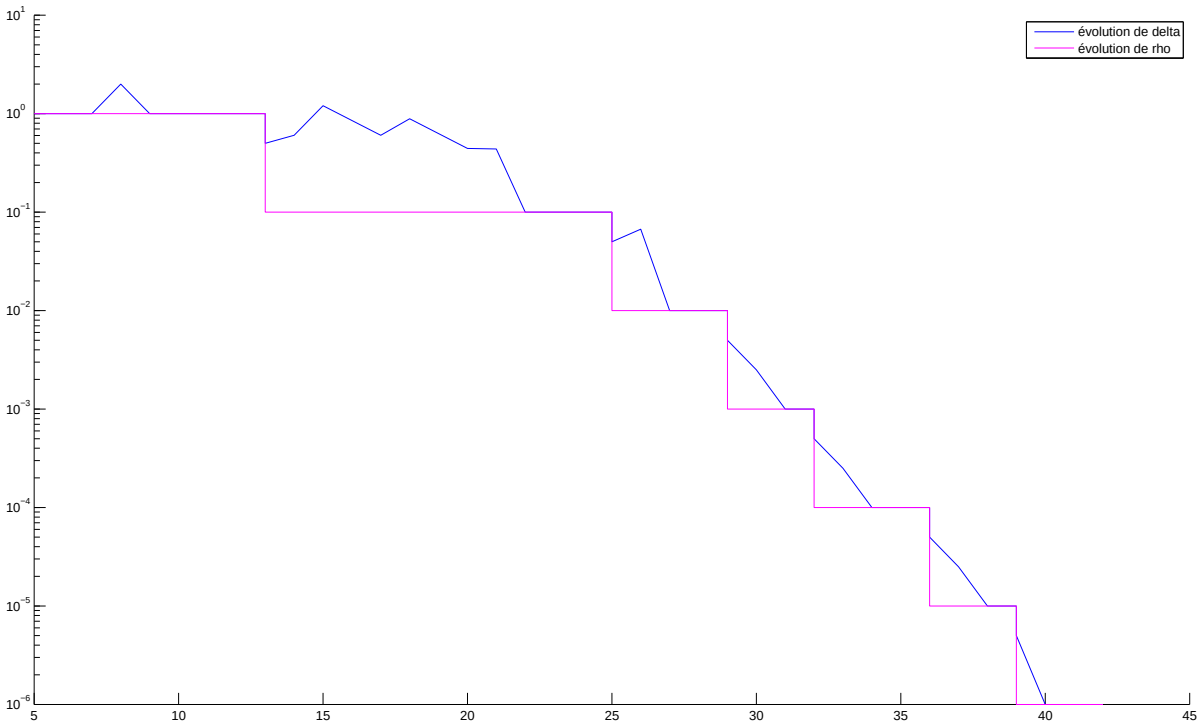


FIGURE 3.9 – Evolution du rayon de la région de confiance Δ et du rayon minimal ρ au cours des itérations

4 Améliorations de la méthode et perspectives

Plusieurs améliorations de notre méthode SQA ont été envisagées ou effectuées, mais n'ont pas encore été validées totalement.

4.1 Prise en compte de contraintes d'égalité

Nous pourrions enrichir la méthode par la prise en compte de contraintes avec dérivées d'égalités linéaires ou non linéaires, ce qui reviendrait à prendre en compte les contraintes d'égalité lors de la minimisation du modèle quadratique dans la région de confiance :

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} Q(x_{opt}+d) \\ &\text{s.t.} \begin{cases} \|d\| \leq \Delta, \\ C_{DB}(x_{opt}+d_k) \leq 0, \\ \tilde{C}_{DF}(x_{opt}+d_k) \leq 0, \\ c_e(x) = 0, \end{cases} \end{aligned} \tag{3.53}$$

où $c_e(x)$ sont les contraintes d'égalité.

4.2 Prise en compte du bruit

Cette méthode est assez peu sensible au bruit provenant de la fonction objectif, car le paramètre ρ permet de maintenir les points d'interpolation assez éloignés les uns des autres afin que le bruit n'ait pas trop d'influence sur le modèle. Toutefois la méthode peut être perfectionnée afin d'être plus robuste au bruit lorsque celui-ci peut être quantifié. Lors de la minimisation du modèle quadratique de l'étape (2)(a)(i), si l'amélioration prédite $\tilde{\varphi}(x_{opt}) - \tilde{\varphi}(x_{opt}+d)$ est inférieure au bruit défini par (3.54) alors la fonction n'est pas évaluée en $x_{opt}+d$ et le test de validité de l'étape (2)(b) suit.

On peut définir le bruit comme suit :

$$noise = \tfrac{1}{2}\max[noise_r|f(x_{opt})|, noise_a \times (1+noise_r)], \tag{3.54}$$

où $noise_r$ et $noise_a$ sont respectivement l'erreur relative et l'erreur absolue sur la fonction objectif. Cette heuristique a été testée dans la section 1.1.2 du chapitre 4.

4.3 Fonction objectif de type moindres carrés

Cette méthode n'est pas très adaptée pour l'optimisation de fonction de moindres carrés (3.55), or ce cas de figure apparaît assez régulièrement dans le monde de l'industrie (voir la section 2.1.1 du chapitre 4) pour le calage de simulateurs sur des données expérimentales par exemple. On considère le problème de mooindre carrés

$$\min \Phi(x) = \frac{1}{2}\sum_{i=1}^{n_{ls}} f_i^2(x), \tag{3.55}$$

où $f_i(x) : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., n_{ls}$ sont des fonctions non linéaires.

L'idée introduite par Zhang et al [185] pour prendre en compte cette caractéristique est de construire un modèle pour chacune de ces fonctions f_i. Notons $m_i(\cdot)$, $i = 1, ..., n_{ls}$ les modèles d'interpolation quadratiques des $f_i(\cdot)$.

Cette idée semble intéressante mais n'est pas évidente en pratique en raison du nombre pouvant être grand n_{ls} de fonctions non linéaires formant la fonction de moindres carrés à minimiser Φ. C'est pourquoi nous avons pensé à grouper les f_i par paquet pour traiter la fonction de moindres carrés.

Pour notre part, nous avons testé deux alternatives différentes :

- la modélisation (nommée LS1) des

$$\sum_{i=0}^{n_j} f_i^2 \tag{3.56}$$

- et la modélisation (nommée LS2) des

$$\sqrt{\sum_{i=0}^{n_j} f_i^2}, \tag{3.57}$$

pour n_j le nombre de données utilisées pour construire le modèle m_j, $j = 1, 2, ..., n_m$, où

$$\sum_{j=1}^{n_m} n_j = n_{ls},$$

où n_m est le nombre de modèles construits. La fonction minimisée par la méthode SQA (voir la section 2.2.2 de ce chapitre) est alors :

- pour LS1 :

$$M_\Phi(x) = \sum_{j=1}^{n_m} m_j(x) \tag{3.58}$$

- pour LS2

$$M_\Phi(x) = \sum_{j=1}^{n_m} m_j^2(x). \tag{3.59}$$

Notons que si n_{ls} n'est pas trop grand, nous pouvons construire un modèle pour chacune des fonctions f_i (soit $n_m = n_{ls}$), comme suggéré par Zhang et al.. Nos deux alternatives reviennent alors à :

- la modélisation des f_i^2 (LS1) et
- la modélisation des f_i (LS2).

Nous avons traité les trois exemples suivants :

-

$$\begin{cases} f_1 = \exp(-x_1) + x_2 - 1 \\ f_2 = \exp(-x_1) + x_2 - 4 \end{cases} \tag{3.60}$$
$$\Phi = \textstyle\sum_{i=1}^{n_{ls}=2} f_i^2(x)$$

∘

$$\begin{cases} f_1(x) = (x_1 - 1) * (x_1 - 1) + (x_2 - 2) * (x_2 - 2) + 3 * x_1 * x_2 \\ f_2(x) = (x_1 + 1) * (x_1 + 1) + x_2 * x_2 \end{cases} \\ \Phi = \sum_{i=1}^{n_{ls}=2} f_i^2(x) \tag{3.61}$$

∘

$$\begin{gathered} \forall i = 1, n_{ls} = 16, \text{ avec } t_i = \frac{i-1}{10} \\ f_i(x) = x_1 * |x_2|^{t_i} * \tanh(x_3 * t_i + \sin(x_4 * t_i)) * \cos(t_i * \exp(x_5)) \\ -53.81 * 1.27^{t_i} + \tanh(3.012 * t_i + \sin(2.13 * t_i)) * \cos(t_i * \exp(0.507)) \\ \Phi = \sum_{i=1}^{n_{ls}=16} f_i^2(x) \end{gathered} \tag{3.62}$$

Les deux fonctions (3.60) et (3.61) ont été minimisées à l'aide de la méthode SQPAL Gauss Newton [44, 159] ainsi que par la méthode SQA avec trois différentes approches : l'approche classique et les deux approches permetant de prendre en compte le caractère moindres carrés de la fonction objectif (LS1 (3.56) et LS2 (3.57), avec un modèle pour chaque fonction $n_m = n_{ls} = 2$) pour deux différents nombres de points d'interpolation : $2n + 1$ et $(n + 1)(n + 2)/2$. La fonction (3.62) a été minimisée à l'aide de la méthode SQPAL Gauss Newton ainsi que par la méthode SQA avec différentes approches pour un nombre de points d'interpolation de $2n + 1$:

- l'approche classique,
- l'approche LS1 construisant un modèle pour chaque fonction ($n_m = n_{ls} = 16$),
- l'approche LS2 construisant un modèle pour chaque fonction ($n_m = n_{ls} = 16$),
- l'approche LS1 construisant $n_m = 4$ modèles, ces modèles étant construits à partir de quatre f_i chacun, et
- l'approche LS2 construisant $n_m = 4$ modèles, ces modèles étant également construits à partir de quatre f_i.

Sur les figures 3.10, 3.11 et 3.12, nous indiquons les valeurs de la fonction objectif aux cours des itérations pour les différentes méthodes utilisées.

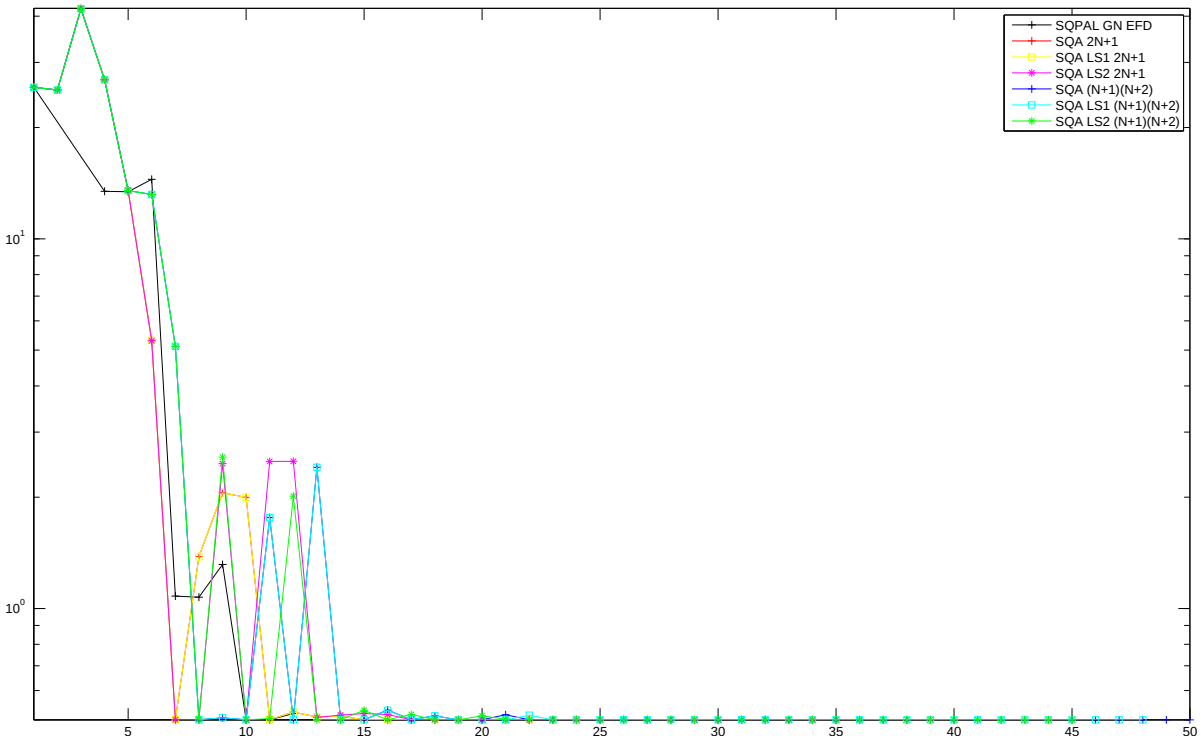


FIGURE 3.10 – Evolution de la valeur de la fonction objectif en fonction du nombre de simulations pour la minimisation de la fonction de moindres carrés (3.60) sans dérivées avec différentes méthodes d'optimisation

Sur la figure 3.10, nous pouvons remarquer que la méthode SQA est plus rapide que la méthode SQPAL Gauss Newton pour obtenir la solution sur cet exemple. Sur cette exemple, la prise en compte du caractère moindres carrés n'accélère pas la convergence de la méthode qui est déjà très rapide avec la version standard.

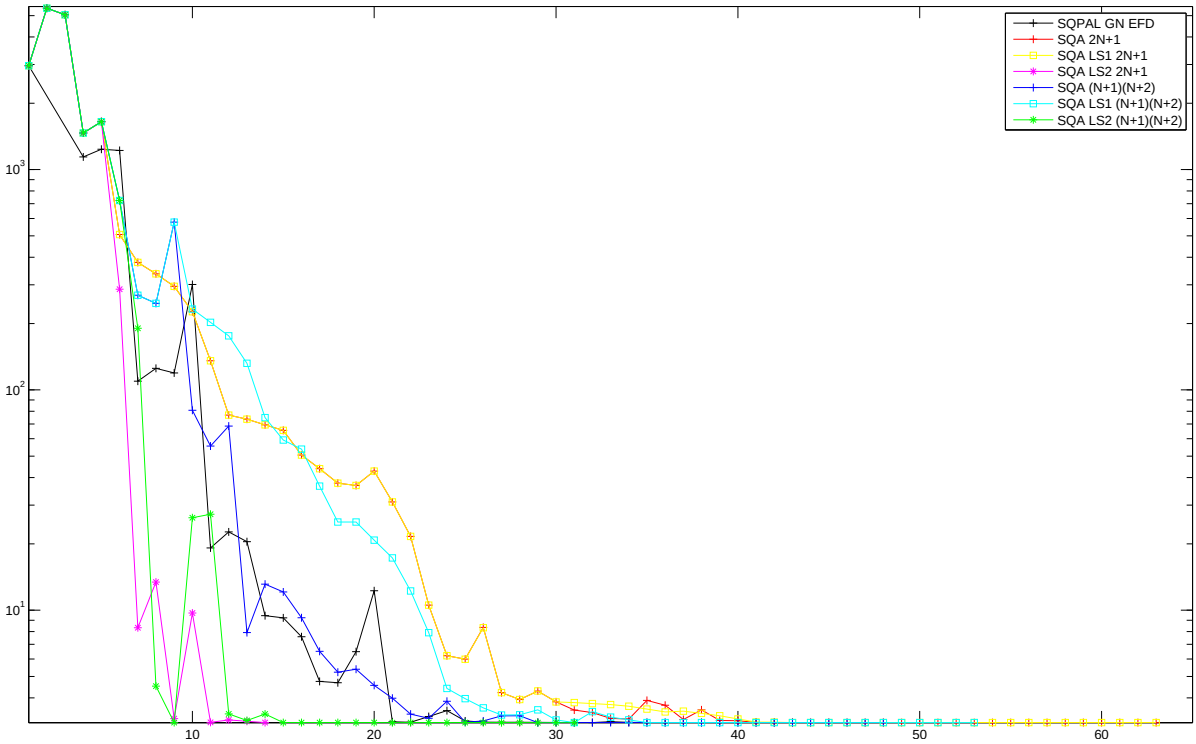


FIGURE 3.11 – Evolution de la valeur de la fonction objectif en fonction du nombre de simulations pour la minimisation de la fonction de moindres carrés (3.61) sans dérivées avec différentes méthodes d'optimisation

Sur la figure 3.11, nous remarquons que le minimum est obtenu plus rapidement pour les méthodes SQA-LS2 avec $2n+1$ ou $(n+1)(n+2)/2$ que pour les mêmes méthodes sans prendre en compte le caractère moindres carrés de la fonction objectif. De plus, ces mêmes méthodes sont plus rapides à obtenir le minimum que la méthode SQPAL Gauss Newton, méthode dédiée à la minimisation de moindres carrés. Les méthodes SQA-LS1 et SQA standard sont semblables lorsque le nombre de modèles construits n_m est égal au nombre des f_i : $n_m = n_{ls}$. Nous pouvons observer sur la figure 3.11 que les résultats obtenus par les méthodes SQA $2n+1$ et SQA-LS1 $2n+1$ d'une part et SQA $(n+1)(n+2)/2$ et SQA-LS1 $(n+1)(n+2)/2$ d'autre part sont relativement semblables deux à deux. Ceci n'est pas étonnant puisque les modèles ne diffèreront que par l'introduction de la minimisation de la norme de Frobenius.

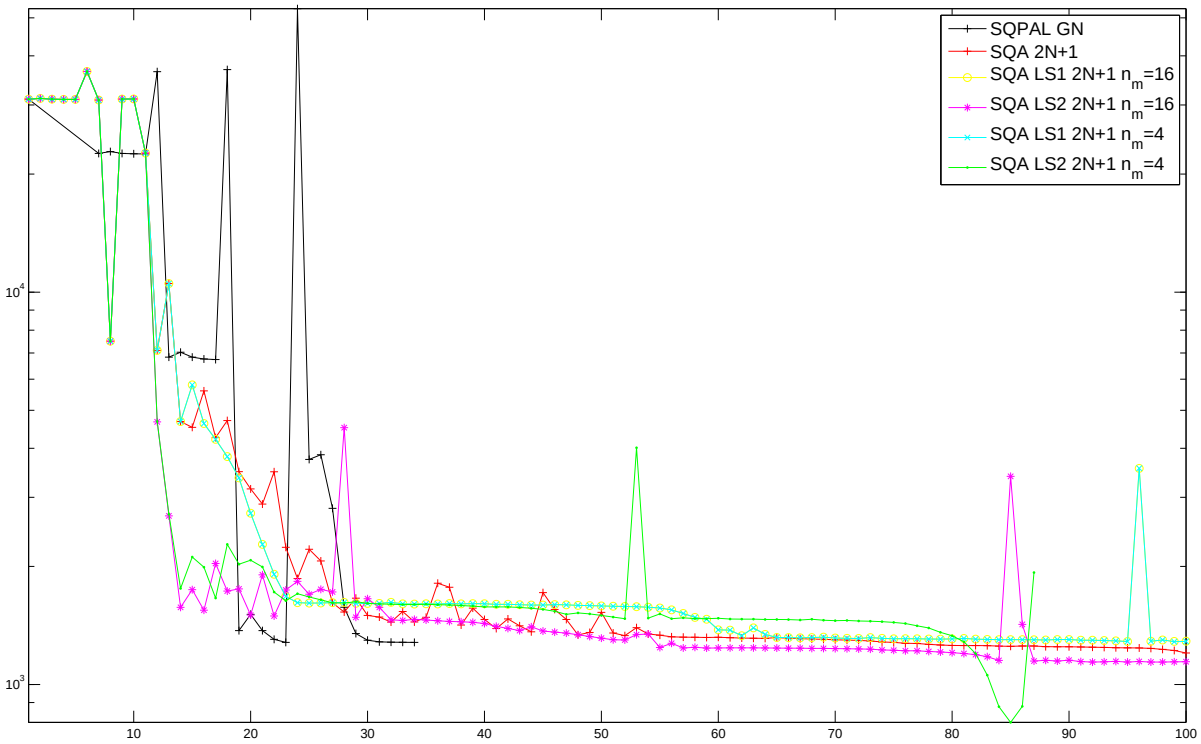


FIGURE 3.12 – Evolution de la valeur de la fonction objectif en fonction du nombre de simulations pour la minimisation de la fonction de moindres carrés (3.62) sans dérivées avec différentes méthodes d'optimisation

Sur la figure 3.12, nous pouvons observer que les méthodes SQA-LS2 $2n+1$, $n_m = 16$ ou $n_m = 4$ rattrapent la méthode SQPAL Gauss Newton qui nécessite beaucoup moins d'évaluations lors de son initialisation. Ces méthodes n'obtiennent la précision de SQPAL Gauss Newton que pour un nombre plus important de simulations. La prise en compte du caractère moindres carrés de la fonction est très intéressante surtout pour lorsque le nombre d'évaluation de la fonction objectif est relativement faible. En effet, les méthodes SQA-LS2 $2n+1$, $n_m = 16$ ou $n_m = 4$ obtiennent une valeur de la fonction objectif de $2.10{-}3$ au bout de 15 évaluations alors que la méthode SQA $2n+1$ obtient cette valeur après 25 évaluations.

Les résultats obtenus sur ces cas tests en dimension peu élevée pour tester la prise en compte du caractère de moindres carrés de la fonction objectif par notre méthode sont prometteurs. Des résultats supplémentaires de la prise en compte de ce caractère de la fonction objectif sur l'application en caractérisation de réservoir seront présentés dans la section 2.1.1 du chapitre 4.

4.4 Mise à jour des poids de pénalisation dans la fonction de mérite

Le choix des poids de pénalisation des contraintes sans dérivées est un problème complexe pour une prise en compte de ces contraintes de manière efficace par notre méthode. L'utilisation d'un poids de pénalisation des contraintes sans dérivées constant pourrait s'avérer trop simpliste pour certaines applications, même si nous n'avons pas rencontré de problème dans nos applications ; la fonction et les contraintes ayant été normalisées convenablement.

Nous avons pensé à une adaptation évolutive des différents poids de pénalisation de ces contraintes sans dérivées λ_i. Notre première tentative est d'emprunter les multiplicateurs de Lagrange μ_i obtenus lors de la minimisation du modèle quadratique Q de la fonction sous contraintes (voir l'équation (3.19)) à l'aide de la méthode SQP. Le problème survient lors de l'initialisation de la méthode, si la première itération ne vérifie pas une contrainte difficile, et si les modèles des contraintes ne sont pas compatibles dans la région de confiance. Dans ce cas, il est impossible de trouver une solution au sous problème (3.19). Par conséquent, celui-ci est remplacé par le sous problème (3.21), qui traite les contraintes difficiles dans une fonction de mérite et donc ne permet pas d'initialiser ou de mettre à jour les multiplicateurs de Lagrange. Un second problème apparait lorsque la méthode effectue une itération d'amélioration du modèle, le sous problème (2.2.3) ne prenant pas en compte les modèles des contraintes sans dérivées, les multiplicateurs de Lagrange associés à ces contraintes ne peuvent être mis à jour dans la fonction de mérite permettant de sélectionner le point optimal courant. Ce cas de figure est critique lorsqu'un minimum local se situe le long d'une contrainte difficile. En effet, le nouvel itéré obtenu par l'amélioration de modèle peut être aprouvé comme le point optimal courant par la fonction de mérite (à l'aide des multiplicateurs de Lagrange de l'itération précédente) alors qu'il ne vérifie pas les contraintes sans dérivées. Dans ce cas, la méthode peut boucler le long de cette contrainte modélisée mais peut également ne pas réussir à revenir vers le domaine admissible pour les contraintes sans dérivées si la région de confiance est réduite.

Une seconde possibilité pour la mise à jour des poids de pénalisation des contraintes sans dérivées est utilisée par Powell [133] dans le cadre de l'optimisation à l'aide de modèle linéaire de la fonction objectif et des contraintes difficiles. La fonction de mérite (prenant contrairement à notre méthode en compte les contraintes en normes L_∞) utilisée est la suivante :

$$\Phi(x) = f(x) + \mu \left(\max\{-C_{DF_{(i)}}(x)/i = 1, ..., n_{DF}\} \right)_{\#} . \tag{3.63}$$

Powell indique qu'une réduction de la fonction de mérite $\Phi(x_{new}) < \Phi(x_{old})$ ait lieu lorsque le paramètre de pénalisation μ ne permet pas de réduire le modèle de la fonction de mérite lors d'une itération de minimisation du modèle dans la région de confiance n'est pas envisageable. Lorsque cela arrive, on recherche le paramètre $\bar{\mu}$ permettant $\tilde{\Phi}(x_{new}) < \tilde{\Phi}(x_{old})$. Le paramètre μ courant reste inchangé si $\mu \geq 3\bar{\mu}/2$, autrement μ est remplacé par $2\bar{\mu}$. Cette augmentation du paramètre μ peut modifier le point optimal courant qui

sera remplacé par un point évalué qui réduit le terme

$$\left(\max\{-C_{DF_{(i)}}(x)/i = 1, ..., n_{DF}\}\right)_{\#}.$$

Cette solution est intéressante, mais la méthode utilisée par Powell [133] ne contenant pas d'étape d'amélioration du modèle, elle ne peut être directement être transposée à notre méthode. Notre choix de poids de pénalisation dans SQA est à perfectionner, mais celui-ci à tout de même permis de traiter les applications du chapitre 4 nécessitant la prise en compte de contraintes sans dérivées.

Chapitre 4

Résultats

Comparer des algorithmes d'optimisation sans dérivées se révèle très délicat. En effet, les critères d'arrêt des algorithmes ne permettent pas d'obtenir des solutions finales comparables. De plus, le nombre d'évaluations de la fonction permettant d'obtenir une réduction importante de la fonction objectif rend certains algorithmes inutilisables en grande dimension. Différents algorithmes d'optimisation sans dérivées sans contraintes sont comparés dans les publications suivantes [4, 5, 121].

Les résultats obtenus par notre méthode SQA (décrite dans le chapitre 3) sur différents benchmarks (CUTEr [77, 121], MOPTA08 [98]) seront tout d'abord présentés puis comparés avec d'autres méthodes. Les résultats obtenus sur des applications réelles de l'IFPEN seront présentés par la suite.

1 Benchmarks

En premier lieu, les résultats obtenus par notre méthode sur un ensemble de problèmes avec des contraintes de bornes issus du benchmark CUTEr [121] sont présentés et comparés avec d'autres méthodes à l'aide de profils de performances et de profils de données (voir annexe A) pour différentes précisions sur la fonction objectif (précision notée $\tau > 0$). Ces problèmes peuvent être lisses, bruités ou non différentiables.

Ensuite, des résultats obtenus sur des fonctions du même benchmark CUTEr [77] avec des contraintes non linéaires seront présentés.

Enfin, on considère un benchmark réputé difficile proposé par D. Jones de General Motors, construit à partir de données de simulations de crash de voitures (MOPTA08 [98]). Les résultats obtenus sur ces benchmarks par de nombreuses méthodes dont la méthode SQA sont décrits et comparés.

1.1 Benchmark avec contraintes de bornes (Moré et Wild)

Nous présentons dans ce paragraphe des résultats obtenus pour un budget limité d'évaluations de la fonction objectif. Nous nous placerons dans le cadre d'un problème d'optimisation sous contraintes de bornes, où la fonction objectif f peut être lisse, bruitée ou non dérivable. Une partie de ces résultats sont présentés dans [113].

Les problèmes considérés sont issus du benchmark CUTEr pour former trois ensembles de 53 problèmes. Les classes de problèmes étudiées sont des problèmes lisses, des problèmes bruités ainsi que des problèmes non différentiables de dimension $n \in [2, 13]$. Nous avons utilisé les problèmes présentés par Moré et Wild dans `www.mcs.anl.gov/~more/dfo` et [121] afin de comparer trois solveurs. Ces problèmes sous contraintes de bornes ont été appliqués pour comparer les différents solveurs (les paramètres ont été normalisés). Les méthodes d'optimisation ainsi comparées sont la méthode d'optimisation SQA présentée au chapitre 3, une implémentation de la méthode EGO de krigeage avec ajout de points par espérance de gain présentée en annexe D (voir [13, 12]), la méthode du simplexe de Nelder-Mead (NMSMAX voir section 1.1 du chapitre 1 et [124]), la stratégie d'évolution CMAES [87, 86, 88, 36] ainsi que la méthode SQPAL [44, 159]. Cette dernière est une méthode SQP avec le gradient estimé par différences finies (avec un pas de différences finies de 10^{-6}) et une approximation BFGS de la matrice Hessienne (voir annexe B). Le tableau 4.1 récapitule différents solveurs d'optimisation sans dérivées classés suivant les différents types de méthodes présentées dans le chapitre 1.

Les problèmes sont construits à partir de plusieurs fonctions de CUTEr, notées f_k. Les problèmes lisses sont définis par les fonctions deux fois continûment différentiables suivantes :

$$f(x) = \sum_{k=1}^{m} f_k(x)^2. \tag{4.1}$$

Les fonctions bruitées sont construites avec

$$f(x) = (1 + \epsilon_f \phi(x)) \sum_{k=1}^{m} f_k(x)^2, \tag{4.2}$$

où $\epsilon_f = 10^{-2}$ est le bruit relatif et la fonction de bruit $\phi : \mathbb{R}^n \mapsto [-1, 1]$ est définie par le polynôme de Chebyshev cubique $\phi(x) = T_3(\phi_0(x))$, $T_3(\alpha) = \alpha(4\alpha^2 - 3)$, où

$$\phi_0(x) = 0.9 \sin(100\|x\|_1) \cos(100\|x\|_\infty) + 0.1 \cos(\|x\|_2).$$

Nous nous intéressons également aux comportements des solveurs sur des fonctions non différentiables qui sont construites de la manière suivante :

$$f(x) = \sum_{k=1}^{m} |f_k(x)|. \tag{4.3}$$

Ces problèmes sont continus mais le gradient n'existe pas lorsque $f_k(x) = 0$ et $\nabla f_k(x) \neq 0$ pour tout indice k. La fonction $f(x)$ est deux fois continûment différentiable dans les régions où f_k ne change pas de signe.

Table 4.1 – Tableau récapitulatif des différents solveurs

Classes	Noms	Commentaires
Recherche directe	NMSMAX	Méthode de Nelder-Mead
	APPSPACK	Pattern Search parallèle
	NOMAD	Pattern Search
Métaheuristiques	CMAES	Stratégie d'évolution
Optim. à base de modèle de ∇f	SQPAL	SQP avec gradients calculés par différences finies et approximation BFGS de la Hessienne
	Implicit Filtering	
Optim. à base de modèle de f	EGO	krigeage avec ajout de point par espérance de gain sans contraintes
	COBYLA	modèle linéaire local valable dans une région de confiance avec contraintes "derivative free"
	DFO	modèle quadratique local valable dans une région de confiance avec contraintes "derivative based" et "derivative free"
	UOBYQA	modèle quadratique complet local valable dans une région de confiance sans contraintes
	CONDOR	modèle quadratique complet local valable dans une région de confiance avec contraintes "derivative based"
	WEDGE	modèle linéaire ou quadratique complet local valable dans une région de confiance sans contrainte
	NEWUOA	modèle quadratique local valable dans une région de confiance sans contraintes
	BOBYQA	modèle quadratique local valable dans une région de confiance avec contraintes de bornes
	BOOSTERS	modèle RBF local valable dans une région de confiance sans contraintes
	ORBIT	modèle RBF local valable dans une région de confiance sans contraintes
	SQA	modèle quadratique local valable dans une région de confiance avec contraintes "derivative based" et "derivative free"

Nous n'utilisons pas la totalité des problèmes définis par Moré et Wild lorsque les méthodes sont comparées avec la méthode EGO, car l'implémentation utilisée ne converge pas dans un temps de calcul raisonnable sur les problèmes dont la dimension est supérieure à 7 (29 problèmes de dimension $n \in [2, 7]$). Mais les comparaisons obtenues à partir des

résultats des autres méthodes sur la totalité des problèmes sont similaires à ceux que nous présentons par la suite. En revanche, lorsque les méthodes sont comparées entre elles sans comparaison à EGO, celles-ci le sont sur la totalité des exemples.

1.1.1 Problèmes lisses

Le test de convergence utilisé s'écrit : $f(x^*) \leq f_L + \tau(f(x_0) - f_L)$, où $\tau > 0$ est la tolérance, x_0 est le point de départ du problème, et f_L est calculée pour chaque problème comme la plus petite valeur de f obtenue par tous les solveurs en un nombre d'évaluations fixé. La tolérance $\tau \in [0, 1]$ représente un pourcentage de réduction de la valeur $f(x_0)$ de départ après un nombre d'évaluations donné (cf. annexe A).

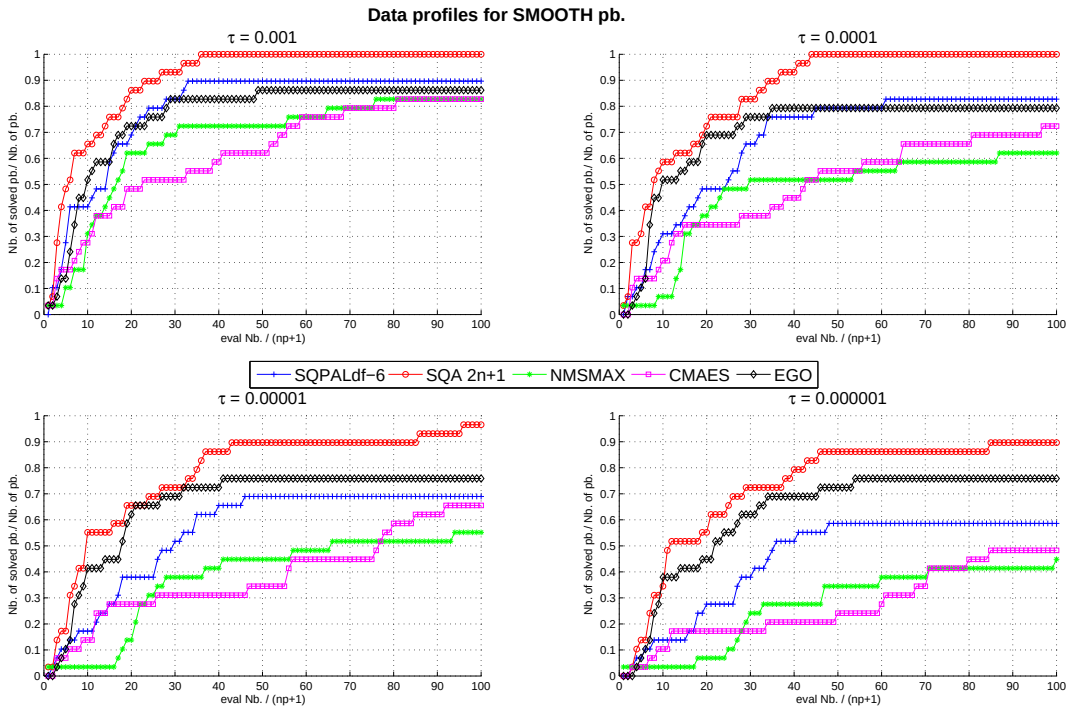


FIGURE 4.1 – Profils de données (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ

Les profils de données de la figure 4.1 montrent que la méthode SQA (voir chapitre 3) résout le plus grand pourcentage de problèmes pour chacune des précisions et pour tous les budgets d'évaluations de f.

Les différences de performance entre la méthode SQA et la méthode SQP avec différences finies pour approcher le gradient sont déjà assez conséquentes pour un budget de $20(n_p+1)$ évaluations (voir figure 4.1) : en effet, pour $\tau = 10^{-5}$ ou 10^{-6}, SQA résout 30 % de problèmes de plus que SQPAL avec un pas de 10^{-6} soit deux fois plus de problèmes.

Seule la méthode de krigeage avec ajout de points par espérance de gain semble pouvoir concurrencer la méthode SQA sur les problèmes de taille inférieure à 7.

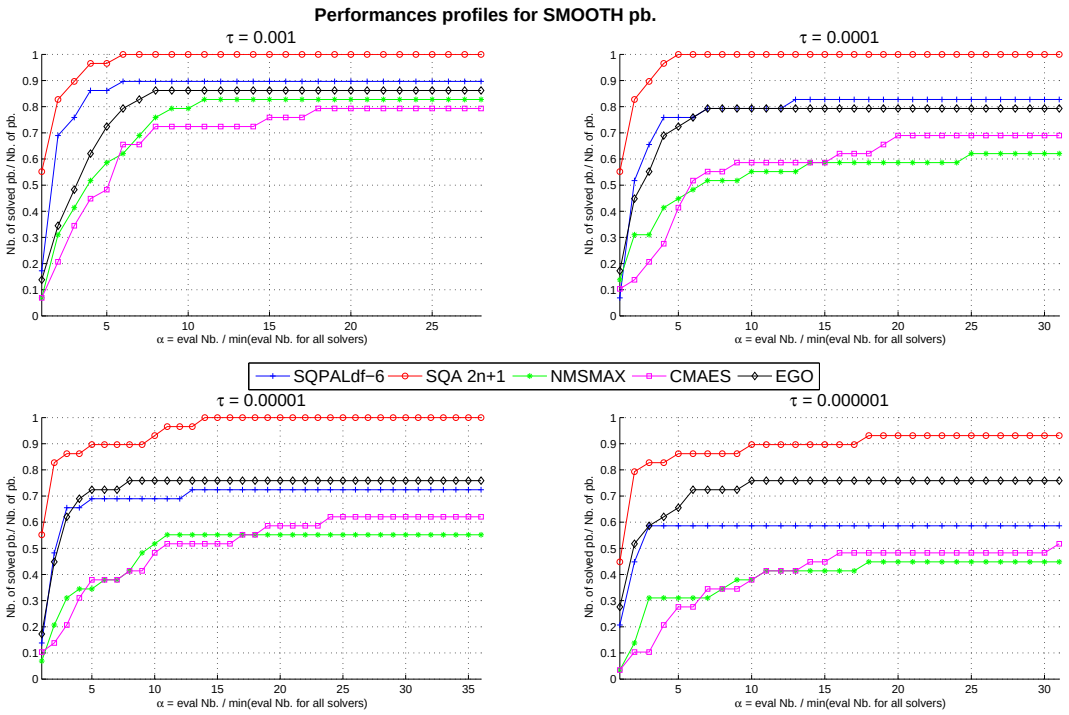


FIGURE 4.2 – Profils de performances (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ

Les profils de performances de la figure 4.2 montrent pour toutes les précisions τ que la méthode SQA est le solveur le plus rapide pour au moins 50 % des problèmes alors que le krigeage et la méthode SQP avec différences finies sont plus rapides seulement pour moins de 30 % des problèmes. Et la stratégie d'évolution CMAES ainsi que le simplexe de Nelder-Mead sont les plus rapides seulement pour moins de 10 % des problèmes. De plus, la méthode SQA est la plus robuste : elle résout le plus grand pourcentage de problèmes pour chacune des précisions. Les différences de performance entre les solveurs considérés sont maintenues lorsque le nombre d'évaluations augmente, le krigeage avec espérance de gain et la méthode SQP avec différences finies ne résolvent pas autant de problèmes que la méthode SQA.

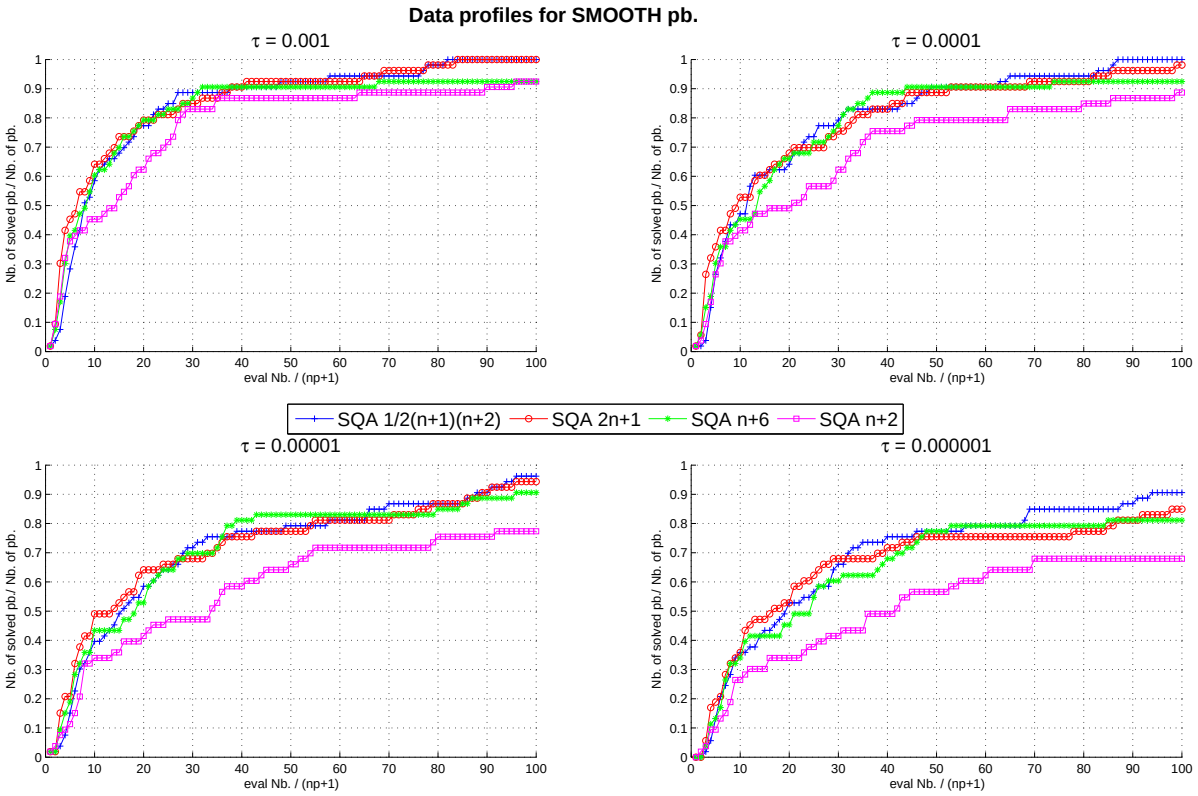


FIGURE 4.3 – Profils de données (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les figures 4.3 et 4.4 indiquent l'influence du choix du nombre de points d'interpolation du modèle quadratique sur les performances de la méthode SQA. Les profils de données de la figure 4.3 montrent que la méthode SQA avec un nombre de points d'interpolation fixe de $(n+1)(n+2)/2$, $2n+1$ et $n+6$ sont toujours plus efficaces que SQA avec $n+2$ points. Nous remarquons également que SQA interpolant la fonction objectif avec un modèle quadratique en $(n+1)(n+2)/2$ points est pour toutes précisions τ le solveur permettant de résoudre le plus grand nombre de problèmes pour un budget de $100\times(np+1)$ simulations. En revanche, SQA avec $2n+1$ et $n+6$ points permettent de résoudre plus de problèmes pour des budgets de simulations bien inférieurs. On pouvait attendre ces résultats, SQA $n+2$ dispose d'un modèle très peu précis, alors que SQA $(n+1)(n+2)/2$ profite d'un modèle très précis mais coûteux lors de la phase d'initialisation.

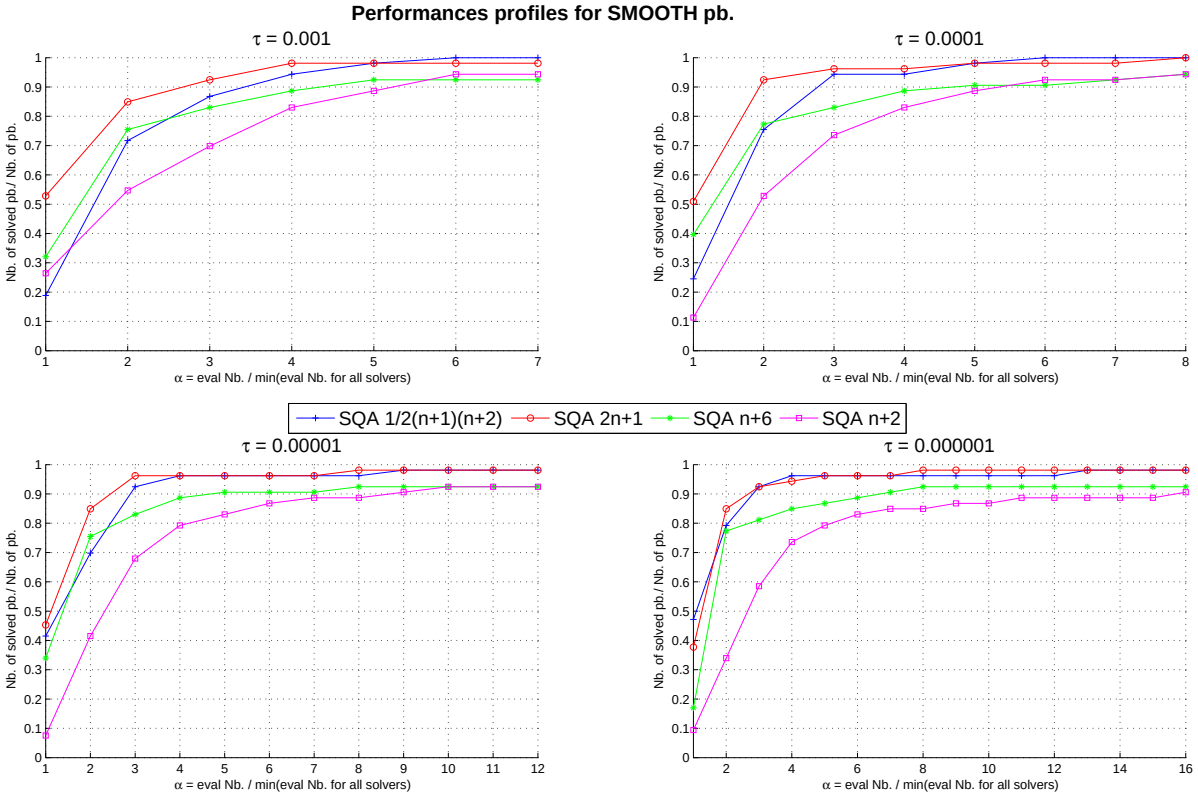


FIGURE 4.4 – Profils de performances (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performances de la figure 4.4 illustrent également l'importance du choix du nombre de points d'interpolation du modèle quadratique de la méthode SQA. En effet si une faible précision sur la convergence est suffisante, $\tau = 10^{-3}$ à 10^{-5}, un nombre de points d'interpolation de $2n+1$ ou $n+6$ est très satisfaisant ; en effet, ces méthodes sont les plus rapides à résoudre respectivement 50 % et 40 % des problèmes pour une précision de $\tau = 10^{-4}$. En revanche pour une meilleure précision de convergence $\tau = 10^{-6}$, la méthode SQA avec un nombre de points d'interpolation de $(n+1)(n+2)/2$ est plus rapide à résoudre pratiquement la moitié des problèmes. De plus la méthode SQA $(n+1)(n+2)/2$ est la plus robuste : elle résout le plus grand pourcentage de problèmes pour chacune des précisions. Notons que la dimension des problèmes de ce benchmark est inférieure à 13, celle-ci est relativement peu élevée ce qui permet à la méthode SQA $(n+1)(n+2)/2$ d'être plutôt plus efficace que la méthode SQA construisant un modèle quadratique à partir de moins de points d'interpolation.

A partir de ces résultats, pour permettre d'obtenir l'avantage d'une convergence rapide dès les $n+2$ premières simulations et une précision plus importante sur la convergence ainsi qu'une méthode plus robuste, il semble judicieux d'adapter le nombre de points d'interpolation au cours des itérations pour associer les avantages de la méthode SQA construisant

un polynôme quadratique sous déterminé et la méthode SQA construisant un polynôme quadratique complet. Cette proposition constitue une perspective d'amélioration de la méthode SQA non implémentée au cours de la thèse.

1.1.2 Problèmes bruités

Nous présentons les résultats pour les problèmes bruités uniquement pour des précisions de τ assez faibles (10^{-2} et 10^{-3}). En effet, des précisions plus fines sont difficiles à atteindre dans le cas d'une minimisation de fonction bruitée (bruit relatif égal à 10^{-2}) et ne sont pas représentatives.

Les profils de données des figures 4.5 montrent que la méthode SQA, résout le plus grand pourcentage de problèmes pour tous les budgets d'évaluations de fonction et pour tous niveaux de précision τ. Les résultats sont sensiblement similaires au cas non bruité mais la méthode SQA et le krigeage avec ajout de points par espérance de gain semblent moins sensibles au bruit que la méthode utilisant les différences finies pour approcher le gradient. L'approximation du gradient est effectivement une difficulté dans le cas de problème bruité, surtout pour de petit pas de différences finies. Cependant, nous signalons que les résultats sur l'optimisation de ces mêmes problèmes bruités avec SQPAL avec un pas de différences finies de 10^{-3}, présentés dans [113]) étaient moins performants. Le choix du pas de différences finies est le réel souci de l'utilisation de ces méthodes sur des fonctions bruités. La stratégie d'évolution et la méthode du simplexe de Nelder-Mead sont sensiblement moins affectées par le bruit que les autres solveurs, effectivement les résultats paraissent meilleurs que sur le cas lisse.

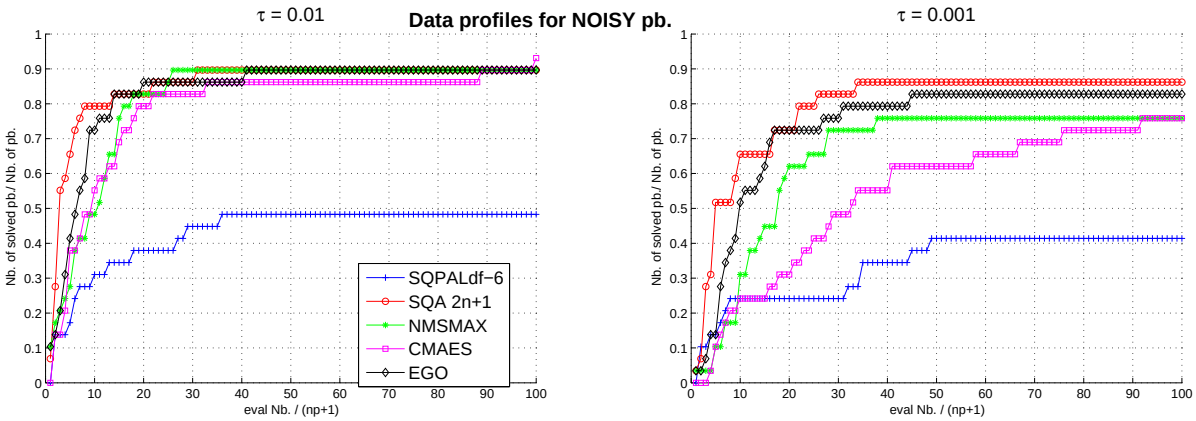


FIGURE 4.5 – Profils de données (voir annexe A) pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ

De plus, les profils de performances des figures 4.2 et 4.6 montrent que la méthode utilisant les différences finies résout deux fois moins de problèmes bruités que de problèmes

lisses pour des niveaux de précision τ assez faibles (10^{-3}). Les deux méthodes SQA et EGO n'ont que 10 % de problèmes non résolus supplémentaires pour le même niveau de précision.

Les profils de performances de la figure 4.6 montrent que la méthode SQA est le solveur le plus rapide pour au moins 55 % des problèmes alors que les 4 autres méthodes sont plus rapides pour moins de 20 % des problèmes. De plus, la méthode SQA est, comme pour les problèmes lisses, la plus robuste car elle résout le plus grand pourcentage de problèmes pour chacune des précisions.

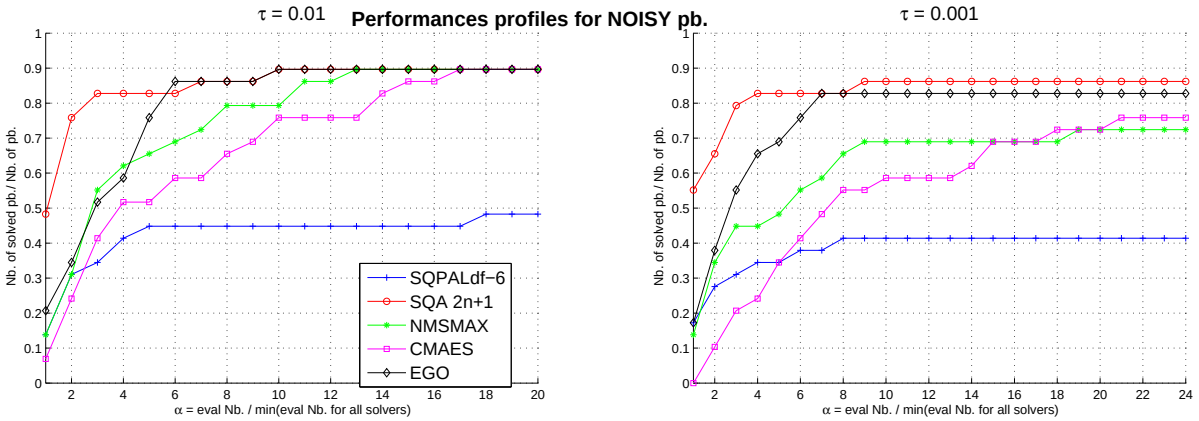


FIGURE 4.6 – Profils de performances (voir annexe A) pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ

Une technique pour prendre en compte le bruit dans la méthode SQA a été implémentée (voir section 4.2 du chapitre 3). Les résultats obtenus étaient sensiblement identiques aux résultats obtenus sans, c'est pourquoi ils ne sont pas présentés parmi ces résultats. Comme expliqué dans la section 4.2 du chapitre 3, la présence de la distance minimale entre les points d'interpolation ρ permet de limiter l'impact du bruit.

Les figures 4.7 et 4.8 indiquent l'influence du nombre de points d'interpolation sur les performances de la méthode SQA sur les problèmes bruités. Les profils de données de la figure 4.7 montrent que la méthode SQA avec un nombre de points de $(n+1)(n+2)/2$ permet de résoudre un plus grand pourcentage de problèmes bruités dès $20(n_p+1)$ évaluations pour une précision $\tau = 10^{-2}$ et 10^{-3} (pour une dimension maximale de problèmes égale à 13). Celle-ci est également la méthode vérifiant le plus grand pourcentage de problèmes pour toutes précisions τ, pour un budget de $100(n_p + 1)$ évaluations. Les méthodes SQA avec un nombre de points d'interpolation de $(n + 6)$ et $2n + 1$ permettent de résoudre un plus grand pourcentage de problèmes pour de petits budgets d'évaluation.

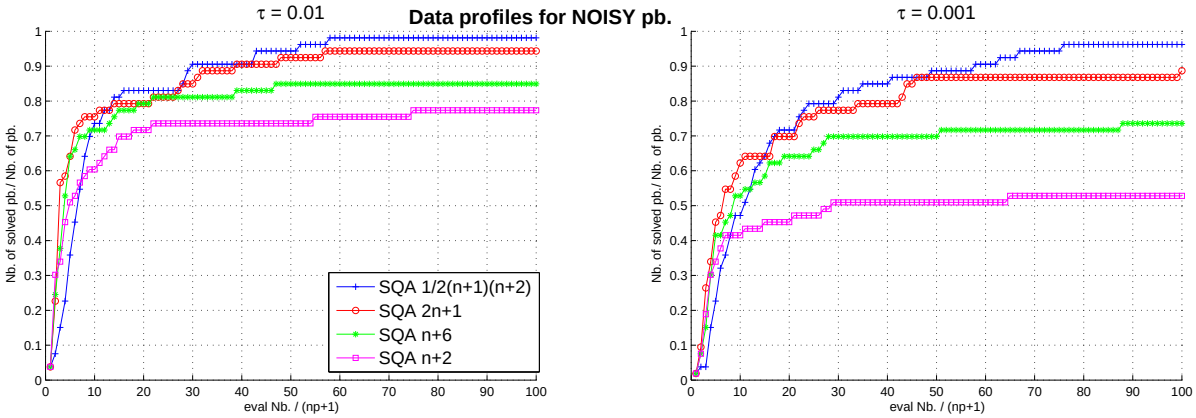


FIGURE 4.7 – Profils de données (voir annexe A) pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performance de la figure 4.8 indiquent que la méthode SQA avec $2n+1$ points d'interpolation est la méthode la plus rapide pour environ 50 % des problèmes pour toutes précisions τ. La méthode SQA $(n+1)(n+2)/2$ est la méthode la plus robuste, elle permet de résoudre environ 95 % des problèmes pour une précision $\tau = 10^{-3}$. Nous pouvons remarquer que plus le modèle de la fonction utilisé par la méthode SQA est basé sur un grand nombre de points d'interpolation, plus la méthode est robuste sur les problèmes bruités.

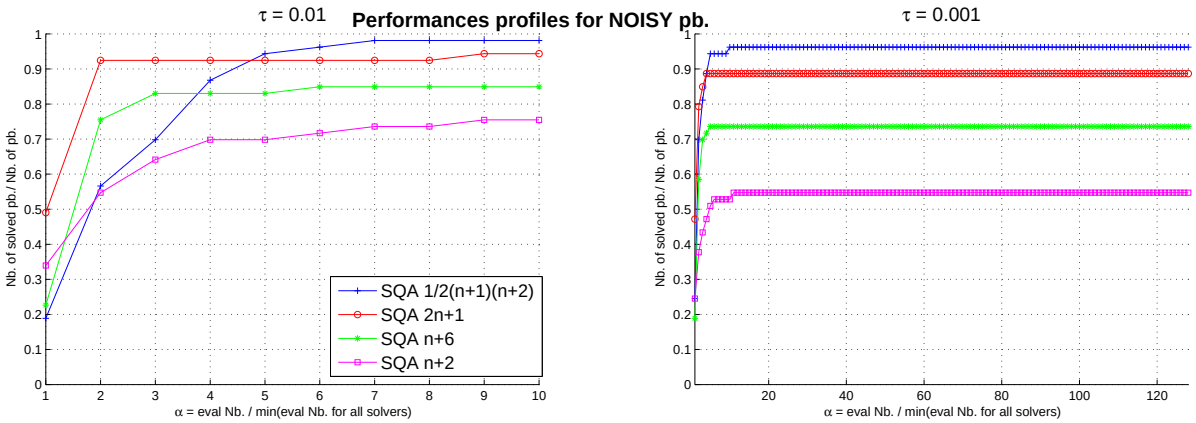


FIGURE 4.8 – Profils de performances (voir annexe A) pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

1.1.3 Problèmes non différentiables

Nous présentons les résultats pour les problèmes non différentiables, qui nous permettent de connaître les performances des solveurs en présence de non-différentiabilités locales, sachant que les preuves de convergence pour l'optimisation sans dérivées supposent la différentiabilité de f.

Les profils de données des figures 4.9 indiquent que les problèmes non différentiables sont plus difficiles à résoudre que les problèmes lisses et bruités. En effet, aucun optimiseur n'est capable de résoudre plus de 45 % de problèmes avec un budget de $100(n_p + 1)$ évaluations de fonction pour une tolérance $\tau = 10^{-6}$. Pour le même budget et le même niveau de précision, environ 90 % de problèmes étaient résolus pour les fonctions lisses. La méthode SQA est bien moins performante sur ces problèmes, elle est sensible à la non-différentiabilité de la fonction objectif. Cependant, elle reste plus efficace que les autres méthodes pour toutes précisions τ.

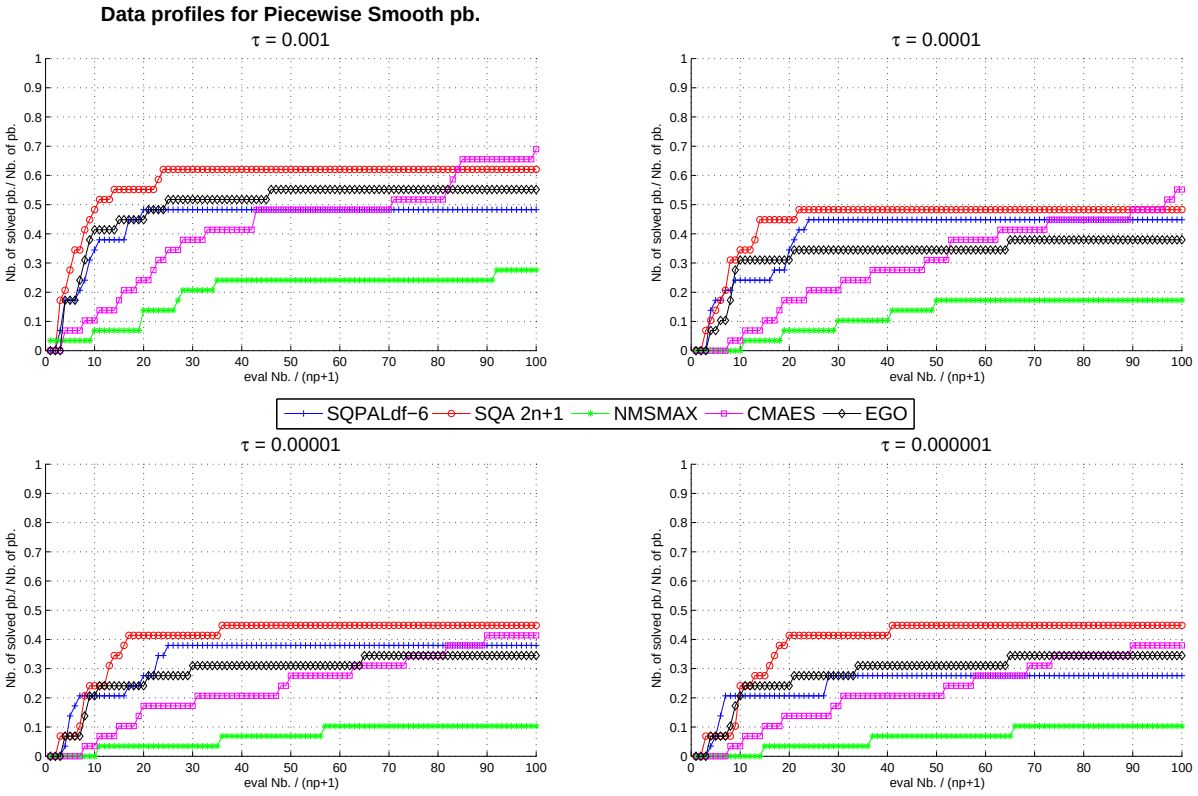


FIGURE 4.9 – Profils de données (voir annexe A) pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ

Les profils de performances des figures 4.10 indiquent que la méthode évolutionnaire CMAES est la méthode la plus robuste pour diminuer la fonction objectif sur les fonctions non différentiables, suivie de la méthode SQA. Les résultats pour $\tau = 10^{-7}$ indiquent que

SQA est le solveur le plus rapide sur 30 % des problèmes, alors que les méthodes SQPAL, EGO et CMAES ne sont les plus rapides que sur 20 % des problèmes.

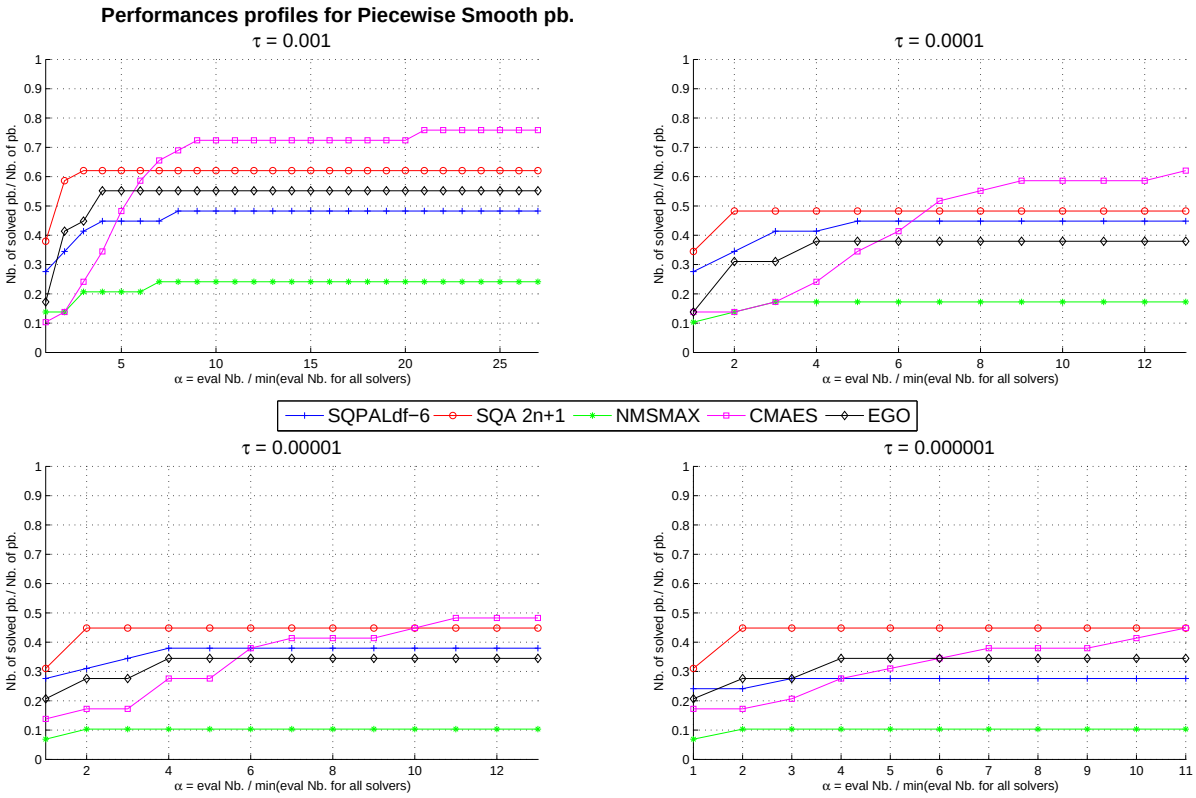


FIGURE 4.10 – Profils de performances (voir annexe A) pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 7 pour différentes précisions τ

Les figures 4.11 et 4.12 indiquent l'influence du choix du nombre de points d'interpolation sur les performances de la méthode SQA. Les profils de données de la figure 4.11 montrent que les performances de la méthode SQA avec un modèle construit à partir de différents nombres d'interpolation de points semblent logiquement s'inverser avec une augmentation du nombre d'évaluations de la fonction. En effet, plus le nombre de points d'interpolation du modèle quadratique est petit, plus l'optimisation démarre rapidement (phase d'initialisation moins coûteuse en évaluation), et donc plus le solveur est rapide pour de petits budgets d'évaluation. Et plus le nombre de points d'interpolation est grand, plus le solveur est capable de résoudre de problèmes. Les profils de données des différentes méthodes s'inversent vers $30(n_p + 1)$ évaluations pour des problèmes de dimension inférieure à 13 pour toutes précisions τ.

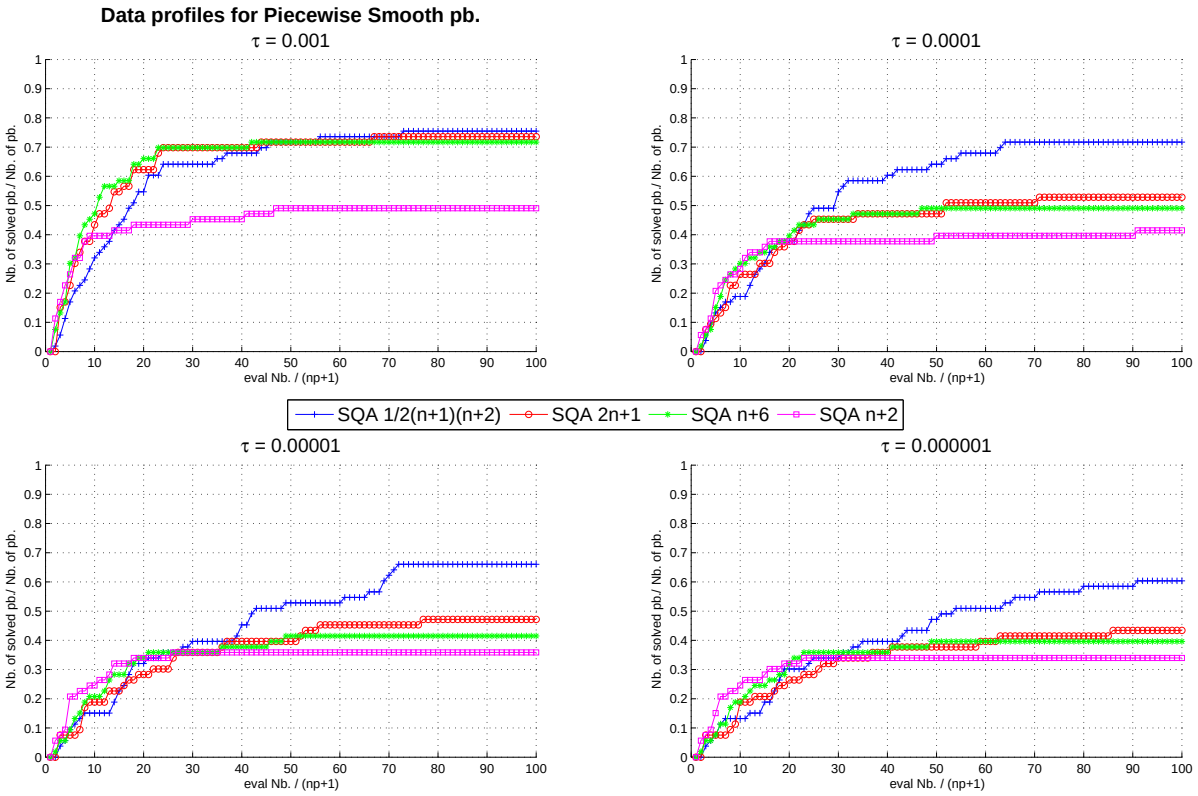


FIGURE 4.11 – Profils de données (voir annexe A) pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performance de la figure 4.12 n'indiquent pas réellement qu'un solveur est le plus rapide à résoudre un grand nombre de problèmes, contrairement aux résultats obtenus sur les fonctions lisses où la méthode SQA $(2n+1)$ ou $(n+1)(n+2)/2$ permettaient de résoudre plus rapidement 50 % des problèmes suivant la précision voulu. En revanche, la robustesse est comme pour les fonctions lisses liée aux nombres de points d'interpolation.

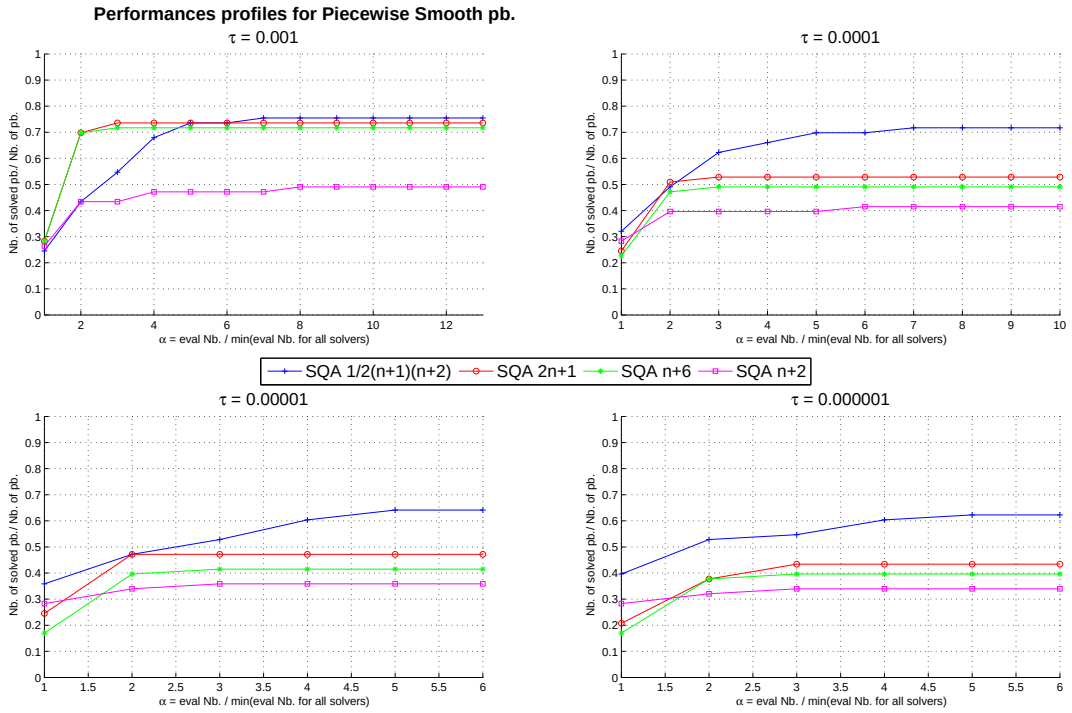


FIGURE 4.12 – Profils de performances (voir annexe A) pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

1.1.4 Conclusions

L'étude des différentes méthodes d'optimisation sans dérivées sur ce benchmark nous indique que le solveur SQA est une alternative très intéressante pour l'optimisation de fonctions objectifs lisses, bruitées et non différentiables sans dérivées. En effet, cette méthode obtient une décroissance de la fonction objectif plus rapide que les autres solveurs sur les différents types de fonctions objectifs testées. De plus, cette méthode est également plus robuste ques les autres méthodes sur les problèmes lisses et bruités. Seule la méthode évolutionnaire CMAES est plus robuste sur les problèmes non-différentiables, ce qui n'est pas étonnant compte tenu des hypothèses de régularité nécessaires pour prouver la convergence de SQA.

L'étude de sensibilité du nombre de points d'interpolation du modèle quadratique de la méthode SQA nous indique que le choix de ce paramètre permet de trancher entre une convergence rapide et la robustesse de la méthode. En effet, si le nombre de points d'interpolation du modèle quadratique est faible (typiquement $n+2$), l'optimisation démarre plus rapidement mais la méthode sera moins robuste que pour un modèle quadratique complet avec $(n+1)(n+2)/2$ points d'interpolation. Comme nous l'avons expliqué dans le paragraphe 1.1.1 de ce chapitre, il serait judicieux d'adapter le nombre de points d'in-

terpolation au cours des évaluations pour permettre d'associer ces différents avantages.

1.2 Benchmark CUTEr sous contraintes avec dérivées

Dans ce paragraphe, nous comparons les résultats obtenus pour un budget limité d'évaluations de la fonction objectif par différents solveurs. Les problèmes étudiés sont un ensemble de 13 problèmes de Hock et Schittkowski [1], problèmes d'optimisation sous contraintes linéaires et non linéaires, étudiés dans [167]. Tous les problèmes proviennent de CUTEr et les points de départ sont donnés. La dimension des problèmes est comprise entre 2 et 13. Nous présentons de nouveau les résultats obtenus sous forme de profils de données et de profils de performance pour différentes précisions. Les solveurs utilisés sont :

- SQA pour $n+2$, $2n+1$ et $(n+1)(n+2)/2$ points d'interpolation,
- SQPAL ([44, 159], méthode SQP avec les gradients calculés par différences finies (avec un pas de différences finies de 10^{-6}) et une approximation BFGS de la matrice Hessienne) et
- COBYLA [133] construit à l'aide d'un modèle d'interpolation linéaire basée sur $n+1$ points d'interpolation.

Le critère d'arrêt de la méthode SQA et de COBYLA est le même que celui utilisé par Vanden Berghen [167], on le fixe à 10^{-4}.

Les profils de données de la figure 4.13 montrent que la méthode SQA $(n+1)(n+2)/2$ résout le plus grand pourcentage de problèmes pour chacune des précisions et pour pratiquement tous les budgets d'évaluations de f. La méthode SQP approchant les gradients par différences finies n'est effectivement pas très économique en évaluations de fonction, la méthode COBYLA approchant la fonction objectif par un modèle linéaire donne des résultats plus proches que SQA mais l'utilisation d'un modèle plus précis (quadratique au lieu de linéaire) permet d'obtenir de meilleures performances.

1. Ce sont les problèmes Hock et Schittkowski : HS22, HS23, HS26, HS34, HS38, HS44, HS65, HS76, HS100, HS106, HS108 et HS116

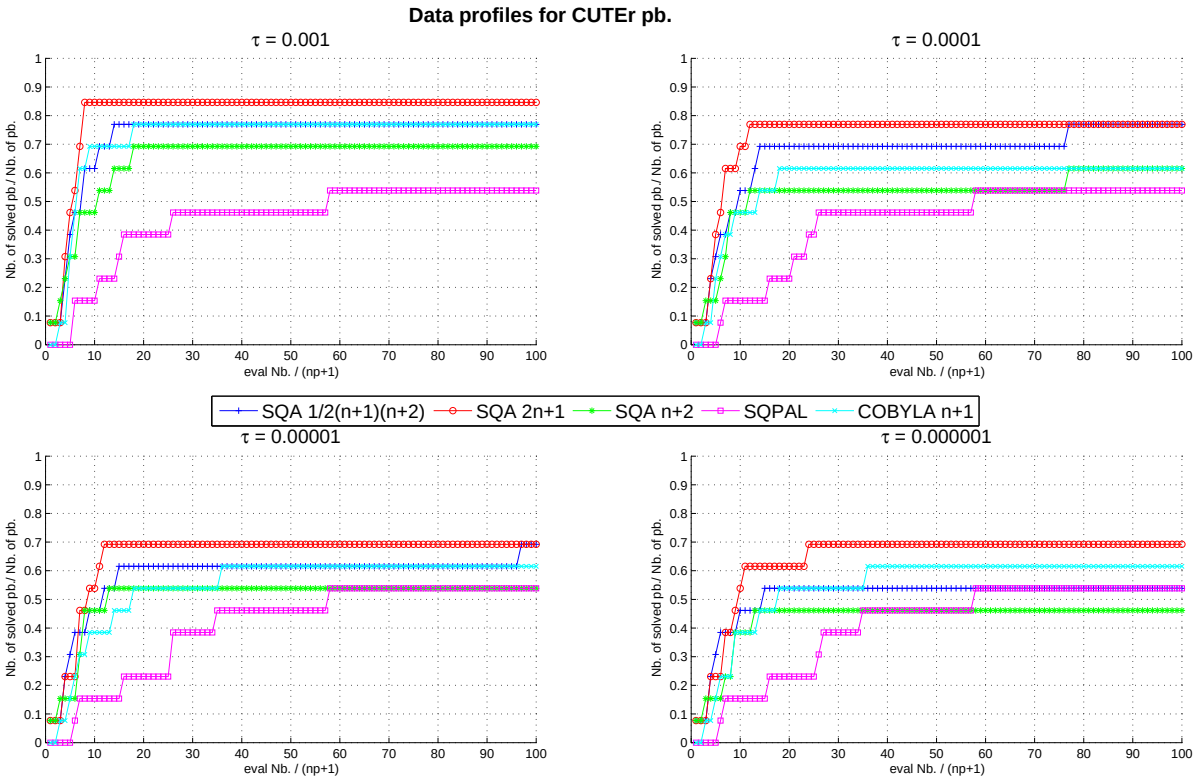


FIGURE 4.13 – Profils de données (voir annexe A) pour les problèmes du benchmark CUTEr sous contraintes de taille inférieure à 13 pour différentes précisions τ

Les profils de performances de la figure 4.14 montrent que la méthode SQA $(n+1)(n+2)/2$ est le solveur le plus rapide sur 30 à 40 % des problèmes alors que les méthodes SQPAL et COBYLA sont les plus rapides sur moins de 25 % des problèmes. De plus, la méthode SQA utilisant un modèle quadratique complet est également la plus robuste. Il s'agit en effet de la méthode résolvant le plus grand pourcentage de problèmes pour chacune des précisions. Notons que les problèmes non résolus par la méthode SQA, mais résolus par d'autres solveurs (par exemple HS108 et HS116), ne vérifient pas les contraintes au cours des itérations. Notre méthode SQA nécessite, en effet, la vérification des contraintes avec dérivées au cours des itérations. Pour résoudre ces problèmes, sans nécessiter la vérification des contraintes au cours des itérations, certaines contraintes peuvent être prise en compte comme des contraintes sans dérivées. Effectivement, la prise en compte de ces contraintes permet d'éviter de les vérifier lors de chaque itération. La méthode SQA $2n+1$ est légérement moins efficace que la méthode SQA $(n+1)(n+2)/2$. Les résultats obtenus par cette dernière sont tout de même meilleurs que ceux obtenus par les méthodes SQPAL et COBYLA.

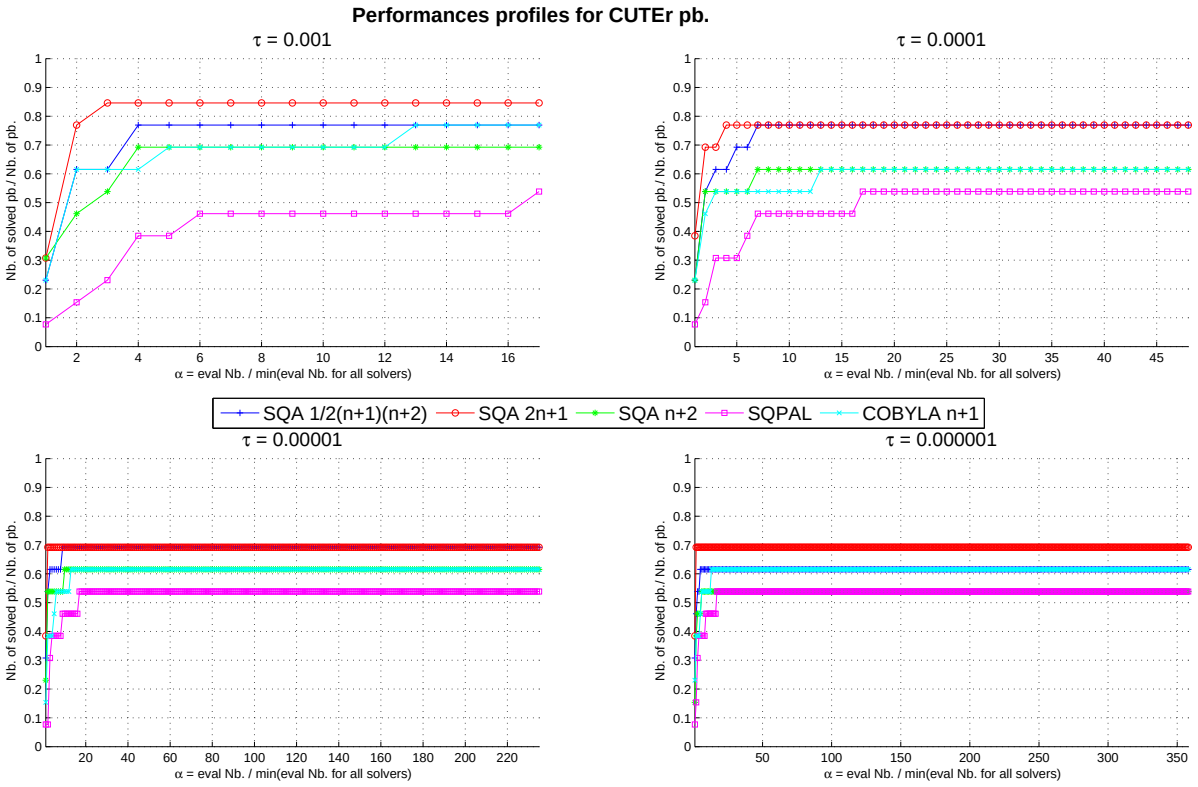


FIGURE 4.14 – Profils de performances (voir annexe A) pour les problèmes du benchmark CUTEr sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les résultats obtenus sur ce benchmark indiquent que la méthode SQA est efficace pour l'optimisation sans dérivées soumise à des contraintes. Cette méthode permet une convergence assez rapide et est plus robuste que les autres méthodes testées.

1.3 MOPTA08

Le benchmark MOPTA08 [98] est un problème de minimisation très complexe proposé par D. Jones, construit à partir d'un problème automobile réel de Général Motors. Le but est d'optimiser les performances d'un véhicule sous certaines contraintes de sécurité. La simulation (crash test) est très coûteuse en termes de temps de calcul (60 simulations au plus peuvent être réalisées par jour) d'où la nécessité d'utiliser des méthodes d'optimisation adaptées. La fonction objectif à minimiser dépend de 124 paramètres et 68 contraintes doivent être vérifiées. Les dérivées de ces fonctions ne sont pas disponibles.

Afin de partager cette application avec la communauté de l'optimisation, Jones a construit des surfaces de réponses à base de krigeage de la fonction objectif et des contraintes à partir de simulations du problème réel. L'objectif principal de ce benchmark est de réduire la valeur de la fonction objectif en dessous d'un seuil de 228 (la valeur initiale

de la fonction objectif est 251) et de vérifier toutes les contraintes en moins de 1800 évaluations c'est-à-dire 15 fois le nombre de paramètres, ce qui correspond environ à un mois de temps CPU pour le simulateur réel.

Jones a tenté de résoudre ce problème avec plusieurs méthodes sans obtenir le résultat espéré.

Les résultats obtenus par deux méthodes SQP classiques avec différences finies (algorithme Harwell VF13 et iSIGHT-FD) sont présentés dans les figures 4.15 et 4.16 (figures de [98]) : la première méthode réduit très lentement la fonction objectif tout en respectant les contraintes et la seconde méthode réduit rapidement la fonction objectif mais est très lente à vérifier les contraintes. Il est très difficile de régler le pas de différences finies pour le calcul des dérivées approchées, la fonction étant également fortement bruitée.

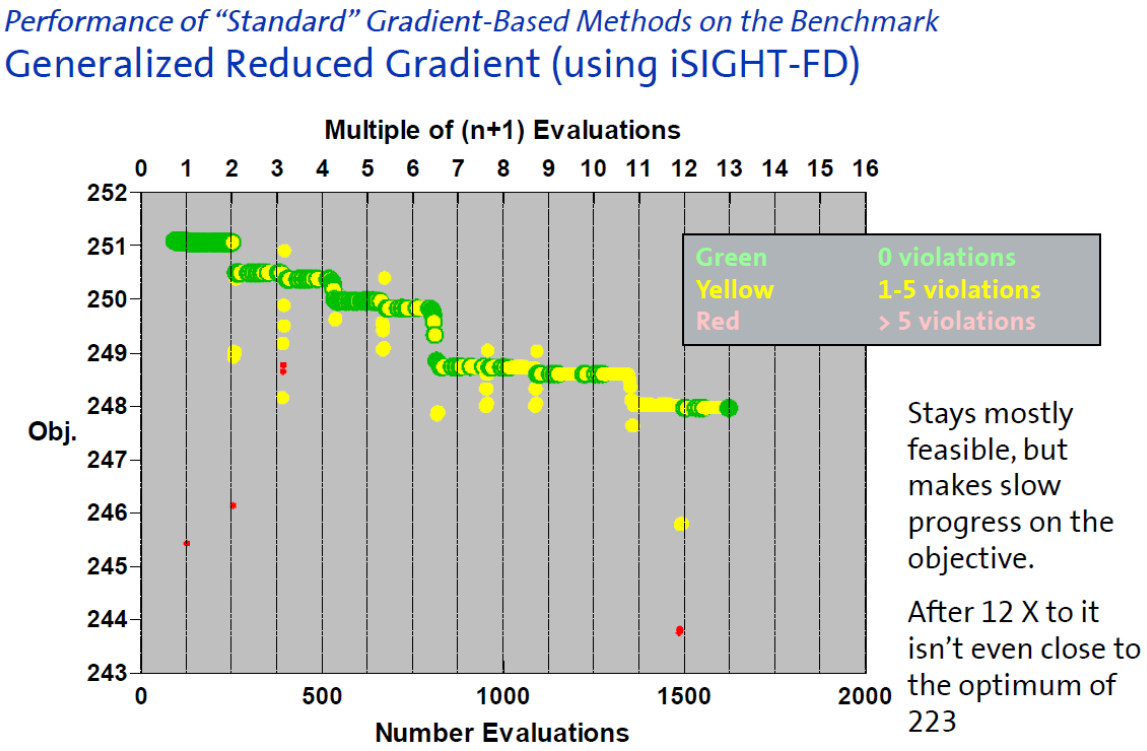


FIGURE 4.15 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation par une méthode SQP classique (iSIGHT-FD) avec une approximation des dérivées par différences finies

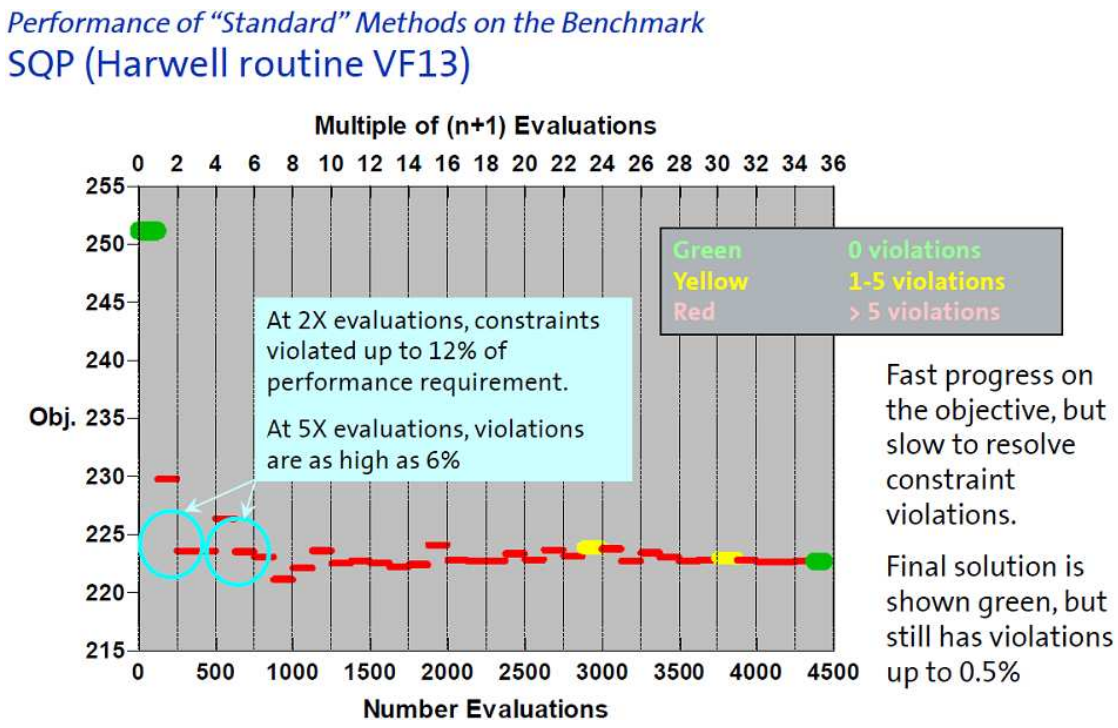


FIGURE 4.16 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation par une méthode SQP classique (Harwell VF13) avec une approximation des dérivées par différences finies

Une méthode de stratégie d'évolution (NUTECH ClearVu) a également été testée avec différentes implémentations pour la prise en compte des contraintes (pénalisation, méthode de filtre, nombre de violations des contraintes voir [98]), les résultats obtenus par pénalisation de la fonction objectif sont présentés dans la figure 4.17. En utilisant comme méthode de prise en compte de contraintes une pénalisation avec un poids de pénalisation élevé, ou le nombre de violations de contraintes, les résultats ne sont pas concluants, une très faible décroissance de la fonction objectif est obtenue pour rester dans le domaine admissible. En revanche, à l'aide de la méthode de filtre, une décroissance de la fonction objectif est obtenue mais les contraintes ne sont pas vérifiées. La prise en compte des contraintes de NUTECH ClearVu permet de réduire légèrement la fonction objectif en vérifiant les contraintes (voir la figure 4.17).

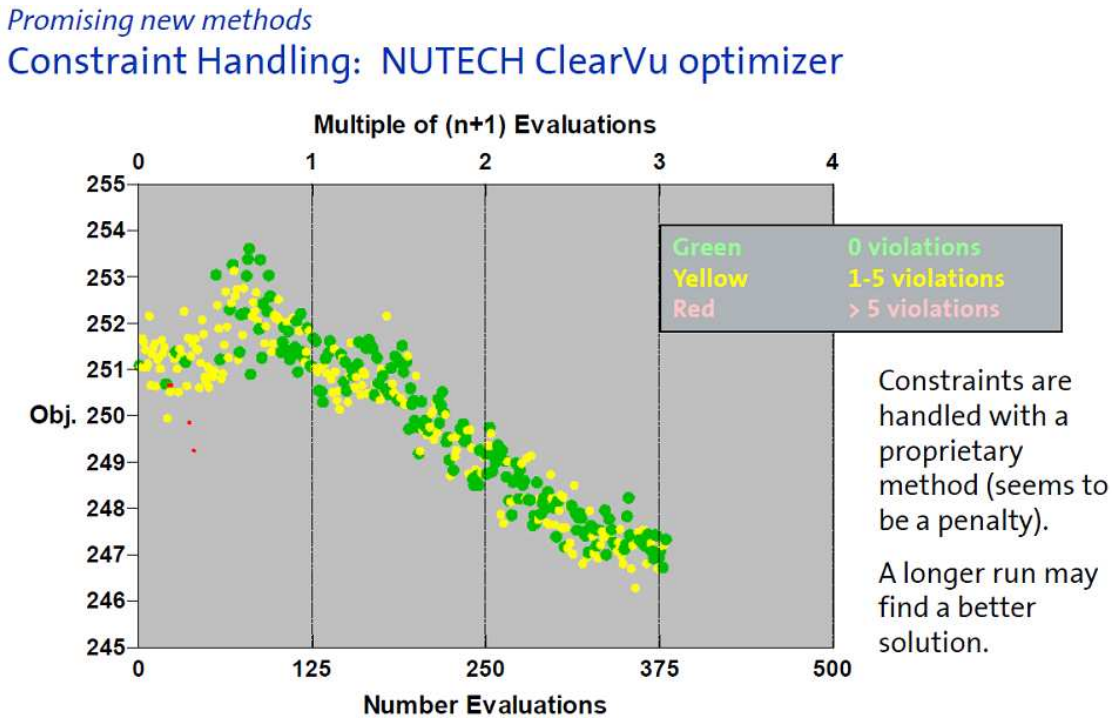


FIGURE 4.17 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation avec une stratégie d'évolution

Une méthode de recherche locale utilisant une approximation linéaire locale a également été testée : COBYLA [133]. Cette méthode a permis une réduction intéressante en moins de 1800 itérations, mais n'atteint pas la valeur cible de 228.

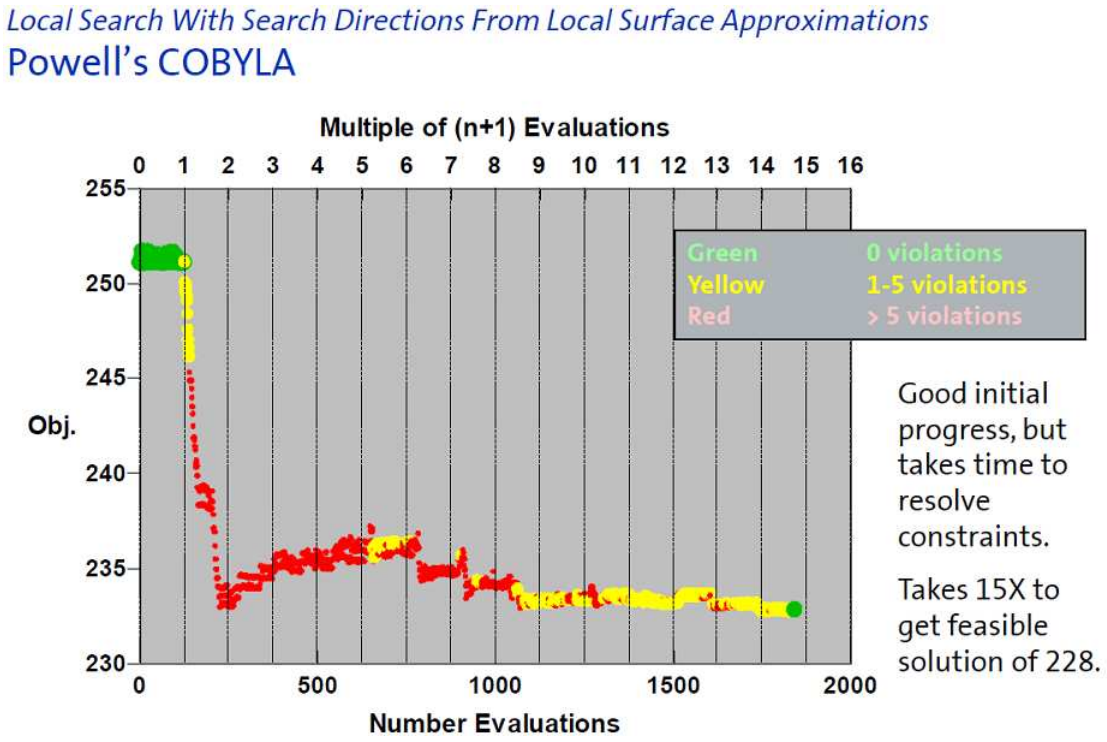


FIGURE 4.18 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation avec la méthode COBYLA d'approximation linéaire locale

Le résultat obtenu par la méthode SQA est présenté dans la figure 4.19. Les résultats de SQA sont très prometteurs : en effet, le seuil de 228 est atteint en un peu plus de 800 simulations et les contraintes sont presque toutes satisfaites à partir de ce nombre de simulations. La seule méthode qui donne des performances proches de celles obtenues par la méthode SQA est une méthode basée sur des approximations RBF (radial basis functions [149, 145]) de la fonction objectif et des contraintes. Avec cette application, nous montrons la capacité de SQA à traiter des problèmes de grande taille avec un grand nombre de contraintes pour lesquelles les dérivées ne sont pas disponibles. De plus, SQA est particulièrement efficace pour le traitement des contraintes par rapport à des méthodes évolutionnaires ou des méthodes SQP où les contraintes sont introduites par pénalisation.

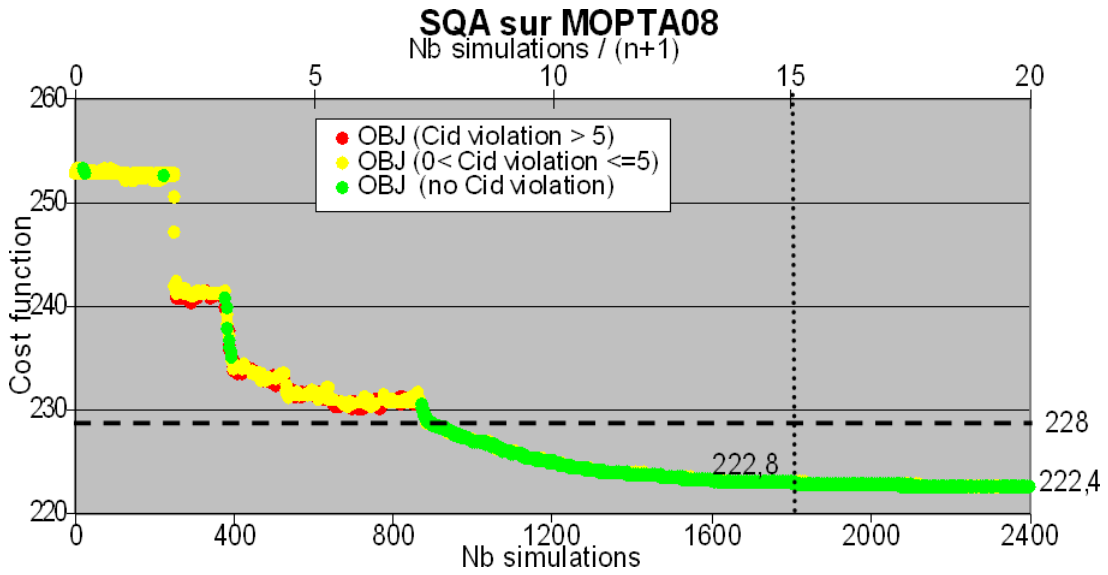


FIGURE 4.19 – Evolution de la valeur de la fonction objectif optimale en fonction du nombre de simulations pour l'optimisation avec la méthode SQA pour un nombre de points d'interpolation de 249 $(2n+1)$

2 Applications

2.1 Applications réservoir

2.1.1 Problème d'inversion de données de production et de données sismiques : problème de calage historique (ou History Matching)

Nous traitons tout d'abord le problème de calage d'historique sans traiter le caractère de moindres carrés de la fonction objectif dans la méthode SQA puis en le prenant en compte.

2.1.1.1 problème de calage d'historique sans prendre en compte le caractère de moindres carrés de la fonction objectif

Le contenu de cette section est basé sur l'article [110].

Introduction

The goal of reservoir characterization is the estimation of unknown reservoir parameters by integrating available data in order to take decisions for production scheme and to predict the production of the field in the future. Reservoir parameters can be classified in two classes: those related to the geological modeling (spatial distribution of porosity, permeability, faults), and those related to the fluid flow modeling (relative permeability curves, productivity index of the wells). These parameters cannot be directly determined by measurements (or only locally using well logs). This is the reason why this parameter estimation problem is formulated as an inverse problem with forward simulators that compute synthetic measurable data from the parameters : production data acquired at production/injection wells (e.g. bottom-hole pressure, gas-oil ratio, oil rate), time lapse seismic data (more precisely compressional and shear wave impedances for different seismic campaigns at different calendar times during the production of the field). The associated forward models consist of a fluid flow simulator and a petro-elastic model (PEM) based on rock physic Gassmann equations. For further details on this application see Roggero et al. (2008). Solving these forward problems is often CPU time consuming and does not provide the derivatives with respect to the parameters.

The optimization problem is formulated as the minimization of a least-squares objective function composed of two terms, one for the production data mismatch and one for the seismic data mismatch. Some weights are introduced to account for data uncertainties and modeling errors. In practice, these problems are often solved by nonlinear optimization methods, as SQP[1] method (Sinoquet and Delbos, 2008) with derivatives approximated by finite differences (FD).
In our application, using the classical Gauss-Newton algorithm is often infeasible because the computation of the Jacobian matrix is CPU time consuming and its storage is impossible for large datasets involving seismic data. Consequently, a natural alternative choice is a Quasi-Newton approach based on classical BFGS approximation of the Hessian of the objective function with derivatives approximated by FD. Although these SQP methods are particularly efficient for the determination of active constraints, the number of function evaluations is usually too high for industrial problems with expensive simulators. Furthermore, the choice of the FD step, crucial for the convergence of this method, is generally cumbersome because it depends on the accuracy of the function computation which is difficult to estimate in practice. This is the reason why, we are interested in Derivative Free Optimization (DFO) methods.

There are mainly four classes of DFO methods in the literature. The first class is composed of direct search methods that explore the variable space by sampling points from a predefined class of geometric patterns or involve some random process (Nelder and Mead, 1965; Audet and Dennis, 2003; Kolda et al., 2003). These methods do not assume generally, smoothness of the objective function and therefore can be applied to a broad class of problems. But, on the other hand, a relatively large number of function evaluations are often required. These methods are theoretically simple and relatively easy to implement. The second class is the class of metaheuristic methods, for instance the evolution strategy (Hansen and Ostermeier, 1996), or the simulated annealing (Kirkpatrick et al., 1983): they explore the parameter space with a population of sampling points evolving towards the global optimum. This class of methods does not assume any regularity assumption on the objective function but requires a large number of function evaluations. The third class of methods combines gradient approximation techniques (e.g. finite difference (Sinoquet and Delbos, 2008) or simplex gradient (Kelley, 1999; Winslow et al., 1991)) with Quasi-Newton methods. These methods are not always robust, especially in the presence of noise, which is often the case for derivative-free applications. The fourth class of methods is based on sequential minimizations of models of the objective function to limit the number of evaluations of the expensive function. These models can be local or global models:

- A global model constructed from a limited number of evaluations of f for parameters values

[1] Sequential Quadratic Programming

chosen according to a relevant criterion (minimum prediction error, space filling, expected gain improvement.), can be, for instance, kriging models (Schonlau, 1997; Jones, 2001; Villemonteix, 2008) or Radial Basis Functions (RBF) (Gutmann, 2001; Regis and Shoemaker, 2007). But they are limited to small size problems (10-30 parameters).

- A local model is generally based on linear or quadratic polynomial interpolation of evaluations of the objective function at sample sets (Conn et al., 2000, 2009a; Powell, 2006, 2007; Marazzi and Nocedal, 2002; Vanden Berghen, 2004). Some recent comparison (Langouët and Sinoquet, 2009) based on a benchmark proposed in Moré and Wild (2009), illustrates good performances of trust region model-based methods compared to three other approaches, even for noisy and piecewise-smooth problems. Even if the three others class of methods are still widely used in the engineering community, they require generally more simulations than the local trust region model-based methods.

In this paper, we propose an adapted method based on local surrogate models (Conn et al., 2000, 2009b; Powell, 2006) belonging to the fourth class of method: these methods are inspired by SQP methods with trust region globalization. The proposed Sequential Quadratic Approximation method (SQA) is an extension of NEWUOA, the efficient Derivative Free Optimization method of Powell (2006), for constrained optimization.

In the first part of this paper, we describe the main features of the SQA method. The second part presents results of this method applied to a challenging benchmark proposed by Jones (2008). The third part is dedicated to a 3D synthetic reservoir application with the joint inversion of production data and 4D seismic data.

SQA Method

The SQA method is a DFO method adapted to constrained optimization of nonlinear function

$$\begin{array}{l} \min\limits_{x\in\mathbb{R}^n} f(x) \\ \text{s.t.} \left\{ \begin{array}{l} C_{DB}(x) \leq 0, \\ C_{DF}(x) \leq 0, \end{array} \right. \end{array} \tag{1}$$

where the derivative free constraints C_{DF} (responses of simulator with unknown derivatives) and the constraints with given derivatives C_{DB} are separated ($C_{DB} : \mathbb{R}^n \to \mathbb{R}^{n_{DB}}$, $C_{DF} : \mathbb{R}^n \to \mathbb{R}^{n_{DF}}$).
The SQA algorithm is given in Algorithm 1. At an initial stage, a quadratic model $\tilde{f}$ of the function f is constructed in a neighborhood of the current point. It interpolates m points[2] (usually $m = 2n + 1$ is chosen) in the admissible domain (at least matching linear constraints). For each derivative free constraint, a quadratic model $\tilde{C}_{DF}$ is also constructed from the same interpolation points. Indeed, for any function f or C_{DF} and any poised set $Y = \{y^0, y^1, ..., y^p\} \subset \mathbb{R}^n$, the minimum-norm interpolating polynomial Q and $\tilde{C}_{DF}$ that interpolate f and C_{DF} on Y can be expressed as

$$\tilde{f}(x) = \sum_{i=0}^{p} f(y^i) l_i(x), \ \tilde{C}_{DF}(x) = \sum_{i=0}^{p} C_{DF}(y^i) l_i(x), \tag{2}$$

where $\{l_i(x), i = 0, ..., p\}$ is the set of minimum-norm Lagrange polynomials for Y (see Conn et al. (2009b)).

Then, at each iteration the problem is solve for the current precision on parameters ρ[3]: the current quadratic model $\tilde{f}$ is minimized under constraints C_{DB} and models of derivative free constraints $\tilde{C}_{DF}$

[2]The incomplete quadratic models $(n + 2 \leq m \leq (n + 1)(n + 2)/2)$ are completed by minimizing the Frobenius norm of the hessian matrix variations of the models (Powell, 2006)

[3]ρ is the current resolution expected on x. It is updated during optimization process (see in Algorithm 1).

in a trust region of radius Δ around the optimal point by a SQP method:

$$\begin{array}{l} \min\limits_{d\in\mathbb{R}^n} \tilde{f}(x_{opt}+d) \\ \text{s.t.} \left\{ \begin{array}{l} C_{DB}(x_{opt}+d) \leq 0, \\ \tilde{C}_{DF}(x_{opt}+d) \leq 0, \\ \|d\| \leq \Delta. \end{array} \right. \end{array} \tag{3}$$

f and C_{DF} are evaluated at $x^* = x_{opt} + d_k$, solution of (2).

If the merit function

$$\varphi(x) = f(x) + \sum_{i=1}^{n_{DF}} \lambda_i C_{DF_{(i)}\#}(x)^4, \tag{4}$$

measuring the constraint violation and f, has decreased, x^* is validated as the new current optimal point x_{opt}, where λ is supposed to represent the unknown Lagrange multipliers in this algorithm. λ is set to a constant value that takes into account the normalization of the function f and the constraints C_{DF}.

Otherwise, another point is added in order to improve the quadratic models: a criterion based on the interpolating Lagrange polynomial is maximized, see Powell (2006). Finally, the trust region radius Δ is updated according to the comparison of the reduction of the merit function based on models $\tilde{f}$ and $\tilde{C}_{DF}$ and the effective reduction of the merit function based on f and C_{DF} :

$$R = \frac{\varphi(x_k) - \varphi(x_k + d_k)}{\tilde{\varphi}(x_k) - \tilde{\varphi}(x_k + d_k)}. \tag{5}$$

So, Δ is increased if the "modeled" merit function has the correct trends compared to the "real" merit function or decreased otherwise. The quadratic models $\tilde{f}$ and $\tilde{C}_{DF}$ are updated to interpolate the m closest points to the current optimal point (where f and C_{DF} were evaluated). The algorithm is stopped when the minimal trust region radius (accuracy on the solution requested by the user) is reached or when the maximum number of simulations given by the user has been performed.

MOPTA08 Benchmark

The MOPTA08 benchmark is a very complex and challenging minimization problem proposed by Jones (2008) : it was built from a real automotive problem. The goal is to optimize vehicle performance subject to security constraints. The simulation (e.g. crash test) is very expensive in terms of computationnal time: 60 simulations can optimistically be computed per day. This function depends on 124 parameters under 68 derivative free constraints (blackbox constraints).

In order to share this application with the optimization community, Jones (2008) built kriging response surface of the objective function and of the constraints from simulations of the real problem. The main objective of this benchmark is to reduce the objective function value below a threshold of 228 (the initial function value is 251) in less than 1800 evaluations (15 times the number parameters which corresponds to one month of CPU time of the real simulator).

The result obtained by SQA method is presented in Figure 1, where the evolution of the objective function is depicted with respect to the number of simulations.

They are very promising: indeed, the threshold of 228 is reach in 800 simulations and all the constraints are satisfied. Jones has shown that lot of tested optimization methods failed to solve this problem: a classical SQP with finite differences, an evolution strategy method,... Only, one method based on RBF (radial basis functions) approximation of the objective function and of the constraints (Regis, 2009)

[4] with $C_\# = maxC, 0$.

Algorithm 1 SQA Algorithm (Sequential Quadratic Approximation)

(1) Initialization : Select m initial interpolation points. x_{opt} is the initial point for which f is minimal among the m points. Determine the first quadratic model $\tilde{f} \approx f$ of the objective function f and the quadratic models $\tilde{C}_{DF}$ of the derivative free constraints C_{DF}.

Δ is the trust region radius: $\rho = \Delta = \rho_{beg}$.

(2) Solve the problem for a precision on x: ρ

(a) (i) Minimize the model :
$\min_d \tilde{f}(x_{opt} + d)$

$$\text{s.t.} \begin{cases} \|d\| \leq \Delta, \\ C_{DB}(x_{opt} + d) \leq 0 \\ \tilde{C}_{DF}(x_{opt} + d) \leq 0 \end{cases}$$

(ii) If $\|d_k\| < \frac{1}{2}\rho$: $\rightarrow$ (2)(b) (necessary to ensure the validity of the model before doing small steps).

(iii) Calculate $f(x_{opt} + d_k)$, $C_{DF}(x_{opt} + d_k)$,
$\varphi(x_{opt}+d_k) = f(x_{opt}+d_k)+\sum_{i=1}^{n_{DF}} \lambda_i C_{DF_{(i)}\#}(x_{opt}+d_k)^4$ and update the trust region radius Δ from the predictivity of the quadratic models $R = \frac{\varphi(x_{opt})-\varphi(x_{opt}+d_k)}{\tilde{\varphi}_k(x_{opt})-\tilde{\varphi}_k(x_{opt}+d_k)}$.
Update interpolation points : $x_{opt} = x_{opt} + d_k$ if $\varphi(x_{opt} + d_k) < \varphi(x_{opt})$ and update the models $\tilde{f}$ and $\tilde{C}_{DF}$ in order to interpolate the function f and the derivative free constraints C_{DF} in $x_{opt} + d_k$.

(iv) If $R > 0.1 \rightarrow$ (2)(a)(i)
Otherwise continue.

(b) Test the validity of the model $\tilde{f}$ and $\tilde{C}_{DF}$. The model is considered valid if all the current interpolation points x_i, $i = 1, ..., m$ are close to the optimal point x_{opt}, i.e. if the Euclidean distance $\|x_i - x_{opt}\| < 2\Delta$ for all x_i, $i = 1, ..., m$.

(i) If the model is not valid :
Improve the quality of the model $\tilde{f}$ and $\tilde{C}_{DF}$:
$\max_{d\in\mathcal{R}^n} |l_t(x_{opt} + d)|$

$$\text{s.t.} \begin{cases} \|d\| \leq \overline{\Delta}, \\ C_{DB}(x_{opt} + d) \leq 0 \end{cases}$$

Calculate $f(x_{opt} + d_k)$ and $C_{DF}(x_{opt} + d_k)$, the worst point of the interpolation set is replaced by a new point closer to the current minimum.
Assuming that the accuracy of the updated models $\tilde{f}$ and $\tilde{C}_{DF}$ has been improved $\rightarrow$ (2)(a)(i).

(ii) If the model is valid : If $\|d_k\| > \rho \rightarrow$ (2)(a)(i)
Otherwise continue

(c) If $\rho > \rho_{end}$: reduction of ρ and $\Delta \rightarrow$ (2)(a) (zoom on x)
Otherwise end of the algorithm.

gives similar performances than SQA method. With this application, we show the ability of SQA to deal with large size problem with a large number of constraints. In the next section, SQA is applied to a realistic reservoir application.

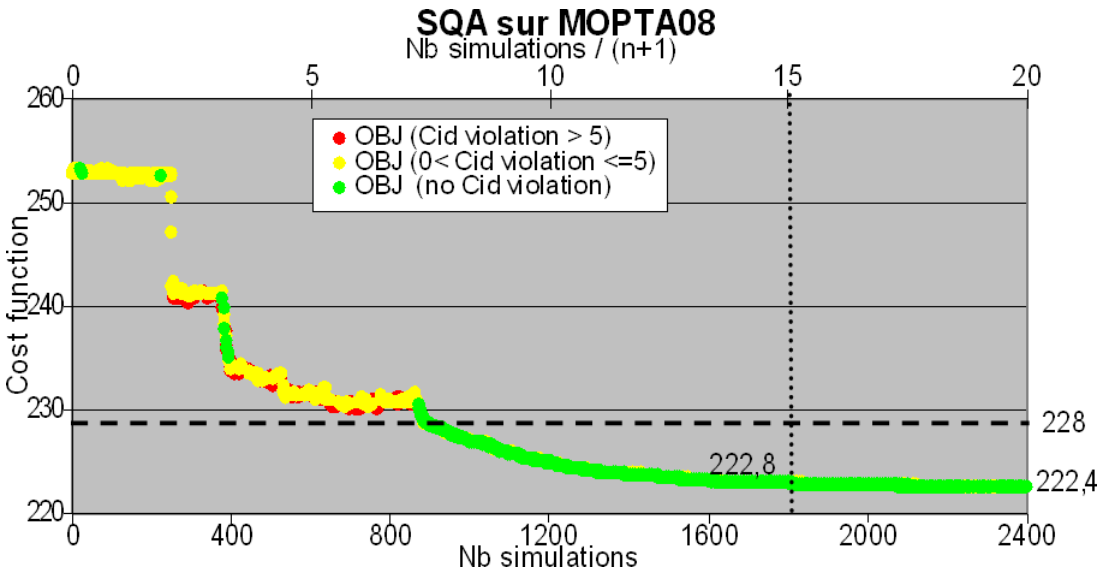


Figure 1: Objective function versus simulation number for optimization with SQA method for 249 interpolation points ($2n + 1$).

Reservoir application: inversion of monitor seismic datasets

The PUNQ test case is a 3D synthetic reservoir model derived from real field data. It was already used for comparative inversion studies in the European PUNQ project (Floris, 2001) and for validation of constrained modeling and optimization scheme development methods (Roggero, 2001). The top structure of the reservoir is presented in Figure 2.

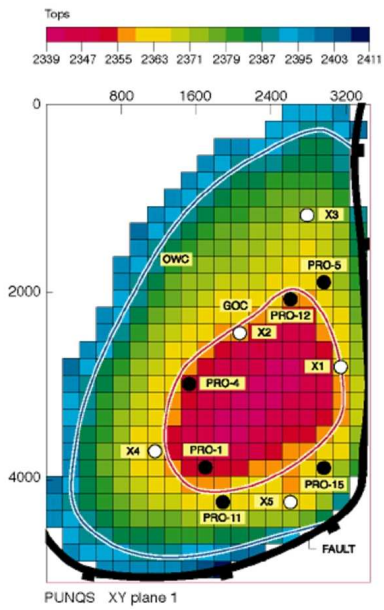


Figure 2: Top structure of the reservoir.

The reservoir is surrounded by an aquifer in the north and the west, and delimited by a fault in the south and the east. A small gas cap is initially present. The geological model is composed of five independent layers. The layers 1, 3, 4 and 5 are assumed to be of good quality, while the layer 2 is of poorer quality.

The initial model consists of a $19 \times 28 \times 5$ grid, with a constant step of 180 m in the horizontal X and Y directions. In order to illustrate the potential of the algorithm on large seismic data sets, we decided to build a larger model. Therefore, our reference model consists of a $76 \times 56 \times 5$ grid, with a constant step of 45 m in the X direction and 90 m in the Y direction. Each geological unit is modeled by one layer, with a Gaussian distribution of the porosities and a spherical variogram. The geostatistical simulation parameters are listed in Table 1. The permeability on each layer is defined by a $(\log K - \phi)$ relationship, i.e. $\log(K_x) = A\phi + B$ with constant ratios K_y/K_x and K_z/K_x. The corresponding reference parameters are given in Table 1.

Table 1: Corresponding reference of geostatistical simulation parameters.

	ϕ mean	ϕ variance	A	B	Ky/Kx	Kz/Kx
Layer 1	0.1722	0.0078	8.585	0.701	1	0.364
Layer 2	0.0802	0.0004	14.383	0.258	1	0.339
Layer 3	0.1677	0.0050	8.683	0.781	1	0.314
Layer 4	0.1615	0.0006	4.209	1.789	1	0.211
Layer 5	0.1892	0.0049	8.98	0.793	1	0.296

Then, the geostatistical simulations are upscaled to come back to the original $19 \times 28 \times 5$ grid in order to work with a faster fluid flow simulation in the reservoir. The synthetic production data are produced by a numerical simulation using the ATHOS model over a eight-year period. The production results selected as synthetic measurements are the gas oil ratio (GOR), the bottomhole pressure (BHP) and the water cut value (WCUT) at the six producing wells (PRO-1, 4, 5, 11, 12 and 15). We give on the left of Figure 3 the reference production data for all wells.

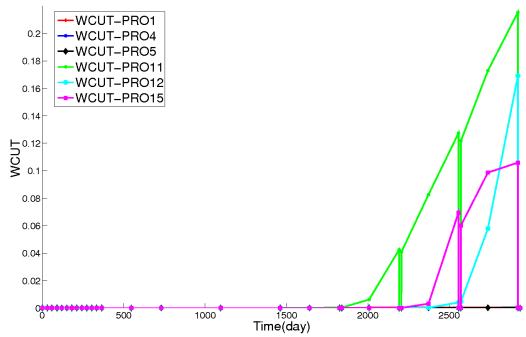
Then, pressure and saturations maps simulated by the ATHOS model at times 0 days, 181 days (half a year) and 2192 days (six years) are extracted and downscaled to the $76 \times 56 \times 5$ grid. A petro-elastic model (PEM) involving Gassmann and Hertz equations is defined with given bulk and dry modulus, bulk densities and Hertz exponents. The combination of downscaled pressure and saturations maps with this petroelastic model allows to compute synthetic P and S impedance maps at times 0, 181 and 2192 days. These maps serve as a synthetic 4D seismic data set. An example of the reference impedance map on layer 5 at time 0 and Delta impedance maps at times 181 and 2192 is depicted on the right of Figure 3.

For history matching, the parameters of the simulation model are constrained by both production and seismic data. An optimal matching is sought by minimization of an objective function defined as follows:

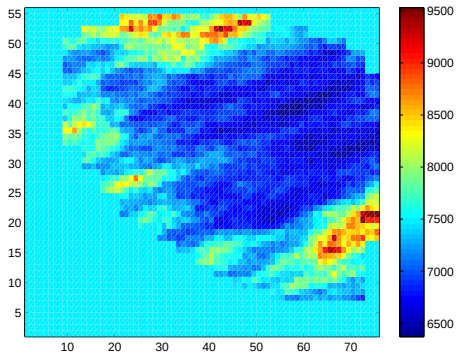
$$f(x) = \frac{1}{2}\sum_{i=1}^{6}\sum_{j=1}^{n_p}(d_{P_i}^{sim}(x,t_j) - d_{P_i}^{obs}(t_j))^2 + \frac{1}{2}\sum_{k=1}^{n_{grid}}(d_S^{sim}(x,k) - d_S^{obs}(k))^2, \tag{6}$$

where $i = 1, ..., 6$ is the well index, t_j, $j = 1, ..., n_p$ are the measurement times of production data and $k = 1, ..., n_{grid}$ denotes the cell indices. The geometric data, the geological structure, the fluid properties and the geomechanical parameters of the PEM are presumed known. The inversion parameters are the porosity means (one for each layer), the A and B coefficients for the permeabilities (two coefficients

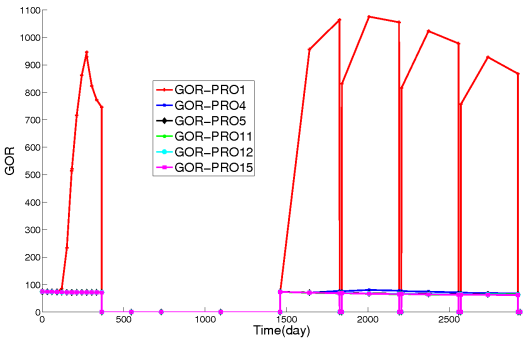
per layer) and the permeability ratios (two ratios per layer). These parameters are submitted to bound constraints.



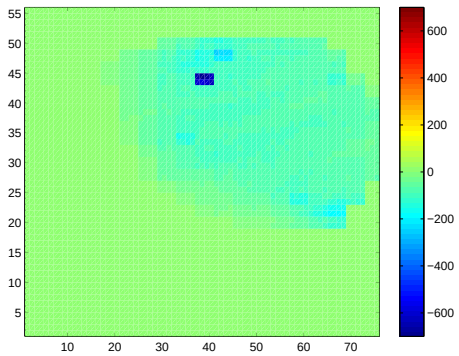
(a) WCUT



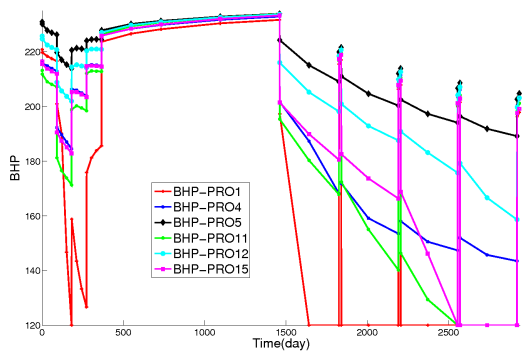
(b) I_P at time 0



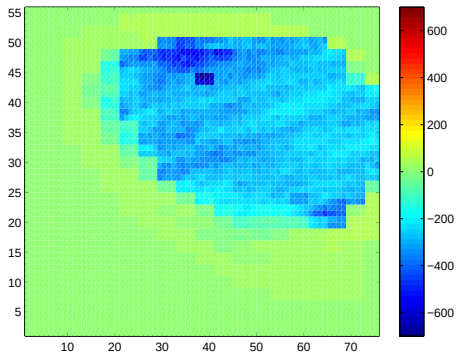
(c) GOR



(d) ΔI_P at time 181



(e) BHP



(f) ΔI_P at time 2192

Figure 3: PUNQ field: Reference production data (left), seismic data (P impedance map) for layer 5 (right).

The SQPAL BFGS method (Sinoquet and Delbos, 2008), the SQA method for differents number of interpolation points ($2n+1$, $n+6$ and $n+2$) and the SQPAL Gauss Newton (GN) method are applied to the minimization problem of the objective function (6) with respect to these 25 inversion parameters un-

der bound constraints. Both BFGS and Gauss-Newton approaches belong to the class of Quasi-Newton methods which solve sequentially a minimization problem based on a quadratic objective function with the same gradient as the original objective function and an approximation of its Hessian matrix:

$$\min_d \frac{1}{2} d^T \tilde{H} d + g^T d. \tag{7}$$

The BFGS formula provides an update of the approximation of the Hessian matrix at each iteration depending on the gradients of the objective function and of the parameter values

$$\tilde{H}_{k+1} = \tilde{H}_k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{H_k \delta_k \delta_k^T H_k^T}{\delta_k^T H_k^T \delta_k}, \tag{8}$$

with $\delta_k = x_{k+1} - x_k$ and $\gamma_k = \nabla_x f(x_{k+1}, \lambda_{k+1}) - \nabla_x f(x_k, \lambda_k)$. For Gauss-Newton method, the approximation of the Hessian matrix is computed from the Jacobian matrix J, first derivatives of the calculated data ($J_{ij} = \frac{\partial d_i}{\partial x_j}$):

$$\tilde{H} = J^T J. \tag{9}$$

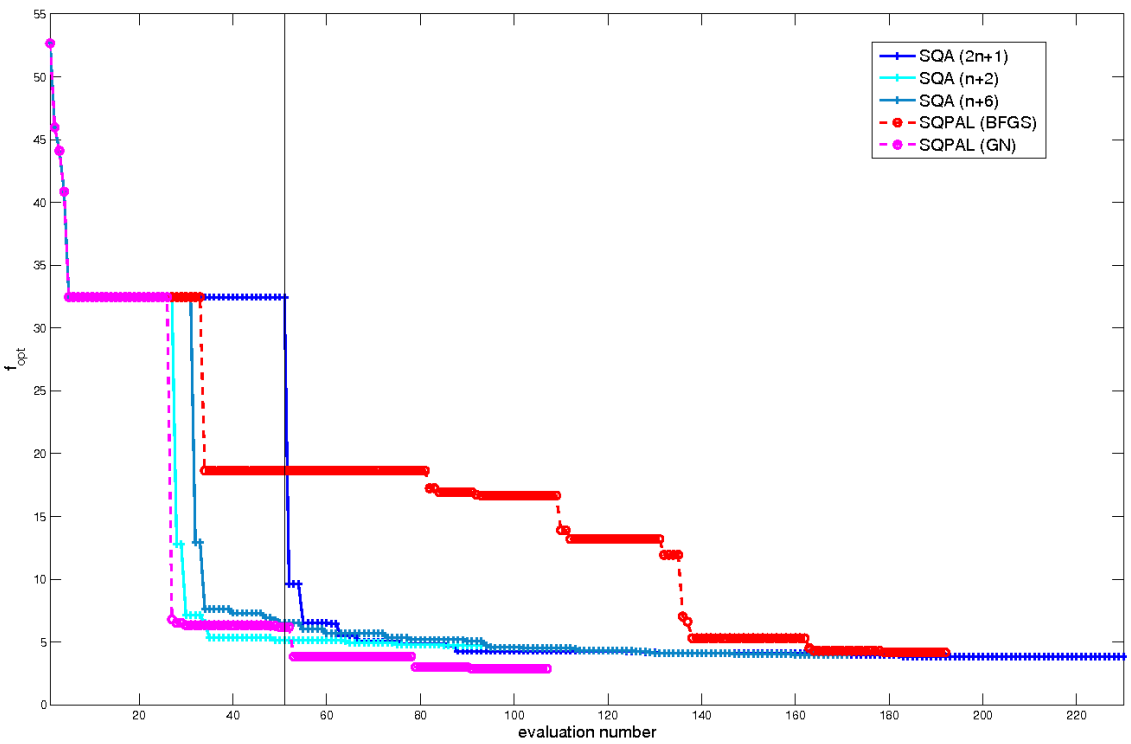


Figure 4: Objective function versus simulation numbers for five optimization runs, respectively with SQA for differents number of interpolation points (solid lines, blue crosses), with BFGS method with line search globalization (dashed line, red circles), and with GN method (dashed line, pink circles).

In both cases, the first derivatives of the objective function (BFGS) or of the calculated data (Gauss-Newton) are computed by finite differences which requires at each iteration $n + 1$ simulations of the forward operator. Moreover, Gauss-Newton approach needs the storage of the Jacobian matrix of size $n \times n_{data}$ ($n_{data} = n_p + n_{grid}$ is the total number of observed data). The storage of such a matrix

becomes cumbersome for joint inversion of production data and seismic data because of the huge number of measurements. In our application case, this storage is still possible for 128500 measurements but it will be untractable for more complex problems with larger number of measurements. The SQA method requires as the BFGS method only the storage of the gradient and of the Hessian matrix of size depending only on the number of parameters.

The results obtained by the five methods are presented in Figure 4, where the evolution of the objective function is depicted with respect to the number of fluid flow and petroelastic simulations. In all cases, we can observe that the objective function decreases: the five different methods seem to converge to a local minimum. As expected, we can also notice that SQA (version $2n+1$ interpolation points) requires more evaluations during the initialization phase than SQPAL ($m = 2n + 1$ compared to FD computations which require $n + 1$) and others SQA (version $n + 6$ and $n + 2$). However, SQA achieves in all cases a greater reduction of the objective function much faster than SQPAL BFGS. This is the key advantage of SQA versus SQPAL BFGS: it allows to reduce the number of evaluations of the simulator, which are very CPU time consuming. We can also observe that SQPAL GN achieves a fastest reduction of the objective function than other methods through the use of information of the jacobian matrix. SQA (version $n + 2$) is the only method with performances closed to SQPAL GN.

The production data at well 12 obtained by the initial model and the SQA matched model are compared to the reference data on the left of Figure 5. SQA provides a model which is much closer to the reference production data than the initial model. The reference, initial, and optimal impedance map of layer 5 at time 0 obtained with the SQA method is given on the right of Figure 5. We can observe that SQA succeeds to give a good match also for seismic data.

SQA method is then able to solve this large size reservoir history matching problem (in terms of number of measurements) without requiring to store the large size Jacobian matrix. It is quite encouraging for future real applications involving complex datasets and computationnaly expensive simulators.

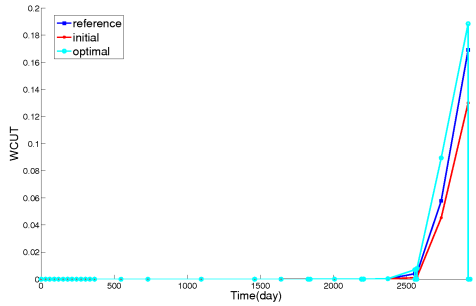
Conclusions

In this paper, we proposed a Derivative Free Optimization method based on Sequential Quadratic Approximations (SQA) of the objective function to solve the joint inversion of production and 4D seismic data. As shown on a realistic application, this method allows to limit the number of evaluations of the computationnaly expensive forward problem compared to classical optimization method coupled with gradient approximated by finite differences as BFGS Quasi-Newton method. We compared also SQA method with Gauss-Newton method : in that case, we obtained comparable results when using a small number of interpolation points to build the quadratic model of the objective function (a quadratic polynomial model built from $n + 2$ points). These results are very encouraging for applications with a large number of measurements: for such cases, Gauss-Newton method is not applicable due to storage requirement (the Jacobian matrix is too huge to be stored).

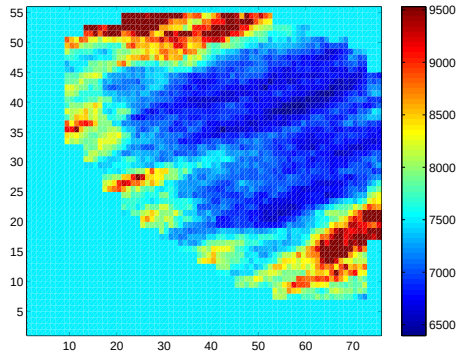
An other interesting feature of SQA method shown in the paper is its ability to deal efficiently with nonlinear constraints (with or without given derivatives). SQA succeeded to solve a challenging optimization benchmark with 68 black-box constraints (with unknown derivatives) and 124 parameters. A study of a reservoir engineering application with nonlinear constraints is in progress and preliminary results are presented in Metla et al. (2010). The aim of this work is to compute parameter uncertainty estimations and confidence intervals of a forecast response based on solutions of history matching problems.

In order to take into account the nature of the objective function in history matching problem, namely its least-square characteristic, we are currently working on adapting SQA method to construct several models to form the objective function (basically, one model for each data misfit or for each type of data misfit), following the idea proposed by Zhang et al. (2009). The objective is to obtain more ac-

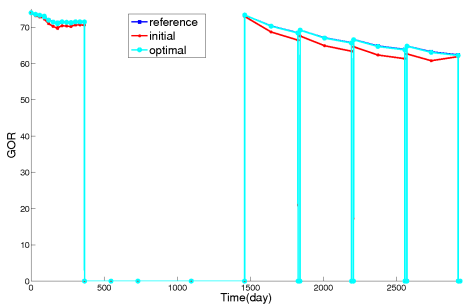
curate models taking into account the characteristics of the objective function, and then obtain a faster convergence.



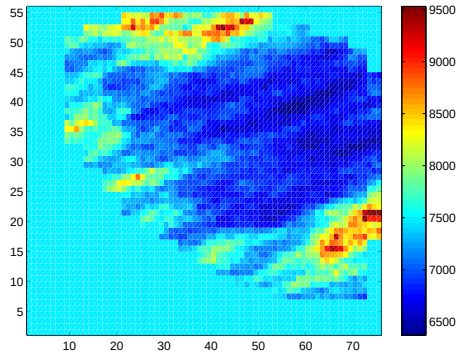
(a) WCUT for well 12



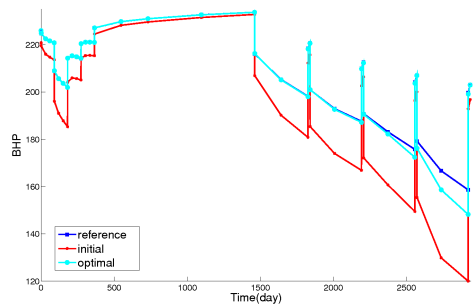
(b) Initial



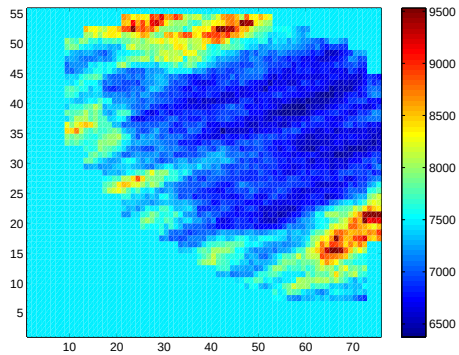
(c) GOR for well 12



(d) Reference



(e) BHP for well 12



(f) Optimal

Figure 5: Comparison between reference production data (blue) and simulated production data for initial point (red) and optimal point obtained with SQA (cyan) (left). P Impedance maps at time 0 obtained with SQA (right).

References

Audet, C. and Dennis, J.E. [2003] Analys of generalized pattern searches. *SIAM J. Optim.*, **13**, 889–903.

Conn, A.R., Gould, N.I.M. and Toint, P.L. [2000] *Trust-Region Methods*. MPS-SIAM Series on Optimization, SOIAM Philadelphia.

Conn, A.R., Scheinberg, K. and Vicente, L.N. [2009a] Global convergence of general derivative-free trust-region algorithms to first and second order critical points. *SIAM J. Optim.*

Conn, A.R., Scheinberg, K. and Vicente, L.N. [2009b] *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics (MPS-SIAM Series on Optimization), Philadelphia, PA, USA.

Floris [2001] *Methods for quantifying the uncertainty of production forecasts - a comparative study*, vol. 7. published in J.Petroleum Geoscience.

Gutmann, H.M. [2001] *Radial Basis Function Methods for Global Optimization*. Ph.D. thesis, University of Cambridge.

Hansen, N. and Ostermeier, A. [1996] Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. *In Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, p. 312-317*.

Jones, D. [20 August 2008] Large-scale multi-disciplinary mass optimization in the auto industry. *Presented at the MOPTA 2008 Conference*.

Jones, D.R. [2001] A taxonomy of global optimization methods based on response surfaces. *J. of Global Optimization*, **21**, 345–383.

Kelley, C.T. [1999] Iterative methods for optimization. *SIAM*.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. [1983] Optimization by simulated annealing. *Science*, **220**, 671–680.

Kolda, T.G., Lewis, R.M. and Torczon, V. [2003] Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, **45, No. 3**, 385–482.

Langouët, H. and Sinoquet, D. [2009] Derivative free optimization under constraints. *ENBIS EMSE, 1-3 July 2009*.

Marazzi, M. and Nocedal, J. [2002] Wedge trust region methods for derivative free optimization. *Math. Program.*, **91**, 289–305.

Metla, N., Delbos, F., Da Veiga, S. and Sinoquet, D. [2010] Constrained nonlinear optimization for extreme scenarii evaluation in reservoir characterization. *ECMOR European Conference on the Mathematics of Oil Recovery, 12th, Oxford, UK, 6-9 september 2010*.

Moré, J.J. and Wild, S.M. [2009] Benchmarking derivative-free optimization algorithms. *Siam J. Optim.*, **20**, 172–191.

Nelder, J.A. and Mead, R. [1965] A simplex method for function minimization. *Comput. J.*, **7**, 308–313.

Powell, M.J.D. [2006] The newuoa software for unconstrained optimization without derivatives. Tech. rep., Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Cambridge.

Powell, M.J.D. [2007] Developments of newuoa for unconstrained minimization without derivatives. Tech. rep., University of Cambridge.

Regis, R.G. [August 2009] Radial basis function algorithms for large-scale nonlinearly constrained black-box optimization. *20th International Symposium on Mathematical Programming Chicago, IL*, vol. 37, 113–135.

Regis, R.G. and Shoemaker, C.A. [2007] Improved strategies for radial basis function methods for global optimization. *J. of Global Optimization*, **37**, 113–135.

Roggero [2001] Constraining stochastic reservoir models to dynamic data: an integrated approach. Rueil, France, OAPEC seminar.

Roggero, F. et al. [2008] Constraining reservoir models to production and 4d seismic data - application to the girassol field. Madrid, Offshore Angola, 19th World Petroleum Congress.

Schonlau, M. [1997] *Computer Experiments and Global Optimizations*. Ph.D. thesis, the University of Waterloo.

Sinoquet, D. and Delbos, F. [2008] Adapted nonlinear optimization method for production data and 4d seismic inversion. *ECMOR European Conference on the Mathematics of Oil Recovery, 11th, Bergen, Norway, 8-11 september 2008*.

Vanden Berghen, F. [2004] *CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions.* Ph.D. thesis, Université Libre de Bruxelles, Faculté des Sciences Appliquées.

Villemonteix, J. [2008] *Optimisation De Fonctions Coûteuses: Modèles Gaussiens Pour Une Utilisation Efficace Du Budget D'évaluations : Théorie Et Pratique Industrielle*. Ph.D. thesis, Université Paris-Sud Xi Faculté Des Sciences D'Orsay.

Winslow, T.A., Trew, R.J., Gilmore, P. and Kelley, C.T. [1991] Doping profiles for optimum class b performance of gaas mesfet amplifiers. *Proceedings of the IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits*, 188–197.

Zhang, H., Conn, A.R. and Scheinberg, K. [2009] A derivative-free algorithm for the least-square minimization. Tech. rep., Department of Mathematics, Louisiana State University.

2.1.1.2 problème de calage d'historique en prenant en compte le caractère de moindres carrés de la fonction objectif

Nous avons tenté de résoudre le problème de calage historique, exposé précédemment, qui est un problème de moindres carrés, à l'aide de l'implémentation SQA prenant en compte cette particularité de la fonction (voir section 4.3 du chapitre 3) en construisant plusieurs modèles pour former la fonction objectif.

Les données de production ont été groupées selon les trois propriétés considérées suivantes :

- le ratio huile / gaz,
- la pression au fond du puits et
- le rapport entre eau produite sur huile produite,

ainsi que pour chacun des 6 puits. Ces données de production représentent donc 18 modèles (3 propriétés, 6 puits) construits à partir de 42 mesures par modèle.

Les données sismiques sont les impédances P et S regroupées à trois pas de temps différents (initial, après 181 jours et après 2 192 jours). Cela représente 6 modèles (2 propriétés, 3 pas de temps) construits à partir de 21 280 mesures par modèle.

Nous cherchons tout d'abord à résoudre le problème précédent uniquement pour caler les données de production. Le nombre de données est alors uniquement de $n_{ls} = 756$ pour ce problème simplifié. Nous construisons soit un modèle par paquet de données ($n_m = 18$ modèles), soit un modèle pour chacune des données $n_m = n_{ls} = 756$. La figure 4.20 représente la décroissance de la fonction objectif en fonction du nombre d'évaluations de celle-ci obtenue par les méthodes SQPAL Gauss Newton, SQA $n + 2$ (classique, LS2 $n_m = 18$ et $n_m = n_{ls} = 756$) et SQA $2n + 1$ LS2 $n_m = 18$.

Les résultats obtenus par SQA $n+2$ LS2 ($n_m = 18$ et $n_m = n_{ls}$) sont plus précis et plus rapides que ceux obtenus par la méthode SQPAL Gauss Newton. De plus la convergence est nettement plus rapide que celle obtenue par la méthode SQA $n + 2$ classique. Le résultat obtenu par la méthode SQA LS2 $2n + 1$ manque de précision, comparé à celui de la méthode SQA $n + 2$ classique.

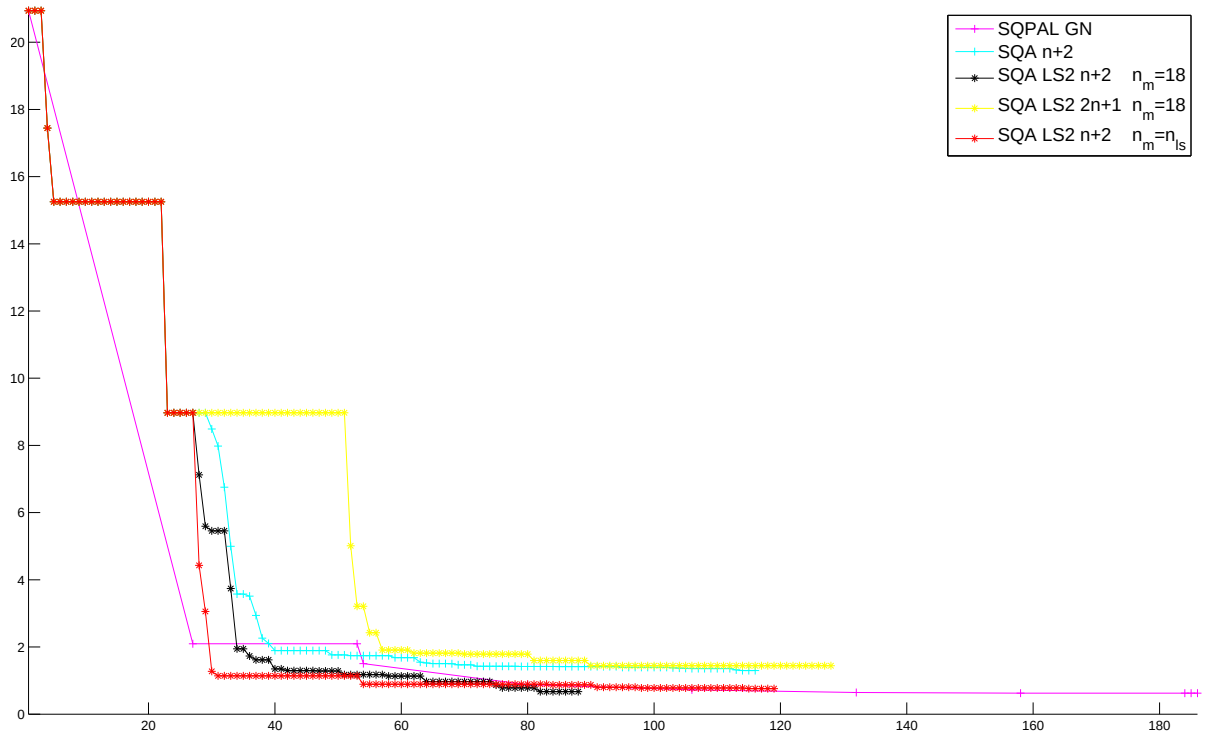


FIGURE 4.20 – Fonction objectif en fonction du nombre d'évaluations, avec SQA et SQA Least Square pour deux nombres de points d'interpolation, puis avec SQPAL avec méthode GN

Nous cherchons maintenant à résoudre le problème initial pour caler les données de production et les données sismiques. Le nombre de modèles construits pour prendre en compte les données est $n_m = 24$ pour $n_{ls} = 128\,436$ mesures, le nombre de données étant maintenant trop élevé pour construire un modèle par données. Les résultats obtenus à l'aide des méthodes SQPAL BFGS et SQPAL Gauss Newton, ainsi que pour les méthodes SQA $n+2$ (classique, LS1 et LS2) et SQA $2n+1$ (classique et LS2) sont représentés sur la figure 4.21.

La méthode SQA LS1 $n+2$, comme sur les exemples de la section 4.3 du chapitre 3, est à peu près équivalente à la méthode SQA $n+2$ classique. La méthode SQA LS2 $n+2$ qui était assez efficace sur les exemples du chapitre précédent ne fonctionne pas mieux que le SQA $n+2$ classique sur ce cas. La méthode SQA LS2 $2n+1$ est à peu prêt équivalente à la méthode SQA $2n+1$ classique, mais permet d'obtenir une légère accélération de la convergence.

La prise en compte du caractère de moindres carrés par la méthode SQA sur cette application n'est pas très efficace. Les résultats seraient plus intéressants si le nombre de modèles construits $n_m = 24$ pouvait être plus important par rapport au nombre de données $n_{ls} = 128\,436$.

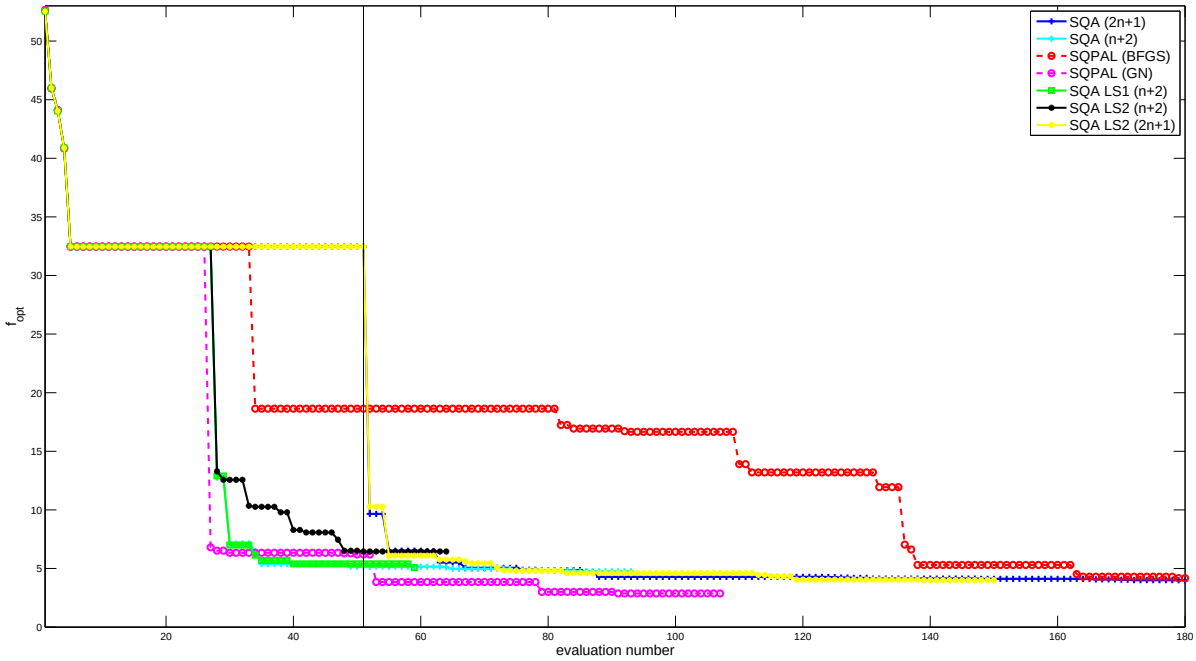


FIGURE 4.21 – Fonction objectif en fonction du nombre d'évaluations, avec SQA et SQA Least Square pour deux nombres de points d'interpolation, puis avec SQPAL avec méthode BFGS ou avec méthode GN

On pourrait envisager de grouper les données de façon plus adéquate en prenant en compte la signification physique des données. L'idée serait de construire un modèle par couche géologique (en utilisant l'information à priori issue de la géostatistique par exemple). On construirait ici 30 modèles pour la sismique (2 propriétés, 3 pas de temps, 5 couches) pour un total de $n_m = 48$ modèles avec les données de production.

Le caractère de moindres carrés de la fonction objectif a été pris en compte par la méthode SQA, comme présenté dans la section 4.3 du chapitre 3. Les résultats obtenus sur le cas avec sismique ne sont pas améliorés de manière significative, le nombre de données étant trop important pour la méthode. En revanche, sur le cas sans sismique, une amélioration très intéressante a été apportée par la prise en compte du caractère de moindres carrés de la fonction objectif.

2.1.2 Calcul des scénarios extrêmes

Une méthodologie d'optimisation non linéaire permettant de prédire la production future de pétrole d'un champ pétrolier ou gazier en exploitation et de donner un intervalle de confiance sur cette prédiction fait l'objet de l'article [118]. L'utilisation de telles méthodes d'optimisation devrait augmenter la précision et donnera ainsi des prévisions de

production plus fiables que les approches basées sur des modèles simplifiés du simulateur (approximation linéaire ou surface de réponse).

Comme présenté dans la section précédente 2.1.1 de ce même chapitre, le problème d'inversion consiste à minimiser la fonction objectif représentant l'erreur quadratique de la réponse du modèle par rapport à la réponse mesurée en chaque puits. On définit ce problème comme suit :

$$\min_{x \in \mathbb{R}^n} HM(x) = \|d(x) - d_{obs}\|^2_{[T_0,T_1]} \tag{4.4}$$

sous des contraintes de bornes, où $x = (x^1, ..., x^n) \in \mathbb{R}^n$ sont les paramètres caractérisant le réservoir, d_{obs} et $d(x)$ sont respectivement les données observées et simulées à partir du temps initial T_0 jusqu'au temps final $T_1 = T_{HM}$ (HM pour history matching ou calage historique). Le problème (4.4) est un problème d'optimisation de moindres carrés non linéaire sous contrainte de bornes. Le calcul de $d(x)$ provient d'un simulateur d'écoulement des fluides dans le réservoir (voir [61, 68] pour plus de détails), ce simulateur est coûteux et les dérivées de ses réponses ne sont pas disponibles. La solution de ce problème est notée dans ce paragraphe x_{HM}.

Le problème de prévision est généralement résolu par une approche probabiliste. Des techniques statistiques [60] permettent l'optimisation de la production cumulée de pétrole et l'estimation des incertitudes associées. Par ailleurs, l'approche en deux étapes proposée par Metla et al (voir [118]) est basée sur l'optimisation non linéaire :

- Tout d'abord, une fois le modèle de réservoir x_{HM} solution de (4.4) obtenu, le simulateur d'écoulement de fluide sera exécuté une fois supplémentaire pour obtenir les réponses jusqu'au temps T_2 futur ($T_2 > T_1$). Ainsi, on peut donner, une première prévision de la production de pétrole cumulée à venir (voir figure 4.22). Cette méthode correspond à la première phase du problème de prévision.
- Ensuite, le problème de prévision ne consiste pas seulement à trouver une valeur de prédiction, mais également à estimer son incertitude. La solution x_{HM} est seulement une solution parmi d'autres qui approchent les données observées d_{obs}. En effet, d'autres paramètres réservoirs $\tilde{x}$, différents de x_{HM} correspondant à des valeurs simulées approchant les données observées d_{obs} avec la même précision que $HM(x_{HM})$ sur $[T_0, T_1]$.

L'approche par optimisation du problème de prévision (proposée en premier dans [79]) équivaut à trouver des modèles de réservoir qui maximisent et minimisent la production de pétrole jusqu'au temps T_2 et qui, en même temps, s'approchent des données observées sur $[T_0, T_1]$ avec une précision suffisante. Cela revient à rechercher les bornes inférieures et supérieures sur cette prédiction pour trouver l'intervalle de confiance de la prévision. Les scénarios de prévision optimiste et pessimiste sont définis comme suit :

- Le scénario optimiste consiste à identifier les paramètres qui maximisent la fonction de production prédite (jusqu'au temps T_2) en approchant les données observées avec une précision donnée.
- Le scénario pessimiste consiste à identifier les paramètres qui minimisent la fonction de production prédite en approchant les données observées avec une précision

donnée.

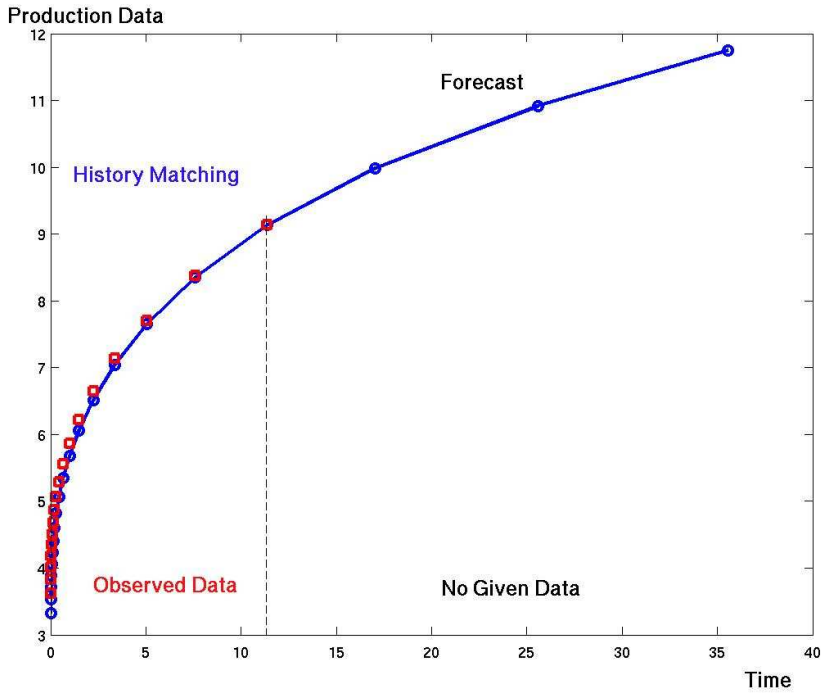


FIGURE 4.22 – Etape 1 du problème de prévision : points rouge – données observées ; ligne bleue – données simulées (de T_0 à T_2)

D'après les définitions précédentes, le problème de prévision est formulé comme le problème d'optimisation suivant :

$$\max_x / \min_x p(x, T_2), \tag{4.5}$$

sous des contraintes de bornes sur x et la contrainte suivante :

$$HM(x) \leq \varepsilon, \tag{4.6}$$

où ε est l'erreur absolue sur la fonction coût de calage historique et HM est défini par l'équation (4.4). L'équation (4.5) consiste à minimiser ou à maximiser la fonction objectif représentant la production cumulée de pétrole totale p au temps T_2. L'équation (4.6) est une contrainte non linéaire de calage historique. Les scénarios de prédiction optimiste et pessimiste pour une valeur donnée ε sont représentés sur la figure 4.23. Notons que si p est pour cette problématique, la production cumulée de pétrole, il peut également être défini par une autre fonction construite à partir des réponses du simulateur.

Les deux problèmes de calage historique (4.4) (voir la section précédente 2.1.1) et de prévision (4.5-4.6) correspondent à des problèmes d'optimisation non linéaire sans dérivées (les dérivées des réponses du simulateur ne sont pas disponibles). Le second problème de prévision présente la difficulté supplémentaire d'être un problème d'optimisation sous une

contrainte non linéaire dont les dérivées ne sont pas disponibles (contrainte sur le calage historique).

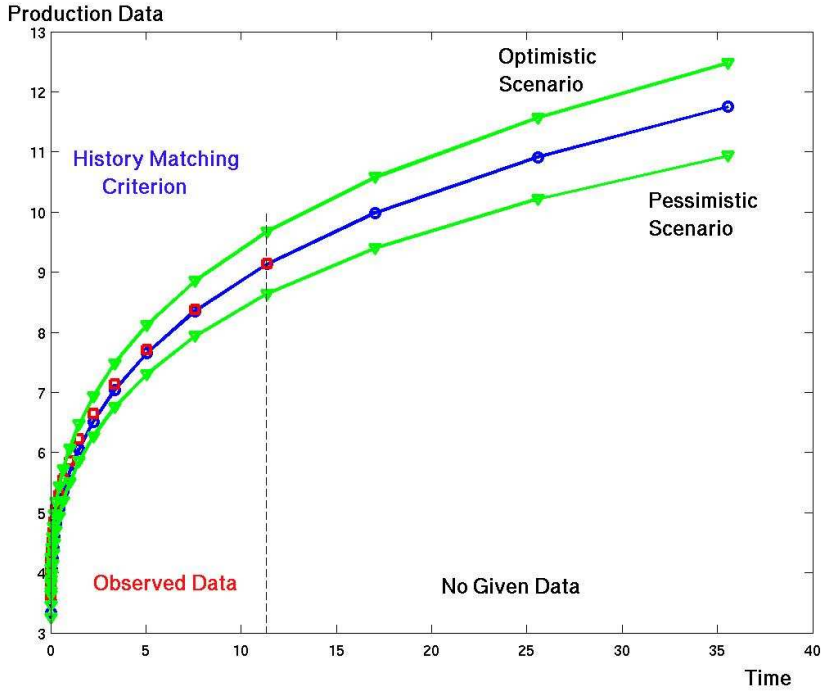


FIGURE 4.23 – Etape 2 du problème de prévision : scénarios optimiste / pessimiste (lignes vertes) estimant les bornes supérieures et inférieures de la prévision

Les résultats obtenus pour le problème de prévision par deux méthodes d'optimisation sont présentés et comparés. Les deux méthodes testées sur ce problème sont : la méthode SQA (voir chapitre 3) et une méthode classique de SQP utilisée avec une approximation BFGS de la Hessienne du lagrangien (annexe B). Le SQP nécessite le calcul des dérivées (par rapport à chacun des paramètres $\frac{\partial}{\partial x^i}$) de la fonction objectif et des données simulées, ces dérivées sont approchées par différences finies (méthode SQPAL, voir [118]).

Le cas PUNQ est un modèle réservoir 3D synthétique dérivé de données d'un champ réel (cf. la partie 2.1 de ce chapitre et [118]). La structure supérieure du réservoir est présentée dans la figure 2 de la section 2.1.1 de ce chapitre.

Les données disponibles sont : le cumul de production d'huile, le rapport gaz-huile et la quantité d'eau au puits pour 6 puits producteurs observés pendant 10 ans. Pour ces données synthétiques, 4 % de bruit relatif gaussien a été ajouté. Les paramètres inconnus sont des paramètres géologiques (i.e. perméabilité) et des paramètres d'écoulement. Soit 7 paramètres et 228 données mesurées. L'objectif est de déterminer ces inconnues et de prévoir la production d'huile cumulée de l'ensemble réservoir après 22 ans.

Les résultats de calage historique sont présentés sur la figure 4.24. La méthode Gauss-Newton de l'optimiseur SQPAL nécessite 75 simulations pour converger. La solution optimale x_{HM} est atteinte après 59 simulations et la valeur de la fonction coût est

$f(x_{HM}) = 2.679$. L'important ici n'est pas la méthode utilisée mais plutôt de partir du même point initial pour chaque méthode lors de la seconde étape de prévision. De plus, le nombre de données du problème n'est pas important, ce qui permet l'utilisation de la méthode Gauss-Newton (pas de problème de stockage de la matrice jacobienne). La figure 4.24 montre l'adéquation entre les données mesurées et calculées de production d'huile cumulée (dans la partie "history matching").

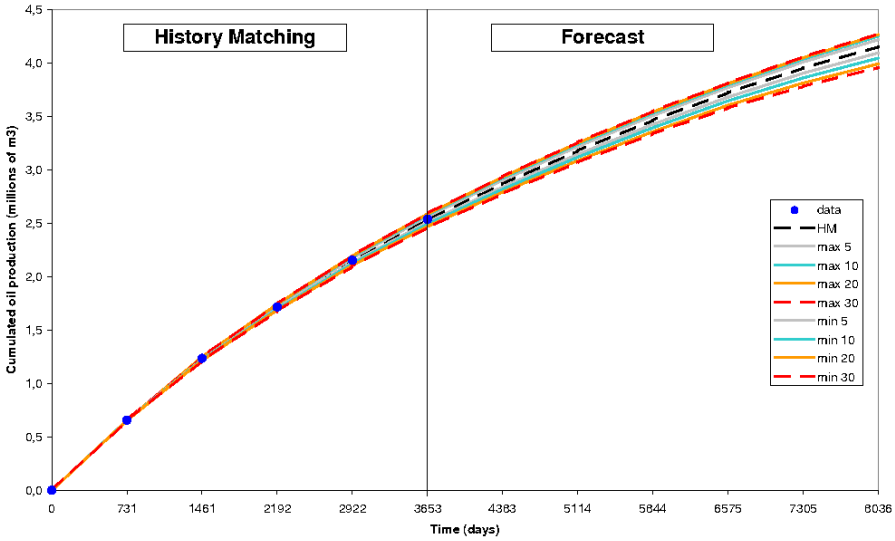


FIGURE 4.24 – Production cumulée d'huile : les points bleus représentent les données observées pendant 10 ans, la ligne pointillée noire représente la prévision de production calculée à partir de la solution du calage historique, les autres lignes montrent max / min de la prévision de production pour différentes valeurs de $\varepsilon = 5, 10, 20, 30$ (précision du calage historique)

La prévision de production cumulée d'huile et ses bornes supérieures et inférieures par rapport à différents $\varepsilon = \{5, 10, 20, 30\}$ peuvent être observées sur la figure 4.24. Des résultats identiques sont présentés sur la figure 4.25, l'axe des ordonnées représentant cette fois l'erreur de prévision (calcul de la différence de courbure avec la courbe de prévision de l'étape de calage historique). Sur ces deux figures, on peut observer que plus ε est grand, plus les bornes sur la prévision de la production cumulée d'huile sont éloignées. Ce comportement est logique car lorsque ε augmente, la taille de l'ensemble des modèles admissibles pour les contraintes linéaires (4.6) augmente également. Et minimiser ou maximiser sur un plus large ensemble peut seulement mener à augmenter (ou égaler) les valeurs maximales et à réduire (ou égaler) les valeurs minimales. Pour $\varepsilon = 30$, l'erreur sur la prévision de production cumulée d'huile après 22 ans de production appartient à

l'intervalle $[-0,2; 0.1]$ en millions de m^3. On observe que la distribution a posteriori de la production pétrolière cumulée n'est pas symétrique (due à la non linéarité des problèmes d'inversion et de prévision). Les scénarios extrêmes obtenus avec la contrainte non linéaire de calage historique ($\varepsilon = \infty$) sont affichés sur la figure 4.26. Dans ce cas, l'intervalle de confiance devient presque deux fois plus grand que celui obtenu avec $\varepsilon = 30$ (de 0.3 à 0.6 millions de m^3). Cela signifie que la définition du seuil de la contrainte non linéaire joue un rôle crucial dans la phase de prédiction.

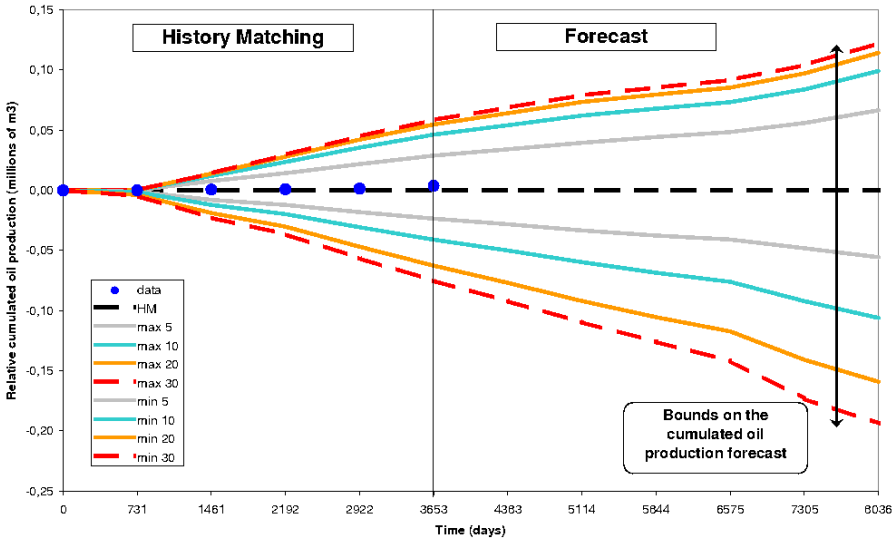


FIGURE 4.25 – Erreur relative sur la prévision de production cumulée d'huile

Des résultats équivalents ont été obtenus par les deux méthodes d'optimisation : la méthode SQA ainsi que la méthode classique SQP avec approximation des gradients par différences finies (avec une technique de globalisation par recherche linéaire) utilisée avec une approximation BFGS de la Hessienne du lagrangien. Dans les deux tableaux 4.2-4.3, les deux méthodes recommandées pour résoudre le problème de prévision (4.5-4.6) sont comparées. La première colonne des tableaux donne les valeurs de ε choisies. Les seconde et troisième colonnes présentent le nombre de simulations nécessaire pour converger respectivement pour SQPAL et SQA. Les deux colonnes suivantes donnent les valeurs de la fonction objectif, i.e. la production cumulée d'huile finale au temps T_2 pour les deux solveurs. Finalement, dans la dernière colonne, nous calculons la distance relative entre les modèles de réservoir obtenus par SQPAL et SQA (cela consiste à calculer la norme l_2 de la différence entre les deux modèles de réservoir divisée par la norme l_2 de x_{HM}). On peut remarquer que les résultats obtenus par les deux optimiseurs sont similaires aussi bien en termes de valeur de fonction objectif qu'en termes de distance relative entre les modèles

de réservoir obtenus par les deux solveurs. On peut remarquer que SQA est capable de résoudre ce problème 1.5 fois plus rapidement que SQPAL (le temps CPU nécessaire pour obtenir une solution du problème de prévision est proportionnel au nombre de simulations nécessaire à la convergence).

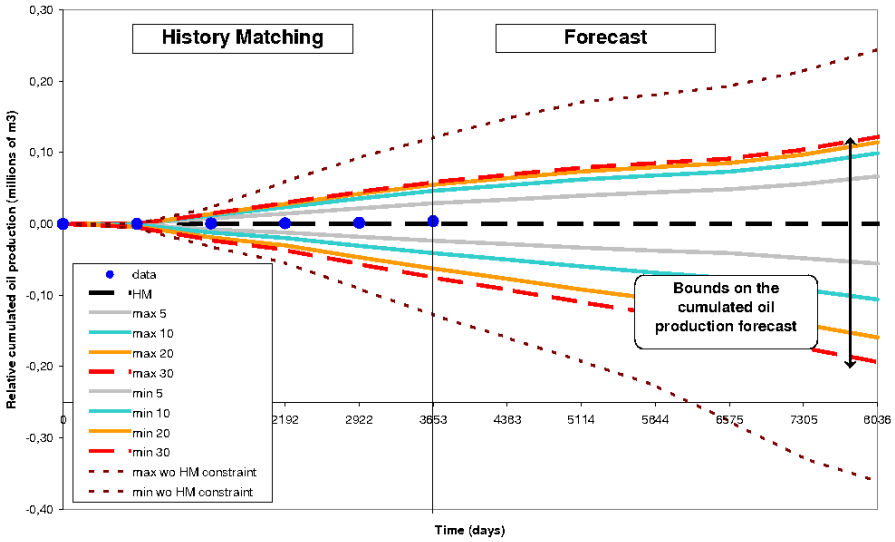


FIGURE 4.26 – Cette figure est similaire à la figure 4.25 excepté pour les lignes pointillées marron qui correspondent à la prévision de la production sans contrainte d'inégalité non linéaire max / min

TABLE 4.2 – SQPAL vs. SQA sur le problème de maximisation (4.5)–(4.6)

Cas d'étude	# Simulations SQPAL	# Simulations SQA	Valeurs Fonc. Obj. de la prévision SQPAL	Valeurs Fonc. Obj. de la prévision SQA	$\frac{\lVert x_{SQPAL}-x_{SQA}\rVert_2}{\lVert x_{HM}\rVert_2}$
"*max*, $\varepsilon = 5$"	126	51	4.22009	4.22255	0.009232
"*max*, $\varepsilon = 10$"	100	72	4.25268	4.25299	0.014125
"*max*, $\varepsilon = 20$"	102	74	4.26783	4.27149	0.01854
"*max*, $\varepsilon = 30$"	120	86	4.27568	4.2748	0.029325
"*max*, $\varepsilon = +\infty$"	81	78	4.39805	4.45779	0.228918

Table 4.3 – SQPAL vs. SQA sur le problème de minimisation (4.5)–(4.6)

Cas d'étude	# Simulations SQPAL	# Simulations SQA	Valeurs Fonc. Obj. de la prévision SQPAL	Valeurs Fonc. Obj. de la prévision SQA	$\frac{\left\|x_{SQPAL}-x_{SQA}\right\|_2}{\left\|x_{HM}\right\|_2}$
"min, $\varepsilon = 5$*"*	81	65	4.09791	4.09856	0.020384
"min, $\varepsilon = 10$*"*	94	77	4.04742	4.04545	0.026749
"min, $\varepsilon = 20$*"*	149	65	3.9944	3.99438	0.033698
"min, $\varepsilon = 30$*"*	65	70	3.95969	3.9663	0.011092
"min, $\varepsilon = +\infty$*"*	162	63	3.7925	3.78991	0.066514

L'approche proposée pour l'évaluation des incertitudes sur la prédiction du cumul de production d'huile est une méthodologie innovante. Nous soulignons que bien que le problème de calage historique est une minimisation de fonction de moindres carrés sous contraintes de bornes, le problème de prévision est plus complexe. Ce n'est plus un simple problème de moindres carrés, et une contrainte d'inégalité non linéaire pour laquelle les dérivées ne sont pas disponibles est introduite. La méthode SQA a donné des résultats prometteurs sur ce problème réservoir 3D présenté : les résultats obtenus sont similaires à ceux obtenus avec SQPAL. Toutefois, SQA est 1.5 fois plus rapide que SQPAL.

Il pourrait être intéressant de comparer ces résultats à ceux obtenus par une approche probabiliste utilisée habituellement par les ingénieurs de réservoir.

2.2 Application en calibration des moteurs

Nous présentons dans cette section les résultats en calibration moteur d'un modèle physique couplé à un modèle statistique. Le contenu de cette section est basé sur l'article [67]. Une autre étude en calibration des moteurs est présentée dans le chapitre 7 avec l'application de méthode d'optimisation en multi-objectifs.

Derivative free optimization method and physical simulations coupled with statistical models for transient engine calibration

Grégory Font, Delphine Sinoquet, Hoël Langouët, Michel Castagné and Sébastien Magand

IFP Energies nouvelles, 1&4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex France

Abstract

Nowadays, automotive manufacturers have to deal with strong constraints: lowest fuel consumption, emission-control legislation and driver requests for driveability and performance. For this purpose, the classical engine has evolved towards a very complex system combining many hi-tech components with advanced control strategies. It leads to an increasingly complexity of the underlying optimization issue.
The classical method to perform this optimization is to use statistical response surface models obtained from optimally designed experiments at test bench in steady-state conditions. In this approach, the transient effects observed during the driving cycle are not taken into account.
A promising idea to include these transient effects in the optimization process is to couple physical modelling of the engine (to get the conditions in the combustion chamber before each combustion event) with statistical models (to obtain engine-out emissions after this combustion event). Powertrain system simulation is shown to provide a helpful tool to reach this objective thanks to a good compromise between accuracy and low CPU time.
A dedicated parameterization of the engine maps and a Derivative Free Optimization method is then used to minimize the cumulative fuel consumption and pollutant emissions, under combustion noise constraints, on a driving cycle.
An application on a real dataset obtained at automated test-bench for a diesel engine is presented.

1. Introduction

Future regulations enforcement concerning emissions, customers requirements and economic competition lead to higher constraints on internal combustion engines development.
It paves the way for increasing complexity in terms of design and control. Diesel engines, with many sophisticated devices for air loop as well as for injection system, have now potentially thousands of control parameters to be calibrated. Fast and reliable tuning of these parameters are key issues for engine manufacturers.
The first stage of the calibration process consists in basic engine maps tuning for steady-state conditions. The conventional method for engine map optimization is based on statistical engine models depending on control parameters obtained from optimally designed experiments at test bench in steady-state conditions. The cumulative emissions on the driving cycle computed with these models are optimized ([1], [2]). In this approach, one assumes a quasi static engine behaviour on engine speed

and load trajectory, indeed the transient effects during the driving cycle are not taken into account.

Engine physical model coupled with statistical and control models is a promising solution to deal with the transient effects optimisation. Eventually the real air and fuel trajectories during the driving cycle could be used as inputs to optimise the control parameters to fulfil noise and emissions targets. Powertrain system simulation, with phenomenological 0D combustion and air loop models, appears as a good compromise between accuracy and low CPU time for engine physical modelling. One of the main challenges concerns the coupling process in order to limit CPU time and loss of accuracy.
Usual optimisation algorithms require derivatives of optimised functions which are not available for such models. Moreover, the large size of the optimization problem (number of parameters for engine map description) prevents from approximating these derivatives by perturbation techniques (finite differences). Hence a derivative free optimization method whose efficiency has been highlighted in several application areas ([3]), is proposed to tackle this topic.
Thus this paper focuses on the calibration process (design of experiments, statistical modelling of engine responses), engine physical models, to couple those models and to design of adequate optimization tools .

2. Experimental data and statistical modelling

2.1. Description of the test case

The engine used during this study is a 1.6 litre four-cylinder Diesel engine (DV6 ATED 4 from PSA Peugeot-Citroën). This engine has a maximum power output of 66 kW. The bore and the stroke of this engine are respectively of 75 mm and 88.3 mm. The compression ratio is 18:1. It is a turbocharged engine with a high pressure cooled exhaust gas recycling circuit. It has a second generation injection system using Bosch CRI 2.2 injectors with injection pressure as high as 1600 bar. The exhaust line has a conventional Euro4 Diesel oxidation catalyst (DOC) and is not equipped with a Diesel particulate filter (DPF).

Moreover, this study is part of a project concerning the use of biofuels to deal with the issues of energy diversification and greenhouse gas emissions. Therefore an Ethanol based Diesel fuel was chosen; the fuel formulation is deeply described in the reference [4]. The vehicle and the powertrain were not modified to use this blend but the engine calibration was updated as explained in previous publications [4], [5].

The most significant point concerning the facilities is the use of a dynamic test bench driven by Morphée® for the reproduction of normalised European driving cycle (NEDC). Such equipment and procedure have been detailed in a previous paper [4]. The test cell provides all the equipment required for transient tests (fuel consumption, opacimeter for smoke measurement) and the design of the fluid circuits has been optimised to ensure good vehicle behaviour reproduction.

2.2. Experimental data

Different approaches to tune engine maps are proposed to deal with the high number of parameters to optimise and to reduce this time-consuming task during the development process. These methods (local, mixed and global) based on mathematical models and Design of Experiments (DoE) were compared in previous publication [6]. Global modelling approach for calibration, considering load and speed as parameters, reduces time spent for whole calibration workflow because an unique DoE is necessary to model engine responses for entire domain (or several DoE if this domain is divided in several zones with common parameters) [7]. In fact the global method considers the trace of the driving cycle (load and speed) as inputs of the models of the engine responses. Figure 1 shows the operating area chosen for the calibration in the speed/load area.

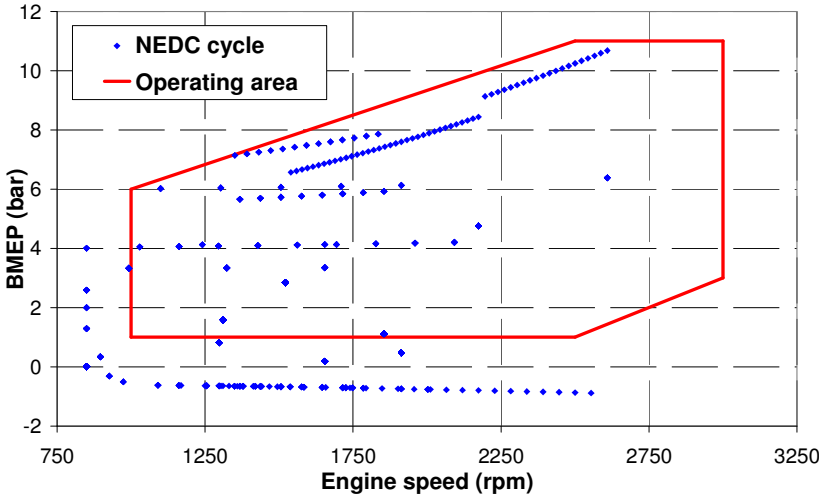


Figure 1: Engine speed and load area for global method

The next stage consists in defining the operating space of each engine control parameter. Concerning injection parameters, previous experiences [7] show that fixed boundaries or simple linear limits compared to the original maps could be a suitable answer to ensure a sufficient potential of optimization.
The main issue involves the air loops limits. It is required to have a wide enough operating space close to the physical domain (to ensure a high potential of optimization) and to make it easy to model. Four air limit parameters should be defined: two for the boost pressure (maximum and minimum limits) as well as two for the air mass flow. Three limits are physical ones and could be straightforwardly obtained. To determine the last one (minimum air mass flow), different criteria could be chosen (air fuel ratio, stability, etc.). Then each limit should be modelled, in this case with low degree polynomial. The accuracy of the model is illustrated in Figure 2 for the minimum air flow limit by comparing experiences used for the modelling (blue crosses) and the validation points (red circles). The coefficient of determination R^2 as well as the root mean squared error (RMSE) show the good estimation obtained with the model.

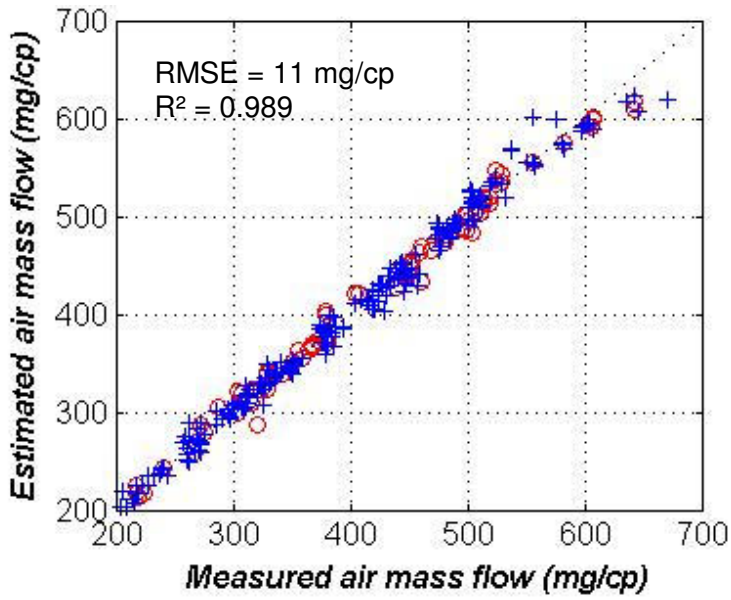


Figure 2: Comparison of estimated and measured minimum air mass flow limit. Blue crosses correspond to experimental data used for modelling whereas red circles are validation points.

Some of the main challenges when using the global modelling approach concern complex constraints for description of parameter variation domain (e.g. non-linear) and increased number of parameters to manage. Thus dedicated statistical tool and method should be used to take into account these issues. This process was described in previous reference [5]. After defining the speed and load boundaries as explained previously, the values of these parameters are fixed; this permits to determine acceptable values for injection parameters and then air loop limits depending on other parameters could be given. The most valuable experience is determined by using a *Maximin* criterion (maximization of minimal distance between experimental points). Finally, the global DoE includes also the corners of the operating space, some mono parametric variations and a space filling.

2.3. Global modelling of the engine responses

Modelling the complex engine response surfaces requires to use advanced statistical models instead of standard polynomial models. In this case the choice of kriging for modelling was the most relevant since it is flexible and takes into account the experimental repeatability and the spatial dependency providing therefore consistent values for close operating points. A total of 1200 experiences, added to 100 points for repeatability and 200 points for validation of the models, were performed.
The engine responses that are modelled are : HC, NOx, CO, CO_2 and PM emissions (g/h), combustion noise (dB) and fuel consumption (kg/h). The statistical criteria are satisfying for all models except HC and PM emissions. Concerning PM emissions, the oxygenated fuel used for these tests implies near-zero soot emissions and there is a lack of accuracy of the measurement device for these levels. Concerning HC emissions the bad repeatability is the main explanation of the poor quality of the model. Some examples of validation of the estimated value from the model are given in Figure 3 and in Figure 4. First NOx emissions model is validated with experimental specific tests. CO_2 emissions estimation during monoparametric variations are repre-

sentative of test bench results in Figure 4. In fact, for the different emissions, the models show good trends and consistent estimated levels.

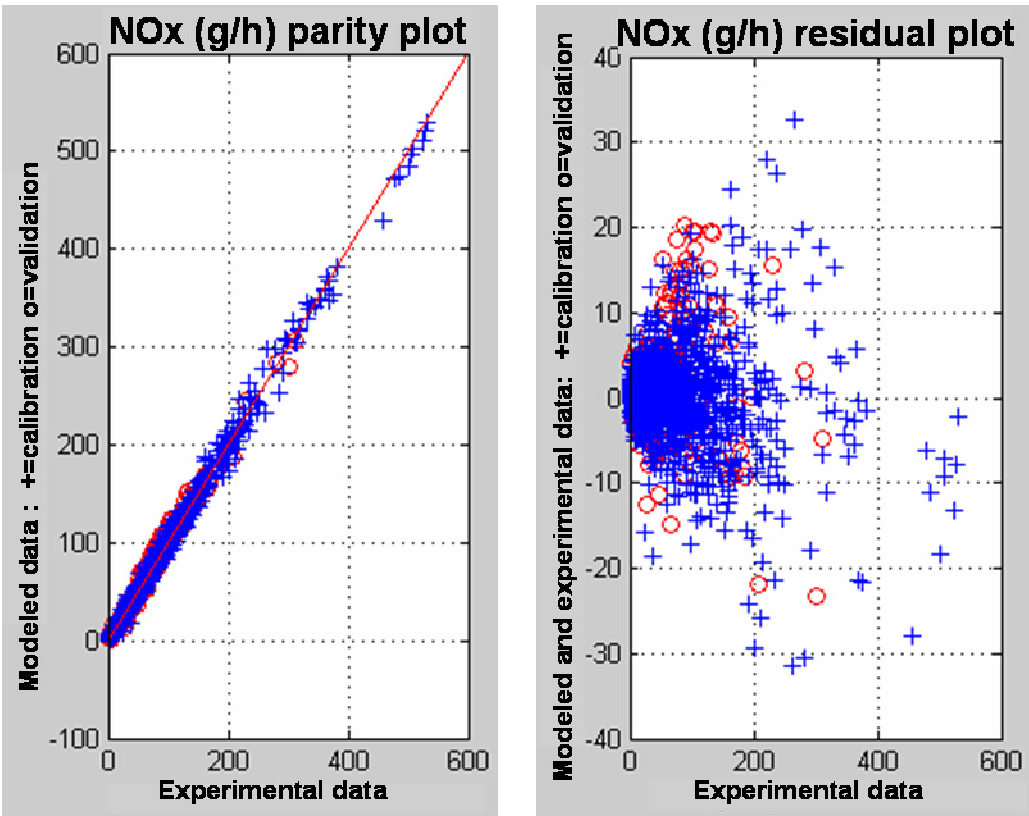


Figure 3: NOx emissions model validation. Blue crosses correspond to experimental data used for modelling whereas red circles are validation points.

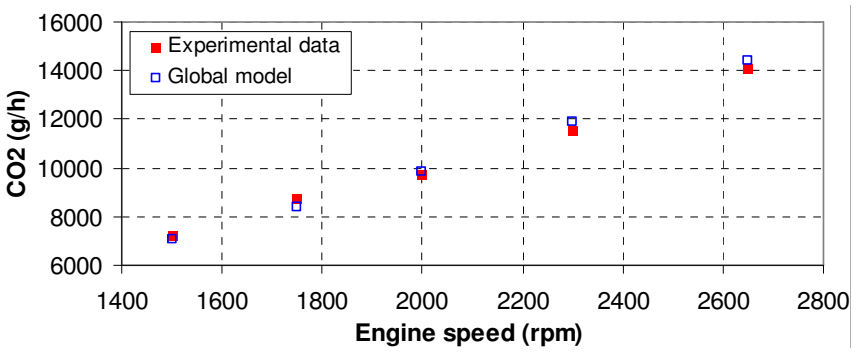


Figure 4: Comparison of estimated and measured CO_2 emissions on an engine speed variation

3. Coupling physical engine simulator with statistical models

IFPEN is strongly involved in powertrain modelling and simulation to support technological development. It develops specific libraries (IFP-Engine, IFP-Drive and IFP-Exhaust) in the LMS.IMAGINE.Lab AMESim environment which includes a large set of commercial libraries used for various applications, in particular for control strategy development, validation and pre-calibration as it is presented in [8].

The purpose of this section is the coupling of a physical engine simulator with statistical models and control algorithm model in order to have both well representative transient effects and accurate pollutants model. The complete engine simulator is based on the LMS.IMAGINE.Lab AMESim platform with in particular the use of the IFP-Engine library. Its description and validation are given in the following sections.

3.1. Complete engine simulator creation and validation in steady state

1. Combustion model

IFP-Barba phenomenological combustion model

In this study, a combustion model based on the Barba approach [9][10][11] is chosen to compute the combustion heat release rate. In this approach, the combustion process is divided into two parts. In a first step, the fuel is burnt using a pre-mixed model with the hypothesis of flame propagation in the pre-mixed zone. When the pre-mixed zone is burnt, the remaining fuel is oxidized using a mixing controlled combustion model.
Using a simple example with a pilot and a main injection, different steps can be isolated during the combustion. At the beginning of the pilot injection, the combustion model is initialised and the different variables are computed. During injection, the auto-ignition delay is computed and the injected fuel is introduced in the premixed zone.
The auto-ignition delay is defined at the beginning of the injection and is computed using a simple Arrhenius law. When the auto-ignition delay is reached at the end of the phase, the pre-mixed combustion starts. In a first step, the ascendant part of the combustion heat release rate (burning mode 1) is generated by the flame propagation in a turbulent field. Then, because of the multiplication of the auto-ignition sites, the hypothesis of the propagation of a single flame in the pre-mixed zone is no longer valid and the flame interactions must be modelled (burning mode 2), leading to a reduction of the total flame surface.
When the main injection starts, the model creates a new pre-mixed zone for the main injection and computes the auto-ignition delay. The definition of the pre-mixed zone is similar to that of the pilot injection.
As for the first injection, the auto-ignition delay of the main injection is computed from the beginning of the injection, leading to the beginning of the combustion. The auto-ignition model is the same as for the pilot injection, but thanks to a higher pressure and temperature, the auto-ignition delay is smaller and reached before the end of the injection.
As the auto-ignition delay is reached before the end of the injection, the fraction of fuel available in the premixed zone is lower than for the pilot injection. The combustion process is the same as for the first injection.
In parallel to the pre-mixed combustion mode, the remaining injected fuel starts to burn progressively with a mixing controlled combustion model. In this burning mode, the combustion is piloted by the mixture speed between the injected fuel and the surrounding air. Generally, the mass of fuel burnt in this mode is defined with a simple equation depending on the turbulent kinetic energy. In order to enlarge the range of applications of this combustion model, there is the possibility to compute more injections and combustions.

Combustion model set up

A single cylinder simulator with closed valves is used to calibrate the combustion chamber. The calibration is made between Intake Valves Closure (IVC) to Exhaust Valve Opening (EVO). The initial in-cylinder thermodynamic conditions are defined using the experimental results for the steady state points. The advantage of this approach is to lower the error generated by the air path on the in-cylinder conditions at the IVC.

Concerning the thermal transfers, a Woschni model is used [12]. In order to fit correctly the combustion, the wall temperatures are functions of the BMEP with the following law :

$$T_{wall}(K) = \frac{100}{19} * BMEP(bar) + 450 - \frac{100}{19} \quad (1)$$

To set up correctly this combustion model, 9 points are chosen among the 218 available in the engine map (cf. Figure 5 – red surrounded crosses). The calibration process consists in fitting the simulated cylinder pressure on the experimental one for each operating condition. Then a validation on the whole operating points is done. Only one set of parameters is used for all the engine map.

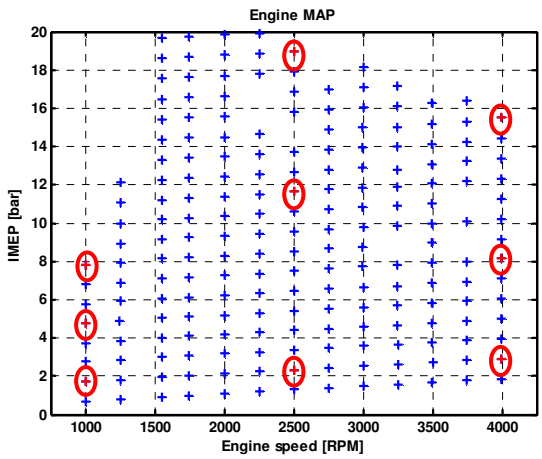


Figure 5: Engine map. 9 points (red circles) are chosen among the 218 available to calibrate the combustion model

The cylinder pressure curves shown in this paper (cf. Figure 6) are three out of the nine points used for the calibration of the combustion model (cf. Figure 5 – red surrounded crosses) . The four experimental cylinder pressures are in blue and the mean of the four is in dashed black. The simulated cylinder pressure is in red. In this paper, all the comparisons are made between the mean experimental cylinder pressure and the simulated one.

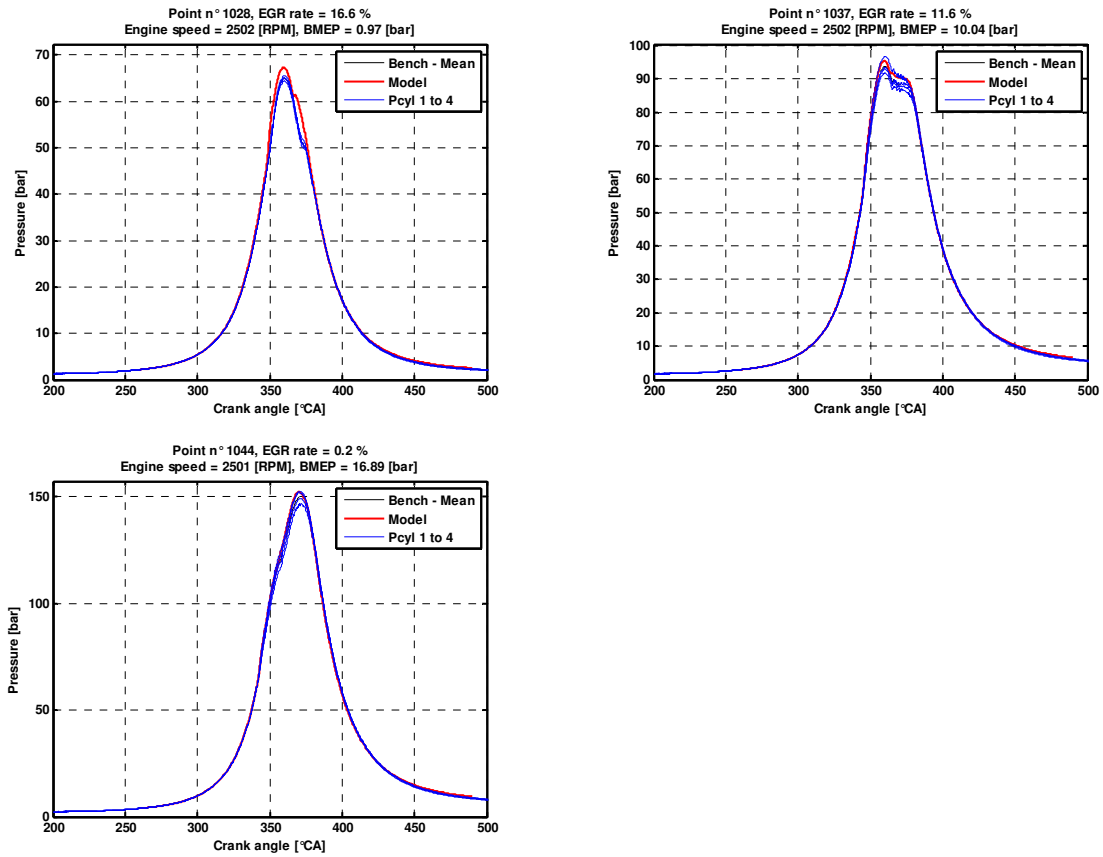


Figure 6 : Cylinder pressure – 2500 RPM – BMEP = 1, 10 and 16.9 bar. The four experimental cylinder pressures are in blue and the mean of the four is in dashed black. The simulated cylinder pressure is in red.

In order to define the combustion model accuracy, five criteria are chosen :

- IMEP : +/- 1 bar
- Max pressure : +/- 4 bar
- Max pressure angle : +/- 2 CA
- ISFC : +/- 5 %
- Noise : +/- 2 dB

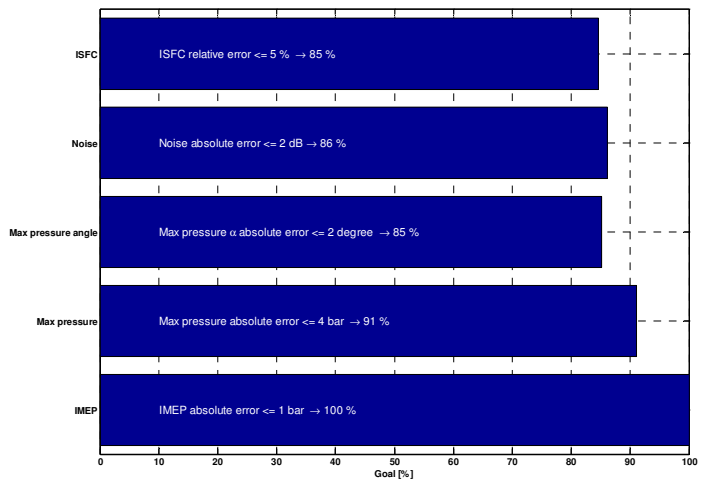


Figure 7 : Percentage of points matching the criteria to check combustion model accuracy

The cylinder pressures of the model are visually closed to the cylinder pressures of the bench. Moreover as described in the Figure 7 the percentage of points matching the criteria are high. It must be reminded that these results are obtained with only one set of parameters for the whole engine map.

2. Steady state engine model

Figure 8 presents the complete engine simulator. The whole air path with the turbo-charger, the charge air cooler, the exhaust line and the EGR loop is modelled. After its tuning the combustion model is embedded in this complete engine simulator.

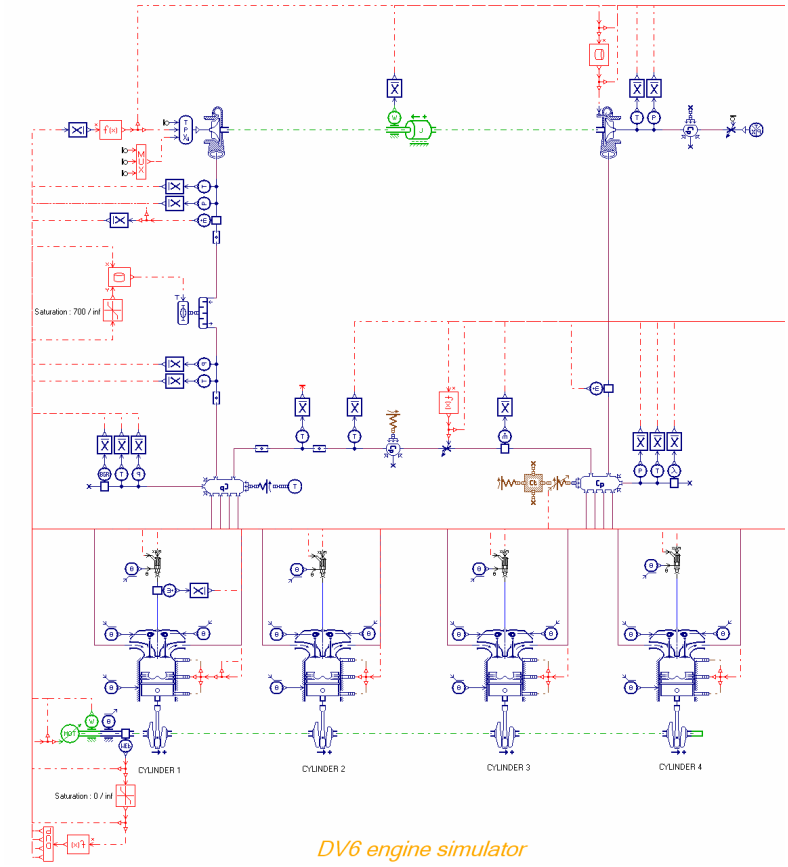


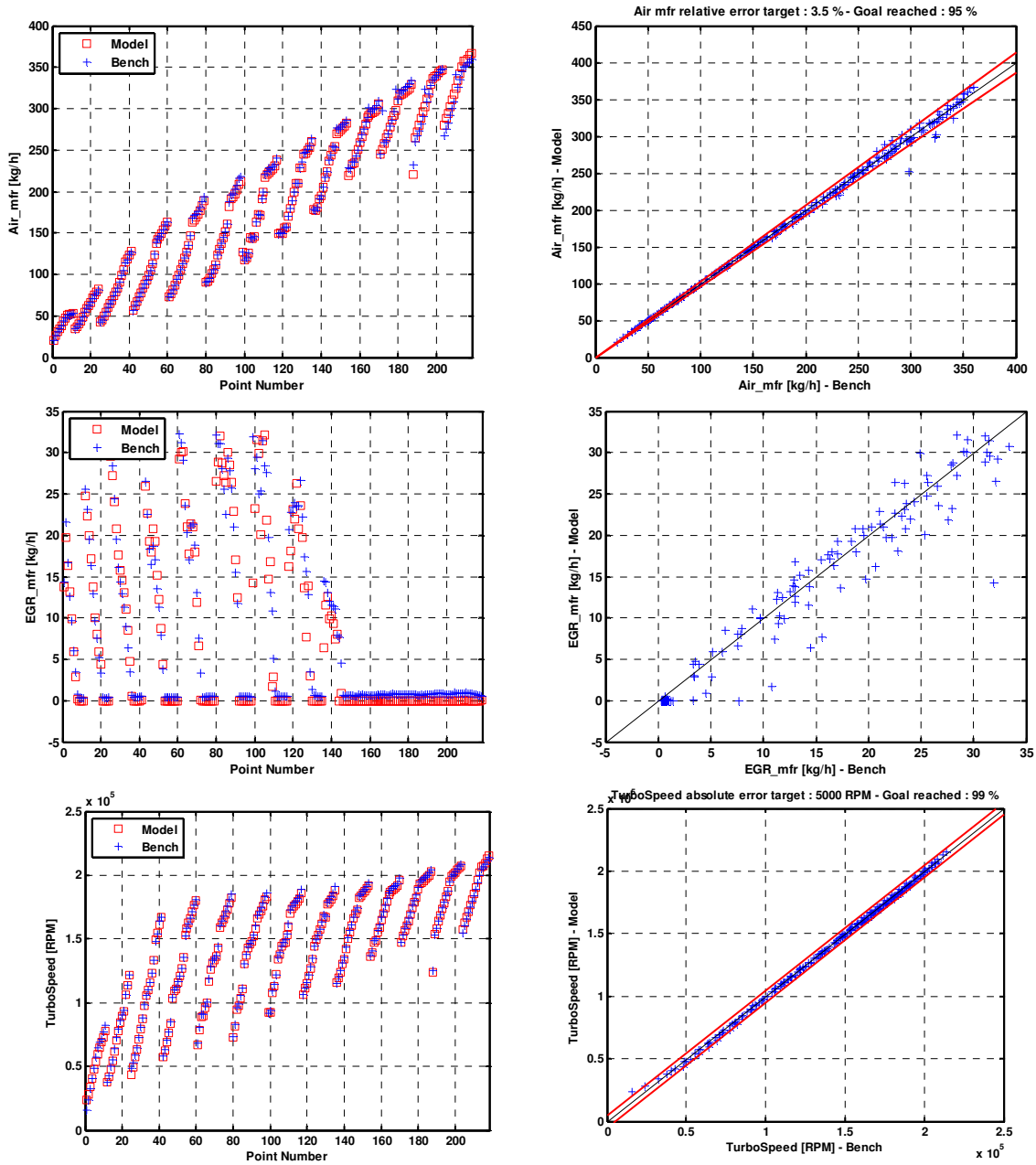
Figure 8 : Four-cylinder simulator in AMESim environment

In order to validate the behaviour of the complete engine simulator, it is coupled with a simple control implemented in Matlab Simulink. Two PID controllers regulate the engine : one for the intake manifold pressure with the control of the VNT position and one for the air mass flow rate with the control of the EGR valve position.

In order to appreciate the quality of the results, some criteria are defined :

- Intake manifold pressure (P2S) : +/- 0.05 bar
- Intake manifold temperature (T2S) : +/- 10 K
- Exhaust manifold pressure (P3) : +/- 0.2 bar
- Exhaust manifold temperature (T3) : +/- 30 K
- Air mass flow rate : +/- 3.5 %
- IMEP: +/- 1 bar

The Figure 9 presents results obtained with the engine simulator on the 218 measurement points. The presented results are air mass flow rate, EGR mass flow rate, turbo speed and IMEP. On left figures, blue crosses represent experimental data and red squares model results for each operating point. On right figures, model results function of experimental data are plotted. Red lines represent criteria to respect.



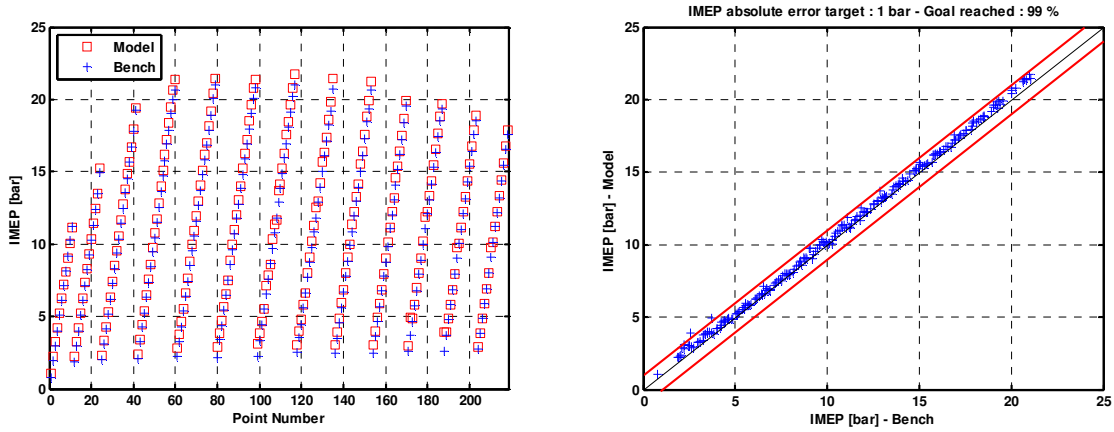


Figure 9: Steady state engine simulator results. Left figures : blue crosses represent experimental data and red squares modelled data results for each operating point. Right figures : red lines represent criteria to respect.

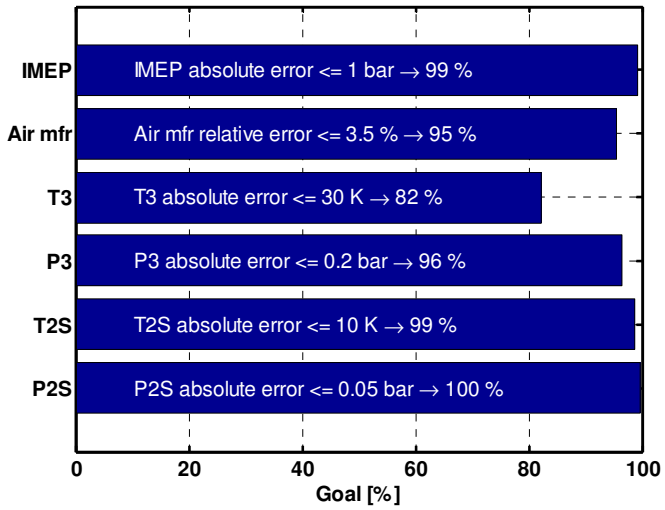


Figure 10: Percentage of points matching the criteria to check the complete engine model accuracy

Figure 10 gives the percentage of the points which respect the criteria on IMEP, air mass flow rate and intake and exhaust manifold temperatures and pressures. According to these criteria, results are consistent with test bench data and the complete engine simulator can be considered as set up.

3.2. Coupling with statistical models : presentation and validation on NEDC cycle

Figure 11 presents a way to estimate pollutant emissions during a transient thanks to a coupling between the AMESim simulator described in previously section, the engine control and the statistical models described in section 2.3. Inputs of these statistical models are air mass flow rate, intake manifold pressure, start of main injection, rail pressure, injected mass in the pilot injection, gap between pilot and main injection, BMEP and engine speed. The coupling is made with Matlab Simulink.

In this study, the real control software was not available. So, a basic open-loop control based on maps is used. Thus maps function of engine speed and total injected mass are used for the pilot and main start of injection, injected mass for the pilot injection, rail pressure, VNT and EGR valve positions.

A NEDC cycle was performed on this platform. As the optimization described in the next section is run on the EUDC part, only the results concerning EUDC are shown. Figure 12 presents comparison between virtual engine (in red line) and experimental data (in blue dashed line) for intake and exhaust manifold pressures (P2S and P3), air mass flow rate (Air_mfr), fuel air equivalence ratio (FAR_sonde), turbo speed (TurboSpeed), BMEP and CO_2, CO, NO_x and HC mass flow rates (CO2GH, COGH, NOxGH and HCGH). A comparison on cumulative emissions (NOx, CO_2, CO and HC) is also presented in Table 1.

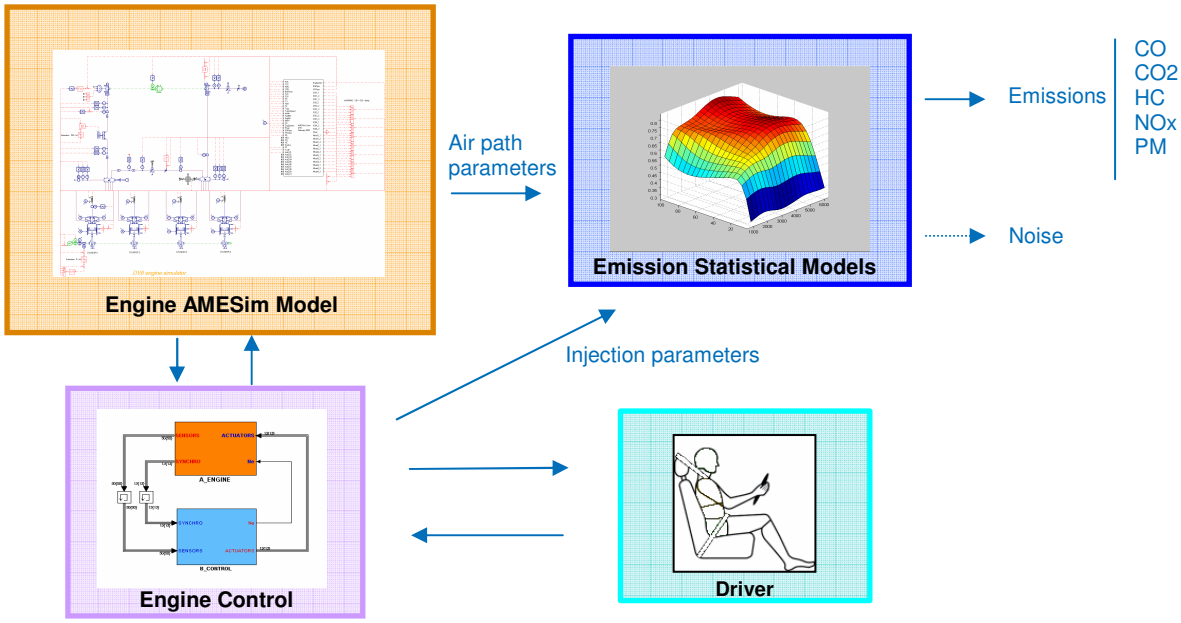
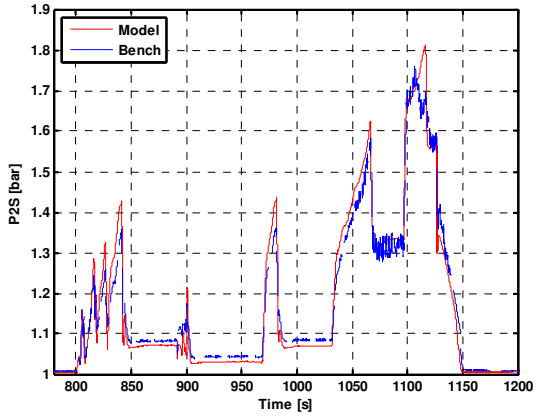
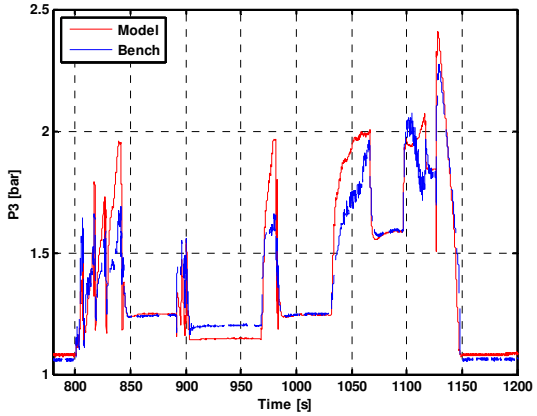
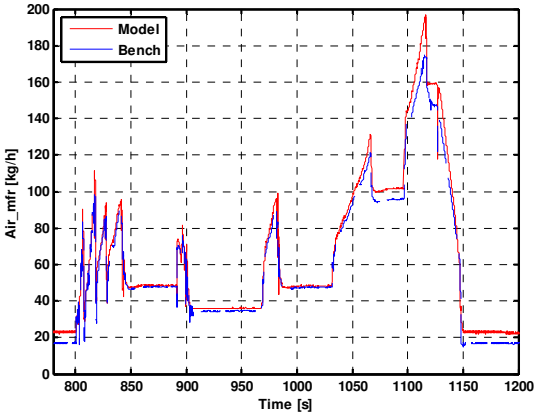
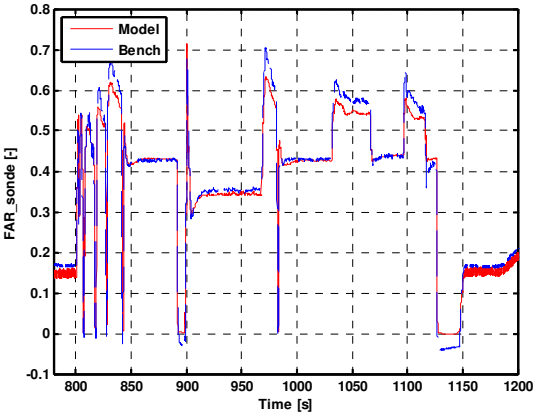


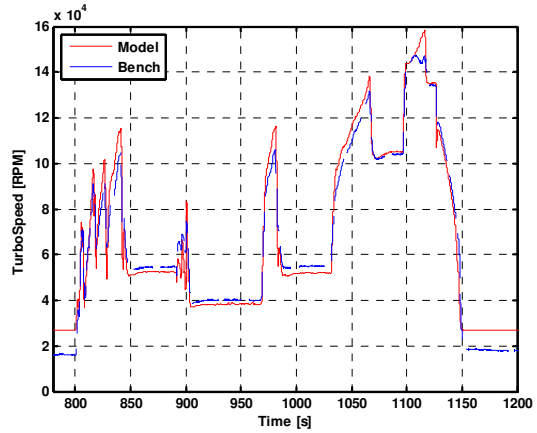
Figure 11 : Description of the coupling between engine Control, engine AMESim model and emission statistical models

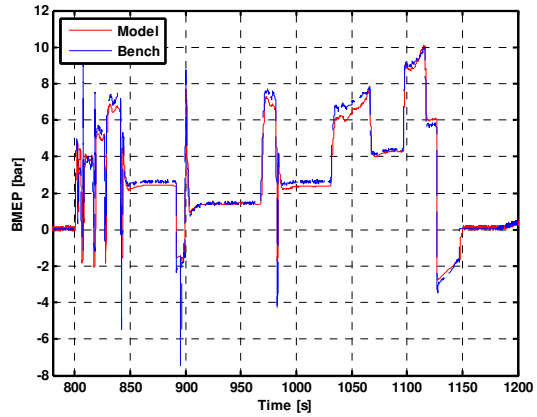

Model
Bench
P2S [bar]
Time [s]


Model
Bench
P3 [bar]
Time [s]


Model
Bench
Air_mfr [kg/h]
Time [s]


Model
Bench
FAR_sonde [-]
Time [s]


x 10^4
Model
Bench
TurboSpeed [RPM]
Time [s]


Model
Bench
BMEP [bar]
Time [s]

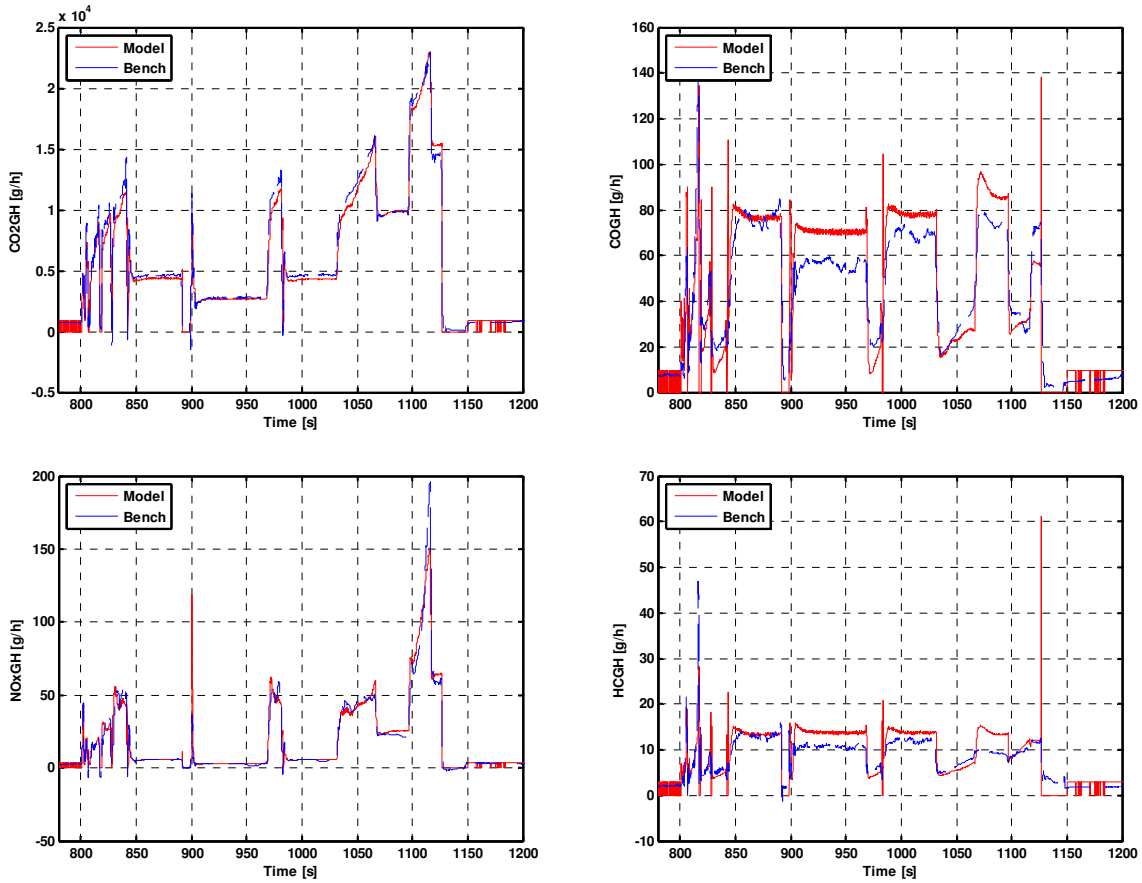


Figure 12 : Comparison between model (red line) and experimental (blue dashed line) data on EUDC

EUDC cycle	NOx emissions	CO2 emissions	CO emissions	HC emissions
Bench	100 %	100 %	100 %	100 %
Virtual engine	100 %	99 %	107 %	110 %

Table 1 Comparison of the cumulative emissions between model and experimental data on EUDC

Keeping in mind that engine control is different for experimental tests and simulation, one can conclude that the behaviour of the engine during the EUDC is well reproduced.

This virtual engine is sufficiently predictive to be used for an optimization process.

4. Engine map optimization with coupled models

4.1. Principle of engine map optimization

The engine map optimization consists in directly optimizing the cumulative pollutant emissions over the cycle via distortions of the engine maps instead of optimizing individually some selected operating points and building afterwards the engine maps by a smoothing step ([6], [13]).
Consequently, the objectives to be minimized (or constrained) are the engine responses cumulated on the considered driving cycle. The engine responses can be modelled by a global statistical model (as described in section 2) or by a physical simulator that models transient behaviour of the engine (as described in section 3).

The map optimization problem is formulated as

$$\begin{cases} \min\limits_{m^p \in R^{N_p}} \int_0^T F_i(r(t),c(t),m^{P_1}(r(t),c(t)),m^{P_2}(r(t),c(t)),\ldots,m^{P_{Np}}(r(t),c(t)))dt \\ subject\ to \\ l(r,c) \le Am^p(r,c) \le u(r,c) \\ \int_0^T F_j(r(t),c(t),m^{P_1}(r(t),c(t)),m^{P_2}(r(t),c(t)),\ldots,m^{P_{Np}}(r(t),c(t)))dt \le S_j \end{cases} \quad (2)$$

where :

- (*r(t); c(t)*) indicate the trace of the considered driving cycle within the engine speed-load domain,
- F_i is the global model of the engine response *i* depending on the engine control parameters but also on the speed and load,
- m^{Pi} are the 2D engine maps of the control parameters in the engine (speed, load) operating domain.

Additional smoothing constraints such as global smoothing constraints (to preserve the regularity of the original maps), robustness constraints (to take into account parameter dispersions) and local constraints (for example limits on the gradients of the maps, or limits on engine noise response) are also introduced.

This formulation requires an adapted parameterisation of the engine maps, m^{Pi}(r,c), in order to limit the total number of unknowns in the optimization process. For this application, an adapted discretization of the engine speed/engine load domain with a bilinear approximation in cells is implemented (see Figure 13):
$m^{P_i}(r,c) = \alpha_0 + \alpha_1 r + \alpha_2 c + \alpha_3 rc$ within a given cell with $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ determined from the 4 cell corners. Other map representations based on LoLiMoT (Local Linear Model Tree) are proposed in [13].

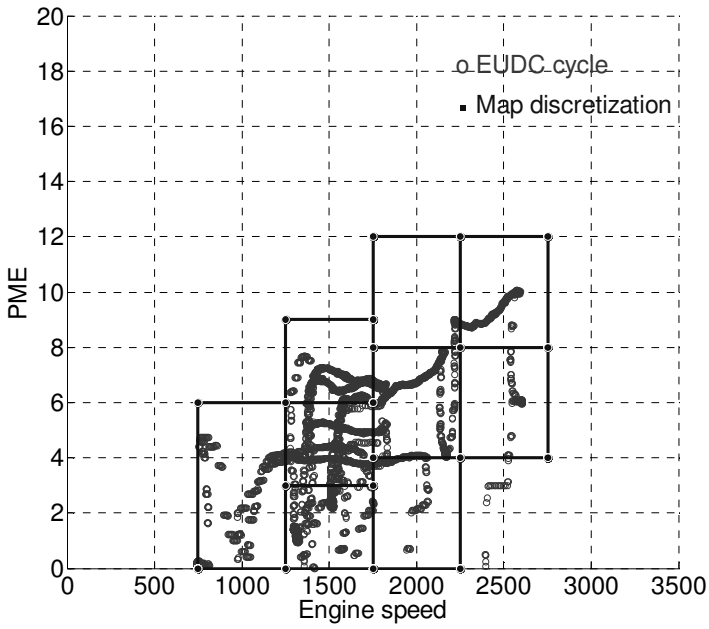


Figure 13 : Map discretization (for bilinear parameterization) in the engine (speed, load) operating domain and EUDC cycle trace.

4.2. Derivative Free Optimization (DFO)

Problem (2) is a nonlinear constrained optimization problem which requires dedicated optimization methods to be solved. For engine responses modelled by statistical models, a classical SQP (Sequential Quadratic Programming) method is applied: first derivatives of objective function and constraints are available.
When the physical model described in section 3 is used, a Derivative Free Optimization method is implemented. In this case, no derivative of modelled engine responses is available and moreover simulations require a high computational effort (a couple of hours of calculation for one simulation).
Therefore, a dedicated optimization method is proposed, the SQA method (Sequential Quadratic Approximation), already applied successfully on several industrial applications [3].
This method is an extension of the DFO method proposed by Powell in [14] to nonlinear constrained problems: in order to save simulations, quadratic interpolating models are used as surrogate of the simulator responses. These quadratic models are minimized and updated thanks to additional simulations performed along the optimization process iterations. The main steps of the SQA algorithm are described in Table 2.

SQA Algorithm

Step 0. INITIALIZATION A quadratic model (QM) of each optimized engine response is constructed in a neighbourhood of the initial guess, namely the trust region. They interpolate *m* points ($n+2 \leq m \leq (n+1)(n+2)/2$ with n the number of parameters), obtained by evaluation of the engine simulator.

Step 1. MINIMIZATION At each iteration, we substitute in problem (2) engine responses with their associated QM and solve it within the trust region (TR).

Step 2. EVALUATION and UPDATE of TR We evaluate the engine responses at the solution of previous problem and update QM with this new point. The size of TR is also updated according to the predictivity of QM at this point :

i. if predictions by QM are very closed to simulation results, TR is enlarged. Go to step 1.

ii. if only the trend is correct, TR is not modified. Go to step 1.

iii. else reduce TR and go to 3.

Step 3. IMPROVEMENT OF QM Choose and simulate a new point (evaluation of engine responses) in order to improve the accuracy of QM, update QM and go to 1.

Table 2 Main steps of SQA algorithm: a DFO (Derivative Free Optimization) method for nonlinear constrained optimization.

5. Application on EUDC driving cycle

First, a quasi-static optimization (QSO) is applied. It is based on cumulative engine responses on the EUDC cycle (1 Hz discretization) modelled by global statistical models presented in section 2.3 (conventional optimization method is used, derivatives of objective and constraint functions being available for surface response models).

Four engine maps controlling injection are optimized: main injection timing, fuel injection pressure, pilot fuel injection quantity, pilot injection timing whereas mass air flow and boost pressure maps remain fixed to initial values. Bilinear modelling of maps allows to limit the number of map parameters to be optimized: 80 parameters.

The minimized objective is the cumulative NOx emissions along the cycle, CO2 emissions are constrained to remain smaller or equal to the initial value, whereas an increase of 50% for HC and CO emissions is allowed.

The optimization process is then applied but this time with physical AMESim model coupled with pollutant statistical models described in section 3.2 with DFO optimization method presented in Table 2. In order to save simulation computational time, the smallest number of interpolation points for a quadratic model: $m=n+2$ is chosen for this application. This incomplete quadratic model is completed during optimization process with a constraint on model variations along iterations. Additional terms of the model are then determined. Therefore, as displayed in Figure 14, the initialization phase costs 82 simulations used to build the initial quadratic model. After this phase, the optimization work starts with a rapid decreasing of NOx emissions.

Results are presented in Table 3 and compared to results obtained with quasi-static optimization. The optimization coupled with the physical AMESim model, compared to the quasi-static optimization, enables to drop off the NOx emissions with a slight increase of HC and CO emissions. But they are still largely under the defined thresholds and even under initial maps results. CO_2 emissions remain close to the initial value.
Figure 15 compares engine emission traces on EUDC cycle for initial engine maps, QSO maps and maps obtained by optimization with coupled physical models. On 70-100km/h acceleration of EUDC cycle, one notices the gain of optimization considering transient effects compared to quasi-static optimization for NOx reduction. Nevertheless, globally, the NOx reduction is low due to the smoothness of EUDC cycle. Larger gains are expected on more realistic cycles.

The physical AMESim model allows to take into account the transient effects of the air loop. It is then possible to take care of boost pressure lag for example to optimize the maps and not only to work on steady-state conditions.

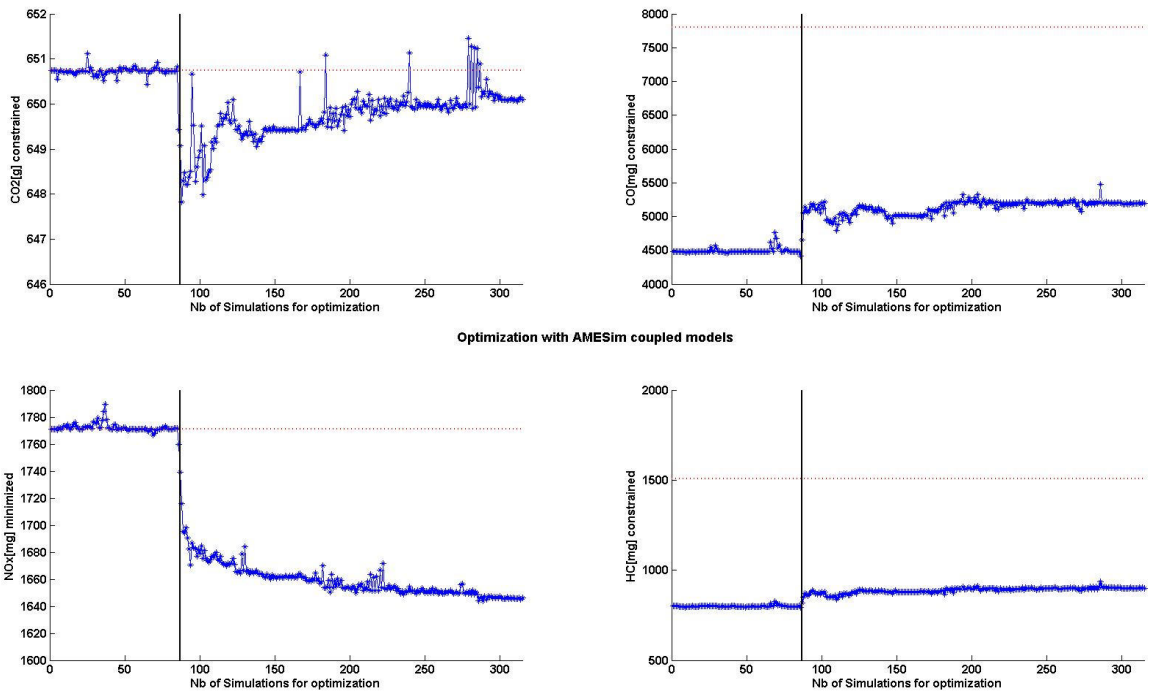


Figure 14 : Objective function decreasing and constraint satisfactions along SQA iterations (Derivative Free Optimization method). Dotted lines indicate upper constraint thresholds. The 82 first simulations belong to the SQA initialization phase (see algorithm in Table 2) to build the initial quadratic model.

Physical coupled models	NOx emissions	CO2 emissions	CO emissions	HC emissions
Threshold	Minimized	100 %	150 %	150 %
Initial maps	100 %	100 %	100 %	100 %
QSO Maps	83.5 %	100 %	84.8 %	82.2 %
Optimized maps	78.0 %	100 %	99.9 %	89.7 %

Table 3 Map optimization results with physical AMESim models coupled with statistical models

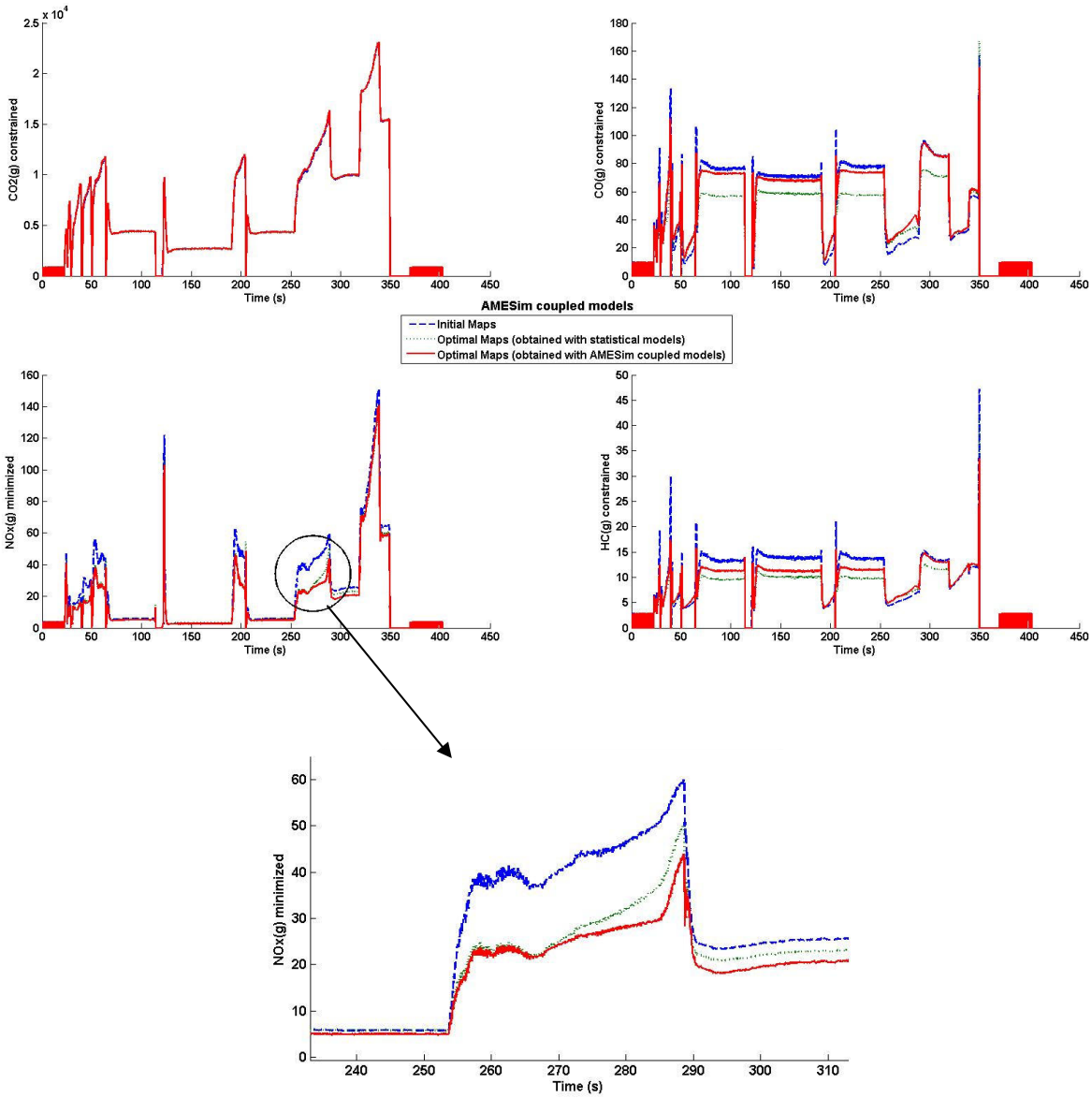


Figure 15 : Pollutant emissions (physical AMESim models coupled with statistical models) along EUDC cycle (1Hz discretization) for initial, Quasi-Static optimized maps, optimized maps (with physical models). (Bottom) Zoom on the 70-100km/h acceleration of EUDC cycle.

6. Conclusions and outlook

Finally an innovative method has been presented to deal with transient effects on engine map calibration process on a Diesel engine, while performing only tests in steady-state conditions. It completes or replaces the standard quasi-static approach.
The main features of the method consist in the different stages listed below:

- Obtain statistical models taking into account engine speed and load as parameters by applying advanced methodologies for domain definition, design of experiments and engine response modelling;
- Tune phenomenological 0D combustion and air loop models on a limited data set, then validate on the engine operating domain;
- Couple of statistical and physical engine models and engine control model to manage the effects of transient operations (especially regarding air loop parameters) on emissions;
- Adapt the formulation of the map optimization and design a derivative-free optimization algorithm to optimize engine responses on a driving cycle.

This methodology has been successfully applied to minimize NOx cumulative emissions on a EUDC cycle of a 1.6 litre engine with biofuel while maintaining initial CO_2 emissions. Nevertheless the results show that the potential of emission reduction remains low on EUDC cycle, because of the smoothness of such a cycle. Transient operations should become a key issue in engine calibration process since driving cycle might change from the NEDC to the World Light Duty Procedure (WLTP).
This method could also be helpful to optimize the engine control transient corrections (for example correction of injection parameters to compensate turbo lag).
One of key issues for system simulation emergence concerning optimization on driving cycles is to have both reduced CPU time and accurate models. Therefore multi-model multi-core simulation platforms ([15]) which allow to reach real-time computation for a virtual engine in keeping the same model accuracy will be investigated in further work.

7. Acknowledgements

The authors kindly acknowledge Fabien Chaudoye for his significant contribution to this paper.

8. References

[1] K. Röpke, R. Gaitzsch, C. Haukap, M. Knaak, C. Knobel, A. Neßler, A. Schaum, U. Schoop, S. Tahl, *DoE – Design of Experiments, Methods and applications in engine development*, Verlag Moderne Industrie, 2005

[2] M. Castagné, Y. Bentolila, A. Hallé, F. Nicolas, D. Sinoquet, *Engine calibration : towards an integrated approach*, DoE in engine development IV, Berlin, Germany, 2007

[3] H. Langouët, F. Delbos, D. Sinoquet and S. Da Veiga (2010) A derivative free optimization method for reservoir characterization inverse problem, 12th European Conference on the Mathematics of Oil Recovery, Oxford, UK, 6 - 9 September 2010 (http://earthdoc.eage.org/detail.php?pubid=41311)

[4] S. Magand, B. Lecointe, F. Chaudoye, M. Castagné (2010), Optimization of a Euro 5 Vehicle Powered by an Ethanol Based Diesel Fuel, SAE Technical Paper 2010-01-1520

[5] F. Chaudoye, M. Castagné, B. Lecointe, S. Magand (2010), Advanced calibration methods applied with an innovative ethanol/diesel fuel formulation, SIA Conference May 2010, Rouen, France

[6] M. Castagné, Y. Bentolila, F. Chaudoye, A. Hallé, F. Nicolas, D. Sinoquet (2008) Comparison of engine calibration methods based on DoE, *Oil and Gas Science and Technology* 63, 4, 563-582. (http://ogst.ifpenergiesnouvelles.fr/)

[7] F. Chaudoye, M. Castagné, D. Sinoquet, F. Wahl (2009), Modelling engine operating space for DoE calibration methods, IAV Conference 2009, Berlin, Germany

[8] A. Albrecht, O. Grondin, F. Le Berr and G. Le Solliec, "Towards a stronger simulation support for engine control design : a methodological point of view", *Oil & Gas Science and Technology – Rev. IFP*, *2007039*, *published on line*, <http:\\ogst.ifp.fr>, 2007

[9] F-A. Lafossas, M. Marbaix and P. Menegazzi, "Development and Application of a 0D D.I. Diesel combustion model for Emissions Prediction", *SAE Technical Paper*, *2007-01-1841*, 2007

[10] C. Barba, C. Burkhardt, K. Boulouchos, M. Bargende, "A phenomenological combustion model for heat release rate prediction in high speed DI Diesel engines with common-rail injection", *SAE Technical Paper*, *2000-01-2933*, 2000

[11] R. Lebas, G. Mauviot, F. Le Berr and A. Albrecht, A Phenomenological Approach to Model Diesel Engine Combustion and In-Cylinder Pollutant Emissions Adapted to Control Strategy, IFAC Paper, ECOSM'09, 2009

[12] G. Woschni, "Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine", *SAE Technical Paper*, *670931*, *SAE Trans, vol. 76*, 1967

[13] D. Sinoquet, H. Langouët, F. Chaudoye, M. Castagné (2009) Multi-objective constrained optimization of engine maps, 5th IAV conference: Design of Experiments (DoE) in Engine Development, Berlin (Germany), June 29-30

[14] M.J.D. Powell (2006) The NEWUOA software for unconstrained optimization without derivatives, Tech. rep., Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Cambridge. (http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2007_05.pdf)

[15] M. Ben Gaid, G. Corde, A. Chasse, B. Léty, R. De La Rubia and M. Ould Abdellahi. Heterogeneous Model Integration and Virtual Experimentation Using xMOD: Application to Hybrid Powertrain Design and Validation. EUROSIM'10, Prague, Czech Republic, September 2010

3 Conclusions sur les résultats numériques obtenus avec la méthode SQA

L'utilisation de la méthode SQA permet d'obtenir de très bons résultats sur le benchmark d'optimisation sous contraintes. Celle-ci est particulièrement économique par rapport aux autres méthodes d'optimisation sans dérivées considérées sur des fonctions lisses, bruités et non-différentiables. Sur les cas sous contraintes linéaires et non linéaires du benchmark CUTEr, la méthode SQA est également capable de résoudre les problèmes sous contraintes de manière plus avantageuse que les autres méthodes d'optimisation sans dérivées considérées. De plus, notre méthode est capable de traiter efficacement les contraintes pour lesquelles les dérivées ne sont pas connues à l'aide de modèles de celles-ci et d'une fonction de pénalisation à poids constants permettant d'évaluer la solution optimale du problème. Notre méthode a permis de résoudre un benchmark complexe de ce type soumis à 68 contraintes sans dérivées et avec 125 paramètres.

Notre méthode s'est également avérée très efficace pour traiter deux applications industrielles en ingénierie de réservoir, et en calibration des moteurs. Notre méthode est performante sur un problème de calage d'historique lorsque le nombre de données est très important, le stockage trop coûteux de l'approximation de la matrice Jacobienne ne permettant pas d'utiliser la méthode Gauss Newton. La prise en compte du caractère moindres carrés de la fonction objectif permet une amélioration notable pour le calage de données de production. Dans le cas de données de production. Dans le cas de calage conjoint de données de production et de données sismiques, les résultats ne sont pas concluants. Le nombre de données empêchant de modéliser séparement tous les termes de la fonction coût associés. Il est alors nécessaire de procéder à un découpage de la fonction coût en termes qui seront modélisés séparemment, ce découpage ayant un sens physique (par puits, par type de données pour les données de production et par couche géologique pour les données sismiques). Une approche permettant d'évaluer les incertitudes sur la prédiction de production de pétrole d'un champ, ainsi qu'un problème de calibration d'un moteur formulé comme un problème d'optimisation sans dérivées sous contraintes sans dérivées ont également été traités à l'aide de notre méthode. Celle-ci permet une réduction efficace de la fonction objectif tout en respectant les contraintes sans dérivées à la convergence.

Deuxième partie

Optimisation Multi-Objectifs sans dérivées

Chapitre 5

Optimisation multi-objectifs

Ce chapitre a pour objectif principal de définir les notions essentielles de l'optimisation multi-objectifs.

Dans un premier temps, les définitions et notations utilisées dans la suite sont décrites. Les différentes approches de résolution de problèmes multi-objectifs sont ensuite introduites. L'utilisation d'objectifs souvent antagonistes pour évaluer une solution rend l'application de méthodes d'optimisation classiques difficile. Le problème nécessite alors d'être transformé et des techniques particulières doivent être mises en place (par exemple en optimisant une somme pondérée des différents objectifs, voir la section 2.1.1 de ce chapitre).

1 Définitions

Considérons le problème d'optimisation de M objectifs $f_1, ..., f_M : X \longrightarrow \mathbb{R}$ à minimiser. La fonction objectif est un vecteur : $f(x) = (f_1(x), ..., f_M(x)) : \mathbb{R}^n \longrightarrow \mathbb{R}^M$, où $\mathbb{R}^M$ est l'espace des objectifs.

Définition 1.1 *Une **solution** $x \in \mathbb{R}^n$ est dite **réalisable** si elle respecte un ensemble de contraintes de la forme $g(x) \leq 0$.*

Le problème s'écrit alors

$$\min_{x \in \mathbb{R}^n} f(x) = (f_1(x), ..., f_M(x)), \tag{5.1}$$

sous les contraintes

$$g(x) \leq 0. \tag{5.2}$$

Les solutions d'un problème d'optimisation multi-objectifs ne peuvent être ordonnées de la même manière que les solutions d'un problème d'optimisation mono-objectif. L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg [75] pour résoudre les problèmes proposés par Schaffer. Il suggère d'utiliser le concept d'optimalité de Pareto pour respecter tous les critères car il refuse de comparer a priori les valeurs de différents

critères. L'utilisation d'une sélection basée sur la notion de dominance de Pareto va faire converger la population vers un ensemble de solutions d'intérêt. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais simplement de les ordonner partiellement.

Vilfredo Pareto [128], mathématicien italien, formule le concept suivant : dans un problème multiobjectifs, il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autres critèresce qui introduit la notion de dominance de Pareto suivante.

Définition 1.2 ***Dominance de Pareto*** *: une solution $x \in X$* ***domine une solution*** *x' si et seulement si*

$$\begin{cases} \forall m \in \{1, ..., M\} : f_m(x) \leq f_m(x') \\ \exists m \in \{1, ..., M\} : f_m(x) < f_m(x') \end{cases} \tag{5.3}$$

On écrit : $x \prec x'$

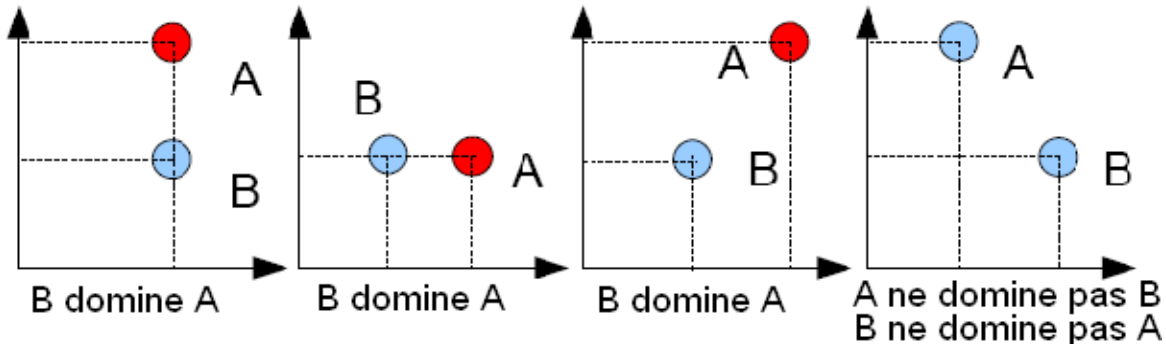


FIGURE 5.1 – Exemples de dominance de Pareto et de non dominance pour une minimisation bi-objectifs

Définition 1.3 *On dit que x* ***domine*** *y* ***au sens de Pareto*** *si pour chacune des fonctions f_m, $\forall m \in \{1, ..., M\}$, x est au moins aussi bon que y, et qu'il existe au moins une fonction f_m pour laquelle x est strictement meilleur que y.*

On définit ainsi l'ensemble des solutions de Pareto qui ne sont pas dominées comme l'ensemble des solutions Pareto optimales (ou solutions non dominées, non inférieures).

Définition 1.4 *Une* ***solution Pareto optimale*** *est :*

$$\{x \,|\, \nexists x' \in X : x' \prec x\} \tag{5.4}$$

Nous constatons qu'il peut exister un grand nombre de solutions Pareto optimales, et qu'aucune de ces solutions n'est meilleure qu'une autre. L'ensemble des valeurs prises par la fonction f pour les solutions Pareto optimales est appelé **front de Pareto**.

Définition 1.5 *Le **front de Pareto** est défini par :*

$$\{f(x) \mid \nexists x' \in X : x' \prec x\} \tag{5.5}$$

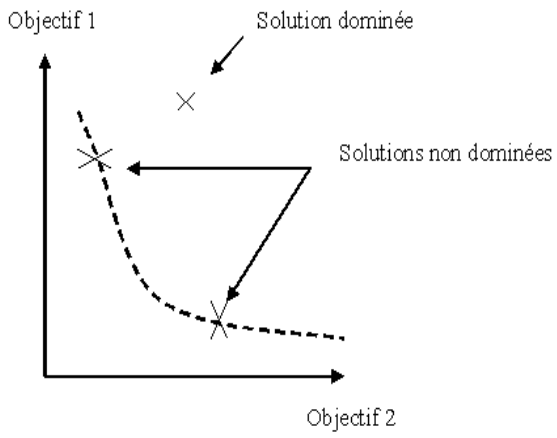


FIGURE 5.2 – Exemple d'un front de Pareto pour l'optimisation de deux objectifs

La figure 5.2 montre l'exemple d'un front de Pareto pour le problème de minimisation de deux objectifs. Les traits en pointillés représentent le front de Pareto.

Les algorithmes d'optimisation multi-objectifs cherchent à estimer le front de Pareto. Pour cela, la diversité des solutions est très recherchée. Ainsi, les différentes solutions du front de Pareto donnent différents compromis entre les différents objectifs visés.

2 Approches de résolution

Deux classifications des différentes stratégies reposant sur deux points de vue différents sont présentées : la classification de l'utilisateur et la classification du créateur.

La solution d'un problème multi-objectifs est un ensemble de solutions. Cependant pour un problème réel, une solution pourra être préférée. L'utilisateur peut faire le choix d'intervenir en amont de la résolution, après celle-ci ou de manière interactive. Les méthodes peuvent alors être réparties dans trois catégories selon ce point de vue (voir [9, 10, 11, 48, 101, 151] certaines méthodes rentrent dans plusieurs catégories différentes).

- **Les méthodes a priori** : l'utilisateur définit ses préférences entre les différents objectifs avant l'utilisation de la méthode d'optimisation.
- **Les méthodes progressives** : l'utilisateur affine son choix de compromis au fur et à mesure du déroulement de la méthode d'optimisation.
- **Les méthodes a posteriori** : l'utilisateur choisit la solution de son choix parmi l'ensemble des solutions fournies par la méthode d'optimisation.

La seconde classification utilisée se place du point de vue du créateur. Il existe trois catégories différentes en fonction de la manière dont elles traitent la présence des objectifs :

- **approches scalaires** : le problème est transformé en problème mono-objectif.
- **approches Pareto** qui utilisent directement la notion d'optimalité Pareto lors de la recherche.
- **approches non scalaires et non Pareto** qui ne transforment pas le problème multi-objectifs en problème mono-objectif mais utilisent des techniques de recherche traitant les objectifs séparément.

2.1 Approches scalaires

Les approches scalaires permettent de résoudre des problèmes multi-objectifs en transformant le problème en un problème mono-objectif. Les trois principales seront détaillées : la méthode d'agrégation, la méthode avec vecteur cible et le méthode ϵ-contraintes.

Ainsi, ces approches permettent d'utiliser des méthodes d'optimisation mono-objectif classiques.

2.1.1 Méthodes d'agrégations

C'est l'une des premières méthodes employées pour la génération de solutions Pareto optimales. Cette méthode consiste à transformer un problème multi-objectifs en un problème qui combine les différentes fonctions objectifs du problème en une seule fonction S de façon linéaire :

$$S(x) = \sum_{m=1}^{M} \lambda_m \cdot f_m(x), \tag{5.6}$$

où les poids λ_m sont compris dans l'intervalle $[0, 1]$ et vérifient $\sum_{m=1}^{M} \lambda_m = 1$. Chaque jeu de poids donne une solution Pareto optimale ou seulement une solution locale. Les résultats obtenus par cette méthode dépendent énormément du choix des poids λ_m, ils indiquent la préférence attribuée aux différents objectifs. Il est donc nécessaire de lancer plusieurs optimisations avec différents poids pour approcher le front de Pareto.

Cette méthode à l'avantage d'être très simple à implémenter. En revanche, aucune agrégation linéaire n'existe afin d'obtenir l'optimalité des parties concaves du front de Pareto (voir [39]).

2.1.2 Méthode avec vecteur cible

Un ensemble de cibles est défini. La méthode d'optimisation tente de minimiser la différence entre les solutions envisagées, les cibles, et les objectifs. Il existe plusieurs approches pour ces méthodes : les méthodes du but à atteindre, du but programmé ou du min-max.

2.1.3 Méthode de ϵ-contraintes

Dans cette méthode, une seule fonction objectif f_k est optimisée tandis que les autres fonctions objectifs sont traitées comme des contraintes. Le problème d'optimisation s'écrit alors :

$$\min_{x \in \mathbb{R}^n} f_k(x), \tag{5.7}$$

sous les contraintes

$$f_j(x) \leq \epsilon_j, j = 1, ..., n, j \neq k. \tag{5.8}$$

L'objectif f_k est l'objectif primaire ou l'objectif préféré. Le front de Pareto optimal est généré en faisant varier les valeurs des ϵ_j. Il est ainsi possible d'approcher l'ensemble Pareto optimal en résolvant des problèmes mono-objectif.

2.2 Approches Pareto

Les approches Pareto utilisent directement la notion de dominance au sens de Pareto. Cette idée a été introduite initialement dans les algorithmes génétiques par Goldberg [75]. Ce concept permet de ne pas favoriser un compromis plutôt qu'un autre mais il fournit une aide précieuse à l'utilisateur. Ce critère de dominance est très utilisé par les algorithmes évolutionnaires multi-objectifs comme NSGA II [41] et MOCMAES (Multi-Objective - Covariance Matrix Adaptation - Evolution Strategy, présenté dans le chapitre 6 suivant).

Les algorithmes évolutionnaires multi-objectifs basés sur le principe de Pareto permettent de trouver plusieurs solutions Pareto optimales en une seule exécution. De plus, ils présentent l'avantage d'être robustes en ce sens qu'ils ne sont pas sensibles aux discontinuités ou à la convexité de la fonction à minimiser.

Les trois opérateurs (sélection, croisement, mutation) du MO-CMA-ES sont spécifiques et en font sa particularité.

D'autres méthodes utilisent l'approche de Pareto comme l'approche géométrique NBI, introduite par Das et Dennis [39, 40], qui consiste à reconstruire le front de Pareto après avoir construit plusieurs aggrégations de fonctions objectifs.

2.3 Approches non scalaires et non Pareto

Ces méthodes ne transforment pas le problème multi-objectifs en un problème mono-objectif mais n'utilisent pas non plus la notion de dominance de Pareto.

Il existe trois types de méthodes.

- **La méthode de sélection parallèle** fait évoluer une population de n individus (de la même manière que les algorithmes évolutionnaires voir la section 2.1 du chapitre 1) puis sélectionne n/M individus pour chacun des M différents objectifs.
- **La méthode lexicographique** définit un certain nombre d'ordres de priorité entre les différents objectifs.

- Et **la méthode avec genre** (et d'attracteur sexuel) affecte un objectif à chaque genre dans un algorithme évolutionnaire. Le genre d'un nouvelle individu sera choisi aléatoirement afin de minimiser simultanément les deux objectifs, car un genre sera toujours jugé d'après l'objectif qui lui a été associé.

L'inconvénient de ce type de méthodes est qu'elles tendent à générer des solutions qui sont largement optimisées pour certains objectifs et très peu pour les autres. Les compromis sont délaissés.

Dans ce chapitre nous avons introduit les notations et définitions nécessaires pour décrire l'optimisation multi-objectifs et présenté brièvement les principales méthodes. Dans ce manuscrit, nous avons privilégié l'utilisation d'un algorithme évolutionnaire utilisant une approche Pareto : la méthode MO-CMA-ES.

Chapitre 6

Multi-Objective - Covariance Matrix Adaptation - Evolution Strategy (MO-CMA-ES)

Le MO-CMA-ES est un algorithme évolutionnaire multi-objectifs. Les algorithmes évolutionnaires ont la particularité d'explorer la totalité du front de Pareto défini précédemment et, ainsi, permettent d'obtenir tous les compromis entre les objectifs visés.

Le MO-CMA-ES utilise le CMA-ES élitiste [85], qui combine le schéma de $(1 + \lambda)$-sélection d'algorithme évolutionnaire [144, 157, 14] avec le CMA-ES de Hansen [87, 86, 88, 36].

Dans ce qui suit, nous nous intéressons au problème de minimisation sans contrainte suivant :

$$\begin{cases} \min f(x) \\ x \in \mathbb{R}^N \end{cases} \tag{6.1}$$

où $f : \mathbb{R}^N \longrightarrow \mathbb{R}^M$ est la fonction coût à minimiser à valeurs réelles et définie sur $\mathbb{R}^N$ (N est le nombre de variables d'optimisation, et M est le nombre d'objectifs à minimiser).

Nous présenterons tout d'abord quelques définitions nécessaires à la sélection multi-objectifs. Puis nous présenterons l'algorithme MO-CMA-ES de Hansen. Enfin, les cas tests de l'optimisation multi-objectifs seront brièvement présentés.

1 Sélection multi-objectifs

La sélection des individus permet d'identifier les "meilleurs" individus d'une population et d'éliminer les "mauvais". Les algorithmes génétiques multi-objectifs les plus efficaces utilisent le concept de Pareto dominance afin d'ordonner les individus et ainsi de sélectionner les "meilleurs". L'algorithme multi-objectifs MO-CMA-ES est basé sur une approche de tri des individus non dominés utilisée dans NSGA II de Deb [41]. Les individus sont triés à l'aide de leur niveau de non dominance. Puis, pour ordonner les individus de même niveau de non dominance, un critère de diversité est défini. Chaque individu

$a_i^{(g)}$ à la génération g représente la solution candidate $x_i^{(g)} \in X \subseteq \mathbb{R}^n$ d'un problème à n dimensions avec M objectifs.

1.1 Premier critère de sélection : tri par non dominance

Le premier critère de sélection effectue un tri des élements de la population A de solutions candidates à l'aide de leur niveau de non dominance. Les solutions non dominées de A sont notées $ndom(A) = \{a \in A | \nexists a' \in A : a' \prec a\}$. Le front de Pareto de A est donné par les coordonnées dans l'espace des objectifs des solutions non dominées

$$\{(f_1(a), ...f_M(a)) \,|a \in ndom(A)\}\,.$$

Les éléments de $ndom(A)$ sont notés de rang 1. Les autres rangs sont définis récursivement en considérant l'espace sans les solutions de rang inférieur. Formellement, $dom_l(A) = dom_{l-1}(A) \backslash ndom_l(A)$ et $ndom_l(A) = ndom(dom_{l-1}(A))$ pour $l \in \{1, ...\}$ avec $dom_0 = A$. Pour $a \in A$, nous définissons le niveau de non dominance $ND(a, A) = i$ si et seulement si $a \in ndom_i(A)$.

Afin d'ordonner les solutions ayant le même rang de non dominance, un second critère de tri est nécessaire. Ce critère est très important pour les performances d'un algorithme génétique : en effet, après quelques générations, il peut y avoir plus de solutions non dominées dans la population que de solutions à choisir. Ce critère a pour but de maintenir la diversité du front de Pareto afin d'obtenir des points bien répartis sur le front et, ainsi, d'obtenir des compromis variés entre les différents objectifs.

1.2 Second critère de sélection

Le second critère de sélection est un critère de diversité des solutions qui permet de tendre à distribuer ces solutions uniformément sur le front de Pareto. La méthode MO-CMA-ES considère deux critères de tri : la distance d'encombrement de Deb [41] et la contribution à l'hypervolume de Emmerich, Beume, et Naujoks [58].

1.2.1 Distance d'encombrement

Dans l'algorithme NSGA II de Deb [41], les solutions non dominées A', qui ont le même niveau de non dominance, sont ordonnées selon leurs contributions à la diversité des valeurs de la fonction objectif dans A'. La distance d'encombrement (Crowding-distance) permet d'estimer la densité de population autour de chaque individu dans l'espace des objectifs, ce qui correspond à leur diversité. Pour M objectifs, la distance d'encombrement de $a \in A'$ est donnée par :

$$c(a, A') = \sum_{m=1}^{M} \frac{c_m(a, A')}{f_m^{max} - f_m^{min}} \tag{6.2}$$

où f_m^{min} et f_m^{max} sont l'estimation des valeurs minimale et maximale du m-ième objectif et

$$c_m(a, A') := \begin{cases} \infty, \text{ si } f_m(a) = \min\{f_m(a')|a' \in A'\} \text{ ou } f_m(a) = \max\{f_m(a')|a' \in A'\} \\ \min\{f_m(a'') - f_m(a')|a', a'' \in A' : f_m(a') < f_m(a) < f_m(a'')\}, \text{ autrement.} \end{cases}$$

La crowding-distance est égale à l'infini pour les solutions extrêmes dans l'espace des objectifs et est égale à la somme des longueurs de l'hyperrectangle (ou orthotope) qui touche les solutions voisines si les solutions ne sont pas extrêmes sur le front de Pareto (voir figure 6.1). A l'aide du niveau de non dominance et de la distance d'encombrement, nous définissons la relation :

$$\begin{aligned} a \prec_{c,A} a' \quad &\Leftrightarrow ND(a, A) < ND(a', A) \text{ ou} \\ &\left[(ND(a, A) = ND(a', A)) \wedge (c(a, ndom_{ND(a',A)}(A)) > c(a', ndom_{ND(a',A)}(A)))\right], \end{aligned} \tag{6.3}$$

pour $a, a' \in A$. En comparant à l'aide de $\prec_{c,A}$, a est "meilleur" que a' si a a un "meilleur" niveau de non dominance ou si a et a' ont le même niveau mais a est dans une région moins encombrée de l'espace des objectifs, ce qui implique plus de diversité.

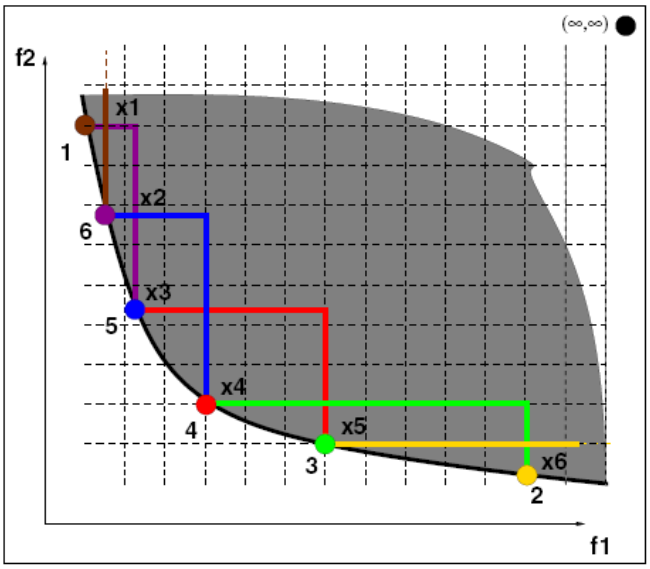


FIGURE 6.1 – Distance d'encombrement en dimension 2

1.2.2 Mesure de l'hypervolume

La mesure de l'hypervolume (Contributing Hypervolume ou S-metric) est introduite par Zitzler et Thiele [186] dans le domaine de l'optimisation multi-objectifs par algorithme évolutionnaire. Elle peut être définie comme la mesure de Lebesgue Λ de l'union des hypercubes dans l'espace des objectifs :

$$S_{a_{ref}}(A') = \Lambda \left(\bigcup_{a \in ndom(A')} \{(f_1(a'), ..., f_M(a'))|a \prec a' \prec a_{ref}\} \right), \tag{6.4}$$

où a_{ref} est un point de référence choisi de manière appropriée. La contribution à l'hypervolume d'un point $a \in ndom(A')$ est donnée par :

$$\Delta_S(a, A') := S_{a_{ref}}(A') - S_{a_{ref}}(A' \backslash a) \tag{6.5}$$

La mesure de l'hypervolume est utilisée dans l'algorithme évolutionnaire proposé par Emmerich, Beume, et Naujoks [58]. Le rang $s(a, A')$ d'un individu a peut être défini récursivement par sa contribution à l'hypervolume. Ainsi, l'individu qui contribue moins à l'hypervolume de A' aura le dernier rang, celui qui contribue le moins à l'hypervolume de A' après celui de dernier rang aura l'avant dernier rang, et ainsi de suite. Formellement, pour $a \in ndom(A')$ on a $s(a, A') = 1$ si $a = \arg\min_{a' \in A'} \{\Delta_S(a', A')\}$ et $s(a, A') = n$ si $a = \arg\min_{a' \in A'} \{\Delta_S(a', A') \backslash \{a''|s(a'', A') < n\})\}$. Le point de référence a_{ref} est choisi dans le but que chaque individu a tel que $f_m(a) = \min f_m(a')|a' \in A'$ pour tout $m \in \{1, ..., M\}$ donne le meilleur rang. Ce sont les individus situés sur le bord du front de Pareto qui sont sélectionnés de préférence.

Pour deux objectifs, ce critère peut être calculé efficacement à l'aide de l'équation donnée par Emmerich, Beume et Naujoks [58] : pour tous les points d'un même rang de non dominance B, un tri par ordre croissant est effectué selon la valeur de la première fonction objectif. Pour $B = \{b_1, ..., b_{|B|}\}$, $\Delta_S(b_i, B) = (f_1(s_{i+1}) - f_1(s_i)) \cdot (f_2(s_{i-1}) - f_2(s_i))$. Le comportement du calcul de la mesure de l'hypervolume est un gros désavantage, celle-ci risque d'être difficile à calculer lorsque le nombre d'objectifs augmente.

La mesure de l'hypervolume est représentée par l'aire des rectangles rattachés à leur point du front de Pareto (voir la figure 6.2).

A l'aide du niveau de non dominance et de la mesure de l'hypervolume, nous définissons la relation :

$$\begin{aligned} a \prec_{s,A} a' \quad & \Leftrightarrow ND(a, A) < ND(a', A) \text{ ou} \\ & \left[(ND(a, A) = ND(a', A)) \wedge (s(a, ndom_{ND(a',A)}(A)) > c(a', ndom_{ND(a',A)}(A))) \right], \end{aligned} \tag{6.6}$$

pour $a, a' \in A$. En comparant à l'aide de $\prec_{s,A}$, a est "meilleur" que a' si a a un "meilleur" niveau de non dominance ou si a et a' ont le même niveau mais a contribue plus à l'hypervolume, ce qui implique plus de diversité.

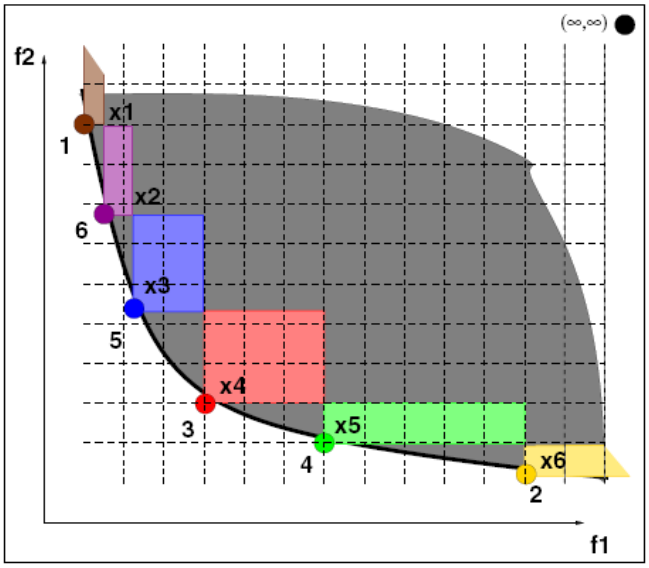


FIGURE 6.2 – Mesure de l'hypervolume en dimension 2

L'optimisation multi-objectifs a pour but de trouver différents compromis entre les objectifs visés. Associés, le critère de non dominance et la distance d'encombrement ou la mesure de l'hypervolume permettent d'effectuer la sélection des individus de la population lors de chaque génération de l'algorithme génétique multi-objectifs.

2 Croisement et mutation

L'opérateur de croisement a pour but d'enrichir la diversité de la population. Il consiste à combiner linéairement les individus entre eux. Cet opérateur n'est pas présent dans la méthode MO-CMA-ES.

L'opérateur de mutation joue un rôle décisif quant à la convergence de l'algorithme génétique. La mutation consiste à perturber un individu par un bruit gaussien centré, c'est-à-dire par l'ajout d'une variable aléatoire gaussienne centrée. L'écart-type de la distribution appelé "pas de mutation" noté σ, caractérise la distribution.

En général, le pas de mutation choisi n'est pas le même pour toutes les composantes afin de prendre en compte les sensibilités différentes de la fonction coût aux différentes composantes.

On note σ_i le pas de mutation correspondant à la $i^{\text{ème}}$ composante d'un vecteur $x^{(g)}$. Alors

$x^{(g+1)}(i) = x^{(g)}(i) + \mathcal{N}(0, \sigma_i^2)$ pour $i = 1, ..., N$,

où $\mathcal{N}(0, \sigma_i^2)$ désigne la distribution normale centrée, de variance σ_i^2.

Le réel σ_i peut être constant auquel cas la mutation est dite "fixe" ou variable d'une itération à l'autre. Dans ce dernier cas, la mutation est dite "adaptative". Il est à noter

que les mutations adaptatives ont été introduites car le choix d'un écart-type constant est délicat. Dans ce qui suit, nous nous intéressons à ce type de mutation.

Parmi les méthodes adaptatives, on distingue la loi du 1/5 de Rechenberg. Si dans moins de 20% des cas, le nouvel individu généré par mutation est meilleur que l'individu muté, alors le pas de mutation est diminué à la génération suivante. Dans le cas contraire, le pas est augmenté. Cette méthode présente l'inconvénient de ne pas pouvoir s'appliquer dans le cas de la mutation des vecteurs composante par composante.

Des méthodes adaptatives différentes permettent de pallier cet inconvénient, notamment celle du MO-CMA-ES basée sur l'algorithme CMA-ES de Hansen [36, 87, 86, 88]. Cette méthode permet d'adapter différents pas de mutation, par la prise en compte d'information locale mais aussi d'information cumulée. Fondamentalement, la méthode consiste à adapter la matrice de covariance de la distribution de mutation de façon à aligner le plus grand axe de la distribution avec la direction de plus grand progrès estimé.

2.1 Equations de base

Dans le $\lambda_{MO} \times (1+\lambda)$-MO-CMA-ES, nous maintenons une population de λ_{MO} individus. Pour simplifier, nous considérons le cas standard $\lambda = 1$, avec λ le nombre d'enfants provenant de chacun des parents. Le k-ième individu de la génération g est noté $a_k^{(g)} = \left[x_k^{(g)}, \bar{p}_{succ,k}^{(g)}, \sigma_k^{(g)}, p_{c,k}^{(g)}, C_k^{(g)}\right]$, où
$x_k^{(g)} \in \mathbb{R}^N$ est le vecteur des paramètres,
$\bar{p}_{succ,k}^{(g)} \in [0,1]$ est le taux de succès moyenné,
$\sigma_k^{(g)}$ est l'écart-type global,
$p_{c,k}^{(g)} \in \mathbb{R}^N$ est le chemin d'évolution et
$C_k^{(g)} \in \mathbb{R}^{N\times N}$ est la matrice de covariance.

A chaque itération g, un tirage aléatoire de $\lambda = 1$ vecteur est effectué pour chaque λ_{MO} parents $x_k^{(g)}$, selon une loi multinormale notée $\mathcal{N}\left(x_k^{(g)}, \sigma_k^{(g)2} \cdot C_k^{(g)}\right)$, centrée en $x_k^{(g)}$, de matrice de covariance $C_k^{(g)}$, et de "variance globale" $\sigma_k^{(g)2}$:

$$x_k'^{(g+1)} \sim \mathcal{N}\left(x_k^{(g)}, \sigma_k^{(g)2} \cdot C_k^{(g)}\right) \text{ pour } k = 1, ..., \lambda_{MO}, \tag{6.7}$$

où $C_k^{(g)}$ est une matrice symétrique définie positive.
Notons que, par abus de langage, on appellera par la suite matrice de covariance, la matrice $C_k^{(g)}$, au lieu de $\sigma_k^{(g)2} \cdot C_k^{(g)}$.

Tous les λ^{MO} parents et les $\lambda \times \lambda^{MO}$ enfants de la génération g forment l'ensemble $Q^{(g)} = \left\{a_k'^{(g+1)}, a_k^{(g)} | 1 \leq k \leq \lambda_{MO}\right\}$ qui représente la population à la génération g.

On rappelle que toute matrice définie positive C est diagonalisable en base orthonormée. La matrice C peut ainsi s'écrire sous la forme : $C = B \cdot D^2 \cdot B^T$, où $D^2 = diag(\sigma_i^2)$ désigne la matrice diagonale dont les éléments diagonaux $\{\sigma_i^2\}_{i=1,...,N}$ sont les valeurs propres de la matrice C. La matrice B est, quant à elle, une matrice orthogonale, c'est-à-dire dont les vecteurs colonnes sont de norme unitaire et deux à deux orthogonaux. Les colonnes de B sont les vecteurs propres normalisés associés aux valeurs propres de C.

On introduit les notations suivantes : $C^{1/2} = B \cdot D \cdot B^T$, où $D = diag(\sigma_i)$. On peut ainsi réécrire la loi de distribution $\mathcal{N}(x, C)$ de la façon suivante :

$$\begin{aligned}\mathcal{N}(x, C) &\sim x + \mathcal{N}(0, C) \\ &\sim x + C^{1/2} \cdot \mathcal{N}(0, Id_N) \\ &\sim x + B \cdot D \cdot B^T \cdot \mathcal{N}(0, Id_N)\end{aligned}$$

où la notation Id_N désigne la matrice identité en dimension N.

Puisque la matrice B est orthogonale, on a également : $B^T \cdot \mathcal{N}(0, Id_N) \sim \mathcal{N}(0, Id_N)$, ce qui donne la relation suivante : $\mathcal{N}(x, C) \sim x + B \cdot D \cdot \mathcal{N}(0, Id_N)$.

Ainsi, à chaque itération de l'algorithme, chacun des $\lambda_{MO} \times (\lambda = 1)$ vecteurs est généré de la façon suivante : N tirages indépendants de loi normale centrée réduite sont effectués (un tirage par composante), constituant ainsi un vecteur aléatoire $z \in \mathbb{R}^N$ qui suit la distribution $\mathcal{N}(0, Id_N)$. Chaque composante $z(i)$ (pour $i = 1, ..., N$) du vecteur z est ensuite dilatée ou rétrécie par le coefficient σ_i constituant le $i^{\text{ème}}$ terme diagonal de la matrice D. Finalement, la multiplication par la matrice B permet d'écrire le vecteur ainsi obtenu dans le système de coordonnées défini par les vecteurs propres de la matrice C.

Ainsi, $x_k'^{(g+1)} \sim x_k^{(g)} + \sigma_k^{(g)} \cdot \underbrace{B_k^{(g)} \cdot D_k^{(g)} \mathcal{N}(0, Id_N)}_{\sim \mathcal{N}\left(0, C_k^{(g)}\right)}$ pour $k = 1, ..., \lambda_{MO}$.

L'écart-type global des parents et des enfants est mis à jour selon la réussite des mutations effectuées, c'est-à-dire si l'enfant est "meilleur" que le parent selon l'un des critères $\prec_{c,A}$ ou $\prec_{s,A}$. La matrice de covariance des enfants est mise à jour en tenant compte de la mutation. Enfin les "meilleurs" λ^{MO} individus de Q^g triés à l'aide des critères de sélection multi-objectifs vus précédemment ($\prec_{c,Q^g}$ ou $\prec_{s,Q^g}$) forment la prochaine génération de parents.

2.2 Estimation des paramètres caractéristiques de la distribution

Le principe de mutation du MO-CMA-ES basé sur le CMA-ES élitiste qui combine le schéma de sélection de Rechenberg avec le CMA-ES standard, consiste à adapter les paramètres caractérisant la distribution $\mathcal{N}\left(x_k^{(g)}, \sigma_k^{(g)2} \cdot C_k^{(g)}\right)$ de façon à diriger les distributions dans le voisinage du front de Pareto optimal.

La mise à jour de la matrice de covariance $C^{(g)}$ peut être raisonnablement appliquée dans une version élitiste, mais l'adaptation de l'écart-type global $\sigma^{(g)}$ est remplacée par une règle de succès basée sur cet écart-type global. Cette méthode ne peut pas être appliquée facilement car la mise à jour du "chemin d'évolution" est inefficace lorsqu'aucun enfant "réussi" n'est produit.

2.2.1 Adaptation de l'écart-type global $\sigma_k^{(g)}$:

Le paramètre $\sigma_k^{(g)}$ permet d'adapter rapidement la distribution de mutation au "terrain" de recherche. Plus précisément, ce paramètre permet de dilater ou rétrécir la distribution selon le nombre de "succès" d'enfants sur leurs parents. Le principe en est présenté ci-dessous.

Pour tout $k = 1, ..., \lambda_{MO}$, soit

$$\lambda_{succ,Q^{(g)},k}^{(g+1)} = \begin{cases} 1 & \text{, si } a_k'^{(g+1)} \prec_{Q^{(g)}} a_k^{(g)} \\ 0 & \text{, autrement} \end{cases}$$

le taux de succès des $\lambda = 1$ enfant $a_k'^{(g+1)}$ sur leur parent $a_k^{(g)}$. Soit $\prec_{Q^{(g)}}$ l'une des deux relations définies précédemment : $\prec_{s,Q^{(g)}}$, $\prec_{c,Q^{(g)}}$.

Principe : Cette règle de mise à jour s'inspire de la loi du 1/5 de Rechenberg [144]. La mise à jour de $\sigma^{(g)}$ fait intervenir un taux de succès moyenné noté $\bar{p}_{succ,k}^{(g)}$. L'idée est que si, au cours de deux itérations successives, la moyenne de la distribution gaussienne évolue dans la même direction et le même sens, il est équivalent d'effectuer un pas unique et plus long, autrement dit une seule itération suffit. On dit alors par abus de langage que les pas sont corrélés. De même, si les pas sont parallèles et de sens opposé (anti-corrélés), il est préférable de réduire le pas de mutation au cours de la dernière itération, pour éviter que les pas ne se compensent mutuellement. Ainsi, la situation idéale se produit lorsque les pas effectués sont orthogonaux (décorrélés). Le taux de succès moyenné $\bar{p}_{succ,k}^{(g)}$ a été introduit afin de tenter de se maintenir dans cette situation idéale. La "longueur" de $\bar{p}_{succ,k}^{(g)}$ est comparée à celle d'un chemin "idéal" où les pas sont générés de façon indépendante et sont par conséquent décorrélés.

Le pas de mutation $\sigma_k^{(g)}$ est mis à jour en conséquence.

Taux de succès moyenné : $\bar{p}_{succ,k}^{(g+1)} = (1 - c_p) \cdot \bar{p}_{succ,k}^{(g)} + c_p \cdot \lambda_{succ,Q^{(g)},k}^{(g+1)}$,
où le coefficient c_p $(0 < c_p \leq 1)$ est appelé taux d'apprentissage ou temps d'accumulation pour le taux de succès moyenné.

Un taux de succès moyenné $\bar{p}_{succ,k}^{(g+1)}$ est utilisé à la place du taux de succès $\lambda_{succ,Q^{(g)},k}^{(g+1)}$ afin de lisser les changements de l'écart-type global et ceci n'a qu'une influence minime sur le taux de changement maximal de l'écart-type global.

Adaptation de $\sigma_k^{(g)}$: L'adaptation de $\sigma_k^{(g)}$ fait intervenir le taux de succès moyenné $\bar{p}_{succ,k}^{(g+1)}$. La mise à jour a donc pour but d'augmenter l'écart-type global si le taux de succès (la fraction d'enfants "meilleurs" que leurs parents) est grand, et de le rabaisser si celui-ci est bas.

$\sigma_k^{(g+1)} = \sigma_k^{(g)} \cdot \exp\left(\frac{1}{d} \cdot \left(\bar{p}_{succ,k}^{(g+1)} - \frac{p_{succ}^{target}}{1-p_{succ}^{target}}(1 - \bar{p}_{succ,k}^{(g+1)})\right)\right)$,
où d est un coefficient d'atténuation.

La règle de mise à jour est effectuée en modifiant l'argument de la fonction exponentielle. L'argument de l'exponentielle est toujouts compris entre $-1/d$ et $1/d$ (car

$p_{succ}^{target} < 0.5$). Donc le coefficient d'atténuation d contrôle le taux d'adaptation de l'écart-type global $\sigma_k^{(g)}$.

On remarque que l'équation permet d'ajuster le pas $\sigma_k^{(g)}$ en fonction de la longueur de $\bar{p}_{succ}$. Plus précisément, si l'on a l'inégalité $\bar{p}_{succ,k}^{(g+1)} > p_{succ}^{target}$, alors l'argument de l'exponentielle est positif donc le pas de mutation est augmenté. Inversement, si $\bar{p}_{succ,k}^{(g+1)} < p_{succ}^{target}$ l'argument de l'exponentielle est négatif donc le pas de mutation est diminué. En cas d'égalité, l'écart-type $\sigma_k^{(g)}$ reste identique.

L'adaptation de la matrice de covariance $C_k^{(g)}$ se fait séparément de celle de $\sigma_k^{(g)}$. En effet, les paramètres $C_k^{(g)}$ et $\sigma_k^{(g)}$ varient sur des échelles de temps différentes : $\sigma_k^{(g)}$ varie plus vite que $C_k^{(g)}$. Cet écart est visible en regardant la valeur des taux d'apprentissage respectivement pour la matrice de covariance et pour l'écart-type global.

2.2.2 Adaptation de la matrice de covariance $C_k^{(g)}$:

Principe : une approximation de la matrice de covariance $C_k^{(g+1)}$ s'obtient par la prise en compte de la totalité des passages d'une génération à la suivante jusqu'à l'itération courante. Elle fait intervenir une quantité appelée "chemin d'évolution" et notée $p_{c,k}^{(g)}$.

Chemin d'évolution : le vecteur $p_{c,k}^{(g)}$ est déterminé à partir de la suite des pas consécutifs effectués par la moyenne au cours des itérations de l'algorithme. Il est également appelé "chemin cumulé". Son expression est donnée par l'équation suivante :

$$p_{c,k}^{\prime(g+1)} = (1-c_c)\cdot p_{c,k}^{(g)} + \begin{cases} \sqrt{c_c\cdot(2-c_c)}\cdot\frac{x_k^{\prime(g+1)}-x_k^{(g)}}{\sigma_k^{(g)}} & \text{, si } \bar{p}_{succ,k}^{\prime(g+1)} < p_{thresh} \\ 0 & \text{, sinon} \end{cases} \tag{6.8}$$

La mise à jour du chemin d'évolution $p_{c,k}^{(g)}$ dépend de la valeur du taux de succès moyenné $\bar{p}_{succ,k}^{(g+1)}$. Si ce taux de succès moyenné $\bar{p}_{succ,k}^{(g)}$ est supérieur au coefficient p_{thresh}, le chemin d'évolution $p_{c,k}^{(g)}$ n'est pas remis à jour afin d'empêcher que les axes de la matrice de covariance $C_k^{(g+1)}$ augmentent lorsque la taille de l'écart-type global est petite. Si le taux de succès moyenné $\bar{p}_{succ,k}^{(g)}$ est inférieur au coefficient p_{thresh}, la mise à jour du chemin d'évolution $p_{c,k}^{(g)}$ est effectuée suivant un lissage exponentiel.

Proposition 2.1 *Le coefficient $\sqrt{c_c\cdot(2-c_c)}$ normalise la variance de $p_{c,k}^{(g)}$. Plus précisément, nous allons montrer par récurrence sur g que sous la condition $p_{c,k}^{(0)} \sim \mathcal{N}(0, C_k^{(0)})$, nous obtenons l'équation suivante : $p_{c,k}^{(g)} \sim \mathcal{N}(0, C_k^{(g)})$ pour tout $g \geq 0$.*

Démonstration 2.2 *On suppose l'équation ci-dessus vérifiée pour g donné. En simplifiant l'équation 6.8, nous avons $p_{c,k}^{(g+1)} = (1-c_c)\cdot p_{c,k}^{(g)} + \sqrt{c_c\cdot(2-c_c)}\cdot\frac{x_k^{\prime(g+1)}-x_k^{(g)}}{\sigma_k^{(g)}}$, or l'équation 6.7 est équivalente à : $\frac{x_k^{\prime(g+1)}-x_k^{(g)}}{\sigma_k^{(g)}} \sim \mathcal{N}(0, C_k^{(g)})$ pour $k = 1, ..., \lambda_{MO}$. De plus, la variable aléatoire $p_{c,k}^{(g)}$ est indépendante des tirages effectués à l'itération $g+1$.*

Ainsi, comme par hypothèse de récurrence nous avons $p_{c,k}^{(g)} \sim \mathcal{N}(0, C_k^{(g)})$*, nous obtenons finalement* $p_{c,k}^{(g+1)} \sim \mathcal{N}\left(0, ((1-c_c)^2 + c_c \cdot (2-c_c)\cdot) \cdot C_k^{(g)}\right)$*. D'où nous avons bien* $p_{c,k}^{(g+1)} \sim \mathcal{N}\left(0, C_k^{(g+1)}\right)$*. Ceci achève la preuve par récurrence.*

Le paramètre c_c $(0 \leq c_c \leq 1)$ est appelé "taux d'apprentissage" ou "temps d'accumulation" pour le chemin d'évolution. Le rapport $1/c_c$ représente le nombre de générations au bout duquel les 2/3 de l'information originale ont disparu. En effet, pour de petites valeurs de c_c, on a : $(1-c_c)^g \approx \exp(-g \cdot c_c)$.

Ainsi, pour $g = 1/c_c$, on a bien $(1-c_c)^g = \frac{1}{e} \approx 0.63$. Autrement dit, après un nombre de générations égal à $1/c_c$, il ne reste plus que 37% de l'information originale, contenue dans le chemin d'évolution $p_{c,k}^{(0)}$. Le choix de ce paramètre est important, puisqu'une valeur faible de c_c conduit à un apprentissage lent. A l'opposé, une valeur élevée peut conduire à la dégénérescence du chemin d'évolution. En général, la valeur de c_c est comprise entre $1/N$ et $1/\sqrt{N}$.

Adaptation de $C_k^{(g)}$: l'expression de $C_k^{(g+1)}$ à l'itération $g+1$ est donnée par l'équation suivante :

$$C_k'^{(g+1)} = (1-c_{cov}) \cdot C_k^{(g)} + \begin{cases} c_{cov} \cdot p_{c,k}'^{(g+1)} \cdot p_{c,k}'^{(g+1)^T}, \text{ si } \bar{p}'^{(g+1)}_{succ,k} < p_{thresh} \\ c_{cov} \cdot (p_{c,k}'^{(g+1)} \cdot p_{c,k}'^{(g+1)^T} + c_c \cdot (2-c_c) \cdot C_k^{(g)}), \text{ sinon} \end{cases} \tag{6.9}$$

De même que précédemment, le paramètre c_{cov} $(0 \leq c_{cov} < c_c \leq 1)$, appelé "taux d'apprentissage" ou "temps d'accumulation" pour la matrice de covariance, représente le nombre de générations au bout duquel il ne reste plus que 37 % de l'information originale contenue dans $C_k^{(0)}$. Typiquement, c_{cov} a une valeur comprise entre $1/N$ et $1/\sqrt{N}$.

La mise à jour de la matrice de covariance $C_k^{(g)}$ dépend du chemin d'évolution $p_{c,k}^{(g)}$. Si le taux de succès moyenné $\bar{p}_{succ,k}^{(g+1)}$ est inférieur au coefficient p_{thresh}, la nouvelle matrice de covariance est une moyenne pondérée de l'ancienne matrice de covariance et du produit de vecteurs de $p_{c,k}^{(g)}$. Si le taux de succès moyenné $\bar{p}_{succ,k}^{(g)}$ est supérieur au coefficient p_{thresh}, le second membre de la mise à jour de $p_{c,k}^{(g)}$ est absent et donc $p_{c,k}^{(g)}$ rétrécit. Bien qu'en introduisant un changement mineur, le terme $c_c(2-c_c)C_k^{(g)}$ compense le rétrécissement de $p_{c,k}^{(g)}$ dans $C_k^{(g)}$. En effet, le facteur de normalisation a été choisi afin que $p_{c,k}^{(g+1)} \sim \mathcal{N}(0, C), \forall k = 1, ..., \lambda_{MO}$. En supposant $p_{c,k}^{(g)} \sim \mathcal{N}(0, C), \forall k = 1, ..., \lambda_{MO}$, si $\bar{p}_{succ} < p_{thresh}$ alors $p_{c,k}^{(g+1)} \sim \mathcal{N}(0, C), \forall k = 1, ..., \lambda_{MO}$ car $\frac{x_k'^{(g+1)} - x_k^{(g)}}{\sigma_k^{(g)}} \sim \mathcal{N}(0, C)$. En revanche, si $\bar{p}_{succ} \geq p_{thresh}$, $\frac{x_k'^{(g+1)} - x_k^{(g)}}{\sigma_k^{(g)}} \sim \mathcal{N}(0, (1-c_c)^2 C)$. Donc dans ce cas, le terme $c_c(2-c_c)C_k^{(g)}$ compensera le rétrécissement de $p_{c,k}^{(g)}$ dans $C_k^{(g)}$.

Le principe de mutation du MO-CMA-ES adapte donc les paramètres caractérisant les distributions $\mathcal{N}\left(x_k^{(g)}, \sigma_k^{(g)2} \cdot C_k^{(g)}\right)$ dans le voisinage du front de Pareto optimal.

3 L'algorithme MO-CMA-ES

Finalement, l'algorithme 6.1 représente $\lambda_{MO} \times (1+1)$-MO-CMA-ES (avec $Q^{(g)}_{\prec:i}$ représentant le i-ème meilleur individu de $Q^{(g)}$ trié à l'aide des critères de sélection multi-objectifs vus précédemment : $\prec_{c,Q^{(g)}}$ ou $\prec_{s,Q^{(g)}}$).

Algorithme 6.1 Algorithme $\lambda_{MO} \times (\lambda+1)$-MO-CMA-ES pour $\lambda = 1$

- Initialisation : $g = 0$, $x^{(0)}$, $\sigma^{(0)}$, $C^{(0)} = Id_N$, $B^{(0)} = Id_N$, $D^{(0)} = Id_N$, $\bar{p}^{(0)}_{succ} = p^{target}_{succ}$, $p^{(0)}_c = 0$, λ_{MO}, $\lambda = 1$.
- Répéter jusqu'au critère d'arrêt :

(1) Pour $k = 1, ..., \lambda_{MO}$
$a'^{(g+1)}_k \leftarrow a^{(g)}_k$
$x'^{(g+1)}_k \sim \mathcal{N}\left(x^{(g)}_k, {\sigma^{(g)}_k}^2 C^{(g)}_k\right)$

(2) $Q^{(g)} = \left\{a'^{(g+1)}_k, a^{(g)}_k | 1 \leq k \leq \lambda_{MO}\right\}$

(3) Pour $k = 1, ..., \lambda_{MO}$

- mise à jour du taux de succès des λ enfants $a'^{(g+1)}_k$ sur leur parent $a^{(g)}_k$

$$\lambda^{(g+1)}_{succ,Q^{(g)},k} = \begin{cases} 1 & \text{, si } a'^{(g+1)}_k \prec_{Q^{(g)}} a^{(g)}_k \\ 0 & \text{, autrement} \end{cases}$$

- mise à jour de l'écart-type global $\sigma^{(g)}_k$ pour les enfants :

(i) $\bar{p}'^{(g+1)}_{succ,k} = (1-c_p) \cdot \bar{p}^{(g)}_{succ,k} + c_p \cdot \lambda^{(g+1)}_{succ,Q^{(g)},k}$

(ii) $\sigma'^{(g+1)}_k = \sigma^{(g)}_k \cdot \exp\left(\frac{1}{d} \cdot \left(\bar{p}'^{(g+1)}_{succ,k} - \frac{p^{target}_{succ}}{1-p^{target}_{succ}}(1 - \bar{p}'^{(g+1)}_{succ,k})\right)\right)$

- mise à jour de l'écart-type global $\sigma^{(g)}_k$ pour les parents :

(i) $\bar{p}^{(g+1)}_{succ,k} = (1-c_p) \cdot \bar{p}^{(g)}_{succ,k} + c_p \cdot \lambda^{(g+1)}_{succ,Q^{(g)},k}$

(ii) $\sigma^{(g+1)}_k = \sigma^{(g)}_k \cdot \exp\left(\frac{1}{d} \cdot \left(\bar{p}^{(g+1)}_{succ,k} - \frac{p^{target}_{succ}}{1-p^{target}_{succ}}(1 - \bar{p}^{(g+1)}_{succ,k})\right)\right)$

- mise à jour de la matrice de covariance $C^{(g+1)}_k$ pour les enfants :

(i) $p'^{(g+1)}_{c,k} = (1-c_c) \cdot p^{(g)}_{c,k} + \begin{cases} \sqrt{c_c \cdot (2-c_c)} \cdot \frac{x'^{(g+1)}_k - x^{(g)}_k}{\sigma^{(g)}_k} & \text{, si } \bar{p}'^{(g+1)}_{succ,k} < p_{thresh} \\ 0 & \text{, sinon} \end{cases}$

(ii) $C'^{(g+1)}_k = (1-c_{cov}) \cdot C^{(g)}_k + \begin{cases} c_{cov} \cdot p'^{(g+1)}_{c,k} \cdot {p'^{(g+1)}_{c,k}}^T & \text{, si } \bar{p}'^{(g+1)}_{succ,k} < p_{thresh} \\ c_{cov} \cdot (p'^{(g+1)}_{c,k} \cdot {p'^{(g+1)}_{c,k}}^T + c_c(2-c_c) \cdot C^{(g)}_k) & \text{, sinon} \end{cases}$

(4) Pour $i = 1, ..., \lambda_{MO}$
$a^{(g+1)}_i \leftarrow Q^{(g)}_{\prec:i}$

(5) $g \leftarrow g + 1$

TABLE 6.1 – Tableau des valeurs par défaut des paramètres du MO-CMA-ES (N : nombre de variables d'optimisation)

Sélection :		
λ_{MO}	nombre d'individus de la population sélectionnés parmi le total des individus $\lambda_{MO} \times (1+\lambda)$	100
λ	nombre d'enfants générés par chacun des λ_{MO} parents	1
Ecart-type global : $\sigma_k^{(g)}$		
d	coefficient d'atténuation qui contrôle le taux d'adaptation de l'écart-type global $\sigma_k^{(g)}$	$1+\frac{N}{2\lambda}$
p_{succ}^{target}	coefficient de comparaison avec le taux de succès moyenné	$\frac{1}{5+\sqrt{\lambda}/2}$
c_p	taux d'apprentissage ou temps d'accumulation pour le taux de succès moyenné	$\frac{p_{succ}^{target}\lambda}{2+p_{succ}^{target}\lambda}$
$\sigma^{(0)}$	valeur initiale de l'écart-type global (si l'on a normalisé les paramètres dans l'espace des solutions $[0,1]$)	0.6
Matrice de covariance : $C_k^{(g)}$		
c_c	taux d'apprentissage pour le chemin d'évolution	$\frac{2}{N+2}$
c_{coc}	taux d'apprentissage pour la matrice de covariance	$\frac{2}{N^2+6}$
p_{thresh}	coefficient de comparaison avec le taux de succès moyenné	0.44

Le tracé du domaine de confiance présenté en annexe G nous permet d'illustrer l'adaptation de la matrice de covariance au cours de l'exécution du MO-CMA-ES. En effet, à chaque itération le tirage de chaque point a 95% de chances de s'effectuer à l'intérieur du domaine de confiance. Autrement dit, à chaque itération, "presque" tous les points sont "emprisonnés" au sein de cet ensemble. Le domaine évolue d'itération en itération en accord avec l'évolution de l'écart-type global et de la matrice de covariance.

Les paramètres λ_{MO} et λ représentant le nombre d'individus de la population et le

nombre d'enfants générés par ces individus permettent d'explorer plus ou moins l'espace des solutions. Ainsi, en augmentant ces paramètres, le front de Pareto optimal sera atteint plus souvent, mais nous pouvons noter également que la complexité de l'algorithme augmente avec ces paramètres.

Le paramètre c_{cov} est le taux d'apprentissage pour la matrice de covariance. Pour de faibles valeurs du paramètre, le minimum global est atteint plus souvent mais le nombre d'évaluations des fonctions coût augmente.

L'écart-type global contrôle la "largeur" de la distribution de mutation, centrée sur le parent, et de la matrice de covariance $\sigma_k^{(g)^2} C_k^{(g)}$; alors que la matrice de covariance oriente les axes de recherche de la distribution de mutation vers le voisinage de solutions de Pareto.

Les paramètres que nous pouvons modifier afin d'accélerer l'algorithme sont la valeur initiale de l'écart-type global $\sigma^{(0)}$ ainsi que la population initiale $x^{(0)}$. La population initiale dépend directement de l'écart-type global initial car $x^{(0)} \sim \mathcal{N}(y, \sigma^{(0)} Id)$, où y est le centre de l'espace des solutions. Si l'écart-type global initial $\sigma^{(0)}$ est grand, le domaine de confiance est grand. Autrement dit, la région de l'espace dans laquelle chaque tirage a 95% de chances de s'effectuer est grande. La valeur par défaut de $\sigma^{(0)}$, si l'on a normalisé les paramètres dans l'espace des solutions $[0, 1]$, est de 0.6.

Dans le paragraphe suivant, le cadre des tests est présenté, particulièrement le nombre d'exécutions et le critère d'arrêt utilisés par la méthode MO-CMA-ES ainsi que l'évaluation de la performance d'un algorithme multi-objectifs.

4 Cadre des tests

4.1 Nombre d'exécutions et critère d'arrêt

Un algorithme évolutionnaire fait intervenir des tirages aléatoires, le comportement de l'algorithme varie donc d'une exécution à l'autre. En particulier, la convergence vers le front de Pareto au cours d'une exécution ne garantit pas que cette convergence sera obtenue lors d'une autre exécution. Pour tester le MO-CMA-ES, on ne peut donc se contenter d'une seule exécution. Nous allons effectuer 100 minimisations successives pour chacune des paires de fonctions testées.

En optimisation multi-objectifs globale, le critère d'arrêt de l'algorithme est généralement le nombre maximal d'évaluations de la fonction coût.

4.2 Evaluation de la performance d'un algorithme d'optimisation multi-objectifs

L'évaluation de la performance d'un algorithme d'optimisation multi-objectifs est plus difficile que celle d'un algorithme mono-objectif. De nombreuses méthodes de mesures de cette performance ont été proposées. Soient deux ensembles de solutions $A, B \subseteq X$. A est meilleur que B (noté $A \rhd B$ si pour chaque élément $a \in B$ il existe un élément $a' \in A$ qui

n'est pas plus "mauvais" que a selon chaque objectif, $\forall m \in \{1, ...M\}, \forall a \in B, \exists a' \in A$: $f_m(a') \leq f_m(a)$, et $ndom(A) \neq ndom(B)$. Autrement, nous avons $A \not\rhd B$. Régulièrement, pour deux ensembles A et B, nous n'avons ni $A \rhd B$ ni $B \rhd A$, c'est pourquoi il est nécessaire d'introduire un indicateur de qualité.

Nous présentons l'indicateur de l'hypervolume [186] qui représente la taille de l'espace des objectifs dominés par l'ensemble des solutions non dominées (voir la figure 6.3). Dans le cas d'un problème à deux dimensions, chaque solution Pareto optimal x couvre l'aire d'un rectangle défini par un point de référence et $(f_1(x), f_2(x))$. Le point de référence a_{ref} peut être soit les plus mauvaises valeurs des fonctions objectifs selon chaque composante, soit un point dominé par tous les points du front. L'union des rectangles couverte par les solutions du front de Pareto constitue l'espace totalement dominé et est utilisée comme mesure. Donc l'ensemble des solutions Pareto optimales qui a le plus grand indicateur de l'hypervolume couvre une plus grande partie de l'espace des objectifs, cet ensemble est par conséquent plus intéressant. Ce concept peut être étendu aux dimensions supérieures. Soit A un ensemble de solutions de Pareto, l'indicateur de l'hypervolume est : $\mathcal{I}_{S,A} = S_{a_{ref}}(A)$. L'avantage de cette mesure est que chaque front de Pareto peut être évalué indépendemment d'un autre. De plus, les régions convexes seront préférées aux régions concaves.

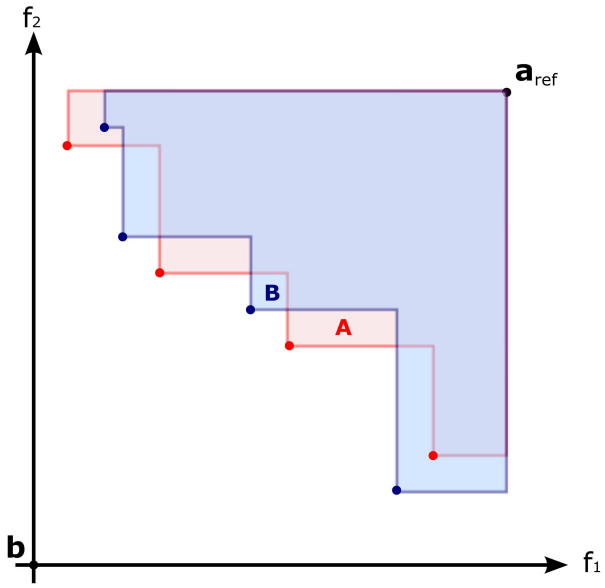


FIGURE 6.3 – Indicateur de l'hypervolume en dimension 2. Nous n'avons ni $A \rhd B$, ni $B \rhd A$, en revanche l'indicateur de l'hypervolume indique $\mathcal{I}_{S,A} = 0.41$ et $\mathcal{I}_{S,B} = 0.42$

Pour utiliser cet indicateur, il est utile de normaliser les fonctions objectifs (par exemple pour qu'elles varient sur $[0, 1]$) et de choisir la valeur de a_{ref}. L'indicateur de l'hypervolume est ensuite normalisé par $S_{a_{ref}}(b)$, où b est un point qui domine l'ensemble des

points du front A. Pour pouvoir comparer différentes performances, le point de référence a_{ref} ainsi que le point b (nous avons choisi $a_{ref} = 2.1$ et $b = 0$ selon chaque objectif normalisé) qui domine l'ensemble des points du front ont l'obligation d'être identiques pour chacune des évaluations. L'indicateur de l'hypervolume représente donc le pourcentage de l'aire (en deux dimensions) du rectangle de diagonale a_{ref} b dominé par le front de Pareto. Sur la figure 6.3, l'aire dominée par le front A est inférieure à l'aire dominée par le front B ($\mathcal{I}_{S,A} = 0.41$ et $\mathcal{I}_{S,B} = 0.42$).

Chapitre 7

Application en calibration des moteurs

De nos jours, les constructeurs automobiles sont confrontés pour la mise au point des moteurs à des contraintes de plus en plus fortes : consommer le moins possible, respecter les normes de pollution en vigueur tout en répondant aux demandes du conducteur en termes d'agrément de conduite et de puissance.

Avec le durcissement des normes et l'apparition de moteurs de plus en plus complexes, le nombre de paramètres à régler a nettement augmenté, rendant impossible la calibration empirique par balayage des valeurs des différents paramètres aux bancs d'essai. Les réponses du moteur sont maintenant modélisées à partir d'essais aux bancs moteur puis optimisées suivant les différents objectifs. Pour atteindre simultanément ces objectifs souvent antagonistes, le motoriste est intéressé par différents compromis afin de choisir au mieux les réglages du moteur.

Dans ce chapitre, nous sommes intéressés par une approche multi-objectifs de ce problème d'optimisation pour obtenir les différents compromis entre les objectifs : consommation en carburant, émission de polluants et bruit. Nous avons choisi d'utiliser l'algorithme évolutionnaire Multi-Objective - Covariance Matrix Adaptation - Evolution Strategy (MO-CMA-ES de Hansen, présenté dans le chapitre précédent).

Dans une première partie, le principe de la calibration des moteurs est présenté ainsi que les problèmes de modélisation et d'optimisation associés. Le problème considéré sera adapté dans une seconde partie afin de permettre l'application de la méthode d'optimisation MO-CMA-ES sur le problème réel de la calibration moteur. Enfin, le moyen de prendre en compte les contraintes dans la méthode MO-CMA-ES est explicité. Finalement, la méthode MO-CMA-ES est appliquée sur le problème réel de calibration moteur.

1 Formulation du problème de calibration

La mise au point des moteurs pour chaque véhicule nécessite la mise en œuvre d'un processus complexe et coûteux [154].

1.1 La mise au point des moteurs

Depuis l'apparition vers la fin des années 80 des normes de pollution limitant les émissions de certains polluants, le motoriste se doit de construire des moteurs respectant ces normes, tout en continuant à en optimiser le rendement. Ce processus implique la calibration de NV paramètres appelés réglages par la suite. Le travail réalisé a porté sur la calibration de six réglages qui influent sur la consommation et la quantité de polluants que le moteur émet en chaque point de fonctionnement considéré :

- l'avance à l'injection (AvPr) permet de fixer la phase entre le début de l'injection de carburant et la position haute du piston dans le cylindre.
- la pression à l'injection (Prail) règle la pression du carburant, lors de son injection dans le cylindre.
- les réglages de la pré-injection (débit : Tipil et phase : Qpil) permettent de faire débuter la combustion de manière plus lente, en injectant une petite quantité de carburant avant l'injection principale.
- le taux de recirculation des gaz (Qair) permet de définir le débit de gaz d'échappement à ré-injecter dans le cylindre.
- le réglage du turbo à géométrie variable (Psural) permet de réguler la quantité d'air injectée dans le cylindre.

1.1.1 Les objectifs

La calibration d'un moteur poursuit plusieurs objectifs, souvent antagonistes.

1.1.1.1 Minimiser la consommation

Pour des raisons évidentes, minimiser la consommation fait partie des premières exigences du client. Il faut également noter que les normes de pollution sont durcies régulièrement ce qui entraîne des contraintes de plus en plus fortes sur la consommation des véhicules afin de limiter les émissions de dioxyde de carbone qui ont un impact sur l'effet de serre.

1.1.1.2 Normes de pollution

Bien que les effets néfastes de la pollution automobile soient connus depuis longtemps, ce n'est que récemment que des normes contraignantes ont été imposées. Ces normes portent d'abord sur les polluants émis par le moteur lors de parcours prédéfinis. En Europe, il existe cinq polluants concernés par ces normes : les oxydes de carbone (CO), les hydrocarbures imbrûlés (HC), les oxydes d'azote (NOx), les particules (Part.) et les Fumées (Fumée) (voir tableau 7.1). Les oxydes de carbone sont dus à une oxydation incomplète du carburant durant la combustion et contribuent à l'augmentation de l'effet de serre. Les hydrocarbures imbrûlés représentent tous les composés du carburant qui ne sont pas totalement éliminés lors de la combustion. Les oxydes d'azote comprennent le monoxyde et le dioxyde d'azote. Ils interviennent, avec les hydrocarbures imbrûlés, dans la création de produits indésirables tels que l'ozone. Les particules comprennent les suies

et les matières minérales présentes à l'échappement, qui peuvent devenir toxiques par leur accumulation et leur taille. Les fumées représentent une des formes visibles des particules.

TABLE 7.1 – Normes de l'Union européenne pour les émissions des moteurs diesel lourds, g/kWh (fumée en m^{-1})

	CO	HC	NOx	Part.	Fumée
Euro 1 (1992)	4.5	1.1	8.0	0.612	
Euro 2 (1996)	4.0	1.1	7.0	0.25	
Euro 3 (2000)	2.1	0.66	5.0	0.10	0.8
Euro 4 (2005)	1.5	0.46	3.5	0.02	0.5
Euro 5 (2008)	1.5	0.46	2.0	0.02	0.5

Afin que le motoriste contrôle les émissions d'un véhicule à l'autre, il est nécessaire d'imposer un protocole reproductible. Ce protocole détermine des cycles de conduite représentant un parcours standard durant lequel la vitesse et l'accélération du véhicule sont fixées, pour une boîte de vitesse manuelle et des changements de rapport fixés (voir figure ci-dessous 7.1). Ce cycle simule une utilisation caractéristique d'un véhicule sur route plane et permet de mesurer les polluants émis lors du parcours.

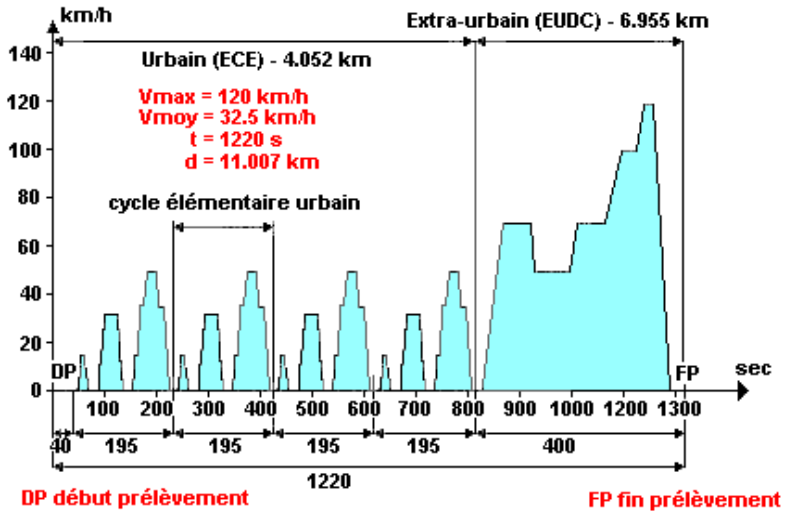


FIGURE 7.1 – Exemple d'un cycle de conduite (cycle européen)

Les normes à respecter permettent de définir des contraintes sur certaines des émissions polluantes du moteur. Des contraintes supplémentaires sont prises en compte par le motoriste.

1.1.1.3 Autres contraintes

Pour l'agrément de conduite, des contraintes supplémentaires sont prises en compte par

le motoriste. Une contrainte garantissant le confort des occupants du véhicule est de limiter le bruit produit par le moteur. Afin de diminuer les à-coups lors de la conduite, les variations de réglages sont limitées.
Plusieurs objectifs sont donc poursuivis par la mise au point des moteurs, minimiser la consommation, respecter les normes de pollution, limiter le bruit produit par le moteur et les variations de réglages. Afin d'y répondre, le motoriste dispose de plusieurs moyens d'action.

1.1.2 Les moyens d'action

La poursuite de ces objectifs par le motoriste implique la calibration des réglages à sa disposition. Afin de régler ces paramètres, le motoriste dispose de plusieurs moyens d'essais, appelés bancs moteur, lui permettant de mesurer les réponses d'un moteur, telles que sa consommation en carburant, les émissions de polluants, le bruit généré...

1.1.2.1 Plan régime/charge

Le régime (tours/minute) représente la vitesse à laquelle tourne l'arbre moteur, et la charge (bar) représente la force (ou couple) que le moteur exerce sur l'arbre, autrement dit la force pour entraîner le véhicule. Les points du plan régime/charge sont appelés points de fonctionnement du moteur. Le plan régime/charge illustre donc la demande en vitesse et en couple que doit fournir le moteur. Pour un type de véhicule donné (poids du véhicule, pneumatiques, boîte de vitesse,...), il est possible de transformer un cycle de conduite associé aux normes de pollution, en une trajectoire dans le plan régime/charge.

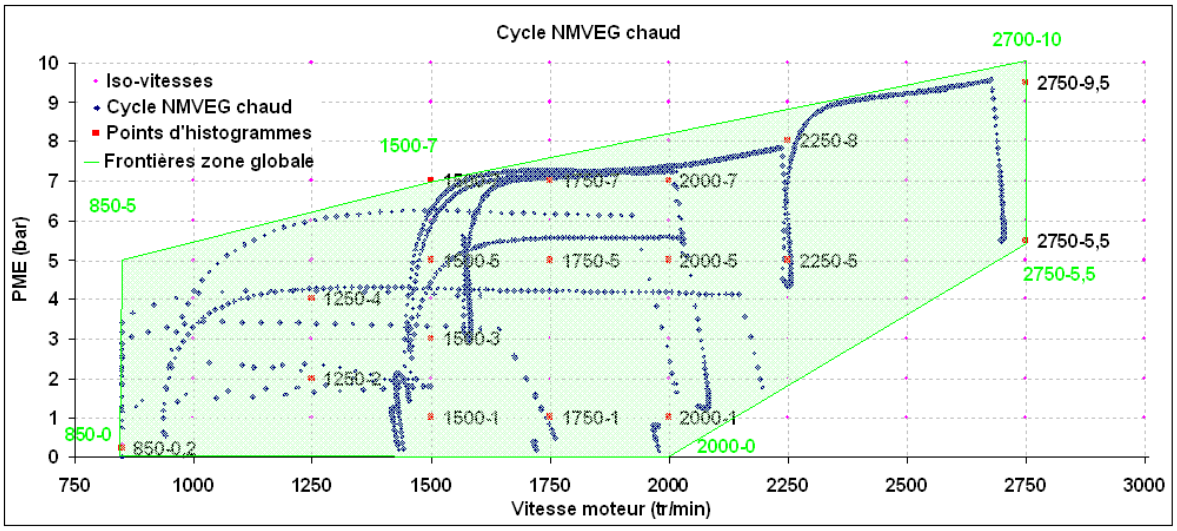


FIGURE 7.2 – Représentation du cycle de conduite européen discrétisé dans le plan régime / charge

Plus formellement pour un type de véhicule donné, il existe un modèle physique noté N permettant de transformer une courbe de vitesse en fonction du temps en une trajectoire dans le plan régime/charge (voir équation 7.1).

$$\begin{aligned} N : [0,T] &\xrightarrow{N} Dom \\ t &\longrightarrow N(t) = (r(t), c(t)) \end{aligned} \tag{7.1}$$

où T (exprimé en secondes) désigne la durée du cycle de vitesse, Dom le domaine de fonctionnement du moteur, et la courbe paramétrée $N(t)_{0 \leq t \leq T}$, la trajectoire dans le plan régime/charge. La figure 7.2 représente le cycle de conduite européen pour un véhicule donné dans le plan régime/charge. Les points représentent le domaine de fonctionnement discrétisé du moteur ($16 \cdot 16 = 256$ points au total).

Le plan régime/charge est utilisé pour définir les cartographies associées aux réglages du contrôle moteur.

1.1.2.2 Cartographies

Le plan régime/charge permet de définir les cartographies. La figure 7.3 illustre une cartographie correspondant au réglage de la pression d'injection dans un moteur diesel.

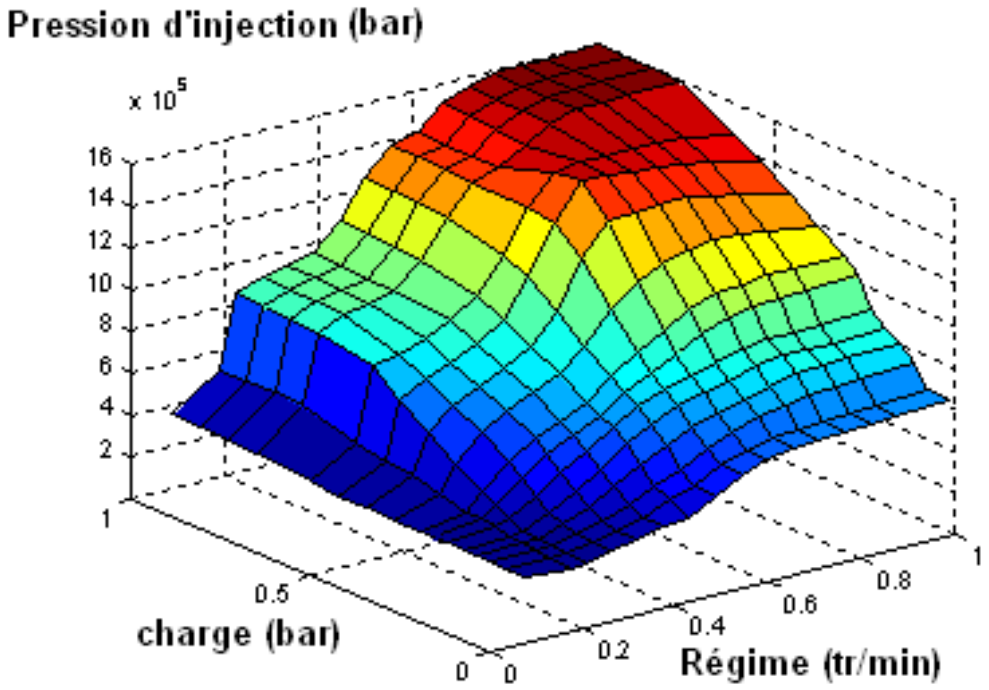


FIGURE 7.3 – Exemple de cartographie associée au réglage de la pression d'injection dans un moteur diesel

En effet, les valeurs d'une même variable de contrôle moteur diffèrent suivant les points de fonctionnement, le moteur ne devant pas y fournir un travail de même nature. Ainsi, à chaque réglage ν ($\nu = 1, ..., NV$) du contrôle moteur (ex. : pression à l'injection), est associée une fonction précisant les valeurs prises par le réglage sur le domaine de fonctionnement du moteur. Cette fonction est appelée cartographie et est notée $cart^{\nu}$.

$$\begin{aligned} cart^{\nu} : Dom &\longrightarrow \mathbb{R} \\ (r, c) &\longrightarrow cart^{\nu}(r, c). \end{aligned} \tag{7.2}$$

1.1.2.3 Moyens d'essai

Les réglages influent sur la consommation et la quantité de polluants que le moteur émet en chaque point de fonctionnement. Pour une certaine valeur des réglages en un certain point de fonctionnement, sont mesurés au banc moteur la consommation en carburant, les émissions de polluant ainsi que le bruit généré par le moteur (voir figure 7.4). Ce test expérimental est appelé essai. Il est à noter que les essais sont coûteux et, par conséquent, le nombre de ceux-ci est en général limité.

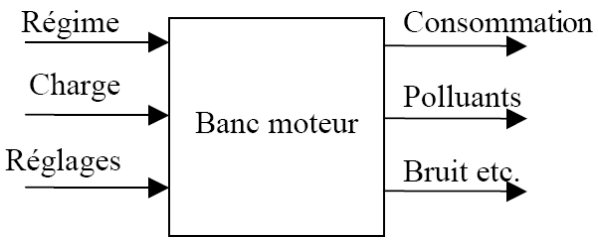


FIGURE 7.4 – Schéma de banc moteur

1.1.3 Le problème d'optimisation des cartographies

Nous considérons le problème d'optimisation (P) qui consiste à optimiser les réglages du moteur par rapport aux réponses de celui-ci tels que la consommation en carburant, les émissions de polluants et le bruit généré par le moteur.

Soient $f_1, ..., f_M$, M objectifs,

$$\begin{aligned} f_1, ..., f_M : X = &\ (Dom, F(Dom \longrightarrow \mathbb{R})^{NV}) \longrightarrow \mathbb{R} \\ x = &(N(t), cart^1(.), ..., cart^{NV}(.)) \longrightarrow f_1(x), ..., f_M(x) \end{aligned}$$

définis par : $f_i(x) = \int_0^T F_i(N(t), cart^1(N(t)), ..., cart^{NV}(N(t)))dt$, où T désigne la durée du cycle de conduite, $F(Dom \longrightarrow \mathbb{R})^{NV}$ l'ensemble des applications définies sur le domaine de fonctionnement du moteur et à valeurs dans $\mathbb{R}$ et f le vecteur des réponses du moteur dont les expressions analytiques ne sont pas connues, et dont seulement un certain nombre de valeurs sont obtenues au travers de mesures au banc moteur pour des réglages fixés.

On considère le problème d'optimisation (P) :

$$(P) \left\{ \min f(x) = (f_1(x), ..., f_M(x)) \right. \tag{7.3}$$

La méthode utilisée pour résoudre le problème d'optimisation consiste, à partir de mesures obtenues au banc moteur, à construire des modèles approchés globaux des réponses du moteur, c'est-à-dire des modèles définis sur le domaine de fonctionnement du moteur, capables de prédire le comportement du moteur pour différentes valeurs des réglages. Ces modèles seront ensuite utilisés pour l'optimisation des réglages.

1.1.3.1 Discrétisation de la trajectoire $N(t)$ (cycle de vitesse)

Afin de résoudre numériquement les calculs des intégrales déterminant la fonction f à optimiser, la trajectoire $N(t)$ dans le plan régime/charge est discrétisée en l'ensemble $\{(r_p, c_p)\}_{1 \leq p \leq NP}$.

Notons que les variables d'optimisation sont des fonctions définies sur le domaine de fonctionnement du moteur puisque nous cherchons à optimiser les cartographies associées aux réglages. Après discrétisation, les variables d'optimisation forment l'ensemble suivant $\{cart^{\nu}(r_p, c_p), 1 \leq \nu \leq NV, 0 \leq p \leq NP\}$. Le nombre de variables est égal au produit entre le nombre de réglages (NV) et le nombre de points représentant le cycle (NP). L'intégrale déterminant la fonction objectif est discrétisée en une somme pondérée de termes représentant chacun la consommation en carburant au voisinage de chaque point de fonctionnement. Les coefficients $\{\alpha_p\}_{1 \leq p \leq NP}$ sont les pondérations temporelles des points de fonctionnement choisis pour la discrétisation de la trajectoire. Le problème d'optimisation peut donc se réécrire sous la forme suivante :

$$\begin{cases} \min f(x) = (f_1(x), ..., f_M(x)) \\ f_i(x) = \displaystyle\sum_{p=1}^{NP} \alpha_{i_p} \cdot F_{i_p}(cart^1(N_p), ..., cart^{NV}(N_p)) \end{cases} \tag{7.4}$$

où F_{i_p} désigne le modèle local de la ième réponse du moteur f au point de fonctionnement $N_p = (r_p, c_p)$ (NP est limité à seize pour le cycle associé aux normes de dépollution européennes). Ici, contrairement à l'approche de la section 2.2 du chapitre 4, nous considérons donc des modèles locaux à la place des modèles globaux.

1.1.3.2 Modélisation des réponses du moteur considérées

On appelle surface de réponse définie sur un domaine de variations des réglages, la « réponse » du moteur calculée à partir de mesures au banc moteur.

$$y_{i_p} = f_i(N_p, cart^1(N_p), ..., cart^{NV}(N_p)), \forall p = 1, ...NP \tag{7.5}$$

où N_p représente le point de fonctionnement (r_p, c_p), et y_{i_p} la ième réponse du moteur en N_p, pour le vecteur des réglages $\big(cart^1(N_p), ..., cart^{NV}(N_p)\big)$. La fonction f_i, donnant un approximation de la mesure effectuée au banc moteur au point N_p pour le vecteur des

réglages $\left(cart^1(N_p), ..., cart^{NV}(N_p)\right)$ pour $p = 1, ...NP$ constitue la surface de réponse du moteur.

Après une phase préliminaire où les domaines de variation des réglages sont déterminés, la méthode de modélisation utilisée se décompose selon les étapes suivantes :

(1) planification d'expériences (voir [23, 24])

Au cours de cette étape, il s'agit de répartir « au mieux » les valeurs des réglages choisies pour réaliser les essais pour un nombre fixé d'essais. Afin de construire à partir de ces données le modèle le plus précis, une technique classique est le plan latin hypercube consistant à choisir au mieux les valeurs de réglages de façon à explorer uniformément l'ensemble des valeurs admissibles par les réglages (plan de type "space filling"). Une autre technique utilisée est le plan D-optimal pour laquelle une forme de modèle est postulée a priori.

(2) mesure des essais sur banc moteur

Les essais définis à l'étape précédente sont effectués au banc moteur.

(3) identification de modèles paramétriques

En chaque point de fonctionnement, un modèle paramétrique pour chaque réponse du moteur considérée est déterminé à partir des mesures permettant de prédire localement au point de fonctionnement la réponse du moteur.

La notation suivante est employée pour désigner les modèles paramétriques locaux :

$$\begin{array}{rcl} F_{i_p} : Dom_{dis} & \longrightarrow & \mathbb{R}^{NV} \\ N_p & \longrightarrow & f_p(cart^1(x_p), ..., cart^{NV}(x_p)) \end{array} \tag{7.6}$$

où Dom_{dis} désigne l'ensemble des points de la trajectoire du cycle discrétisée, et f_p le modèle paramétrique local associé au point de fonctionnement N_p.

Les réponses du moteur sont, dans la version actuelle, modélisées par des polynômes de degré 2 ou 3.

Ensuite, une optimisation sur ces modèles est effectuée : cette étape consiste à optimiser les réglages des moteurs aux points de fonctionnement choisis $\{N_p\}_{1 \leq p \leq NP}$, selon les objectifs définis, à partir des réponses modélisées à l'étape 3.

Nous pouvons noter que le nombre de variables d'optimisation dépend de NP, nombre de points de discrétisation du cycle dans le plan régime/charge. Or, pour obtenir une bonne approximation de l'intégrale, il faudrait choisir NP élevé, ce qui accroît le nombre de variables à optimiser. Pour pallier cet inconvénient, les cartographies sont également approchées par des modèles paramétriques.

1.1.3.3 Modélisation des cartographies par des modèles Chaque cartographie $cart^\nu$ (pour $\nu = 1, ..., NV$) est approchée par un modèle. Ce modèle peut être de type LOLIMOT (Local Linear Model Tree [154, 122, 111, 112]) ou par des modèles bilinéaires, comme dans le chapitre 4 et [67]. Ces modèles sont caractérisés par un nombre important de paramètres ($150 - 300$ paramètres pour les LOLIMOT, 20 paramètres pour les modèles bilinéaires).

1.1.3.4 Problème d'optimisation étudié

Le problème d'optimisation des cartographies étant complexe (nombre de variables de l'ordre de 1000, pour $NV = 6$ cartographies LOLIMOT de $150 - 300$ paramètres chacune), l'optimisation multi-objectifs sera effectuée indépendamment en chaque point de fonctionnement choisi : $\{N_p\}_{1 \leq p \leq NP}$. Le problème de minimisation traité est alors le suivant :

$$\begin{cases} \min f_p(x) = (f_{1_p}(x), ..., f_{M_p}(x)) \\ f_{i_p}(x) = F_{i_p}(cart^1(N_p), ..., cart^{NV}(N_p)) \end{cases} \tag{7.7}$$

où F_{i_p} désigne le modèle local de la ième réponse du moteur f au point de fonctionnement $N_p = (r_p, c_p)$ et $(cart^1(N_p), ..., cart^{NV}(N_p))$ sont les réglages au point de fonctionnement N_p étudié. Le nombre de variables d'optimisation est égal à NV.

2 Prise en compte des contraintes dans MOCMAES

Quelques notions permettant la prise en compte de contraintes pour les algorithmes évolutionnaires sont présentées dans [119] et dans [151] pour l'optimisation multi-objectifs. Une pénalisation adaptative pour les individus infaisables et rejetant seulement les individus infaisables éloignés des bornes [19] pourrait également être testée. Tout comme un algorithme génétique [57] lié à une technique Genocop III (Genetic Algorithm for Numerical Optimization of Constrained Problems [120]) utilisant une technique de coévolution permettant de faire évoluer deux populations parallèlement (une population comprenant les individus qui vérifient les contraintes linéaires et l'autre comprenant les individus qui vérifient toutes les contraintes).

Afin de prendre en compte les contraintes, nous avons pensé à pénaliser les fonctions objectifs suivant les valeurs de non-respect de chaque contrainte.

$$\begin{cases} \min f(x) \\ g(x) \leq b \end{cases} \Longleftrightarrow \begin{cases} \min \Phi_\lambda(x) & \text{avec } \Phi_\lambda(x) = f(x) + \lambda p(x) \\ & p(x) = \begin{cases} 0 & \text{si } g(x) \leq b \\ r(x) & \text{si } g(x) > b \end{cases} \end{cases} \tag{7.8}$$

où λ est le poids de pénalisation et $r(x) > 0$ est une fonction de pénalisation. Les fonctions $f(x)$, $\Phi_\lambda(x)$, $p(x)$ et $r(x)$ sont à valeurs vectorielles.

La notion de domination-contrainte a été introduite pour enrichir la notion de dominance dans les problèmes multi-objectifs contraints. La plupart des méthodes de résolution de problèmes mono-objectif utilisent une approche basée sur l'attribution de pénalités en fonction des contraintes violées. Mais cette approche a deux inconvénients :

- elle exige de fixer les règles de pénalité,
- elle ne fait pas la différence entre deux individus dont l'un viole des contraintes et l'autre pas si le premier réalise une diminution importante de l'objectif par rapport au second.

Une première définition de la domination-contrainte a été donnée par Jiménez et Verdegay [97]. Ils utilisent un tournoi à deux individus i et j. Si i donne une solution admissible et j donne une solution non admissible, alors i est choisi. Si les deux individus donnent

des solutions admissibles alors ils sont comparés avec un sous-ensemble d'individus choisis aléatoirement. Si les individus donnent des solutions non admissibles, alors ils sont comparés à un ensemble d'individus du même type. Le meilleur des deux, ou celui qui est situé dans la plus petite niche, sera sélectionné.

Une autre définition a été proposée par Deb, Pratap et Meyarivan [42]. Une solution i domine avec contrainte une solution j si un des cas suivants est réalisé :

- la solution i est admissible et la solution j ne l'est pas.
- les deux solutions i et j ne sont pas admissibles mais i a le plus petit total de niveau de violations de contrainte.
- les deux solutions sont admissibles mais i domine j.

L'utilisation de cette notion exige que l'on puisse mesurer le niveau de violation de chaque contrainte du problème.

Cette définition favorise énormément les solutions vérifiant toutes les contraintes (surtout lorsque le nombre de contraintes est assez grand), c'est pourquoi nous avons pensé à la remplacer par une pénalisation non continue permettant à terme de n'obtenir que des solutions vérifiant toutes les contraintes.

Dans le rapport [109], les trois fonctions de pénalisation suivantes ont été comparées : les pénalisations continues par les fonctions $r(x) = x^2$ et $r(x) = |x|$ des solutions ne vérifiant pas les contraintes (où le poids de pénalisation λ est de 100) et une pénalisation non continue (une pénalité de α est ajoutée au vecteur des fonctions objectifs pour chacune des contraintes non vérifiées). On a alors le problème : $\phi_\alpha(x) = f(x) + \sum_{i=1}^{NC} \alpha \times \mathbb{1}_i$, où NC est le nombre de contraintes du problème et $\mathbb{1}_i$ vaut 1 si la i-ème contrainte n'est pas vérifiée. On fixe le paramètre de pénalisation α à 100 dans le cadre de ces tests. Ces comparaisons permettent de choisir la bonne technique de pénalisation pour certains types de contraintes pour l'application réelle.

Ces comparaisons ont montré que l'utilisation d'une pénalisation non continue permet d'obtenir des résultats tout aussi satisfaisants que l'utilisation de la pénalité égale à la valeur absolue de l'écart à la contrainte. Mais la pénalisation non continue a l'avantage de simplifier les réglages de ce poids de pénalisation. En effet, la pénalisation par l'écart aux contraintes nécessite le réglage d'un poids de pénalisation pour chacune des contraintes. C'est pourquoi, nous avons utilisé ces deux modes de prise en compte des contraintes.

Dans le paragraphe suivant, l'algorithme MO-CMA-ES est appliqué sur un cas réel de la calibration moteur en deux points de fonctionnement.

3 Application du MO-CMA-ES sur un cas réel de calibration de moteur

3.1 Présentation des résultats

Pour un nombre donné d'évaluations de la fonction, un tableau donne pour chaque couple de fonctions soumis à des contraintes testé, la meilleure valeur d'indicateur de

l'hypervolume obtenue.

En revanche, pour tester l'efficacité de la prise en compte des contraintes, on notera le nombre moyen d'individus de la population des λ_{MO} individus de la dernière itération de l'algorithme ne vérifiant pas les contraintes. Bien entendu, si les individus ne vérifient pas chacune des contraintes, l'indicateur de l'hypervolume obtenu n'est pas représentatif de la performance de l'algorithme.

3.2 Reformulation du problème

La mesure de l'hypervolume ayant été jugée plus efficace que la distance d'encombrement par Hansen [85], nous utiliserons la relation $\prec_{s,Q^{(g)}}$ pour sélectionner les individus lors de chaque génération.

De plus, pour un nombre d'objectifs supérieur à deux, la complexité de calcul de cet hypervolume, ainsi que le tri des individus de la population limitent l'application à la minimisation de deux fonctions objectifs. Le problème formulé dans la section précédente est soumis à de nombreuses contraintes linéaires ou non linéaires.

L'optimisation multi-objectifs sera effectuée en chaque point de fonctionnement choisi : $\{N_p\}_{1\leq p\leq NP}$. Soit $V_p = (cart^1(N_p), ..., cart^{NV}(N_p)) \in \mathbb{R}^{NV}$ le vecteur des réglages au point de fonctionnement N_p étudié. Le problème de minimisation traité est donc :

$$\min f_p(V_p) = (f_{1_p}(V_p), f_{2_p}(V_p))$$

sous les contraintes :

$$\begin{cases} p_{min} \leq V_p \leq p_{max} \\ A \cdot V_p \leq b \\ \begin{pmatrix} f_{1_p}(V_p) \\ \vdots \\ f_{M_p}(V_p) \end{pmatrix} < \begin{pmatrix} c_{1_p} \\ \vdots \\ c_{M_p} \end{pmatrix} \end{cases} \tag{7.9}$$

où $f_{i_p}(V_p) = F_{i_p}(V_p)$. Le nombre de variables d'optimisation est égal à NV (voir section 1.1 et 1.1.3.4). Nous cherchons à minimiser simultanément deux objectifs f_1 et f_2 mais le nombre d'objectifs est souvent supérieur à M. Les $M-2$ autres objectifs sont traités comme des contraintes à l'aide de la méthode des ϵ-contraintes (voir section 2.1.3 du chapitre 5). Ces contraintes doivent être inférieures à des seuils fixés par la législation. Les contraintes sur les 2 objectifs poursuivis nous permettent de n'obtenir que les compromis qui respectent ces normes.

Nous avons trois types de contraintes :

- des contraintes avec dérivées sur les paramètres :
  - contraintes de bornes : $p_{min} \leq V_p \leq p_{max}$, où p_{min} et p_{max} sont des vecteurs de taille NV qui délimitent l'espace des paramètres.
  - contraintes linéaires : $A \cdot V_p \leq b$ qui sont des contraintes provenant du fonctionnement du moteur, définissent des bornes sur les variations d'un paramètre

dépendantes des autres paramètres. A est une matrice $N_{CL} \times NV$ et b est un vecteur de taille N_{CL} (N_{CL} est le nombre de contraintes linéaires qui varie suivant le point de fonctionnement optimisé).

- des contraintes sans dérivées sur les réponses : des contraintes sur les réponses sont introduites, chaque réponse du moteur peut être contrainte à ne pas dépasser un seuil : $f_{i_p}(V_p) < c_{i_p} \in \mathbb{R}$, pour tout $i \in \{1, ..., M\}$. Ces contraintes non linéaires proviennent des objectifs que l'on ne peut minimiser, mais aussi des objectifs minimisés afin d'obtenir directement les compromis qui nous semblent intéressants.

Pour les points de fonctionnement étudiés, nous avons choisi d'optimiser suivant les deux objectifs à minimiser :

- oxydes d'azote (NOx),
- particules et fumées (IOFCL),

car les points de fonctionnement que nous avons étudiés sont dans la partie du plan régime/charge qui correspond au cycle de conduite européen en ville. En effet, en ville, la réduction de l'émission de ces deux polluants est prioritaire (au détriment d'autres objectifs comme la consommation). De plus, ces deux objectifs sont antagonistes, les compromis sont donc difficiles à obtenir.
Le problème de calibration moteur formulé précédemment est étudié en deux points de fonctionnement. Le critère d'arrêt pour chaque exécution est le nombre maximal d'évaluations de la fonction coût que nous avons fixé à 100×1500. La population de $\lambda_{MO} = 100$ individus générée par l'algorithme MO-CMA-ES évolue sur 1500 générations.

Nous présenterons nos résultats suivant la méthodologie présentée dans la section 3.1 précédente : la meilleure valeur de l'hypervolume obtenue ainsi que le nombre moyen d'individus ne vérifiant pas chacune des contraintes à l'itération finale de l'algorithme. Nous indiquerons également la moyenne et la variance de l'indicateur de l'hypervolume des générations finales.

Les solutions ne vérifiant pas les contraintes sont pénalisées :

- soit par la valeur absolue de l'écart à la contrainte : $\phi_\lambda(x) = f(x) + \sum_{i=1}^{NC} \lambda \, |r_i(x)|$, où $r_i(x)$ est l'écart à la i-ème contrainte et NC est le nombre de contraintes.
- Soit par l'ajout d'un terme $\alpha = 100$: $\phi_\alpha(x) = f(x) + \sum_{i=1}^{NC} \alpha \times \mathbb{1}_i$, où NC est le nombre de contraintes du problème et $\mathbb{1}_i$ vaut 1 si la i-ème contrainte n'est pas vérifiée.

Les résultats obtenus seront ensuite comparés avec une méthode d'optimisation mono-objectif locale utilisée classiquemement qui permet l'optimisation d'une fonction non linéaire sous contraintes linéaires et/ou non linéaires (fmincon : méthode SQP de la toolbox optimization de Mathworks). Ainsi, un tel algorithme peut donner une solution correspondant à un minimum local très différent de l'optimum global. Pour multiplier les chances d'obtenir ce minimum global, la méthode d'optimisation est lancée à partir de plusieurs jeux de paramètres initiaux. Cette méthode ne permet que la minimisation mono-objectif, en revanche le nombre de contraintes est quelconque. Pour minimiser plusieurs objectifs simultanément, nous avons recours à une somme pondérée des deux objectifs (une méthode d'agrégation des deux objectifs visés, voir section 2.1.1 du chapitre

5 sachant que les $M-2$ autres sont introduits sous forme de contraintes), les poids étant choisis en fonction de la variabilité des fonctions considérées sur le domaine de variation des paramètres. Ainsi, si la fonction objectif normalisée par sa valeur moyenne n°1 varie plus sur le domaine que la fonction objectif normalisée n°2, son poids associé sera plus fort, le gain de minimisation étant potentiellement plus fort pour cet objectif. Cette méthode n'est pas idéale, c'est pourquoi nous avons étudié une méthode dédiée à l'optimisation multi-objectifs qui devrait permettre de trouver plusieurs optima associés aux différents compromis entre les différentes fonctions objectifs.

La méthode actuelle a été utilisée pour obtenir l'optimum associé à différentes réponses (voir figure 7.1) :

- les émissions de fumées et de particules (IOFCL),
- les émissions d'oxydes d'azote (NO_x),
- les émissions de CO_2, proportionnelles à la consommation de carburant (CO_2),
- les émissions de fumées et d'oxydes d'azote (IOFCL+NO_x).

Dans le cas étudié, le nombre de paramètres est $NV = 6$ et le nombre d'objectifs est $M = 6$. Au premier point de fonctionnement étudié, le nombre de contraintes linéaires avec dérivées est $N_{CL} = 2$ alors qu'au second, $N_{CL} = 4$.

3.3 Optimisation aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar

Les solutions obtenues par MO-CMA-ES sont représentées dans l'espace des objectifs aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar (voir les figures 7.5 et 7.6). Ces solutions sont aussi intéressantes que les solutions obtenues par la méthode classique. Mais nous obtenons de nombreux autres compromis entre les objectifs visés. Nous pouvons également remarquer que l'optimum de la minimisation de la consommation de carburant ne correspond pas aux minima selon les émissions de particules et de fumées et de d'oxydes d'azote.

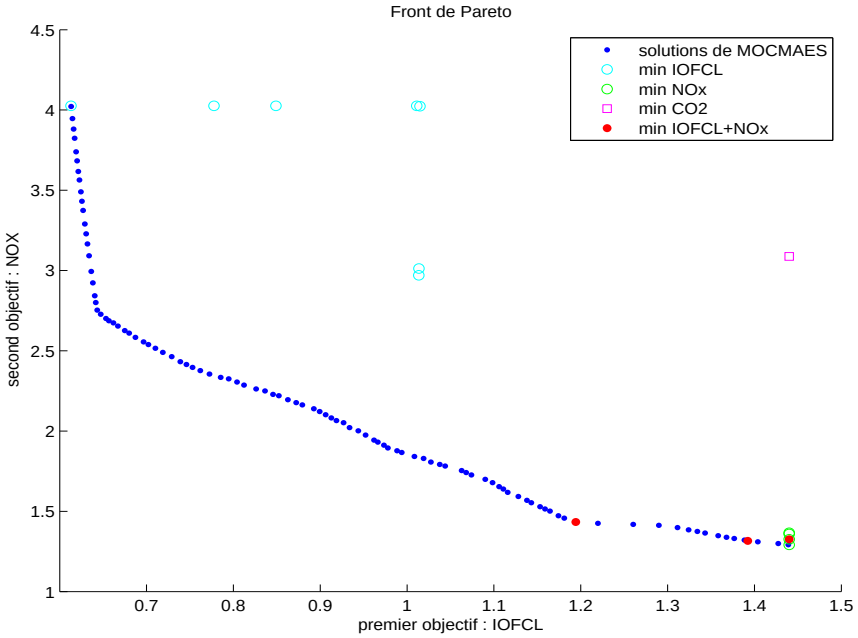


FIGURE 7.5 – Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 1 bar

Nous remarquons que la moyenne de la mesure de l'hypervolume est plus petite lorsque la pénalisation est non continue que lorsque la pénalisation est continue, c'est-à-dire que le front de Pareto obtenu est plus souvent meilleur (au sens de l'indice de l'hypervolume) avec la pénalisation continue (voir tableau 7.2). Nous avons également pu remarquer que les solutions dans l'espace des paramètres ne s'approchent pas autant des bornes lorsque la pénalisation est non continue que lorsqu'elle est continue. En revanche, la valeur maximale de l'indicateur de l'hypervolume est à peu près équivalente. La pénalisation non continue a l'avantage par rapport à la pénalisation continue de n'obtenir que des solutions vérifiant chacune des contraintes, mais davantage d'exécutions sont nécessaires pour avoir le meilleur front possible. Nous avons envisagé l'utilisation d'une fonction de pénalisation évolutive dépendante des générations : la pénalisation est effectuée de façon continue tant que la génération 1 000 n'est pas atteinte, puis les parents de la génération 1 000 sont évalués à nouveau et pénalisés de façon non continue, puis tous les individus sont pénalisés de façon non continue (nous aurions pu également faire varier les poids de pénalisation au cours des générations).

Les solutions ne vérifiant pas les contraintes sont donc pénalisées par

$$\phi_\lambda(x) = f(x) + \begin{cases} \sum_{i=1}^{NC} \lambda \left| r_i(x) \right| & \text{, si } g < 1000 \\ \sum_{i=1}^{NC} \alpha \times \mathbb{1}_i & \text{, si } g \geq 1000 \end{cases}$$

où NC est le nombre de contraintes du problème, g est la génération courante, $r_i(x)$ est

l'écart à la i-ème contrainte et $\mathbb{1}_i$ vaut 1 si la i-ème contrainte n'est pas vérifiée.

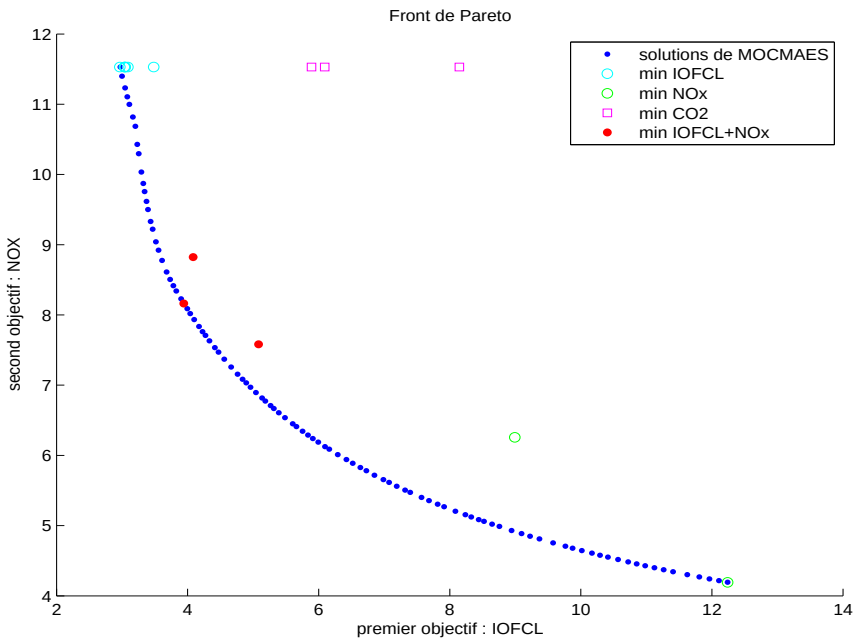


FIGURE 7.6 – Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 5 bar

TABLE 7.2 – Résultats de l'optimisation aux points de fonctionnement 1 750-1 et 1 750-5. La valeur maximale, moyenne et la variance de la mesure de l'hypervolume, ainsi que le nombre maximum d'individu ne vérifiant pas les contraintes sont indiqués pour la génération finale.

Cas réel		$\lambda = 1000\,\lvert x\rvert$	$\lambda = 10000\,\lvert x\rvert$	α	pénalisation évolutive
1 750-1	Maximum	0.81012 1	0.810473 0	0.809222 0	0.8105 0
	Moyenne Variance	0.800199 5.5×10^{-5}	0.802025 4.2×10^{-5}	0.799236 7.3×10^{-5}	0.8021 4.3×10^{-5}
1 750-5	Maximum	0.247136 1	0.247972 0	0.248075 0	0.2485 0
	Moyenne Variance	0.234677 4.1×10^{-4}	0.23793 2.1×10^{-4}	0.218187 8.8×10^{-4}	0.238 2.2×10^{-4}

La moyenne d'indicateur de l'hypervolume est plus grande pour la méthode de pénalisation évolutive, ce qui nous permet d'obtenir en général un meilleur front de Pareto pour chaque exécution et donc de réduire le nombre d'exécutions de l'algorithme. Pour

une meilleure utilisation des résultats de chacune des exécutions, λ_{MO} individus sont sélectionnés à l'aide du critère $\prec_s$ appliqué à l'union des populations finales de λ_{MO} individus des 100 exécutions.

3.4 Optimisation aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar avec robustesse

En plus des contraintes de bornes, des contraintes linéaires et non linéaires, des contraintes de robustesse ont été ajoutées par le motoriste afin d'avoir une marge d'erreur lors de la calibration moteur. La dispersion des réglages est ainsi prise en compte dans l'optimisation pour définir un voisinage autour de l'optimum, tous les points de ce voisinage devant rester dans le domaine défini par les contraintes décrites dans la partie précédente. Introduire la robustesse consiste à ajouter des contraintes un peu plus fortes sur les paramètres : en effet elle réduit l'espace des paramètres admissibles pour l'optimisation.

Table 7.3 – Résultats de l'optimisation aux points de fonctionnement 1 750-1 et 1 750-5 avec robustesse. La valeur maximale, moyenne et la variance de la mesure de l'hypervolume, ainsi que le nombre maximum d'individu ne vérifiant pas les contraintes sont indiqués pour la génération finale.

Cas réel		$\lambda = 1000\,\lvert x\rvert$	α	pénalisation évolutive
1 750-1 avec robustesse	Maximum	0.756326 1	0.753175 0	0.7521 0
	Moyenne Variance	0.726052 4.9×10^{-4}	0.71429 7×10^{-4}	0.7322 3.7×10^{-4}
1 750-5 avec robustesse	Maximum	0.187367 1	0.188855 0	0.1885 0
	Moyenne Variance	0.176629 2×10^{-4}	0.173207 3×10^{-4}	0.1806 9×10^{-5}

La méthode de pénalisation évolutive testée permet d'obtenir une meilleure moyenne d'indicateur de l'hypervolume que la pénalisation continue ou non continue (cf. tableau 7.3) ce qui nous confirme les conclusions précedentes.

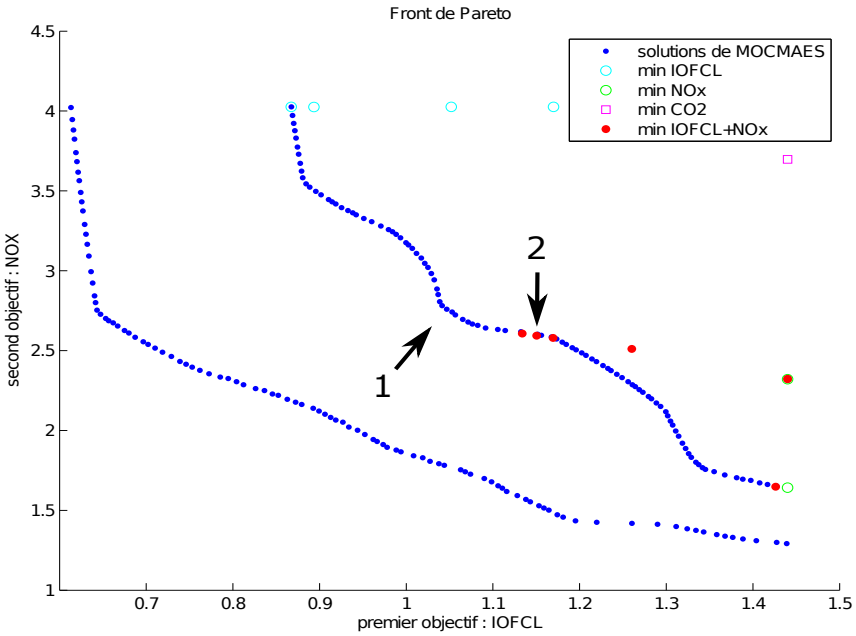


FIGURE 7.7 – Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 1 bar avec l'ajout de la contrainte de robustesse

Les solutions obtenues par MO-CMA-ES sont représentées dans l'espace des objectifs aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar avec robustesse (voir les figures 7.7 et 7.8). Nous obtenons également de nombreux compromis entre les objectifs visés que nous ne pouvions obtenir par la méthode classique.

L'utilisation de la méthode MO-CMA-ES nous a permis de trouver une partie convexe très intéressante pour les motoristes dans l'optimisation au point de fonctionnement 1 750 tours/min - 1 bar avec la prise en compte de la robustesse (flêche 1 sur la figure 7.7). En effet, la partie convexe située au milieu du front constitue un très bon compromis entre les deux objectifs de minimisation des émissions d'oxydes d'azote et des émissions de particules et de fumées. En revanche, nous constatons qu'une des solutions trouvées par la méthode classique (flêche 2 sur la figure 7.7) domine au sens de Pareto quelques solutions du front de Pareto obtenu à l'aide du MO-CMA-ES dans le cas du point de fonctionnement 1 750 tours/min - 1 bar avec robustesse, certainement dans une zone non explorée par MO-CMA-ES.

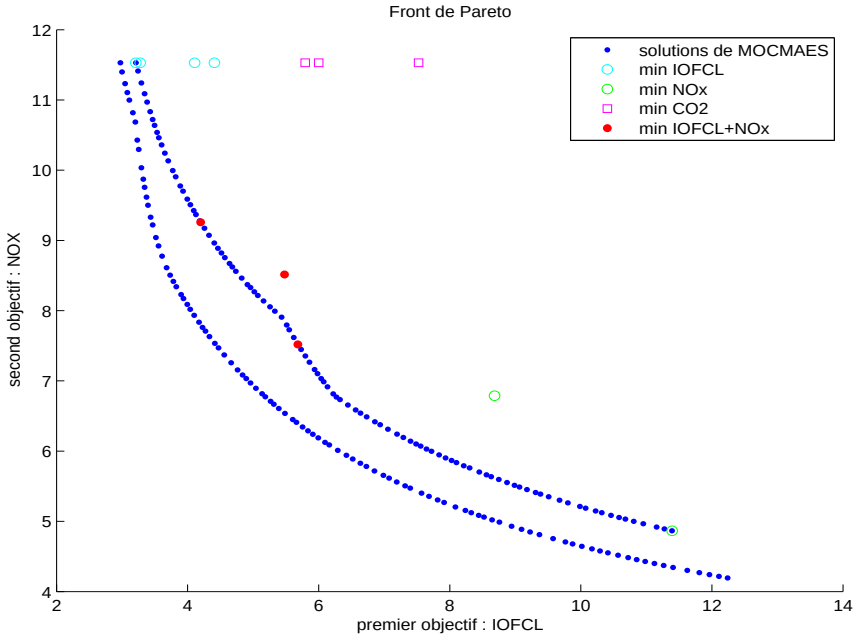


FIGURE 7.8 – Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 5 bar avec l'ajout de la contrainte de robustesse

Les résultats obtenus aux points de fonctionnement 1 750 tours/min - 1 bar et 1 750 tours/min - 5 bar en prenant en compte la robustesse sont comparés dans la section suivante à ceux obtenus à l'aide d'une méthode d'agrégation du problème à deux objectifs sous contraintes traité.

3.5 Comparaison entre les résultats du MO-CMA-ES et une méthode d'agrégation

Un problème multi-objectifs peut être approché par une fonction scalaire d'agrégation linéaire de tous les objectifs : $f(x) = \sum_{m=1}^{M} \alpha_m \cdot f_m(x)$ (voir section 2.1.1 du chapitre 5). Chaque poids des objectifs donne un sous-ensemble de solutions Pareto optimales ou seulement une solution simple. Il est nécessaire de lancer plusieurs optimisations avec différents poids pour obtenir le front de Pareto.

Pour approcher le problème d'optimisation multi-objectifs étudié, nous avons utilisé une fonction d'agrégation linéaire des deux objectifs poursuivis (oxydes d'azote : NOx, particules et fumées : IOFCL) sous les mêmes ϵ-contraintes que précédemment. L'optimisation mono-objectif effectuée à l'aide de la méthode d'optimisation locale actuelle est : $\beta \times f_{IOFCL} + (1 - \beta) \times f_{NOx}$, où $\beta \in [0, 1]$.

Nous avons effectué cette optimisation deux fois pour une centaine de valeurs de β uniformément réparties sur $[0, 1]$. Une première fois en partant de points initiaux quelconques (en rose sur les figures 7.9 et 7.10), et une seconde fois en partant des points du

front de Pareto obtenus à l'aide de l'optimisation multi-objectifs par MO-CMA-ES (en cyan sur ces mêmes figures).

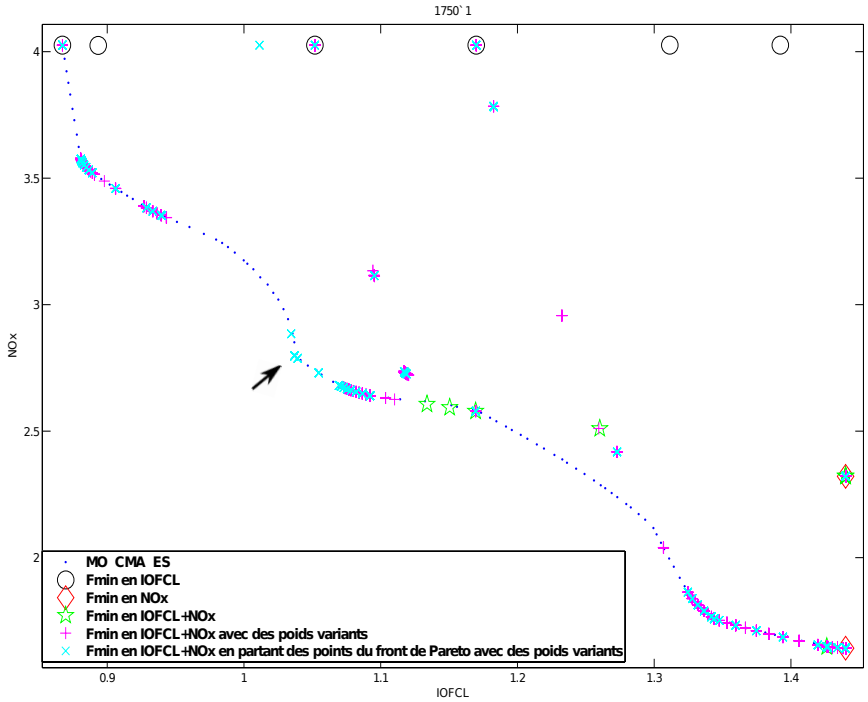


FIGURE 7.9 – Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 1 bar avec l'ajout de la contrainte de robustesse

Dans le cas de l'optimisation au point de fonctionnement 1 750 tours/min - 1 bar avec robustesse, les solutions obtenues par la méthode d'agrégation du problème ne couvrent que les parties convexes du front de Pareto obtenu par la méthode MO-CMA-ES, comme nous l'attendions. Les deux méthodes semblent alors conduire à des résultats similaires en considérant que les parties convexes sont les plus intéressantes pour le motoriste, mais les quatre objectifs passés en contraintes intéressent également le motoriste. De plus une trop forte variation des réglages dans le plan régime/charge peut donner des à-coups lors de la conduite et nuire au confort des occupants du véhicule. Ainsi, une solution sur une partie non convexe du front n'impliquant pas une trop forte variation des réglages dans le plan régime/charge peut être choisie comme réglage si les réponses des quatre objectifs passés en contraintes sont plus intéressantes que celle d'une solution d'une partie concave. En revanche, nous constatons qu'une solution trouvée par la méthode d'agrégation du problème (voir la flêche sur la figure 7.9), en partant des solutions du front de Pareto obtenu à l'aide de l'optimisation multi-objectifs par MO-CMA-ES, domine au sens de

Pareto ces mêmes solutions du front de Pareto dans le cas du point de fonctionnement 1 750 tours/min - 1 bar avec robustesse. En effet, la méthode MO-CMA-ES n'avait pas forcément totalement converger.

Dans le cas de l'optimisation au second point de fonctionnement 1 750 tours/min - 5 bar avec robustesse, la totalité du front est convexe, donc les solutions du front de Pareto obtenues par les deux méthodes sont plus proches. Cependant, le critère de diversité utilisé par le MO-CMA-ES permet d'obtenir une répartition uniforme des solutions du front de Pareto.

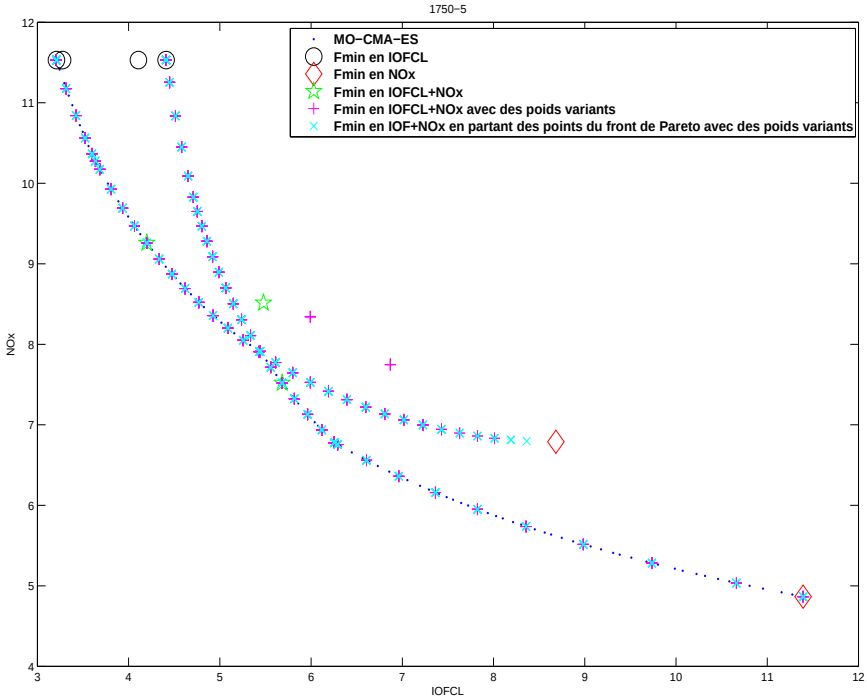


FIGURE 7.10 – Front de Pareto de l'optimisation des réglages au point de fonctionnement 1 750 tours/min - 5 bar avec l'ajout de la contrainte de robustesse

Nous notons que les résultats obtenus par la méthode d'agrégation pour la centaine de valeurs de β requiert 0.5×10^6 évaluations de la fonction objectif, alors que l'algorithme MO-CMA-ES requiert, lui, 1.5×10^6.

4 Conclusions

L'utilisation d'un algorithme évolutionnaire multi-objectifs adapté à l'optimisation sous contraintes nous a permis de trouver des compromis entre les deux objectifs anta-

gonistes de minimisation des émissions d'oxydes d'azote et des émissions de fumées et particules.

Une méthode de pénalisation évolutive est utilisée afin d'obtenir, à la génération finale, un front de Pareto intéressant pour chacune des exécutions effectuées et permettant que chacune des solutions du front de Pareto de la génération finale vérifie toutes les contraintes du problème. Cette méthode consiste à pénaliser les solutions ne vérifiant pas les contraintes suivant leurs niveaux de violation de chaque contrainte du problème jusqu'à une certaine génération, afin de s'approcher des bornes fixées par ces contraintes dans l'espace des paramètres. Puis, à partir de la génération fixée, les solutions ne vérifiant pas les contraintes sont pénalisées de façon discontinue (avec un paramètre adapté) afin de n'avoir que des solutions vérifiant toutes les contraintes dans les solutions du front de Pareto de la génération finale. Les solutions obtenues par cette méthode sont en moyenne "meilleures" que celles obtenues par les autres systèmes de pénalisation, ce qui permet de réduire le nombre d'exécutions de l'algorithme.

Les solutions que nous avons obtenues aux deux points de fonctionnement étudiés forment de bons compromis qui n'avaient pas été trouvés par la méthode classique. Ces solutions ont été comparées à celles obtenues à l'aide d'une méthode d'agrégation du problème à deux objectifs sous contraintes en faisant varier les poids de pénalisation. Les solutions obtenues par la méthode MO-CMA-ES et la méthode d'agrégation semblent équivalentes sur les parties convexes du front de Pareto. Mais le motoriste peut être amené à utiliser un réglage situé sur une partie concave du front de Pareto, obtenu par MO-CMA-ES, afin de prendre en compte les quatre réponses du moteur que sont la consommation, le bruit produit par le moteur, les émissions d'oxydes de carbone et d'hydrocarbures imbrûlés. De plus, les variations de réglage dans le plan régime/charge sont prises en compte par le motoriste et peuvent l'amener à utiliser un réglage situé sur une partie concave du front de Pareto.

Cette méthode est nécessairement plus coûteuse en temps de calcul que la méthode actuelle mais permet d'obtenir plusieurs compromis du problème posé et, ainsi, de vérifier les objectifs visés.

Conclusions générales

Au cours du chapitre 1 de la première partie de ce mémoire, différentes méthodes d'optimisation de fonctions coûteuses sans dérivées ont été exposées. On distingue quatre classes de méthodes d'optimisation sans dérivées : les méthodes de recherches directes n'utilisant que les évaluations de la fonction objectif, f, pour diriger l'optimisation, les métaheuristiques, les méthodes avec modèles de substitution et, enfin, les méthodes hybrides combinant certaines des catégories précédemment citées. Les méthodes directes et métaheuristiques sont peu sensibles aux imprécisions du calcul de la fonction f mais sont gourmandes en évaluation de f.

Les méthodes d'optimisation basées sur des modèles de substitution de la fonction objectif, peu coûteux à évaluer, permettent de déterminer un point qui minimise la fonction à partir du modèle de substitution. Ce modèle est mis à jour pour être suffisamment précis autour du point courant de l'optimisation. Un maximum d'informations concernant la fonction est alors exploitable au cours de chaque itération, permettant de limiter le nombre d'évaluations de la fonction coûteuse. La méthode SQA (Sequential Quadratic Approximation, présentée au chapitre 3) est une solution originale apportée au problème d'optimisation sans dérivées avec contraintes. Celle-ci appartient aux méthodes de région de confiance basées sur des modèles de substitution approchant la fonction objectif localement (chapitre 2). La méthode SQA permet, de prendre en compte des contraintes classiques, pour lesquelles les dérivées sont connues, mais également des contraintes sans dérivées. Ces contraintes sans dérivées sont prises en compte à l'aide de modèles locaux représentant chacune d'elles. Par ailleurs, une fonction de mérite avec des poids fixes permet de comparer les différents itérés.

Présentés dans le chapitre 4, les résultats obtenus par la méthode SQA sur des benchmarks d'optimisation sous contraintes sont très satisfaisants. Cette méthode permet de limiter le nombre d'évaluations coûteuses du problème, comparativement à d'autres méthodes d'optimisation sans dérivées sous contraintes. De plus, elle s'est révélée très efficace sur un problème posé en ingénierie de réservoir, le problème, de moindres carrés, de calage d'historique sous contraintes de bornes, pour lequel le nombre de données à caler est important.

Un autre résultat intéressant de la méthode SQA est sa capacité à traiter efficacement des contraintes non linéaires sans dérivées. SQA a ainsi permis de résoudre un

problème très complexe avec 68 contraintes sans dérivées et 124 paramètres, ainsi que deux problèmes réels d'optimisation sans dérivées, sous contraintes sans dérivées, de manière très performante : une approche permettant d'évaluer les incertitudes sur la prédiction de production d'huile d'un champ pétrolier, ainsi qu'un problème de calibration de moteurs.

Dans la partie II du présent mémoire, nous illustrons l'efficacité de la méthode d'optimisation multi-objectifs globale MO-CMA-ES (implémentée par Hansen).

Nous l'avons adaptée pour l'optimisation sous contraintes. Pour une application de calibration de moteur multi-objectifs, la méthode MO-CMA-ES a permis d'obtenir de nombreux compromis entre deux objectifs antagonistes. La méthode de prise en compte des contraintes par pénalisation évolutive, a été appliquée avec succès sur cette application.

Perspectives

La méthode de prise en compte des contraintes dans la méthode SQA a démontré son efficacité sur différentes applications. Néanmoins, l'étude de la mise à jour du poids de pénalisation des contraintes sans dérivées dans la fonction de mérite reste à faire. Une adaptation évolutive des différents poids de pénalisation des contraintes sans dérivées empruntant les multiplicateurs de Lagrange obtenus lors de la résolution des sous-problèmes (cf. section 4.4 du chapitre 3).

Une amélioration majeure pourrait être apportée à la méthode SQA présentée dans la première partie de ce manuscrit en permettant que le nombre de points d'interpolation du modèle quadratique utilisé pour optimiser la fonction objectif puisse évoluer au cours des itérations. Ainsi l'optimisation commencerait avec la construction d'un modèle linéaire (après $n+1$ évaluations). Wild, dans ces travaux, augmente le nombre de points d'interpolation du modèle au cours des itérations jusqu'à la construction d'un modèle quadratique complet. Le modèle est alors enrichie au cours des itérations grâce aux différentes simulations de la méthode. Wild, lorsqu'une réduction de région de confiance est effectuée, réduit le nombre de points d'interpolation pour n'utiliser que les informations locales de courbures autour du point optimal. Lorsque le nombre de points devient important, la méthode peut utiliser un modèle de régression quadratique.

Une troisième plus-value intéressante de notre méthode serait de pouvoir introduire de l'information a priori sur la fonction objectif (par exemple pour le calage d'historique [50, 49]). Les modèles quadratiques sous déterminés, présentés dans la section 2.2.3 du chapitre 2, pourraient permettre de prendre en compte la structure creuse du gradient et de la Hessienne de la fonction objectif. On pourrait alors résoudre de manière plus performante des problèmes nécessitant l'optimisation de fonctions séparables ($f(x_1, x_2) = f_1(x_1) + f_2(x_2)$). De façon analogue, certaines dérivées connues pourraient également être prises en considération.

D'autre part, certains auteurs, Oeuvray [126] ou Wild, Regis et Shoemaker [178], pro-

posent de remplacer le modèle de substitution quadratique par des modèles plus complexes RBF pour une optimisation plus globale et une meilleure prise en compte du bruit sur l'évaluation de la fonction objectif. Ces approches pourraient être étudiées afin d'en cerner le potentiel sur nos applications. On pourrait également étudier les modèles de krigeage. La principale difficulté serait la complexité du sous-problème d'optimisation à résoudre (minimisation d'une fonction de type RBF ou krigeage dans la région de confiance sous contraintes). On pourrait imaginer utiliser des modèles de substitution polynomiaux locaux (d'ordre plus élevé que les modèles quadratiques) de la fonction objectif mis à jour à chaque itération. La solution de la minimisation de ces modèles dans une région de confiance peut être obtenue à l'aide des résultats de Lasserre [114, 115].

Enfin, on pourrait envisager de paralléliser la méthode SQA. En effet, après la phase d'initialisation, les évaluations de la fonction objectif sont effectuées une à une par la méthode SQA. La méthode pourrait tirer partie de l'évolution de plusieurs modèles de la fonction (construits en parallèle), ayant différentes caractéristiques, au cours de l'algorithme. Un réel progrès serait apporté au cours des itérations, provoqué par l'échange de l'information de courbure, entre les différents modèles, leur permettant d'être mis à jour plus précisément.

Pour l'optimisation multi-objectifs, il serait intéressant d'une part d'utiliser les améliorations dans le calcul de la mesure d'hypervolume en trois dimensions apportées par Fleischer [63] en vue de l'optimisation avec trois objectifs de l'application en calibration des moteurs. D'autre part, différentes approches autres que MO-CMA-ES, existent et pourraient être évaluées sur nos applications d'optimisation multi-objectifs : le NSGA II de Deb [41] est un algorithme génétique multi-objectifs basé sur le principe de Pareto, tout comme le MO-CMA-ES, qui utilise également la sélection par non dominance et la crowding-distance (voir paragraphe 1.2.1 du chapitre 6), l'approche géométrique NBI de Dennis [45] qui vise à reconstruire le front de Pareto par des minimisations particulières à l'aide d'une méthode d'optimisation non linéaire différentiable. Dans ce domaine, une étude multi-objectifs sous contraintes sur deux applications réelles de l'automobile a été effectuée par Roudenko à l'aide du NSGA II [151].

Annexes

Annexe A

Profils de performances et profils de données

Pour comparer de façon synthétique différents optimiseurs sur une série de problèmes, nous utilisons des outils adaptés comme les profils de performances et de données [121].

Les profils de performances permettent de comparer sur de nombreux problèmes tests les performances obtenues par différents algorithmes d'optimisation. Soit $\mathcal{P}$ l'ensemble des problèmes, $\mathcal{S}$ l'ensemble des solveurs d'optimisation et $\mathcal{T}$ le test de convergence. Les profils de performances sont définis par une mesure de performances $t_{p,s}$ obtenue pour $p \in \mathcal{P}$ et $s \in \mathcal{S}$, cette mesure peut être le nombre d'évaluations de la fonction objectif f nécessaire pour atteindre une certaine précision sur celle-ci. Pour chaque paire (p, s) de problème p et de solveur s, le rapport de performance est défini par

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} | 1 \leq s \leq |\mathcal{S}|\}},$$

où $|\mathcal{S}|$ est le cardinal de $\mathcal{S}$. Dans le cas particulier où le solveur s ne parvient pas à satisfaire le test de convergence sur le problème p, la convention $r_{p,s} = \infty$ est utilisée.

Le profil de performances d'un solveur s est défini comme la fonction de distribution du rapport de performance,

$$p_s(\alpha) = \frac{1}{|\mathcal{P}|} size\{p \in \mathcal{P} | r_{p,s} \leq \alpha\}.$$

La valeur $p_s(\alpha = 1)$ est la fraction de problèmes pour lesquels le solveur $s \in \mathcal{S}$ est le plus rapide. Pour α suffisamment grand, $p_s(\alpha)$ est la fraction de problèmes résolus par $s \in \mathcal{S}$. En comparant ces valeurs pour les différents solveurs, nous pouvons savoir lequel est le plus rapide et le plus robuste (celui qui parvient à résoudre le plus de problèmes pour α grand). La comparaison des valeurs de α prises par les différents profils de performances pour un même pourcentage de problèmes permet de savoir quel est le solveur le plus efficace pour résoudre ce pourcentage de problèmes.

Le test de convergence utilisé pour l'optimisation sans dérivée de fonction coûteuse à évaluer ne dépend pas des gradients, et peut s'écrire :

$$f(x^*) \leq f_L + \tau(f(x_0) - f_L), \tag{A.1}$$

où $\tau > 0$ est une tolérance, x_0 est le point de départ du problème, et f_L est calculée pour chaque problème $p \in \mathcal{P}$ comme la plus petite valeur de f obtenue par tous les solveurs en un nombre d'évaluations maximum fixé (c'est une estimation de f à la solution, le nombre d'évaluations maximum est donc important). Ce test de convergence s'écrit également :

$$f(x_0) - f(x^*) \geq (1 - \tau)(f(x_0) - f_L), \tag{A.2}$$

ce qui signifie que la réduction $f(x_0) - f(x^*)$ réalisée par x^* est supérieure à un facteur $(1 - \tau)$ de la meilleure réduction possible $f(x_0) - f_L$.

La tolérance $\tau \in [0, 1]$ dans (A.1) représente un pourcentage de réduction de la valeur $f(x_0)$ de départ après un nombre d'évaluations donné.

Les profils de performances fournissent une vue précise sur les performances des solveurs les uns par rapport aux autres en un nombre d'évaluations de fonction donné. Par contre, ils ne fournissent pas suffisamment d'informations pour les problèmes d'optimisation de fonctions coûteuses.

Les profils de données permettent d'illustrer la performance d'un solveur en fonction du nombre d'évaluations de la fonction, et ainsi de choisir le solveur permettant d'utiliser au mieux un budget d'évaluations fixé afin d'atteindre la réduction de la fonction objectif voulue. Soit $t_{p,s}$ le nombre d'évaluations de la fonction requis pour satisfaire (A.1) pour une tolérance τ donnée. Le profil de données d'un solveur s est défini comme la fonction de distribution de $\frac{t_{p,s}}{n_p+1}$,

$$d_s(\alpha) = \frac{1}{|\mathcal{P}|} size\{p \in \mathcal{P} | \frac{t_{p,s}}{n_p + 1} \leq \alpha\}, \tag{A.3}$$

est le pourcentage de problèmes qui peuvent être résolus (pour une tolérance τ) avec α évaluations de la fonction. n_p est le nombre de variables du problème d'optimisation $p \in \mathcal{P}$, ce qui permet de rendre la mesure de performance indépendante de la dimension de chaque problème. $d_s(k)$ correspond au pourcentage de problèmes qui peuvent être résolus avec l'équivalent de k estimations de gradient par différences finies (qui nécessite $n_p + 1$ évaluations de la fonction). Une limite sur le total d'évaluations de fonction est également fixée, et $t_{p,s} = \infty$ si le test de convergence (A.1) n'est pas vérifié lors de la dernière évaluation permise.

Les profils de données fournissent le nombre d'évaluations requis pour résoudre tous les problèmes pour chaque solveur. De plus, nous remarquons que le profil de données d'un solveur $s \in \mathcal{S}$, contrairement au profil de performances, est indépendant des autres solveurs.

Pour des exemples de profils de performances et de données, le lecteur pourra se référer aux résultats de la section 1 du chapitre 4 et à l'annexe F.

Annexe B

Méthode d'optimisation classique

Méthode de Quasi-Newton

La méthode de Quasi-Newton [125] la plus populaire est la méthode Broyden-Fletcher-Goldfarb-Shanno (BFGS). Celle-ci consiste à construire un modèle quadratique tangent de la fonction objectif f à l'itération courante x_k : $m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$, où B_k est une approximation symétrique semi-définie positive de la Hessienne de f. Le pas de descente est alors $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$ avec α_k la taille du pas choisi de manière à satisfaire les conditions de Armijo ou Wolfe et B_k^{-1} une approximation de l'inverse de la Hessienne de f symétrique semi-définie positive qui peut être mise à jour à partir des formules BFGS. En effet, la méthode de BFGS permet de construire une approximation de l'inverse de la matrice Hessienne de la fonction à optimiser : à la place de calculer B_k^{-1} à chaque itération, une simple mise à jour de celle-ci afin de prendre en compte la courbure mesurée lors de la dernière itération est effectuée. La mise à jour de l'approximation B_{k+1}^{-1} doit être symétrique semi-définie positive et doit vérifier l'équation sécante : $B_{k+1}^{-1}(\nabla f_{k+1} - \nabla f_k) = x_{k+1} - x_k$. Mais afin de déterminer B_{k+1}^{-1} de manière unique, une condition supplémentaire de proximité de B_k^{-1} est formulée :

$$\min_{B^{-1}} \|B^{-1} - B_k^{-1}\| \tag{B.1}$$

sous les contraintes

$$B^{-1} = (B^{-1})^T,\ B^{-1}(\nabla f_{k+1} - \nabla f_k) = x_{k+1} - x_k. \tag{B.2}$$

La norme utilisée est la norme de Frobenius pondérée $\|A\|_W = \|W^{1/2} A W^{1/2}\|_F$, où $\|\cdot\|_F$ représente la norme de Frobenius :

$$\|A\|_F = \{\sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^2\}^{1/2}, \quad A \in \mathbb{R}^{n \times n}. \tag{B.3}$$

L'unique solution H_{k+1} vérifiant B.1 sous les contraintes B.2 est donnée par :

$$B_{k+1}^{-1} = (I - \rho_k s_k y_k^T) B_k^{-1} (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \tag{B.4}$$

où $\rho_k = 1/(y_k^T s_k)$, $s_k = x_{k+1} - x_k$ et $y_k = \nabla f_{k+1} - \nabla f_k$.

Cette formule est connue sous le nom de formule BFGS.

Annexe C

Optimisation avec contraintes : résultats généraux

Dans cette annexe, nous reprenons les principales définitions et résultats de l'optimisation sous contraintes puis les principales méthodes pour résoudre ce type de problème (pour plus de détéails voir [64], [65], [125],[71] et [17]).

1 Problème d'optimisation sous contraintes d'égalité et d'inégalité

1.1 Formulation du problème

Considérons le problème d'optimisation

$$(P) \begin{cases} \min\limits_{x} f(x) \\ c_i(x) = 0, \quad i \in E \\ c_i(x) \leq 0, \quad i \in I, \end{cases} \tag{C.1}$$

dans lequel on minimise une fonction $f : \Omega \mapsto \mathbb{R}$ (avec Ω un ouvert de $\mathbb{R}^n$) et où E et I forment une partition de $\{1, ..., m\}$ ($E \cup I = \{1, ..., m\}$ et $E \cap I = \emptyset$). Les contraintes sont définies par la fonction $c : \mathbb{R}^n \mapsto \mathbb{R}^m$. Si $v \in \mathbb{R}^m$, on note v_E (resp. v_I) le vecteur de $\mathbb{R}^{|E|}$ (resp. $\mathbb{R}^{|I|}$) formé des composantes v_i de v avec $i \in E$ (resp. $i \in I$). De même, on notera c_E (resp. c_I) la fonction définissant les contraintes d'égalité (resp. les contraintes d'inégalité).

Définition 1.1 *On appelle* ***ensemble admissible du problème*** (P)*, l'ensemble* $\mathcal{X}$ *défini par*

$$\mathcal{X} := \{x \in \Omega : c_E(x) = 0, c_I(x) \leq 0\}.$$

Définition 1.2 *On appelle* ***minimum (global) de*** (P) *tout point* $x_* \in \mathcal{X}$ *vérifiant*

$$f(x_*) \leq f(x), \quad \forall x \in \mathcal{X}.$$

Définition 1.3 *On dit que $x_* \in \mathcal{X}$ est un* ***minimum local de*** *(P) s'il existe un voisinage V de x_* tel que*

$$f(x_*) \leq f(x), \quad \forall x \in \mathcal{X} \cap V.$$

Définition 1.4 *On dit que c_i $(i \in I)$ est* ***active en*** *x si $c_i(x) = 0$. On note*

$$I^0(x) := \{i \in I : c_i(x) = 0\}$$

l'ensemble des indices des contraintes d'inégalité actives en x, on note également $I^0_* := I^0(x_*)$.

Définition 1.5 *On dit qu'un* ***problème d'optimisation est convexe*** *si sa fonction coût et ses contraintes d'inégalité sont convexes et si ses contraintes d'égalité sont linéaires.*

1.2 Conditions nécessaires d'optimalité du premier ordre de Karush-Kuhn-Tucker

Les conditions nécessaires d'optimalité du problème d'optimisation (P) sont une réunion d'équations et/ou d'inéquations et/ou de propriétés que vérifient les solutions de (P). On parle de conditions du premier ordre lorsque celles-ci ne font intervenir que les dérivées premières de f et de c. Avant de donner les conditions d'optimalité du problème C.1, on donne quelques définitions utiles par la suite.

Définition 1.6 *On dit que $d \in \mathbb{R}^n$ est* ***tangent à*** *$\mathcal{X}$ en x s'il existe une suite $\{d_k\} \subset \mathbb{R}^n$ et une suite $\{t_k\} \subset \mathbb{R}^*_+ = \{t \in \mathbb{R} : t > 0\}$ telles que*

$$d_k \to d, \;\; t_k \to 0, \;\; x + t_k d_k \in \mathcal{X}.$$

On note $T_x\mathcal{X}$ l'ensemble des vecteurs tangents à $\mathcal{X}$ en x et on l'appelle le cône tangent.

Définition 1.7 *On appelle* ***cône linéarisant*** *$T'_x\mathcal{X}$* ***de*** *$\mathcal{X}$* ***en*** *x l'ensemble*

$$T'_x\mathcal{X} := \{d \in \mathbb{R}^n : \; c'_E(x) \cdot d = 0, \; c'_{I^0(x)}(x) \cdot d \leq 0\}.$$

Définition 1.8 *On dit que les* ***contraintes*** *sont* ***qualifiées en*** *$x \in \mathcal{X}$ si*

$$T_x\mathcal{X} = T'_x\mathcal{X}. \qquad (QC)$$

Définition 1.9 *On dit que les* ***contraintes vérifient la condition de qualification (QC-A) en*** *$x \in \mathbb{R}^n$ lorsque $c_{E \cup I^0(x)}$ est affine dans un voisinage de x*

Les propositions 1.10 et 1.11 sont reprises des propositions 3.25 et 3.21 de [71].

Proposition 1.10 *Soit x vérifiant les contraintes de (P). Supposons que $c_{E \cup I^0(x)}$ sont dérivables en x et que $c_{I \setminus I^0(x)}$ est continu en x. Si la condition de qualification des contraintes (QC-A) est satisfaite en x alors les contraintes sont qualifiées en x.*

Théorème 1.11 *Soit x_* un minimum local de (P), supposons que f et $c_{E\cup I_*^0}$ sont dérivables en x_* et que les contraintes soient qualifiées en x_*. Alors, il existe un multiplicateur $\lambda_* \in \mathbb{R}^m$ tel que l'on ait*

$$(KKT) \quad \begin{cases} (a)\ \nabla f(x_*) + A(x_*)^\top \lambda_* = 0 \\ (b)\ c_E(x_*) = 0,\ c_I(x_*) \leq 0 \\ (c)\ (\lambda_*)_I \geq 0 \\ (d)\ (\lambda_*)_I^\top c_I(x_*) = 0, \end{cases} \tag{C.2}$$

où ∇ représente le gradient associé au produit scalaire euclidien, et où $A(x)$ est la matrice jacobienne de c en x ($A(x) = \nabla c(x)^\top$)

Quelques remarques sur le système d'optimalité du théorème 1.11

1. Le système (C.2) est connu sous le nom de conditions de Karush, Kuhn et Tucker ou (KKT).
2. L'équation (a) peut aussi s'écrire

$$\nabla_x L_{(P)}(x_*, \lambda_*) = 0,$$

avec $L_{(P)}$ le lagrangien associé au problème (P) :

$$L_{(P)}(x, \lambda) = f(x) + \lambda^\top c(x). \tag{C.3}$$

3. Le vecteur λ_* constitué des multiplicateurs de Lagrange associés aux contraintes c et à la solution x_*.
4. Un couple (x_*, λ_*) vérifiant (C.2) est appelé solution primale-duale de (P) : x_* est la solution primale et λ_* la solution duale.
5. Un point x_* pour lequel il existe $\lambda_* \in \mathbb{R}^m$ tel que (C.2) est vérifiée est dit stationnaire.
6. Dans la condition (b), on reconnaît la condition d'admissibilité des contraintes en x_* ($x_* \in \mathcal{X}$).
7. Les conditions (c) et (d) concernent uniquement les contraintes d'inégalité. La condition (c) est la condition de signe constant des multiplicateurs et la condition (d) s'appelle la condition de complémentarité. La condition (d) peut aussi se reformuler composante par composante (en notant que l'on a à la fois $(\lambda_*)_I \geq 0$ et $c_I(x_*) \leq 0$) :

$$(\lambda_*)_i c_i(x_*) = 0,\ \ \forall i \in I.$$

Cette dernière expression implique que le multiplicateur correspondant à une contrainte inactive est nul :

$$c_i(x_*) < 0 \quad \Longrightarrow \quad (\lambda_*)_i = 0.$$

Définition 1.12 *On dit que la* ***condition de complémentarité stricte est satisfaite en une solution primale-duale*** (x_*, λ_*) ***de*** (P) *lorsque* $\forall i \in I$ *:*

$$c_i(x_*) < 0 \quad \Longleftrightarrow \quad (\lambda_*)_i = 0.$$

Voici ci-dessous une condition suffisante d'optimalité du premier ordre reprise de la proposition 3.22 de [71].

Proposition 1.13 *Considérons le problème* (P) *et supposons que ce problème est convexe (voir définition 1.5). Soit* x_* *un point vérifiant les contraintes de* (P)*. Si* f *et* c *sont dérivables en* x_* *et s'il existe* $\lambda_* \in \mathbb{R}^m$ *tel que les conditions de KKT (C.2) soient vérifiées, alors* x_* *est un minimum global de* (P)*.*

2 Méthodes de pénalisation

Historiquement, les méthodes de pénalisation sont les toutes premières méthodes de résolution du problème (P). Elle consiste à transformer un problème d'optimisation avec contraintes en un problème (ou une suite de problèmes) d'optimisation sans contrainte. Les différentes méthodes de pénalisation suivent généralement le principe suivant : le problème original (P_{EI}) est remplacé par un ou des problèmes sans contrainte de la forme

$$(\widetilde{P}_\sigma) : \quad \min_x \Theta_\sigma(x),$$

où

$$\Theta_\sigma(x) := f(x) + \sigma p(x),$$

est la fonction de pénalisation, p est la fonction pénalisant les contraintes et $\sigma > 0$ est le facteur de pénalisation. On observe que les contraintes de (P) ont été remplacées par le terme additionnel $\sigma p(x)$ à minimiser dans la fonction objectif de $(\widetilde{P}_\sigma)$. L'une des questions essentielles est de savoir si en résolvant $(\widetilde{P}_\sigma)$ on résout bien (P). La réponse à cette question dépend du choix de la fonction p et du facteur de pénalisation σ, et conduit à la notion de pénalisation exacte :

Définition 2.1 *On dit que* Θ_σ *est une* ***fonction de pénalisation exacte en un minimum local*** x_* ***de*** (P) *si* x_* *est un minimum local de* Θ_σ*.*

Parmi les différentes méthodes de pénalisation on distingue 2 grandes sous-classes de méthodes :

* méthodes de pénalisation extérieure,
* méthodes de pénalisation intérieure.

Dans les méthodes de pénalisation extérieure la fonction p pénalise la violation des contraintes. Elle doit respecter les propriétés suivantes :

$$\begin{cases} (i) \quad p \text{ est continue sur } \mathbb{R}^n \\ (ii) \quad p(x) \geq 0, \forall x \in \mathbb{R}^n \\ (iii) \quad p(x) = 0 \Longleftrightarrow x \in \mathcal{X}. \end{cases}$$

Un exemple simple et intuitif de fonction de pénalisation extérieure est la fonction de pénalisation quadratique des contraintes (fonction utilisée par [37]). Cette fonction qui est à la base des méthodes de lagrangien augmenté (LA) (voir [43]) s'écrit :

$$\Theta_\sigma(x) = f(x) + \frac{1}{2\sigma}(||c_E(x)||_2^2 + ||c_I(x)^+||_2^2), \tag{C.4}$$

où $(v^+)_i = \max(v_i, 0)$. La méthode de pénalisation consiste alors à minimiser la fonction Θ_σ précédente pour une suite décroissante de valeurs de σ jusqu'à ce que les conditions d'optimalité de (P) (équation (C.2)) soient respectées à une précision donnée. Le talon d'Achille de cette méthode est que la minimisation de la fonction Θ_σ est de plus en plus difficile à effectuer lorsque σ diminue. Le mauvais conditionnement du Hessien $\nabla^2_{xx}\Theta_\sigma$ proche de la solution de (P) est la source de cette difficulté. La méthode du lagrangien augmenté, construite sur les bases de la pénalisation quadratique (voir section [43]), permet en général d'obtenir de meilleurs résultats en éliminant ce mauvais conditionnement.

Dans les méthodes de pénalisation intérieure la fonction p pénalise l'abord de la frontière du domaine admissible. Elle doit respecter les propriétés suivantes : (on note $\mathcal{X}^0$ l'intérieur de $\mathcal{X}$, $\partial\mathcal{X}$ la frontière de $\mathcal{X}$ et on suppose que $\mathcal{X}^0 \neq \emptyset$)

$$\begin{cases} (i) \quad p \text{ est continue sur } \mathcal{X}^0 \\ (ii) \quad p(x) \geq 0, \forall x \in \mathcal{X}^0 \\ (iii) \quad p(x) \to +\infty, \text{ quand } x \in \mathcal{X}^0 \to \partial\mathcal{X}. \end{cases}$$

On parle aussi de méthodes (resp. fonction) "barrière" pour évoquer les méthodes (resp. fonction) de pénalisation intérieure : la condition (iii) crée une barrière au bord de l'ensemble admissible rendant les solutions de $(\widetilde{P}_\sigma)$ admissibles (ces solutions restent dans $\mathcal{X}^0$). L'une des méthodes de pénalisation intérieure les plus connues est la pénalisation logarithmique (les méthodes de pénalisation logarithmique remontent à [70] et [62]) qui est à la base des algorithmes de points intérieurs (voir section [43]). Pour les contraintes d'inégalité seulement, cette fonction s'écrit :

$$\Theta_\sigma(x) = f(x) - \sigma \sum_{i \in I} \log(-c_i(x)) \tag{C.5}$$

La méthode de pénalisation consiste alors à minimiser la fonction Θ_σ précédente pour une suite décroissante de valeurs de σ jusqu'à ce que les conditions d'optimalité de (P) (équation (C.2)) soient respectées à une précision donnée. Similairement au cas de la pénalisation quadratique, la minimisation de la fonction Θ_σ est de plus en plus difficile à effectuer lorsque σ diminue : le mauvais conditionnement du Hessien $\nabla^2_{xx}\Theta_\sigma$ proche de la solution de (P) en est la cause. Afin de prendre aussi en compte les contraintes d'égalité de (P), on peut utiliser la fonction de pénalisation mixte suivante :

$$\Theta_\sigma(x) = f(x) + \frac{1}{2\sigma}||c_E(x)||_2^2 - \sigma \sum_{i \in I} \log(-c_i(x)). \tag{C.6}$$

Cette fonction combine les techniques de pénalisation quadratique et logarithmique : les contraintes d'égalité de (P) sont pénalisées par la fonction de pénalisation quadratique

(fonction (C.4)), alors que les contraintes d'inégalité sont pénalisées par la fonction de pénalisation logarithmique (fonction (C.5)). Les propriétés de convergence de la méthode de pénalisation avec cette fonction mixte ont été présentées dans [76].

3 Méthode de programmation quadratique successive

On trouve les premières traces de la méthode SQP dans [180]. Cependant, il faut attendre le milieu des années 70 pour voir le développement de cette méthode (voir [127], [83], [84], [132], [130], [131]). Dans cette section, on décrit la méthode de programmation quadratique successive (SQP : "Sequential Quadratic Programming") appliquée à la minimisation du problème (P). S'apparentant aux algorithmes newtoniens, la méthode SQP est actuellement l'une des méthodes les plus efficaces pour résoudre les problèmes d'optimisation non linéaire. Elle consiste à transformer la résolution d'un problème d'optimisation non linéaire en la résolution d'une suite de problèmes d'optimisation quadratiques. Ceux-ci sont obtenus par linéarisation des conditions d'optimalité (C.2) au point (x_k, λ_k) courant. Le système d'optimalité linéarisé a pour inconnues le déplacement (d, μ) à apporter à (x_k, λ_k) :

$$\begin{cases} M_k d + A_k^\top \mu = -\nabla_x L_k \\ (c_k)_E + (A_k)_E d = 0, \quad (c_k)_I + (A_k)_I d \leq 0 \\ (\lambda_k + \mu)_I \geq 0 \\ (\lambda_k + \mu)_I^\top (c_k)_I + (\lambda_k)_I^\top (A_k)_I d = 0, \end{cases} \tag{C.7}$$

avec les notations $c_k := c(x_k)$ ($(c_k)_E := c_E(x_k)$ et $(c_k)_I := c_I(x_k)$), $A_k = c'(x_k)$ la matrice jacobienne de c, $\nabla_x L_k := \nabla_x L_{(P)}(x_k, \lambda_k)$ et $M_k := \nabla^2_{xx} L_{(P)}(x_k, \lambda_k)$ le Hessien du lagrangien en (x_k, λ_k). On peut obtenir à partir du système (C.7) un problème plus facilement résoluble en ajoutant le terme $(\mu)_I^\top (A_k)_I d$ dans la dernière équation (ce terme est négligeable lorsque le déplacement (d, μ) est petit, ce qui est le cas lorsque (x_k, λ_k) est proche d'une solution de (P)). Sur ce système linéarisé modifié, on réalise le changement de variable $\lambda^{QP} := \lambda_k + \mu$, ce qui nous donne :

$$\begin{cases} M_k d + A_k^\top \lambda^{QP} = -\nabla f_k \\ (c_k)_E + (A_k)_E d = 0, \quad (c_k)_I + (A_k)_I d \leq 0 \\ (\lambda^{QP})_I \geq 0 \\ (\lambda^{QP})_I^\top ((c_k)_I + (A_k)_I d) = 0, \end{cases} \tag{C.8}$$

où $\nabla f_k = \nabla f(x_k)$ est le gradient de f en x_k. On peut facilement vérifier que le système (C.8) est le système d'optimalité du problème quadratique suivant :

$$\begin{cases} \min_d \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c_E(x_k) + (A_k)_E d = 0 \\ c_I(x_k) + (A_k)_I d \leq 0. \end{cases} \tag{C.9}$$

Le problème quadratique (C.9) s'obtient facilement à partir du problème initial (P).

Remarques 3.1

1. *Les contraintes de (C.9) s'obtiennent en linéarisant les contraintes de (P) en x_k.*
2. *La fonction objectif de (C.9) est hybride, avec le gradient de f pour la partie linéaire et le Hessien du lagrangien pour la partie quadratique.*
3. *Rappelons que le lagrangien de (P) s'écrit*

$$L_{(P)}(x, \lambda) = f(x) + \lambda^\top c(x).$$

La matrice M_k est égale au Hessien du Lagrangien c'est à dire :

$$M_k = \nabla^2_{xx} f(x) + \lambda^\top \nabla^2_{xx} c(x).$$

On remarque dans cette dernière équation que le calcul de M_k fait intervenir deux termes différents qui sont le Hessien de f et la courbure des contraintes. Une conséquence directe de cette observation est que dans le cas de contraintes linéaires ($\nabla^2_{xx} c(x) = 0$) le Hessien du lagrangien est égal au Hessien de f.

Définition 3.2 *Le problème d'optimisation (C.9) est appelé* ***Problème Quadratique Tangent (PQT) de*** (P) ***en*** x_k.

La méthode SQP consiste alors à produire une suite de points $\{(x_k, \lambda_k)\}$ qui converge vers une solution primale-duale (x_*, λ_*) de (P). En chaque point (x_k, λ_k), on recherche une solution primale-duale (d_k, λ_k^{QP}) du problème quadratique tangent (C.9). Une fois cette solution obtenue il ne reste plus qu'à calculer le nouveau point (x_{k+1}, λ_{k+1}) par :

$$x_{k+1} = x_k + d_k \quad \text{et} \quad \lambda_{k+1} = \lambda_k^{QP}.$$

Plus généralement, on parlera de méthode SQP lorsque la matrice M_k ($M_k \approx \nabla^2_{xx} L_{(P)}(x_k, \lambda_k)$) définie dans le PQT (C.9) est seulement une approximation du Hessien du lagrangien. Ainsi, M_k, généralement une matrice symétrique (semi) définie positive. On peut utiliser l'algorithme SQP local C.1 pour résoudre (P).

Nous noterons que la résolution de (C.8) est l'étape coûteuse de l'algorithme SQP. Ainsi, il est important de pouvoir résoudre ce problème en un temps raisonnable pour valider le choix de l'utilisation de la méthode SQP par rapport aux autres méthodes envisageables (méthode de pénalisation, méthode de Points Intérieurs).

Le théorème suivant, repris du théorème 13.2 de [17], donne un résultat standard de convergence locale de la méthode SQP.

Théorème 3.3 *Supposons que f et c sont de classe C^2 dans un voisinage d'un point stationnaire x_* de (P) avec λ_* un multiplicateur de Lagrange associé. Supposons aussi que la condition de complémentarité stricte est vérifiée (voir définition 1.12) et que $(x_*, (\lambda_*)_{E \cup I_0^*})$*

Algorithme C.1 Algorithme SQP local

- **Initialisation** : Un itéré initial est donné : (x_1, λ_1). $k = 1$. Calculer $c(x_1)$, $\nabla f(x_1)$ et $A(x_1)$.
- Tant que (C.2) n'est pas satisfaite
  (1) Calculer M_k et trouver une solution primale-duale de (C.9), i.e., une solution (d_k, λ_k^{QP}) de (C.8).
  (2) Calculer le nouveau point :
  $$x_{k+1} = x_k + d_k \quad \text{et} \quad \lambda_{k+1} = \lambda_k^{QP}.$$
  (3) Calculer $c(x_k)$, $\nabla f(x_k)$ et $A(x_k)$.
  (4) Accroître k de 1 : $k := k + 1$.

est un point stationnaire régulier (voir remarque 1. de 3.4) du problème avec contraintes d'égalité

$$(P_{E\cup I_0^*}) \begin{cases} \min_x f(x) \\ c_i(x) = 0, \quad i \in E \cup I_*^0, \end{cases} \tag{C.10}$$

Considérons l'algorithme SQP local (voir algorithme C.1), dans lequel d_k est un point stationnaire de norme minimale du problème quadratique tangent (C.9). Alors, il existe un voisinage V de (x_*, λ_*) tel que, si le premier itéré $(x_1, \lambda_1) \in V$:

(i) l'algorithme SQP local est bien défini et il génère une suite $\{(x_k, \lambda_k)\}$ qui converge superlinéairement vers (x_*, λ_*) ;

(ii) les contraintes actives du problème quadratique tangent (C.9) sont celles du problème (P) ;

(iii) si, en plus, f et c sont de classe $C^{2,1}$ dans un voisinage de x_*, la convergence de $\{(x_k, \lambda_k)\}$ est quadratique.

Remarques 3.4

1. *Soit K la matrice définie par*
$$K := \begin{pmatrix} (L_{E\cup I_0^*})_* & (A_{E\cup I_0^*})_*^\top \\ (A_{E\cup I_0^*})_* & 0 \end{pmatrix},$$
où $(A_{E\cup I_0^*})_* = (c_{E\cup I_0^*})'(x_*)$ et $(L_{E\cup I_0^*})_* = \nabla^2_{xx} L_{(P_{E\cup I_0^*})}(x_*, \lambda_*)$. Un point stationnaire de (C.10) est dit régulier si K n'est pas singulière (voir proposition 12.1 et définition 12.2 de [17]).

2. *La propriété (ii) du théorème 3.3 est connue sous le nom de propriété d'**identification finie des contraintes actives** de l'algorithme SQP. Il est souhaitable que la méthode de résolution du problème quadratique tangent (C.9) utilisée dans l'algorithme SQP tire parti de cette propriété.*

De nombreux codes SQP ont été développés, voici ci-dessous les plus connus :

* NPSOL, pour les problèmes de taille moyenne (voir [73])
* SNOPT, pour les problèmes de grande taille creux (voir [72])
* NLPQL (voir [153])

Nous noterons que, par comparaison avec d'autres types de méthodes (méthode de pénalisation ou de LA), la méthode SQP demande en général moins d'évaluation de la fonction coût (voir [125, chapitre 15]). Plus précisément, elle demande en général moins d'évaluations de la fonction coût par rapport à une méthode de points intérieurs (voir [43]). Cependant, chaque résolution de problème quadratique tangent est plus coûteuse à effectuer que la résolution des systèmes linéaires provenant des méthodes de pénalisation ou de LA.

4 Pénalisation exacte et globalisation par recherche linéaire de la méthode SQP

La méthode SQP décrite dans la section 3 de cette annexe converge si le premier itéré (x_1, λ_1) est assez proche d'un point stationnaire régulier (voir théorème 3.3). Comme un tel point n'est généralement pas disponible dans la plupart des applications, il est nécessaire d'utiliser des techniques dites de globalisation d'un algorithme local qui permettent de "forcer" la convergence même si le point de départ est loin d'une solution primale-duale de (P).

A l'heure actuelle, il existe deux grandes classes de techniques de globalisation :

* la recherche linéaire (RL),
* les régions de confiance (RC, cf. section 2 du chapitre 1).

Ces deux techniques de globalisation utilisent le même principe : elles mesurent le progrès effectué lors du passage de l'itération x_k à l'itération x_{k+1} par l'intermédiaire d'une fonction auxiliaire dite fonction de mérite. Pour le problème (P), cette fonction auxiliaire, doit non seulement prendre en compte la minimisation effective de f, mais aussi la satisfaction des contraintes. Pour ce faire, les méthodes d'optimisation utilisent souvent la fonction de pénalisation $\Theta_\sigma(x) = f(x) + \sigma p(x)$ comme fonction de mérite, où p est une fonction qui pénalise la violation des contraintes ($p(x) = 0$ si $x \in \mathcal{X}$ et $p(x) > 0$ si $x \notin \mathcal{X}$) et où $\sigma > 0$ est le facteur de pénalisation (voir section 2 de cette annexe).

Dans cette section, nous nous intéressons plus particulièrement à la fonction de pénalisation par norme générale définie par

$$\Theta_\sigma(x) = f(x) + \sigma ||c(x)^{\#}||_P, \tag{C.11}$$

où $||.||_P$ est une norme quelconque et $.^{\#} : \mathbb{R}^m \mapsto \mathbb{R}^m$ est la fonction définie par

$$(v^{\#})_i = \begin{cases} v_i & \text{if } \ i \in E \\ v_i^+ = \max(0, v_i) & \text{if } \ i \in I \end{cases} .$$

Nous nous limiterons aux résultats concernant la technique de globalisation de la méthode SQP par la méthode de recherche linéaire sur la fonction de pénalisation par norme Θ_σ. Dans cette méthode, on génère une suite de points $\{(x_k, \lambda_k)\}$ qui converge vers une solution primale-duale (x_*, λ_*) de (P). Cette suite est définie par la formule de récurrence

$$(x_{k+1}, \lambda_{k+1}) = (x_k, \lambda_k) + \alpha_k(d_k, \lambda_k^{QP} - \lambda_k),$$

où (d_k, λ_k^{QP}) est une solution primale-duale du PQT (C.9) et $\alpha_k > 0$ est le pas de la RL servant à faire décroître la fonction de mérite Θ_σ dans la direction d_k (ce pas est calculé par un algorithme de rebroussement). Cette approche est originale car elle utilise la solution du PQT (C.9) pour faire décroître la fonction de mérite Θ_σ et non pas une direction fondée sur un sous-gradient de la fonction non différentiable Θ_σ (ce qui aurait conduit à un algorithme moins rapide). La proposition suivante, reprise de la proposition 15.1 de [17], est essentielle : elle assure qu'il existe un $\sigma > 0$ tel que la direction primale d_k trouvée par la méthode SQP est bien une direction de descente de la fonction de mérite Θ_σ en x_k.

Proposition 4.1 *Si (d_k, λ_k^{QP}) satisfait les conditions d'optimalité (C.8), alors on a*

$$\Theta'_\sigma(x_k; d_k) \leq \nabla f_k^\top d_k - \sigma ||c_k^\#||_P = -d_k^\top M_k d_k + (\lambda_k^{QP})^\top c_k - \sigma ||c_k^\#||_P.$$

Si, en plus, $\sigma > ||\lambda_k^{QP}||_D$, alors on a

$$\Theta'_\sigma(x_k; d_k) \leq -d_k^\top M_k d_k.$$

En conclusion : $\Theta'_\sigma(x_k; d_k) < 0$, si $\sigma > ||\lambda_k^{QP}||_D$, si M_k est définie positive, et si x_k n'est pas un point stationnaire de (P).

Ainsi, à chaque itération k de l'approche décrite précédemment, il faut adapter la valeur de σ pour que d_k soit bien une direction de descente de la fonction de mérite Θ_σ (la fonction Θ_σ peut changer à chaque itération). Le concept de fonction de pénalisation exacte (voir définition 2.1) joue un rôle important dans la convergence de cette approche : il permet de stabiliser la valeur de σ. Voici ci-dessous une condition suffisante d'exactitude de Θ_σ reprise de la proposition 14.7 de [17] :

Proposition 4.2 *Supposons que f et $c_{E \cup I_*^0}$ sont deux fois différentiables en un minimum local x_* de (P) pour lequel la condition nécessaire d'optimalité (C.2) est satisfaite. En ce minimum x_* on suppose également que la condition suffisante d'optimalité du second ordre faible est respectée, et que*

$$\sigma > \sup_{\lambda_* \in \Lambda_*} ||\lambda_*||_D.$$

Alors, Θ_σ a un minimum local strict en x_*.

Annexe D

Optimisation globale par Krigeage

1 Modèle de Krigeage

Le principe du krigeage est de modéliser la fonction objectif f comme un processus stochastique :

$$Y(x) = \beta + Z(x) \tag{D.1}$$

avec $\beta \in \mathbb{R}$ la moyenne de Y sur le domaine de recherche et $Z(x)$ un processus stochastique tel que $\mathbb{E}(Z(x)) = 0$ et $Cov(Z(w), Z(x)) = \sigma^2 R(w, x)$. La fonction de corrélation $R(\cdot, \cdot)$ peut être écrite telle que :

$$R(w, x) = \prod_{j=1}^{k} \exp(-\theta_j |w_j - x_j|^{p_j}) \tag{D.2}$$

avec $\theta_j \geq 0$ et $0 < p_j \leq 2$. Les p_j peuvent être interprétés comme des paramètres de lissage (ou a priori sur f) et les θ_j indiquent si l'approximation est plutôt locale ou plutôt globale.

Le meilleur estimateur linéaire non-biaisé est alors :

$$\hat{y}(x) = \hat{\beta} + r^T(x) R^{-1}(y - 1\hat{\beta}) \tag{D.3}$$

où $r^T(x)$ est le vecteur de corrélations entre x et chacun des n points échantillonnés, $\hat{\beta}$ est l'estimateur des moindres carrés généralisés de β, R est la matrice de corrélation définie par l'équation (D.2) et 1 est un vecteur de un selon chaque coordonnée.

L'erreur moyenne quadratique peut être définie par

$$MSE[\hat{y}(x)] = \sigma_z^2 \left[1 - (1 r_x^t) \begin{pmatrix} 0 & 1^t \\ 1 & R \end{pmatrix} \begin{pmatrix} 1 \\ r_x \end{pmatrix} \right] \tag{D.4}$$

2 Algorithme d'optimisation par Krigeage

On présente ici une méthode d'optimisation globale d'une fonction objectif f coûteuse avec un budget d'évaluations limité. Cette limitation implique que chacune des évaluations doit être choisie de manière à améliorer le plus possible l'estimation du minimum de f. Il est donc plus avantageux de choisir chaque évaluation en fonction du résultat des précédentes. En effet, le principe commun aux méthodes d'optimisation reposant sur le krigeage est d'évaluer f de manière itérative en un point qui optimise un critère d'échantillonnage construit à l'aide des résultats d'évaluation déjà disponibles. Une approche simple serait d'utiliser un minimiseur de la prédiction par krigeage $Y(x)$ comme nouveau point à évaluer. Toutefois, cette approche accorderait une confiance trop grande à la prédiction qui n'est construite qu'à partir d'un nombre limité d'évaluations de f et l'optimisation risque de stagner en un minimum local.

Pour obtenir un compromis satisfaisant entre recherche locale et recherche globale, il est nécessaire d'utiliser l'erreur de prédiction (D.4) pour diriger la recherche vers les zones où la prédiction doit être améliorée. Cette idée a conduit à la création de critère d'échantillonnage utilisant simultanément la prédiction $\hat{y}$ et l'évaluation de l'erreur associée : la maximisation de l'espérance de gain (Expected Improvement [156, 155, 100, 99] classiquement utilisée) permet ainsi d'obtenir un bon compromis entre recherche locale et recherche globale.

Après avoir mis à jour l'estimateur linéaire non-biaisé de (D.3) à l'aide de tous les points échantillonnés à l'itération courante, on peut prédire $Y(x)$ en un point non-échantillonné x par $\hat{y}(x)$ avec une erreur moyenne quadratique donnée par (D.4).

Pour plus de simplicité, nous notons $\hat{\mu} = \hat{y}(x)$, et la racine de l'erreur quadratique moyenne $\hat{\sigma} = \sqrt{MSE}$ (d'où $Y(x) \sim \mathbb{N}(\hat{\mu}, \hat{\sigma}^2)$). Soit $f^* = \min_{x \in \mathbb{X}} f(x)$ le minimum global inconnu de f sur $\mathbb{X}$ et $\mathbb{S}_n = x_1, ..., x_n$ les n premiers points évalués et $f_{min} = min_{x_i \in \mathbb{S}_n} f(x_i)$ la valeur minimale de la fonction objectif aux points échantillonnés.

Le gain en un nouveau point x est défini par :

$$I(x) = \begin{cases} f_{min} - y & \text{si } y < f_{min} \\ 0 & \text{sinon} \end{cases}, \tag{D.5}$$

avec $y = f(x)$.

Alors l'espérance de gain est :

$$\mathbb{E}(I) = \begin{cases} (f_{min} - \hat{\mu})\Phi(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}) + \hat{\sigma}\Phi'(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}) & \text{si } \hat{\sigma} > 0 \\ 0 & \text{si } \hat{\sigma} = 0 \end{cases} \tag{D.6}$$

où Φ est la fonction de répartition d'une variable gaussienne centrée réduite.

D'autres critères d'échantillonnage existent, parmi eux, la variance de krigeage, l'espérance de gain généralisée (ci-dessous), l'entropie conditionnelle des minimiseurs [13, 12, 74, 172, 173, 174]. L'espérance de gain généralisée $\mathbb{E}(I^g)$ peut aussi être utilisée :

$$I^g = \begin{cases} (f_{min} - y)^g & \text{si } y < f_{min} \\ 0 & \text{sinon} \end{cases} \tag{D.7}$$

avec $g \in \mathbb{N}$. Dans ce cas, plus g est grand, plus la recherche est globale et à l'inverse plus g est proche de zéro, plus la recherche est locale.

L'algorithme EGO (Efficient Global Optimization), basé sur l'algorithme 1.6 proposé par Jones et al. ([100] 1998), choisit comme nouveau point d'évaluation un maximiseur d'espérance de gain, et met à jour les paramètres de la covariance ainsi que la prédiction par krigeage après chaque nouvelle évaluation. Notons qu'une recherche globale du point d'espérance de gain maximale est effectuée à chaque itération, ce qui est un problème d'optimisation difficile en grande dimension.

Annexe E

Initialisation de la méthode SQA

Afin de pouvoir démarrer l'optimisation de la fonction objectif sous contraintes, la méthode SQA nécessite de tester certaines propriétés : le point inital doit vérifier chacune des contraintes avec dérivées et les autres points d'interpolation de l'étape d'initialisation ont besoin de vérifier chacune des contraintes dures.

Les m points initiaux sont placés autour du point de départ $x_0 \in \mathbb{R}^n$ (choisis lors de l'étape d'initialisation décrite par la suite) de la façon suivante.

- Lorsque le nombre de conditions d'interpolation satisfait $m \geq 2n+1$, les $2n+1$ premiers points $x_i, i = 1, 2, ..., m$ sont choisis pour être les vecteurs

$$x_1 = x_0 \quad et \quad \left.\begin{array}{r} x_{i+1} = x_0 + \rho_{beg} e_i \\ x_{i+n+1} = x_0 - \rho_{beg} e_i \end{array}\right\}, \quad i = 1, 2, ..., n, \tag{E.1}$$

  où e_i est un vecteur de $\mathbb{R}^n$ nul selon chaque composante exceptée la i-ème composante qui est égale à un. Ainsi $Q(x_0)$, $\nabla Q(x_0)$ et les éléments diagonaux $(\nabla^2 Q)_{ii}$, $i = 1, 2, ..., n$, sont donnés par les $2n+1$ premières équations (3.6). Si $m = 2n+1$, tous les éléments non diagonaux de $\nabla^2 Q$ sont fixés à zéro, ce qui complète le modèle quadratique initial (3.10).
- Dans le cas où $n+2 \leq m \leq 2n$, les points d'interpolation initiaux sont les m premiers vecteurs de (E.1). On peut alors déterminer $Q(x_0)$, les $m-n-1$ premières composantes de $\nabla Q(x_0)$ et $(\nabla^2 Q)_{ii}$, $i = 1, 2, ..., m-n-1$ définis comme précédemment. Les autres éléments diagonaux de $\nabla^2 Q$ sont fixés à zéro, ainsi les autres composantes de $\nabla Q(x_0)$ prennent les valeurs $(F(x_0 + \Delta_0 e_i) - F(x_0))/\Delta_0$, pour $m - n \leq i \leq n$.
- Dans le cas où $m > 2n+1$, les points initiaux x_i, $i = 1, 2, ..., m$, sont choisis de manière à ce que les conditions (3.6) fournissent aussi $2(m-2n-1)$ éléments non diagonaux de $\nabla^2 Q$ (le facteur 2 provient de la symétrie de la matrice Hessienne du modèle). Spécifiquement, pour $i \in [2n+2, m]$, le point x_i a la forme

$$x_i = x_0 + \sigma_p \Delta_0 e_p + \sigma_q \Delta_0 e_q, \tag{E.2}$$

  où p et q sont des entiers différents de $[1, n]$, et σ_p et σ_q vérifient

$$\sigma_j = \left.\begin{array}{rl} -1 & F(x_0 - \Delta_0 e_j) < F(x_0 + \Delta_0 e_j) \\ +1 & F(x_0 - \Delta_0 e_j) \geq F(x_0 + \Delta_0 e_j), \end{array}\right\} \quad j = 1, 2, ..., n, \tag{E.3}$$

ce qui rapproche (E.2) vers les valeurs de la fonction objectif les plus petites. Les éléments p et q de la formule (E.2) sont choisis de la manière suivante. Soit j, la partie entière du quotient $(i-n-2)/n$ qui satisfait $j \geq 1$ (car $i \geq 2n+2$), nous fixons $p = i-n-1-jn$ (dans l'intervalle $[1, n]$) et nous fixons q à $p+j$ ou $p+j-n$ (le second choix dans le cas $p+j > n$). Ainsi les éléments $(\nabla^2 Q)_{pq} = (\nabla^2 Q)_{qp}$ sont donnés par les équations (3.6), et les éléments non diagonaux de $\nabla^2 Q$ qui ne sont pas fournis sont fixés à zéro.

Par conséquent, les paramètres ρ_{beg}, rayon de la région de confiance initiale et x_0 point de départ peuvent être modifiés afin de vérifier ces propriétés.

(1) Si une faible réduction de ρ_{beg} permet à tous les points initiaux de vérifier les contraintes dures, celle-ci est effectuée,

(2) dans le cas contraire, le point de départ x_0 est déplacé pour permettre aux m points initiaux de vérifier les contraintes dures, tout en vérifiant lui-même toutes les contraintes avec dérivées.

La première étape consiste donc à trouver un point initial, vérifiant les contraintes avec dérivées, le plus proche possible du point de départ donné par l'utilisateur de la méthode x_{user}.

$$\min_{x \in \mathbb{R}^n} (x - x_{user})^2 \tag{E.4}$$

sous les contraintes

$$\begin{cases} C_{DB_h}(x) \leq 0, \\ C_{DB_s}(x) \leq 0, \end{cases} \tag{E.5}$$

Ensuite, la seconde étape consiste à trouver le plus grand rayon de région de confiance Δ (inférieur à ρ_{beg}) permettant aux $m-1$ autres points initiaux de vérifier les contraintes dures. Pour ce faire,

$$\min_{\Delta \in \mathbb{R}} -\Delta \tag{E.6}$$

sous les contraintes

$$\left\{ \ 0 \leq \Delta \leq \rho_{beg} C_{DB_h}(x_i) \leq 0, \ \text{pour tout } x_i, i = 1, ..., m-1, \right. \tag{E.7}$$

où $x_i, i = 2, ..., m$ sont les points initiaux autour de x_0.

Si le rayon de la région de confiance est inférieur à $\rho_{beg}/10$, une troisième étape consistant à trouver un point de départ x_0 (le plus proche possible du point obtenu lors de la première étape) admissible pour chacune des contraintes avec dérivées et de vérifier les contraintes dures pour les $m-1$ autres points initiaux.

$$\min_{x \in \mathbb{R}^n, \Delta \in \mathbb{R}} (x - x_{user})^2 \tag{E.8}$$

sous les contraintes

$$\begin{cases} \rho_{beg}/10 \leq \Delta \leq \rho_{beg}, \\ C_{DB_h}(x) \leq 0, \\ C_{DB_s}(x) \leq 0, \\ C_{DB_h}(x_i) \leq 0, \text{ pour tout } x_i, i = 2, ..., m, \end{cases} \tag{E.9}$$

Si une solution à ce problème existe, la méthode SQA peut alors commencer à résoudre le problème d'optimisation à partir de ce nouveau point initial et dans cette nouvelle région de confiance.

Annexe F

Résultats obtenus sur le benchmark Moré et Wild sous contraintes de bornes de taille inférieure à 13

Les résultats suivants, obtenus par notre méthode SQA (voir chapitre 3) sur l'ensemble de problèmes avec des contraintes de bornes issus du benchmark CUTEr [121] de dimension inférieure à 13 sont présentés et comparés (comme dans la section 1.1 du chapitre 4) avec d'autres méthodes à l'aide de profils de performances et de profils de données (voir annexe A) pour différentes précisions $\tau > 0$. Ces problèmes peuvent être lisses, bruités ou non différentiables.

Les méthodes d'optimisation ainsi comparées sont la méthode d'optimisation SQA présentée au chapitre 3 avec des modèles construits à partir de différents nombres de points d'interpolation ($n+2$, $n+6$, $2n+1$ et $(n+1)(n+2)/2$) ainsi que trois méthodes d'optimisation sans dérivées par région de confiance permettant de prendre en compte des contraintes de bornes utilisant un modèle local plus ou moins riche. La méthode COBYLA [133] construit un modèle d'interpolation linéaire basé sur $n+1$ points d'interpolation, la méthode BOBYQA [142] construit, comme la méthode SQA, un modèle d'interpolation, pouvant être aussi bien sous déterminé que complet, celui-ci est basé sur un nombre fixe de point d'interpolation, maintenu dans notre cas à $2n+1$. Enfin la méthode CONDOR [167, 169, 168] construit un modèle quadratique complet à partir de $(n+1)(n+2)/2$ points d'interpolation.

1 Problèmes lisses

Les profils de données de la figure F.1 montrent que les résultats de la méthode SQA $(n+1)(n+2)/2$ et de la méthode CONDOR $(n+1)(n+2)/2$ sont similaires pour chacune des précisions et pour tous les budgets d'évaluations de f. Nous remarquons également que la méthode SQA résout légérement plus de problèmes que la méthode CONDOR pour un budget de $100(n_p+1)$ évaluations pour toutes précisions τ.

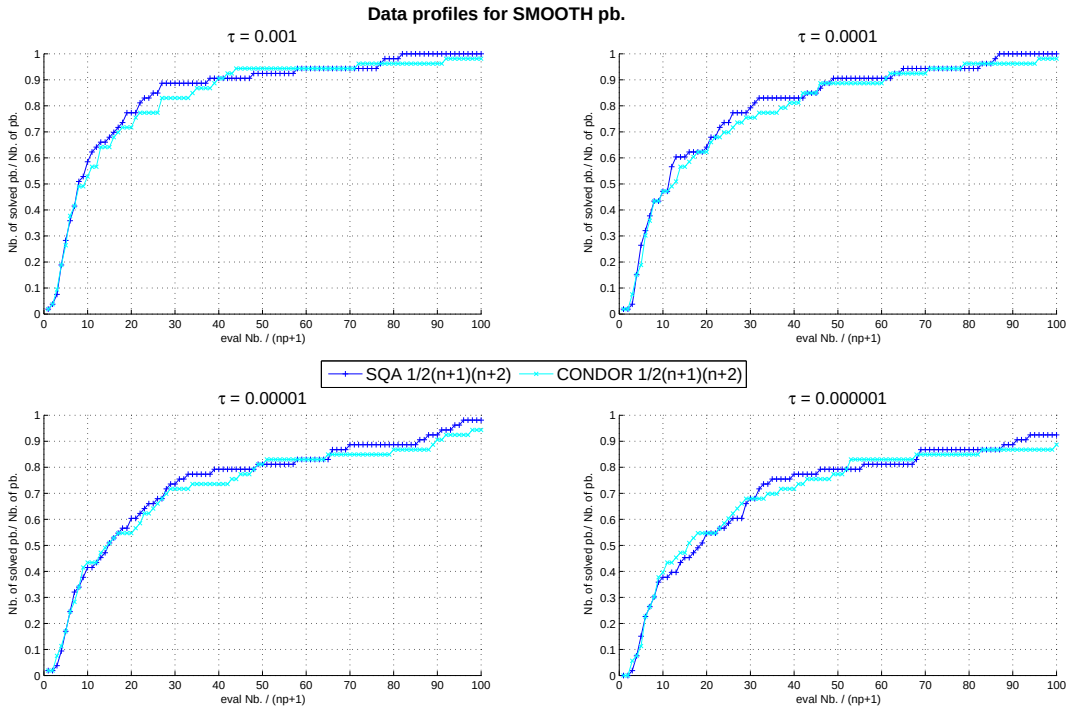


FIGURE F.1 – Profils de données (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performances de la figure F.2 indiquent que, pour toutes précisions τ, la méthode SQA est plus rapide que CONDOR pour au moins 60% des problèmes. De plus, la méthode SQA est également plus robuste que CONDOR : en effet, elle permet de résoudre pratiquement la totalité des problèmes pour toutes précisions τ.

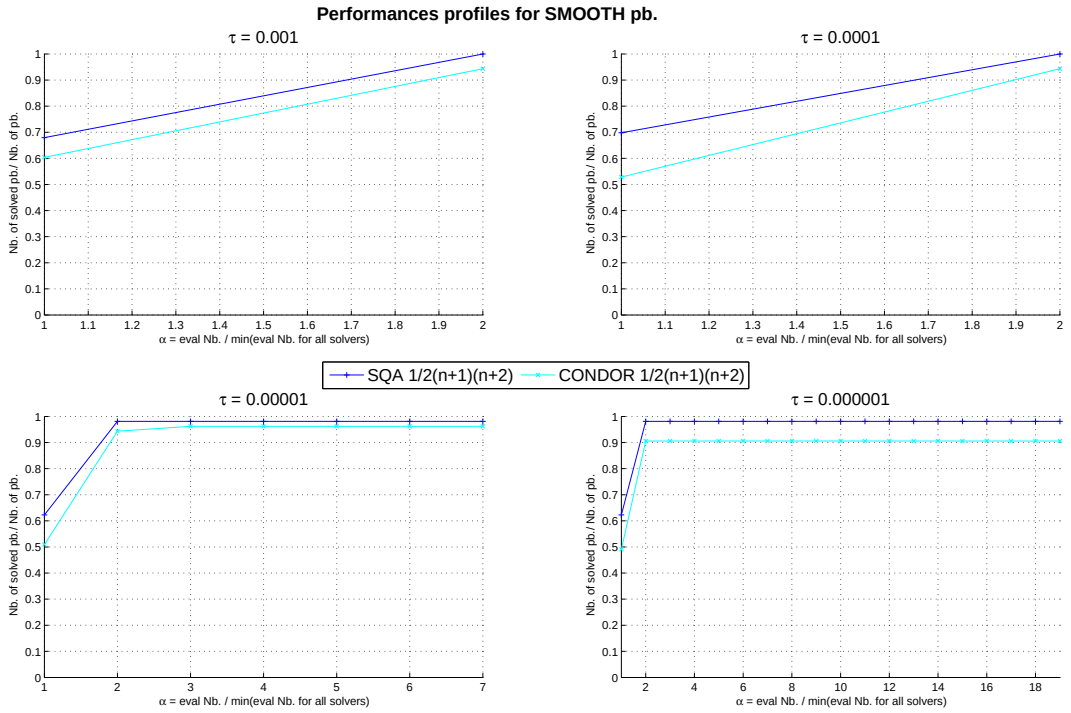


FIGURE F.2 – Profils de performances (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de données de la figure F.3 montrent que les résultats de la méthode SQA $2n+1$ et la de la méthode BOBYQA $2n+1$ sont similaires pour chacunes des précisions et pour tout budget d'évaluations de f.

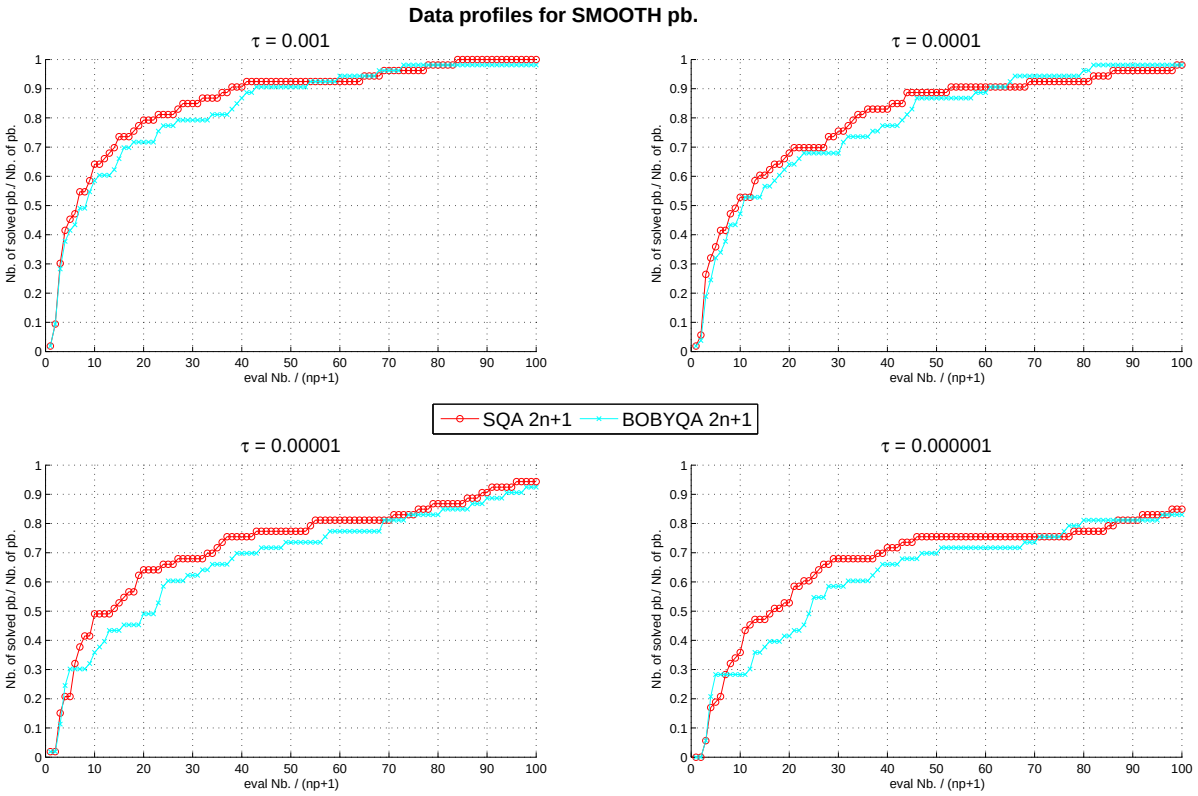


FIGURE F.3 – Profils de données (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performances de la figure F.4 indiquent que pour toutes précisions τ, la méthode SQA est plus rapide que BOBYQA pour au moins 60% des problèmes. La méthode SQA est également plus robuste, en effet, celle-ci permet de résoudre un plus grand pourcentage de problèmes que la méthode BOBYQA (plus de 95% des problèmes).

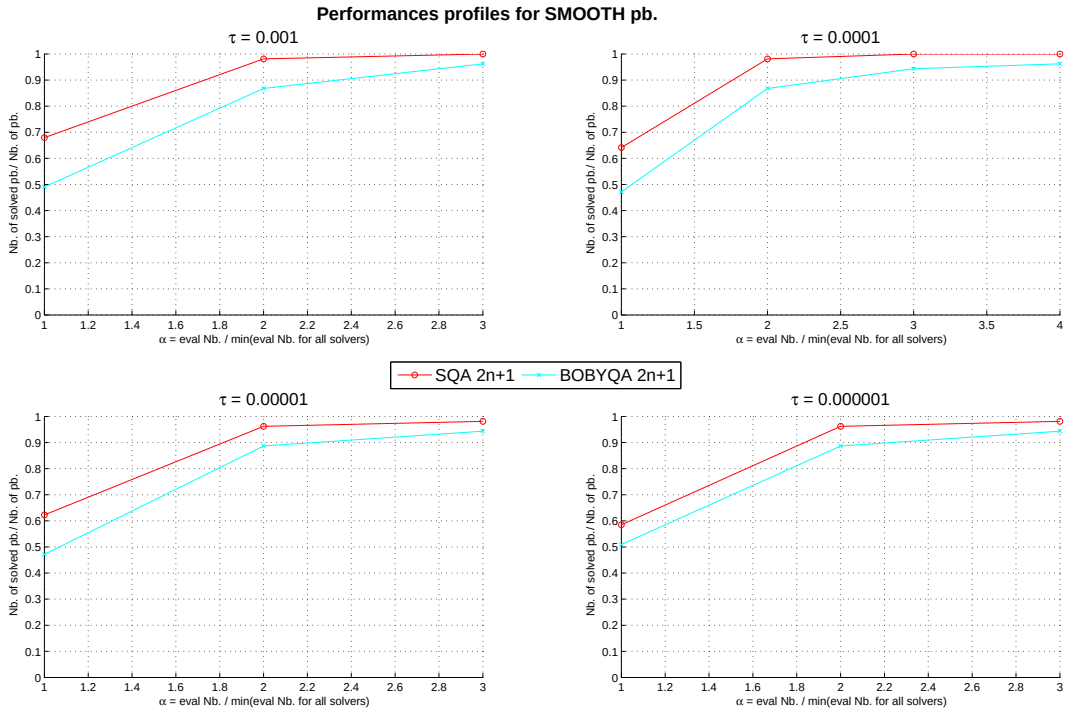


FIGURE F.4 – Profils de performances (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de données de la figure F.5 montrent que les résultats de la méthode SQA $n+6$ et $n+2$ et de la méthode COBYLA $n+1$ sont assez différents. En effet, la méthode COBYLA permet de résoudre un plus grand pourcentage de problèmes pour de petits budgets d'évaluations. Pour des précisions τ de 10^{-3}, 10^{-4}, 10^{-5} et 10^{-6}, la méthode COBYLA permet de résoudre un plus grand pourcentage de problèmes jusqu'à respectivement $10(n_p+1)$, $15(n_p+1)$, $25(n_p+1)$ et $40(n_p+1)$. La méthode COBYLA permet de résoudre jusqu'à 15% de problèmes de plus que SQA $n+6$ et jusqu'à 20% de problèmes de plus que SQA $n+2$ pour de petits budgets d'évaluations. En revanche pour un budget de $100(n_p+1)$, SQA $n+6$ résout 20% de problèmes de plus que COBYLA. La méthode SQA $n+2$ résout, elle, moins de problèmes que la méthode SQA $n+6$ pour un budget de $100(n_p+1)$: plus la précision demandée est importante, plus le pourcentage de problèmes résolus s'éloigne de celui de SQA $n+6$ et se rapproche de celui de COBYLA.

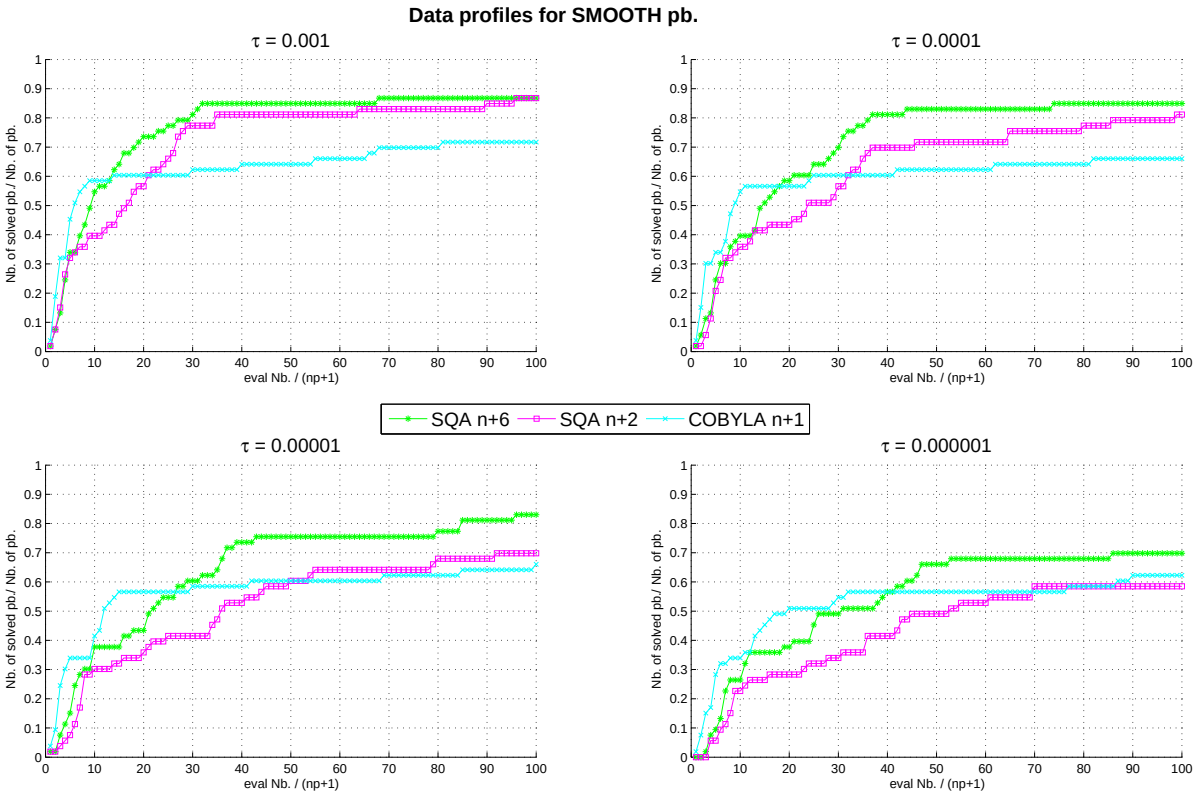


FIGURE F.5 – Profils de données (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performances de la figure F.6 indiquent que pour toutes précisions les méthodes SQA $n+6$ et COBYLA sont plus rapides que la méthode SQA $n+2$ pour 40 à 50 % des problèmes. La méthode SQA $n+2$ n'est la plus rapide des méthodes que sur moins de 20 % des problèmes. Nous remarquons également que les méthodes SQA $n+6$ et $n+2$ sont capables de résoudre un plus grand pourcentage de problèmes que la méthode COBYLA pour des précisions importantes : environ 85 % des problèmes sont résolus par SQA $n+6$ et $n+2$ pour une précision de 10^{-6}, alors que COBYLA ne résout que 75 % des problèmes.

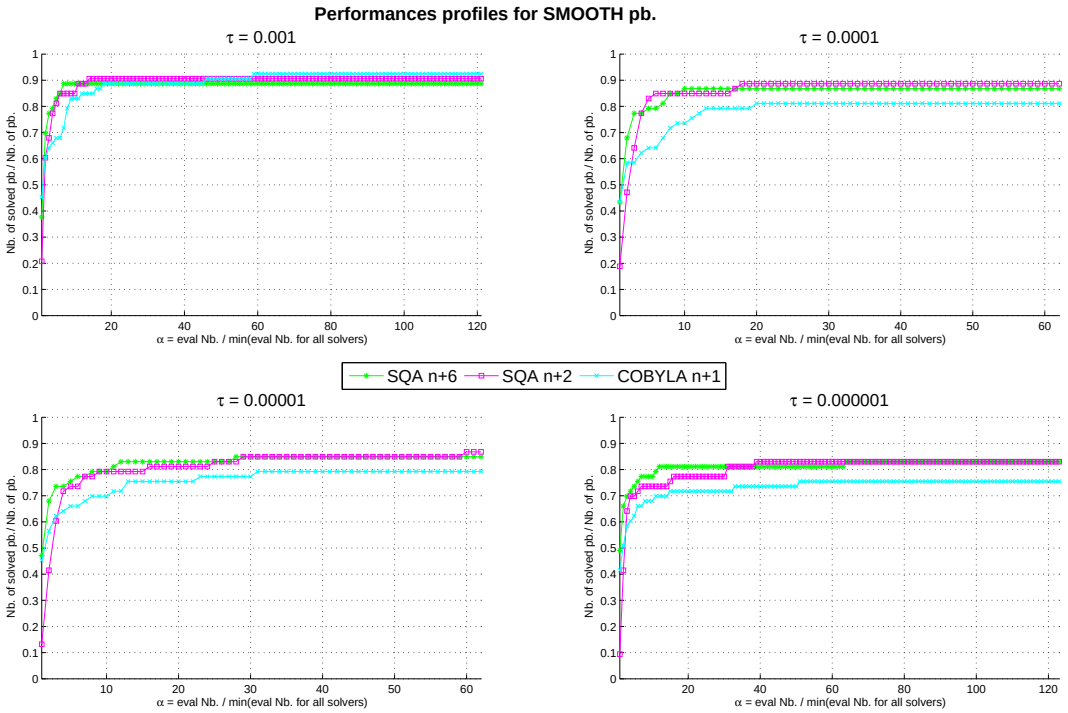


FIGURE F.6 – Profils de performances (voir annexe A) pour les problèmes lisses du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

2 Problèmes bruités

Les profils de données de la figure F.7 montrent que plus le modèle construit par chacune des méthodes est précis, plus le pourcentage de problèmes résolus pour $100(np+1)$ simulations est grand. Ainsi les méthodes SQA $(n+1)(n+2)/2$ et CONDOR permettent de résoudre plus de problèmes que SQA $2n+1$ et BOBYQA $2n+1$, qui permettent de résourdre plus de problèmes que COBYLA $n+1$, SQA $n+6$ et SQA $n+2$. En revanche, lorsque le budget d'évaluations est bien moins élevé, les meilleurs solveurs sont à l'inverse les solveurs ayant un modèle basé sur peu d'évaluations. C'est ce que nous avons remarqué sur les figures 4.5 et 4.6 du chapitre 4. De plus, nous remarquons comme pour les problèmes lisses que les performances de SQA $(n+1)(n+2)/2$ sont similaires à celles de CONDOR $(n+1)(n+2)/2$, et que les performances de SQA $2n+1$ et BOBYQA $2n+1$ le sont également. Les résultats obtenus par COBYLA sont, contrairement aux problèmes lisses, plus rapides et plus précis pour résoudre les problèmes bruités que SQA $n+6$ et SQA $n+2$.

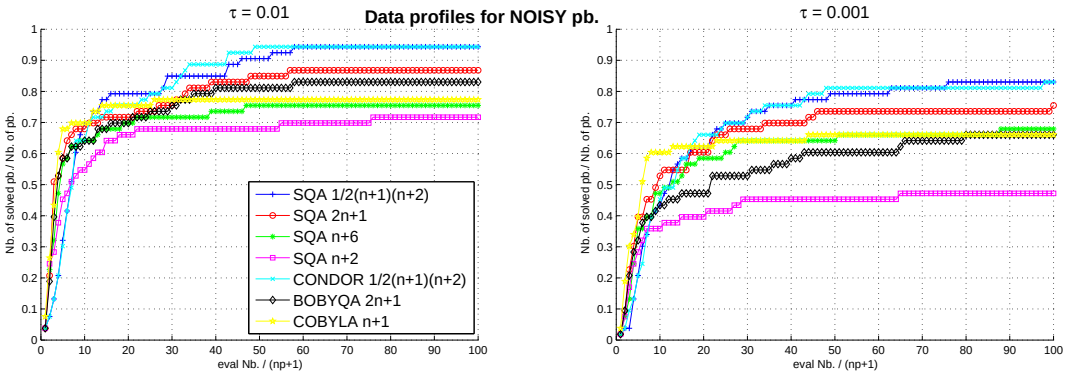


FIGURE F.7 – Profils de données (voir annexe A) pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

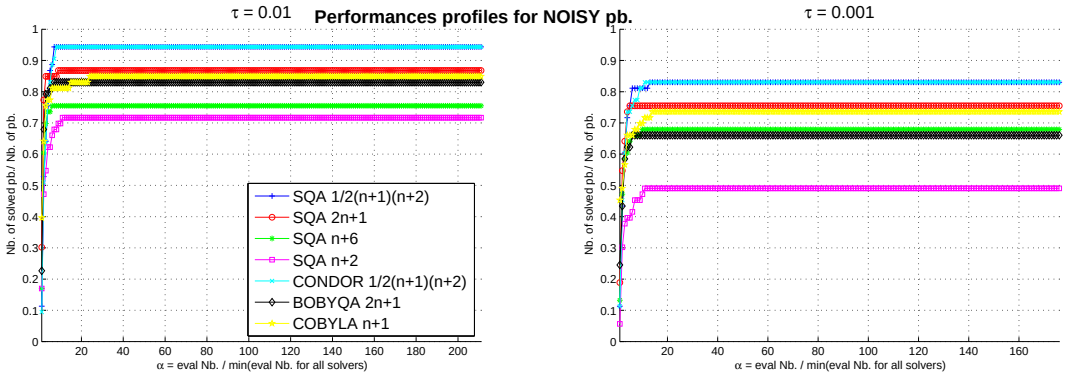


FIGURE F.8 – Profils de performances (voir annexe A) pour les problèmes bruités du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performances de la figure F.8 nous indiquent que les performances de la méthode SQA $(n+1)(n+2)/2$ et de la méthode CONDOR $(n+1)(n+2)/2$ sont équivalentes pour toutes les précisions τ des problèmes bruités. Elles sont plus rapides que les autres méthodes pour 10 % des problèmes, et ce sont les plus robustes avec 85 % des problèmes résolus. Nous pouvons également remarquer que les méthodes SQA $2n+1$ et BOBYQA $2n+1$ obtiennent des résultats comparables. Elles sont plus rapides que les autres méthodes pour 25 % des problèmes, et SQA $2n+1$ est légèrement plus robuste que BOBYQA (75 % contre 65 % de problèmes résolus pour une précision $\tau = 10^{-3}$). Les résultats indiquent que la méthode COBYLA $n+1$ est la méthode permettant de résoudre le plus grand pourcentage de problèmes plus rapidement que les autres méthodes : 45 %

des problèmes (pour $\tau = 10^{-3}$) contre 15 % pour SQA $n+6$ et 5 % pour SQA $n+2$. La méthode COBYLA est plus robuste que les méthodes SQA $n+6$ et $n+2$: 75 % de problèmes résolus pour COBYLA contre 65 % pour SQA $n+6$ et 50 % pour SQA $n+2$.

3 Problèmes non différentiables

Les profils de données de la figure F.9 montrent que les résultats obtenus sur les problèmes non différentiables par SQA $(n+1)(n+2)/2$ sont similaires à ceux obtenus par CONDOR ; cependant CONDOR résout 10 à 20 % de problèmes supplémentaires pour un budget de $100(np+1)$ simulations. De façon analogue, les résultats obtenus par SQA $2n+1$ et BOBYQA sont similaires pour toutes les précisions. La méthode COBYLA semble plus efficace que les méthodes SQA $n+6$ et SQA $n+2$: celle-ci résout 10 % de problèmes supplémentaires pour une précision $\tau = 10^{-6}$.

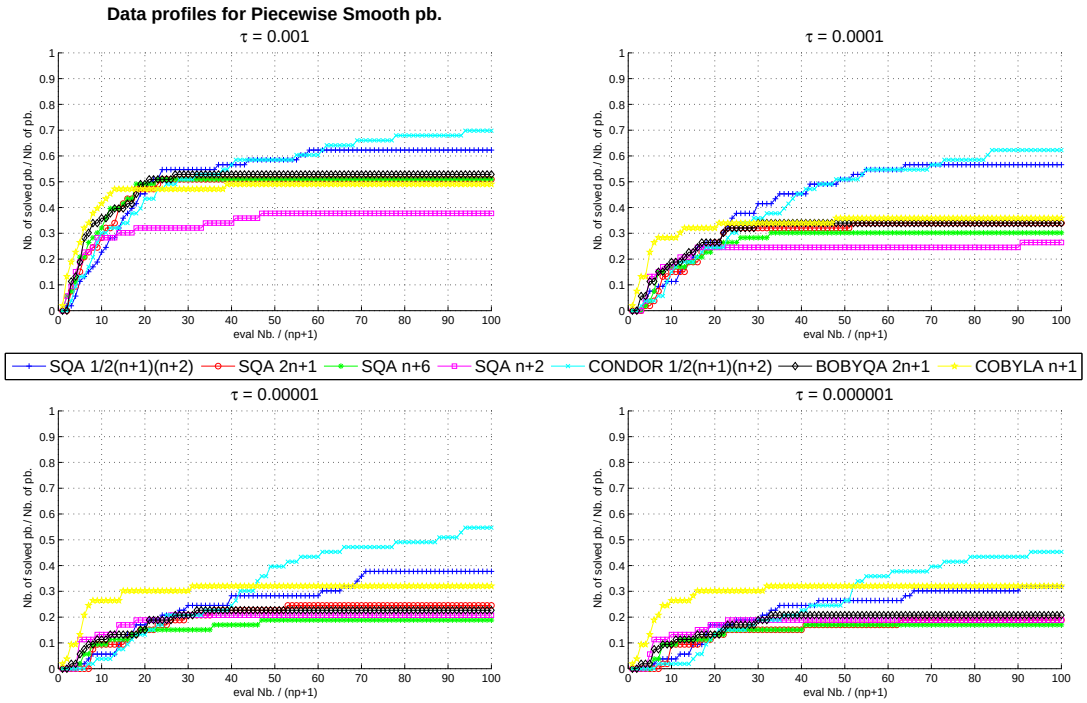


FIGURE F.9 – Profils de données (voir annexe A) pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Les profils de performance de la figure F.10 indiquent que la méthode CONDOR $(n+1)(n+2)/2$ permet de résoudre 10 à 20 % de problèmes supplémentaires que la méthode SQA $(n+1)(n+2)/2$. C'est la méthode la plus rapide sur le plus grand pourcentage de problèmes. Les résultats obtenus par SQA $2n+1$ et BOBYQA $2n+1$ sont identiques. La

méthode COBYLA est plus robuste et plus rapide que les méthodes SQA $n+6$ et SQA $n+2$ sur un plus grand pourcentage de problèmes.

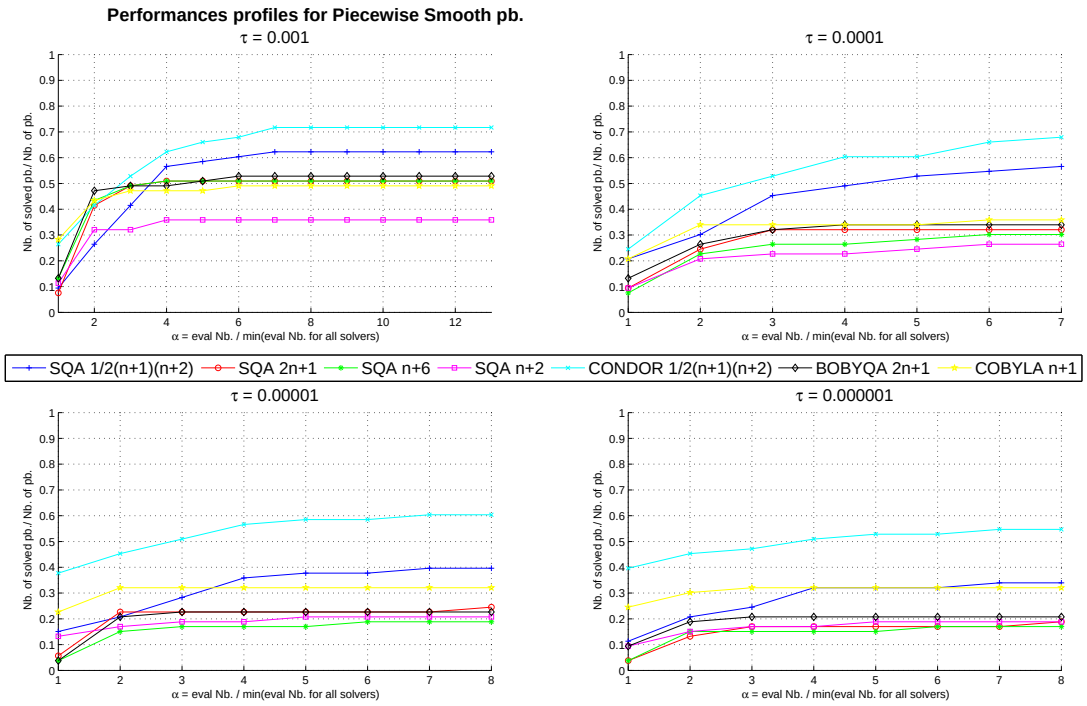


FIGURE F.10 – Profils de performances (voir annexe A) pour les problèmes non différentiables du benchmark sous contraintes de bornes de taille inférieure à 13 pour différentes précisions τ

Sur ces exemples lisses sous contraintes de bornes, la méthode SQA se comporte de manière satisfaisante par rapport aux algorithmes de mêmes type. La méthode COBYLA, qui n'utilise qu'un modèle linéaire est plus rapide que la méthode SQA $n+6$ ou SQA $n+2$ pour de petits budgets d'évaluations, en revanche c'est l'inverse pour de gros budgets d'évaluations.

Sur les problèmes bruités, les méthodes utilisant les modèles les plus précis de la fonction objectif (polynôme quadratique complet) sont plus robustes. A l'inverse, la méthode COBYLA résout plus de problèmes pour de petits budgets d'évaluations et est plus efficace sur un plus grand pourcentage de problèmes que les autres méthodes. Celle-ci ne subit pas autant l'influence du bruit que les autres méthodes.

Sur les problèmes non différentiables, la méthode la plus robuste est la méthode CONDOR $(n+1)(n+2)/2$ suivie de la méthode SQA $(n+1)(n+2)/2$. Les méthodes utilisant les modèles les plus précis sont les plus robustes, malgré la non différentiabilité (excepté COBYLA).

Annexe G

Distribution normale

1 Distribution normale à une dimension

La densité de probabilité f d'une variable aléatoire réelle $x \in \mathbb{R}$ normale ou gaussienne a pour équation : $f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(-\frac{(x-m)^2}{2 \cdot \sigma^2}\right)$.

L'espérance mathématique ou moyenne de f est égale à m, la variance à σ^2. Dans le cas particulier où la moyenne est nulle et la variance égale à l'unité, la distribution est dite centrée réduite.

Ainsi, la probabilité que x prenne des valeurs comprises entre $-a$ et a ($x \in [-a, a]$) est donnée par : $P = \mathbb{P}(|x - m| \leq a) = \displaystyle\int_{-a:\sigma}^{a/\sigma} \phi(x)dx$, où ϕ est la distribution normale centrée réduite.

Pour la valeur particulière $a = 1.96\sigma$, P vaut 0.9544. L'intervalle $[-1.96\sigma; 1.96\sigma]$ est appelé domaine de confiance à 95%.

2 Distribution normale à N dimensions

On appelle distribution normale à N dimensions la densité de probabilité associée à un vecteur aléatoire $x \in \mathbb{R}^N$ dont chacune des composantes suit une loi normale. Elle est caractérisée par une matrice symétrique définie positive appelée matrice de covariance. Les éléments diagonaux de la matrice sont appelés coefficients de corrélation entre les composantes constituant le vecteur aléatoire. La fonction densité f caractérisant la distribution normale d'un vecteur aléatoire $x \in \mathbb{R}^N$ donnée par la matrice de covariance C est définie par :

$$f(x) = \frac{\exp\left(-\frac{1}{2} \cdot (x-m)^T \cdot C^{-1} \cdot (x-m)\right)}{\sqrt{(2 \cdot \pi)^N \cdot \det(C)}} \tag{G.1}$$

où $\det(C)$ désigne le déterminant de la matrice C et m est un vecteur de $\mathbb{R}$ dont les composantes $(m(i))_{i=1,\ldots,N}$ sont égales à l'espérance mathématique des variables aléatoires $(x(i))_{i=1,\ldots,N}$.

La détermination du domaine de confiance à 95% est généralisable au cas à N dimensions. Dans ce qui suit, nous présentons le calcul dans le cas bidimensionnel.

Sans perte de généralité, nous pouvons supposer la matrice C diagonale. Ainsi, l'équation G.1 devient : $f(x,y) = \dfrac{1}{2 \cdot \pi \cdot \sigma_1 \cdot \sigma_2} \cdot \exp\left(-\dfrac{1}{2} \cdot \left(\dfrac{x^2}{\sigma_1^2} + \dfrac{y^2}{\sigma_2^2}\right)\right)$, où σ_1^2 et σ_2^2 sont les valeurs propres de la matrice C.

Afin de généraliser la notion de domaine de confiance au cas bidimensionnel, nous cherchons un ensemble Ω_w inclus dans $\mathbb{R}^2$ tel que : $\displaystyle\int\int_{(x,y)\in\Omega_w} f(x,y)dxdy = 0.95$. Plus précisément, l'ensemble Ω_w est défini par : $\Omega_w = z = (x,y) \in \mathbb{R}^2$ tels que $z^T \cdot C^{-1} \cdot z \in [-w; w]$, où la valeur de w est à déterminer.

Après avoir effectué le changement de variables suivant : $\begin{cases} x = r \cdot \sqrt{w} \cdot \sigma_1 \cdot \cos(\theta) \\ y = r \cdot \sqrt{w} \cdot \sigma_2 \cdot \sin(\theta) \end{cases}$, où $r \in [0;1]$ et $\theta \in [0; 2 \cdot \pi]$, l'intégrale est égale après calculs à : $\displaystyle\int\int_{(x,y)\in\Omega_w} f(x,y)dxdy = 1 - \exp\left(-\frac{w}{2}\right)$.

L'intégrale précédente prend alors la valeur 0.95 lorsque w est égal à $2 \cdot \ln(20) \approx 6$.

Ainsi, dans le cas bidimensionnel, le domaine de confiance Ω_w à 95% est obtenu lorsque w est égal à 6.

La définition d'un domaine de confiance dans le cas bidimensionnel permet en effet d'avoir une interprétation géométrique du fonctionnement du MO-CMA-ES présenté dans ce rapport.

Bibliographie

[1] Akteke, B. Derivative free optimization methods : Application in stirrer configuration and data clustering. Master's thesis, MIDDLE EAST TECHNICAL UNIVERSITY, 2005.

[2] Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V. A trust region framework for managing the use of approximation models in optimization. *Structural Optimization 15* (1998), 16–23.

[3] Audet, C., and Dennis, J. E. Analysis of generalized pattern searches. *SIAM J. Optim. 13* (2003), 889–903.

[4] Auger, A., Hansen, N., Perez Zerpa, J. M., Ros, R., and Schoenauer, M. Empirical comparisons of several derivative free optimization algorithms.

[5] Auger, A., Hansen, N., Perez Zerpa, J. M., Ros, R., and Schoenauer, M. Experimental comparisons of derivative free optimization algorithms. *8th International Symposium on Experimental Algorithms* (2009).

[6] Bakr, M. H., Bandler, J. W., Madsen, K., and Sondergaard, J. Review of the space mapping approach to engineering optimization and modeling. *Optimization and Engineering 1*, 3 (2000), 241–276.

[7] Bandler, J. W., Biernacki, R. M., Chen, S. H., Grobelny, P. A., and Hemmers, R. H. Space mapping technique for electromagnetic optimization. *IEEE Trans. Microwave Theory Tech. 42* (1994), 2536–2544.

[8] Bandler, J. W., Cheng, Q. S., Dakroury, S., Mohamed, A. S., Bakr, M. H., Madsen, K., and Sondergaard, J. Space mapping : The state of the art. *IEEE Trans. Microwave Theory Tech. 52* (2004), 337–361.

[9] Barichard, V. *Approches Hybrides Pour Les Problèmes multiobjectifs.* PhD thesis, Laboratoire d'Etude et de Recherche en Informatique d'Angers, Université d'Angers, France, 2003.

[10] Basseur, M. *Design of cooperative metaheuristics for multiobjective optimization : Application to the FlowShop scheduling Problem.* PhD thesis, Université des Sciences et Technologies de Lille, France, 2005.

[11] Berro, A. *Optimisation Multiobjectif et Stratégies d'Evolution en Environnement Dynamique.* PhD thesis, Université des Sciences Sociales Toulouse I, Toulouse, France, 2001.

[12] Bettinger, R. *Inversion d'un système par krigeage appliquée à la synthèse de catalyseurs à haut débit.* PhD thesis, université de Nice-Sophia Antipolis, 2009.

[13] Bettinger, R., Duchene, P., Pronzato, L., and Thierry, E. Design of experiments for response diversity. In *Journal of Physics : Conference Series* (2008), In Proc. 6th International Conference on Inverse Problems in Engineering (ICIPE), Dourdan (Paris).

[14] Beyer, H.-G., and Schwefel, H.-P. Evolution strategies : A comprehensive introduction. *Natural Computing 1* (2002), 3–52.

[15] Bilchev, G., and armee, P. The ant colony metaphor for searching continuous design spaces. *Lecture Notes in Computer Science 993* (1995), 25–39.

[16] Björkman, M., and Holmström, K. Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering 1*, 4 (2000), 373–397.

[17] Bonnans, J. F., Gilbert, J. C., Lemarechal, C., and Sagastizabal, C. A. *Numerical Optimization : Theoretical And Practical Aspects.* Springer-Verlag, 2003.

[18] Booker, A. J., J.E. Dennis, J., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization 17*, 98-47 (November 1998), 1–13.

[19] Bouzarkouna, Z., and Ding, D. and Auger, A. Using evolution strategy with meta-models for well placement optimization. In *In Proceedings of the 12th European Conference on the Mathematics of Oil Recovery ECMOR 2010* (2010).

[20] Box, G. E. P. Evolutionary operation : A method for increasing industrial productivity. *Appli. Statist. 6* (1957), 81–101.

[21] Box, G. E. P., and Wilson, K. B. On the experimental attainment of optimum conditions. *Roy. Statist. Soc. Ser. B. 13* (1951), 1–45.

[22] Casse, M. *Sparse Grid : de l'approximation des EDP au calcul de sensibilité.* PhD thesis, Institut de Mathématiques de Toulouse, 2010.

[23] Castagné, M., Bentolila, Y., Chaudoye, F., Hallé, A., Nicolas, F., and Sinoquet, D. Comparison of engine calibration methods based on design of experiments (doe). *Oil & Gas Science and Technology 63* (2008), 563–582.

[24] Chaudroye, F., Castagne, M., Sinoquet, D., and Wahl, F. Modelling engine operating space for doe calibration methods. In *5th Conference Design fo Experiments (DoE) in Engine Development, Berlin (Allemagne), 26 juin 2009* (2009).

[25] Colorni, A., Dorigo, M., and Maniezzo, V. Distributed optimization by ant colonies. *actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing* (1991), 134–142.

[26] Colson, B. *Trust-regions algorithms for derivative-free optimization and nonlinear bilevel programming.* PhD thesis, The University of Namur, 2003.

[27] Colson, B., and Toint, P. L. Optimizing partially separable functions without derivatives. *Optimization Methods and Software Vol. 20* (2005), 493–508.

[28] Conn, A. R. Bilevel derivative-free optimization and its application to robust optimization. In *IBM Research, Toulouse* (2010).

[29] Conn, A. R., Gould, N. I. M., and Toint, P. L. *Trust-Region Methods.* MPS-SIAM Series on Optimization, SOIAM Philadelphia, 2000.

[30] Conn, A. R., and Le Digabel, S. Use of quadratic models with mesh adaptive direct search for constrained black box optimization.

[31] Conn, A. R., Scheinberg, K., and Toint, P. L. On the convergence of derivative-free methods for unconstrained optimization. *A. Iserles and M. Buhmann, eds., Approximation Theory and Optimization : Tributes to M. J. D. Powell* (1997), 83–108.

[32] Conn, A. R., Scheinberg, K., and Toint, P. L. A derivative free optimization algorithm in practice. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (St Louis, MO, September 1998).

[33] Conn, A. R., Scheinberg, K., and Vicente, L. N. Global convergence of general derivative-free trust-region algorithms to first and second order critical points. *SIAM J. Optim.* (2009).

[34] Conn, A. R., Scheinberg, K., and Vicente, L. N. *Introduction to Derivative-Free Optimization.* Society for Industrial and Applied Mathematics (MPS-SIAM Series on Optimization), Philadelphia, PA, USA, 2009.

[35] Conn, A. R., and Toint, P. L. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In *Nonlinear Optimization and Applications* (Plenum, New York, 1996), e. In G. Di Pillo et F. Gianessi, Ed., pp. 27–47.

[36] Contou-Carrere, P. Algorithmes d'optimisation globale efficaces pour des fonctions fortement non linéaires. *Rapport de stage IFP* (2006).

[37] Courant, R. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society 49* (1943), 1–23.

[38] Darwin, C. *On the Origin of Species by Means of Natural Selection.* John Murray, 1859.

[39] Das, I., and Dennis, J. A closer look at drawbacks of minimizing wieighted sums of objectives for pareto set generation in multicriteria optimization problems. In *Structural Optimization 14 63-69* (1997), Springer, Ed., Structural Optimization.

[40] Das, I., and Dennis, J. Normal-boundary intersection : A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization 8* (1998), 631–657.

[41] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm : Nsga-ii. *IEEE Transactions on Evolutionay Computation 6*, 2 (2002), 182–197.

[42] Deb, K., Pratap, A., and Meyarivan, T. Constrained test problems for multiobjective evolutionary optimization. In *First International Conference on Evolutionary Multi-Criterion Optimization* (2001), Springer Verlag, pp. 284–298.

[43] Delbos, F. *Problèmes d'optimisation non linéaire avec contraintes en tomographie de réflexion 3D.* PhD thesis, université Paris 6, 2004.

[44] Delbos, F., Gilbert, J.-C., and Sinoquet, D. Non linear optimization for reservoir characterization. In *ENGOPT International conference on engineering optimization, Rio de Janeiro, Brazil, 1-5 june 2008* (2008).

[45] Dennis, J. E., and Das, I. Normal-boundary intersection : A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization 8* (July 1998), 631–657.

[46] Dennis, J. E., and Schnabel, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall (Englewood Cliffs), 1983.

[47] Dennis, J. E., and Torczon, V. Direct search methods on parallel machines. *SIAM J. Optim. 1* (1991), 448–474.

[48] Dhaenens-Flipo, C. *Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à L'Extraction de Connaissances.* PhD thesis, Université des Sciences et Technologies de Lille, Lille, France, 2005.

[49] Ding, D. Development of a data partition technique for gradient-based optimization methods in history matching. In *SPE 130473* (2010).

[50] Ding, D. Improving perturbation designs for gradientbased optimization methods in history matching. In *ECMOR European Conference on the Mathematics of Oil Recovery, 12th, Oxford, UK, 6-9 september 2010* (2010).

[51] Diniz-Ehrhardt, M., Martinez, J., and Pedroso, L. Derivative-free methods for nonlinear programming with general lower-level constraints. Tech. rep., University of Campinas, Brazil, 2010.

[52] do Nascimento Guimarães, L. J., Horowitz, B., and Afonso, S. M. B. Global optimal solutions for reservoir engineering applications. In *EngOpt 2008 - International Conference on Engineering Optimization Rio de Janeiro, Brazil, 01 - 05 June 2008.* (2008).

[53] Dorigo, M. *Optimization, Learning and Natural Algorithms.* PhD thesis, Politecnico di Milano, Italie, 1992.

[54] Dréo, J. *Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical.* PhD thesis, Paris 12, 2004.

[55] Dumas, L. *Optimization and CFD.* Springer, 2008, ch. CFD-based Optimization for Automotive Aerodynamics.

[56] Dumas, L., and Alaoui, L. E. *Medical Applications of Genetic and Evolutionary Computation.* John Wiley and Sons, 2010, ch. How Genetic Algorithms can Improve Pacemaker Efficiency.

[57] Emerick, A., Silva, E., Messer, B., Almeida, L., Szwarcman, D., Pacheco, M., and Vellasco, M. Well placement optimization using a genetic algorithm with nonlinear constraints. In *In SPE reservoir simulation symposium.* (2009).

[58] Emmerich, M., Beume, N., and Naujoks, B. An emo algorithm using the hypervolume measure as selection criterion, third international conference on evolutionary multy-criterion optimization (emo 2005). *In C. A. C. Coello, E. Zitzler, and A. H. Aguirre (Eds.), Springer-Verlag 3410 of LNCS* (2005), 62–76.

[59] Fan, S.-K. S., Liang, Y.-C., and Zhara, E. A genetic algorithm and a particle swarm optimizer hybridized with nelder-mead simplex search. *Pergamon Press 50* (2006), 401–425.

[60] Feraille, M., and Busby, D. Uncertainty management on a reservoir workflow. In *International Petroleum Technology Conference* (2009).

[61] Feraille, M., Roggero, F., Manceau, E., Hu, L. Y., Zabalza-Mezghani, I., and Reis, L. Application of advanced history matching techniques to an integrated field case study. In *SPE Annual technical conference and exhibition* (2003).

[62] Fiacco, A., and McCormick, G. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques.* John Wiley, New York, 1968.

[63] Fleischer, M. The measure of pareto optima. applications to multi-objective metaheuristics. *EMO of LNCS Springer-Verlag 2003* (2632), 519–533.

[64] Fletcher, R. *Practical Methods of Optimization. Volume 1 : Unconstrained Optimization.* John Wiley & Sons, Chichester, 1980.

[65] Fletcher, R. *Practical Methods of Optimization. Volume 2 : Constrained Optimization.* John Wiley & Sons, Chichester, 1981.

[66] Fletcher, R., Gould, N. I. M., Leyffer, S., Toint, P. L., and Wächter, A. Global convergence of trust-region sqp-filter algorithms for general nonlinear programming. In *SIAM J. Optim.* (2002).

[67] Font, G., Sinoquet, D., Langouët, H., Castagne, M., and Magand, S. Derivative free optimization method and physical simulations coupled with statistical models for transient engine calibration. In *6th Conference Design of Experiments (DoE) in Engine Development, Berlin (Allemagne)* (2011).

[68] Fornel, A., Noetinger, B., and Roggero, F. Using time domain seismic attributes for history matching. In *IEA - International workshop and symposium of the International Energy Agency* (2007).

[69] Frimannslund, L., and Steihaug, T. A generating set search method using curvature information. *Comput. Optim. Appl. 38* (2007), 105–121.

[70] Frisch, K. The logarithmic potential method for convex programming. Memorandum, Institute of Economics, University of Oslo, Oslo, Norway, 1955.

[71] Gilbert, J. *Optimisation Différentiable – Théorie et Algorithmes, Syllabus de cours à l'ENSTA*. http ://www-rocq.inria.fr/∼gilbert/ensta/optim.html, 2003.

[72] Gill, P., Murray, W., and Saunders, M. SNOPT : an SQP algorithm for large-scale constrained optimization. Numerical Analysis Report 96-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1996.

[73] Gill, P., Murray, W., Saunders, M., and Wright, M. User's guide for NPSOL (version 4.0) : a Fortran package for nonlinear programming. Technical Report SOL-86-2, Department of Operations Research, Stanford University, Stanford, CA 94305, 1986.

[74] Ginsbourger, D. *Métamodèles Multiples pour l'Approximation et l'Optimisation de Fonctions Numériques Multivariables.* PhD thesis, Ecole des Mines de Saint-Etienne, 2009.

[75] Goldberg, D. E. Genetic algorithms in search, optimization and machine learning. *Addison-Wesley Publishing Company, Reading, Massachusetts* (1989).

[76] Gould, N. On the convergence of a sequential penalty function method for constrained minimization. *SIAM Journal on Numerical Analysis 26* (1989), 107–126.

[77] Gould, N., Orban, D., and Toint, P. L. Cuter and sifdec : A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software 29* (2003), 373–394.

[78] Gratton, S., Toint, P. L., and Tröltzsch, A. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software,* (2010).

[79] Guérillot, D., and Roggero, F. Matching the future for the evaluation of extreme reservoir development scenarios. In *8th European Symposium on Improved Oil Recovery* (1995).

[80] Gutmann, H.-M. A radial basis function method for global optimization. *J. of Global Optimization 19*, 3 (2001), 201–227.

[81] Gutmann, H.-M. *Radial Basis Function Methods for Global Optimization.* PhD thesis, University of Cambridge, 2001.

[82] Haftka, R. T. Combining global and local approximations. *AIAA J. 29* (1991), 1523–1525.

[83] Han, S.-P. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming 11* (1976), 263–282.

[84] Han, S.-P. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications 22* (1977), 297–309.

[85] Hansen, N., Igel, C., and Roth, S. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation 15*, 1 (2007), 1–28.

[86] Hansen, N., and Kern, N. Evaluating the cma evolution strategy on multimodal test functions. In *In Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings, p. 282-291, Berlin : Springer.* (2004).

[87] Hansen, N., and Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies : The covariance matrix adaptation. In *In Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, p. 312-317* (1996).

[88] Hansen, N., and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation 9*, 2 (2001), 159–195.

[89] Holland, J. H. Adaptation in natural and artificial systems. *University of Michigan Press* (1975).

[90] Hooke, R., and Jeeves, T. A. "direct search" solution of numerical and statistical problems. *ACM 8* (1961), 212–229.

[91] Huang, D., TT, A., WI, N., and N., Z. Global optimization of stochastic black-box systems via sequential kriging meta-models. *J. of Global Optimization 34* (2006), 441–466.

[92] Ishikawa, T., and M., M. An optimization method based on radial basis functions. *IEEE Transactions on Magnetics 33*, 2 (1997), 1868–1871.

[93] Ishikawa, T., Y., T., and M., M. A combined method for the global optimization using radial basis function and deterministic approach. *IEEE Transactions on Magnetics 35*, 3 (1999), 1730–1733.

[94] Jakobsson, S., Patriksson, M., Rudholm, J., and Wojciechowski, A. A method for simulation based optimization using radial basis functions. *Optimization and Engineering Online* (2009).

[95] Jakobsson, S., Saif-Ul-Hasnain, M., Rundqvist, R., Edelvik, F., Andersson, B., Patriksson, M., Ljungqvist, M., Lortet, D., and Johan-Wallesten. Combustion engine optimization : a multiobjective approach. *Optimization and Engineering Online* (2009).

[96] Jebalia, M. *Optimisation par Stratégies d'Évolution : Convergence et vitesses de convergence pour des fonctions bruitées - Résolution d'un problème d'identification.* PhD thesis, Université Pierre et Marie Curie, 2008.

[97] Jiménez, F., and Verdegay, J. L. Constrained multiobjective optimization by evolutionary algorithms. *In Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98), University of La Laguna, Tenerife, Spain* (1998), 266–271.

[98] Jones, D. Large-scale multi-disciplinary mass optimization in the auto industry. *Presented at the MOPTA 2008 Conference* (20 August 2008).

[99] Jones, D. R. A taxonomy of global optimization methods based on response surfaces. *J. of Global Optimization 21* (2001), 345–383.

[100] Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization 13* (1998), 455–492.

[101] Jozefowiez, N. *Modélisation et Résolution Approchée de Problèmes de Tournées Multi-Objectif.* PhD thesis, Université des Sciences et Technologies de Lille, France, 2004.

[102] Käck, J.-E. Constrained global optimization with radial basis functions. Tech. rep., Department of Mathematics and Physics Mälardalen University P.O. Box 883 SE-721 23 Västerås, Sweden, 2004.

[103] Kelley, C. T. Iterative methods for optimization. *SIAM* (1999).

[104] Kennedy, J., and Eberhart, R. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks 4* (1995), 1942–1948.

[105] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *Science 220* (1983), 671–680.

[106] Knowles, J. Parego : a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation 10* (2006), 50–66.

[107] Koduru, P., Das, S., and Welch, S. M. A particle swarm optimization-nelder mead hybrid algorithm for balanced exploration and exploitation in multidimensional search space. *Proceedings, International Conference on Artificial Intelligence, Las Vegas, Nevada* (2006), 457–464.

[108] Kolda, T. G., Lewis, R. M., and Torczon, V. Optimization by direct search : New perspectives on some classical and modern methods. *SIAM Review 45, No. 3* (2003), 385–482.

[109] Langouët, H. Optimisation multi-objectifs pour la calibration des moteurs. Master's thesis, Université Pierre et Marie Curie, 2007.

[110] Langouët, H., Delbos, F., Sinoquet, D., and Da Veiga, S. A derivative free optimization method for reservoir characterization inverse problem. In *ECMOR European Conference on the Mathematics of Oil Recovery, 12th, Oxford, UK, 6-9 september 2010* (2010).

[111] Langouët, H., Métivier, L., Sinoquet, D., and Tran, Q. H. Optimization for engine calibration. In *ENGOPT International conference on engineering optimization, Rio de Janeiro, Brazil, 1-5 june 2008* (2008).

[112] Langouët, H., Métivier, L., Sinoquet, D., and Tran, Q.-H. Engine calibration : multi-objective constrained optimization of engine maps. *Optimization and Engineering* (soumis en 2010).

[113] Langouët, H., and Sinoquet, D. Derivative free optimization under constraints. In *ENBIS EMSE, 1-3 July 2009* (2009).

[114] Lasserre, J. B. Global optimization with polynomials and the problem of moments. *Siam J. Optim. 11*, 3 (2000), 796–817.

[115] Lasserre, J. B. *Moments, Positive Polynomials and Their Applications.* Imperial College Press, 2009.

[116] Marazzi, M., and Nocedal, J. Wedge trust region methods for derivative free optimization. *Math. Program. 91* (2002), 289–305.

[117] McKay, M., Beckman, R., and Conover, W. A comparison of three methods for selecting values of input variables in the analysis of outpit from a computer code. *Technometrics 21* (1979), 239–246.

[118] Metla, N., Delbos, F., Da Veiga, S., and Sinoquet, D. Constrained nonlinear optimization for extreme scenarii evaluation in reservoir characterization. In *ECMOR European Conference on the Mathematics of Oil Recovery, 12th, Oxford, UK, 6-9 september 2010* (2010).

[119] Michalewicz, Z., Dasgupta, D., Riche, R., and Schoenauer, M. Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal 30* (1996), 851–870.

[120] Michalewicz, Z., and Nazhiyath, G. Genocop iii : a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. *In Proceedings of the 2nd IEEE International Conference on Evolutionary Computation* (1995), 647–651.

[121] Moré, J. J., and Wild, S. M. Benchmarking derivative-free optimization algorithms. *Siam J. Optim. 20* (2009), 172–191.

[122] Métivier, L. Modélisation et optimisation des cartographies d'un moteur. *Rapport de stage IFP* (2006).

[123] Myers, R. H., and Montgomery, D. C. *Response Surface Methodology : Process and Product Optimization Using Designed Experiments.* Wiley series in probability and statistics, 1995.

[124] Nelder, J. A., and Mead, R. A simplex method for function minimization. *Comput. J. 7* (1965), 308–313.

[125] Nocedal, J., and Wright, S. *Numerical Optimization.* Springer, 1999.

[126] Oeuvray, R. *Trust-Region Methods Based On Radial Basis Functions With Application To Biomedical Imaging.* PhD thesis, Institute of Mathematics, Swiss Federal Insitute of Technology, Lausanne, 2005.

[127] Palomares, U. G., and Mangasarian, O. Superlinear convergent quasi-Newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming 11* (1976), 1–13.

[128] Pareto, V. Cours d'économie politique. *vol. 1 et 2, F. Rouge, Lausanne* (1896).

[129] Powell, M. J. D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J. 7* (1964), 155–162.

[130] Powell, M. J. D. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming 14* (1978), 224–248.

[131] Powell, M. J. D. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In *Nonlinear Programming 3* (1978), O. Mangasarian, R. Meyer, and S. Robinson, Eds., Academic Press, New York, pp. 27–63.

[132] Powell, M. J. D. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical Analysis Dundee 1977* (1978), G. Watson, Ed., no. 630 in Lecture Notes in Mathematics, Springer-Verlag, Berlin, pp. 144–157.

[133] Powell, M. J. D. *Advances In Optimization And Numerical Analysis*, gomez ed. Kluwer Academic Publishers, 1994, ch. A direct search optimization method that models the objective and constraint functions by linear interpolation, pp. 51–67.

[134] Powell, M. J. D. A direct search optimization method that models the objective by quadratic interpolation. In *Presentation at the 5th Stockholm Optimization Days, Stockholm, BC, Canada* (1994).

[135] Powell, M. J. D. Direct search algorithms for optimization calculations. *Acta Numerica 7* (1998), 287–336.

[136] Powell, M. J. D. A quadratic model trust region method for unconstrained minimization without derivatives. *Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong* (1998).

[137] Powell, M. J. D. Uobyqa : unconstrained optimization by quadratic approximation. *Mathematical Programming 92* (2002), 555–582.

[138] Powell, M. J. D. Least frobenius norm updating of quadratic models that satisfy interpolation conditions. *Mathematical Programming 100* (2004), 183–215.

[139] Powell, M. J. D. The newuoa software for unconstrained optimization without derivatives. Tech. rep., Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Cambridge, 2006.

[140] Powell, M. J. D. Developments of newuoa for unconstrained minimization without derivatives. Tech. rep., University of Cambridge, 2007.

[141] Powell, M. J. D. A view of algorithms for optimization without derivatives. Tech. rep., University of Cambridge, 2007.

[142] Powell, M. J. D. The bobyqa algorithm for bound constrained optimization without derivatives. Tech. rep., Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Cambridge, 2009.

[143] Powell, M. J. D. On the convergence of trust region algorithms for unconstrained minimization without derivatives. Tech. rep., Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, Cambridge, 2011.

[144] Rechenberg, I. Evolutionsstrategie : Optimierung technisher systeme nache prinzipien der biologischen evolution. *Werkstatt Bionik und Evolutionstechnik. Stuttgart : Frommann-Holzboog* (1973).

[145] Regis, R. G. Radial basis function algorithms for large-scale nonlinearly constrained black-box optimization. In *20th International Symposium on Mathematical Programming Chicago, IL* (August 2009), vol. 37, pp. 113–135.

[146] Regis, R. G., and Shoemaker, C. A. Constrained global optimization of expensive black box functions using radial basis functions. *J. of Global Optimization 31* (2005), 153–171.

[147] Regis, R. G., and Shoemaker, C. A. Improved strategies for radial basis function methods for global optimization. *J. of Global Optimization 37* (2007), 113–135.

[148] Regis, R. G., and Shoemaker, C. A. Parallel radial basis function methods for the global optimization of expensive functions. *European J. Oper. Res. 182* (2007), 514–535.

[149] Regis, R. G., and Shoemaker, C. A. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput. 19* (2007), 497–509.

[150] Rosenbrock, H. H. An automatic method for finding the greatest or least value of a function. *Comput. J. 3* (1960), 175–184.

[151] Roudenko, O. *Application des Algorithmes Evolutionnaires aux problèmes d'optimisation multi-critère avec contraintes.* PhD thesis, Université Paris 6, Paris, France, 2004.

[152] Rudholm, J., and Wojciechowsjki, A. A method for simulation based optimization using radial basis functions. Master's thesis, university of Göteborg, 2007.

[153] Schittkowski, K. NLPQL : a FORTRAN subroutine solving constrained nonlinear programming problems. *Annals of Operations Research 5* (1985), 485–500.

[154] Schmied, A. *Méthodes stochastiques d'optimisation appliquées à la mise au point des moteurs.* PhD thesis, Thèse de doctorat, Université Paris 12 — Val de Marne, UFR de Sciences et Technologies, 2003.

[155] Schonlau, M. *Computer Experiments and Global Optimizations.* PhD thesis, University of Waterloo, 1997.

[156] Schonlau, M., and Welch, W. Global optimization with nonparametric function fitting. *Proceedings of the Section on Physical and Engineering Sciences, American Statistical Association* (1996), 183–186.

[157] Schwefel, H.-P. Evolution and optimum seeking. *Sixth-Generation Computer Technology Series, John Wiley and Sons* (1995).

[158] Serafini, D. B. *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1998.

[159] Sinoquet, D., and Delbos, F. Adapted nonlinear optimization method for production data and 4d seismic inversion. In *ECMOR European Conference on the Mathematics of Oil Recovery, 11th, Bergen, Norway, 8-11 september 2008* (2008).

[160] Søndergaard, J. *Optimization Using Surrogate Models — by the space mapping technique.* PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2003.

[161] Spendley, W., Hext, G., and Himsworth, F. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics 4* (1962), 441–461.

[162] Torczon, V. On the convergence of the multidirectional search algorithm. *SIAM J. Optim. 1* (1991), 123–145.

[163] Torczon, V. On the convergence of pattern search algorithms. *SIAM J. Optim. 7* (1997), 1–25.

[164] Toropov, V. V. Simulation approach to structural optimization. *Struct. Multidiscip. Optim. 1* (1989), 37–46.

[165] Touzani, S. *Response surface methods based on analysis of variance expansion for sensitivity analysis.* PhD thesis, Université de Grenoble, 2011.

[166] Ugur, O., Karasozen, B., Schäfer, M., and Yapici, K. Derivative free optimization methods for optimizing stirrer configurations. *Derivative free optimization methods for optimizing stirrer configurations, European Journal of Operational Research, Elsevier, December 191(3)* (2008), 855–863.

[167] Vanden Berghen, F. *CONDOR : a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions.* PhD thesis, Université Libre de Bruxelles, Faculté des Sciences Appliquées, 2004.

[168] Vanden Berghen, F. Constrained, non-linear, direct, parallel optimization algorithm using a trust region method for high-computing load, continuous, noisy functions : Condor. Tech. rep., Université Libre de Bruxelles, 2007.

[169] Vanden Berghen, F., and Bersini, H. Condor, a new parallel, constrained extension of powell's uobyqa algorithm : Experimental results and comparison with the dfo algorithm. *Journal of computational and applied mathematics 181* (2005), 157–175.

[170] VanGrieken, M. *Optimisation pour l'apprentisage et apprentisage pour l'optimisation.* PhD thesis, Université Paul Sabatier, Toulouse, 2004.

[171] Vaz, A., and L.N.Vicente. A particle swarm pattern search method for bound constrained global optimization. *Journal of Global Optimization 39* (2007), 197–219.

[172] Villemonteix, J. *Optimisation De Fonctions Coûteuses : Modèles Gaussiens Pour Une Utilisation Efficace Du Budget D'évaluations : Théorie Et Pratique Industrielle.* PhD thesis, Université Paris-Sud 11, Faculté Des Sciences D'Orsay, 2008.

[173] Villemonteix, J., Vazquez, E., Sidorkiewicz, M., and Walter, E. Informational approach to the global optimization (iago) of expensive to evaluate functions. In *GdR MASCOT-NUM CEA Cadarache* (2008).

[174] Villemonteix, J., Vazquez, E., and Walter, E. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization 44, 4* (2009), 509–534.

[175] Weise, T. *Global Optimization Algorithms - Theory and Application.* Thomas Weise, 2008.

[176] Wild, S. M. Mnh : A derivative-free optimization algorithm using minimal norm hessians. Tech. rep., School of Operations Research & Information Engineering, Cornell University, 2008.

[177] Wild, S. M. *Derivative-Free Optimization Algorithms For Computationally Expensive Functions.* PhD thesis, Faculty of the Graduate School of Cornell University, 2009.

[178] Wild, S. M., Regis, R. G., and Shoemaker, C. A. Orbit : Optimization by radial basis function interpolation in trust-regions. *Siam J. Sci. Comput. 30* (2008), 3197–3219.

[179] Wild, S. M., and Shoemaker, C. Global convergence of radial basis function trust region derivative-free algorithms. Tech. rep., Mathematics and Computer Science Division, 2011.

[180] Wilson, R. *A simplicial algorithm for concave programming.* PhD thesis, Graduate School of Business Administration, Harvard University, Cambridge, MA, USA, 1963.

[181] Winfield, D. *Function and functional optimization by interpolation in data tables.* PhD thesis, Harvard University, Cambridge, MA, USA, 1969.

[182] Winfield, D. Function minimization by interpolation in a data table. *Journal of the Institute of Mathematics and Its Applications 12* (1973), 339–347.

[183] Winslow, T. A., Trew, R. J., Gilmore, P., and Kelley, C. T. Doping profiles for optimum class b performance of gaas mesfet amplifiers. *Proceedings of the IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits* (1991), 188–197.

[184] Zahara, E., and Kao, Y.-T. Hybrid nelder-mead simplex search and particle swarm optimization for constrained engineering design problems. *Pergamon Press 36* (2009), 3880–3886.

[185] Zhang, H., Conn, A. R., and Scheinberg, K. A derivative-free algorithm for the least-square minimization. Tech. rep., Department of Mathematics, Louisiana State University, 2009.

[186] Zitzler, E., and Thiele, L. Multiobjective optimization using evolutionary algorithms – a comparative case study. *In A. E. Eiben, T. Bäck, M. Schoenauer, et H.-P. Schwefel (Eds.), Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Springer-Verlag* (1998), 292–301.