



Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique

Université Mohamed El-Bachir El-Ibrahimi
Bordj Bou Arréridj

Faculté des Mathématiques et de l'Informatique



Thèse de doctorat

Présentée par

Ramdani Zoubir

En vue de l'obtention du grade de **Docteur** (3^{ème} cycle, LMD)

Domaine : Mathématiques et informatique

Filière : Mathématiques

Spécialité : Mathématiques appliquées

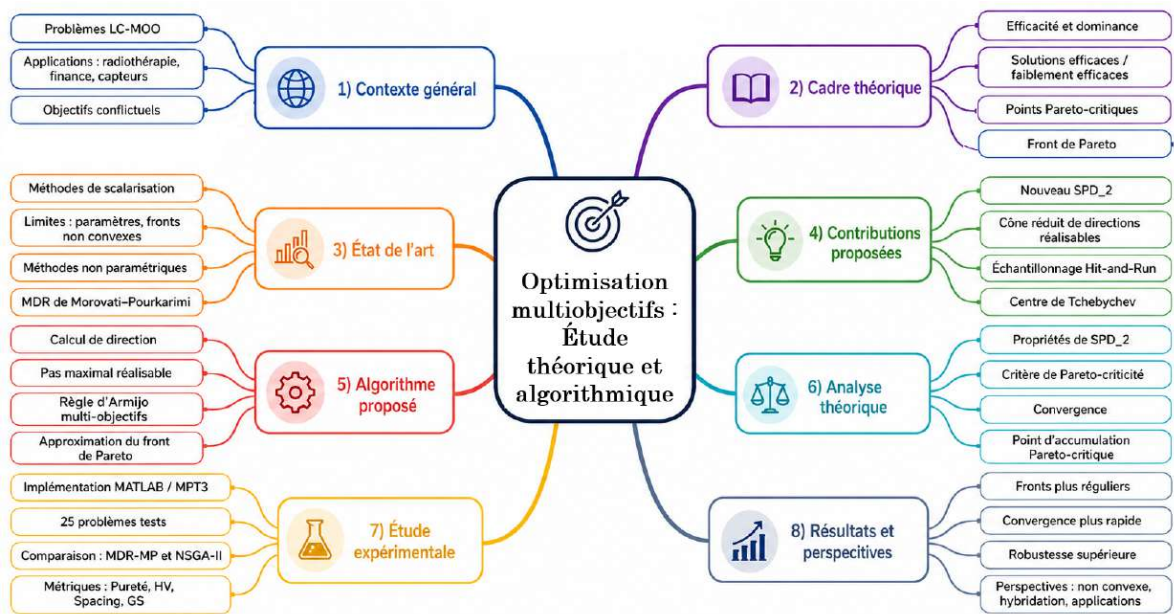
**Optimisation multi-objectifs :
Étude théorique et algorithmique**

Soutenue publiquement le : 12 / 05 / 2026

Devant le jury composé de :

Dr. Zeghdane Rebiha	Maître de Conférences A	Université de BBA	Présidente
Dr. Addoune Smail	Maître de Conférences A	Université de BBA	Directeur
Dr. Brahmi Boualem	Maître de Conférences A	Université de BBA	Co-directeur
Pr. Zouache Djaafar	Professeur	Université de BBA	Examineur
Dr. Brahmi Belkacem	Maître de Conférences A	Université de Béjaïa	Examineur
Dr. Khimoum Noureddine	Maître de Conférences A	Université de Béjaïa	Examineur

Année universitaire
2025/2026



Résumé

Cette thèse traite de l'optimisation multi-objectif sous contraintes linéaires (LC-MOO), dont l'objectif est d'approximer efficacement le front de Pareto. Elle propose une amélioration d'une méthode de directions réalisables reposant sur deux contributions complémentaires : d'une part, la reformulation du sous-problème de recherche de direction au moyen d'un cône de directions réalisables réduit; d'autre part, l'introduction d'une procédure d'échantillonnage Hit-and-Run permettant de générer une population initiale mieux répartie dans le polyèdre réalisable. La convergence théorique de l'approche est établie, et une validation numérique réalisée sous MATLAB avec la boîte à outils MPT3, sur 25 problèmes LC-MOO, compare la MDR proposée à une autre MDR ainsi qu'à NSGA-II. Les résultats mettent en évidence une amélioration significative tant en qualité d'approximation du front de Pareto qu'en rapidité de convergence.

MOTS CLÉS : Optimisation multi-objectif; Contraintes linéaires; Front de Pareto; Directions réalisables; Hit-and-Run.

Abstract

This thesis addresses linearly constrained multi-objective optimization (LC-MOO) to efficiently approximate the Pareto front. It proposes an improved Method of Feasible Directions based on two complementary contributions : first, a reformulation of the Direction-Finding Subproblem by introducing a reduced cone of feasible directions; second, the incorporation of a Hit-and-Run sampling procedure to generate a better-distributed initial population. The theoretical convergence of the approach is established, and a numerical experiment conducted in MATLAB using the MPT3 toolbox on 25 LC-MOO problems compares the proposed MFD with another MFD and with NSGA-II. The results show a significant improvement in both the quality of the Pareto front and convergence speed.

KEYWORDS : Multi-objective optimization; Linear constraints; Pareto front; Feasible directions; Hit-and-Run.

الملخص

تتناول هذه الأطروحة مسائل التحسين متعدد الأهداف تحت قيود خطية (LC-MOO) بهدف تقريب جبهة Pareto بكفاءة ضمن إطار لا معلمي قائم على طريقة الاتجاهات المقبولة. وتسعى الطريقة إلى بناء تقريب دقيق ومتوازن لجبهة Pareto الإمكان. تركز المقاربة على مساهمتين متكاملتين: أولاً، إعادة صياغة مسألة البحث عن الاتجاه عبر إدخال مخروط مخفض للاتجاهات المقبولة يدمج جميع قيود متعدد السطوح، بما يعزز استقرار المسار ويحد من الاهتزازات. ثانياً، اعتماد إجراء أخذ العينات Hit-and-Run لتوليد مجموعة أولية أكثر تجانساً داخل المجال الممكن، مما يحسن تغطية فضاء البحث. وقد تم إثبات تقارب الطريقة نظرياً. كما أُجريت دراسة عددية باستخدام MATLAB مع الحزمة MPT3 على 52 مسألة من نوع LC-MOO، حيث تمت مقارنة الطريقة المقترحة مع نسخة أخرى من MFD ومع خوارزمية NSGA-II. وأظهرت النتائج تحسناً ملحوظاً في جودة جبهة صرّة وسرعة التقارب مقارنة بالطرائق المرجعية.

الكلمات المفتاحية: التحسين متعدد الأهداف؛ القيود الخطية؛ الحلول الكفوءة؛ جبهة Pareto؛ طريقة الاتجاهات المقبولة؛ استخراج؛ أخذ العينات Hit-and-Run.

REMERCIEMENTS ET DÉDICACES

Je rends tout d'abord grâce à Dieu, le Tout-Puissant, pour la force, la patience et la persévérance qu'Il m'a accordées tout au long de ce parcours, et sans lesquelles l'aboutissement de ce travail n'aurait pas été possible.

Je tiens à exprimer ma profonde gratitude à mes directeurs de thèse, Monsieur Adoune Smail et Monsieur Brahmi Boualem, pour leur encadrement, leurs conseils, leur disponibilité et la qualité de leurs orientations scientifiques. Leurs remarques et leurs encouragements ont été déterminants pour mener à bien cette recherche.

J'adresse également mes sincères remerciements aux membres du jury pour l'honneur qu'ils me font en acceptant d'examiner cette thèse, ainsi que pour le temps consacré à son évaluation et pour leurs observations constructives.

Je remercie chaleureusement ma famille, en particulier mes parents, pour leur soutien constant, leurs sacrifices et leur confiance. Je n'oublie pas mes frères et sœurs, dont la présence et les encouragements ont toujours été une source de motivation. J'exprime enfin une reconnaissance particulière à mon épouse pour sa patience, son soutien indéfectible et sa compréhension durant toute la période de préparation de cette thèse.

Je dédie ce travail à tous les membres de ma famille.

Table des matières

1	Notations et notions préliminaires	8
1.1	Notions préliminaires	9
1.1.1	Normes et distances dans \mathbb{R}^n	9
1.1.2	Boules, voisinages et notions locales	10
1.2	Quelques notions sur les polyèdres	11
1.2.1	Intérieur, frontière et intérieur relatif	11
1.2.2	Points extrêmes	12
1.2.3	Faces d'un polyèdre	12
1.2.4	Boîte englobante	12
1.2.5	Volume du polyèdre	13
1.3	Fonctions différentiables	14
1.4	Fonction convexe	15
1.5	Minimum local, minimum global et point stationnaire	17
1.6	Conditions d'optimalité	19
1.7	Direction réalisable et direction de descente	21
2	Méthodes de directions réalisables pour l'optimisation mono-objectif	25
2.0.1	Schéma général d'une MDR	26
2.1	SPD de Zoutendijk	26
2.2	Contraintes quasi-actives, jamming et limites de la variante originale	30
2.2.1	Contraintes quasi-actives	30
2.2.2	Jamming et contre-exemple de Wolfe	31
2.3	SPD de Topkis–Veinott	33
2.4	SPD de Pironneau–Polak	34
2.5	SPD de Wiest–Polak	35
2.6	SPD de Cawood–Kostreva (normes relaxées)	35
2.7	SPD modifié de Chen–Kostreva	36
2.8	Méthode de Frank–Wolfe	36

2.9	Méthode de projection du gradient (Rosen)	37
2.10	Méthode du gradient réduit	40
3	Optimisation multi-objectifs	43
3.1	Définitions et notions de base	44
3.2	Caractérisation des solutions efficaces	47
3.3	Classification des méthodes de résolution	49
3.4	Méthodes scalaires	50
3.4.1	Méthode de la somme pondérée des fonctions objectif	50
3.4.2	Méthode des ϵ -contraintes	52
3.4.3	Méthode des normes pondérées	53
3.4.4	Autres méthodes scalaires	56
3.5	Algorithme NSGA-II	56
4	Méthode de directions réalisables pour l'optimisation multi-objectifs	58
4.1	Pourquoi les méthodes non paramétriques	59
4.2	Notions de directions réalisables pour le contexte multi-objectifs	64
4.3	Méthode de directions réalisables de Morovati et Pourkarimi	69
4.4	Méthode de directions réalisables de Ramdani et al. [80]	74
4.5	Méthodes d'échantillonnage pour générer des populations initiales de points réalisables	86
4.5.1	Méthode d'échantillonnage de Morovati et Pourkarimi [71]	86
4.5.2	Procédure d'échantillonnage Hit-and-Run de Ramdani et al. [80]	88
4.5.3	Exemple d'illustration	93
4.6	Conclusion	97
5	Étude expérimentale et comparative	98
5.1	Description de la liste des problèmes testés	99
5.2	Description des codes en MATLAB	101
5.3	Intégration de la toolbox MPT3	103
5.4	Les métriques d'évaluation de Front Pareto	104
5.5	Résultats et discussions	107
	Bibliographie	117

Table des figures

1.1	Illustration d'une fonction convexe	16
1.2	Interprétation géométrique de la direction de descente réalisable	23
2.1	Effet des contraintes quasi-actives : impact sur la génération de directions.	31
2.2	Interprétation géométrique de la méthode de projection du gradient de Rosen	39
3.1	Exemples de formes possibles d'un <i>front de Pareto</i>	45
3.2	Front de Pareto et ses bornes	47
3.3	Cas d'ensemble \mathcal{Y} convexe	51
3.4	Cas d'ensemble \mathcal{Y} non convexe	51
3.5	Fonctionnement de la méthode ε -contrainte	53
4.1	Espace de décision et ensembles faiblement efficace et efficace pour l'Exemple (4.1).	61
4.2	Ensembles faiblement efficace et efficace, et front de Pareto pour l'Exemple (4.2).	63
4.3	Interprétation géométrique de la direction de descente réalisable de l'exemple (4.3).	68
4.4	Interprétation géométrique de la méthode d'échantillonnage de Morovati et Pourkarimi [71].	94
4.5	Interprétation géométrique de la méthode Hit-and-Run.	96
5.1	Métriques d'évaluation des fronts de Pareto	107
5.2	Population de points réalisables générée par la procédure d'échantillon- nage Hit-and-Run pour le problème test <i>HANNE3</i>	107
5.3	Population de points réalisables générée par la procédure d'échantillon- nage de Morovati et Pourkarimi pour le problème test <i>HANNE3</i>	107
5.4	Ensemble des solutions efficaces en espace de décision pour le problème test <i>HANNE3</i>	110
5.5	Ensemble des solutions efficaces en espace de décision pour le problème test <i>KATHRIN2008</i>	110
5.6	Front de Pareto du problème test <i>HANNE3</i>	110

5.7	Front de Pareto du problème test <i>EX004</i>	110
5.8	Fronts de Pareto du problème test <i>EXAMPLE 7.0</i>	111
5.9	Fronts de Pareto du problème test <i>PEKKA2</i>	112
5.10	Comparaison des scores de pureté et de l'hypervolume.	112
5.11	Comparaison des scores d'espacement et des scores d'étendue généralisée.	112

Liste des tableaux

5.1	Résumé des problèmes tests et de leurs caractéristiques	100
5.2	Ce tableau présente le nombre d'itérations nécessaires pour atteindre une solution Pareto-optimale, en indiquant les valeurs minimale, maximale et moyenne. Il fournit également le temps CPU moyen (en secondes) pour une population initiale de 200 points réalisables, ainsi que la taille du front de Pareto $ P^* $, après suppression des solutions dominées et redondantes. Les résultats sont reportés pour trois méthodes : notre <i>MDR</i> , la <i>MDR</i> de <i>MDR-MP</i> et <i>NSGA-II</i>	109

Introduction générale

L'optimisation multi-objectifs est une discipline centrale de la recherche opérationnelle et de l'aide à la décision. Elle traite des problèmes où plusieurs objectifs doivent être optimisés simultanément, souvent de manière conflictuelle, l'amélioration de l'un pouvant entraîner la dégradation d'un autre. Contrairement à l'optimisation mono-objectif, il n'existe généralement pas de solution optimale unique ; on recherche plutôt des solutions dites *solutions efficaces* ou *solutions optimales au sens de Pareto*, c'est-à-dire des solutions réalisables pour lesquelles aucun objectif ne peut être amélioré sans en détériorer au moins un autre. Ces solutions représentent des compromis entre les objectifs et constituent un support d'analyse pour le décideur, qui choisit parmi elles celle qui correspond le mieux à ses préférences et aux exigences du contexte considéré.

Dans cette thèse, nous considérons plus particulièrement une classe importante de problèmes, à savoir les problèmes d'optimisation multi-objectifs sous contraintes linéaires, notés *LC-MOO* (pour *Linearly Constrained Multi-Objective Optimization*), dont l'ensemble réalisable est défini par un système d'inégalités linéaires décrivant une région polyédrale. De tels problèmes apparaissent dans de nombreux domaines d'application, notamment la planification de radiothérapie [36], la conception de réseaux de capteurs sans fil [21], l'optimisation des systèmes de distribution d'eau [66], la gestion de trésorerie [83], l'optimisation de portefeuilles d'investissement [73, 45, 79], la planification agricole [95], ainsi que la logistique humanitaire en situation de crise [94].

À titre d'illustration en finance, *Oliva et al.* [73] étudient un problème d'optimisation multi-objectifs de portefeuille, formulé comme un *LC-MOO*, où les variables de décision représentent les poids d'allocation du capital entre les actifs. Les auteurs considèrent conjointement des objectifs de performance et de maîtrise du risque, c'est-à-dire maximiser le rendement et minimiser une mesure de risque, sous des contraintes réalistes de construction de portefeuille : budget, non-négativité, limites réglementaires, borne sur l'écart de suivi par rapport à un indice, exigences de diversification et contraintes de car-

dinalité. Ces objectifs étant naturellement conflictuels, le décideur ne peut pas viser une satisfaction simultanée parfaite et doit raisonner en termes de solutions de compromis.

Le concept de solution efficace, également appelée solution optimale au sens de Pareto, formalise la notion de compromis en optimisation multi-objectifs. Une solution réalisable est dite efficace s'il n'existe aucune autre solution réalisable permettant d'améliorer au moins un objectif sans en détériorer au moins un autre. L'ensemble de ces solutions constitue le front de Pareto, qui regroupe tous les compromis admissibles entre les différents objectifs. Ce front représente une base rationnelle pour l'aide à la décision. L'enjeu algorithmique consiste alors à en construire une approximation fiable, bien répartie et exploitable, afin de fournir au décideur une représentation claire des alternatives possibles et de lui permettre de sélectionner une solution conforme à ses préférences et aux exigences du contexte étudié.

Une première famille d'approches, couramment utilisée pour approximer le front de Pareto des problèmes *LC-MOO*, repose sur les *techniques de scalarisation*, également appelées méthodes paramétriques. Leur principe consiste à reformuler le problème multi-objectifs en une suite de problèmes mono-objectifs à l'aide d'un ou de plusieurs paramètres. Cette transformation permet d'utiliser des algorithmes et des solveurs classiques, particulièrement performants lorsque l'ensemble réalisable est polyédral, tout en intégrant explicitement les préférences du décideur à travers des poids, des niveaux de contrainte ou un point de référence. Parmi les techniques les plus répandues figurent la méthode de pondération des objectifs, la méthode des ε -contraintes, ainsi que les approches fondées sur des normes ou des points de référence. Leur simplicité de mise en œuvre, leur efficacité numérique et leur compatibilité avec les outils standards expliquent leur rôle central dans la littérature [70, 35].

Cependant, malgré leur popularité et leur simplicité conceptuelle, les méthodes de scalarisation présentent des limites bien identifiées. Selon *Morovati et Pourkarimi* [71], deux interrogations émergent : pourquoi privilégier des approches non paramétriques face aux schémas classiques de scalarisation, et pourquoi y recourir même dans le cas convexe, où la méthode de pondération et celle des ε -contraintes sont censées, en théorie, générer l'ensemble des solutions efficaces ? Plusieurs travaux apportent des éléments de réponse. *Das et Dennis* [29] montrent qu'une discrétisation uniforme des poids ne conduit pas nécessairement à une distribution uniforme des solutions sur le *front de Pareto*, y compris pour des problèmes convexes. L'exemple analysé dans [42] met en évidence que la méthode de pondération peut n'être réellement efficace que sur un intervalle de pa-

ramètres très restreint, tandis que *Morovati et Pourkarimi* [71] soulignent qu'elle peut échouer à produire une approximation satisfaisante du front, même dans le cas linéaire des fonctions objectif. Pour la méthode des ε -contraintes, *Mavrotas* [68] met en lumière les difficultés liées au choix du paramètre ε , qui peut rendre certains sous-problèmes irréalisables. De manière plus générale, *Ehrgott* [35] insiste sur la sensibilité de ces approches aux paramètres et sur leur capacité limitée à explorer de manière homogène des fronts non convexes. *Marler et Arora* [67] relèvent également leur forte dépendance paramétrique ainsi que le risque d'obtenir des optima non stricts, tandis que *Messac et al.* [69] soulignent qu'un mauvais calibrage peut engendrer un coût de calcul élevé sans amélioration significative de la qualité de l'approximation obtenue.

Face à ces limites, une littérature non paramétrique s'est développée autour des méthodes à directions, dont l'objectif est de réduire la dépendance aux paramètres tout en exploitant directement les informations différentielles des fonctions objectif. Le travail fondateur de *Fliege et Svaiter* [40] constitue une étape décisive en introduisant la notion de direction de descente adaptée au cadre multi-objectifs. Depuis, ce cadre a été enrichi par de nombreuses contributions, parmi lesquelles les méthodes de gradient projeté [50], les approches de régions de confiance [17, 14], les variantes quasi-Newton [42, 78, 2, 72], ainsi que les méthodes de gradient réduit dédiées aux problèmes sous contraintes linéaires [37, 38]. Des développements plus récents incluent les stratégies imposant une distance minimale entre solutions [84], la méthode des directions réalisables de type *Zoutendijk* [71], les schémas à ensembles actifs [47], les méthodes de gradient conditionnel [5, 23] et les approches de gradient conjugué non linéaire sur variétés riemanniennes [22]. L'idée commune à ces méthodes consiste à calculer, à chaque itération, une direction assurant une amélioration simultanée des objectifs tout en maintenant la faisabilité, ce qui les rend particulièrement adaptées aux problèmes *LC-MOO*.

Cette thèse s'inscrit dans le courant des méthodes non paramétriques fondées sur la recherche de directions, et plus précisément dans la famille des méthodes à directions réalisables. Initialement développées pour résoudre des problèmes d'optimisation non linéaire mono-objectif, ces méthodes ont été introduites par *Zoutendijk* [103]. Elles consistent à construire une suite de points réalisables en déterminant, à chaque itération, une direction réalisable assurant une amélioration de la fonction objectif, puis en choisissant une longueur de pas compatible avec les contraintes. Leur intérêt réside dans leur simplicité de mise en œuvre et dans le maintien de la faisabilité tout au long du processus itératif. De nombreuses variantes et extensions ont ensuite été proposées et étudiées dans le cadre mono-objectif [89, 77, 93, 91, 18].

Dans le contexte multi-objectifs, Morovati et Pourkarimi [71] proposent une extension d'une méthode à directions réalisables, que nous noterons désormais *MDR*, au cas des problèmes multi-objectifs sous contraintes générales. Toutefois, lorsque cette extension est appliquée aux problèmes *LC-MOO*, plusieurs limites peuvent être observées. Premièrement, comme cela est discuté dans [10] dans le cadre mono-objectif, le sous-problème de recherche de direction (*SPD*) repose principalement sur les contraintes actives, ce qui peut entraîner des trajectoires oscillatoires et ralentir la convergence. Deuxièmement, le contrôle directionnel simultané des objectifs et de certaines contraintes peut, selon les situations, privilégier le maintien de la faisabilité au détriment d'une amélioration significative des objectifs. Enfin, la procédure d'échantillonnage utilisée dans [71] pour générer une population initiale de points réalisables, destinée à l'approximation du front de Pareto, repose sur un point extrême et sur des directions aléatoires avec des pas fixes ; cette stratégie tend à concentrer les points autour de la zone initiale, créant ainsi un effet de regroupement qui réduit la diversité de la population et limite l'exploration globale de la région réalisable.

L'objectif principal de cette thèse est d'améliorer l'approche de Morovati et Pourkarimi [71] pour la classe des problèmes *LC-MOO*, tout en conservant l'esprit d'une méthode non paramétrique fondée sur des directions réalisables. À cette fin, deux contributions complémentaires sont proposées.

La première concerne le calcul de la direction : nous reformulons le *SPD* de manière à prendre en compte l'ensemble des contraintes du polyèdre, et non seulement les contraintes actives, au moyen d'un cône de directions réalisables réduit. Cette reformulation vise à stabiliser le comportement de l'algorithme au voisinage des frontières, à limiter d'éventuelles oscillations et à produire des déplacements plus efficaces du point de vue de l'amélioration simultanée des objectifs, tout en garantissant le maintien de la faisabilité.

La seconde contribution concerne l'étape d'initialisation, qui joue un rôle déterminant dans la qualité de l'approximation du front de Pareto. Contrairement à la procédure proposée par Morovati et Pourkarimi [71], qui génère les points initiaux à partir d'un point extrême et peut conduire à une concentration des solutions autour de ce point, nous proposons une procédure d'échantillonnage de type *Hit-and-Run* [85, 96]. Cette technique débute au centre de Tchebychev du polyèdre, ce qui permet de générer une population de points réalisables mieux répartie dans l'ensemble réalisable. En évitant l'effet de re-

groupement autour d'un point extrême, cette stratégie favorise une exploration plus homogène du polyèdre et contribue ainsi à une meilleure approximation du front de Pareto.

Ces deux améliorations sont intégrées dans un schéma algorithmique structuré en deux niveaux. Dans un premier temps, une population initiale de points réalisables est générée. Dans un second temps, chaque point est amélioré au moyen d'une procédure itérative jusqu'à satisfaire des conditions d'optimalité au sens multi-objectifs, appelées conditions de *Pareto-criticité*. Chaque itération comporte deux étapes : la première consiste à déterminer une direction de descente réalisable par la résolution d'un sous-problème de recherche de direction (*SPD*) ; la seconde consiste à calculer une longueur de pas le long de cette direction, de manière à garantir simultanément l'amélioration des fonctions objectif et le maintien de la faisabilité. Cette progression est pilotée par une recherche linéaire multi-objectifs de type Armijo, assurant une amélioration conjointe au sens de la dominance ainsi qu'une évolution contrôlée à l'intérieur de la région réalisable. Sous des hypothèses de convexité, une analyse théorique établit des garanties de convergence vers des solutions optimales au sens de Pareto.

Une étude expérimentale et comparative est ensuite menée sous MATLAB, en s'appuyant sur la boîte à outils *MPT3* [52] pour la manipulation des polyèdres et la gestion de la faisabilité. Les performances de la méthode proposée sont comparées à celles de la méthode *MDR* de *Morovati et Pourkarimi* [71], ainsi qu'à l'algorithme évolutionnaire de référence *NSGA-II* [31]. L'évaluation porte à la fois sur la rapidité de convergence, mesurée par le nombre d'itérations et le temps CPU, et sur la qualité des fronts obtenus, mesurée à l'aide de quatre indicateurs : la *pureté*, l'*hypervolume*, l'*espacement* (*spacing*) et l'*étendue généralisée* (*generalized spread*).

La thèse est organisée en cinq chapitres complémentaires, qui suivent une progression logique allant des fondements théoriques aux développements méthodologiques et à la validation expérimentale.

- **Chapitre 1** : présentation des notions de base et des préliminaires mathématiques nécessaires à la compréhension du travail, notamment la convexité, la géométrie des polyèdres, les conditions d'optimalité en optimisation non linéaire, ainsi que les concepts de direction réalisable et de direction de descente. Ce chapitre établit le cadre théorique général dans lequel s'inscrit la suite de la thèse.
- **Chapitre 2** : étude détaillée des méthodes à directions réalisables en optimisation non linéaire mono-objectif, avec un accent particulier sur la méthode de *Zoutendijk* et ses principales variantes. L'objectif est d'analyser les mécanismes fondamentaux

de ces approches afin d'identifier les éléments transférables et adaptables au cadre multi-objectifs sous contraintes linéaires.

- **Chapitre 3** : exposé des fondements de l'optimisation multi-objectifs, incluant les notions de solution efficace, de front de Pareto et les principales méthodes de scalarisation. Ce chapitre présente également une synthèse de l'algorithme évolutionnaire *NSGA-II*, utilisé par la suite comme méthode de référence pour la comparaison expérimentale.
- **Chapitre 4** : présentation de la contribution principale de la thèse. Nous y proposons une reformulation du sous-problème de recherche de direction à l'aide d'un cône de directions réalisables réduit, ainsi qu'une nouvelle procédure d'échantillonnage de type *Hit-and-Run* pour la génération d'une population initiale bien répartie. Le chapitre inclut l'analyse théorique de la méthode proposée, ses propriétés de convergence sous hypothèses appropriées, ainsi qu'une comparaison conceptuelle et algorithmique avec la méthode *MDR* de [71].
- **Chapitre 5** : étude expérimentale et comparative de la méthode proposée. L'implémentation est réalisée sous *MATLAB* en s'appuyant sur la boîte à outils *MPT3* pour la manipulation des polyèdres et la gestion de la faisabilité. Différents problèmes tests sont considérés, et les performances sont évaluées à l'aide de plusieurs indicateurs de qualité du front, accompagnés d'une analyse détaillée des résultats numériques.

La thèse se termine par une conclusion générale qui récapitule la contribution, les démarches suivies et quelques perspectives de recherche.

1

Notations et notions préliminaires

Introduction

Dans ce chapitre, nous considérons le problème d'optimisation *mono-objectif* sous contraintes linéaires, formulé comme suit :

$$(P) \quad \begin{cases} \min_x f(x), \\ \text{s.c. } Ax \leq b, \\ x \in \mathbb{R}^n, \end{cases} \quad (1.1)$$

où s.c. est l'abréviation de *sous contraintes*. Le vecteur $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ représente les variables de décision, $A \in \mathbb{R}^{m \times n}$ et $b \in \mathbb{R}^m$ sont des données du problème, et $f : \mathbb{R}^n \rightarrow \mathbb{R}$ désigne la fonction objectif.

L'ensemble des points satisfaisant les contraintes, noté \mathcal{S} , est appelé *ensemble réalisable* et s'écrit

$$\mathcal{S} = \{x \in \mathbb{R}^n : Ax \leq b\} \subseteq \mathbb{R}^n.$$

Dans toute la suite, nous supposons que \mathcal{S} est non vide et borné.

Ce chapitre regroupe les notions fondamentales et les notations qui serviront de socle aux développements ultérieurs. La thèse porte sur le développement et l'analyse de *mé-*

thodes à directions réalisables (MDR) pour la résolution de problèmes d'optimisation *multi-objectifs* sous contraintes linéaires. À l'instar des problèmes d'optimisation *mono-objectif*, ces approches se déroulent en deux étapes principales : une itération consiste typiquement à déterminer une direction de descente réalisable en résolvant un problème d'optimisation secondaire, appelé *sous-problème de recherche de direction*, noté simplement (SPD), puis à calculer une longueur de pas appropriée au moyen d'une procédure de type *recherche linéaire*.

Afin d'établir les fondations théoriques nécessaires à l'étude de ces méthodes, nous commençons par examiner certaines propriétés géométriques des polyèdres formant l'ensemble \mathcal{S} , notamment la structure convexe, les points extrêmes et les faces. Nous rappelons ensuite des notions analytiques relatives aux fonctions considérées, en particulier la convexité, la différentiabilité et les conditions d'optimalité. Au fil des sections, nous fixons enfin les notations et les conventions utilisées tout au long de la thèse afin d'assurer la cohérence et la clarté des développements. Pour cette partie, nous nous appuyons notamment sur les références bibliographiques suivantes [10, 57, 81, 16].

1.1 Notions préliminaires

Soit $\mathcal{M} = \{1, 2, \dots, m\}$ l'ensemble des indices des contraintes. Pour tout $\bar{x} \in \mathcal{S}$, l'ensemble des contraintes actives en \bar{x} est défini par

$$J(\bar{x}) = \{j \in \mathcal{M} : A_j \bar{x} - b_j = 0\},$$

où A_j désigne la j -ième ligne de la matrice A .

1.1.1 Normes et distances dans \mathbb{R}^n

Les notions de norme et de distance jouent un rôle central en optimisation, notamment pour mesurer la taille des directions, définir des voisinages, exprimer des critères d'arrêt, et analyser la stabilité ainsi que la convergence des méthodes numériques.

Définition 1.1 (Norme). *Une application $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}_+$ est appelée norme si, pour tous $x, y \in \mathbb{R}^n$ et tout $\alpha \in \mathbb{R}$, elle vérifie :*

1. $\|x\| \geq 0$ et $\|x\| = 0 \Leftrightarrow x = 0$,
2. $\|\alpha x\| = |\alpha| \|x\|$,
3. $\|x + y\| \leq \|x\| + \|y\|$.

Définition 1.2 (Distance induite par une norme). Soit $\|\cdot\|$ une norme sur \mathbb{R}^n . La fonction $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$ définie par

$$d(x, y) = \|x - y\|$$

est une distance.

Pour $x \in \mathbb{R}^n$, on définit, pour $q \in [1, +\infty)$, la norme ℓ_q par

$$\|x\|_q = \left(\sum_{i=1}^n |x_i|^q \right)^{1/q},$$

et la norme ℓ_∞ par

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

Les cas $q = 1$ et $q = 2$ correspondent respectivement à la norme ℓ_1 et à la norme euclidienne. Lorsque $\|\cdot\| = \|\cdot\|_2$, la norme euclidienne d'un vecteur $\bar{x} \in \mathbb{R}^n$ est donnée par

$$\|\bar{x}\|_2 = \sqrt{\sum_{i=1}^n \bar{x}_i^2}.$$

1.1.2 Boules, voisinages et notions locales

Dans l'espace euclidien \mathbb{R}^n , de nombreuses notions fondamentales en optimisation, telles que le minimum local, l'intérieur, la continuité, la convergence et les directions réalisables, sont intrinsèquement *locales*. Pour formaliser rigoureusement ce que signifie « proche de \bar{x} », on utilise les *boules* associées à une norme.

Définition 1.3. Soit $\|\cdot\|$ une norme sur \mathbb{R}^n . Pour un point $\bar{x} \in \mathbb{R}^n$ et un rayon $\epsilon > 0$, la boule ouverte de centre \bar{x} et de rayon ϵ est définie par

$$B(\bar{x}, \epsilon) = \{y \in \mathbb{R}^n : \|y - \bar{x}\| < \epsilon\}.$$

Autrement dit, $B(\bar{x}, \epsilon)$ est l'ensemble de tous les points y dont la distance à \bar{x} , mesurée par la norme $\|\cdot\|$, est strictement inférieure à ϵ . La boule fermée correspondante est

$$\bar{B}(\bar{x}, \epsilon) = \{y \in \mathbb{R}^n : \|y - \bar{x}\| \leq \epsilon\}.$$

La notion de boule joue un rôle structurant, car elle fournit une notion précise de *voisinage* d'un point. Elle constitue l'outil standard pour exprimer des propriétés qui ne concernent pas tout l'espace, mais seulement un comportement dans un périmètre arbitrairement petit autour de \bar{x} .

1.2 Quelques notions sur les polyèdres

Rappelons que l'ensemble réalisable du problème (1.1) est donné par

$$\mathcal{S} = \{x \in \mathbb{R}^n : Ax \leq b\},$$

où $A \in \mathbb{R}^{m \times n}$ et $b \in \mathbb{R}^m$. Un tel ensemble est une intersection finie de demi-espaces fermés ; il constitue donc un ensemble convexe et fermé.

Dans la littérature anglo-saxonne, un *polyhedron* désigne une intersection finie de demi-espaces, éventuellement non bornée, tandis qu'un *polytope* correspond à un *polyhedron* borné, équivalent à l'enveloppe convexe d'un nombre fini de points [101]. En revanche, dans la littérature francophone, le terme *polyèdre* est couramment utilisé pour désigner un ensemble convexe défini par des inégalités linéaires, généralement borné [13]. Conformément à cet usage, nous appellerons \mathcal{S} un polyèdre.

1.2.1 Intérieur, frontière et intérieur relatif

Définition 1.4 (Point intérieur). *Un point $\bar{x} \in \mathcal{S}$ est un point intérieur de \mathcal{S} s'il existe $\epsilon > 0$ tel que*

$$B(\bar{x}, \epsilon) \subseteq \mathcal{S}.$$

L'ensemble des points intérieurs est noté $\text{int}(\mathcal{S})$.

Lorsque $\text{int}(\mathcal{S}) \neq \emptyset$, les contraintes sont localement inactives autour de tout point intérieur, ce qui simplifie l'analyse locale et la recherche de directions réalisables.

Cependant, il est fréquent en optimisation que \mathcal{S} soit contenu dans un sous-espace affine strict de \mathbb{R}^n , par exemple lorsqu'on se restreint à une face. Dans ce cas, l'intérieur au sens usuel peut être vide, ce qui motive la notion suivante.

Définition 1.5 (Enveloppe affine). *L'enveloppe affine de \mathcal{S} est le plus petit sous-espace affine contenant \mathcal{S} , noté $\text{aff}(\mathcal{S})$.*

Définition 1.6 (Intérieur relatif). *Soit $\text{aff}(\mathcal{S})$ l'enveloppe affine de \mathcal{S} . Un point $\bar{x} \in \mathcal{S}$ appartient à l'intérieur relatif de \mathcal{S} , noté $\text{ri}(\mathcal{S})$, s'il existe $\epsilon > 0$ tel que*

$$B(\bar{x}, \epsilon) \cap \text{aff}(\mathcal{S}) \subseteq \mathcal{S}.$$

L'intérieur relatif joue un rôle fondamental dans les théorèmes de séparation, les conditions de qualification des contraintes et l'analyse des méthodes d'optimisation évoluant sur des faces de \mathcal{S} [81].

1.2.2 Points extrêmes

Définition 1.7 (Point extrême). *Un point $\bar{x} \in \mathcal{S}$ est un point extrême, ou sommet, si, pour tous $x^1, x^2 \in \mathcal{S}$ et tout $\lambda \in (0, 1)$,*

$$\bar{x} = \lambda x^1 + (1 - \lambda)x^2 \quad \Rightarrow \quad x^1 = x^2 = \bar{x}.$$

Un point extrême ne peut donc pas être exprimé comme une combinaison convexe stricte de deux points distincts de \mathcal{S} . En particulier, aucun point intérieur n'est extrême.

Les points extrêmes sont essentiels en optimisation linéaire et convexe. Le théorème de Krein–Milman garantit que tout polyèdre convexe borné est l'enveloppe convexe de l'ensemble fini de ses points extrêmes [28]. De nombreuses méthodes reposent sur leur énumération, telles que la méthode de description double [44], les méthodes pivotales issues du simplexe [7], ou encore les algorithmes de recherche inverse [8].

1.2.3 Faces d'un polyèdre

Définition 1.8 (Face). *Un sous-ensemble non vide $F \subseteq \mathcal{S}$ est une face de \mathcal{S} s'il existe un ensemble d'indices $J \subseteq \{1, \dots, m\}$ tel que*

$$F = \{x \in \mathcal{S} : A_j x = b_j, \forall j \in J\}.$$

La dimension de la face associée à J est donnée par

$$\dim(F) = n - \text{rank}(A_J),$$

où A_J est la sous-matrice de A correspondant aux contraintes actives.

Les points extrêmes correspondent aux faces de dimension zéro, les arêtes aux faces de dimension un, et le polyèdre \mathcal{S} lui-même est une face, d'une dimension égale à la dimension affine de \mathcal{S} .

1.2.4 Boîte englobante

Définition 1.9 (Boîte englobante). *La boîte englobante du polyèdre $\mathcal{S} \subset \mathbb{R}^n$, notée $[LB, UB]$, est le plus petit pavé aligné sur les axes contenant \mathcal{S} . Elle est définie composante par composante par*

$$LB_i = \min_{x \in \mathcal{S}} x_i, \quad UB_i = \max_{x \in \mathcal{S}} x_i, \quad i = 1, \dots, n.$$

Autrement dit,

$$[LB, UB] = \{x \in \mathbb{R}^n : LB_i \leq x_i \leq UB_i, i = 1, \dots, n\}.$$

Le calcul exact des bornes LB_i et UB_i requiert la résolution de $2n$ programmes linéaires, chacun correspondant à la minimisation ou à la maximisation d'une composante x_i sur l'ensemble réalisable \mathcal{S} [10].

En pratique, lorsque la dimension est élevée, on recourt souvent à des bornes approchées afin de limiter le coût de calcul. La boîte englobante joue néanmoins un rôle important : elle fournit un encadrement simple de l'ensemble réalisable, facilite la visualisation géométrique, sert de domaine de tirage pour les méthodes d'échantillonnage aléatoire et intervient dans certaines procédures d'approximation du volume.

1.2.5 Volume du polyèdre

Le volume du polyèdre réalisable \mathcal{S} , noté $\text{Vol}(\mathcal{S})$, constitue une caractéristique géométrique fondamentale lorsque l'on analyse un problème d'optimisation sous un angle global ou probabiliste. En effet, au-delà de la simple description de l'ensemble réalisable, le volume fournit une mesure quantitative de sa « taille » dans l'espace \mathbb{R}^n . Cette information est particulièrement utile pour évaluer la difficulté géométrique d'un problème, comparer différentes régions réalisables ou encore apprécier l'effet de contraintes supplémentaires sur la réduction de l'espace de recherche.

Dans le cadre des méthodes d'échantillonnage aléatoire, notamment de type Monte Carlo ou Hit-and-Run, le volume joue un rôle central. Supposons que l'on génère des points uniformément dans un ensemble englobant B contenant \mathcal{S} , par exemple une boîte englobante. La probabilité qu'un point tiré appartienne à \mathcal{S} s'écrit

$$\mathbb{P}(x \in \mathcal{S}) = \frac{\text{Vol}(\mathcal{S})}{\text{Vol}(B)}.$$

Ainsi, le rapport des volumes détermine directement le taux d'acceptation d'une méthode de rejet et, par conséquent, le coût nécessaire pour obtenir un échantillon réalisable de taille donnée. Plus généralement, dans l'analyse de complexité des algorithmes aléatoires en grande dimension, des bornes faisant intervenir le volume, ou des rapports de volumes entre ensembles convexes, apparaissent naturellement pour caractériser la difficulté intrinsèque du problème.

Le calcul exact du volume d'un polyèdre est cependant un problème #P-difficile dès que la dimension $n \geq 5$ [33]. Parmi les approches exactes figurent la décomposition en simplexes, la méthode de Lasserre [62] et la méthode de Lawrence [63], dont la complexité croît généralement de manière exponentielle avec la dimension. Ces méthodes restent donc limitées à des problèmes de taille modérée.

Pour les dimensions élevées, on privilégie des techniques d'approximation fondées

sur des méthodes de Monte Carlo. Le principe consiste à estimer le volume par

$$\text{Vol}(\mathcal{S}) \approx \frac{G_{\mathcal{S}}}{G} \text{Vol}(B(\mathcal{S})),$$

où G désigne le nombre total de points générés uniformément dans une boîte englobante $B(\mathcal{S})$ et $G_{\mathcal{S}}$ le nombre de points appartenant effectivement à \mathcal{S} [74]. Ces méthodes sont particulièrement adaptées aux grandes dimensions et constituent le fondement théorique et pratique des stratégies d'échantillonnage utilisées dans cette thèse, notamment celles de type Hit-and-Run.

1.3 Fonctions différentiables

Soit $f : \mathcal{S} \rightarrow \mathbb{R}$ et soit $x \in \mathcal{S}$. Pour tout $i \in \{1, 2, \dots, n\}$, si la limite

$$\frac{\partial f}{\partial x_i}(x) = \lim_{t \rightarrow 0} \frac{f(x + te_i) - f(x)}{t}$$

existe, alors elle est appelée la i -ième dérivée partielle de f au point x , où e_i désigne le i -ième vecteur canonique.

Si toutes les dérivées partielles de f existent au point x , alors le gradient de f en x , noté $\nabla f(x)$, est le vecteur

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^{\top}.$$

La fonction f est dite continûment différentiable sur \mathcal{S} si toutes ses dérivées partielles existent et sont continues sur \mathcal{S} .

En outre, la fonction f est dite deux fois continûment différentiable sur \mathcal{S} si toutes ses dérivées partielles d'ordre deux existent et sont continues sur \mathcal{S} . Dans ce cadre, les dérivées partielles croisées sont symétriques : pour tous i, j et tout $x \in \mathcal{S}$,

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \frac{\partial^2 f}{\partial x_j \partial x_i}(x).$$

La matrice hessienne de f au point $x \in \mathcal{S}$ est alors la matrice nn définie par

$$[\nabla^2 f(x)]_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, \dots, n.$$

Dans la suite, nous rappelons les développements de Taylor au premier et au second ordre de f au voisinage d'un point \bar{x} . Ces approximations locales sont essentielles pour

analyser le comportement de f et justifier les propriétés de descente des méthodes de directions réalisables (MDR).

Supposons que f soit continûment différentiable au voisinage de \bar{x} . Alors, pour x proche de \bar{x} , on a

$$f(x) = f(\bar{x}) + (x - \bar{x})^\top \nabla f(\bar{x}) + R_1(x),$$

où $R_1(x)$ est un terme d'erreur continu vérifiant

$$\lim_{x \rightarrow \bar{x}} \frac{R_1(x)}{\|x - \bar{x}\|} = 0.$$

Supposons maintenant que f soit deux fois continûment différentiable au voisinage de \bar{x} . Alors, pour x proche de \bar{x} ,

$$f(x) = f(\bar{x}) + (x - \bar{x})^\top \nabla f(\bar{x}) + \frac{1}{2}(x - \bar{x})^\top \nabla^2 f(\bar{x})(x - \bar{x}) + R_2(x),$$

où $R_2(x)$ est un terme d'erreur continu vérifiant

$$\lim_{x \rightarrow \bar{x}} \frac{R_2(x)}{\|x - \bar{x}\|^2} = 0.$$

1.4 Fonction convexe

La notion de convexité des fonctions occupe une place centrale en optimisation. Elle permet de relier les propriétés locales aux propriétés globales du problème étudié. En effet, la convexité simplifie considérablement l'analyse théorique et la conception des algorithmes.

Définition 1.10. Soit $f : \mathcal{S} \rightarrow \mathbb{R}$. La fonction f est convexe sur \mathcal{S} si, pour tous $x_1, x_2 \in \mathcal{S}$ et tout $\lambda \in [0, 1]$,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

De plus, f est strictement convexe sur \mathcal{S} si l'inégalité précédente est stricte ($<$) pour $x_1 \neq x_2$ et $\lambda \in (0, 1)$.

Géométriquement, dans la figure (1.1), la courbe rouge représente le graphe de f . La droite pointillée reliant $(x_1, f(x_1))$ à $(x_2, f(x_2))$ est une sécante. La convexité se traduit par le fait que le graphe de f reste entièrement au-dessous, ou sur, cette sécante entre x_1 et x_2 . Cela correspond précisément à l'inégalité de convexité précédente, valable pour tout $\lambda \in [0, 1]$.

Dans le cas où f est continûment différentiable sur \mathcal{S} , les résultats suivants donnent des caractérisations équivalentes de la convexité.

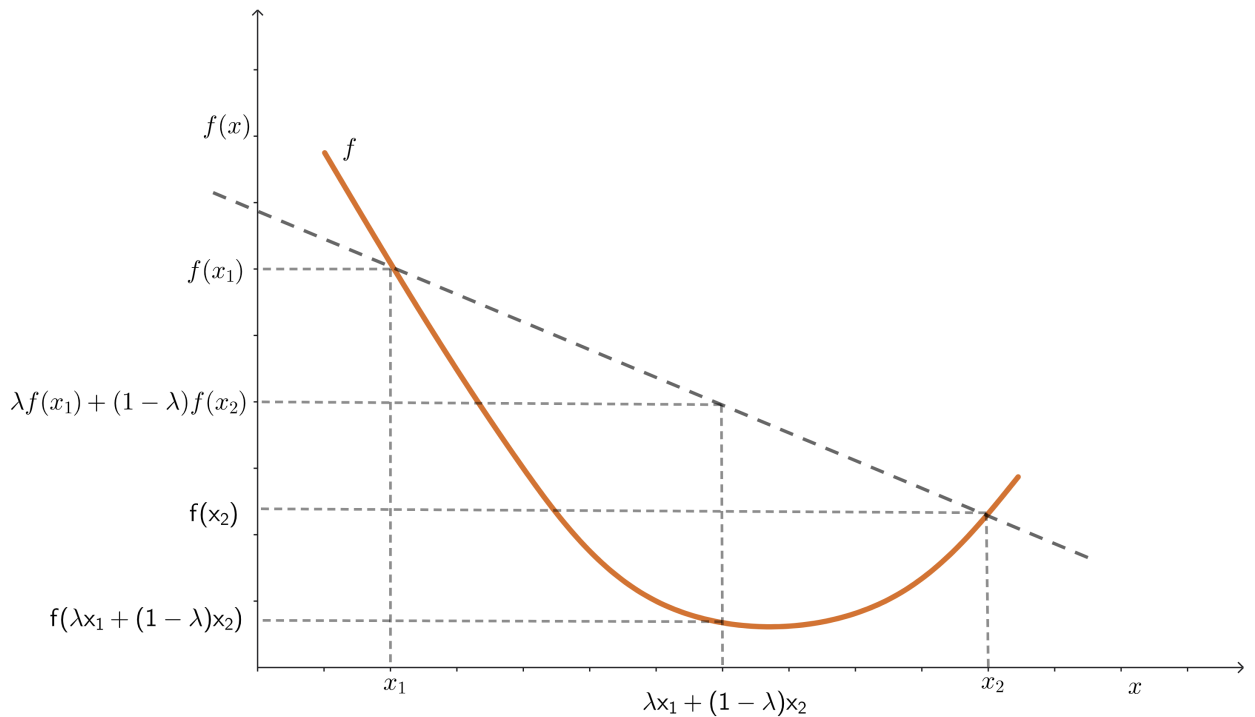


FIGURE 1.1 – Illustration d'une fonction convexe

Lemme 1.1. La fonction f est convexe sur \mathcal{S} si et seulement si, pour tout $\bar{x} \in \mathcal{S}$,

$$f(x) \geq f(\bar{x}) + \nabla f(\bar{x})^\top (x - \bar{x}) \quad \forall x \in \mathcal{S}.$$

De même, f est strictement convexe sur \mathcal{S} si et seulement si, pour tout $\bar{x} \in \mathcal{S}$,

$$f(x) > f(\bar{x}) + \nabla f(\bar{x})^\top (x - \bar{x}) \quad \forall x \in \mathcal{S}, x \neq \bar{x}.$$

Lemme 1.2. La fonction f est convexe sur \mathcal{S} si et seulement si

$$(\nabla f(x_2) - \nabla f(x_1))^\top (x_2 - x_1) \geq 0 \quad \forall x_1, x_2 \in \mathcal{S}.$$

Elle est strictement convexe sur \mathcal{S} si et seulement si

$$(\nabla f(x_2) - \nabla f(x_1))^\top (x_2 - x_1) > 0 \quad \forall x_1, x_2 \in \mathcal{S}, x_1 \neq x_2.$$

Définition 1.11. Une matrice symétrique $H \in \mathbb{R}^{n \times n}$ est dite définie positive, respectivement définie négative, si $d^\top H d > 0$, respectivement $d^\top H d < 0$, pour tout $d \in \mathbb{R}^n \setminus \{0\}$. Elle est dite semi-définie positive, respectivement semi-définie négative, si $d^\top H d \geq 0$, respectivement $d^\top H d \leq 0$, pour tout $d \in \mathbb{R}^n$. Si aucun de ces cas ne s'applique, la matrice H est dite indéfinie.

Lemme 1.3. Supposons que f est deux fois différentiable sur \mathcal{S} . Alors f est convexe sur \mathcal{S} si et

seulement si sa hessienne $\nabla^2 f(x)$ est semi-définie positive en tout point $x \in \mathcal{S}$, c'est-à-dire

$$d^\top \nabla^2 f(x) d \geq 0 \quad \forall x \in \mathcal{S}, \forall d \in \mathbb{R}^n.$$

De même, f est strictement convexe sur \mathcal{S} si et seulement si $\nabla^2 f(x)$ est définie positive en tout point $x \in \mathcal{S}$, c'est-à-dire

$$d^\top \nabla^2 f(x) d > 0 \quad \forall x \in \mathcal{S}, \forall d \in \mathbb{R}^n \setminus \{0\}.$$

1.5 Minimum local, minimum global et point stationnaire

Dans cette section, nous clarifions les notions de minimum local et de minimum global pour le problème (1.1), ainsi que leur lien avec la notion de point stationnaire. Nous mettons ensuite en évidence le rôle central de la convexité de la fonction objectif lorsque l'ensemble réalisable \mathcal{S} est convexe.

Définition 1.12 (Minimum local). *Un point $x^* \in \mathcal{S}$ est appelé minimum local du problème (P) s'il existe $\varepsilon > 0$ tel que*

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{S} \cap B(x^*, \varepsilon).$$

Ainsi, un minimum local est un point qui minimise la fonction objectif par rapport aux points réalisables suffisamment proches.

Définition 1.13 (Minimum global). *Un point $x^* \in \mathcal{S}$ est appelé minimum global du problème (P) si*

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{S}.$$

Un minimum global est donc optimal sur l'ensemble réalisable tout entier, et non seulement dans un voisinage.

Lorsque f est continûment différentiable sur \mathcal{S} , la notion de point stationnaire se définit comme suit.

Définition 1.14 (Point stationnaire). *Un point $x^* \in \mathcal{S}$ est appelé point stationnaire du problème (P) si*

$$\nabla f(x^*)^\top (x - x^*) \geq 0 \quad \forall x \in \mathcal{S}.$$

Le résultat suivant établit le lien fondamental entre minimum local et stationnarité.

Lemme 1.4. *Si $x^* \in \mathcal{S}$ est un minimum local du problème (P) et si f est différentiable en x^* , alors x^* est un point stationnaire.*

Démonstration. Supposons que x^* soit un minimum local et qu'il ne soit pas stationnaire. Alors il existe $x \in \mathcal{S}$ tel que

$$\nabla f(x^*)^\top (x - x^*) < 0.$$

Posons $d = x - x^*$. Comme \mathcal{S} est convexe, pour tout $t \in (0, 1)$ on a

$$x^* + td \in \mathcal{S}.$$

Le développement de Taylor au premier ordre donne

$$f(x^* + td) = f(x^*) + t\nabla f(x^*)^\top d + o(t),$$

avec $\lim_{t \rightarrow 0^+} \frac{o(t)}{t} = 0$. Comme $\nabla f(x^*)^\top d < 0$, il existe $t_0 > 0$ tel que, pour tout $t \in (0, t_0)$,

$$f(x^* + td) < f(x^*),$$

ce qui contredit le caractère minimal local de x^* . □

Ainsi, la stationnarité constitue une condition nécessaire d'optimalité locale. En revanche, en général, un point stationnaire n'est pas nécessairement un minimum; il peut s'agir d'un point selle.

Dans toute la suite, l'ensemble réalisable \mathcal{S} est supposé convexe, car il est défini par des contraintes linéaires. Dans ce cadre, la convexité du problème (P) repose uniquement sur la convexité de la fonction objectif f .

— Si f est convexe sur \mathcal{S} , alors tout minimum local est automatiquement un minimum global.

— De plus, dans ce cas, tout point stationnaire est un minimum global.

En effet, si f est convexe, on a pour tout $x \in \mathcal{S}$

$$f(x) \geq f(x^*) + \nabla f(x^*)^\top (x - x^*).$$

Ainsi, si x^* est stationnaire, le membre de droite est supérieur ou égal à $f(x^*)$, ce qui implique

$$f(x) \geq f(x^*) \quad \forall x \in \mathcal{S},$$

et donc l'optimalité globale.

Enfin, si f est strictement convexe sur \mathcal{S} , le minimum global est unique. En effet, l'existence de deux minimums distincts contredirait la stricte convexité.

1.6 Conditions d'optimalité

Dans cette section, nous établissons les conditions d'optimalité du problème (1.1) sous contraintes linéaires. Lorsque l'ensemble réalisable \mathcal{S} est convexe et que la fonction objectif f est différentiable, les conditions de Karush–Kuhn–Tucker (KKT) fournissent une caractérisation analytique précise des minima locaux, et, sous convexité, des minima globaux.

Le lemme de Farkas suivant joue un rôle central dans la démonstration des conditions d'optimalité sous contraintes linéaires.

Lemme 1.5 (Lemme de Farkas). *Soient $c \in \mathbb{R}^n$ et $A \in \mathbb{R}^{mn}$. Alors exactement l'un des deux systèmes suivants admet une solution :*

$$\text{(Système I)} : \exists x \in \mathbb{R}^n \text{ tel que } Ax \leq 0, c^\top x > 0,$$

$$\text{(Système II)} : \exists y \in \mathbb{R}^m \text{ tel que } A^\top y = c, y \geq 0.$$

Une formulation équivalente, dite forme implicationnelle du lemme de Farkas, est la suivante.

Lemme 1.6 (Forme implicationnelle). *Soient $c \in \mathbb{R}^n$ et $A \in \mathbb{R}^{mn}$. Les deux assertions suivantes sont équivalentes :*

$$(A) : \forall x \in \mathbb{R}^n, Ax \leq 0 \Rightarrow c^\top x \leq 0,$$

$$(B) : \exists y \in \mathbb{R}^m \text{ tel que } y \geq 0 \text{ et } A^\top y = c.$$

Ce résultat constitue le fondement théorique des conditions d'optimalité sous contraintes linéaires.

Théorème 1.1 (Conditions nécessaires de Karush–Kuhn–Tucker). *Soit $x^* \in \mathcal{S}$ un minimum local du problème (1.1), supposé différentiable. Alors il existe un vecteur de multiplicateurs $\alpha \in \mathbb{R}^m$, $\alpha \geq 0$, tel que*

$$\nabla f(x^*) + \sum_{j=1}^m \alpha_j A_j^\top = 0, \tag{1.2}$$

$$\alpha_j (A_j x^* - b_j) = 0, \quad j = 1, \dots, m. \tag{1.3}$$

Démonstration. Comme x^* est un minimum local, il n'existe aucune direction de descente réalisable en x^* .

Pour $t > 0$ suffisamment petit, une direction d est réalisable en x^* si et seulement si

$$A_j d \leq 0 \quad \forall j \in J(x^*),$$

où $J(x^*)$ désigne l'ensemble des contraintes actives.

L'absence de direction de descente réalisable signifie donc que le système

$$A_j d \leq 0 \quad (j \in J(x^*)), \quad \nabla f(x^*)^\top d < 0$$

n'admet aucune solution.

En appliquant le lemme de Farkas à ce système, on obtient l'existence de multiplicateurs $\alpha_j \geq 0$ ($j \in J(x^*)$) tels que

$$\nabla f(x^*) + \sum_{j \in J(x^*)} \alpha_j A_j^\top = 0.$$

En posant $\alpha_j = 0$ pour $j \notin J(x^*)$, on obtient (1.2).

La condition de complémentarité (1.3) résulte directement de la définition de l'ensemble actif. \square

Lorsque le problème est convexe, les conditions KKT ne sont plus seulement nécessaires, mais également suffisantes.

Théorème 1.2 (Conditions suffisantes sous convexité). *Supposons que f soit convexe et continûment différentiable sur \mathcal{S} . Alors $x^* \in \mathcal{S}$ est solution optimale globale de (1.1) si et seulement s'il existe $\alpha \geq 0$ satisfaisant (1.2)–(1.3).*

Démonstration. La nécessité découle du théorème précédent.

Pour la suffisance, supposons que les conditions KKT soient vérifiées. Considérons la fonction de Lagrange

$$h(x) = f(x) + \sum_{j=1}^m \alpha_j (A_j x - b_j).$$

Comme f est convexe et que les fonctions $x \mapsto A_j x - b_j$ sont affines, la fonction h est convexe. La condition (1.2) implique $\nabla h(x^*) = 0$, donc x^* minimise h sur \mathbb{R}^n .

Pour tout $x \in \mathcal{S}$, on a $\sum_{j=1}^m \alpha_j (A_j x - b_j) \leq 0$ et, par complémentarité, $\sum_{j=1}^m \alpha_j (A_j x^* - b_j) = 0$. On en déduit

$$f(x^*) \leq f(x),$$

ce qui montre que x^* est optimal global. \square

1.7 Direction réalisable et direction de descente

Soit $\bar{x} \in \mathcal{S}$. On s'intéresse au comportement local du problème au voisinage de \bar{x} , en étudiant les directions dans lesquelles il est possible de se déplacer tout en conservant la faisabilité, et éventuellement en améliorant la valeur de la fonction objectif.

Définition 1.15. *Un vecteur non nul $d \in \mathbb{R}^n$ est appelé direction réalisable en \bar{x} s'il existe $\delta > 0$ tel que*

$$\bar{x} + \lambda d \in \mathcal{S} \quad \forall \lambda \in (0, \delta). \quad (1.4)$$

Il est appelé direction de descente en \bar{x} s'il existe $\delta > 0$ tel que

$$f(\bar{x} + \lambda d) < f(\bar{x}) \quad \text{et} \quad \bar{x} + \lambda d \in \mathcal{S} \quad \forall \lambda \in (0, \delta). \quad (1.5)$$

On note

$$\Phi(\bar{x}) = \{d \in \mathbb{R}^n \setminus \{0\} : \exists \delta > 0, \bar{x} + \lambda d \in \mathcal{S} \quad \forall \lambda \in (0, \delta)\}$$

le cône des directions réalisables en \bar{x} , et

$$\Delta(\bar{x}) = \{d \in \mathbb{R}^n \setminus \{0\} : \exists \delta > 0, f(\bar{x} + \lambda d) < f(\bar{x}) \quad \forall \lambda \in (0, \delta)\}$$

le cône des directions de descente en \bar{x} .

Dans les méthodes de directions réalisables (MDR), la détermination d'une nouvelle itération repose sur la recherche d'une direction $d \in \Phi(\bar{x})$, et si possible $d \in \Phi(\bar{x}) \cap \Delta(\bar{x})$, au moyen d'un sous-problème de recherche de direction (SPD).

Lemme 1.7. *Soit $\bar{x} \in \mathcal{S}$. Alors :*

— *d est une direction réalisable en \bar{x} si et seulement si*

$$A_j d \leq 0 \quad \forall j \in J(\bar{x}).$$

— *Si f est continûment différentiable et si $\nabla f(\bar{x})^\top d < 0$, alors d est une direction de descente en \bar{x} .*

Démonstration. Supposons d'abord que d soit réalisable. Pour tout $j \in J(\bar{x})$, on a $A_j \bar{x} = b_j$ et, pour $\lambda > 0$ suffisamment petit,

$$A_j(\bar{x} + \lambda d) \leq b_j.$$

On en déduit $b_j + \lambda A_j d \leq b_j$, donc $A_j d \leq 0$.

Réciproquement, supposons que $A_j d \leq 0$ pour tout $j \in J(\bar{x})$. Pour tout $i \notin J(\bar{x})$, on a

$A_i \bar{x} < b_i$; par continuité, il existe $\delta_i > 0$ tel que $A_i(\bar{x} + \lambda d) \leq b_i$ pour tout $\lambda \in (0, \delta_i)$. En posant $\delta = \min_{i \in J(\bar{x})} \delta_i$, on obtient $\bar{x} + \lambda d \in \mathcal{S}$ pour tout $\lambda \in (0, \delta)$, ce qui établit l'admissibilité.

Enfin, si f est continûment différentiable, le développement de Taylor au premier ordre donne

$$f(\bar{x} + \lambda d) = f(\bar{x}) + \lambda \nabla f(\bar{x})^\top d + o(\lambda), \quad \frac{o(\lambda)}{\lambda} \rightarrow 0.$$

Si $\nabla f(\bar{x})^\top d < 0$, il existe $\delta' > 0$ tel que $f(\bar{x} + \lambda d) < f(\bar{x})$ pour tout $\lambda \in (0, \delta')$. \square

On peut ainsi écrire

$$\Phi(\bar{x}) = \{d \in \mathbb{R}^n \setminus \{0\} : A_j d \leq 0, \forall j \in J(\bar{x})\}, \quad \Delta(\bar{x}) = \{d \in \mathbb{R}^n \setminus \{0\} : \nabla f(\bar{x})^\top d < 0\}.$$

Un déplacement infinitésimal dans une direction de $\Phi(\bar{x})$ préserve la faisabilité, tandis qu'un déplacement dans une direction de $\Delta(\bar{x})$ diminue la valeur de la fonction objectif. Une condition nécessaire d'optimalité locale s'énonce alors comme suit.

Théorème 1.3. *Si \bar{x} est un minimum local du problème (1.1), alors*

$$\Phi(\bar{x}) \cap \Delta(\bar{x}) = \emptyset.$$

Démonstration. S'il existait $d \in \Phi(\bar{x}) \cap \Delta(\bar{x})$, alors pour $\lambda > 0$ suffisamment petit, on aurait simultanément $\bar{x} + \lambda d \in \mathcal{S}$ et $f(\bar{x} + \lambda d) < f(\bar{x})$, ce qui contredirait l'optimalité locale de \bar{x} . \square

Pour illustrer géométriquement la notion de direction de descente réalisable, considérons le problème suivant :

$$\left\{ \begin{array}{l} \min_{x \in \mathbb{R}^2} f(x) = (x_1 - 6)^2 + (x_2 - 1)^2, \\ \text{s.c. } A_1 := 3x_1 + 2x_2 - 12 \leq 0, \\ \quad A_2 := -x_1 + 2x_2 - 4 \leq 0, \\ \quad A_3 := -x_1 \leq 0, \\ \quad A_4 := -x_2 \leq 0. \end{array} \right. \quad (1.6)$$

L'ensemble réalisable est donc le polyèdre

$$\mathcal{S} = \{x \in \mathbb{R}^2 : A_j x \leq b_j, j = 1, \dots, 4\}.$$

Dans cet exemple, au point $b = (2, 3)$, les contraintes actives sont A_1 et A_2 ; donc

$$J(b) = \{1, 2\}.$$

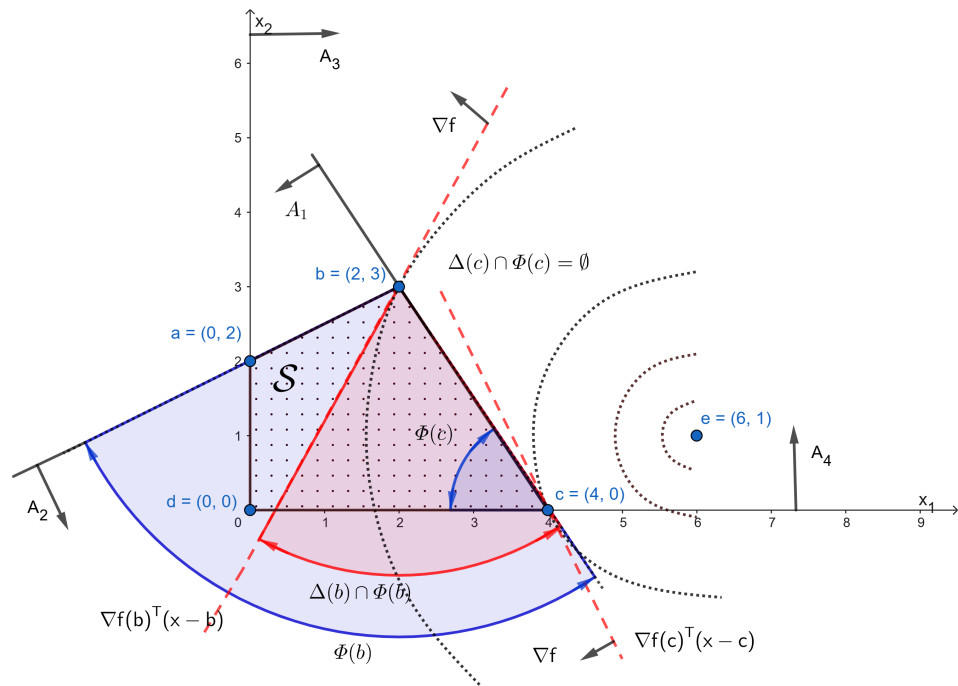


FIGURE 1.2 – Interprétation géométrique de la direction de descente réalisable

Le cône des directions réalisables en b s'écrit donc

$$\Phi(b) = \{d \in \mathbb{R}^2 \setminus \{0\} : 3d_1 + 2d_2 \leq 0, -d_1 + 2d_2 \leq 0\}.$$

Par ailleurs, $\nabla f(b) = (-8, 4)^\top$; le cône des directions de descente au point b est

$$\Delta(b) = \{d \in \mathbb{R}^2 \setminus \{0\} : -8d_1 + 4d_2 < 0\}.$$

On vérifie que l'intersection $\Phi(b) \cap \Delta(b)$ est non vide, par exemple la direction $d = (1, -2)$ appartient à cette intersection. Il existe donc en b une direction qui permet simultanément de rester dans l'ensemble réalisable et de diminuer la valeur de la fonction objectif. Par conséquent, le point b n'est pas optimal.

En revanche, au point $c = (4, 0)$, les contraintes actives sont A_1 et A_4 ; donc

$$J(c) = \{1, 4\}.$$

Le cône des directions réalisables en c est alors

$$\Phi(c) = \{d \in \mathbb{R}^2 \setminus \{0\} : 3d_1 + 2d_2 \leq 0, -d_2 \leq 0\}.$$

D'autre part,

$$\nabla f(c) = (-4, -2)^\top, \quad \Delta(c) = \{d \in \mathbb{R}^2 \setminus \{0\} : -4d_1 - 2d_2 < 0\}.$$

Puisque $-d_2 \leq 0$, on a $d_2 \geq 0$. Or, pour qu'un vecteur d appartienne à $\Delta(c)$, il faut que

$$2d_1 + d_2 > 0.$$

Cette condition est incompatible avec l'inégalité précédente. Par conséquent,

$$\Phi(c) \cap \Delta(c) = \emptyset.$$

Cela signifie qu'il n'existe aucune direction qui soit à la fois réalisable et de descente. Autrement dit, toute direction permettant de diminuer la fonction objectif fait sortir de l'ensemble réalisable, tandis que toute direction réalisable ne permet pas une décroissance locale de f . Cette situation traduit un état de stationnarité sous contraintes; ici, elle correspond au fait que $c = (4, 0)$ est la solution optimale du problème considéré.

La figure (1.2) illustre ainsi de manière géométrique les cônes $\Phi(\bar{x})$ et $\Delta(\bar{x})$ en deux points réalisables. Lorsque leur intersection est non vide, il existe une direction de descente réalisable et le point considéré ne peut pas être optimal. À l'inverse, lorsque $\Phi(\bar{x}) \cap \Delta(\bar{x}) = \emptyset$, il n'existe aucune direction réalisable permettant de diminuer localement la fonction objectif, ce qui est cohérent avec les conditions de Karush–Kuhn–Tucker.

2

Méthodes de directions réalisables pour l'optimisation mono-objectif

Introduction

La méthode des directions réalisables (MDR), initialement développée par Zoutendijk [103] pour résoudre des problèmes d'optimisation non linéaire à objectif unique, s'appuie sur un principe simple : à chaque itération, on construit une direction de descente réalisable au point courant, puis on détermine une longueur de pas garantissant à la fois la faisabilité (toutes les itérées restent dans l'ensemble réalisable) et l'amélioration de la fonction objectif. Cette philosophie confère à la MDR plusieurs avantages : (i) une propriété de descente itérative, (ii) la conservation de la faisabilité à toutes les itérations, (iii) une mise en œuvre conceptuellement directe, et (iv) une efficacité numérique appréciable reposant principalement sur des dérivées du premier ordre et la résolution d'un unique sous-problème de recherche de direction à chaque itération.

Dans cette thèse, l'intérêt principal porte sur la construction et l'analyse de sous-problèmes de recherche de direction, notés *SPD* (sous-problème de recherche de direction), qui génèrent des directions réalisables, et si possible de descente, pour les problèmes sous contraintes linéaires. Les variantes de *MDR* se distinguent essentiellement par :

- la définition du *SPD* (ensemble de contraintes utilisées, normalisation de la direction, relaxation, termes quadratiques, etc.);
- la stratégie de calcul du pas (recherche linéaire exacte ou inexacte, pas maximal réalisable, conditions de type Armijo/Wolfe, etc.).

On rappelle que le problème étudié est

$$(P) \quad \begin{cases} \min_x f(x), \\ \text{s.c.} \quad Ax \leq b, \\ \quad \quad x \in \mathbb{R}^n, \end{cases} \quad (2.1)$$

avec ensemble réalisable $\mathcal{S} = \{x \in \mathbb{R}^n : Ax \leq b\}$, supposé non vide et borné. Pour un point faisable $x \in \mathcal{S}$, on note $J(x) = \{j \in \{1, \dots, m\} : A_j x - b_j = 0\}$ l'ensemble des contraintes actives.

2.0.1 Schéma général d'une MDR

Le schéma général suivant décrit la structure commune des méthodes de directions réalisables.

Algorithm 1 Schéma général d'une méthode des directions réalisables (MDR)

- 1: *Initialisation* : choisir $x^0 \in \mathcal{S}$ et poser $k = 0$.
 - 2: *Calcul de direction* : résoudre un SPD au point x^k et obtenir une direction d^k .
 - 3: *Test d'arrêt* : si le SPD ne fournit pas de direction de descente réalisable, arrêter. Alors x^k est stationnaire (et, sous convexité, optimal).
 - 4: *Calcul du pas* : déterminer $\lambda^k > 0$ tel que $x^k + \lambda^k d^k \in \mathcal{S}$ et $f(x^k + \lambda^k d^k) < f(x^k)$.
 - 5: *Mise à jour* : poser $x^{k+1} = x^k + \lambda^k d^k$ puis $k \leftarrow k + 1$.
-

Les sections suivantes détaillent plusieurs SPD classiques, puis présentent des méthodes apparentées (Frank–Wolfe, projection du gradient, gradient réduit) qui exploitent également des directions réalisables, mais avec une construction différente.

2.1 SPD de Zoutendijk

Étant donné $x \in \mathcal{S}$, le Lemme (1.7) montre qu'un vecteur non nul d est une direction réalisable en x si et seulement si

$$A_j d \leq 0, \quad \forall j \in J(x), \quad (2.2)$$

et que si, de plus, $\nabla f(x)^\top d < 0$, alors d est une direction de descente en x . Ainsi, la recherche d'une direction de descente réalisable peut être formulée via le système

$$\begin{cases} \nabla f(x)^\top d < 0, \\ A_j d \leq 0, \quad \forall j \in J(x), \\ d \neq 0. \end{cases} \quad (2.3)$$

Cependant, ce système, pris isolément, ne fournit pas une procédure numérique stable : il est invariant par changement d'échelle (si d convient, alors td convient pour tout $t > 0$), ce qui motive l'introduction d'une normalisation.

Zoutendijk [103] propose de construire d en minimisant une majoration commune (variable auxiliaire z) de la baisse linéarisée de l'objectif et des quantités $A_j d$ associées aux contraintes actives. Cette idée, systématisée par Pironneau et Polak [77], conduit à poser

$$z \geq \nabla f(x)^\top d, \quad z \geq A_j d \quad (j \in J(x)),$$

et à minimiser z sous une contrainte de normalisation sur d . On obtient alors les trois sous-problèmes de recherche de direction de Zoutendijk suivants.

$$\text{SPD_Z1}(x) \begin{cases} \min_{z,d} z, \\ \nabla f(x)^\top d \leq z, \\ A_j d \leq z, \quad \forall j \in J(x), \\ -1 \leq d \leq 1, \\ z \in \mathbb{R}, d \in \mathbb{R}^n, \end{cases} \quad (2.4a)$$

$$\text{SPD_Z2}(x) \begin{cases} \min_{z,d} z, \\ \nabla f(x)^\top d \leq z, \\ A_j d \leq z, \quad \forall j \in J(x), \\ d^\top d \leq 1, \\ z \in \mathbb{R}, d \in \mathbb{R}^n, \end{cases} \quad (2.4b)$$

$$\text{SPD_Z3}(x) \begin{cases} \min_{z,d} z, \\ \nabla f(x)^\top d \leq z, \\ A_j d \leq z, \quad \forall j \in J(x), \\ \nabla f(x)^\top d \geq -1, \\ z \in \mathbb{R}, d \in \mathbb{R}^n. \end{cases} \quad (2.4c)$$

Dans $\text{SPD_Z1}(x)$, l'inégalité $-1 \leq d \leq 1$ est comprise composante par composante. Les problèmes $\text{SPD_Z1}(x)$ et $\text{SPD_Z3}(x)$ sont des programmes linéaires, tandis que $\text{SPD_Z2}(x)$ est un problème convexe avec contrainte quadratique.

Lemme 2.1. *Pour tout $x \in \mathcal{S}$, chacun des sous-problèmes $\text{SPD_Zi}(x)$, $i = 1, 2, 3$, admet au moins une solution réalisable.*

Démonstration. Le couple $(z, d) = (0, 0)$ satisfait les contraintes de $\text{SPD_Zi}(x)$ pour $i = 1, 2, 3$. Ainsi, l'ensemble réalisable de chaque SPD est non vide. \square

Lemme 2.2. *Pour tout $x \in \mathcal{S}$ et pour $i \in \{1, 2, 3\}$, la valeur optimale $z_i^*(x)$ de $\text{SPD_Zi}(x)$ vérifie $z_i^*(x) \leq 0$.*

Démonstration. Par le lemme (2.1), $(z, d) = (0, 0)$ est réalisable et donne la valeur $z = 0$. Comme chaque SPD minimise z , on obtient nécessairement $z_i^*(x) \leq 0$. \square

Intuitivement, si $z^* < 0$, alors on obtient une direction d telle que $\nabla f(x)^\top d \leq z^* < 0$ et $A_j d \leq z^* < 0$ pour $j \in J(x)$. La direction est donc strictement réalisable au premier ordre

pour les contraintes actives et strictement de descente au premier ordre. En revanche, si $z^* = 0$, le SPD ne permet pas de produire une direction de descente réalisable; cela correspond précisément à la condition « absence de direction de descente réalisable », équivalente aux conditions KKT pour des contraintes linéaires (Chapitre 1).

Lemme 2.3. Soit $\bar{x} \in \mathcal{S}$. Pour $i \in \{1, 2, 3\}$, on a :

$$\bar{x} \text{ est un point stationnaire du problème (2.1)} \iff z_i^*(\bar{x}) = 0.$$

Démonstration. (\Rightarrow) Si \bar{x} est stationnaire, alors il n'existe aucune direction $d \neq 0$ telle que $A_j d \leq 0$ pour tout $j \in J(\bar{x})$ et $\nabla f(\bar{x})^\top d < 0$ (voir la caractérisation « absence de direction de descente réalisable » et le lien avec les conditions KKT établi au Chapitre 1). Supposons par contradiction que $z_i^*(\bar{x}) < 0$. Alors il existe (z, d) réalisable avec $z < 0$. En particulier, $\nabla f(\bar{x})^\top d \leq z < 0$ et $A_j d \leq z < 0$ pour tout $j \in J(\bar{x})$, ce qui fournit une direction de descente réalisable, contradiction. Ainsi $z_i^*(\bar{x}) = 0$ (en utilisant aussi $z_i^*(\bar{x}) \leq 0$, lemme (2.2)).

(\Leftarrow) Si $z_i^*(\bar{x}) = 0$, alors aucun couple réalisable ne peut avoir $z < 0$. En particulier, il n'existe pas de direction $d \neq 0$ satisfaisant simultanément $A_j d \leq 0$ ($j \in J(\bar{x})$) et $\nabla f(\bar{x})^\top d < 0$. On retrouve alors la condition d'absence de direction de descente réalisable, équivalente à la stationnarité sous contraintes linéaires, et donc aux conditions KKT du Chapitre 1. \square

Conformément au schéma général (Algorithme (1)), à l'itération k , une fois la direction d^k obtenue par un SPD, on calcule le nouvel itéré

$$x^{k+1} = x^k + \lambda^k d^k, \quad (2.5)$$

où $\lambda^k > 0$ doit préserver la faisabilité et assurer la descente.

Pour un point faisable $x^k \in \mathcal{S}$ et une direction réalisable d^k , on introduit l'ensemble

$$\Omega(x^k) = \left\{ j \in \{1, \dots, m\} \setminus J(x^k) \mid A_j d^k > 0 \right\}. \quad (2.6)$$

Il s'agit des contraintes inactives en x^k , donc strictement satisfaites, mais qui peuvent devenir violées lorsque l'on avance dans la direction d^k .

Proposition 2.1. Pour tout $x^k \in \mathcal{S}$, si d^k est une direction réalisable en x^k , alors $\Omega(x^k) \neq \emptyset$.

Démonstration. Supposons par l'absurde que $\Omega(x^k) = \emptyset$. Alors $A_j d^k \leq 0$ pour tout $j \notin J(x^k)$. Comme d^k est réalisable, on a aussi $A_j d^k \leq 0$ pour tout $j \in J(x^k)$. Ainsi $A d^k \leq 0$, composante par composante, ce qui implique

$$A(x^k + \lambda d^k) \leq A x^k \leq b, \quad \forall \lambda \geq 0,$$

et donc $x^k + \lambda d^k \in \mathcal{S}$ pour tout $\lambda \geq 0$. Cela contredit la bornitude de \mathcal{S} . On conclut que $\Omega(x^k) \neq \emptyset$. \square

Pour les contraintes actives $j \in J(x^k)$, on a $A_j x^k = b_j$ et, par admissibilité, $A_j d^k \leq 0$. Donc

$$A_j(x^k + \lambda d^k) = b_j + \lambda A_j d^k \leq b_j, \quad \forall \lambda \geq 0.$$

En revanche, pour $j \in \Omega(x^k)$, on a $A_j d^k > 0$ et $b_j - A_j x^k > 0$. La contrainte $A_j(x^k + \lambda d^k) \leq b_j$ impose alors

$$\lambda \leq \frac{b_j - A_j x^k}{A_j d^k}.$$

On définit le pas maximal réalisable par

$$\lambda_{\max}^k = \min_{j \in \Omega(x^k)} \left\{ \frac{b_j - A_j x^k}{A_j d^k} \right\} > 0. \quad (2.7)$$

Ainsi, $x^k + \lambda d^k \in \mathcal{S}$ pour tout $\lambda \in (0, \lambda_{\max}^k)$, et le calcul du pas peut se ramener à une recherche linéaire sur l'intervalle $[0, \lambda_{\max}^k]$:

$$\min_{0 < \lambda \leq \lambda_{\max}^k} f(x^k + \lambda d^k). \quad (2.8)$$

Dans la pratique, on peut utiliser une recherche exacte, comme dans certains exemples classiques, ou une recherche inexacte, par exemple avec des conditions d'Armijo/Wolfe, afin de réduire le coût numérique.

2.2 Contraintes quasi-actives, jamming et limites de la variante originale

Le SPD de Zoutendijk n'utilise que les contraintes actives $J(x^k)$. Or, lorsqu'une contrainte est inactive mais très proche de l'être, c'est-à-dire lorsque la marge $b_j - A_j x^k$ est faible, l'ignorer dans le SPD peut produire une direction qui conduit à un pas maximal λ_{\max}^k extrêmement petit, ce qui ralentit fortement la progression.

2.2.1 Contraintes quasi-actives

Une stratégie standard consiste à enrichir l'ensemble des contraintes prises en compte dans le SPD en ajoutant des contraintes dites quasi-actives. Pour un seuil $\varepsilon > 0$, on définit

$$J_\varepsilon(x) = \{j \in \{1, \dots, m\} \mid b_j - A_j x \leq \varepsilon\}. \quad (2.9)$$

Cette idée, discutée notamment par Bazaraa et al. [10], favorise la production de directions permettant des pas plus longs, donc une progression plus régulière. Le choix de ε doit toutefois être calibré : un ε trop grand peut activer artificiellement trop de contraintes, rendre le SPD inutilement conservatif, voire provoquer un arrêt prématuré.

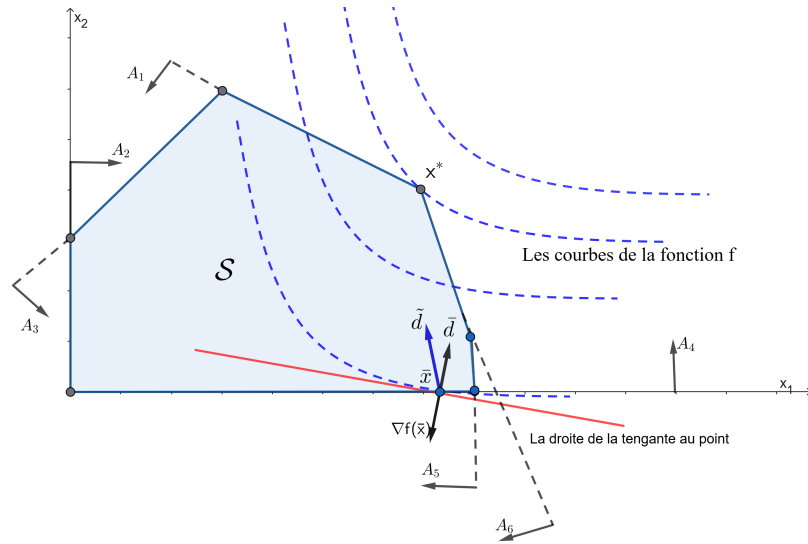


FIGURE 2.1 – Effet des contraintes quasi-actives : impact sur la génération de directions.

Comme illustré dans la Figure (2.1), au point \bar{x} , une seule contrainte est active, mais une autre est quasi-active, avec une marge faible. Si le SPD ne tient compte que de $J(\bar{x})$, la direction obtenue peut être immédiatement bloquée par la contrainte quasi-active, imposant un pas très petit. En ajoutant cette contrainte au SPD, via $J_\varepsilon(\bar{x})$, on obtient une direction plus robuste, autorisant un déplacement plus significatif tout en conservant l'admissibilité.

2.2.2 Jamming et contre-exemple de Wolfe

Malgré ces ajustements, la formulation originale de Zoutendijk peut souffrir d'une limitation théorique : la convergence vers un point satisfaisant les conditions KKT n'est pas garantie en général. Un phénomène classique est le *jamming* (blocage), où les itérations stagnent à un point non stationnaire. Le contre-exemple suivant, dû à Wolfe [93], met en évidence ce comportement.

Exemple 2.1 (Contre-exemple de Wolfe, 1967). *Considérons le problème convexe suivant :*

$$\begin{cases} \min_{x \in \mathbb{R}^3} f(x) = \frac{4}{3}(x_1^2 - x_1 x_2 + x_2^2)^{3/4} - x_3, \\ \text{s.c. } x_1 \geq 0, x_2 \geq 0, 0 \leq x_3 \leq 2. \end{cases} \quad (2.10)$$

La solution optimale unique est $\bar{x} = (0, 0, 2)^\top$, avec $f(\bar{x}) = -2$.

Initialisons l'algorithme au point réalisable $x_1 = (0, a, 0)^\top$, où $0 < a \leq \frac{1}{2\sqrt{2}}$. À chaque itération k , la direction d^k est obtenue en résolvant SPD_Z2, avec normalisation quadratique, puis λ^k est choisie par une recherche linéaire exacte, dans l'esprit de l'analyse de Wolfe.

Au point initial x_1 , les contraintes actives incluent $x_2 \geq 0$ et $x_3 \geq 0$, soit, sous forme $Ax \leq b$, $-x_2 \leq 0$ et $-x_3 \leq 0$. On a alors

$$A_1 = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \nabla f(x_1) = (-\sqrt{a}, 2\sqrt{a}, -1)^\top. \quad (2.11)$$

La direction optimale de SPD_Z2 est

$$d_1 = -\frac{\nabla f(x_1)}{\|\nabla f(x_1)\|},$$

et la recherche linéaire conduit à

$$x_2 = \left(\frac{a}{2}, 0, \frac{\sqrt{a}}{2} \right)^\top. \quad (2.12)$$

En poursuivant, la suite $\{x^k\}$ alterne entre les frontières $x_1 = 0$ et $x_2 = 0$. Plus précisément, pour $k \geq 3$,

$$x^k = \begin{cases} \left(0, \left(\frac{1}{2}\right)^{k-1} a, \frac{1}{2} \sum_{j=0}^{k-2} \left(\frac{a}{2^j}\right)^{1/2} \right)^\top, & \text{si } k \text{ impair,} \\ \left(\left(\frac{1}{2}\right)^{k-1} a, 0, \frac{1}{2} \sum_{j=0}^{k-2} \left(\frac{a}{2^j}\right)^{1/2} \right)^\top, & \text{si } k \text{ pair.} \end{cases} \quad (2.13)$$

Cette suite converge vers

$$\hat{x} = \left(0, 0, \left(1 + \frac{\sqrt{2}}{2}\right) \sqrt{a} \right)^\top, \quad (2.14)$$

et, sous l'hypothèse $a \leq \frac{1}{2\sqrt{2}}$, on obtient $x_3 = \left(1 + \frac{\sqrt{2}}{2}\right) \sqrt{a} \leq \sqrt{2} < 2$, donc la contrainte $x_3 \leq 2$ est strictement inactive au point limite \hat{x} .

Au point \hat{x} , les contraintes actives sont $x_1 \geq 0$ et $x_2 \geq 0$, c'est-à-dire $-x_1 \leq 0$ et $-x_2 \leq 0$, tandis que $\nabla f(\hat{x}) = (0, 0, -1)^\top$. Il n'existe pas de multiplicateurs $\lambda_1, \lambda_2 \geq 0$ tels que

$$\nabla f(\hat{x}) + \lambda_1(-1, 0, 0)^\top + \lambda_2(0, -1, 0)^\top = 0,$$

ce qui viole les conditions KKT. De plus,

$$f(\hat{x}) = -\left(1 + \frac{\sqrt{2}}{2}\right)\sqrt{a} > -2 = f(\bar{x}),$$

confirmant que \hat{x} n'est pas optimal.

Ce blocage est typique du jamming : les directions d^k et/ou les pas λ^k se dégradent, c'est-à-dire tendent vers des valeurs très petites, entraînant une oscillation stérile sans progression vers la contrainte $x_3 \leq 2$, qui reste pourtant déterminante pour atteindre l'optimum \bar{x} . Une approche naturelle pour éviter ce phénomène consiste à inclure davantage de contraintes, actives et inactives, dans le SPD, comme le proposent Topkis et Veinott [89].

2.3 SPD de Topkis–Veinott

Pour remédier aux limitations de la variante de Zoutendijk, notamment l'exclusion des contraintes inactives, Topkis et Veinott [89] proposent un SPD intégrant *toutes* les contraintes, ce qui stabilise les directions obtenues et améliore les garanties théoriques de convergence vers un point satisfaisant les conditions KKT.

Pour harmoniser les notations avec le problème $Ax \leq b$, on introduit la fonction contrainte linéaire

$$e_j(x) = A_j x - b_j \leq 0, \quad j = 1, \dots, m. \quad (2.15)$$

On note également $g(x) = (g_1(x), \dots, g_m(x))^T$. Pour les contraintes actives, $e_j(x) = 0$, et pour les contraintes strictement inactives, $e_j(x) < 0$.

Le SPD de Topkis–Veinott, avec normalisation en boîte, s'écrit alors :

$$\text{SPD_TV}(x) \begin{cases} \min_{z,d} z, \\ \nabla f(x)^T d \leq z, \\ e_j(x) + A_j d \leq z, \quad \forall j = 1, \dots, m, \\ -1 \leq d \leq 1, \\ z \in \mathbb{R}, d \in \mathbb{R}^n. \end{cases} \quad (2.16)$$

Ici, les contraintes $e_j(x) + A_j d \leq z$ combinent l'information locale sur la marge $e_j(x)$ et l'effet directionnel $A_j d$. Lorsque $e_j(x) < 0$, c'est-à-dire lorsque la contrainte est inactive, la marge permet d'autoriser des directions qui pourraient temporairement augmenter $A_j x$, tout en restant contrôlées par le niveau $z \leq 0$. Cette anticipation réduit le risque de jamming et rend la direction plus robuste vis-à-vis des contraintes proches de l'activation.

Lemme 2.4 (Critère d'arrêt pour SPD_TV). *Soit $x \in \mathcal{S}$. Alors x satisfait les conditions KKT du problème (2.1) si et seulement si la valeur optimale z^* de SPD_TV(x) est nulle.*

Démonstration. L'argument est analogue à celui du lemme (2.3). Si $z^* < 0$, alors le SPD fournit une direction d telle que $\nabla f(x)^\top d < 0$ et $e_j(x) + A_j d < 0$ pour tout j , ce qui garantit, pour un pas suffisamment petit, une direction de descente strictement réalisable. Réciproquement, si $z^* = 0$, l'absence de solution avec $z < 0$ signifie qu'il n'existe pas de direction de descente réalisable, ce qui équivaut aux conditions KKT sous contraintes linéaires (Chapitre 1). \square

2.4 SPD de Pironneau–Polak

Pironneau et Polak [77] ont étudié la convergence de plusieurs variantes et ont proposé des SPD quadratiques où l'on ajoute un terme $\frac{1}{2}\|d\|^2$ dans la fonction objectif afin de contrôler implicitement la taille de la direction, sans imposer une normalisation explicite. Cette modification produit un programme quadratique convexe, souvent plus stable numériquement, et particulièrement intéressant lorsque $m \ll n$, car la formulation duale s'exprime en dimension $m + 1$.

En conservant $e_j(x) = A_j x - b_j \leq 0$, on définit l'ensemble des contraintes ε -actives par

$$J_\varepsilon(x) = \{j \in \{1, \dots, m\} \mid e_j(x) \geq -\varepsilon\} = \{j \mid b_j - A_j x \leq \varepsilon\}. \quad (2.17)$$

Le SPD avec contraintes ε -actives peut s'écrire comme suit :

$$\text{SPD_PP1}(x) \begin{cases} \min_{z,d} & z + \frac{1}{2}\|d\|^2, \\ \nabla f(x)^\top d & \leq z, \\ e_j(x) + A_j d & \leq z, \quad \forall j \in J_\varepsilon(x), \\ z \in \mathbb{R}, d \in \mathbb{R}^n. \end{cases} \quad (2.18)$$

La variante avec toutes les contraintes est la suivante :

$$\text{SPD_PP2}(x) \begin{cases} \min_{z,d} & z + \frac{1}{2}\|d\|^2, \\ \nabla f(x)^\top d & \leq z, \\ e_j(x) + A_j d & \leq z, \quad \forall j = 1, \dots, m, \\ z \in \mathbb{R}, d \in \mathbb{R}^n. \end{cases} \quad (2.19)$$

Les auteurs montrent dans [77] que la MDR associée à SPD_PP2(x) est globalement convergente vers un point stationnaire; sous hypothèses de convexité, la convergence

est linéaire, et une borne sur le taux est dérivée dans [76, 77]. Chaney [19, 20] relâche certaines hypothèses et obtient des résultats analogues sous des conditions du second ordre adaptées à un minimum local.

2.5 SPD de Wiest–Polak

Le SPD de Wiest–Polak [91] étend des approches basées sur la programmation quadratique généralisée en introduisant des termes quadratiques, interprétés comme une régularisation ou une approximation de second ordre, dans les inégalités du SPD, tout en conservant une normalisation en boîte. Pour le problème (2.1), on peut l'écrire sous la forme :

$$\text{SPD_WP}(x) \begin{cases} \min_{z,d} z, \\ \nabla f(x)^\top d + \frac{\gamma}{2} \|d\|^2 \leq z, \\ e_j(x) + A_j d + \frac{\gamma}{2} \|d\|^2 \leq z, \quad \forall j = 1, \dots, m, \\ -1 \leq d \leq 1, \\ z \in \mathbb{R}, d \in \mathbb{R}^n, \end{cases} \quad (2.20)$$

où $\gamma > 0$. Cette variante vise à améliorer la stabilité et la vitesse de convergence ; des expériences rapportées, par exemple dans [24], indiquent une supériorité sur certains ensembles de tests.

2.6 SPD de Cawood–Kostreva (normes relaxées)

Cawood et Kostreva [18] proposent une généralisation de Pironneau–Polak en remplaçant la norme euclidienne par une norme elliptique induite par une matrice symétrique définie positive H . Le paramètre $\gamma > 0$ règle l'intensité de la régularisation. Le SPD associé s'écrit :

$$\text{SPD_CK}(x) \begin{cases} \min_{z,d} z + \frac{\gamma}{2} d^\top H d, \\ \nabla f(x)^\top d \leq z, \\ e_j(x) + A_j d \leq z, \quad \forall j = 1, \dots, m, \\ z \in \mathbb{R}, d \in \mathbb{R}^n. \end{cases} \quad (2.21)$$

Ce SPD est un programme quadratique convexe, strictement convexe si H est définie positive, qui se résout efficacement, notamment via sa formulation duale. Cawood et Kostreva [18] établissent une convergence globale sous des hypothèses standards, notamment la compacité de l'ensemble de niveau $\{x \in \mathcal{S} : f(x) \leq f(x^0)\}$ et l'existence d'un intérieur non vide.

Korycki et Kostreva [59, 60] dérivent une convergence linéaire et proposent des mises à jour de type BFGS pour H , vue comme une approximation du Hessien du Lagrangien, afin d'améliorer l'efficacité locale et la robustesse sur de grands problèmes, notamment en optimisation structurale.

2.7 SPD modifié de Chen–Kostreva

Chen et Kostreva [25] introduisent une variante pondérée de SPD_CK, avec des coefficients $\gamma_j > 0$, et $\gamma_0 > 0$ pour l'objectif, permettant d'ajuster la relaxation de manière différenciée entre l'objectif et les contraintes. Le SPD s'écrit :

$$\text{SPD_CKmod}(x) \begin{cases} \min_{z,d} z + \frac{1}{2}d^\top H d, \\ \nabla f(x)^\top d \leq \gamma_0 z, \\ e_j(x) + A_j d \leq \gamma_j z, \quad \forall j = 1, \dots, m, \\ z \in \mathbb{R}, d \in \mathbb{R}^n. \end{cases} \quad (2.22)$$

Lorsque $\gamma_j = \gamma$, constant pour tout j , on retrouve la famille Cawood–Kostreva; lorsque $\gamma_j = 1$ et $H = I$, on se rapproche de Pironneau–Polak. Les poids γ_j permettent un ajustement heuristique ou adaptatif visant à accélérer la convergence, en particulier lorsque certaines contraintes gouvernent la dynamique des itérations.

Chen et Kostreva [25] prouvent la convergence globale sous des hypothèses de compacité d'ensemble de niveau, sans exiger une convexité stricte. Le taux est linéaire, et les tests numériques indiquent qu'un choix approprié des γ_j réduit souvent le nombre d'itérations et d'évaluations de f .

2.8 Méthode de Frank–Wolfe

La méthode de Frank et Wolfe [43], souvent appelée *conditional gradient*, est une méthode de base pour l'optimisation convexe différentiable sur un polyèdre. Elle se distingue des MDR au sens strict par le fait que la direction est obtenue en résolvant un programme linéaire sur \mathcal{S} , par linéarisation de l'objectif, plutôt qu'un SPD construit à partir des contraintes actives.

À l'itération k , on résout le problème linéaire

$$\min_{y \in \mathcal{S}} \nabla f(x^k)^\top y, \quad (2.23)$$

et on note y^k une solution optimale. La direction de déplacement est alors $d^k = y^k - x^k$, et

on choisit un pas $\lambda^k \in [0, 1]$ en résolvant

$$\min_{\lambda \in [0, 1]} f(x^k + \lambda(y^k - x^k)). \quad (2.24)$$

Le nouvel itéré est $x^{k+1} = x^k + \lambda^k(y^k - x^k)$, qui reste faisable par convexité de \mathcal{S} .

Algorithm 2 Algorithme de Frank–Wolfe

- 1: Choisir un point initial $x^0 \in \mathcal{S}$. Poser $k = 0$.
 - 2: **repeat**
 - 3: Résoudre $y^k \in \arg \min_{y \in \mathcal{S}} \nabla f(x^k)^\top y$.
 - 4: Poser $d^k = y^k - x^k$.
 - 5: Calculer le *gap* de Frank–Wolfe : $g_k = \nabla f(x^k)^\top (x^k - y^k)$.
 - 6: Si $g_k \leq \varepsilon$, arrêter.
 - 7: Choisir $\lambda^k \in [0, 1]$ en résolvant $\min_{\lambda \in [0, 1]} f(x^k + \lambda d^k)$, ou via une règle inexacte.
 - 8: Mettre à jour $x^{k+1} = x^k + \lambda^k d^k$, puis $k \leftarrow k + 1$.
 - 9: **until** critère d'arrêt
-

Le critère $g_k \leq \varepsilon$ est naturel : lorsque f est convexe, g_k est un certificat d'optimalité, ou écart dual, et $g_k = 0$ caractérise l'optimalité sur \mathcal{S} .

2.9 Méthode de projection du gradient (Rosen)

La méthode de Rosen [82] prolonge l'idée de descente du gradient aux problèmes contraints en projetant le gradient sur l'espace tangent défini par les contraintes actives. Contrairement aux SPD de type Zoutendijk, qui optimisent explicitement une direction sous contraintes linéarisées, la direction est ici obtenue par projection orthogonale, ce qui simplifie le calcul mais peut produire des directions moins agressives en descente.

Définition 2.1. Une matrice $M \in \mathbb{R}^{nn}$ est une matrice de projection si $M = M^\top$ et $M^2 = M$.

Lemme 2.5. Soit $M \in \mathbb{R}^{nn}$. Alors :

1. si M est une matrice de projection, alors M est semi-définie positive ;
2. M est une matrice de projection si et seulement si $I - M$ est une matrice de projection.

Démonstration. (1) Pour tout $x \in \mathbb{R}^n$, $x^\top M x = x^\top M^\top M x = \|Mx\|^2 \geq 0$, donc $M \geq 0$. (2) Si $M = M^\top$ et $M^2 = M$, alors $(I - M)^\top = I - M$ et $(I - M)^2 = I - 2M + M^2 = I - M$. La réciproque est identique en remplaçant M par $I - M$. \square

Soit $\bar{x} \in \mathcal{S}$. Notons $J(\bar{x})$ l'ensemble des contraintes actives et A_0 la sous-matrice de A composée des lignes A_j pour $j \in J(\bar{x})$. Sous l'hypothèse que A_0 est de rang plein, la

projection orthogonale sur $\ker(A_0)$ est

$$M = I - A_0^\top (A_0 A_0^\top)^{-1} A_0. \quad (2.25)$$

Lemme 2.6. *Soit $\bar{x} \in \mathcal{S}$. Si M est une matrice de projection telle que $M\nabla f(\bar{x}) \neq 0$, alors $d = -M\nabla f(\bar{x})$ est une direction de descente en \bar{x} . De plus, si A_0 est de rang plein et M est donné par (2.25), alors d est tangent aux contraintes actives au premier ordre, donc réalisable pour un pas suffisamment petit.*

Démonstration. Posons $g = \nabla f(\bar{x})$. Comme M est une projection, elle est semi-définie positive (lemme (2.5)), et

$$g^\top d = -g^\top M g = -\|Mg\|^2 < 0$$

si $Mg \neq 0$. Donc d est de descente. Par ailleurs, si $M = I - A_0^\top (A_0 A_0^\top)^{-1} A_0$, alors $A_0 M = 0$, d'où $A_0 d = -A_0 M g = 0$. Ainsi d respecte les contraintes actives au premier ordre, ce qui assure l'admissibilité locale, en tenant compte des contraintes inactives via un pas suffisamment petit. \square

La méthode itérative est la suivante : au point x^k , on construit A_0 à partir de $J(x^k)$, puis M et $d^k = -M\nabla f(x^k)$. Si $d^k = 0$, on évalue les multiplicateurs associés aux contraintes actives ; si tous sont ≥ 0 , on conclut à un point KKT, sinon on relâche une contrainte, à savoir celle associée à un multiplicateur négatif, et on recommence.

Algorithm 3 Processus de projection du gradient (Rosen)

- 1: Choisir un point faisable $x^0 \in \mathcal{S}$. Poser $k = 0$.
- 2: Déterminer $J(x^k)$ et former A_0 .
- 3: **if** $J(x^k) = \emptyset$ **then**
- 4: **if** $\nabla f(x^k) = 0$ **then**
- 5: arrêter.
- 6: **end if**
- 7: poser $d^k = -\nabla f(x^k)$ et aller à l'étape 5.
- 8: **end if**
- 9: Construire $M = I - A_0^\top(A_0A_0^\top)^{-1}A_0$ (si A_0 est de rang plein) et poser $d^k = -M\nabla f(x^k)$.
- 10: **if** $d^k = 0$ **then**
- 11: Calculer $\alpha = (A_0A_0^\top)^{-1}A_0\nabla f(x^k)$.
- 12: **if** $\alpha \geq 0$ **then**
- 13: arrêter; x^k est un point KKT.
- 14: **end if**
- 15: choisir un indice j tel que $\alpha_j < 0$, supprimer la ligne correspondante de A_0 , et retourner à l'étape 4.
- 16: **end if**
- 17: Calculer λ_{\max}^k comme en (2.7). Choisir $\lambda^k \in (0, \lambda_{\max}^k]$ (recherche linéaire) et mettre à jour $x^{k+1} = x^k + \lambda^k d^k$.
- 18: Poser $k \leftarrow k + 1$ et retourner à l'étape 2.

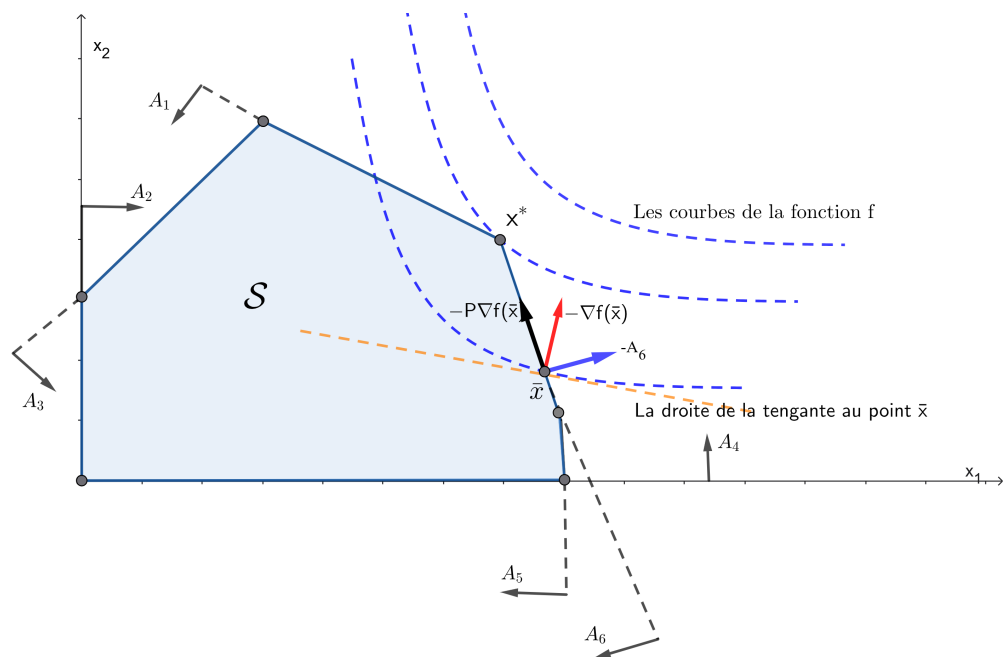


FIGURE 2.2 – Interprétation géométrique de la méthode de projection du gradient de Rosen

Rosen [82] discute essentiellement des propriétés locales sous des hypothèses telles que la convexité stricte de f et l'indépendance linéaire des contraintes actives. Du et Zhang [32] soulignent plusieurs limitations, comme le zigzagging, la lenteur près de solutions non extrêmes et la sensibilité à la recherche linéaire, et proposent des ajustements visant à renforcer la convergence globale.

2.10 Méthode du gradient réduit

La méthode du gradient réduit, développée par Wolfe [92] puis généralisée par Abadie et Carpentier [1], vise des problèmes avec contraintes d'égalité et non-négativité, typiquement

$$\min f(x) \quad \text{s.c.} \quad Ax = b, \quad x \geq 0. \quad (2.26)$$

Le problème (2.1) peut être transformé sous cette forme en introduisant des variables d'écart et un changement de variables adéquat ; ce point est classique en programmation non linéaire sous contraintes linéaires.

Soit x une solution réalisable. On décompose $A = [B \ N]$, où $B \in \mathbb{R}^{mm}$ est inversible (base) et $N \in \mathbb{R}^{m(n-m)}$. De même, on écrit $x = (x_B, x_N)$ avec $x_B > 0$. En imposant $Ax = b$, on obtient

$$Ad = 0 \iff Bd_B + Nd_N = 0 \iff d_B = -B^{-1}Nd_N. \quad (2.27)$$

En notant $\nabla f(x) = (\nabla_B f(x), \nabla_N f(x))$, le *gradient réduit* est défini par

$$\begin{aligned} r^\top &= (r_B^\top, r_N^\top), \\ r^\top &= \nabla f(x)^\top - \nabla_B f(x)^\top B^{-1}A, \\ r^\top &= (0, \nabla_N f(x)^\top - \nabla_B f(x)^\top B^{-1}N). \end{aligned} \quad (2.28)$$

La direction d_N est choisie de sorte que $r_N^\top d_N < 0$ et $d_j \geq 0$ si $x_j = 0$. Une règle classique, due à Wolfe, consiste, pour chaque composante hors base j , à définir

$$d_j = \begin{cases} -r_j, & \text{si } r_j \leq 0, \\ -x_j r_j, & \text{si } r_j > 0, \end{cases} \quad (2.29)$$

ce qui garantit $d_j \geq 0$ lorsque $x_j = 0$ et évite des pas trop petits lorsque $x_j > 0$ est petit mais $r_j > 0$. On complète ensuite d_B via (2.27).

Lemme 2.7. *Considérons (2.26). Soit x réalisable avec décomposition (B, N) telle que B soit inversible et $x_B > 0$. Soit r le gradient réduit défini par (2.28). On construit d_N par (2.29) et*

$d_B = -B^{-1}Nd_N$. Si $d \neq 0$, alors d est une direction de descente réalisable. De plus, $d = 0$ si et seulement si x satisfait les conditions KKT.

Démonstration. Par construction, $Ad = Bd_B + Nd_N = 0$, donc la direction respecte l'égalité. Si $x_j = 0$, alors j est hors base et (2.29) donne $d_j \geq 0$, ce qui préserve la non-négativité localement. Enfin,

$$\nabla f(x)^\top d = \nabla_B f(x)^\top d_B + \nabla_N f(x)^\top d_N = \left(\nabla_N f(x)^\top - \nabla_B f(x)^\top B^{-1}N \right) d_N = r_N^\top d_N.$$

Or, chaque terme $r_j d_j \leq 0$: si $r_j \leq 0$, alors $r_j d_j = -r_j^2 \leq 0$; si $r_j > 0$, alors $r_j d_j = -x_j r_j^2 \leq 0$. L'inégalité est stricte dès que $d_N \neq 0$, donc dès que $d \neq 0$. On obtient ainsi $\nabla f(x)^\top d < 0$, donc d est une direction de descente. Enfin, l'équivalence $d = 0 \iff$ KKT est obtenue en remarquant que $d = 0$ équivaut à $r_N \geq 0$ et $r_N^\top x_N = 0$, ce qui correspond aux conditions de complémentarité sur les variables hors base, avec multiplicateurs déduits de $v^\top = -\nabla_B f(x)^\top B^{-1}$. \square

Algorithm 4 Algorithme du gradient réduit

- 1: Choisir un point initial x^0 satisfaisant $Ax^0 = b$ et $x^0 \geq 0$. Poser $k = 0$.
- 2: **repeat**
- 3: Choisir une base B inversible (ensemble d'indices de base) et poser $A = [B \ N]$, $x = (x_B, x_N)$.
- 4: Calculer le gradient réduit r via (2.28).
- 5: Construire d_N par (2.29) et $d_B = -B^{-1}Nd_N$. Poser $d = (d_B, d_N)$.
- 6: **if** $d = 0$ **then**
- 7: arrêter ; x^k satisfait KKT.
- 8: **end if**
- 9: Déterminer λ_{\max}^k afin de préserver $x^k + \lambda d \geq 0$:

$$\lambda_{\max}^k = \begin{cases} \min_{1 \leq j \leq n} \left\{ \frac{-x_{k,j}}{d_j} : d_j < 0 \right\}, & \text{s'il existe } d_j < 0, \\ +\infty, & \text{sinon.} \end{cases} \quad (2.30)$$

- 10: Choisir $\lambda^k \in (0, \lambda_{\max}^k]$ (recherche linéaire) et mettre à jour $x^{k+1} = x^k + \lambda^k d$, $k \leftarrow k + 1$.
 - 11: **until** critère d'arrêt
-

Zangwill [98] propose la méthode Convex-Simplex, variante où une seule variable hors base est modifiée par itération, selon un mécanisme rappelant le simplexe, tandis que les variables de base sont ajustées pour maintenir $Ax = b$. Cette simplification peut

accroître la robustesse sur certaines classes de fonctions convexes, au prix d'une progression parfois plus lente. Huard [53] analyse en détail la convergence du gradient réduit et propose une variante générant une suite infinie de solutions réalisables dont les points d'accumulation sont optimaux pour le problème (2.1), après transformation adéquate.

3

Optimisation multi-objectifs

Introduction

Nous considérons dans ce chapitre des problèmes d'optimisation multi-objectifs avec contraintes linéaires (LC-MOO) de la forme

$$\text{(LC - MOO)} \quad \begin{cases} \min_{x \in \mathcal{S}} F(x) = (f_1(x), f_2(x), \dots, f_p(x))^T, \\ p \geq 2, \end{cases} \quad (3.1)$$

où $F : \mathbb{R}^n \rightarrow \mathbb{R}^p$ est le vecteur des fonctions objectif. Dans toute la suite, on suppose que f_i est continûment différentiable sur \mathcal{S} pour tout $i = 1, \dots, p$, et l'on note $\nabla f_i(x)$ son gradient en x . Rappelons que l'ensemble réalisable est le polyèdre $\mathcal{S} = \{x \in \mathbb{R}^n : Ax \leq b\}$, supposé non vide, fermé et borné. L'image de \mathcal{S} dans l'espace des objectifs est notée $\mathcal{Y} = F(\mathcal{S}) \subset \mathbb{R}^p$. Le problème multi-objectifs se distingue du cas mono-objectif par l'absence, en général, d'un optimum unique : les objectifs peuvent être conflictuels, et l'on recherche alors un ensemble de solutions de compromis décrites par la notion d'optimalité au sens de *Pareto*.

Ce chapitre rappelle les concepts fondamentaux de la discipline, notamment la dominance, l'efficacité, la faible efficacité, le *front de Pareto*, ainsi que les points idéal, utopique et Nadir. Il présente ensuite des familles classiques de méthodes de résolution : (i) des

méthodes de scalarisation, comme la somme pondérée, les ε -contraintes et les normes pondérées, qui convertissent le problème LC – MOO en problèmes mono-objectifs paramétrés ; (ii) des méthodes non paramétriques fondées sur des directions de descente ; (iii) des approches stochastiques de type métaheuristique et des algorithmes évolutionnaires, en particulier NSGA-II. Les notions et méthodes présentées s'appuient notamment sur [70, 35, 67].

3.1 Définitions et notions de base

Dans un problème multi-objectifs, comparer deux solutions revient à comparer leurs vecteurs d'objectifs. On utilise pour cela l'ordre partiel produit sur \mathbb{R}^p . Pour $u, v \in \mathbb{R}^p$, on définit

$$u \leq v \iff u_i \leq v_i \quad \forall i = 1, \dots, p, \quad u < v \iff u_i < v_i \quad \forall i = 1, \dots, p.$$

On dira que u domine v au sens de *Pareto*, et l'on notera $u < v$, si

$$u \leq v \text{ et } u \neq v, \quad \text{c.-à-d.} \quad u_i \leq v_i \quad \forall i, \text{ et } \exists k : u_k < v_k.$$

Cet ordre partiel formalise les compromis propres aux objectifs multiples et sert de base à la notion d'optimalité au sens de *Pareto*¹.

Définition 3.1 (Efficacité). Soit $\bar{x} \in \mathcal{S}$. Le point \bar{x} est dit efficace ou optimal au sens de *Pareto* pour le problème LC – MOO s'il n'existe aucun $x \in \mathcal{S}$ tel que $f_i(x) \leq f_i(\bar{x})$ pour tout $i = 1, \dots, p$, et $f_k(x) < f_k(\bar{x})$ pour un certain $k \in \{1, \dots, p\}$.

Le point correspondant $\bar{y} = F(\bar{x}) \in \mathcal{Y}$ dans l'espace des objectifs est appelé *point non dominé*.

Définition 3.2. \bar{x} est dit faiblement efficace ou faiblement *Pareto-optimal* s'il n'existe aucun $x \in \mathcal{S}$ tel que $f_i(x) < f_i(\bar{x})$ pour tout $i = 1, \dots, p$.

Dans ce cas, le point correspondant $\bar{y} = F(\bar{x}) \in \mathcal{Y}$ dans l'espace des objectifs est appelé *point faiblement non dominé*.

Définition 3.3 (Efficacité locale). Soit $\bar{x} \in \mathcal{S}$. Le point \bar{x} est dit localement efficace ou localement optimal au sens de *Pareto* pour le problème LC – MOO s'il existe $\delta > 0$ tel que \bar{x} soit *Pareto-optimal* dans $\mathcal{S} \cap B(\bar{x}, \delta)$, où $B(\bar{x}, \delta)$ désigne la boule ouverte centrée en \bar{x} de rayon δ .

1. La notion d'efficacité au sens de *Pareto* est attribuée à Vilfredo *Pareto* (1906) dans *Manuale di economia politica*.

Le point correspondant $\bar{y} = F(\bar{x}) \in \mathcal{Y}$ dans l'espace des objectifs est appelé point localement non dominé.

Par ailleurs, un point est faiblement efficace s'il n'existe aucun autre point qui améliore simultanément tous les objectifs. En revanche, un point est efficace s'il n'existe aucun autre point qui améliore au moins un objectif sans en détériorer un autre.

Par souci de brièveté, et sauf indication contraire, le terme point efficace sera employé dans la suite de cette thèse pour désigner une efficacité globale. Nous ne distinguerons l'efficacité locale de l'efficacité globale que lors de l'analyse de fonctions objectif non convexes. En effet, sous l'hypothèse de convexité, toute solution localement efficace est également globalement efficace. De plus, toute solution efficace est localement efficace. La réciproque n'est valide que si $F(x)$ est \mathbb{R}^p -convexe.

Dans la suite, nous désignerons par \mathcal{X}_E l'ensemble des solutions efficaces, ou *Pareto-optimales*, et par \mathcal{X}_{WE} l'ensemble des solutions faiblement efficaces. De même, dans l'espace des objectifs, \mathcal{Y}_N représente l'ensemble des points non dominés et \mathcal{Y}_{WN} l'ensemble des points faiblement non dominés. D'après les définitions précédentes, il est possible d'observer que :

$$\mathcal{X}_E \subseteq \mathcal{X}_{WE}, \quad \mathcal{Y}_N \subseteq \mathcal{Y}_{WN}.$$

Cela signifie que les points efficaces sont faiblement efficaces, mais que l'inverse n'est pas vrai.

L'ensemble de tous les points faiblement non dominés, \mathcal{Y}_{WN} , constitue ce qu'on appelle le *front de Pareto*. Ce front représente la frontière de l'espace des objectifs où aucune amélioration simultanée de tous les critères n'est possible. Il regroupe ainsi les solutions de compromis qui ne peuvent être améliorées globalement sans dégrader au moins un objectif.

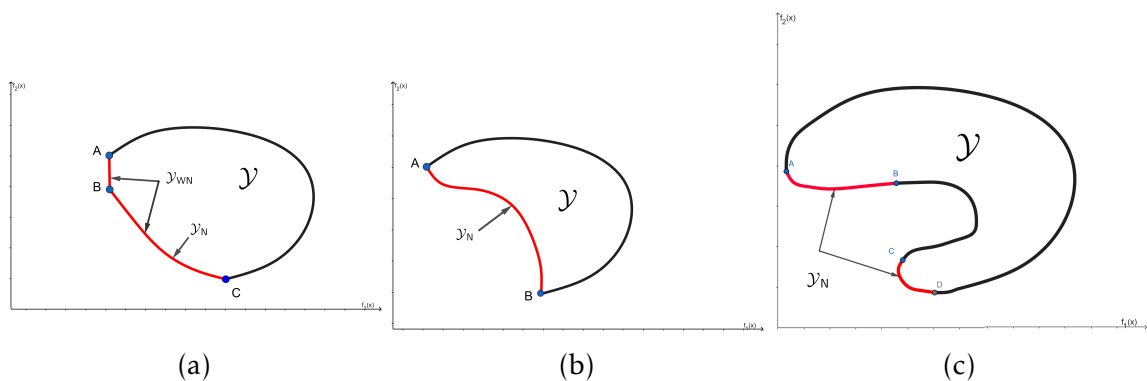


FIGURE 3.1 – Exemples de formes possibles d'un front de Pareto.

Les figures (3.1a), (3.1b) et (3.1c) montrent trois cas distincts de fronts de Pareto pour

un problème bi-objectif. La première illustre un front contenant des points faiblement non dominés sur le segment $[A, B]$ et des points non dominés sur la partie $[B, C]$. La seconde figure (3.1b) présente un front avec une partie concave. La troisième figure (3.1c) illustre un front discontinu, composé de segments isolés.

Les différentes méthodes connues dans la littérature pour résoudre le problème LC – MOO consistent généralement à identifier le *front de Pareto* complet ou une partie de celui-ci, ce qui équivaut souvent à une approximation de ce front.

Puisque l'ensemble de décision \mathcal{S} est borné et fermé, les fonctions objectif f_i sont bornées sur \mathcal{S} , et admettent ainsi des bornes inférieures et supérieures finies du fait de leur continuité. Par conséquent, le *front de Pareto* possède une borne inférieure appelée *point idéal* et une borne supérieure appelée *point Nadir*.

Définition 3.4. Le *point idéal*, noté z^* , est défini comme le vecteur des minima indépendants de chaque objectif : $z_i^* = \min_{x \in \mathcal{S}} f_i(x)$ pour $i = 1, \dots, p$.

Ce point incarne une solution théoriquement parfaite, souvent inatteignable en raison des conflits entre objectifs, mais il sert de référence pour normaliser les fonctions, mesurer les écarts et faciliter les agrégations scalaires [70], ainsi que le calcul de distances ou de compromis dans les méthodes dédiées [100].

Définition 3.5. Le *point utopique*, noté z^{utop} , est défini comme le vecteur obtenu en soustrayant un petit scalaire $\epsilon > 0$ au point idéal pour chaque composante : $z_i^{utop} = z_i^* - \epsilon$ pour $i = 1, \dots, p$.

Ce point est introduit dans les méthodes des normes pondérées, particulièrement dans les variantes augmentées de la norme de Tchebycheff, comme celles proposées par Steuer et Choo [86], afin d'éviter des problèmes liés aux solutions faiblement efficaces.

Définition 3.6. Le *point Nadir*, noté z^{nad} , approxime les maximums des objectifs sur le front de Pareto, offrant ainsi une estimation de sa borne supérieure. Il est souvent calculé en maximisant chaque objectif sur l'ensemble des solutions Pareto-optimales.

Bien que sa détermination exacte soit complexe pour les problèmes non convexes, le *point Nadir* représente les pires valeurs des objectifs sur le front et sert de référence pour estimer l'*hypervolume*, un indicateur mesurant le volume dominé par l'approximation du front par rapport à un point fixe [55]. Il facilite ainsi l'analyse, la normalisation et l'évaluation des compromis [70]. La figure (3.2) illustre le point idéal z^* , qui correspond à la meilleure solution théorique, et le point Nadir z^{nad} , qui représente les pires valeurs sur le front. Ces deux points sont souvent inatteignables.

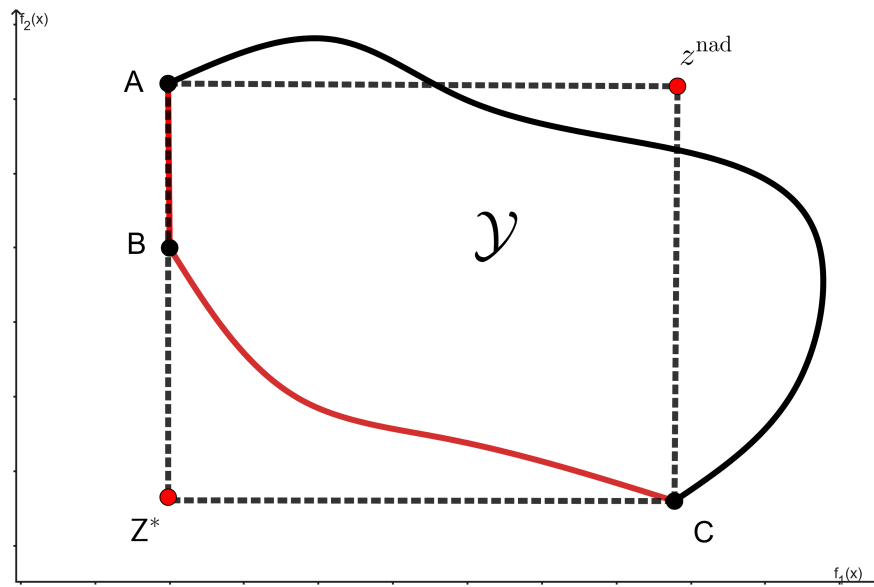


FIGURE 3.2 – Front de Pareto et ses bornes

3.2 Caractérisation des solutions efficaces

Cette section présente des tests d'efficacité reposant sur la résolution d'un sous-problème scalaire. Ces tests sont utiles pour vérifier si un point donné est efficace, produire une solution efficace initiale ou certifier l'absence de solutions efficaces dans certains cas. Le principe remonte à Ecker et Kouada [34] pour le cas linéaire, puis a été étendu notamment par Wendell et Lee [90].

Soit $\bar{x} \in \mathcal{S}$ fixé. Considérons le problème auxiliaire

$$\begin{cases} \min_{x \in \mathcal{S}} \sum_{i=1}^p f_i(x), \\ \text{s.c. } f_i(x) \leq f_i(\bar{x}), \quad i = 1, \dots, p. \end{cases} \quad (3.2)$$

Théorème 3.1. *Le point $\bar{x} \in \mathcal{S}$ est efficace pour le problème LC – MOO si et seulement si \bar{x} est solution optimale du problème (3.2).*

Démonstration. Si \bar{x} n'est pas efficace, il existe $x \in \mathcal{S}$ tel que $F(x) < F(\bar{x})$, donc $f_i(x) \leq f_i(\bar{x})$ pour tout i , avec au moins une inégalité stricte. Alors x est réalisable pour (3.2) et $\sum_{i=1}^p f_i(x) < \sum_{i=1}^p f_i(\bar{x})$, ce qui exclut l'optimalité de \bar{x} .

Réciproquement, si \bar{x} est optimal pour le problème (3.2) et s'il existait $x \in \mathcal{S}$ tel que $F(x) < F(\bar{x})$, alors x serait réalisable pour le problème (3.2) et donnerait une valeur strictement plus petite, ce qui constitue une contradiction. \square

Ce type de test a été reformulé et généralisé par Benson [12], notamment dans des cadres convexes.

Théorème 3.2 (Test de Benson). Soit $\bar{x} \in \mathcal{S}$. Considérons le sous-problème suivant :

$$\begin{cases} \max_{x, \epsilon} & \sum_{i=1}^p \epsilon_i \\ \text{s.t.} & f_i(x) + \epsilon_i = f_i(\bar{x}), \quad \forall i = 1, \dots, p \\ & \epsilon_i \geq 0, \quad x \in \mathcal{S} \end{cases} \quad (3.3)$$

Les résultats suivants sont valides :

1. $\bar{x} \in \mathcal{X}_E$ si et seulement si la valeur optimale de (3.3) est égale à zéro.
2. Si la valeur optimale de (3.3) est finie et non nulle, obtenue au point \bar{x} , alors $\bar{x} \in \mathcal{X}_E$.
3. Si $F(x)$ est \mathbb{R}^p -convexe et que le sous-problème (3.3) est non borné, alors $\mathcal{X}_E = \emptyset$.

Démonstration. Nous démontrerons chaque partie du théorème séparément.

1. Supposons que la valeur optimale de (3.3) soit égale à zéro. Par l'absurde, supposons que $\bar{x} \notin \mathcal{X}_E$. Alors, il existe un $x \in \mathcal{S}$ tel que $f_i(x) \leq f_i(\bar{x})$ pour tout $i = 1, \dots, p$, avec au moins une inégalité stricte, disons pour k , $f_k(x) < f_k(\bar{x})$. Définissons $\epsilon_i = f_i(\bar{x}) - f_i(x) \geq 0$ pour tout i , avec $\epsilon_k > 0$. Cela implique que (x, ϵ) est réalisable pour (3.3) et que $\sum_{i=1}^p \epsilon_i > 0$, ce qui contredit la valeur optimale nulle. Donc, $\bar{x} \in \mathcal{X}_E$.
Réciproquement, supposons que $\bar{x} \in \mathcal{X}_E$. Aucun $x \in \mathcal{S}$ ne peut satisfaire $f_i(x) \leq f_i(\bar{x})$ pour tout i , avec au moins une inégalité stricte. La valeur optimale de (3.3) est donc nulle, car tout $\epsilon > 0$ violerait l'efficacité.
2. Supposons que la valeur optimale soit finie et positive, obtenue à (x^*, ϵ^*) , avec $\sum_{i=1}^p \epsilon_i^* > 0$. Alors, $f_i(x^*) = f_i(\bar{x}) - \epsilon_i^* \leq f_i(\bar{x})$ pour tout i , et au moins une $\epsilon_k^* > 0$, donc $f_k(x^*) < f_k(\bar{x})$. Cela signifie que x^* domine \bar{x} . Par l'absurde, supposons que $x^* \notin \mathcal{X}_E$. Il existerait un $x' \in \mathcal{S}$ dominant x^* , ce qui impliquerait une valeur optimale supérieure, contredisant l'optimalité. Donc, $x^* \in \mathcal{X}_E$.
3. Supposons que $F(x)$ soit \mathbb{R}^p -convexe et que (3.3) soit non borné, c'est-à-dire que sa valeur optimale soit $+\infty$. Cela signifie qu'il existe une direction où les ϵ_i peuvent croître indéfiniment tout en restant réalisables, impliquant l'absence de bornes supérieures sur les améliorations potentielles. Dans un problème convexe, cela équivaut à l'absence de points *Pareto-optimaux*, car tout point peut être amélioré indéfiniment dans au moins une direction, rendant $\mathcal{X}_E = \emptyset$.

□

3.3 Classification des méthodes de résolution

Selon Miettinen [70], les méthodes de résolution des problèmes d'optimisation multi-objectifs, incluant le problème LC – MOO, peuvent être classées selon divers critères. L'un de ces critères est l'implication du décideur, qui dépend du moment où ses préférences sont exprimées : a priori, de manière interactive ou a posteriori. Ces trois approches permettent respectivement de fixer les priorités en amont, d'ajuster les choix en cours d'exécution ou de sélectionner une solution parmi les solutions efficaces générées.

Dans cette thèse, où la résolution du problème LC – MOO vise l'identification complète ou partielle du front de Pareto, l'accent est mis sur les approches a posteriori. Dans ce cas, le décideur n'intervient qu'en fin de processus, une fois l'ensemble des solutions efficaces disponible pour l'évaluation.

Une autre classification repose sur la nature de la technique employée, opposant les méthodes déterministes exactes aux approches stochastiques approximatives, qui favorisent l'exploration globale des espaces multimodaux. Cette taxonomie regroupe trois catégories principales :

1. Méthodes scalaires, ou paramétriques, qui reformulent le problème LC – MOO en un sous-problème mono-objectif paramétré, dont la solution optimale correspond à un point efficace du problème original. En variant les paramètres de manière systématique, ces méthodes produisent des points discrets sur le front de Pareto, facilitant son approximation partielle ou totale au moyen d'itérations successives, tout en intégrant des considérations de convexité pour une meilleure convergence. Exemples : la pondération des objectifs, la méthode des ε -contraintes et les normes pondérées [70, 36, 35].
2. Méthodes évolutionnaires et métaheuristiques, qui sont des algorithmes stochastiques adaptés aux paysages d'optimisation complexes et non convexes. Elles exploitent des principes bio-inspirés pour effectuer une recherche parallèle efficace. Opérant sur des populations de candidats, elles appliquent des opérateurs tels que la sélection, le croisement et la mutation pour approximer le front de Pareto, en intégrant parfois des mécanismes d'hybridation avec des optimiseurs locaux afin d'accélérer la convergence. Parmi les algorithmes évolutionnaires notables, citons VEGA (*Vector Evaluated Genetic Algorithm*), qui divise la population en sous-populations optimisées par objectif, MOGA (*Multi-Objective Genetic Algorithm*), qui met l'accent sur les solutions non dominées via un partage de fitness, NSGA (*Non-dominated Sorting Genetic Algorithm*) et sa version améliorée NSGA-II, qui utilise un tri non dominé et une distance de crowding pour obtenir une répartition uniforme des solutions *Pareto-optimales*, ainsi que SPEA2 (*Strength Pareto*

Evolutionary Algorithm 2), qui emploie une archive externe pour maintenir les élites. Les métaheuristiques incluent le recuit simulé, la recherche tabou, MOPSO (*Multi-Objective Particle Swarm Optimization*), qui étend l'optimisation par essaim particulière avec une archive pour guider la recherche vers un front diversifié, et l'optimisation par colonies de fourmis. Ces approches renforcent la diversité et la robustesse face aux contraintes et aux conflits d'objectifs, malgré un coût computationnel élevé pour les grandes instances; leur efficacité en ingénierie est bien établie [30, 26, 102]. Dans ce qui suit, nous détaillerons la méthode NSGA-II, car elle sera comparée à la méthode de direction réalisable que nous développerons par la suite.

3. Méthodes non paramétriques : approches numériques sans paramètres explicites, qui favorisent une flexibilité accrue dans les itérations directionnelles. Introduites par *Fliege et Svaiter* [40] à travers l'extension de la descente la plus raide au cadre multi-objectifs, elles ont inspiré des variantes intégrant des stratégies de globalisation pour éviter les minima locaux. Parmi ces méthodes, on peut citer les directions réalisables de *Zoutendijk* [71], le gradient réduit [38, 37], les régions de confiance [14], le gradient conditionnel [5], le gradient projeté [50] et la programmation quadratique successive [41].

3.4 Méthodes scalaires

3.4.1 Méthode de la somme pondérée des fonctions objectif

Introduite par Gass et Saaty [46] et étayée théoriquement par Zadeh [97], la méthode de la somme pondérée consiste à ramener le problème LC – MOO à un problème de minimisation scalaire. À chaque fonction objectif f_i , on associe un poids $w_i > 0$, puis on minimise la combinaison linéaire pondérée des objectifs. Sous certaines hypothèses, une solution optimale du problème ainsi paramétré correspond à une solution efficace du problème LC – MOO. Les poids sont en général normalisés de sorte que $\sum_{i=1}^p w_i = 1$. Le problème de pondération s'écrit alors :

$$\min_{x \in S} \sum_{i=1}^p w_i f_i(x). \quad (3.4)$$

Le théorème ci-dessous précise le lien entre optimalité pour (3.4) et efficacité.

- Théorème 3.3.**
1. Toute solution optimale de (3.4) est faiblement efficace.
 2. Si $w_i > 0$ pour tout $i = 1, \dots, p$, toute solution optimale de (3.4) est efficace.

3. Si (3.4) admet une solution optimale unique, alors cette solution est efficace.

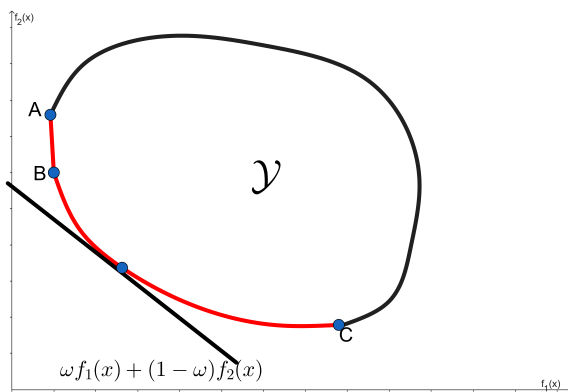


FIGURE 3.3 – Cas d'ensemble \mathcal{Y} convexe

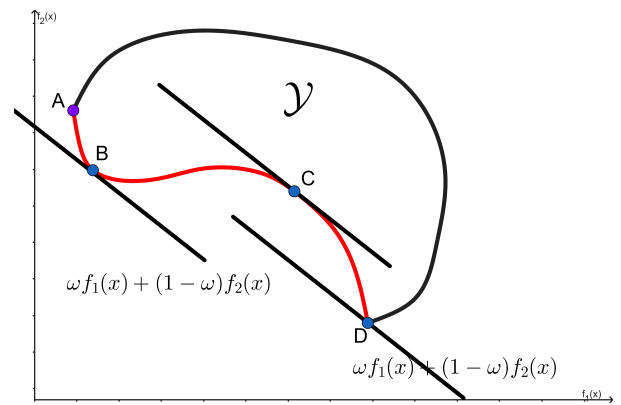


FIGURE 3.4 – Cas d'ensemble \mathcal{Y} non convexe

Les figures (3.3) et (3.4) illustrent deux ensembles réalisables \mathcal{Y} dans l'espace des objectifs d'un problème bi-objectif, ainsi que leurs *fronts de Pareto* en rouge. Pour un poids $\omega \in [0, 1]$, la minimisation de la combinaison linéaire $\omega f_1(x) + (1 - \omega)f_2(x)$ revient à déplacer une droite de niveau $\omega f_1 + (1 - \omega)f_2 = \text{cste}$, de pente $-\frac{\omega}{1-\omega}$, jusqu'à son premier point de contact avec \mathcal{Y} . Le point obtenu est alors une solution optimale du problème paramétré et constitue un point non dominé. Dans la figure (3.3), le *front de Pareto* se décompose en deux parties : un segment $[A, B]$ et une portion courbe $[B, C]$. Lorsque ω varie, la droite de niveau balaye la partie $[B, C]$, ce qui met en évidence un avantage important de la méthode : en ajustant les poids, on peut générer tous les points non dominés et reconstruire une partie du front. En revanche, la présence du segment $[A, B]$ souligne une première limite : pour une même valeur de ω , le problème pondéré peut admettre plusieurs solutions optimales, ce qui correspond géométriquement à un contact le long d'un segment ; les points de $[A, B]$ sont alors seulement faiblement non dominés, tandis que B est non dominé.

La figure (3.4) met en évidence la limite principale de la méthode de la somme pondérée : lorsque l'ensemble réalisable \mathcal{Y} n'est pas convexe, la droite de niveau ne peut atteindre que les points de la portion $[A, y_1]$ ainsi que l'extrémité B . Par conséquent, toute la portion comprise entre y_1 et B , c'est-à-dire la partie concave du front, ne peut pas être obtenue, quelle que soit la valeur de ω . Le point y_2 illustre précisément ce type de point non dominé mais non atteignable par la droite de niveau.

En résumé, la méthode de la somme pondérée des fonctions objectif est simple et facile à mettre en œuvre. Elle est efficace pour générer des solutions non dominées lorsque l'ensemble \mathcal{Y} est convexe. Toutefois, elle peut produire des solutions multiples ou seulement faiblement non dominées pour certains poids et, surtout, elle ne permet pas de retrouver

l'ensemble du *front de Pareto* lorsque \mathcal{Y} est non convexe.

3.4.2 Méthode des ϵ -contraintes

Introduite par Haimes *et al.* [51], la méthode des ϵ -contraintes ramène le problème LC – MOO à une suite de problèmes d'optimisation mono-objectif. Le principe consiste à choisir l'une des fonctions objectif, notée f_i , comme objectif principal à minimiser, tandis que les autres fonctions objectif f_l , avec $l \neq i$, sont reformulées sous forme de contraintes en leur imposant des bornes supérieures ϵ_l . On obtient alors le problème paramétré suivant :

$$\begin{cases} \min_{x \in \mathcal{S}} & f_i(x), \\ \text{s.c.} & f_l(x) \leq \epsilon_l, \quad \forall l = 1, \dots, p, l \neq i. \end{cases} \quad (3.5)$$

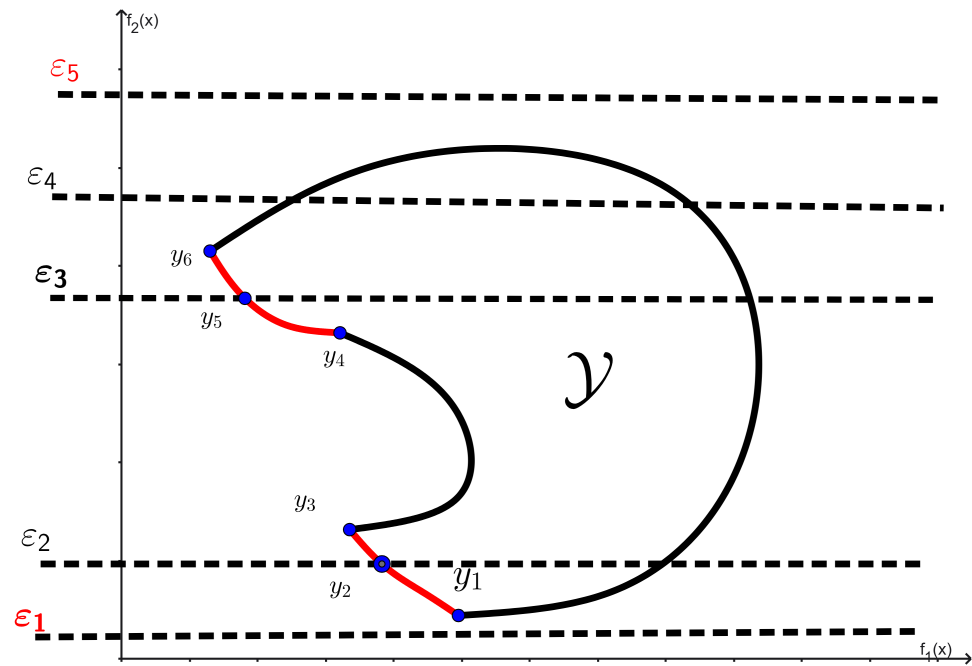
En faisant varier les paramètres ϵ_l , on génère différentes solutions du problème (3.5). Sous des hypothèses appropriées, ces solutions correspondent à des solutions faiblement efficaces ou efficaces du problème LC – MOO.

Théorème 3.4. 1. Toute solution optimale du problème des ϵ -contraintes (3.5) est faiblement efficace pour le problème LC – MOO.

2. Un point $x^* \in \mathcal{S}$ est efficace pour le problème LC – MOO si et seulement si, pour tout $i = 1, \dots, p$, x^* est optimal pour (3.5) lorsque les bornes sont fixées par $\epsilon_l = f_l(x^*)$ pour tout $l \neq i$.
3. Un point $x^* \in \mathcal{S}$ est efficace pour le problème LC – MOO s'il est l'unique solution optimale de (3.5) pour un certain i et pour des bornes telles que $\epsilon_l = f_l(x^*)$ pour tout $l \neq i$.
4. Pour tout vecteur de bornes supérieures $\epsilon = (\epsilon_1, \dots, \epsilon_{i-1}, \epsilon_{i+1}, \dots, \epsilon_p)^\top$, si x^* est l'unique solution optimale de (3.5) pour un certain i , alors x^* est efficace pour le problème LC – MOO.

La figure (3.5) représente l'espace des objectifs d'un problème LC – MOO bi-objectif, où le *front de Pareto* discontinu est indiqué en rouge. Dans cette figure, différentes bornes supérieures sont imposées à la fonction objectif f_2 , tandis que la fonction f_1 est minimisée. La borne ϵ_1 est trop restrictive, ce qui rend l'ensemble réalisable vide. À l'inverse, la borne ϵ_5 n'impose pratiquement aucune contrainte sur l'ensemble réalisable \mathcal{Y} ; lorsqu'elle est utilisée, la solution optimale obtenue est le point y_6 , qui est efficace d'après le Théorème (3.4). De même, pour la borne ϵ_4 , on obtient également le point y_6 . Enfin, le point y_2 correspond à la solution optimale associée à la borne ϵ_2 , et constitue aussi une solution efficace.

Malgré sa capacité à générer des points non dominés lorsque le *front de Pareto* est discontinu et lorsque l'ensemble réalisable \mathcal{Y} est non convexe, cas où la méthode de la somme

FIGURE 3.5 – Fonctionnement de la méthode ε -contraainte

pondérée peut échouer, la méthode des ε -contraintes présente plusieurs limites. Le choix des bornes ε_l est délicat : des valeurs trop restrictives rendent le problème (3.5) irréalisable, tandis que des valeurs trop larges influencent peu la solution et peuvent conduire, pour plusieurs bornes, au même point non dominé. De plus, l'approximation du front exige la résolution d'une suite de problèmes (3.5) pour de nombreuses valeurs de ε_l , ce qui peut être coûteux, notamment lorsque p est élevé. Enfin, la qualité de l'échantillonnage dépend fortement de la stratégie de variation des ε_l . La méthode peut aussi produire des solutions seulement faiblement non dominées lorsque l'optimum n'est pas unique, tout en restant sensible aux tolérances numériques lorsque les contraintes $f_l(x) \leq \varepsilon_l$ sont presque actives [70].

3.4.3 Méthode des normes pondérées

Les méthodes des métriques pondérées (*weighted metrics*) s'inscrivent historiquement dans la famille des approches dites du critère global ou du point de référence. Elles consistent à rechercher une solution efficace en minimisant une distance, exprimée par une norme de type L_q ou L_∞ , entre le vecteur des fonctions objectif $F(x)$ et un vecteur de référence, généralement choisi comme le point idéal z^* . Cette idée est étroitement liée au courant du *compromise programming*, développé par Zeleny [99].

La construction de la méthode se fait comme suit : étant donné un vecteur de poids $\omega \in \mathbb{R}^p$ tel que $\omega_i \geq 0$ pour tout $i = 1, \dots, p$ et que $\sum_{i=1}^p \omega_i = 1$, soit $z^* = (z_1^*, \dots, z_p^*)$ un vecteur de référence, souvent le point idéal.

Pour $1 \leq q < +\infty$, le problème L_q pondéré s'écrit :

$$\min_{x \in \mathcal{S}} \left(\sum_{i=1}^p \omega_i |f_i(x) - z_i^*|^q \right)^{1/q}. \quad (3.6)$$

Remarquons que la méthode de la somme pondérée est un cas particulier du problème (3.6) pour $q = 1$: le problème (3.6) se réduit alors à la minimisation de la somme pondérée des fonctions objectif.

En faisant varier le vecteur de poids $\omega = (\omega_1, \dots, \omega_p)$, on obtient différentes solutions faiblement efficaces ou efficaces. Le théorème suivant établit la relation entre la solution optimale du problème pondéré (3.6) et son efficacité.

Théorème 3.5. *La solution optimale du problème pondéré (3.6) pour $1 \leq q < \infty$ est efficace pour le problème LC – MOO, sous l'une des conditions suivantes :*

1. *La solution est unique, ou*
2. *Tous les coefficients de pondération sont strictement positifs.*

Introduite initialement par *Bowman* [15], pour la métrique L_∞ , la méthode pondérée de Tchebycheff est donnée par :

$$\min_{x \in \mathcal{S}} \max_{i=1, \dots, p} \left\{ \omega_i |f_i(x) - z_i^*| \right\}, \quad (3.7)$$

Ce problème est non différentiable. *Bowman* [15] a proposé une formulation équivalente en introduisant une variable auxiliaire $\alpha \in \mathbb{R}$:

$$\begin{cases} \min_{x \in \mathcal{S}, \alpha \in \mathbb{R}} & \alpha, \\ \text{s.c.} & \alpha \geq \omega_i (f_i(x) - z_i^*), \quad i = 1, \dots, p. \end{cases} \quad (3.8)$$

Le théorème suivant établit la relation entre la solution optimale de (3.8) et son efficacité.

Théorème 3.6. *1. La solution optimale du problème de Tchebycheff pondéré (3.8) est faiblement efficace si tous les coefficients de pondération ω_i sont positifs.*

2. *Si le problème pondéré (3.8) possède une solution optimale unique, celle-ci est efficace pour le problème LC – MOO.*

Démonstration. 1. Soit (x^*, α^*) une solution optimale du problème pondéré (3.8). Supposons, par l'absurde, que x^* ne soit pas faiblement efficace pour le problème LC – MOO. Il existerait alors un point $x \in \mathcal{S}$ tel que $f_i(x) < f_i(x^*)$ pour tout $i = 1, \dots, p$. Puisque $\omega_i > 0$ pour tout i , et que z^* est fixe, on obtient :

$$\omega_i (f_i(x) - z_i^*) < \omega_i (f_i(x^*) - z_i^*) \leq \alpha^*, \quad \forall i = 1, \dots, p.$$

Par conséquent,

$$\max_{i=1,\dots,p} \omega_i(f_i(x) - z_i^*) < \alpha^*.$$

Cela implique que (x, α) , avec $\alpha = \max_i \omega_i(f_i(x) - z_i^*)$, est réalisable pour (3.8) et fournit une valeur objective strictement inférieure à α^* , ce qui contredit l'optimalité de (x^*, α^*) . Donc, l'hypothèse est fausse, et x^* est faiblement efficace pour le problème LC – MOO.

2. Soit $x^* \in \mathcal{S}$ la solution optimale unique du problème de Tchebycheff pondéré (3.8), avec valeur optimale α^* . Supposons, par l'absurde, que x^* ne soit pas efficace pour le problème LC – MOO. Il existerait alors un point $x \in \mathcal{S}$, avec $x \neq x^*$, tel que $f_i(x) \leq f_i(x^*)$ pour tout $i = 1, \dots, p$, et $f_k(x) < f_k(x^*)$ pour au moins un indice k . Puisque $\omega_i > 0$ pour tout i et que z^* est fixe, on obtient :

$$\omega_i(f_i(x) - z_i^*) \leq \omega_i(f_i(x^*) - z_i^*) \leq \alpha^*, \quad \forall i = 1, \dots, p,$$

avec une inégalité stricte pour $i = k$: $\omega_k(f_k(x) - z_k^*) < \omega_k(f_k(x^*) - z_k^*) \leq \alpha^*$. Deux cas se présentent pour la valeur $\tilde{\alpha} = \max_{i=1,\dots,p} \omega_i(f_i(x) - z_i^*)$.

- (1) $\tilde{\alpha} < \alpha^*$. Dans ce cas, x fournit une valeur objective strictement inférieure à α^* pour le problème (3.8), ce qui contredit l'optimalité de x^* .
- (2) $\tilde{\alpha} = \alpha^*$. Dans ce cas, x est également une solution optimale du problème (3.8). Cependant, l'unicité de la solution optimale implique que $x = x^*$, ce qui contredit l'hypothèse $x \neq x^*$.

Dans les deux cas, on obtient une contradiction. Par conséquent, l'hypothèse est fausse, et x^* est efficace pour le problème LC – MOO. □

En effet, si la solution optimale du problème (3.8) n'est pas unique, ou si l'unicité est difficile à garantir, le problème peut générer des solutions faiblement efficaces. Steuer et Choo [86] suggèrent d'ajouter un terme d'augmentation, menant au problème de Tchebycheff pondéré augmenté de la forme suivante :

$$\begin{cases} \min_{x \in \mathcal{S}, \alpha \in \mathbb{R}} & \alpha + \rho \sum_{i=1}^p (f_i(x) - z_i^{utop}), \\ \text{s.c.} & \alpha \geq \omega_i (f_i(x) - z_i^*), \quad i = 1, \dots, p. \end{cases} \quad (3.9)$$

où ρ est un scalaire positif suffisamment petit et z^{utop} est le point utopique tel que $z_i^{utop} = z_i^* - \epsilon$.

Les bases théoriques du problème (3.9) se trouvent dans l'ouvrage de Kaliszewski [56].

3.4.4 Autres méthodes scalaires

Parmi les autres méthodes scalaires qui peuvent résoudre le problème LC – MOO, on peut citer une approche hybride combinant la méthode de la somme pondérée et celle des ε -contraintes [70]. La méthode des contraintes élastiques (*Elastic Constraint Method*) représente une variante améliorée de la méthode classique des ε -contraintes, transformant le problème en une séquence de sous-problèmes mono-objectifs. Dans cette approche, les contraintes sur les objectifs sont rendues flexibles : les violations sont autorisées, mais incorporées par des termes de pénalisation dans la fonction objectif, comme détaillé par Ehrgott [35].

La méthode de Benson [12], exploitant le test d'efficacité (3.2), permet d'identifier des solutions efficaces. Son principe repose sur la sélection d'un point initial réalisable $x_0 \in \mathcal{S}$. La résolution du sous-problème paramétré (3.2) confirme l'efficacité de x_0 si celui-ci est optimal ; sinon, elle produit une nouvelle solution efficace x , dont le vecteur objectif $F(x)$ domine $F(x_0)$.

Enfin, la méthode de scalarisation de Pascoletti–Serafini, introduite en 1984, convertit le problème multi-objectifs en un problème scalaire paramétré par des vecteurs a et r dans \mathbb{R}^m , en minimisant une variable auxiliaire t sous la contrainte $a + tr - f(x) \in K$, où K désigne un cône d'ordre convexe fermé et pointu. Géométriquement, elle traduit le cône opposé $-K$ le long de la droite paramétrée par $a + tr$ jusqu'au contact avec la frontière de \mathcal{Y} , le point tangent correspondant à une solution non dominée [36].

3.5 Algorithme NSGA-II

L'algorithme NSGA-II (*Non-dominated Sorting Genetic Algorithm II*), proposé par Deb *et al.* [31] en 2002, constitue une avancée importante dans les algorithmes génétiques. D'une manière générale, les algorithmes génétiques classiques, inspirés des mécanismes de l'évolution naturelle, reposent sur la création d'une population initiale aléatoire, suivie de l'évaluation de la valeur de fitness de chaque individu. Ensuite, des étapes essentielles, telles que la sélection des parents, souvent par tournoi ou par roulette, le croisement pour générer des descendants, la mutation pour introduire de nouvelles variations et le remplacement des individus, sont effectuées. Ce processus est répété jusqu'à ce qu'un critère d'arrêt, comme un nombre fixe de générations ou une stabilisation des résultats, soit atteint.

Dans ce contexte, NSGA-II se distingue des algorithmes génétiques classiques par l'introduction du tri non dominé, qui permet d'organiser les solutions en fronts de Pareto, et par l'évaluation de la distance de crowding, un mécanisme qui soutient la diversité de la population sans nécessiter de paramètres externes. Cette approche diffère de son pré-

décenseur, *NSGA*, qui faisait appel à un mécanisme de partage nécessitant un paramètre pour gérer la diversité. Parmi les améliorations significatives de *NSGA-II* par rapport aux algorithmes génétiques standards, on note la réduction de la complexité du tri non dominé, l'intégration de l'élitisme par la fusion des populations parentale et descendante, ainsi qu'une meilleure convergence et une meilleure diversité sur les *fronts de Pareto*. De plus, *NSGA-II* aborde les contraintes en modifiant la notion de dominance : une solution valide est toujours préférée à une solution non valide, et parmi les solutions non valides, celle qui présente la violation la plus faible est choisie.

Algorithm 5 Algorithme NSGA-II [31]

- 1: INITIALISATION : générer une population P_0 de taille N .
 - 2: Évaluer $F(x)$ pour tous les individus de P_0 . Poser $t = 0$.
 - 3: **repeat**
 - 4: REPRODUCTION : sélectionner des parents (tournoi binaire basé sur rang et crowding), puis appliquer croisement et mutation pour obtenir Q_t (taille N).
 - 5: Évaluer $F(x)$ pour Q_t .
 - 6: ÉLITISME : former $R_t = P_t \cup Q_t$ (taille $2N$) et trier R_t par non-dominance en fronts $\mathcal{F}_1, \mathcal{F}_2, \dots$
 - 7: Construire P_{t+1} en ajoutant successivement les fronts complets tant que la taille reste $\leq N$; pour le front partiel, trier par distance de crowding décroissante et compléter jusqu'à N .
 - 8: $t \leftarrow t + 1$.
 - 9: **until** critère d'arrêt
 - 10: **Retour** : P_t comme approximation du front.
-

4

Méthode de directions réalisables pour l'optimisation multi-objectifs

Introduction

Dans ce chapitre, nous considérons les problèmes d'optimisation multi-objectifs avec des contraintes linéaires *LC-MOO* de la forme suivante :

$$(LC-MOO) \begin{cases} \min_x & F(x) = [f_1(x), f_2(x), \dots, f_p(x)]^T, \\ s.c & x \in \mathcal{S}. \end{cases} \quad (4.1)$$

On suppose que les fonctions objectif $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$, sont continûment différentiables sur l'ensemble réalisable $\mathcal{S} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. De plus, \mathcal{S} est supposé non vide, fermé et borné.

L'objectif de ce chapitre est de proposer et d'analyser des méthodes itératives basées sur des directions réalisables afin d'obtenir une solution (faiblement) efficace du problème *LC-MOO*, tout en cherchant à explorer correctement le front de Pareto. Pour cela, nous poursuivons trois objectifs.

Premièrement, nous expliquons pourquoi les approches paramétriques classiques de scalarisation (pondération, méthode des ε -contraintes) peuvent poser des difficultés en

pratique : elles demandent un choix de paramètres parfois délicat, peuvent produire une couverture partielle ou déséquilibrée du front, et deviennent peu adaptées lorsque le front comporte des zones non convexes. Ces constats justifient l'intérêt des méthodes non paramétriques, en particulier des méthodes de directions qui traitent directement le problème *LC-MOO* sans le transformer en un problème scalaire.

Deuxièmement, nous présentons deux méthodes à directions réalisables pour caractériser une solution efficace du problème *LC-MOO*. À chaque itération, une direction est calculée en résolvant un sous-problème de recherche de direction *SPD* de type quadratique, avec ou sans prise en compte explicite des contraintes non actives. Ensuite, un pas est choisi au moyen d'une recherche linéaire de type Armijo multi-objectif, de façon à rester dans \mathcal{S} et à améliorer les valeurs des fonctions objectif. Nous décrivons d'abord la méthode de Morovati et Pourkarimi [71], puis celle proposée par Ramdani *et al.* [80], en précisant leurs *SPD*, leurs critères d'arrêt et leurs règles de choix du pas.

Troisièmement, pour construire une approximation du front de Pareto, nous introduisons deux procédures d'échantillonnage permettant de générer des ensembles initiaux de points réalisables dans \mathcal{S} . Ces points servent de départs multiples aux algorithmes de directions, ce qui permet d'obtenir un ensemble de solutions (Pareto-de descente que ou (faiblement) efficaces). L'image de ces solutions dans l'espace des objectifs fournit alors une approximation du front, et nous illustrons le comportement des procédures proposées sur un exemple.

4.1 Pourquoi les méthodes non paramétriques

Parmi les approches classiques pour résoudre le problème *LC-MOO* figurent les méthodes de scalarisation. Elles introduisent un ou plusieurs paramètres afin de convertir le problème vectoriel en une famille de problèmes scalaires à objectif unique. Sous certaines hypothèses, chaque choix de paramètres mène à une solution (faiblement) efficace, et la variation de ces paramètres est supposée permettre d'approximer l'ensemble de Pareto et son front.

Dans l'esprit de Morovati et Pourkarimi [71], deux questions se posent alors :

- (Q1) En quoi les méthodes non paramétriques sont-elles préférables aux schémas de scalarisation classiques ?
- (Q2) La méthode de pondération et la méthode des ε -contraintes sont réputées efficaces sur les problèmes convexes et, en théorie, permettent de générer l'ensemble des solutions efficaces en faisant varier leurs paramètres. Dans ce contexte, pourquoi recourir à des méthodes non paramétriques, y compris dans le cas convexe ?

Plusieurs travaux apportent des éléments de réponse. *Das et Dennis* [29] montrent qu'une discrétisation uniforme des poids dans la méthode de pondération n'induit pas, en général, une répartition uniforme des points sur le front de Pareto, même pour un problème convexe. L'exemple discuté à la Section 7 de [42] indique aussi que la méthode de pondération peut n'être vraiment efficace que sur une plage de paramètres très étroite, alors que des méthodes non paramétriques (par exemple de type Newton), sous certaines hypothèses, convergent de manière plus stable vers une solution efficace. Pour la méthode des ε -contraintes, *Mavrotas* [68] met en avant les difficultés pratiques liées au choix des niveaux ε , qui peuvent rendre certains sous-problèmes infaisables ou conduire à des solutions seulement faiblement efficaces. Plus généralement, *Ehrgott* [35] souligne la sensibilité des méthodes de scalarisation aux paramètres, ainsi que leur capacité réduite à explorer de façon homogène des fronts non convexes. *Marler et Arora* [67] de descente quant également ces approches en raison de leur dépendance paramétrique et du risque d'obtenir des optima non stricts. Enfin, *Messac et al.* [69] notent qu'un calibrage inadéquat peut entraîner un coût de calcul élevé sans amélioration proportionnelle de l'approximation du front.

Au-delà de ces constats, nous montrons ci-dessous que construire une approximation de bonne qualité du front de Pareto à l'aide de la méthode de pondération et de la méthode des ε -contraintes peut rester difficile, y compris sur des problèmes simples. Pour cela, nous considérons deux problèmes bi-objectifs : le premier est linéaire (donc convexe) et le second présente un front non convexe. Comme les problèmes linéaires constituent une classe convexe élémentaire, il est particulièrement instructif d'observer que, même dans ce cas, les méthodes paramétriques peuvent produire une couverture partielle ou mal répartie du front. Les deux exemples suivants illustrent ces limites.

Exemple 4.1.

$$\left\{ \begin{array}{l} \min_{x \in \mathbb{R}^2} \quad f_1(x) = x_1, \\ \min_{x \in \mathbb{R}^2} \quad f_2(x) = x_2, \\ \text{s.c.} \quad x_1 + 2x_2 \leq 6, \\ \quad \quad x_1 + x_2 \geq 2, \\ \quad \quad x_1 \geq 0, \quad x_2 \geq 0. \end{array} \right. \quad (4.2)$$

Il s'agit d'un problème *LC-MOO* bi-objectif linéaire. La Figure (4.1) représente l'ensemble réalisable

$$\mathcal{S} = \left\{ x \in \mathbb{R}^2 : x_1 + 2x_2 \leq 6, x_1 + x_2 \geq 2, x_1 \geq 0, x_2 \geq 0 \right\},$$

qui coïncide ici avec l'espace des objectifs puisque $f_1(x) = x_1$ et $f_2(x) = x_2$. L'ensemble des

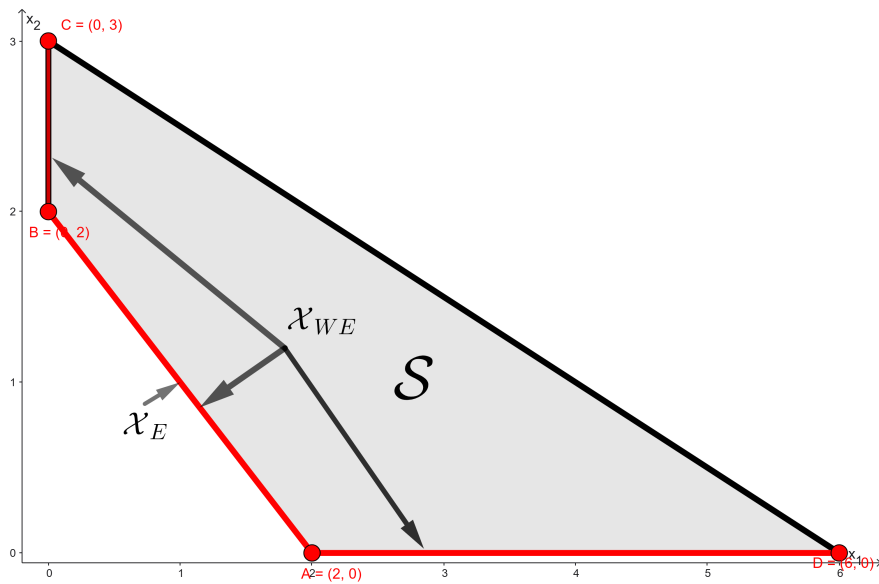


FIGURE 4.1 – Espace de décision et ensembles faiblement efficace et efficace pour l'Exemple (4.1).

solutions faiblement efficaces \mathcal{X}_{WE} est l'union des trois segments CB , BA et AD , tandis que l'ensemble des solutions efficaces \mathcal{X}_E se réduit au segment BA .

Méthode de pondération. On considère la famille scalaire

$$(P_\omega) \quad \min_{x \in \mathcal{S}} \omega x_1 + (1 - \omega)x_2, \quad \omega \in [0, 1].$$

On constate que, pour tout $\omega \in [0, 1]$ sauf $\omega = \frac{1}{2}$, une solution optimale de (P_ω) est un point extrême A ou B . Ainsi, lorsque $\omega \neq \frac{1}{2}$, la méthode de pondération ne génère pas d'autres points efficaces. De plus, lorsque l'optimum scalaire n'est pas unique, (P_ω) admet des solutions alternatives sur des faces entières de \mathcal{S} : pour $\omega = 0$, tout point du segment AD est solution de (P_ω) et n'est que faiblement efficace ; pour $\omega = 1$, tout point du segment CB est également solution et faiblement efficace. Cet exemple convexe met donc en évidence deux difficultés pratiques : la sélection fréquente de solutions extrêmes et la possibilité d'obtenir des ensembles de solutions seulement faiblement efficaces dès que l'optimum scalaire est multiple.

Méthode des ε -contraintes. Appliquons la méthode des ε -contraintes à l'Exemple (4.1) en minimisant $f_1(x)$ sous la contrainte supplémentaire $f_2(x) \leq \varepsilon$. Pour $\varepsilon = 0.1$, on obtient

le sous-problème

$$\left\{ \begin{array}{l} \min \quad x_1, \\ \text{s.c.} \quad x_2 \leq 0.1, \\ \quad \quad x_1 + 2x_2 \leq 6, \\ \quad \quad x_1 + x_2 \geq 2, \\ \quad \quad x_1 \geq 0, \quad x_2 \geq 0, \end{array} \right.$$

dont une solution optimale est $x = (1.9, 0.1)$ et $f(x) = (1.9, 0.1)$, située sur le front. En revanche, pour $\varepsilon < 0$, le sous-problème devient infaisable. À l'opposé, si $\varepsilon = 2.5$, le sous-problème admet un ensemble de solutions optimales de la forme $x = (0, x_2)$ avec $x_2 \in [2, 2.5]$. Un solveur peut alors retourner $x = (0, 2.5)$, donnant $f(x) = (0, 2.5)$, qui est dominé par $(0, 2)$ (même valeur de f_1 et meilleure valeur de f_2). Ce type de sortie correspond à une solution faiblement Pareto-optimale ; il peut induire le décideur en erreur et accroître inutilement l'effort de calcul.

Enfin, pour couvrir l'ensemble du front, il faut choisir ε dans une plage pertinente (ici $[0, 2]$) et adopter une discrétisation adaptée : des valeurs hors plage ou trop espacées laissent des zones non explorées. En pratique, l'absence d'information a priori sur les bornes de f_2 conduit souvent à des essais successifs, ce qui renforce la dépendance paramétrique et compromet une couverture complète du front.

Considérons maintenant le problème bi-objectif non convexe suivant (HANNE3 [27]) :

Exemple 4.2.

$$\left\{ \begin{array}{l} \min_{x \in \mathbb{R}^2} \quad f_1(x) = \sqrt{x_1}, \\ \min_{x \in \mathbb{R}^2} \quad f_2(x) = \sqrt{x_2}, \\ \text{s.c.} \quad x_1 + x_2 \geq 5, \\ \quad \quad 0.1 \leq x_1 \leq 10, \quad 0.1 \leq x_2 \leq 10. \end{array} \right. \quad (4.3)$$

La Figure (4.2) illustre l'espace de décision et l'espace des objectifs associés à l'Exemple (4.2). La sous-figure (4.2a) met en évidence \mathcal{X}_{WE} , composé des segments CB , BA et AD , et \mathcal{X}_E , réduit au segment BA . La sous-figure (4.2b) représente le front de Pareto : \mathcal{Y}_{WE} est formé des segments AB et CD ainsi que de l'arc BA , tandis que \mathcal{Y}_E correspond à l'arc BC , concave.

Méthode de pondération. On considère

$$(P_\omega) \quad \min_{x \in \mathcal{S}} \quad \omega \sqrt{x_1} + (1 - \omega) \sqrt{x_2}, \quad \omega \in [0, 1].$$

On observe que, pour tout $\omega \in [0, 1]$ sauf $\omega = \frac{1}{2}$, une solution optimale de (P_ω) est l'un des

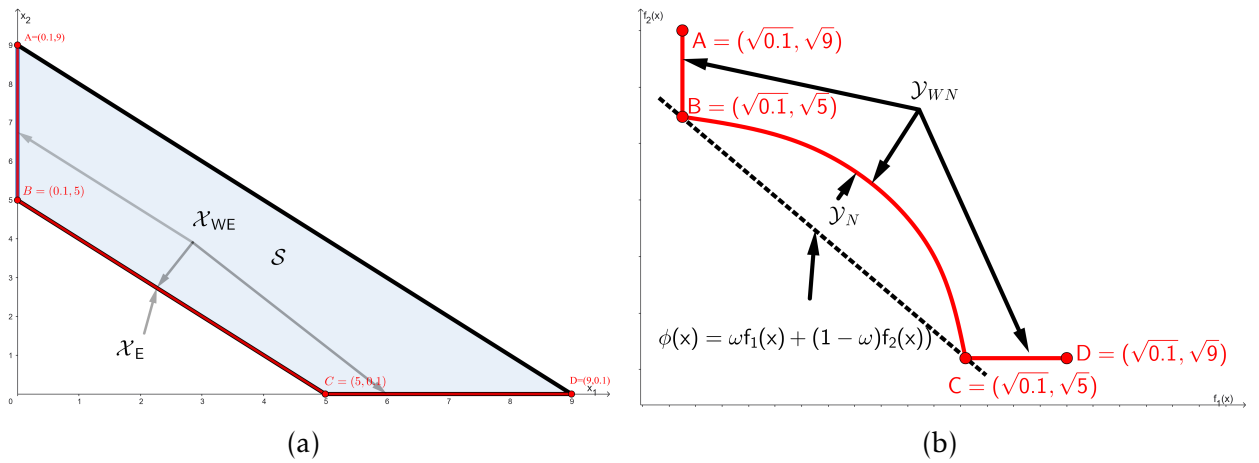


FIGURE 4.2 – Ensembles faiblement efficace et efficace, et front de Pareto pour l'Exemple (4.2).

points B ou C . De plus, pour $\omega = 0$ (resp. $\omega = 1$), tout point du segment AB (resp. CD) est une solution alternative de (P_ω) et n'est que faiblement efficace. Enfin, aucun point de l'arc BC ne peut être produit par la méthode de pondération, quelle que soit la valeur de ω , précisément parce que cette portion du front est concave. Cet exemple met ainsi en évidence la limite structurelle classique : la méthode de pondération ne couvre pas les parties non convexes du front de Pareto.

Méthode des ϵ -contraintes. En minimisant $f_1(x) = \sqrt{x_1}$ sous la contrainte $f_2(x) = \sqrt{x_2} \leq \epsilon$, on obtient le sous-problème

$$\begin{cases} \min_{x \in \mathbb{R}^2} & \sqrt{x_1}, \\ \text{s.c.} & x_1 + x_2 \geq 5, \\ & \sqrt{x_2} \leq \epsilon, \\ & 0.1 \leq x_1 \leq 10, \quad 0.1 \leq x_2 \leq 10. \end{cases}$$

Comme $\sqrt{x_1}$ est croissante en x_1 , l'optimum consiste à minimiser x_1 sous $x_2 \leq \min(\epsilon^2, 10)$ et $x_1 + x_2 \geq 5$, d'où

$$x_1 = \max(0.1, 5 - \min(\epsilon^2, 10)), \quad x_2 = \min(\epsilon^2, 10).$$

En faisant varier ϵ de $\sqrt{0.1} \approx 0.316$ à $\sqrt{10} \approx 3.162$, la méthode peut, en principe, balayer l'arc concave BC , contrairement à la méthode de pondération. Toutefois, elle reste fortement dépendante du choix de ϵ : des valeurs trop petites rendent le sous-problème infaisable (par exemple $\epsilon < \sqrt{0.1}$), tandis que des valeurs trop grandes peuvent conduire à des solutions seulement faiblement efficaces (sur AB ou CD). De plus, construire une approximation régulière du front exige une connaissance préalable (ou une estimation

fiable) des plages de variation des objectifs ainsi qu'un nombre potentiellement élevé de résolutions de sous-problèmes, ce qui augmente le coût de calcul.

Les méthodes de scalarisation, bien qu'efficaces dans de nombreux contextes, demeurent contraintes par une forte dépendance aux paramètres et par une couverture souvent incomplète lorsque le front est non convexe. Ces limites motivent le recours à des approches non paramétriques, à l'instar de Fliege et Svaiter [40], qui construisent des directions de descente communes sans transformation scalaire préalable. Ce cadre a donné lieu à de nombreux prolongements, notamment les méthodes de gradient projeté en optimisation vectorielle [50], les algorithmes de régions de confiance [39, 14], ainsi que des stratégies imposant une distance minimale entre solutions [84]. Parmi les autres contributions, on peut citer les méthodes de gradient réduit pour les problèmes *LC-MOO* [37, 38], les extensions de la méthode des directions réalisables de Zoutendijk [71] et les schémas de descente à ensembles actifs [47]. Plus récemment, des développements incluent les méthodes de gradient conditionnel [5, 23] et les méthodes de gradient conjugué non linéaire sur variétés riemanniennes [22].

Conformément aux objectifs énoncés précédemment, nous examinerons dans la suite deux méthodes de directions réalisables : celle proposée par Morovati et Pourkarimi [71], ainsi que celle développée par Ramdani *et al.* [80], destinées à résoudre le problème *LC-MOO*.

Les deux méthodes reposent sur le même schéma général. L'algorithme part d'un point réalisable, puis répète deux étapes principales :

1. Calculer une direction qui améliore les objectifs tout en préservant la faisabilité, via la résolution d'un *SPD* quadratique.
2. Déterminer un pas le long de cette direction à l'aide d'une recherche linéaire de type Armijo multi-objectif.

4.2 Notions de directions réalisables pour le contexte multi-objectifs

La définition suivante généralise les notions usuelles de direction réalisable et de direction de descente réalisable au cadre de l'optimisation multi-objectifs.

Définition 4.1. Soit $\bar{x} \in \mathcal{S}$. Un vecteur non nul $d \in \mathbb{R}^n$ est appelé direction réalisable en \bar{x} s'il existe $\delta > 0$ tel que, pour tout $\lambda \in (0, \delta)$, on ait $\bar{x} + \lambda d \in \mathcal{S}$. De plus, d est une direction de

descente réalisable au point \bar{x} s'il existe $\delta > 0$ tel que, pour tout $\lambda \in (0, \delta)$, $F(\bar{x} + \lambda d) < F(\bar{x})$ et $\bar{x} + \lambda d \in \mathcal{S}$.

La notion de direction réalisable en un point $\bar{x} \in \mathcal{S}$ coïncide avec celle du cas mono-objectif. En revanche, la direction de descente réalisable est adaptée au contexte multi-objectifs : elle exige, en plus de la faisabilité, une diminution stricte et simultanée de toutes les composantes de F pour des pas $\lambda \in (0, \delta)$.

Lemme 4.1. Soit $\bar{x} \in \mathcal{S}$, et soit $d \in \mathbb{R}^n \setminus \{0\}$ vérifiant

$$\begin{cases} \nabla f_i(\bar{x})^T d < 0, & \forall i = 1, 2, \dots, p, \\ A_j d \leq 0, & \forall j \in J(\bar{x}), \end{cases} \quad (4.4)$$

alors d est une direction de descente réalisable au point \bar{x} .

Démonstration. Soit $\bar{x} \in \mathcal{S}$ et $d \in \mathbb{R}^n \setminus \{0\}$ satisfaisant le système (4.4).

Premièrement, les inégalités $A_j d \leq 0$ pour $j \in J(\bar{x})$ garantissent que les contraintes actives restent satisfaites pour des pas suffisamment petits : si $A_j \bar{x} = b_j$, alors $A_j(\bar{x} + \lambda d) \leq b_j$ pour tout $\lambda > 0$. Pour les contraintes inactives, par continuité, il existe $\delta' > 0$ tel que $\bar{x} + \lambda d \in \mathcal{S}$ pour tout $\lambda \in (0, \delta')$.

Deuxièmement, comme les fonctions f_i sont continûment différentiables, un développement de Taylor d'ordre 1 donne $f_i(\bar{x} + \lambda d) = f_i(\bar{x}) + \lambda \nabla f_i(\bar{x})^T d + o(\lambda)$. La condition $\nabla f_i(\bar{x})^T d < 0$ implique alors l'existence de $\delta_i > 0$ tel que $f_i(\bar{x} + \lambda d) < f_i(\bar{x})$ pour tout $\lambda \in (0, \delta_i)$. En posant $\delta'' = \min_i \delta_i > 0$, on obtient la diminution simultanée de toutes les composantes sur $(0, \delta'')$.

En prenant finalement $\delta = \min(\delta', \delta'') > 0$, on conclut que d est une direction de descente réalisable en \bar{x} . \square

Soit $\bar{x} \in \mathcal{S}$. On note par $\Phi(\bar{x})$ le cône des directions réalisables au point \bar{x} , défini par

$$\Phi(\bar{x}) = \left\{ d \in \mathbb{R}^n, \|d\| \neq 0 : A_j d \leq 0, \text{ pour tout } j \in J(\bar{x}) \right\} \quad (4.5)$$

où $J(\bar{x})$ désigne l'ensemble des indices des contraintes actives au point \bar{x} . et notons par $\Delta(\bar{x})$ le cône des directions de descente au point \bar{x} , tel que

$$\Delta(\bar{x}) = \left\{ d \in \mathbb{R}^n, \|d\| \neq 0 : \nabla f_i(\bar{x})^T d < 0, \text{ pour tout } i = 1, 2, \dots, p \right\}. \quad (4.6)$$

Définition 4.2. Une solution $x^* \in \mathcal{S}$ est appelée point Pareto-de descente que ou point stationnaire du problème (3.1) si, pour tout $d \in \Phi(x^*)$, il existe $i \in \{1, 2, \dots, p\}$ tel que $\nabla f_i(x^*)^T d \geq 0$.

Le corollaire suivant est une conséquence immédiate de la Définition (4.2). Il fournit un système d'inégalités assurant l'existence d'une direction de descente réalisable en tout point non Pareto-de descente que.

Corollaire 4.1. *Soit $x^* \in \mathcal{S}$ un point non Pareto-de descente que alors il existe une direction d au point x^* ; c'est-à-dire*

$$\begin{cases} \nabla f_i(x^*)^T d < 0 & \forall i \in \{1, 2, \dots, p\}, \\ A_j d \leq 0 & \forall j \in J(x^*). \end{cases} \quad (4.7)$$

Ce résultat met en évidence une propriété centrale des MDR : tant que x^* n'est pas Pareto-de descente que, on peut construire une direction d qui améliore simultanément tous les objectifs, tout en respectant les contraintes actives. Dans le cas où $J(x^*) = \emptyset$ (point intérieur de \mathcal{S}), le système se réduit à $\nabla f_i(x^*)^T d < 0$ pour tout $i \in \{1, \dots, p\}$, ce qui correspond à la condition naturelle d'existence d'une direction de descente en l'absence de contraintes actives.

En effet, la Pareto-de descente calité constitue une condition nécessaire d'efficacité. De plus, lorsque chaque fonction objectif f_i est convexe, elle devient une condition suffisante pour l'efficacité faible. Si toutes les fonctions f_i sont strictement convexes, elle fournit en outre une condition suffisante pour l'efficacité. Ces liens sont résumés dans le lemme suivant.

Lemme 4.2. *Soit $x^* \in \mathcal{S}$. Les affirmations suivantes sont vraies :*

1. *Si $x^* \in \mathcal{X}_{WE}$, alors x^* est un point Pareto-de descente que.*
2. *Si toutes les fonctions objectifs f_i sont convexes et x^* est un point Pareto-de descente que, alors $x^* \in \mathcal{X}_{WE}$.*
3. *Si toutes les fonctions objectifs f_i sont strictement convexes et x^* est un point Pareto-de descente que, alors $\bar{x} \in \mathcal{X}_E$.*

Démonstration. Soit $x^* \in \mathcal{S}$.

1. Supposons, par l'absurde, que $x^* \in \mathcal{X}_{WE}$ ne soit pas un point Pareto-de descente que. D'après le Corollaire (4.1), il existe une direction d de descente réalisable en x^* . Ainsi, il existe $\delta > 0$ tel que $x^* + \lambda d \in \mathcal{S}$ et $F(x^* + \lambda d) < F(x^*)$ pour tout $\lambda \in (0, \delta)$, ce qui contredit la faible efficacité de x^* .
2. Supposons que toutes les fonctions objectifs f_i soient convexes et que x^* soit un point Pareto-de descente que, mais $x^* \notin \mathcal{X}_{WE}$. Alors, il existe $\bar{x} \in \mathcal{S}$ tel que $f_i(\bar{x}) < f_i(x^*)$ pour tout $i \in \mathcal{P}$. En posant $d = \bar{x} - x^*$, on obtient une direction réalisable (car \mathcal{S} est convexe). Par convexité de chaque f_i ,

$$f_i(\bar{x}) \geq f_i(x^*) + \nabla f_i(x^*)^T (\bar{x} - x^*),$$

et comme $f_i(\bar{x}) < f_i(x^*)$, on en déduit $\nabla f_i(x^*)^T d < 0$ pour tout $i \in \mathcal{P}$.

Ainsi,

$$\max_{i \in \mathcal{P}} \nabla f_i(x^*)^T d < 0$$

, ce qui contredit la Pareto-de descente calité de x^* .

3. De la même manière, supposons que toutes les fonctions objectifs f_i soient strictement convexes et que x^* soit un point Pareto-de descente que, mais $x^* \notin \mathcal{X}_E$. Alors, il existe $\bar{x} \in \mathcal{S}$, avec $\bar{x} \neq x^*$, tel que $f_i(\bar{x}) \leq f_i(x^*)$ pour tout $i \in \mathcal{P}$, et $f_k(\bar{x}) < f_k(x^*)$ pour un certain $k \in \mathcal{P}$. En posant $d = \bar{x} - x^*$, on obtient une direction réalisable. Par stricte convexité,

$$f_i(\bar{x}) > f_i(x^*) + \nabla f_i(x^*)^T (\bar{x} - x^*),$$

et la condition $f_i(\bar{x}) \leq f_i(x^*)$ implique alors $\nabla f_i(x^*)^T d < 0$ pour tout $i \in \mathcal{P}$. On obtient donc $\max_{i \in \mathcal{P}} \nabla f_i(x^*)^T d < 0$, ce qui contredit la Pareto-de descente calité de x^* . □

Pour illustrer la notion de direction de descente réalisable, considérons l'exemple suivant.

Exemple 4.3.

$$\left\{ \begin{array}{l} \min_{x \in \mathbb{R}^2} f_1(x) = (x_1 - 6)^2 + (x_2 - 2)^2, \\ \min_{x \in \mathbb{R}^2} f_2(x) = x_1^2 + x_2^2, \\ \text{s.c.} \quad -x_1 + 2x_2 \leq 4, \\ \quad \quad 3x_1 + 2x_2 \leq 12, \\ \quad \quad x_1 \geq 0, \quad x_2 \geq 0. \end{array} \right. \quad (4.8)$$

Notons que les deux fonctions objectifs sont quadratiques, avec des matrices hessiennes définies positives ; elles sont donc strictement convexes. Les Figures (4.3a) et (4.3b) illustrent l'espace de décision \mathcal{S} et l'ensemble Pareto-efficace \mathcal{S}_E associés. L'ensemble \mathcal{S} est le polyèdre délimité par les points extrêmes $d = (0, 0)$, $a = (0, 2)$, $b = (2, 3)$ et $c = (4, 0)$. L'ensemble des solutions efficaces \mathcal{X}_E (en rouge) est constitué de deux segments : $[d, e]$ et $[e, c]$, où $e \approx (3.27, 1.09)$.

La Figure (4.3a) met l'accent sur les cônes au point $b = (2, 3)$, à savoir le cône des directions réalisables $\Phi(b)$, où les contraintes actives sont labellisées A_1 et A_2 , donc

$$\Phi(b) = \left\{ d \in \mathbb{R}^2 \setminus \{0\} : -d_1 + 2d_2 \leq 0, \quad 3d_1 + 2d_2 \leq 0 \right\}.$$

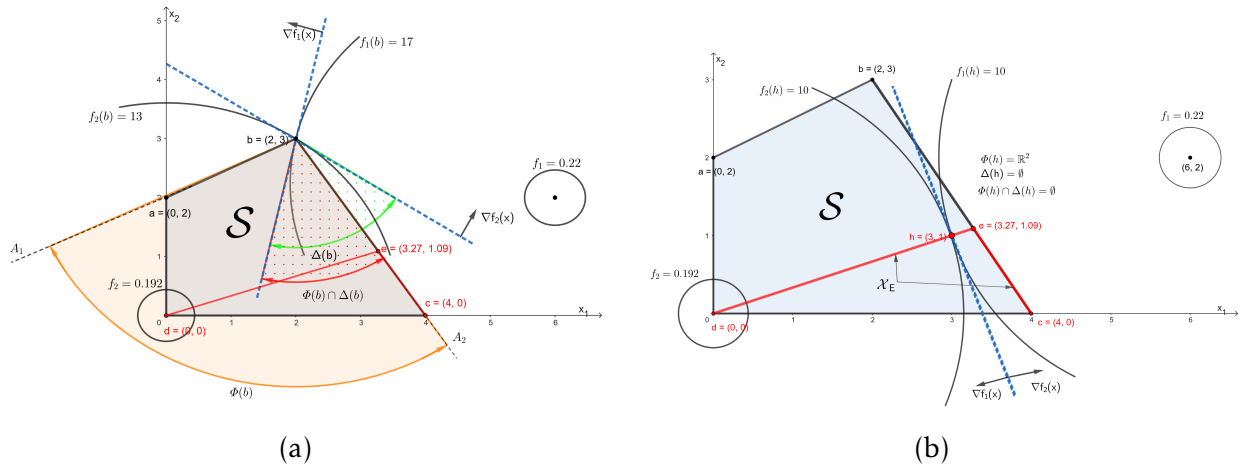


FIGURE 4.3 – Interprétation géométrique de la direction de descente réalisable de l'exemple (4.3).

et le cône des directions de descente $\Delta(b)$ (secteur où $\nabla f_i(b)^\top d < 0$ pour $i = 1, 2$) tel que

$$\Delta(b) = \left\{ d \in \mathbb{R}^2 \setminus \{0\} : -8d_1 + 2d_2 < 0, \quad 4d_1 + 6d_2 < 0 \right\}.$$

Le cône des directions de descente réalisables en b correspond à l'intersection $\Delta(b) \cap \Phi(b)$, donnée par le système d'inégalités suivant :

$$\Delta(b) \cap \Phi(b) = \left\{ d \in \mathbb{R}^2, d \neq 0 : \begin{array}{l} -8d_1 + 2d_2 < 0, \\ 4d_1 + 6d_2 < 0, \\ -d_1 + 2d_2 \leq 0, \\ 3d_1 + 2d_2 \leq 0. \end{array} \right\}. \quad (4.9)$$

Comme on peut le constater sur la Figure (4.3a), le cône de directions de descente réalisable $\Delta(b) \cap \Phi(b)$ est non vide. Cela montre l'existence de directions de descente réalisables et confirme que b n'est pas Pareto-de descente que (Lemme (4.1) et Corollaire (4.1)).

La Figure (4.3b) prolonge l'analyse au point $h = (3, 1)^\top$, qui est un point intérieur de \mathcal{S} (donc $J(h) = \emptyset$). Dans ce cas, $\Phi(h) = \mathbb{R}^2 \setminus \{0\}$, tandis que $\Delta(h) = \emptyset$ en raison de l'orientation des droites tangentes représentées par des lignes pointillées en ce point. On en déduit que h est Pareto-de descente que (Définition (4.2)). En vertu du Lemme (4.2), tout point Pareto-de descente que est un point efficace.

3. Il existe $\lambda \in \mathbb{R}_+^{p+|J(x)|}$ tel que $\sum_{i=1}^{p+|J(x)|} \lambda_i = 1$ et, pour tout $d \in \mathbb{R}^n \setminus \{0\}$,

$$\sum_{i=1}^{p+|J(x)|} \lambda_i \nabla G_i(x)^\top d + \frac{1}{2} \|d\|^2 > 0. \quad (4.13)$$

4. Il existe $\lambda \in \mathbb{R}_+^{p+|J(x)|}$ tel que $\sum_{i=1}^{p+|J(x)|} \lambda_i = 1$ et

$$\sum_{i=1}^{p+|J(x)|} \lambda_i \nabla G_i(x) = 0. \quad (4.14)$$

L'idée centrale est la suivante : le terme $\frac{1}{2} \|d\|^2$ rend le sous-problème fortement convexe, ce qui garantit l'existence et l'unicité du minimum pour toute scalarisation par somme pondérée. En exploitant cette propriété, on obtient des conditions d'optimalité en gradients, qui coïncident avec les conditions de Pareto-de descente calité pour (3.1).

D'après le Théorème (4.1), si $x \in \mathcal{S}$ n'est pas Pareto-de descente que, toute solution faiblement efficace de (4.12) fournit une direction réalisable améliorante. Afin de sélectionner une direction unique, *Morovati et Pourkarimi* utilisent une scalarisation min-max et considèrent le problème scalaire non contraint :

$$\min_{d \in \mathbb{R}^n} \left\{ \frac{1}{2} \|d\|^2 + \max_{i \in \{1, 2, \dots, p+|J(x)\}} \nabla G_i(x)^\top d \right\}. \quad (4.15)$$

Le lemme suivant assure que ce sous-problème admet au moins une solution optimale.

Lemme 4.3. *La fonction*

$$\phi(d) = \frac{1}{2} \|d\|^2 + \max_{i \in \{1, 2, \dots, p+|J(x)\}} \nabla G_i(x)^\top d$$

atteint son minimum sur \mathbb{R}^n .

En particulier, le terme quadratique rend ϕ coercive, ce qui force l'existence d'un minimum. De plus, la stricte convexité de $\frac{1}{2} \|d\|^2$ implique l'unicité de la solution.

Pour $x \in \mathcal{S}$, on note $\gamma(x)$ la valeur optimale de (4.15) et $d(x)$ son minimum :

$$\gamma(x) := \min_{d \in \mathbb{R}^n} \left\{ \frac{1}{2} \|d\|^2 + \max_{i \in \{1, 2, \dots, p+|J(x)\}} \nabla G_i(x)^\top d \right\}, \quad (4.16)$$

$$d(x) := \arg \min_{d \in \mathbb{R}^n} \left\{ \frac{1}{2} \|d\|^2 + \max_{i \in \{1, 2, \dots, p+|J(x)\}} \nabla G_i(x)^\top d \right\}. \quad (4.17)$$

Lemme 4.4. *Le sous-problème (4.15) admet une solution optimale unique.*

La justification est classique : si deux minimums distincts existaient, l'évaluation au point milieu donnerait une valeur strictement plus petite que la valeur optimale, ce qui est impossible.

Le sous-problème (4.15) est non différentiable à cause de l'opérateur max. Pour obtenir un problème lisse, on introduit une variable auxiliaire z qui majore $\nabla G_i(x)^\top d$ pour tout i . On obtient alors la reformulation quadratique suivante, avec contraintes linéaires :

$$\left\{ \begin{array}{l} \min_{z,d} \quad z + \frac{1}{2} \|d\|^2, \\ \text{s.t.} \quad \nabla f_i(x)^\top d \leq z, \quad \forall i = 1, \dots, p \\ \quad \quad A_j d \leq z, \quad \forall j \in J(x) \\ \quad \quad z \in \mathbb{R}, d \in \mathbb{R}^n. \end{array} \right. \quad (4.18)$$

Cette reformulation est équivalente : pour un d donné, le meilleur choix est $z = \max_i \nabla G_i(x)^\top d$, et toute paire réalisable (z, d) fournit une majoration de ce maximum. Ainsi, les deux formulations partagent la même valeur optimale et le même $d(x)$.

La résolution du SPD (4.18) fournit à la fois une direction de descente réalisable (quand elle existe) et un indicateur de critère d'arrêt pour l'algorithme. Le lemme suivant regroupe les propriétés utilisées par l'algorithme.

Lemme 4.5. *Considérons le problème (3.1). Pour tout $x \in \mathcal{S}$, les affirmations suivantes sont vérifiées :*

1. $\gamma(x) \leq 0$.
2. *Les conditions ci-après sont équivalentes :*
 - (i) *Le point x n'est pas un point Pareto-de descente que.*
 - (ii) $\gamma(x) < 0$.
 - (iii) $d(x) \neq 0$.
3. *Les fonctions $x \mapsto d(x)$ et $x \mapsto \gamma(x)$ sont continues.*

On peut interpréter ces points comme suit. D'abord, $\gamma(x) \leq 0$ car la valeur de (4.15) en $d(x) = 0$ est nulle, ce qui fournit une borne supérieure. Ensuite, x est Pareto-de descente que exactement lorsque la meilleure valeur atteignable est $\gamma(x) = 0$, ce qui correspond aussi à $d(x) = 0$. Enfin, la continuité de $x \mapsto d(x)$ et $x \mapsto \gamma(x)$ relève de l'optimisation paramétrique et s'appuie notamment sur des arguments similaires à ceux du lemme 4 de Fliege et al. [42].

Comme dans toute méthode de direction réalisable, une fois la direction $d(x)$ calculée (si x n'est pas Pareto-de descente que), il faut choisir un pas $\lambda > 0$ qui préserve la

faisabilité et améliore tous les objectifs. Morovati et Pourkarimi [71] adoptent une règle d'Armijo adaptée au cas multi-objectifs. En suivant Fliege et Svaiter [40], la condition s'écrit :

$$F(x + \lambda d(x)) \leq F(x) + \beta \lambda \gamma(x) \quad (4.19)$$

où $\beta \in (0, 1)$ est le paramètre d'Armijo. L'inégalité est comprise composante par composante, ce qui impose une amélioration simultanée de toutes les fonctions objectif lorsque la condition $\gamma(x) < 0$ soit vérifiée.

Le lemme suivant garantit qu'un pas strictement positif satisfaisant (4.19) existe.

Lemme 4.6. *Supposons que $x \in \mathcal{S}$ soit un point non de descente que. Alors, pour tout $\beta \in (0, 1)$, il existe $\bar{\lambda} \in (0, 1]$ tel que, pour tout $\lambda \in (0, \bar{\lambda}]$, on ait*

- $x + \lambda d(x) \in \mathcal{S}$ et
- $F(x + \lambda d(x)) \leq F(x) + \beta \lambda \gamma(x)$.

La preuve est de même nature que celle du Lemme 15 de [40].

L'algorithme complet de Morovati et Pourkarimi [71] est rappelé ci-dessous. Il alterne : (i) la résolution du SPD (4.18), (ii) un test d'arrêt basé sur $\gamma(x^k)$, puis (iii) une recherche de pas garantissant la faisabilité et l'amélioration de tout les objectif en utilisant la règle d'Armijo multi-objectifs.

Algorithm 6 Méthode de Morovati et Pourkarimi

- 1: **Entrées** : choisir $0 < \beta < 1$, $x^0 \in \mathcal{S}$, $\varepsilon > 0$.
- 2: **Initialiser** $k \leftarrow 0$.
- 3: **repeat**
- 4: **(Recherche de direction de descente réalisable)** Résoudre le sous-problème (4.18) au point x^k et obtenir

$$\gamma_k \leftarrow \gamma(x^k), \quad d^k \leftarrow d(x^k).$$

- 5: **(Test d'arrêt)** Vérifier la condition $|\gamma_k| < \varepsilon$.
- 6: **(Recherche d'un pas réalisable)** Initialiser $\alpha_k \leftarrow 1$.
- 7: Remplacer $\alpha_k \leftarrow \alpha_k/2$ tant que $A(x^k + \alpha_k d^k) \not\leq 0$.
- 8: Fixer $\lambda^k \leftarrow \alpha_k$.
- 9: Remplacer $\lambda^k \leftarrow \lambda^k/2$ tant qu'il existe $i \in \{1, 2, \dots, p\}$ tel que

$$f_i(x^k + \lambda^k d^k) > f_i(x^k) + \beta \lambda^k \gamma(x^k).$$

- 10: **(Mise à jour)** Poser $x^{k+1} \leftarrow x^k + \lambda^k d^k$ et $k \leftarrow k + 1$.
- 11: **until** $|\gamma_k| < \varepsilon$
- 12: **Sortie** : retourner x^k .

Le théorème suivant énonce la convergence : sous des hypothèses de convexité adaptées et une bornitude d'un ensemble de niveau, tout point d'accumulation de la suite produite par l'Algorithme (6) est une solution faiblement efficace (ou efficace).

Théorème 4.2. *Considérons le Problème (3.1). Supposons que $\{x^k\}$ soit la suite générée par l'Algorithme (6). On suppose que F est \mathbb{R}^p -convexe (respectivement strictement \mathbb{R}^p -convexe) et continûment différentiable. Enfin, on suppose que l'ensemble de niveau*

$$\Gamma(x^0) = \{x \in \mathcal{S} \mid F(x) \leq F(x^0)\}$$

est borné. Alors, la suite $\{x^k\}$ admet au moins un point d'accumulation. De plus, tout point d'accumulation de $\{x^k\}$ est une solution faiblement efficace (respectivement une solution efficace) du Problème (3.1).

L'argument de convergence peut se résumer ainsi (voir [71] pour les détails). La règle d'Armijo entraîne une diminution des objectifs à chaque itération non de descente que, ce qui maintient la suite $\{x^k\}$ dans l'ensemble de niveau $\Gamma(x^0)$. Comme cet ensemble est borné, la suite admet au moins un point d'accumulation. Ensuite, on établit que $\gamma(x^k) \rightarrow$

0. Par continuité (Lemme (4.5)), tout point d'accumulation \bar{x} vérifie $\gamma(\bar{x}) = 0$, donc \bar{x} est Pareto-de descente que. Enfin, sous \mathbb{R}^p -convexité (ou stricte \mathbb{R}^p -convexité) et en utilisant les relations entre de descente calité et efficacité, on conclut que \bar{x} est faiblement efficace (ou efficace).

4.4 Méthode de directions réalisables de Ramdani et al. [80]

Rappelons que le *SPD* de Morovati et Pourkarimi [71], adapté au problème (3.1), s'écrit, pour un point $x \in \mathcal{S}$, sous la forme suivante :

$$\left\{ \begin{array}{ll} \min_{z,d} & z + \frac{1}{2}\|d\|^2, \\ \text{s.c.} & \nabla f_i(x)^\top d \leq z, \quad \forall i \in \mathcal{P}, \\ & A_j d \leq z, \quad \forall j \in J(x), \\ & z \in \mathbb{R}, \quad d \in \mathbb{R}^n. \end{array} \right. \quad (4.20)$$

Dans ce *SPD*, la variable auxiliaire z joue le rôle d'une borne supérieure commune : elle majore à la fois les dérivées directionnelles des fonctions objectif et les contributions directionnelles des contraintes actives. Le terme $\frac{1}{2}\|d\|^2$ sert de régularisation : il contrôle la norme de la direction et assure l'unicité de la solution optimale.

Cependant, pour résoudre le problème (3.1), nous avons mis en évidence deux limites du *SPD* (4.20) qui peuvent dégrader l'efficacité de convergence de la méthode de Morovati et Pourkarimi [71] :

1. Le *SPD* (4.20) n'intègre que les contraintes actives, en laissant de côté les contraintes inactives. Cette approximation peut provoquer des trajectoires en zigzag et une progression lente. En optimisation mono-objectif, le contre-exemple de Wolfe [93] illustre qu'ignorer les contraintes inactives peut mener à une convergence vers des points non optimaux. En pratique, contraintes actives et inactives influencent toutes deux la géométrie des déplacements réalisables, ce qui contribue à limiter les changements brusques près des frontières associées aux contraintes inactives [10].
2. La même variable z majore simultanément les dérivées directionnelles des objectifs et celles des contraintes actives. Lorsque l'échelle des gradients des objectifs est très différente de celle des contraintes actives, ce couplage peut orienter la résolution du *SPD* vers une compensation défavorable, au détriment d'une amélioration marquée des objectifs, ce qui peut ralentir la convergence.

Pour surmonter ces deux difficultés, *Ramdani et al.* [80] introduisent un nouveau *SPD* qui (i) tient compte de toutes les contraintes via un cône réduit de directions réalisables, et (ii) choisit une variable z dépendant uniquement des gradients des fonctions objectif. Ainsi, la direction obtenue privilégie l'amélioration des objectifs tout en préservant la faisabilité.

Afin d'illustrer la forme du cône réduit $\Phi_2(x)$ en un point donné, considérons l'ensemble réalisable \mathcal{S} associé à l'Exemple (4.3). Dans ce cas, les contraintes s'écrivent

$$\begin{aligned} e_1(x) &= -x_1 + 2x_2 - 4 \leq 0, \\ e_2(x) &= 3x_1 + 2x_2 - 12 \leq 0, \\ e_3(x) &= -x_1 \leq 0, \\ e_4(x) &= -x_2 \leq 0. \end{aligned}$$

Sur la Figure (4.3a), considérons le point extrême $b = (2, 3)$. On a alors

$$e_1(b) = -2 + 6 - 4 = 0, \quad e_2(b) = 6 + 6 - 12 = 0, \quad e_3(b) = -2, \quad e_4(b) = -3.$$

Dès lors, pour une direction $d = (d_1, d_2)$, les conditions

$$e_j(b) + A_j d \leq 0, \quad j = 1, \dots, 4,$$

se traduisent par

$$\begin{aligned} j = 1 : 0 + (-d_1 + 2d_2) \leq 0 &\iff -d_1 + 2d_2 \leq 0, \\ j = 2 : 0 + (3d_1 + 2d_2) \leq 0 &\iff 3d_1 + 2d_2 \leq 0, \\ j = 3 : -2 + (-d_1) \leq 0 &\iff -d_1 \leq 2 \iff d_1 \geq -2, \\ j = 4 : -3 + (-d_2) \leq 0 &\iff -d_2 \leq 3 \iff d_2 \geq -3. \end{aligned}$$

Par conséquent,

$$\Phi_2(b) = \left\{ d \in \mathbb{R}^2 \setminus \{0\} : \begin{array}{l} -d_1 + 2d_2 \leq 0, \\ 3d_1 + 2d_2 \leq 0, \\ d_1 \geq -2, \\ d_2 \geq -3 \end{array} \right\}.$$

Par ailleurs, en évaluant les gradients des deux fonctions objectif au point $b = (2, 3)$, on obtient

$$\nabla f_1(b) = (-8, 2), \quad \nabla f_2(b) = (4, 6).$$

Ainsi, les conditions de descente

$$\nabla f_i(b)^\top d < 0, \quad i = 1, 2,$$

s'écrivent explicitement sous la forme

$$-8d_1 + 2d_2 < 0, \quad 4d_1 + 6d_2 < 0.$$

On en déduit que le cône des directions de descente au point b est

$$\Delta(b) = \{d \in \mathbb{R}^2 \setminus \{0\} : -8d_1 + 2d_2 < 0, 4d_1 + 6d_2 < 0\}.$$

Par conséquent, la recherche d'une direction de descente réalisable en b revient à déterminer un vecteur non nul

$$d \in \Phi_2(b) \cap \Delta(b),$$

c'est-à-dire à résoudre le système d'inégalités

$$\begin{cases} -d_1 + 2d_2 \leq 0, \\ 3d_1 + 2d_2 \leq 0, \\ d_1 \geq -2, \\ d_2 \geq -3, \\ -8d_1 + 2d_2 < 0, \\ 4d_1 + 6d_2 < 0. \end{cases}$$

Tout vecteur d satisfaisant ce système définit alors une direction de descente réalisable au point b .

Comme dans les méthodes classiques de directions réalisables introduites par Zoutendijk [103], la recherche d'une direction satisfaisant simultanément les contraintes de faisabilité et de descente peut être reformulée comme la résolution d'un problème d'optimisation auxiliaire. C'est dans cet esprit que, pour tout point courant $x \in \mathcal{S}$, nous introduisons le sous-problème de recherche de direction (SPD) suivant :

$$\begin{cases} \min_{z,d} & z + \frac{1}{2}\|d\|^2, \\ \text{s.c.} & \nabla f_i(x)^\top d \leq z, \quad \forall i \in \mathcal{P}, \\ & e_j(x) + A_j d \leq 0, \quad \forall j \in \mathcal{M}, \\ & z \in \mathbb{R}, \quad d \in \mathbb{R}^n. \end{cases} \quad (4.21)$$

Ce sous-problème de recherche de direction a été construit précisément pour répondre aux limites identifiées dans le *SPD* de Morovati et Pourkarimi (4.20). D'une part, il remplace la prise en compte exclusive des contraintes actives par le cône réduit $\Phi_2(x)$, qui incorpore l'ensemble des contraintes du polyèdre, qu'elles soient actives ou inactives. Cette reformulation permet de mieux refléter la géométrie réelle de l'ensemble réalisable, de prévenir certaines trajectoires oscillatoires au voisinage des frontières et de produire des directions plus stables.

D'autre part, le rôle de la variable auxiliaire z dans (4.20), majore simultanément les dérivées directionnelles des objectifs et celles des contraintes actives, elle est ici dédiée uniquement au contrôle des gradients des fonctions objectif. Les contraintes interviennent désormais séparément à travers les conditions définissant $\Phi_2(x)$. Cette dissociation évite les effets de compensation entre amélioration des objectifs et contrôle des contraintes, et oriente davantage le *SPD* vers la recherche de directions réellement favorables du point de vue multi-objectif, tout en maintenant la faisabilité.

Pour tout $x \in \mathcal{S}$, le *SPD* (4.21) est équivalent au problème suivant :

$$\min_{d \in \Phi_2(x)} \Gamma(d, x), \quad \text{où} \quad \Gamma(d, x) = \frac{1}{2} \|d\|^2 + \max_{i \in \mathcal{P}} \nabla f_i(x)^\top d. \quad (4.22)$$

Dans la suite, on note $\varphi(x)$ la valeur optimale, et $(d(x), z(x))$ une solution optimale du *SPD* (4.21) :

$$\varphi(x) = \min_{d \in \Phi_2(x)} \left\{ \frac{1}{2} \|d\|^2 + \max_{i \in \mathcal{P}} \nabla f_i(x)^\top d \right\}, \quad (4.23)$$

$$d(x) = \arg \min_{d \in \Phi_2(x)} \left\{ \frac{1}{2} \|d\|^2 + \max_{i \in \mathcal{P}} \nabla f_i(x)^\top d \right\}, \quad (4.24)$$

$$z(x) = \max_{i \in \mathcal{P}} \nabla f_i(x)^\top d(x). \quad (4.25)$$

On donne maintenant quelques propriétés utiles du *SPD* (4.21), qui serviront ensuite pour l'arrêt et la convergence.

Lemme 4.7. *Pour tout $x \in \mathcal{S}$, on a $\varphi(x) \leq 0$.*

Démonstration. Pour tout $x \in \mathcal{S}$, comme $d = 0 \in \Phi_2(x)$,

$$\Gamma(0, x) = \frac{1}{2} \|0\|^2 + \max_{i \in \mathcal{P}} \nabla f_i(x)^\top 0 = 0.$$

Ainsi,

$$\varphi(x) = \min_{d \in \Phi_2(x)} \Gamma(d, x) \leq \Gamma(0, x) = 0.$$

□

Le lemme suivant relie la valeur $z(x)$ à l'existence d'une direction de descente réalisable.

Lemme 4.8. *Pour tout $x \in \mathcal{S}$, on a $z(x) \leq 0$. De plus, si $z(x) < 0$, alors $d(x)$ est une direction de descente réalisable en x .*

Démonstration. D'après le Lemme (4.7), on a $\varphi(x) = z(x) + \frac{1}{2}\|d(x)\|^2 \leq 0$, donc $z(x) \leq 0$. Si $z(x) < 0$, alors, pour $(z(x), d(x))$ solution optimale du SPD (4.21),

$$\begin{cases} \nabla f_i(x)^\top d(x) \leq z(x) < 0, & \forall i \in \mathcal{P}, \\ e_j(x) + A_j d(x) \leq 0, & \forall j \in \mathcal{M}. \end{cases}$$

La seconde famille d'inégalités donne $d(x) \in \Phi_2(x)$, donc $d(x)$ est une direction réalisable en x . La première famille garantit une dérivée directionnelle strictement négative pour chaque objectif. Ainsi, $d(x)$ est une direction de descente réalisable en x . \square

Lemme 4.9. *Le SPD (4.21) admet une solution optimale unique pour tout $x \in \mathcal{S}$.*

Démonstration. Pour tout $x \in \mathcal{S}$, le SPD (4.21) est réalisable car $(z, d) = (0, 0)$ satisfait les contraintes. De plus, le SPD (4.21) est équivalent à (4.22), qui minimise, sur l'ensemble non vide, fermé et convexe $\Phi_2(x)$, une fonction fortement convexe en d (somme d'un terme quadratique strictement convexe et d'une fonction convexe). Par conséquent, le minimum est unique. \square

Lemme 4.10. *Pour tout $x \in \mathcal{S}$, les applications $x \mapsto d(x)$, $x \mapsto z(x)$ et $x \mapsto \varphi(x)$ sont continues.*

Démonstration. La continuité de $d(x)$ découle du Lemme 3.3 de [49]. Comme $z(x) = \max_{i \in \mathcal{P}} \nabla f_i(x)^\top d(x)$ et que $\nabla f_i(x)$ et $d(x)$ sont continues, $z(x)$ est continue en tant que maximum de fonctions continues. Ainsi, $\varphi(x) = z(x) + \frac{1}{2}\|d(x)\|^2$ est continue comme somme de fonctions continues. \square

Le lemme suivant relie Pareto-de descente qualité, solution optimale et valeur optimale du SPD. Ces équivalences servent directement de critères d'arrêt pour l'algorithme de MDR présenté ensuite.

Lemme 4.11. *Soit $x \in \mathcal{S}$. Les propositions suivantes sont équivalentes :*

1. x est un point Pareto-de descente que du problème (3.1).
2. $z(x) = 0$.
3. $d(x) = 0$.
4. $\varphi(x) = 0$.

Démonstration. — (1) \Rightarrow (2) Supposons, par contradiction, que $z(x) < 0$. D'après le Lemme (4.8), $d(x)$ est alors une direction de descente réalisable en x , ce qui contredit la Pareto-de descente calité (Corollaire (4.1)).

— (2) \Rightarrow (3) Si $z(x) = 0$, alors

$$\varphi(x) = z(x) + \frac{1}{2}\|d(x)\|^2 = \frac{1}{2}\|d(x)\|^2.$$

Comme $\varphi(x) \leq 0$ (Lemme (4.7)), on obtient $\|d(x)\|^2 \leq 0$, donc $d(x) = 0$.

— (3) \Rightarrow (4) Si $d(x) = 0$, alors

$$z(x) = \max_{i \in \mathcal{P}} \nabla f_i(x)^\top 0 = 0,$$

et par conséquent $\varphi(x) = 0$.

— (4) \Rightarrow (1) Supposons que $\varphi(x) = 0$ et que x ne soit pas Pareto-de descente que. Alors, il existe $d \in \Phi_2(x)$ tel que $\nabla f_i(x)^\top d < 0$ pour tout $i \in \mathcal{P}$. Posons

$$z = \max_{i \in \mathcal{P}} \nabla f_i(x)^\top d < 0.$$

Pour $\lambda \geq 0$, le vecteur $\bar{d} = \lambda d$ appartient à $\Phi_2(x)$, et

$$\Gamma(\bar{d}, x) = \frac{1}{2}\lambda^2\|d\|^2 + \lambda z.$$

La fonction en λ est strictement convexe et atteint son minimum en

$$\lambda^* = -\frac{z}{\|d\|^2} > 0,$$

ce qui donne

$$\Gamma(\lambda^* d, x) = -\frac{z^2}{2\|d\|^2} < 0.$$

Ainsi, $\varphi(x) \leq \Gamma(\lambda^* d, x) < 0$, contradiction avec $\varphi(x) = 0$. Donc x est Pareto-de descente que. □

À l'itération k , on calcule d'abord la direction $d(x^k)$ en résolvant le SPD (4.21). La seconde étape consiste à déterminer un pas λ^k tel que le nouveau point

$$x^{k+1} = x^k + \lambda^k d(x^k)$$

reste dans \mathcal{S} et améliore simultanément tous les objectifs. Plus précisément, on cherche

$\lambda^k > 0$ tel que

$$F(x^k + \lambda^k d(x^k)) < F(x^k).$$

Soit donc $x^k \in \mathcal{S}$ non Pareto-de descente que, et posons $d^k = d(x^k)$. On introduit l'ensemble

$$\Omega(x^k) = \{j \in \mathcal{M} \setminus J(x^k) \mid A_j d^k > 0\}, \quad (4.26)$$

qui regroupe les contraintes inactives en x^k mais susceptibles de devenir actives en avançant dans la direction d^k .

Proposition 4.1. *Pour tout $x^k \in \mathcal{S}$, si d^k est une direction réalisable à x^k , alors $\Omega(x^k) \neq \emptyset$.*

Démonstration. Supposons par l'absurde que $\Omega(x^k) = \emptyset$. Alors $A_j d^k \leq 0$ pour tout $j \in \mathcal{M} \setminus J(x^k)$. Comme d^k est réalisable, on a aussi $A_j d^k \leq 0$ pour tout $j \in J(x^k)$. Donc $A_j d^k \leq 0$ pour tout $j \in \mathcal{M}$, ce qui implique $x^k + \lambda d^k \in \mathcal{S}$ pour tout $\lambda \geq 0$. Ceci contredit la bornitude de \mathcal{S} . Ainsi, $\Omega(x^k) \neq \emptyset$. \square

On déduit maintenant un pas maximal garantissant la faisabilité. Pour les contraintes actives ($j \in J(x^k)$), on a $A_j x^k = b_j$ et $A_j d^k \leq 0$, donc

$$A_j(x^k + \lambda d^k) \leq b_j, \quad \forall \lambda \geq 0.$$

Pour les contraintes inactives, $b_j - A_j x^k > 0$. Si $j \in \Omega(x^k)$, il faut imposer

$$\lambda \leq \frac{b_j - A_j x^k}{A_j d^k}.$$

On définit alors

$$\lambda_{\max}^k = \min_{j \in \Omega(x^k)} \left\{ \frac{b_j - A_j x^k}{A_j d^k} \right\} > 0, \quad (4.27)$$

qui est le pas maximal préservant la faisabilité à partir du point x^k tout au long de la direction d^k .

Contrairement à la MDR de Morovati et Pourkarimi [71] qui fixe $\lambda_{\max}^k = 1$ à chaque itération, la méthode de Ramdani *et al.* [80] calcule λ_{\max}^k à partir de (4.27). La proposition suivante précise que le point obtenu au pas maximal est réalisable et qu'au moins une contrainte devient active.

Proposition 4.2. *Soit $x^k \in \mathcal{S}$. Si d^k est une direction faisable améliorante en x^k , alors*

$$x^{k+1} = x^k + \lambda_{\max}^k d^k \in \mathcal{S},$$

où λ_{\max}^k est défini par (4.27), et $J(x^{k+1}) \neq \emptyset$.

Démonstration. Soit $x^k \in \mathcal{S}$ et soit d^k une direction faisable améliorante en x^k . D'après (4.27) et la Proposition (4.1), il existe $j_0 \in \Omega(x^k)$ tel que

$$\lambda_{\max}^k = \frac{b_{j_0} - A_{j_0} x^k}{A_{j_0} d^k}.$$

Alors,

$$A_{j_0}(x^k + \lambda_{\max}^k d^k) = A_{j_0} x^k + \left(\frac{b_{j_0} - A_{j_0} x^k}{A_{j_0} d^k} \right) A_{j_0} d^k = b_{j_0}.$$

La contrainte j_0 est donc active en x^{k+1} , ce qui implique $J(x^{k+1}) \neq \emptyset$. \square

Le lemme suivant garantit l'existence d'un pas $\lambda^k \in (0, \lambda_{\max}^k]$ satisfaisant la condition d'Armijo multi-objectifs.

Lemme 4.12. *Soit $x^k \in \mathcal{S}$ un point non Pareto-de descente que. Alors, pour tout $\beta \in (0, 1)$, il existe $\bar{\lambda}^k \in (0, \lambda_{\max}^k]$ tel que, pour tout $\lambda^k \in (0, \bar{\lambda}^k]$,*

$$F(x^k + \lambda^k d^k) \leq F(x^k) + \beta \lambda^k \varphi(x^k). \quad (4.28)$$

Démonstration. Soit $x^k \in \mathcal{S}$ un point non Pareto-de descente que. D'après le Lemme (4.11), on a $\varphi(x^k) < 0$ et $d^k \neq 0$. Pour chaque $i \in \mathcal{P}$, le développement de Taylor d'ordre 1 donne

$$f_i(x^k + \lambda d^k) = f_i(x^k) + \lambda \nabla f_i(x^k)^\top d^k + o(\lambda),$$

avec $\lim_{\lambda \rightarrow 0^+} \frac{o(\lambda)}{\lambda} = 0$. Comme $\nabla f_i(x^k)^\top d^k \leq z(x^k)$ et $z(x^k) = \varphi(x^k) - \frac{1}{2} \|d^k\|^2$, on obtient

$$\beta \varphi(x^k) - \nabla f_i(x^k)^\top d^k \geq (\beta - 1) \varphi(x^k) + \frac{1}{2} \|d^k\|^2.$$

Or $\varphi(x^k) < 0$, $\beta \in (0, 1)$ et $d^k \neq 0$, donc le membre de droite est strictement positif. Par conséquent, il existe $\delta_i > 0$ tel que, pour tout $\lambda \in (0, \delta_i]$,

$$f_i(x^k + \lambda d^k) \leq f_i(x^k) + \beta \lambda \varphi(x^k).$$

En posant

$$\bar{\lambda}^k = \min \left\{ \lambda_{\max}^k, \min_{i \in \mathcal{P}} \delta_i \right\} > 0,$$

on obtient le résultat. \square

À l'itération k , lorsque x^k n'est pas Pareto-de descente que, on calcule d^k par la résolution de SPD (4.21) et λ_{\max}^k par (4.27). Le pas λ^k est ensuite déterminé par une recherche d'Armijo par retour arrière.

Algorithm 7 Détermination de pas λ^k par la règle d'Armijo multi-objectifs

- 1: **Entrées** : choisir $0 < \beta < 1$, $x^k \in \mathcal{S}$, $d^k \in \mathbb{R}^n$.
- 2: **(Pas maximal réalisable)** Calculer λ_{\max}^k par (4.27).
- 3: Fixer $\lambda^k \leftarrow \lambda_{\max}^k$.
- 4: Remplacer $\lambda^k \leftarrow \lambda^k/2$ tant qu'il existe $i \in \mathcal{P}$ tel que

$$f_i(x^k + \lambda^k d^k) > f_i(x^k) + \beta \lambda^k \varphi(x^k).$$

- 5: **Sortie** : retourner λ^k .

On résume maintenant la MDR de Ramdani *et al.* [80].

Algorithm 8 Méthode de Ramdani et al.

- 1: **Entrées** : choisir $0 < \beta < 1$, $x^0 \in \mathcal{S}$, $\varepsilon > 0$.
- 2: Initialiser $k \leftarrow 0$.
- 3: **repeat**
- 4: **(Recherche de direction de descente réalisable)** Résoudre le SPD (4.21) au point x^k et obtenir

$$\varphi_k \leftarrow \varphi(x^k), \quad d^k \leftarrow d(x^k).$$

- 5: **(Test d'arrêt)** Vérifier la condition $|\varphi_k| < \varepsilon$.
- 6: **(Recherche d'un pas λ^k)** Calculer λ_{\max}^k par (4.27).
- 7: Fixer $\lambda^k \leftarrow \lambda_{\max}^k$.
- 8: Remplacer $\lambda^k \leftarrow \lambda^k/2$ tant qu'il existe $i \in \mathcal{P}$ tel que

$$f_i(x^k + \lambda^k d^k) > f_i(x^k) + \beta \lambda^k \varphi(x^k).$$

- 9: **(Mise à jour)** Poser $x^{k+1} \leftarrow x^k + \lambda^k d^k$ et $k \leftarrow k + 1$.
- 10: **until** $|\varphi_k| < \varepsilon$
- 11: **Sortie** : retourner x^k : Point Pareto-de descente que du problème (3.1).

L'algorithme (8) vise à produire un point Pareto-de descente que du problème (3.1). Le théorème suivant établit la convergence vers la Pareto-de descente calité des points d'accumulation.

Théorème 4.3. *Considérons le problème (3.1), où toutes les fonctions objectif f_i sont supposées continûment différentiables et où l'ensemble réalisable \mathcal{S} est supposé être un polyèdre non vide,*

fermé et borné. Soit $\{x^k\}$ la suite générée par l'Algorithme (8). Alors, $\{x^k\}$ admet au moins un point d'accumulation, et tout point d'accumulation est un point Pareto-de descente que du problème (3.1).

Démonstration. Comme $x^k \in \mathcal{S}$ pour tout $k \in \mathbb{N}$ et comme \mathcal{S} est borné, la suite $\{x^k\}$ est bornée. Par le théorème de Bolzano–Weierstrass, une suite bornée dans \mathbb{R}^n admet toujours une sous-suite convergente : il existe un ensemble infini $\mathcal{K} \subset \mathbb{N}$ et un point $x^* \in \mathbb{R}^n$ tels que

$$x^k \rightarrow x^* \quad \text{pour } k \rightarrow +\infty, k \in \mathcal{K}.$$

Comme \mathcal{S} est fermé, on a $x^* \in \mathcal{S}$.

Montrons maintenant que tout point d'accumulation x^* est un point Pareto-de descente que. Supposons, par l'absurde, que x^* ne soit pas un point Pareto-de descente que. D'après la Définition (4.2) et le Corollaire (4.1), il existe une direction de descente réalisable $\bar{d} \in \Phi(x^*)$ telle que

$$\max_{i \in \mathcal{P}} \nabla f_i(x^*)^\top \bar{d} < 0.$$

Posons

$$z := \max_{i \in \mathcal{P}} \nabla f_i(x^*)^\top \bar{d} < 0.$$

Puisque \bar{d} est une direction réalisable en x^* , il existe $\tau > 0$ suffisamment petit tel que $x^* + \tau \bar{d} \in \mathcal{S}$. On pose alors $d_\tau := \tau \bar{d}$. Ainsi, $d_\tau \in \Phi_2(x^*)$.

Considérons la fonction

$$\Gamma(d, x^*) = \frac{1}{2} \|d\|^2 + \max_{i \in \mathcal{P}} \nabla f_i(x^*)^\top d.$$

En évaluant cette fonction en $d = d_\tau = \tau \bar{d}$, on obtient

$$\Gamma(d_\tau, x^*) = \frac{1}{2} \|\tau \bar{d}\|^2 + \max_{i \in \mathcal{P}} \nabla f_i(x^*)^\top (\tau \bar{d}) = \frac{1}{2} \tau^2 \|\bar{d}\|^2 + \tau z.$$

La fonction unidimensionnelle

$$g(\lambda) = \frac{1}{2} \lambda^2 \|\bar{d}\|^2 + \lambda z$$

est strictement convexe et atteint son minimum en

$$\lambda^* = -\frac{z}{\|\bar{d}\|^2} > 0.$$

De plus,

$$g(\lambda^*) = -\frac{z^2}{2\|\bar{d}\|^2} < 0.$$

Ainsi,

$$\varphi(x^*) \leq g(\lambda^*) < 0,$$

ce qui implique $\varphi(x^*) < 0$.

Par continuité de φ (Lemme 4.10), il existe un voisinage de x^* dans lequel $\varphi(x) < 0$. Ainsi, pour tous les itérés x^k suffisamment proches de x^* (avec $k \in \mathcal{K}$), on a $\varphi(x^k) < 0$. L'algorithme ne s'arrête donc pas à ces itérés et continue à produire des directions de descente.

Soit maintenant $\mathcal{K} \subseteq \mathbb{N}$ un ensemble infini tel que $x^k \rightarrow x^*$ pour $k \in \mathcal{K}$. Par le Lemme 4.10,

$$\varphi(x^k) \rightarrow \varphi(x^*) < 0, \quad d^k = d(x^k) \rightarrow d(x^*) \neq 0.$$

Donc, pour $k \in \mathcal{K}$ assez grand, $|\varphi(x^k)| \geq \varepsilon$: l'algorithme ne s'arrête pas et calcule d^k , λ_{\max}^k et λ^k .

Comme $d^k \in \Phi_2(x^k)$, le pas maximal est donné par (4.27) :

$$\lambda_{\max}^k = \min_{j \in \Omega(x^k)} \left\{ \frac{b_j - A_j x^k}{A_j d^k} \right\} > 0.$$

D'après la Proposition 4.2, $x^k + \lambda_{\max}^k d^k \in \mathcal{S}$ et au moins une contrainte devient active. De plus, comme $d^k \rightarrow d(x^*) \neq 0$ et que \mathcal{M} est fini, les ensembles $\Omega(x^k)$ se stabilisent le long de \mathcal{K} . Par continuité de $x \mapsto A_j x$ et de la fonction $d(\cdot)$, on obtient

$$\lambda_{\max}^k \rightarrow \lambda_{\max}(x^*) > 0, \quad k \in \mathcal{K}.$$

D'après le Lemme 4.12, pour tout $\beta \in (0, 1)$, il existe $\bar{\lambda}^k \in (0, \lambda_{\max}^k]$ tel que, pour tout $\lambda^k \in (0, \bar{\lambda}^k]$,

$$f_i(x^k + \lambda^k d^k) \leq f_i(x^k) + \beta \lambda^k \varphi(x^k), \quad \forall i \in \mathcal{P}. \quad (4.29)$$

Comme $\varphi(x^k) < 0$, on a en particulier

$$f_i(x^k + \lambda d^k) < f_i(x^k), \quad \forall i \in \mathcal{P}.$$

Ainsi, la suite $\{F(x^k)\}$ est décroissante composante par composante. Comme \mathcal{S} est borné et les f_i sont continues, $\{F(x^k)\}$ converge vers $F(x^*)$. Donc

$$\lim_{k \rightarrow \infty} F(x^k) = F(x^*),$$

ce qui implique

$$\lim_{k \rightarrow \infty} \|F(x^{k+1}) - F(x^k)\| = 0. \quad (4.30)$$

De l'équation (4.29), on obtient

$$F(x^{k+1}) - F(x^k) \leq \lambda^k \beta \varphi(x^k) \leq 0. \quad (4.31)$$

En combinant (4.30) et (4.31), on obtient

$$\lim_{k \rightarrow \infty} \lambda^k \beta \varphi(x^k) = 0.$$

Comme $\beta > 0$, on en déduit

$$\lim_{k \rightarrow \infty} \lambda^k \varphi(x^k) = 0. \quad (4.32)$$

Puisque $\varphi(x^k)$ converge vers une constante strictement négative, il existe $\delta < 0$ tel que $\varphi(x^k) \leq \delta$ pour $k \in \mathcal{K}$ assez grand. Le terme $\lambda^k \varphi(x^k)$ est donc le produit d'un terme positif $\lambda^k > 0$ par un terme négatif $\varphi(x^k) \leq \delta < 0$. Supposons par l'absurde que

$$\liminf_{k \rightarrow \infty, k \in \mathcal{K}} \lambda^k > 0.$$

Il existe alors $\bar{\lambda} > 0$ tel que $\lambda^k \geq \bar{\lambda}$ pour $k \in \mathcal{K}$ assez grand. Par Armijo,

$$f_i(x^{k+1}) - f_i(x^k) \leq \beta \lambda^k \varphi(x^k) \leq \beta \bar{\lambda} \delta =: c < 0,$$

où $c = \beta \bar{\lambda} \delta$ est une constante fixe (négative, indépendante de k).

Donc, à chaque itération, on a l'inégalité

$$f_i(x^{k+1}) \leq f_i(x^k) + c, \quad c < 0.$$

En sommant successivement depuis le point initial x^0 , on obtient la relation télescopique

$$f_i(x^{k+1}) \leq f_i(x^0) + \sum_{j=0}^k \beta \bar{\lambda} \delta = f_i(x^0) + (k+1)c.$$

Puisque $c < 0$ et $k \rightarrow +\infty$ le long de \mathcal{K} , on a $(k+1)c \rightarrow -\infty$. Ainsi,

$$f_i(x^k) \rightarrow -\infty \quad (k \rightarrow +\infty, k \in \mathcal{K}).$$

Ceci contredit la bornitude inférieure de f_i sur le compact \mathcal{S} . Donc nécessairement

$$\liminf_{k \rightarrow \infty, k \in \mathcal{K}} \lambda^k = 0.$$

Or, toujours d'après le Lemme 4.12 appliqué à x^k , on dispose, pour $k \in \mathcal{K}$ assez grand,

d'un intervalle $(0, \bar{\lambda}_k]$ de pas acceptables, avec $\bar{\lambda}_k$ uniformément minoré (car $\bar{\lambda}_k \leq \lambda_{\max}^k$ et $\lambda_{\max}^k \rightarrow \lambda_{\max}(x^*) > 0$). En particulier, il existe $\eta > 0$ tel que $\bar{\lambda}_k \geq \eta$ pour $k \in \mathcal{K}$ assez grand. Dans ces conditions, la procédure de retour arrière ne peut pas produire une sous-suite $\lambda^k \rightarrow 0$, puisque tout pas $\lambda \in (0, \eta]$ vérifie déjà Armijo. Ceci contredit $\liminf_{k \rightarrow \infty, k \in \mathcal{K}} \lambda^k = 0$.

La contradiction provient de l'hypothèse initiale x^* n'est pas Pareto-de descente que . Par conséquent, x^* est un point Pareto-de descente que du problème (3.1). \square

4.5 Méthodes d'échantillonnage pour générer des populations initiales de points réalisables

Dans cette section, nous rappelons deux procédures d'échantillonnage permettant de générer une population initiale de points réalisables dans le polyèdre $\mathcal{S} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ supposé non vide et borné. Ces populations initiales servent de points de départ à l'exécution des deux méthodes de directions réalisables présentées précédemment, afin de faciliter l'exploration de l'ensemble réalisable et l'approximation des fronts de Pareto.

Nous présentons d'abord la procédure de Morovati et Pourkarimi [71], puis la procédure *Hit-and-Run* de [80] adaptée à l'échantillonnage uniforme sur les ensembles convexes. Une section d'illustration en dimension 2 est ensuite proposée pour comparer le comportement des deux approches.

4.5.1 Méthode d'échantillonnage de Morovati et Pourkarimi [71]

La procédure proposée par Morovati et Pourkarimi [71] construit une population POP de N points réalisables dans un polyèdre $\mathcal{S} \subset \mathbb{R}^n$. Elle démarre d'un point réalisable $x^0 \in \mathcal{S}$, puis explore \mathcal{S} à partir de x^0 selon n_d directions aléatoires d^1, \dots, d^{n_d} (tirées uniformément sur la sphère unité). Pour chaque direction d^k , la méthode détermine une longueur maximale $\lambda_{\max,k} > 0$ telle que $x^0 + \lambda_{\max,k} d^k \in \mathcal{S}$, puis génère des points le long de cette direction avec un pas commun $t > 0$. Lorsque le nombre de points obtenus est inférieur à N , la population est complétée par insertion de points médians.

La recherche de $\lambda_{\max,k}$ repose sur deux facteurs $t_u > 1$ et $t_l \in (0, 1)$. On commence par tester la faisabilité de $y = x^0 + d^k$. Lorsque $y \in \mathcal{S}$, on agrandit le déplacement à l'aide de t_u ; lorsque $y \notin \mathcal{S}$, on réduit le déplacement à l'aide de t_l . Dans les deux cas, on considère les puissances λ_k^q avec $q \in \{0, 1, \dots, Q\}$ (avec $Q = 25$ afin d'obtenir la plus grande valeur réalisable). Enfin, les directions conduisant à une longueur $\lambda_{\max,k}$ trop petite sont rejetées.

Les paramètres t_u et t_l contrôlent respectivement l'agrandissement et la réduction du

déplacement le long d'une direction. Ils vérifient $t_l < 1 < t_u$ et sont fixés dans [71] à $t_u = \frac{3}{2}$ et $t_l = \frac{2}{3}$. Le paramètre Q borne la recherche sur les puissances λ_k^q . À titre indicatif, $(t_l)^Q = (\frac{2}{3})^{25}$ est de l'ordre de 10^{-5} alors que $(t_u)^Q = (\frac{3}{2})^{25}$ est de l'ordre de 10^4 , ce qui fournit un intervalle de longueurs explorées suffisamment large en pratique. La condition $\lambda_{\max,k} > \tau$ (avec $\tau = 10^{-4}$) évite de conserver des directions quasi nulles, qui produiraient des points trop proches de x^0 .

Algorithm 9 Échantillonnage de Morovati–Pourkarimi : génération d'une population de points réalisables

- 1: **Entrées** : choisir la taille de la population $|\text{POP}| = N$, le nombre de directions $n_d \in \mathbb{N}$, une tolérance $\varepsilon > 0$. Fixer $t_u = \frac{3}{2}, t_l = \frac{2}{3}, Q = 25, \tau = 10^{-4}$.
- 2: **Étape 1 : calcul d'un point réalisable initial**. Calculer un point réalisable $x^0 \in \mathcal{S}$.
Dans le cas des contraintes linéaires, cela revient à résoudre :

$$\min_x 0 \quad \text{s.c.} \quad x \in \mathcal{S}.$$

- 3: **Étape 2 : génération des directions**. Générer n_d directions aléatoires unitaires $d^1, \dots, d^{n_d}, \|d^k\| = 1$, à partir du point x^0 .
- 4: **Étape 3 : choix du facteur de déplacement**. Pour chaque direction $d^k, k = 1, \dots, n_d$, poser $y = x^0 + d^k$. Si $y \in \mathcal{S}$, poser $\lambda_k := t_u$. Sinon, poser $\lambda_k := t_l$.
- 5: **Étape 4 : recherche de la longueur maximale réalisable**. Pour chaque direction d^k :
 - si $\lambda_k = t_u$, choisir la plus grande valeur de $q \in \{0, 1, \dots, Q\}$ telle que $x^0 + \lambda_k^q d^k \in \mathcal{S}$;
 - si $\lambda_k = t_l$, choisir la plus petite valeur de $q \in \{0, 1, \dots, Q\}$ telle que $x^0 + \lambda_k^q d^k \in \mathcal{S}$.
 Si $\lambda_k^q > \tau$, poser $\lambda_{\max,k} := \lambda_k^q$. Sinon, rejeter la direction d^k et la remplacer par une nouvelle direction aléatoire.
- 6: **Étape 5 : calcul du pas commun**. Calculer $M = \sum_{k=1}^{n_d} \lambda_{\max,k}, t = \frac{M}{N}$.
- 7: **Étape 6 : génération des points réalisables**. Pour chaque direction d^k , définir

$$\text{POP}(k) = \left\{ x^0 + it d^k \mid i = 1, \dots, \left\lfloor \frac{\lambda_{\max,k}}{t} \right\rfloor, x^0 + it d^k \in \mathcal{S} \right\} \cup \{x^0 + \lambda_{\max,k} d^k\}.$$

- 8: **Étape 7 : construction de la population finale**. Poser

$$\text{POP} = \bigcup_{k=1}^{n_d} \text{POP}(k).$$

- 9: Si $|\text{POP}| < N$, compléter la population par insertion de points médians le long des directions disponibles.
 - 10: **Sortie** : retourner POP.
-

4.5.2 Procédure d'échantillonnage Hit-and-Run de Ramdani et al. [80]

La méthode *Hit-and-Run* appartient à la famille des chaînes de Markov Monte Carlo (MCMC) conçues pour produire un échantillonnage (asymptotiquement) uniforme à l'intérieur d'ensembles bornés. Elle a été introduite par Smith [85] dans le contexte de procédures Monte Carlo sur des ensembles mesurables. Son nom provient de son mécanisme : *Hit* correspond au tirage d'une direction aléatoire à partir du point courant, et *Run* correspond au déplacement le long de la corde faisable maximale dans cette direction, en choisissant un nouveau point uniformément sur cette corde. Des extensions (par exemple Bélisle et al. [11]) permettent de l'adapter à des distributions multivariées sur des corps convexes, et des analyses théoriques (par exemple Lovász [65]) garantissent une convergence rapide (mélange polynomial) sous des hypothèses usuelles. La méthode a également été exploitée pour l'échantillonnage dans des polyèdres en optimisation globale stochastique (voir Zabinsky [96]).

Le schéma de base pour générer une population POP de N points réalisables dans \mathcal{S} s'écrit comme suit.

Algorithm 10 Hit-and-Run pour générer une population POP de N points réalisables dans \mathcal{S}

- 1: *Étape 1* : Trouver un point réalisable initial $x^0 \in \mathcal{S}$, poser $k = 0$ et initialiser $\text{POP} = \{x^0\}$.
- 2: *Étape 2 (Hit)* : Générer une direction aléatoire d^k uniformément distribuée sur la sphère unité $\{d \in \mathbb{R}^n \mid \|d\| = 1\}$.
- 3: *Étape 3 (Run)* : Déterminer l'intervalle réalisable

$$\mathcal{I}(x^k, d^k) = \{\lambda \in \mathbb{R} \mid x^k + \lambda d^k \in \mathcal{S}\}.$$

- 4: Tirer λ_k uniformément sur $\mathcal{I}(x^k, d^k)$, puis calculer $x^{k+1} = x^k + \lambda_k d^k$.
 - 5: Ajouter x^{k+1} à POP, poser $k \leftarrow k + 1$.
 - 6: Terminer lorsque $|\text{POP}| = N$; sinon reprendre à l'étape 2.
-

Dans ce qui suit, nous précisons les améliorations introduites dans [80] à chaque étape afin de favoriser une population mieux répartie et plus proche d'une distribution uniforme dans \mathcal{S} .

1. *Initialisation (Étape 1)*. La méthode de Morovati et Pourkarimi [71] obtient x^0 via un programme linéaire, ce qui conduit souvent à un point situé sur un sommet de \mathcal{S} et peut provoquer un regroupement des échantillons au voisinage des facettes. Pour limiter cet effet, nous choisissons comme point initial le *centre de Chebyshev*

de \mathcal{S} .

Le centre de Chebyshev d'un polyèdre \mathcal{S} est, dans ce contexte, le centre de la plus grande boule euclidienne entièrement contenue dans \mathcal{S} . Autrement dit, le centre de Chebyshev est le point réalisable $x^0 \in \mathcal{S}$ maximisant le rayon $r > 0$ de la plus grande boule euclidienne inscrite dans \mathcal{S} ,

$$B(x^0, r) = \{x \in \mathbb{R}^n \mid \|x - x^0\| \leq r\}.$$

Ce rayon correspond à la distance minimale de x^0 aux hyperplans support des facettes de \mathcal{S} , garantissant que la boule reste entièrement contenue dans le polyèdre. Le couple (x^0, r) est solution de

$$\begin{cases} \max_{x^0, r} r, \\ \text{s.c. } B(x^0, r) \subseteq \mathcal{S}. \end{cases} \quad (4.33)$$

Pour expliciter la contrainte d'inclusion, écrivons $x = x^0 + u$ avec $\|u\| \leq r$. La faisabilité impose

$$A(x^0 + u) \leq b, \quad \forall u \text{ tel que } \|u\| \leq r.$$

Ce qui est équivalent, pour chaque ligne A_j de A (avec $j = 1, \dots, m$),

$$A_j x^0 + \sup_{\|u\|_2 \leq r} (A_j u) \leq b_j.$$

Par l'inégalité de Cauchy-Schwarz, $\sup_{\|u\|_2 \leq r} (A_j u) = r \|A_j\|_2$. Ainsi, (4.33) se reformule en programme linéaire :

$$\begin{cases} \max_{x^0, r} r, \\ \text{s.c. } A_j x^0 + r \|A_j\|_2 \leq b_j, \quad j = 1, \dots, m, \\ x^0 \in \mathbb{R}^n, r \geq 0. \end{cases} \quad (4.34)$$

2. *Génération de direction (Étape 2)*. Une direction uniformément distribuée sur la sphère unité de \mathbb{R}^n peut être obtenue en tirant n variables indépendantes $g_i \sim \mathcal{N}(0, 1)$, $i = 1, \dots, n$, puis en normalisant $g = (g_1, \dots, g_n)$:

$$d^k = \frac{g}{\|g\|}. \quad (4.35)$$

3. *Génération d'un point aléatoire (Étape 3)*. À partir de x^k et d'une direction unitaire d^k , on considère la droite $x^k + \lambda d^k$ et on calcule l'intervalle des λ tels que $x^k + \lambda d^k \in$

\mathcal{S} . Pour $j \in \{1, \dots, m\}$, la contrainte $A_j(x^k + \lambda d^k) \leq b_j$ s'écrit

$$(A_j d^k) \lambda \leq b_j - A_j x^k.$$

Les contraintes avec $A_j d^k = 0$ ne bornent pas λ : elles sont soit satisfaites (si $A_j x^k \leq b_j$), soit elles rendent l'intervalle vide (si $A_j x^k > b_j$). On introduit l'ensemble d'indices $\mathcal{M} = \{1, \dots, m\}$ et l'ensemble actif

$$J(x^k) = \{j \in \mathcal{M} \mid A_j x^k = b_j\}.$$

En notant

$$\Omega^-(x^k) = \{j \in \mathcal{M} \setminus J(x^k) \mid A_j d^k < 0\}, \quad \Omega^+(x^k) = \{j \in \mathcal{M} \setminus J(x^k) \mid A_j d^k > 0\},$$

on obtient les bornes (lorsque la corde est non vide) :

$$\begin{aligned} \lambda_{\min}^k &= \max_{j \in \Omega^-(x^k)} \left\{ \frac{b_j - A_j x^k}{A_j d^k} \right\}, \\ \lambda_{\max}^k &= \min_{j \in \Omega^+(x^k)} \left\{ \frac{b_j - A_j x^k}{A_j d^k} \right\}. \end{aligned} \tag{4.36}$$

L'intervalle réalisable est alors $[\lambda_{\min}^k, \lambda_{\max}^k]$. Les ensembles $\Omega^-(x^k)$ et $\Omega^+(x^k)$ sont typiquement non vides lorsque x^k est dans l'intérieur relatif de \mathcal{S} . Ils peuvent néanmoins être vides si $J(x^k) \neq \emptyset$ et si d^k pointe vers l'extérieur, auquel cas la corde faisable dans cette direction peut être dégénérée ; dans cette situation, on relance le tirage de direction (éventuellement en repartant de x^0) afin d'obtenir un intervalle non vide.

Une fois l'intervalle déterminé, on tire aléatoirement λ^k dans l'intervalle

$$\left[\lambda_{\min}^k, \lambda_{\max}^k \right]$$

et l'on calcule le nouveau point réalisable.

$$x^{k+1} = x^k + \lambda^k d^k.$$

4. *Espacement de la population (Étape 4)*. Afin de limiter la formation de *clusters* et de favoriser une couverture homogène, on impose une distance minimale $\rho > 0$ entre

tout nouveau point x^{k+1} et les points déjà présents dans POP :

$$\|x^{k+1} - \bar{x}\| \geq \rho, \quad \forall \bar{x} \in \text{POP}. \quad (4.37)$$

La valeur de ρ est choisie à partir d'un raisonnement volumique : si les N points étaient uniformément répartis, chacun "occuperait" approximativement un volume moyen

$$\text{Vol}_{\text{moy}} = \frac{\text{Vol}(S)}{N} \quad (4.38)$$

On modélise ce volume moyen par le volume d'une boule euclidienne de dimension n et de rayon ρ . Selon Gipple [48], le volume d'une boule n -dimensionnelle de rayon ρ est

$$\text{Vol}_n(\rho) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)} \rho^n, \quad (4.39)$$

où Γ désigne la fonction gamma.

La fonction gamma est l'extension continue de la factorielle. Pour $z > 0$, elle admet la représentation intégrale d'Euler

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt,$$

et elle satisfait l'équation fonctionnelle

$$\Gamma(z+1) = z\Gamma(z).$$

En particulier, pour tout entier $n \geq 1$, on a $\Gamma(n+1) = n!$ et donc $\Gamma(n) = (n-1)!$. Pour les demi-entiers, on dispose de la forme fermée (pour tout entier $n \geq 0$)

$$\Gamma\left(n + \frac{1}{2}\right) = \frac{(2n-1)!}{2^n} \sqrt{\pi},$$

d'où, par exemple,

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}, \quad \Gamma\left(\frac{3}{2}\right) = \frac{\sqrt{\pi}}{2}, \quad \Gamma\left(\frac{5}{2}\right) = \frac{3\sqrt{\pi}}{4}, \quad \Gamma\left(\frac{7}{2}\right) = \frac{15\sqrt{\pi}}{8}.$$

La présence de la fonction gamma dans (4.39) s'explique par le fait qu'elle fournit une expression *unifiée* des volumes des boules euclidiennes dans toutes les dimensions. En particulier, pour $n = 2$, la formule (4.39) donne l'aire d'un disque de rayon ρ :

$$\text{Vol}_2(\rho) = \frac{\pi^1}{\Gamma(2)} \rho^2 = \frac{\pi}{1!} \rho^2 = \pi \rho^2,$$

car $\Gamma(2) = 1$. Pour $n = 3$, elle fournit le volume d'une boule de rayon ρ :

$$\text{Vol}_3(\rho) = \frac{\pi^{3/2}}{\Gamma(\frac{5}{2})} \rho^3 = \frac{\pi^{3/2}}{\frac{3\sqrt{\pi}}{4}} \rho^3 = \frac{4}{3} \pi \rho^3,$$

où l'on utilise $\Gamma(\frac{5}{2}) = \frac{3\sqrt{\pi}}{4}$.

En égalisant le volume moyen et le volume de la boule, comme suit :

$$\text{Vol}_{\text{moy}} = \frac{\mathcal{S}}{N} \approx \text{Vol}_n(\rho) \quad (4.40)$$

on obtient alors

$$\rho = \left(\frac{\text{Vol}(\mathcal{S}) \Gamma(\frac{n}{2} + 1)}{N \pi^{n/2}} \right)^{1/n}. \quad (4.41)$$

Le calcul exact du volume du polyèdre \mathcal{S} est *NP-difficile* en général, et devient très coûteux en dimension élevée ($n \geq 5$) [33]. En dimension 2, une méthode exacte simple consiste à énumérer les sommets, à les ordonner dans le sens trigonométrique, puis à appliquer la *shoelace formula* [61] :

$$\text{Vol}(\mathcal{S}) = \frac{1}{2} \left| \sum_{i=1}^{N_v} (x_i y_{i+1} - x_{i+1} y_i) \right|, \quad (4.42)$$

où N_v désigne le nombre de sommets et $(x_{N_v+1}, y_{N_v+1}) = (x_1, y_1)$.

Plus généralement, les approches de calcul de volume $\text{Vol}(\mathcal{S})$ se répartissent en méthodes exactes tel que (méthode récursive de *Lasserre* [62], méthode de *Lawrence* [63]) et méthodes d'approximation. Les méthodes exactes sont efficaces en faible dimension ou pour des formes particulières, mais deviennent rapidement difficiles lorsque la dimension augmente à cause de la complexité combinatoire (énumération des sommets/facettes), elle-même NP-difficile [33].

Pour des dimensions même modérées ($n \geq 5$), on privilégie souvent des d'approximation de type *Monte-Carlo*. Une approche classique consiste à échantillonner uniformément dans une boîte englobante $B(\mathcal{S})$ et à estimer

$$\text{Vol}(\mathcal{S}) \approx \frac{G_S}{G} \text{Vol}(B(\mathcal{S})),$$

où G est le nombre total de points tirés dans $B(\mathcal{S})$ et G_S le nombre de points appartenant à \mathcal{S} [74]. Lorsque la boîte englobante n'est pas disponible, ses bornes LB_i et UB_i peuvent être obtenues en résolvant $2n$ programmes linéaires (minimi-

sation/maximisation de chaque variable x_i sur \mathcal{S}). Alternativement, l'algorithme de *reverse search* d'Avis et Fukuda [7] permet d'énumérer les sommets dans certains cas, puis la projection sur chaque axe fournit directement les bornes. Le volume de la boîte s'obtient alors par

$$\text{Vol}(B(\mathcal{S})) = \prod_{i=1}^n (\text{UB}_i - \text{LB}_i).$$

Enfin, lorsque l'ajout de nouveaux points satisfaisant la contrainte (4.37) devient difficile (par saturation géométrique), on ajuste ρ à la baisse via

$$\rho_{\text{new}} = \vartheta \rho, \quad \vartheta \in (0, 1],$$

où ϑ est un paramètre de réduction fixé de manière à conserver un compromis entre densité et régularité spatiale.

4.5.3 Exemple d'illustration

Pour illustrer le fonctionnement des deux procédures d'échantillonnage destinées à générer des populations de points réalisables, considérons le polyèdre suivant :

$$\mathcal{S} = \left\{ \begin{array}{l} \left(\begin{array}{l} x_1 \\ x_2 \end{array} \right) \in \mathbb{R}^2 \\ \left. \begin{array}{l} x_1 + 6x_2 \leq 39, \\ x_1 + 2x_2 \leq 15, \\ 5x_1 + x_2 \leq 30, \\ 3x_1 + x_2 \geq 3, \\ x_1 \geq 0, \\ x_2 \geq 0 \end{array} \right\}. \end{array} \right. \quad (4.43)$$

Les contraintes sont numérotées dans l'ordre A_1, A_2, \dots, A_6 . L'ensemble réalisable est le polyèdre délimité par les six points extrêmes $a^1, a^2, a^3, a^4, a^5, a^6$, tel qu'il est lustré à la figure (4.5).

1. Illustration de la méthode de Morovati et Pourkarimi [71].

Nous appliquons l'échantillonnage de Morovati et Pourkarimi [71] afin de générer deux points réalisables.

Étape 1 : Initialiser $N = 2$, $n_d = 2$, $t_u = \frac{3}{2}$ et $t_l = \frac{2}{3}$.

Étape 2 : Résoudre le programme linéaire

$$\min_x 0 \quad \text{s.c.} \quad x \in \mathcal{S}$$

La résolution de ce dernier donne le sommet $a^1 = (0, 3)^\top$ (voir la figure).

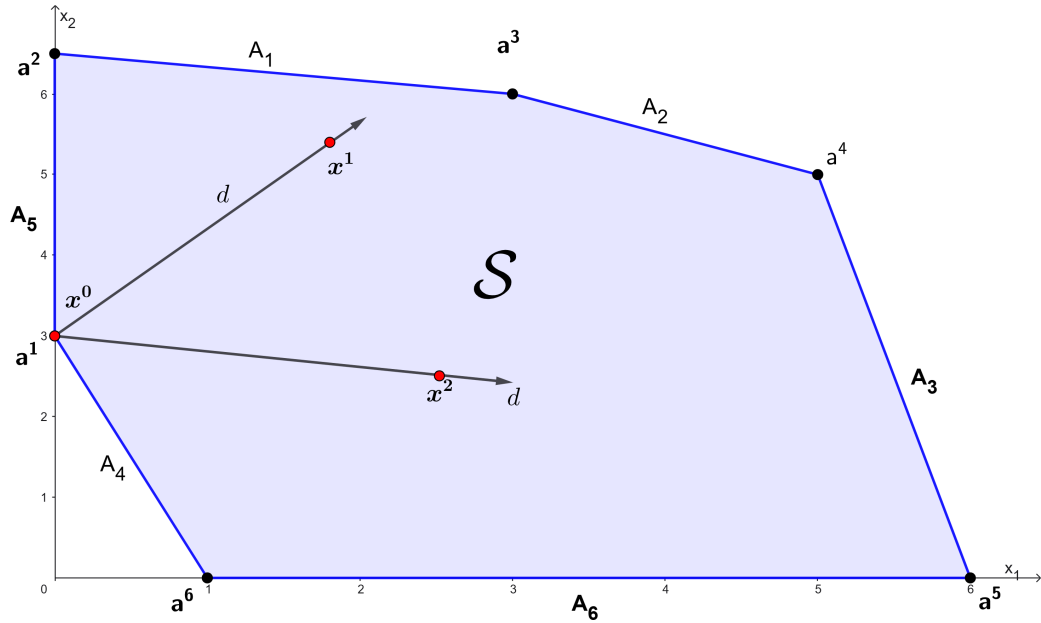


FIGURE 4.4 – Interprétation géométrique de la méthode d'échantillonnage de Morovati et Pourkarimi [71].

Étape 3 : Générer $n_d = 2$ directions aléatoires, uniformément distribuées puis normalisées. Supposons que le générateur fournisse $d^1 \simeq (0.6, 0.8)^\top$ et $d^2 \simeq (-0.4, 0.9)^\top$. Pour chaque direction, on calcule $y = x^0 + d^k$ et l'on décide si l'on prend t_u ou t_l . Par exemple, pour d^1 ,

$$y = x^0 + d^1 = (0, 3)^\top + (0.6, 0.8)^\top = (0.6, 3.8)^\top.$$

On vérifie que $y \in \mathcal{S}$ (toutes les contraintes sont satisfaites), donc $\lambda_1 = t_u = \frac{3}{2}$. En revanche, pour d^2 , on obtient $x^0 + d^2 = (-0.4, 3.9)^\top \notin \mathcal{S}$ (car $x_1 \geq 0$ est violée), donc $\lambda_2 = t_l = \frac{2}{3}$.

Étape 4 : Pour chaque d^k , déterminer une puissance $q \in \{0, \dots, 25\}$ telle que $y = x^0 + \lambda_k^q d^k \in \mathcal{S}$, et retenir la plus grande longueur de pas réalisable $\lambda_{\max, k} = \lambda_k^q$ (après tests de faisabilité). Supposons que l'on obtienne $\lambda_{\max, 1} \simeq 4.2$ et $\lambda_{\max, 2} \simeq 1.8$.

Étape 5 : Calculer $M = \lambda_{\max, 1} + \lambda_{\max, 2} \simeq 6$, puis le pas

$$t = \frac{M}{N} \simeq 3.$$

Étape 6 : Générer des points réalisables le long de chaque direction. Pour d^1 ,

$$x^0 + td^1 \simeq (0, 3)^\top + 3(0.6, 0.8)^\top = (1.8, 5.4)^\top,$$

et

$$x^0 + \lambda_{\max,1} d^1 \simeq (0, 3)^\top + 4.2(0.6, 0.8)^\top = (2.52, 6.36)^\top,$$

en vérifiant à chaque fois la faisabilité. De manière analogue, on procède pour d^2 . Finalement, on obtient une population de taille $N = 2$, par exemple $p_1 \simeq (1.8, 5.4)^\top$ et $p_2 \simeq (2.52, 6.36)^\top$.

2. Illustration de la méthode Hit-and-Run.

(1) Les normes euclidiennes des lignes de contraintes A_j (calculées par $\|A_j\|_2 = \sqrt{A_{j1}^2 + A_{j2}^2}$) sont :

$$\begin{aligned} \|A_1\|_2 &= \sqrt{1^2 + 6^2} = \sqrt{37}, & \|A_2\|_2 &= \sqrt{1^2 + 2^2} = \sqrt{5}, \\ \|A_3\|_2 &= \sqrt{5^2 + 1^2} = \sqrt{26}, & \|A_4\|_2 &= \sqrt{(-3)^2 + (-1)^2} = \sqrt{10}, \\ \|A_5\|_2 &= 1, & \|A_6\|_2 &= 1. \end{aligned}$$

Le problème du centre et du rayon de Chebyshev se reformule en le programme linéaire :

$$\left\{ \begin{array}{l} \max_{x_1^0, x_2^0, r} \quad r \\ \text{s.c.} \quad x_1^0 + 6x_2^0 + \sqrt{37}r \leq 39, \\ \quad \quad x_1^0 + 2x_2^0 + \sqrt{5}r \leq 15, \\ \quad \quad 5x_1^0 + x_2^0 + \sqrt{26}r \leq 30, \\ \quad \quad -3x_1^0 - x_2^0 + \sqrt{10}r \leq -3, \\ \quad \quad -x_1^0 + r \leq 0, \\ \quad \quad -x_2^0 + r \leq 0, \\ \quad \quad r \geq 0, \quad x_1^0, x_2^0 \in \mathbb{R}. \end{array} \right. \quad (4.44)$$

La résolution fournit le centre de Chebyshev $x^0 \simeq (2.6834, 3.2306)^\top$ et le rayon $r \simeq 2.6186$. Ce rayon correspond à la distance minimale de x^0 aux hyperplans des facettes actives, garantissant que la boule $B(x^0, r)$ est incluse dans \mathcal{S} (figure (4.5)).

(2) Une direction est générée en tirant un vecteur gaussien $d \sim \mathcal{N}(0, 1)$ puis en le normalisant. Supposons que l'on obtienne $d = (-0.7247, 1.0936)^\top$, d'où la direction unitaire $d \simeq (-0.5524, 0.8336)^\top$.

(3) Les ensembles d'indices sont déterminés par le signe de $A_j d$:

$$\Omega^-(x^0) = \{j \mid A_j d < 0\} = \{3, 6\}, \quad \Omega^+(x^0) = \{j \mid A_j d > 0\} = \{1, 2, 4, 5\}.$$

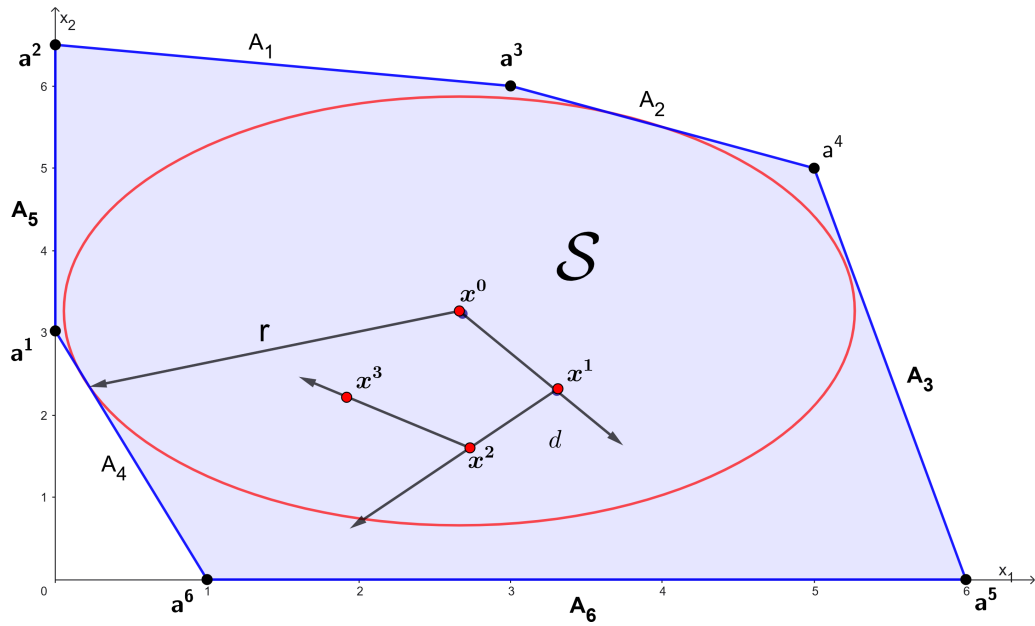


FIGURE 4.5 – Interprétation géométrique de la méthode Hit-and-Run.

Les bornes du segment réalisable dans la direction d depuis x^0 sont alors :

$$\lambda_{\min} = \max_{j \in \{3,6\}} \frac{b_j - A_j x^0}{A_j d} \simeq -3.8755, \quad \lambda_{\max} = \min_{j \in \{1,2,4,5\}} \frac{b_j - A_j x^0}{A_j d} \simeq 3.8059.$$

Ainsi, l'intervalle faisable est approximativement $[\lambda_{\min}, \lambda_{\max}] = [-3.8755, 3.8059]$. Un tirage uniforme dans cet intervalle, par exemple $\lambda = -1.2443$, conduit au nouveau point

$$x^1 = x^0 + \lambda d \simeq (3.3708, 2.1933)^\top.$$

- (4) Le point x^1 sert ensuite de point courant pour générer x^2 : on tire une nouvelle direction unitaire, on calcule les bornes $[\lambda_{\min}, \lambda_{\max}]$ associées, puis on échantillonne λ uniformément dans cet intervalle pour poser $x^2 = x^1 + \lambda d$. La même procédure est répétée itérativement.
- (5) Pour imposer un espacement minimal entre les points, on s'appuie sur le volume (ici l'aire) $\text{Vol}(S)$ afin d'estimer une distance caractéristique. Dans cet exemple (2D), l'aire peut être calculée exactement à l'aide de la *shoelace formula* [61], en ordonnant les six sommets dans le sens trigonométrique (deux sommets consécutifs étant adjacents), par exemple :

$$(0, 3) \rightarrow \left(0, \frac{13}{2}\right) \rightarrow (3, 6) \rightarrow (5, 5) \rightarrow (6, 0) \rightarrow (1, 0) \rightarrow (0, 3).$$

En appliquant (4.42), on obtient :

$$\begin{aligned}
 S &= \sum_{i=1}^6 (x_i y_{i+1} - x_{i+1} y_i) \\
 &= \left(0 \cdot \frac{13}{2} - 0 \cdot 3\right) + \left(0 \cdot 6 - 3 \cdot \frac{13}{2}\right) + (3 \cdot 5 - 5 \cdot 6) + (5 \cdot 0 - 6 \cdot 5) + (6 \cdot 0 - 1 \cdot 0) + (1 \cdot 3 - 0 \cdot 0) \\
 &= -61.5.
 \end{aligned}$$

Ainsi,

$$\text{Vol}(\mathcal{S}) = \frac{1}{2}|S| = \frac{1}{2}61.5 = 30,75.$$

Cependant, même en dimension 2, on peut aussi estimer le volume (aire) par une procédure *Monte Carlo* (Ong et al. [74]) : on détermine d'abord une boîte englobante $[LB, UB]$, on génère un grand nombre de points (par exemple $G = 100\,000$) dans cette boîte, puis on calcule le sous-ensemble G_S des points appartenant à \mathcal{S} .

4.6 Conclusion

Dans ce chapitre, nous avons présenté une méthode de directions réalisables adaptée aux problèmes d'optimisation multi-objectifs sous contraintes linéaires. Notre première contribution consiste à reformuler le sous-problème de recherche de direction en introduisant un cône réduit de directions réalisables, permettant de tenir compte de l'ensemble des contraintes du polyèdre.

Cette reformulation améliore la stabilité des directions calculées et favorise une progression plus efficace vers des points Pareto-stationnaire. Notre seconde contribution concerne l'introduction d'une procédure d'échantillonnage de type Hit-and-Run, destinée à générer une population initiale mieux répartie dans l'ensemble réalisable.

5

Étude expérimentale et comparative

Introduction

Ce chapitre présente une étude expérimentale et comparative menée sous MATLAB R2022a. Elle vise à évaluer l'efficacité de la Méthode des Directions réalisables (*MDR*) proposée, couplée à une procédure d'échantillonnage *Hit-and-Run* pour générer une population initiale de points réalisables dans le polyèdre. Les performances obtenues sont comparées à celles (i) de la méthode de *Morovati et Pourkarimi* [71], qui s'appuie sur un sous-problème différent pour la recherche d'une direction réalisable (*SPD*) ainsi que sur une autre procédure d'échantillonnage pour la génération d'une population de points réalisables dans le polyèdre, et (ii) d'un algorithme évolutionnaire de référence reconnu pour ses performances, *NSGA-II* [31].

Les objectifs de cette étude expérimentale et comparative sont fixés comme suit :

- (O1) *Visualiser la qualité géométrique de la population initiale de points réalisables, générée par Hit-and-Run, en comparaison avec la procédure de Morovati et Pourkarimi [71]. Cet objectif consiste à visualiser la répartition spatiale des points dans le polyèdre de 2D et 3D au regard de trois critères : (i) la dispersion, (ii) la couverture de l'ensemble réalisable et (iii) l'absence de regroupements locaux.*
- (O2) *Évaluer l'apport de la modification du sous-problème de recherche de direction*

(SPD). Comparativement à la méthode de Morovati et Pourkarimi [71], cet objectif consiste à analyser l'effet du SPD modifié sur : (i) la capacité à résoudre l'ensemble des problèmes testés, c'est-à-dire la faculté de l'algorithme à produire une approximation du front de Pareto exploitable, sans échec numérique ni stagnation, et (ii) la rapidité de convergence, appréciée au moyen du nombre d'itérations nécessaires pour déterminer un point Pareto-de descente que et du temps d'exécution CPU correspondant.

- (O3) *Évaluer la qualité des fronts de Pareto approximés à l'aide d'indicateurs de performance.* Cet objectif consiste à comparer les fronts produits par les trois méthodes étudiées au moyen des métriques suivantes :
 - *Pureté* : évalue la proportion de solutions non dominées du front considéré par rapport à un front de référence global.
 - *Hypervolume* : mesure le volume de l'espace objectif dominé, reflétant la couverture globale du front.
 - *Spacing* : calcule la régularité de l'espacement entre solutions non dominées, indiquant l'uniformité de la distribution.
 - *Generalized spread* : évalue la diversité et la capacité à représenter les zones extrêmes et intermédiaires du front.

5.1 Description de la liste des problèmes testés

L'étude expérimentale et comparative a été conduite intégralement sous MATLAB R2022a. Les trois approches évaluées — (i) la méthode proposée MDR couplée à la procédure d'échantillonnage *Hit-and-Run*, (ii) la méthode de Morovati et Pourkarimi [71], et (iii) l'algorithme évolutionnaire NSGA-II [31] — ont été implémentées dans un même environnement logiciel et soumises à des conditions d'exécution homogènes, de manière à assurer une comparaison rigoureuse, équitable et reproductible.

Les expériences portent sur un ensemble de 25 problèmes *LC-MOO*. Cet ensemble a été constitué de manière à couvrir un spectre large de configurations rencontrées dans la littérature :

- différentes dimensions (nombre de variables n) et tailles (nombre de contraintes m);
- plusieurs cardinalités d'objectifs (nombre de fonctions p);
- des objectifs de nature variée (linéaires, quadratiques et non linéaires);
- des propriétés géométriques contrastées (problèmes convexes et non convexes).

Cette diversité vise à tester la robustesse des méthodes, leur capacité à traiter des ins-

tances hétérogènes, ainsi que l’impact (i) de la stratégie d’échantillonnage initial et (ii) du *SPD* sur la stabilité numérique et l’efficacité globale.

La liste complète des problèmes tests, implémentée en MATLAB et directement intégrable dans nos scripts d’expérimentation, est mise à disposition sur *GitHub* dans le fichier suivant :

https://github.com/Ramdani-sudo/MFD-for-MOQP/blob/main/test_problems.m

TABLE 5.1 – Résumé des problèmes tests et de leurs caractéristiques

Problèmes	Référence	n	m	p	LB	UB	Type d’objectifs	Convexité
HANNE3	[27]	2	3	2	[0,0]	[10,10]	Non linéaire	Non convexe
CONSTR	[38]	2	6	2	[0.1,0]	[1,5]	Non linéaire	Convexe
KITA	[38]	2	5	2	[0,0]	–	Quadratique	Convexe
EX004	[38]	2	5	2	[0,0]	[2,6]	Non linéaire	Convexe
KM	[38]	2	5	2	[0,0]	–	Non linéaire	Convexe
EXEMPLE 7.1	[4]	2	4	2	[0,-2]	[1,2]	Non linéaire	Non convexe
EXEMPLE 7.0	[4]	2	4	2	[0,-2]	[1,2]	Non linéaire	Non convexe
ABC-COMP	[54]	2	5	2	[0,0]	–	Quadratique	Non convexe
BEEME	[54]	2	4	2	[-1,1]	[2,2]	Non linéaire	Non convexe
COMET	[36]	3	6	3	[0,-2,1]	[3.5,2,1]	Non linéaire	Non convexe
SW1	[38]	3	6	3	[0,0,0]	–	Linéaire	Convexe
SW2	[38]	3	6	3	[0,0,0]	–	Linéaire	Convexe
PEKKA2	[38]	3	6	3	[0,0,0]	–	Linéaire	Convexe
EXEMPLE 2.1	[3]	2	6	2	[0,0]	–	Linéaire	Convexe
LISWETM	[38]	7	5	2	[0,0,...,0]	[10,10,...,10]	Quadratique	Convexe
NABETANI2008	[88]	2	3	2	[0,0]	–	Non linéaire	Non convexe
KATHRIN2008	[58]	2	5	5	[0,0]	–	Non linéaire	Convexe
WIECEK2001	[88]	2	5	2	[0,0]	–	Non linéaire	Non convexe
YOUNESS2004	[88]	2	4	2	[0,0]	–	Non linéaire	Non convexe
MOLPG-1	[87]	8	16	3	[0,0,...,0]	–	Linéaire	Convexe
MOLPG-2	[87]	12	28	3	[0,0,...,0]	–	Linéaire	Convexe
MOLPG-3	[87]	10	28	3	[0,0,...,0]	–	Linéaire	Convexe
MOQP-1	[64]	20	30	3	[0,0,...,0]	[10,10,...,10]	Quadratique	Convexe
MOQP-2	[64]	20	30	3	[0,0,...,0]	[10,10,...,10]	Quadratique	Convexe
MOQP-3	[64]	20	30	3	[0,0,...,0]	[10,10,...,10]	Quadratique	Convexe

Le Tableau (5.1) synthétise les 25 problèmes tests en précisant leur structure et leurs principales caractéristiques. La colonne « Référence » indique la source de chaque instance dans la littérature, ce qui assure la traçabilité des formulations et des paramètres. Les colonnes n et m indiquent respectivement le nombre de variables de décision et de contraintes : les instances les plus grandes en dimension sont MOQP-1 à MOQP-3 (jusqu’à $n = 20$), et le plus grand nombre de contraintes apparaît pour la famille MOQP ($m = 30$). La colonne p donne le nombre d’objectifs, le problème KATHRIN2008 se distinguant par un cas à cinq objectifs ($p = 5$). Les bornes $[LB, UB]$ définissent une boîte englobante du polyèdre, utile notamment pour l’estimation de son volume dans la procédure Hit-and-Run ; lorsque UB n’est pas fourni (noté “–”), il doit être déterminé numériquement. Enfin, les colonnes « Type d’objectifs » et « Convexité » mettent en évidence la diversité des fonctions (linéaires, quadratiques, non linéaires) et des géométries (convexes et non

convexes), permettant une évaluation comparative sur des cas de complexité variée.

5.2 Description des codes en MATLAB

L'étude expérimentale et comparative repose sur un ensemble de scripts et de fonctions MATLAB, organisés de manière modulaire. L'objectif est (i) d'exécuter, sur une même liste de problèmes, les trois approches considérées, (ii) de collecter des sorties directement comparables (fronts de Pareto approchés, temps CPU, nombre d'itérations), et (iii) d'évaluer la qualité des fronts obtenus au moyen des indicateurs de performance retenus.

L'implémentation s'appuie sur le *Multi-Parametric Toolbox 3* (MPT3), présenté dans la section suivante, pour la manipulation des polyèdres (tests de faisabilité, calcul du centre de Chebyshev, détermination d'une boîte englobante, etc.).

Le programme associé à cette étude contient cinq scripts `.m`, dont les principaux sont les suivants :

- `test_problems.m` : bibliothèque des 25 problèmes tests présentés précédemment.
- `main_function.m` : script de pilotage des expériences (fonction principale). Il charge les problèmes depuis `test_problems.m`, fixe les paramètres communs (taille de population, tolérances, itérations maximales, paramètres de recherche linéaire), puis exécute successivement les trois méthodes de résolution, à savoir :
 - `MDR_Ramdani_et_al.m` : implémentation principale de notre méthode *MDR* couplée à l'échantillonnage *Hit-and-Run*, et `MDR_Morovati_Pourkarimi.m` : implémentation de la méthode de direction de Morovati et Pourkarimi, incluant leur procédure d'échantillonnage (basée sur un point initial obtenu par un programme linéaire et une exploration directionnelle) ainsi que leur sous-problème de recherche de direction (*SPD*).
 - `nsgaII_linear_constraints.m` : implémentation de NSGA-II adaptée au cas de contraintes linéaires, en s'appuyant sur un code open-source développé par Persson [75]. Ce module gère la génération de la population, la sélection par tournoi, le croisement, la mutation, le tri non dominé (*non-dominated sorting*), ainsi qu'un mécanisme de traitement des contraintes (vérification de faisabilité et/ou pénalisation selon la configuration).
- `Comparaison_Methods.m` : module d'évaluation et de comparaison des fronts. À partir des fronts produits par les trois méthodes, il construit un front global de référence, normalise les points en espace objectif, calcule les métriques de qualité (pureté, hypervolume, spacing, generalized spread), puis génère des visualisations (fronts filtrés et histogrammes comparatifs).

Le déroulement computationnel de l'étude expérimentale et comparative est piloté par `main_function.m`. Pour chaque problème test, ce script charge les fonctions objectifs, les données de contraintes (A, b) ainsi que les bornes disponibles, puis exécute successivement les trois méthodes en imposant une configuration homogène (taille de population, tolérances et nombre maximal d'itérations, lorsque cela est compatible avec la logique propre à chaque algorithme). Les trois fronts obtenus sont ensuite transmis à `Comparison_Methods.m` pour le calcul des indicateurs de performance et la génération des figures en dimension 2 et 3.

L'ensemble des scripts MATLAB, ainsi que les fonctions utilisées pour la comparaison des méthodes, sont mis à disposition sur [GitHub](https://github.com/Ramdani-sudo/MFD-for-MOOP)¹, tracer et reproduire précisément les expériences numériques :

<https://github.com/Ramdani-sudo/MFD-for-MOOP.git>

Les expériences ont été conduites avec une configuration commune visant à garantir une comparaison équitable entre les méthodes. Les paramètres utilisés sont récapitulés ci-dessous.

- Taille de la population initiale (points réalisables). La population initiale est fixée à $N = 200$ points réalisables dans l'ensemble réalisable.
- Paramètres communs aux deux méthodes de type *MDR* (notre *MDR* et *MDR-MP*).
 - Multiplicateurs de pas : $t_u = 3/2$ (augmentation) et $t_l = 2/3$ (réduction).
 - Exposant de pas : $q \in \{0, 1, \dots, 25\}$, avec bornes de pas $[10^{-5}, 10^5]$.
 - Nombre de directions aléatoires : $n_d = 150$.
 - Tolérance d'activation des contraintes : $\epsilon_1 = 10^{-5}$.
 - Précision de convergence : $\epsilon_2 = 10^{-6}$.
 - Paramètre d'Armijo (recherche linéaire) : $\sigma = 0.5$.
 - Nombre maximal d'itérations : `max_iter` = 500.
 - Critère d'arrêt : l'algorithme s'arrête lorsque la mesure de descente citée de Pareto satisfait $|\varphi(x_k)| \leq \epsilon_2$.
- Paramètre spécifique à l'échantillonnage *Hit-and-Run* (notre *MDR*). Le facteur de réduction associé à la contrainte de séparation des points est fixé à $\theta = 0.75$.
- Paramètres de NSGA-II.
 - Nombre maximal de générations : `genMax` = 1000.
 - Taille du tournoi : `kTour` = 2.

1. GitHub est une plateforme d'hébergement et de gestion de code basée sur le système de contrôle de versions Git, fondée en 2008 par Tom Preston-Werner, Chris Wanstrath, PJ Hyett et Scott Chacon. Elle permet de collaborer sur des projets, de suivre les modifications du code et de gérer différentes versions de manière structurée. Aujourd'hui largement utilisée dans le développement logiciel et la recherche, la plateforme est accessible à l'adresse suivante : <https://github.com>.

- Probabilité initiale de mutation : $n_p = 0.2$.

5.3 Intégration de la toolbox MPT3

Pour les opérations géométriques sur les polyèdres faisables, nos scripts s'appuient sur le *Multi-Parametric Toolbox 3* (MPT3), une boîte à outils open-source sous MATLAB dédiée à l'optimisation paramétrique et à la géométrie computationnelle des ensembles convexes. Développée notamment par *Kvasnica et Jones*, et documentée dans [52], MPT3 est distribuée publiquement et peut être obtenue via le site officiel : <https://www.mpt3.org/>. Dans cette étude, la classe `Polyhedron` est mobilisée pour construire le polyèdre \mathcal{S} , vérifier des propriétés structurelles (non-vacuité, bornitude), calculer des objets géométriques utiles (boîte englobante, centre de Chebyshev) et visualiser les ensembles en dimension 2D/3D.

- Déclaration d'un polyèdre : Le polyèdre est défini à partir de contraintes linéaires (inégalités et, le cas échéant, égalités) via la classe `Polyhedron`. Une écriture standard est : `P = Polyhedron('A',A,'b',b,'Ae',Ae,'be',be)` ;
La structure `P` encapsule la description de \mathcal{S} et fournit des méthodes pour les calculs géométriques requis par l'algorithme.
- Détermination d'une boîte englobante : Les bornes `LB` et `UB` sont utiles pour construire une boîte englobante et, plus généralement, pour des opérations d'approximation (par exemple, l'estimation de volume et certains diagnostics numériques liés à l'échantillonnage). Lorsque ces bornes ne sont pas fournies par le problème, elles peuvent être obtenues à partir d'une approximation externe : `bbox = P.outerApprox()` ;
`LB = bbox.Internal.lb` ; `UB = bbox.Internal.ub` ;
Si la représentation par sommets est disponible, une alternative consiste à exploiter `P.V` : `LB = min(P.V,[],1)` ; `UB = max(P.V,[],1)` ;.
- Centre de Chebyshev et rayon inscrit : Pour initialiser certaines procédures d'échantillonnage et améliorer la stabilité géométrique (point initial éloigné des facettes), nous calculons le centre de Chebyshev et le rayon de la plus grande boule euclidienne inscrite : `data = P.chebyCenter()` ;
`x0 = data.x` ; `r0 = data.r` ;.
- Tests de validité et requêtes d'appartenance : Plusieurs vérifications sont utilisées afin de sécuriser l'exécution des routines d'échantillonnage et de génération : `P.isEmptySet()` pour tester la non-vacuité, `P.isBounded()` pour vérifier la bornitude, et `P.contains(x)` pour tester si un point x appartient à \mathcal{S} .
- Visualisation en 2D et 3D : Pour l'analyse qualitative des populations générées et

de la géométrie de l'ensemble faisable, la visualisation est réalisée via `P.plot`, qui s'adapte automatiquement à la dimension (2D/3D). Par exemple, un rendu en fil de fer peut être obtenu par : `P.plot('wire', true, 'linewidth', 2);`

5.4 Les métriques d'évaluation de Front Pareto

Dans ce qui suit, nous présentons les quatre métriques retenues pour évaluer la qualité des fronts de Pareto générés par les trois méthodes, en termes de convergence, de couverture, d'uniformité et de diversité. Ces métriques constituent un cadre d'évaluation complet permettant de réaliser des comparaisons quantitatives entre les trois approches d'optimisation appliquées à des problèmes *LC-MOO*. Plus précisément, la *convergence* décrit la proximité du front approché par rapport à l'ensemble Pareto-optimal réel; la *couverture* quantifie l'étendue de l'espace objectif dominé par le front; l'*uniformité* mesure la régularité de la distribution des solutions le long du front; et la *diversité* reflète la capacité du front à représenter l'ensemble des compromis, incluant les zones extrêmes et intermédiaires.

Soient P_1 , P_2 et P_3 les fronts de Pareto produits respectivement par notre méthode *MDR*, par la méthode de Morovati et Pourkarimi *MDR-MP*, et par *NSGA-II*. Dans la majorité des cas pratiques, le front de Pareto *vrai* n'est pas disponible, notamment en raison du coût computationnel associé à l'identification exacte de l'ensemble optimal. Pour cette raison, nous construisons un front de référence global

$$P_G = P_1 \cup P_2 \cup P_3,$$

qui regroupe l'ensemble des solutions candidates obtenues par les trois méthodes. Une étape de filtrage est ensuite appliquée pour supprimer les points dominés, via des comparaisons d'ordre partiel par paires. Le front P_G ainsi obtenu sert de front vrai et permet d'effectuer des évaluations de performance relatives de manière cohérente.

Chaque front $P_k = \{y_1, \dots, y_{N_k}\}$, de taille N_k , est ensuite normalisé dans l'hypercube unité $[0, 1]^p$, où p désigne le nombre d'objectifs. Cette normalisation est essentielle afin de limiter les biais dus à des échelles objectives hétérogènes, et d'éviter qu'un objectif numériquement dominant n'influence de manière disproportionnée les métriques. La normalisation est réalisée composante par composante, en utilisant les bornes globales observées sur P_G :

$$y'_{ij} = \frac{y_{ij} - \min_{y \in P_G} y_j}{\max_{y \in P_G} y_j - \min_{y \in P_G} y_j}, \quad \forall i = 1, \dots, N_k, \quad j = 1, \dots, p. \quad (5.1)$$

Ici, y_{ij} est la j -ième valeur objective du i -ième point du front P_k , tandis que les minima et maxima sont calculés sur l'ensemble des points du front de référence global P_G . Cette mise à l'échelle min–max projette tous les objectifs sur une plage commune, ce qui favorise une comparaison équitable entre fronts.

Les métriques de performance sont définies ci-après. Chacune est conçue pour capturer un aspect particulier de la qualité d'un front de Pareto approché.

1. **Pureté.** La métrique de pureté, notée $\text{Purity}(P_k)$, quantifie la convergence en mesurant la proportion de points de P_k qui demeurent non dominés lorsque l'on considère le front de référence P_G :

$$\text{Purity}(P_k) = \frac{|\{y \in P_k \mid y \text{ est non dominé dans } P_G\}|}{|P_k|}. \quad (5.2)$$

Cette métrique peut être interprétée comme un "taux de survie" des solutions de P_k face à la concurrence induite par la fusion des trois fronts. Un point est dit non dominé si aucun autre point de P_G ne le domine strictement. Une valeur proche de 1 indique que la méthode produit des solutions de haute qualité, robustes au sens où elles restent compétitives dans le front combiné, traduisant une forte convergence vers le front de Pareto. Comme discuté dans [6], la pureté est particulièrement adaptée aux contextes où le front vrai n'est pas connu, car elle fournit une mesure relative directement exploitable.

2. **Hypervolume.** L'hypervolume, noté $\text{HV}(P_k)$, évalue la couverture en calculant le volume de l'espace objectif dominé par le front P_k relativement à un point de référence r . Le point de référence est défini, pour chaque objectif m , par

$$r_m = \max_{y' \in P_k} y'_m + \delta,$$

avec $\delta = 0.1$, de manière à garantir que r se situe au-delà du front normalisé. Pour les problèmes bi- ou tri-objectifs ($p = 2$ ou 3), un calcul exact peut être effectué :

$$\text{HV}(P_k) = \sum_{i=1}^{N_k} \left(\prod_{j=1}^{p-1} (r_j - y'_{ij}) \right) (y'_{(i-1),p} - y'_{i,p}), \quad y'_{0,p} = r_p, \quad (5.3)$$

en supposant que les points sont triés de manière appropriée selon le dernier objectif. En dimension supérieure ($p > 3$), le calcul exact devient NP-difficile ; on emploie alors une approximation de type Monte Carlo [9] :

$$\text{HV}(P_k) \approx \frac{\text{dominated_count}}{\text{size_samples}} \cdot \prod_{m=1}^p (r_m - r_m^*), \quad (5.4)$$

où $r_m^* = \min_{y' \in P_k} y'_m$, et `dominated_count` désigne le nombre de points tirés aléatoirement (typiquement de l'ordre de 10^5 ou plus) dans l'hyper-rectangle $[r^*, r]$ qui sont dominés par au moins un point de P_k . Des valeurs plus élevées de l'hypervolume traduisent une meilleure couverture, car elles indiquent une région dominée plus large, incorporant à la fois un effet de convergence (solutions proches de l'optimum) et de diversité (front étendu).

3. **Espacement.** La métrique d'espacement, notée $SP(P_k)$, évalue l'uniformité en quantifiant la variance des distances inter-solutions sur le front normalisé :

$$SP(P_k) = \frac{1}{N_k - 1} \sum_{j=1}^{N_k} (d_j - \bar{d})^2, \quad (5.5)$$

où

$$d_j = \min_{l \neq j} \|y'_j - y'_l\|^2 \quad \text{et} \quad \bar{d} = \frac{1}{N_k} \sum_{j=1}^{N_k} d_j.$$

Ici, d_j est la distance euclidienne *carrée* au plus proche voisin du point y'_j , tandis que \bar{d} représente la moyenne de ces distances. Cette mesure basée sur la variance pénalise les regroupements locaux et les zones de faible densité : une valeur proche de 0 indique une distribution très régulière des solutions non dominées dans le front de Pareto.

4. **Étendue généralisée.** L'étendue généralisée, notée $GS(P_k)$, mesure la diversité et la couverture globale en comparant la distribution de P_k aux points extrémaux du front de référence global P_G [31] :

$$GS(P_k) = \frac{\sum_{m=1}^p d(v_m, P_k) + \sum_{j=1}^{N_k-1} |d_j - \bar{d}|}{\sum_{m=1}^p d(v_m, P_k) + (N_k - 1)\bar{d}}, \quad (5.6)$$

où

$$v_m = \arg \min_{y' \in P_G} y'_m \quad \text{et} \quad d(v_m, P_k) = \min_{y' \in P_k} \|v_m - y'\|^2.$$

Le point v_m correspond au point extrémal de P_G minimisant le m -ième objectif, et $d(v_m, P_k)$ est la distance (carrée) au point de P_k le plus proche de cet extrémum. Les quantités d_j et \bar{d} sont définies comme pour la métrique d'espacement, en supposant ici que les points sont ordonnés de manière adéquate afin de calculer les écarts séquentiels. Des valeurs plus faibles de $GS(P_k)$ indiquent de meilleures performances, car elles correspondent à (i) une bonne capture des extrêmes du front et (ii) une répartition plus régulière des solutions.

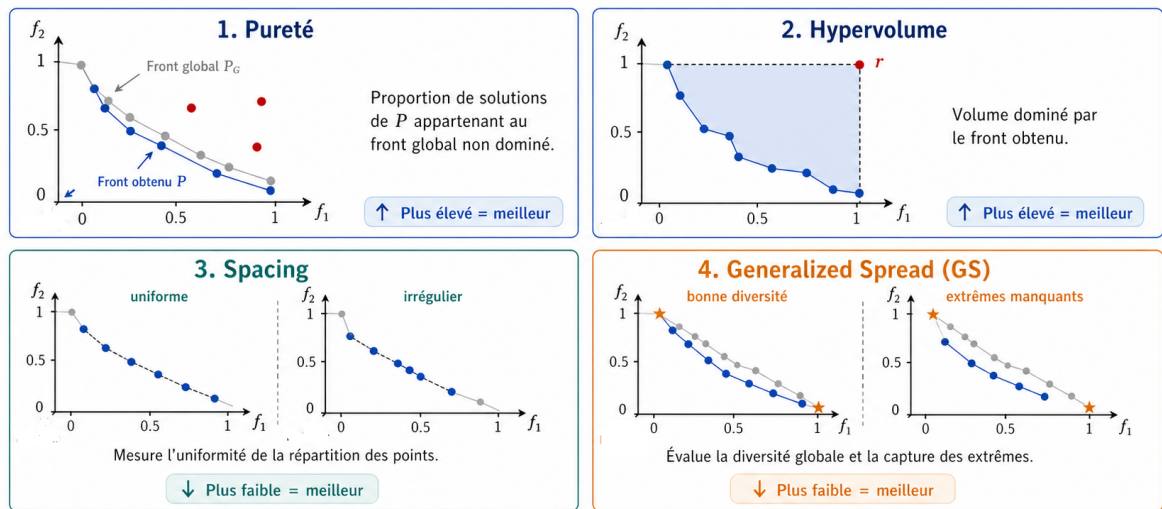


FIGURE 5.1 – Métriques d'évaluation des fronts de Pareto

5.5 Résultats et discussions

Dans cette section, nous analysons les performances des méthodes évaluées selon une démarche progressive. Dans un premier temps, nous comparons notre *MDR* à *MDR-MP* en mettant l'accent sur la qualité de la population initiale de points réalisables, en particulier la population générée par la procédure Hit-and-Run, ainsi que sur les principes d'échantillonnage utilisés dans la *MDR-MP*. Dans un second temps, nous comparons les deux méthodes de type *MDR* du point de vue de la convergence et de la vitesse de convergence, à travers le nombre d'itérations nécessaires et le temps CPU requis. Enfin, nous étendons l'analyse aux trois méthodes (notre *MDR*, *MDR-MP* et *NSGA-II*) en évaluant la qualité des fronts de Pareto obtenus à l'aide des quatre métriques introduites précédemment.

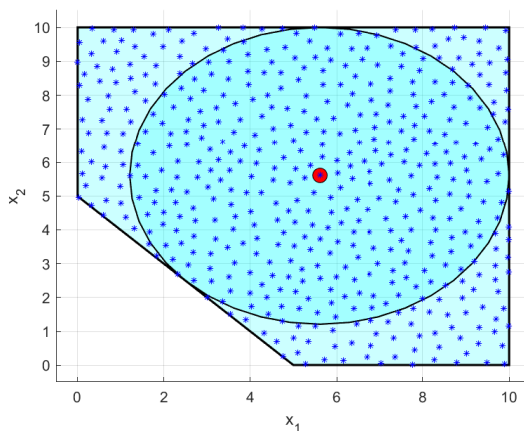


FIGURE 5.2 – Population de points réalisables générée par la procédure d'échantillonnage Hit-and-Run pour le problème test *HANNE3*.

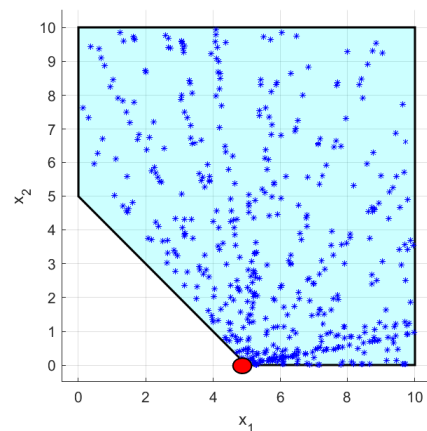


FIGURE 5.3 – Population de points réalisables générée par la procédure d'échantillonnage de Morovati et Pourkarimi pour le problème test *HANNE3*.

Les Figures (5.2) et (5.3) illustrent deux populations de 500 points réalisables associées au problème test *HANNE3*, générées respectivement par notre procédure Hit-and-Run et par la stratégie d'échantillonnage introduite par Morovati et Pourkarimi. Un point essentiel réside dans le choix du point de départ : dans notre approche, le point initial x_0 (marqué en rouge) est pris au centre de Chebyshev du polyèdre, c'est-à-dire au point qui maximise la distance minimale aux facettes. Ce choix fournit un démarrage "central" au sens géométrique, ce qui favorise une exploration plus équilibrée de l'ensemble réalisable et conduit généralement à une distribution plus homogène des points, avec une meilleure couverture des régions internes et une limitation des regroupements locaux.

À l'inverse, la technique de Morovati et Pourkarimi démarre typiquement à partir d'un sommet du polyèdre, obtenu via la résolution d'un programme linéaire. Cette initialisation, proche des frontières, a pour effet d'orienter la génération vers une zone particulière de l'ensemble faisable, ce qui se traduit par une concentration plus marquée des échantillons et par une couverture moins uniforme. Dans un contexte multi-objectif, une telle inhomogénéité de la population initiale peut limiter l'exploration des compromis, car certaines régions du polyèdre sont sous-échantillonnées dès le départ. Cette limitation se répercute ensuite sur la qualité des solutions non dominées et tend à produire des écarts plus importants entre points consécutifs sur le front de Pareto, ce qui est cohérent avec les valeurs d'espacement reportées (voir la Figure (5.11a)).

Le Tableau (5.2) récapitule, pour chaque problème test, le nombre d'itérations, le temps CPU moyen (en secondes) et la taille du front de Pareto P^* après suppression des solutions dominées et redondantes. Une première lecture met en évidence un point important : notre *MDR* ainsi que *NSGA-II* parviennent à résoudre l'ensemble des instances considérées, tandis que la *MDR-MP* échoue sur plusieurs problèmes, en particulier *MOLPG-1* à *MOLPG-3* et *MOQP-1* à *MOQP-3*. Ces échecs indiquent une moindre robustesse de *MDR-MP* lorsque la dimension augmente et/ou lorsque le nombre de contraintes devient plus élevé, ce qui peut être lié à la manière dont certaines contraintes sont traitées (notamment lorsque des contraintes inactives sont exclues), et à l'impact de cette sélection sur la stabilité numérique de la recherche de directions réalisables.

Au-delà de la robustesse, les résultats montrent que notre *MDR* est globalement plus rapide que *MDR-MP* en termes de vitesse de convergence, tant du point de vue du nombre d'itérations que du temps CPU. Par exemple, pour le problème *EXAMPLE 2.1*, notre méthode converge en moyenne en 2.96 itérations, contre 12.4550 itérations pour *MDR-MP*, ce qui traduit un gain significatif en efficacité itérative. De même, en termes de coût computationnel, la différence se reflète clairement : pour *EXAMPLE 7.1*, notre méthode

TABLE 5.2 – Ce tableau présente le nombre d’itérations nécessaires pour atteindre une solution Pareto-optimale, en indiquant les valeurs minimale, maximale et moyenne. Il fournit également le temps CPU moyen (en secondes) pour une population initiale de 200 points réalisables, ainsi que la taille du front de Pareto $|P^*|$, après suppression des solutions dominées et redondantes. Les résultats sont reportés pour trois méthodes : notre *MDR*, la *MDR* de *MDR-MP* et *NSGA-II*.

Problèmes	Itérations		Temps CPU			$ P^* $	NSGA-II
	Notre <i>MDR</i>	<i>MDR-MP</i>	Notre <i>MDR</i>	<i>MDR-MP</i>	Notre <i>MDR</i>		
HANNE3	[2,2]2	[1,2]1.9700	0.0014	0.0011	200	200	108
CONSTR	[2,2]2	[2,2]1.9100	0.0014	0.0010	200	200	129
KITA	[2,3]2.0905	[2,3]2.0790	0.0016	0.0015	198	200	132
EX004	[2,2]2	[2,2]2	0.0018	0.0013	200	200	170
KM	[2,5]2.2850	[1,8]2.8300	0.0015	0.0016	200	200	185
EXAMPLE 7.1	[2,9]3.9740	[1,10]5.4000	0.0027	0.0049	146	121	177
EXAMPLE 7.0	[2,14]4.5968	[1,14]5.9782	0.0031	0.0055	190	181	194
ABC-COMP	[2,70]6.0774	[1,143]12.9320	0.0100	0.0174	149	159	191
BEEME	[2,5]2.7950	[2,29]4.6850	0.0023	0.0067	200	200	199
COMET	[2,12]5.4000	[1,93]6.7600	0.0035	0.0049	200	200	194
SW1	[2,3]2.0205	[1,2]1.8100	0.0015	0.0010	200	200	156
SW2	[2,3]2.9045	[1,16]13.1950	0.0021	0.0081	200	200	198
PEKKA2	[2,3]2.1100	[2,3]2.0150	0.0015	0.0012	200	200	127
EXAMPLE 2.1	[2,4]2.9600	[1,19]12.4550	0.0021	0.0076	200	200	200
LISWETM	[6,10]8.5350	–	0.0065	–	200	–	200
NABETANI2008	[2,6]2.7150	[2,14]7.5800	0.0019	0.0060	200	200	200
KATHRIN2008	[1,6]1.5600	[1,7]1.8200	0.0013	0.0018	200	200	133
WIECEK2001	[2,223]90.3641	[2,219]59.9585	0.0676	0.0570	195	190	199
YOUNESS2004	[1,2]1.8800	[1,2]1.9100	0.0012	0.0013	200	200	200
MOLPG-1	[2,3]2.7100	–	0.0029	–	199	–	179
MOLPG-2	[2,7]4.5100	–	0.0063	–	200	–	134
MOLPG-3	[2,6]4.0550	–	0.0040	–	200	–	156
MOQP-1	[5,63]10.0250	–	0.0131	–	197	–	67
MOQP-2	[7,309]9.1684	–	0.0117	–	200	–	91
MOQP-3	[5,10]7.5050	–	0.0097	–	200	–	122

nécessite en moyenne 0.0027 secondes, tandis que *MDR-MP* atteint 0.0049 secondes. Ce comportement est cohérent avec l’idée que (i) la population initiale plus homogène obtenue par *Hit-and-Run* facilite l’exploration et (ii) le sous-problème de recherche de direction adopté dans notre approche renforce la qualité des directions générées, ce qui accélère la progression vers des points Pareto-stationnaire.

En outre, notre *MDR* produit, pour l’ensemble des problèmes testés, des fronts P^* complets et exploitables. À l’opposé, la *MDR-MP* peut rencontrer des difficultés sur certaines instances, ce qui se manifeste soit par un échec explicite (entrées “–” dans le tableau), soit par une dynamique moins stable selon les problèmes. Enfin, bien que *NSGA-II* résolve tous les problèmes, on observe qu’après filtrage des solutions dominées, la taille du front obtenu peut être plus faible que celle des deux méthodes *MDR* (par exemple 108 points pour *HANNE3* et 133 points pour *KATHRIN2008*). Cette réduction suggère que, dans notre protocole de comparaison, une proportion plus importante des solutions générées par *NSGA-II* devient dominée lorsqu’elle est confrontée au front de référence global, ce qui est cohérent avec des scores de pureté plus faibles (voir la Figure (5.10a)).

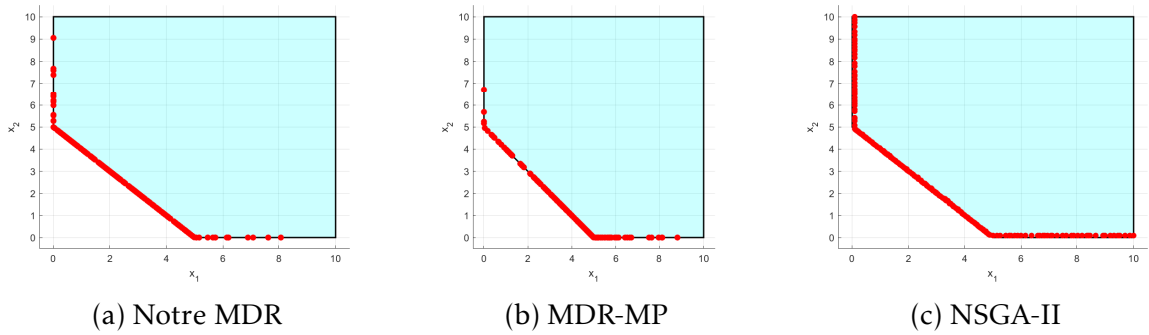


FIGURE 5.4 – Ensemble des solutions efficaces en espace de décision pour le problème test *HANNE3*.

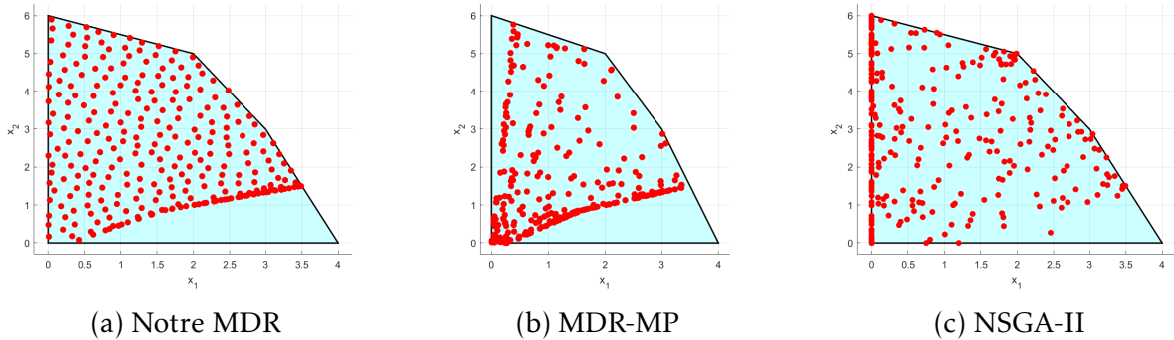


FIGURE 5.5 – Ensemble des solutions efficaces en espace de décision pour le problème test *KATHRIN2008*.

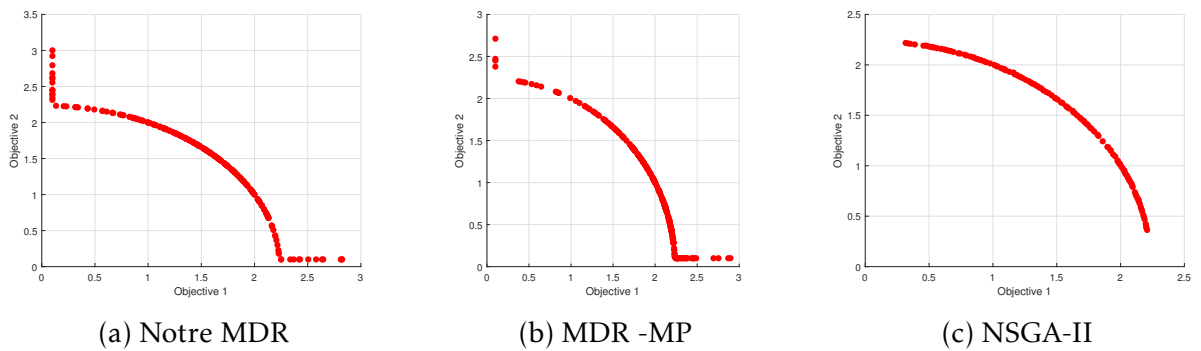


FIGURE 5.6 – Front de Pareto du problème test *HANNE3*.

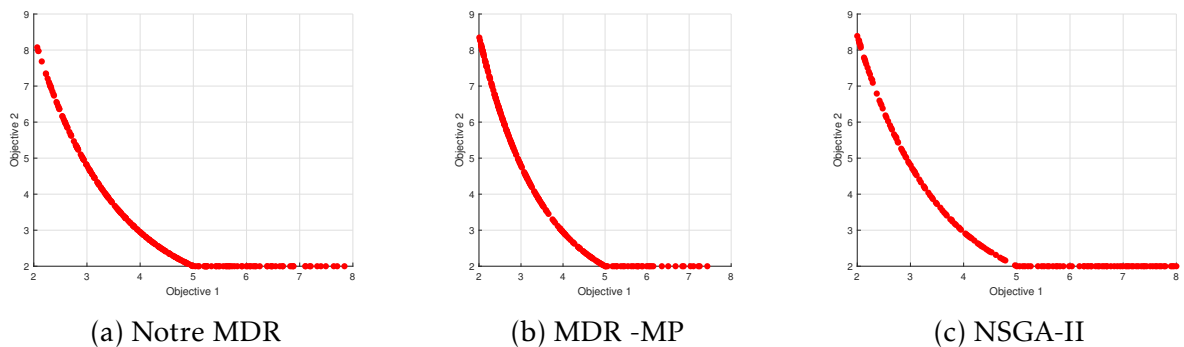


FIGURE 5.7 – Front de Pareto du problème test *EX004*.

Les Figures (5.4), (5.5), (5.6), (5.7), (5.8) et (5.9) comparent la qualité des solutions en espace de décision et en espace des objectifs.

Pour *HANNE3* (Figure (5.4)), notre *MDR* produit des solutions bien réparties, avec une pureté élevée (Figure (5.10a)) et un espacement faible (Figure (5.11a)), tandis que le front de *NSGA-II* est réduit à 108 points après suppression des solutions dominées (Tableau (5.2)). La *MDR-MP* présente des lacunes plus marquées (espacement plus élevé, Figure (5.11a)) en raison de l'échantillonnage initial à partir d'un sommet. Pour *EX004* (Figure (5.7)), les trois méthodes produisent des fronts identiques, avec des hypervolumes comparables (Figure (5.10b)). Pour *KATHRIN2008* (Figure (5.5)), qui comporte cinq objectifs, notre *MDR* et *NSGA-II* obtiennent des distributions plus régulières, tandis que la *MDR-MP* montre une couverture plus inégale (espacement plus élevé, Figure (5.11a)). Le front de *NSGA-II* est réduit à 133 points, ce qui traduit une pureté plus faible.

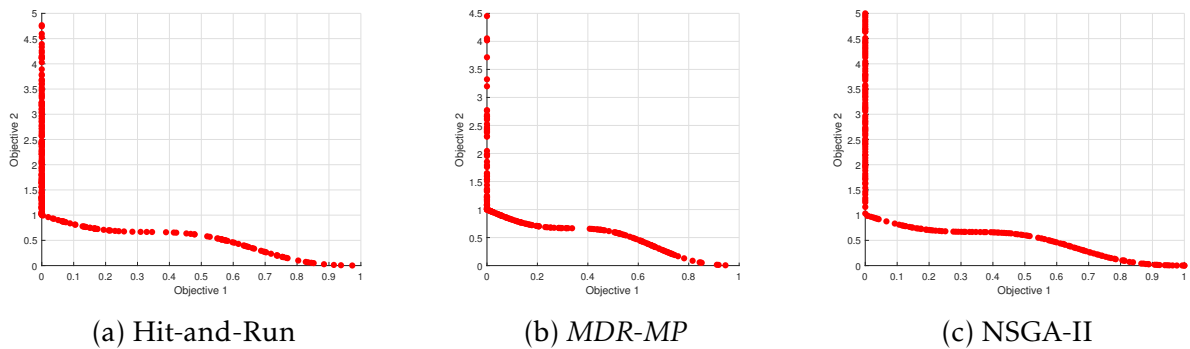


FIGURE 5.8 – Fronts de Pareto du problème test *EXAMPLE 7.0*.

Comme l'illustrent les Figures (5.8) et (5.9), notre *MDR* (couplée à *Hit-and-Run*) et *NSGA-II* génèrent, pour les problèmes *EXAMPLE 7.0* et *PEKKA2*, des ensembles de solutions non dominées globalement bien répartis le long des compromis, ce qui se traduit par de faibles valeurs de la métrique d'espacement (Figure (5.11a)). Cette régularité indique que les solutions couvrent de manière plus uniforme les différentes zones du front, ce qui facilite l'analyse des compromis et réduit la présence de "vides" entre solutions successives.

En revanche, la méthode *MDR-MP* produit, sur ces mêmes problèmes, des distributions moins régulières, avec des regroupements locaux et des zones moins couvertes. Cette structure se reflète par des valeurs d'espacement plus élevées (Figure (5.11a)), traduisant une uniformité moindre. Ce comportement est cohérent avec l'effet cumulé (i) d'une population initiale moins homogène et (ii) d'une exploration directionnelle qui peut privilégier certaines régions du polyèdre, ce qui réduit la capacité à maintenir un maillage régulier des solutions non dominées sur tout le front.

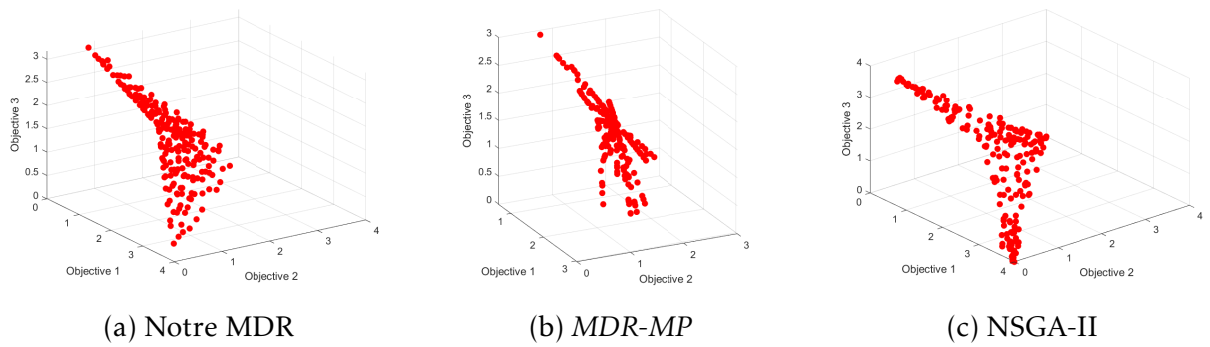


FIGURE 5.9 – Fronts de Pareto du problème test PEKKA2.

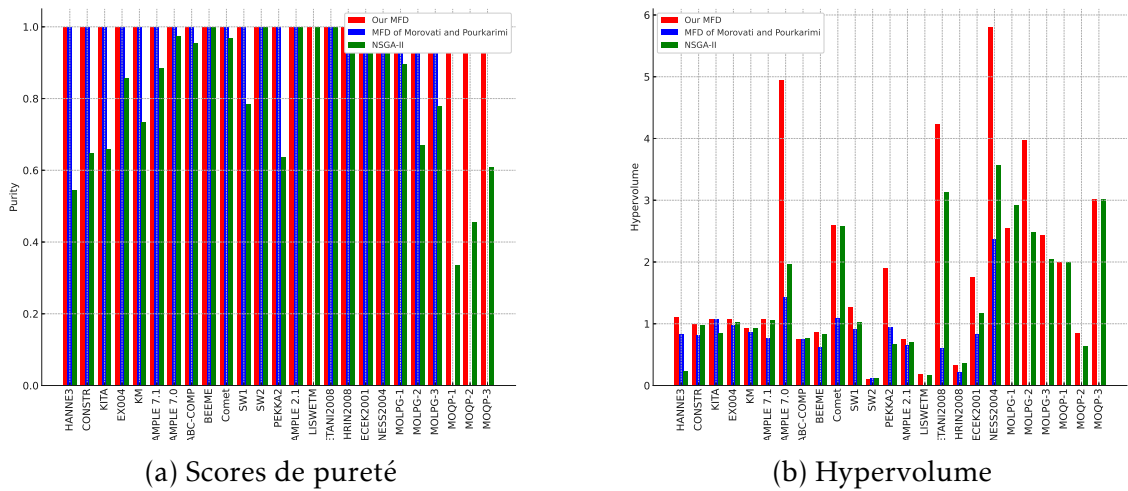


FIGURE 5.10 – Comparaison des scores de pureté et de l'hypervolume.

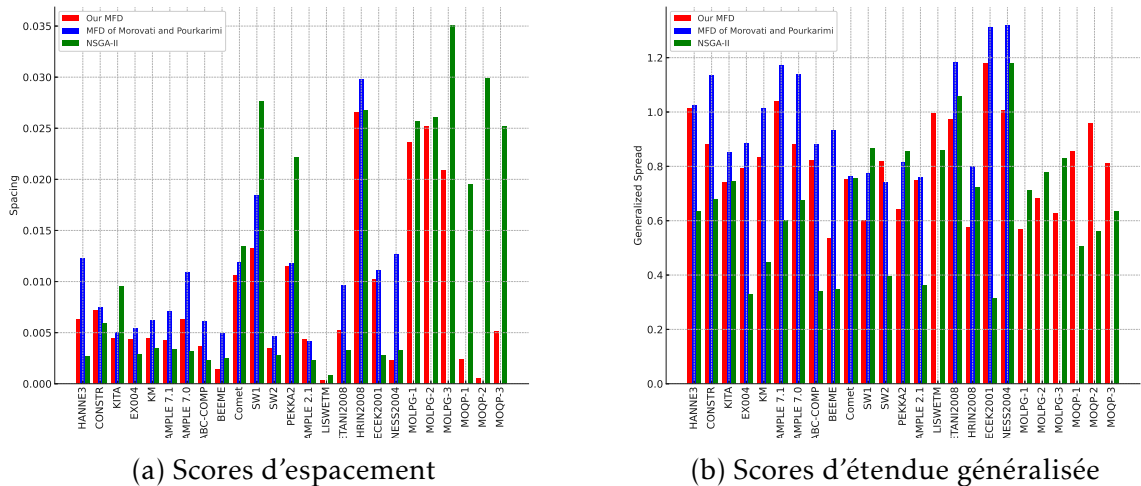


FIGURE 5.11 – Comparaison des scores d'espacement et des scores d'étendue généralisée.

Les Figures (5.10a), (5.10b), (5.11a) et (5.11b) présentent des histogrammes comparant les métriques de performance, et permettent une lecture synthétique des comportements observés sur la même liste de problème LC-MOO. En premier lieu, les deux variantes de

MDR (notre *MDR* et *MDR-MP*) obtiennent un score de pureté égal à 1.0 pour l'ensemble des problèmes (Figure (5.10a)). Cela signifie que, relativement au front de référence global P_G construit à partir des trois méthodes, les solutions conservées sur les fronts *MDR* sont toutes non dominées, ce qui indique que ces deux approches produisent des approximations compétitives et stables en termes de convergence.

En second lieu, notre *MDR* fournit de manière systématique des valeurs d'hypervolume plus élevées (Figure (5.10b)), ce qui traduit une meilleure couverture de l'espace objectif dominé. Concernant l'uniformité, notre *MDR* présente des valeurs d'espacement faibles et relativement stables (Figure (5.11a)), ce qui confirme que la distribution des solutions non dominées est régulière sur un grand nombre d'instances. À l'inverse, la *MDR-MP* affiche des valeurs d'espacement plus élevées pour certains problèmes tels que *HANNE3*, *EXAMPLE 7.0* et *NABETANI2008*, ce qui correspond à des fronts présentant des regroupements et des zones moins couvertes. De son côté, *NSGA-II* produit des valeurs d'espacement plus élevées sur plusieurs instances, notamment *SW1*, *KATHRIN2008* et la famille *MOQP-1* à *MOQP-3*, ce qui suggère une distribution moins régulière des solutions non dominées dans ces cas, malgré sa robustesse globale en termes de faisabilité.

Enfin, pour la métrique d'étendue généralisée, *NSGA-II* obtient les plus faibles valeurs (Figure (5.11b)), ce qui indique, au sens de cette métrique, une meilleure capacité à couvrir certains points extrêmes du front de référence. Cependant, notre *MDR* fournit des approximations globalement plus équilibrées sur l'ensemble des problèmes, en combinant une bonne couverture (hypervolume) et une bonne uniformité (espacement), tout en maintenant une pureté maximale. Elle surpasse également la *MDR-MP*, qui présente des lacunes notables sur plusieurs instances, notamment *CONSTR*, *EXAMPLE 7.0*, *KATHRIN2008* et *WIECEK2001*.

Conclusion

Dans ce chapitre, nous avons présenté une étude expérimentale permettant d'évaluer les performances de notre méthode *MDR* sur plusieurs problèmes d'optimisation multi-objectifs sous contraintes linéaires. Les résultats obtenus montrent que l'échantillonnage *Hit-and-Run* permet de générer des points initiaux mieux répartis dans l'ensemble réalisable, ce qui améliore l'exploration du front de Pareto. La comparaison avec la méthode de Morovati et Pourkarimi et avec *NSGA-II* montre que notre approche est généralement plus stable, plus rapide et capable de produire des fronts de bonne qualité. Les indicateurs utilisés, notamment la pureté, l'hypervolume, l'espacement et l'étendue générali-

sée, confirment que notre méthode donne des solutions bien réparties et compétitives. Ainsi, cette étude numérique met en évidence l'intérêt pratique de notre contribution et confirme l'efficacité de la méthode proposée pour résoudre les problèmes *LC-MOO* de petites et moyennes dimension.

Conclusion générale

Cette thèse s’inscrit dans le cadre de l’optimisation *multi-objectif* sous contraintes linéaires (*LC-MOO*), un domaine occupant une place centrale dans de nombreuses applications telles que l’optimisation de portefeuilles en finance, l’ingénierie des systèmes, la planification et l’allocation de ressources, la logistique ou encore la gestion de réseaux complexes. Contrairement à l’optimisation mono-objectif, il ne s’agit pas d’identifier une solution unique, mais de caractériser l’ensemble des solutions Pareto-optimales constituant le front de Pareto. Celui-ci met en évidence les compromis inhérents à des objectifs souvent conflictuels et fournit un cadre structuré d’aide à la décision, permettant au décideur de sélectionner, parmi les solutions efficaces, celle qui correspond le mieux à ses préférences et aux exigences du contexte considéré.

Bien que les méthodes de scalarisation soient largement utilisées pour approximer le front de Pareto, leur efficacité pratique demeure fortement tributaire du choix des paramètres (poids, niveaux ε , points de référence). Un paramétrage inadapté peut conduire à une exploration partielle, déséquilibrée, voire à l’omission de certaines régions du front. Ces limites motivent le recours à des approches non paramétriques fondées sur l’exploitation directe des propriétés différentielles des fonctions objectif. Dans cette perspective, plusieurs travaux ont proposé d’étendre au cadre multi-objectifs les méthodes classiques de directions réalisables développées en optimisation mono-objectif, en conservant leur principe fondamental : générer itérativement des directions améliorantes tout en maintenant la faisabilité vis-à-vis des contraintes.

L’objectif principal de cette thèse a consisté à proposer une amélioration substantielle d’une méthode de directions réalisables appliquée à la détermination du front de Pareto des problèmes *LC-MOO*. Deux contributions complémentaires ont été développées. La première concerne la reformulation du sous-problème de recherche de direction, afin d’intégrer l’ensemble des contraintes du polyèdre — et non uniquement les contraintes actives — au moyen d’un *cône de directions réalisables réduit*. Cette approche vise à stabili-

ser le comportement de l'algorithme au voisinage des frontières, à limiter les phénomènes d'oscillation et à produire des directions plus pertinentes du point de vue multi-objectifs, tout en garantissant le maintien de la faisabilité. La seconde contribution porte sur l'étape d'initialisation : une procédure d'échantillonnage de type *Hit-and-Run*, amorcée au centre de *Tchebychev* du polyèdre, a été proposée afin de générer une population initiale mieux répartie dans l'ensemble réalisable, favorisant ainsi une exploration plus homogène et une approximation plus fidèle du front de Pareto.

Une validation numérique approfondie a été conduite sous *Matlab*, en s'appuyant sur la boîte à outils *MPT3* pour la manipulation des polyèdres et le contrôle rigoureux de la faisabilité. La méthode développée a été comparée à deux approches de référence : la méthode *MDR* de Morovati et Pourkarimi ainsi que l'algorithme évolutionnaire *NSGA-II*. L'étude expérimentale a porté sur un ensemble de vingt-cinq problèmes *LC-MOO*, sélectionnés de manière à couvrir un large éventail de configurations issues de la littérature : différentes dimensions (nombres de variables n et de contraintes m), diverses cardinalités d'objectifs (p), fonctions objectif linéaires, quadratiques et non linéaires, ainsi que des structures géométriques convexes et non convexes. L'évaluation a reposé à la fois sur des critères d'efficacité numérique (nombre d'itérations et temps de calcul) et sur des indicateurs de qualité du front obtenu (pureté, hypervolume, espacement et étendue généralisée). Les résultats mettent en évidence la supériorité globale de l'approche proposée, qui fournit des fronts de meilleure qualité tout en assurant une convergence plus rapide et plus stable. Ainsi, la version améliorée de la méthode à directions réalisables développée dans cette thèse surpasse les deux méthodes de comparaison, tant en termes de performance computationnelle que de précision d'approximation du front de Pareto.

Plusieurs perspectives de recherche prolongent naturellement ces travaux. Une première orientation consiste à approfondir l'analyse théorique des deux variantes de la méthode *MDR*, en étudiant de manière détaillée leurs propriétés de convergence et en établissant explicitement leurs taux de convergence respectifs, afin de renforcer les fondements analytiques de l'approche. Une seconde perspective concerne le développement d'une procédure d'échantillonnage de type *Hit-and-Run* adaptée à des ensembles réalisables non convexes, éventuellement combinée à d'autres méthodes non paramétriques. Une telle extension constituerait une avancée significative pour le traitement de problèmes multi-objectifs plus généraux et contribuerait à élargir le champ d'applications pratiques de la méthode proposée.

Bibliographie

- [1] J. Abadie and J. Carpentier. Generalization of the wolfe reduced gradient method to the case of nonlinear constraints. In R. Fletcher, editor, *Optimization*, pages 37–47. Academic Press, New York, 1969.
- [2] M. A. Ansary and G. Panda. A modified quasi-newton method for vector optimization problem. *Optimization*, 64(11) :2289–2306, 2015. URL <https://doi.org/10.1080/02331934.2014.971008>.
- [3] P. Armand and C. Malivert. Determination of the efficient set in multiobjective linear programming. *Journal of Optimization Theory and Applications*, 70(3) :467–489, 1991. URL <https://doi.org/10.1007/BF00941298>.
- [4] R. K. Arora. *Optimization : Algorithms and Applications*. CRC Press, Boca Raton, 2015. URL <https://doi.org/10.1201/b18469>.
- [5] P. B. Assunção, O. P. Ferreira, and L. F. Prudente. Conditional gradient method for multiobjective optimization. *Computational Optimization and Applications*, 78(3) : 741–768, 2021. URL <https://doi.org/10.1007/s10589-020-00260-5>.
- [6] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon. Performance indicators in multiobjective optimization. *European Journal of Operational Research*, 292 :397–422, 2020. doi : 10.1016/j.ejor.2020.11.016. URL <https://doi.org/10.1016/j.ejor.2020.11.016>.
- [7] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8 : 295–313, 1992. URL <https://doi.org/10.1007/BF02293050>.
- [8] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1) :21–46, 1996. doi : 10.1016/0166-218X(95)00026-N. First International Colloquium on Graphs and Optimization.

-
- [9] J. Bader and E. Zitzler. Hype : An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1) :45–76, 2011. URL https://doi.org/10.1162/EVC0_a_00009.
- [10] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming : Theory and Algorithms*. John Wiley and Sons, 3 edition, 2006.
- [11] C. J. P. Bélisle, H. E. Romeijn, and R. L. Smith. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2) :255–266, 1993. URL <https://doi.org/10.1287/moor.18.2.255>.
- [12] H. P. Benson. Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Applications*, 26(4) :569–580, 1978. URL <https://doi.org/10.1007/BF00933152>.
- [13] M. Berger. *Geometry I*. Universitext. Springer Berlin Heidelberg, 2009. URL <https://books.google.ro/books?id=5W6cnfQegYcC>.
- [14] N. K. Bisui and G. Panda. A trust-region scheme for constrained multi-objective optimization problems with superlinear convergence property. *Optimization Methods and Software*, 39 :1–41, 2024. URL <https://doi.org/10.1080/10556788.2024.2372303>.
- [15] V. J. Bowman. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thiriez and S. Zionts, editors, *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, Heidelberg, 1976. URL https://doi.org/10.1007/978-3-642-87563-2_5.
- [16] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [17] G. A. Carrizo, P. A. Lotito, and M. C. Maciel. Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem. *Mathematical Programming*, 159(1–2) :339–369, 2016. URL <https://doi.org/10.1007/s10107-015-0962-6>.
- [18] M. E. Cawood and M. M. Kostreva. Norm-relaxed method of feasible directions for solving nonlinear programming problems. *Journal of Optimization Theory and Applications*, 83 :311–320, 1994. URL <https://doi.org/10.1007/BF02190059>.
- [19] R. W. Chaney. On the pironneau-polak method of centers. *Journal of Optimization Theory and Applications*, 20 :269–295, 1976.

- [20] R. W. Chaney. On the rate of convergence of some feasible direction algorithms. *Journal of Optimization Theory and Applications*, 20 :297–313, 1976.
- [21] J. Chen, S. He, Y. Sun, P. Thulasiraman, and X. Shen. Optimal flow control for utility-lifetime tradeoff in wireless sensor networks. *Computer Networks*, 53 :3031–3041, 2009. URL <https://doi.org/10.1016/j.comnet.2009.07.014>.
- [22] K. Chen, E.H. Fukuda, and H. Sato. Nonlinear conjugate gradient method for vector optimization on riemannian manifolds with retraction and vector transport. *Applied Mathematics and Computation*, 486 :129001, 2025. URL <https://doi.org/10.1016/j.amc.2024.129001>.
- [23] W. Chen, X. Yang, and Y. Zhao. Conditional gradient method for vector optimization. *Computational Optimization and Applications*, 85 :857–896, 2023. URL <https://doi.org/10.1007/s10589-023-00478-z>.
- [24] X. Chen and M. M. Kostreva. Methods of feasible directions : A review. In X. Yang, A. I. Mees, M. Fisher, and L. Jennings, editors, *Progress in Optimization*, volume 39 of *Applied Optimization*. Springer, 2000. doi : 10.1007/978-1-4613-0301-5_14.
- [25] X. B. Chen and M. M. Kostreva. A generalization of the norm-relaxed method of feasible directions. *Applied Mathematics and Computation*, 102(2-3) :257–272, 1999. URL [https://doi.org/10.1016/S0096-3003\(98\)10025-5](https://doi.org/10.1016/S0096-3003(98)10025-5).
- [26] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2nd edition, 2007. doi : 10.1007/978-0-387-36797-2.
- [27] Y. Collette and P. Siarry. *Optimisation Multiobjectif*. Eyrolles, 2002.
- [28] G. E. Danninger-Uchida. Krein–milman theorem. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization*. Springer US, 2009. URL https://doi.org/10.1007/978-0-387-74759-0_316.
- [29] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1) :63–69, 1997. URL <https://doi.org/10.1007/BF01197559>.
- [30] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm : NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2) : 182–197, 2002. URL <https://doi.org/10.1109/4235.996017>.
- [32] D. Ding-Zhu, F. Wu, and X.S. Zhang. On Rosen’s gradient projection methods. *Annals of Operations Research*, 24 :9–28, 1990. URL <https://doi.org/10.1007/BF02216813>.
- [33] M. E. Dyer and A. M. Friez. On the complexity of computing the volume of a polyhedron. *SIAM Journal on Computing*, 17(5) :967–974, 1988. URL <https://doi.org/10.1137/0217060>.
- [34] J. G. Ecker and I. A. Kouada. Finding efficient points for linear multiple objective programs. *Mathematical Programming*, 8(1) :375–377, 1975. URL <https://doi.org/10.1007/BF01580453>.
- [35] M. Ehrgott. *Multicriteria Optimization*. Springer, 2 edition, 2005-03. ISBN 3-540-213-98-8.
- [36] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer, Berlin, 2008. URL <https://doi.org/10.1007/978-3-540-79159-1>.
- [37] M. El Moudden and A. El Ghali. A new reduced gradient method for solving linearly constrained multiobjective optimization problems. *Computational Optimization and Applications*, 71(3) :719–741, 2018. URL <https://doi.org/10.1007/s10589-018-0023-1>.
- [38] M. El Moudden and A. El Ghali. Multiple reduced gradient method for multiobjective optimization problems. *Numerical Algorithms*, 79(4) :1257–1282, 2018. URL <https://doi.org/10.1007/s11075-018-0483-5>.
- [39] B. El-Sobky and Y. Abo-Elnaga. An active-set trust-region algorithm for solving constrained multi-objective optimization problem. *Applied Mathematical Sciences*, 6(33) :1599–1612, 2012. URL <https://www.m-hikari.com/ams/ams-2012/ams-33-36-2012/e1sobkyAMS33-36-2012.pdf>.
- [40] J. Fliege and B. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51 :479–494, 2000. URL <https://doi.org/10.1007/s001860000043>.
- [41] J. Fliege and A. I. F. Vaz. A method for constrained multiobjective optimization based on sqp techniques. *SIAM Journal on Optimization*, 26(4) :2091–2119, 2016. URL <https://doi.org/10.1137/15M1016424>.

- [42] J. Fliege, L. M. Graña Drummond, and B. Svaiter. Newton’s method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2) :602–626, 2009. URL <https://doi.org/10.1137/08071692X>.
- [43] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3 :95–110, 1956. URL <https://doi.org/10.1002/nav.3800030109>.
- [44] K. Fukuda and A. Prodon. Double description method revisited. In Michel Deza, Reinhardt Euler, and Ioannis Manoussakis, editors, *Combinatorics and Computer Science*, pages 91–111. Springer Berlin Heidelberg, 1996. URL https://doi.org/10.1007/3-540-61576-8_77.
- [45] A. Garcia-Bernabeu and J. M. Herrero. Computing the mean-variance sustainability nondominated surface by ev-moga. *Complexity*, 2019 :6095712, 2019. URL <https://doi.org/10.1155/2019/6095712>. Article ID 6095712, 12 pages.
- [46] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1–2) :39–45, 1955. URL <https://doi.org/10.1002/nav.3800020106>.
- [47] B. Gebken, S. Peitz, and M. Dellnitz. A descent method for equality and inequality constrained multiobjective optimization problems. In Leonardo Trujillo, Oliver Schütze, Yazmin Maldonado, and Paul Valle, editors, *Numerical and Evolutionary Optimization – NEO 2017*, volume 785 of *Studies in Computational Intelligence*, pages 29–61. Springer, Cham, 2019. URL https://doi.org/10.1007/978-3-319-96104-0_2.
- [48] J. Gipple. The volume of n-balls. *Rose-Hulman Undergraduate Mathematics Journal*, 15(1) :Article 14, 2014. URL <https://scholar.rose-hulman.edu/rhumj/vol15/iss1/14>.
- [49] L.M. Graña Drummond and B.F. Svaiter. A steepest descent method for vector optimization. *Journal of Computational and Applied Mathematics*, 175(2) :395–414, 2005. URL <https://doi.org/10.1016/j.cam.2004.06.020>.
- [50] L.M. Graña Drummond and A.N. Iusem. A projected gradient method for vector optimization problems. *Computational Optimization and Applications*, 28(1) :5–29, 2004. URL <https://doi.org/10.1023/B:COAP.0000018877.76336.43>.
- [51] Yacov Y. Haimes, Leon S. Lasdon, and David A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3) :296–297, 1971.

- doi : 10.1109/TSMC.1971.4308298. URL <https://doi.org/10.1109/TSMC.1971.4308298>.
- [52] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-parametric toolbox 3.0. In *Proceedings of the European Control Conference*, pages 502–511, 2013. URL <https://www.mpt3.org/>.
- [53] P. Huard. Convergence of the reduced gradient method. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming 2*, pages 29–54. Academic Press, 1975. URL <https://doi.org/10.1016/B978-0-12-468650-2.50006-7>.
- [54] C.L. Hwang and A.d. Masud. *Multiple Objective Decision Making—Methods and Applications (A State-of-the-Art Survey)*. Springer, Berlin, 1979. URL <https://doi.org/10.1007/978-3-642-45511-7>.
- [55] .H Ishibuchi, .R Imada, .Y Setoguchi, and .Y Nojima. Reference point specification in hypervolume calculation for fair comparison and efficient search. In *Proceedings of the 2017 Genetic and Evolutionary Computation Conference (GECCO '17)*, pages 585–592, Berlin, Germany, July 2017. ACM. URL <https://doi.org/10.1145/3071178.3071281>.
- [56] I. Kaliszewski. *Quantitative Pareto Analysis by Cone Separation Technique*. Kluwer Academic Publishers, 1994.
- [57] V. Karmanov. *Programmation mathématique*. Mir, 1989.
- [58] K. Klamroth and K. Miettinen. Integrating approximation and interactive decision making in multicriteria optimization. *Operations Research*, 56(1) :222–234, 2008. URL <https://doi.org/10.1287/opre.1070.0425>.
- [59] J. A. Korycki and M. M. Kostreva. Convergence analysis of norm-relaxed method of feasible directions. *Journal of Optimization Theory and Applications*, 91 :389–418, 1996. URL <https://doi.org/10.1007/BF02190102>.
- [60] J. A. Korycki and M. M. Kostreva. Norm-relaxed method of feasible directions : application in structural optimization. *Structural Optimization*, 11 :187–194, 1996. URL <https://doi.org/10.1007/BF01197034>.
- [61] A. Laaksonen. *Guide to Competitive Programming : Learning and Improving Algorithms Through Contests*. Springer, 2018.

- [62] J. B. Lasserre. An analytical expression and an algorithm for the volume of a convex polyhedron in \mathbb{R}^n . *Journal of Optimization Theory and Applications*, 39(3) :363–377, 1983. URL <https://doi.org/10.1007/BF00934543>.
- [63] J. Lawrence. Polytope volume computation. *Mathematics of Computation*, 57(195) : 259–271, 1991. URL <https://doi.org/10.2307/2938672>.
- [64] S. Leyffer. Multiobjective test problems. Argonne National Laboratory, 2025. URL <https://wiki.mcs.anl.gov/leyffer/index.php/MacMOOP>. Accessed : 2025-04-15.
- [65] L. Lovász. Hit-and-run mixes fast. *Mathematical Programming*, 86 :443–461, 1999. URL <https://doi.org/10.1007/s101070050099>.
- [66] H. Mala-Jetmarova, N. Sultanova, and D. Savic. Lost in optimisation of water distribution systems? A literature review of system operation. *Environmental Modelling and Software*, 93 :209–254, 2017. URL <https://doi.org/10.1016/j.envsoft.2017.02.009>.
- [67] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6) :369–395, 2004. URL <https://doi.org/10.1007/s00158-003-0368-6>.
- [68] .G Mavrotas. Effective implementation of the ε -constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2) :455–465, 2009. URL <https://doi.org/10.1016/j.amc.2009.03.037>.
- [69] A. Messac, A. Ismail-Yahaya, and C.A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25 :86–98, 2003. URL <https://doi.org/10.1007/s00158-002-0276-1>.
- [70] K. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Kluwer Academic Publishers, Boston, 1999.
- [71] V. Morovati and L. Pourkarimi. Extension of zoutendijk method for solving constrained multiobjective optimization problems. *European Journal of Operational Research*, 273 :44–57, 2019. URL <https://doi.org/10.1016/j.ejor.2018.08.018>.
- [72] V. Morovati, H. Basirzadeh, and L. Pourkarimi. Quasi-newton methods for multiobjective optimization problems. *4OR : A Quarterly Journal of Operations Research*, 16(3) :261–294, 2017. URL <https://doi.org/10.1007/s10288-017-0355-8>.

- [73] C. Oliva, Pedro R. Ventura, and Luis F. Lago-Fernández. Multi-objective portfolio optimization via gradient descent. arXiv preprint arXiv :2507.16717, 2025. URL <https://arxiv.org/abs/2507.16717>.
- [74] H. L. Ong, H. C. Huang, and W. M. Huin. Finding the exact volume of a polyhedron. *Advances in Engineering Software*, 34(6) :351–356, 2003. URL [https://doi.org/10.1016/S0965-9978\(03\)00030-9](https://doi.org/10.1016/S0965-9978(03)00030-9).
- [75] J. Persson. Constrained nsga-ii with seeding enabled. MATLAB Central File Exchange, 2025. URL <https://www.mathworks.com/matlabcentral/fileexchange/75338-constrained-nsga-ii-with-seeding-enabled>. Accessed : 2025-03-30.
- [76] O. Pironneau and E. Polak. On the rate of convergence of certain methods of centers. *Mathematical Programming*, 2 :230–257, 1972.
- [77] O. Pironneau and E. Polak. Rate of convergence of a class of methods of feasible directions. *SIAM Journal on Numerical Analysis*, 10(1) :161–174, 1973. URL <https://doi.org/10.1137/0710017>.
- [78] Z. Povalej. Quasi-newton’s method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 255 :765–777, 2014. URL <https://doi.org/10.1016/j.cam.2013.06.022>.
- [79] S. Qu, Y. Ji, J. Jiang, and Q. Zhang. Nonmonotone gradient methods for vector optimization with a portfolio optimization application. *European Journal of Operational Research*, 263(2) :356–366, 2017. URL <https://doi.org/10.1016/j.ejor.2017.05.027>.
- [80] Z. Ramdani, S. Addoune, and .B Brahmi. Method of feasible directions with hit-and-run sampling for solving linearly constrained multi-objective optimization problems. *Journal of Optimization Differential Equations and their Applications*, 33 (2) :158–186, 2025. URL <https://doi.org/10.15421/142519>.
- [81] R. T. Rockafellar. *Convex analysis*. University Press, Princeton, 1970.
- [82] J. B. Rosen. The gradient projection method for nonlinear programming, i : Linear constraints. *SIAM Journal on Applied Mathematics*, 8 :181–217, 1960. URL <http://www.jstor.org/stable/2098960>.
- [83] F. Salas-Molina, J. A. Rodríguez-Aguilar, and P. Díaz-García. Selecting cash management models from a multiobjective perspective. *Annals of Operations Research*, 261 :275–288, 2018. URL <https://doi.org/10.1007/s10479-017-2634-9>.

-
- [84] Y. Sato, K. Izui, T. Yamada, and S. Nishiwaki. Gradient-based multiobjective optimization using a distance constraint technique and point replacement. *Engineering Optimization*, 48(7) :1226–1250, 2016. URL <https://doi.org/10.1080/0305215X.2015.1111068>.
- [85] R. L. Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6) :1296–1308, 1984. URL <https://doi.org/10.1287/opre.32.6.1296>.
- [86] R. E. Steuer and E. U. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3) :326–344, 1983. doi : 10.1007/BF02591870. URL <https://doi.org/10.1007/BF02591870>.
- [87] R.E. Steuer. *Multiple Criteria Optimization : Theory, Computation and Application*. John Wiley and Sons, New York, 1986. URL <https://doi.org/10.1002/oca.4660100109>.
- [88] E. Tarek. Method of centers algorithm for multi-objective programming problems. *Acta Mathematica Scientia*, 29(5) :1128–1142, 2009. URL [https://doi.org/10.1016/S0252-9602\(09\)60091-6](https://doi.org/10.1016/S0252-9602(09)60091-6).
- [89] D. M. Topkis and A. F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM Journal on Control and Optimization*, 5(2) :268–279, 1967. URL <https://doi.org/10.1137/0305018>.
- [90] R. E. Wendell and D. N. Lee. Efficiency in multiple objective optimization problems. *Mathematical Programming*, 12(1) :406–414, 1977. URL <https://doi.org/10.1007/BF01593811>.
- [91] E. J. Wiest and E. Polak. A generalized quadratic programming-based phase i-phase ii method for inequality constrained optimization. *Journal of Applied Mathematics and Optimization*, 26 :223–252, 1992. URL <https://doi.org/10.1007/BF01371083>.
- [92] P. Wolfe. Methods of nonlinear programming. In R. L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 67–86. McGraw-Hill, New York, 1963.
- [93] P. Wolfe. On the convergence of gradient methods under constraints. Technical report, IBM Research, Yorktown Heights, 1967.
- [94] M. Yaghoubi and F. Dadmand. A multi-objective model for humanitarian logistics during an earthquake crisis : A case study of iran. *Journal of Control, Optimization*

- and Applied Mathematics*, 9 :21–41, 2024. URL <https://doi.org/10.30473/coam.2024.31.1238>.
- [95] G. Yang, X. Li, L. Huo, and Q. Liu. A solving approach for fuzzy multi-objective linear fractional programming and application to an agricultural planting structure optimization problem. *Chaos, Solitons & Fractals*, 141 :110352, 2020. URL <https://doi.org/10.1016/j.chaos.2020.110352>.
- [96] Z. B. Zabinsky. *Stochastic Adaptive Search for Global Optimization*. Nonconvex Optimization and Its Applications. Springer, 2003. URL <https://doi.org/10.1007/978-1-4419-9182-9>.
- [97] L.A. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1) :59–60, 1963. URL <https://doi.org/10.1109/TAC.1963.1105511>.
- [98] W.I. Zangwill. The convex simplex method. *Management Science*, 14(3) :221–238, 1967. URL <https://doi.org/10.1287/mnsc.14.3.221>.
- [99] M. Zeleny. Compromise programming. In J. L. Cochrane and Milan Zeleny, editors, *Multiple Criteria Decision Making*, pages 262–301. University of South Carolina Press, Columbia, South Carolina, 1973.
- [100] M. Zeleny. *Multiple Criteria Decision Making*. McGraw-Hill, New York, 1982.
- [101] G. M. Ziegler. *Lectures on Polytopes*. Graduate Texts in Mathematics. Springer New York, 2012. URL <https://books.google.ro/books?id=xd25TXSSUcgC>.
- [102] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2 : Improving the strength pareto evolutionary algorithm. Technical Report 103, Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland, May 2001.
- [103] G. Zoutendijk. *Methods of Feasible Directions*. A study in linear and non-linear programming. Elsevier publishing company, 1960. URL https://books.google.ro/books/about/Methods_of_Feasible_Directions.html?id=dT0tnQAACAAJ&redir_esc=y.