



HAL
open science

La Méthode Adaptée et l'Optimisation Multicritère à Plusieurs Niveaux

Mustapha Kaci

► **To cite this version:**

Mustapha Kaci. La Méthode Adaptée et l'Optimisation Multicritère à Plusieurs Niveaux. Mathématiques [math]. Université des Sciences et de la Technologie d'Oran Mohamed-Boudiaf USTOMB, 2023. Français. NNT: . tel-04184442

HAL Id: tel-04184442

<https://hal.science/tel-04184442>

Submitted on 21 Aug 2023

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

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



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



THÈSE

En vue de l'obtention du Diplôme de Doctorat

Présentée par: KACI Mustapha

Intitulé

La méthode adaptée et l'optimisation multicritère à plusieurs niveaux

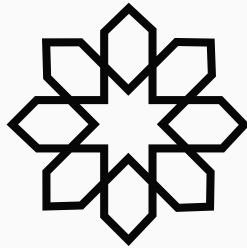
Faculté : *Mathématiques et Informatique*
Département : *Mathématique*
Domaine : *Mathématiques et Informatique*
Filière : *Mathématiques*
Intitulé de la Formation : *Mathématiques*

Soutenue le 14/06/2023 devant le Jury Composé de :

<i>Membres de Jury</i>	<i>Grade</i>	<i>Qualité</i>	<i>Domiciliation</i>
<i>M. TLEMCANI Mounir</i>	<i>Professeur</i>	<i>Président</i>	<i>Université USTO-MB</i>
<i>Mme. RADJEF Sonia</i>	<i>Professeure</i>	<i>Directrice de thèse</i>	<i>Université USTO-MB</i>
<i>M. MOULAÏ Mustapha</i>	<i>Professeur</i>	<i>Examineur</i>	<i>Université USTHB</i>
<i>M. TERBECHÉ Mekki</i>	<i>Professeur</i>	<i>Examineur</i>	<i>Université USTO-MB</i>
<i>M. LEBBAH Yahia</i>	<i>Professeur</i>	<i>Examineur</i>	<i>Université Oran 1</i>

Année Universitaire : 2022-2023

محمد



الله

**Au nom d'Allah, le Tout Miséricordieux,
le Très Miséricordieux**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

**In the name of Allah, the Most Gracious,
the Most Merciful**

السلام عليكم ورحمة الله وبركاته

Que la bénédiction et la paix d'Allah soient avec vous

REMERCIEMENTS

Je tiens en premier lieu à exprimer ma profonde reconnaissance envers Allah le Tout-Puissant pour m'avoir guidé et soutenu tout au long de ces trois années de recherche, me permettant ainsi de mener à bien cette thèse.



Je souhaite adresser mes sincères remerciements à ma directrice de thèse, Madame Radjef-Douar Sonia, Professeur à l'Université des sciences et de la technologie d'Oran Mohamed Boudiaf. Son engagement indéfectible, sa confiance inébranlable, ses conseils avisés et son assistance précieuse ont été d'une importance capitale pour la réalisation de ce travail. Travailler sous sa supervision a été une expérience enrichissante et inspirante, empreinte de rigueur scientifique et de bienveillance académique. Je tiens à lui exprimer ma profonde gratitude pour tout ce qu'elle m'a enseigné au cours de ces trois années. Mes remerciements les plus chaleureux, Professeur Radjef-Douar.

Je suis également extrêmement reconnaissant envers M. Tlemcani Mounir, éminent Professeur et chercheur à l'Université des sciences et de la technologie d'Oran Mohamed Boudiaf, qui a accepté d'assumer la présidence du jury malgré ses multiples obligations scientifiques et pédagogiques. Je tiens à lui exprimer ma plus grande estime et mes remerciements les plus sincères. Merci infiniment, Professeur Tlemcani.

J'aimerais également témoigner ma sincère gratitude envers M. Moulaï Mustapha, Professeur à l'Université des sciences et de la technologie Houari Boumediene, pour avoir accepté de faire partie de mon jury et pour avoir consacré son temps précieux à l'évaluation de cette thèse. Son expertise et ses commentaires éclairés ont grandement enrichi mon travail. Je lui suis profondément reconnaissant pour sa générosité intellectuelle. Merci, Professeur Moulaï.

Mes remerciements vont également à M. Terbeche Mekki, Professeur à l'Université des



sciences et de la technologie d'Oran Mohamed Boudiaf, qui a accepté avec bienveillance d'examiner cette thèse. Sa contribution et ses précieuses observations ont été d'une grande valeur ajoutée à mon travail. Je lui adresse ma gratitude la plus sincère. Merci, Professeur Terbeche.

Je souhaite exprimer ma reconnaissance envers M. Lebbah Yahia, Professeur distingué à l'Université des sciences et de la technologie d'Oran 1 Ahmed Ben Bella, pour sa disponibilité et sa participation active en tant qu'évaluateur de cette thèse. Sa contribution éclairée a grandement enrichi la qualité de mon travail. Je lui suis infiniment reconnaissant et lui adresse mes sincères remerciements. Merci, Professeur Lebbah.

Je tiens également à exprimer ma reconnaissance envers tous mes amis et collègues qui m'ont soutenu tout au long de cette aventure intellectuelle. Leurs encouragements, leurs discussions fructueuses et leur collaboration ont été sources de motivation et de stimulation. Leur présence bienveillante dans ma vie académique a contribué à faire de cette thèse une expérience enrichissante et mémorable. Je leur adresse mes sincères remerciements.

Je voudrais profiter de cette occasion pour adresser un hommage spécial à mon cher grand-père Kaci Mohamed. Son influence positive dans ma vie a été inestimable, et je suis profondément reconnaissant pour tous les sacrifices qu'il a consentis et tout ce qu'il a fait pour moi. Son amour inconditionnel, sa sagesse et son soutien constant ont été une source d'inspiration et de force tout au long de mon parcours. Ses précieux enseignements, son encouragement inébranlable et ses précieux conseils m'ont guidé et ont nourri ma détermination à poursuivre mes objectifs. Je suis honoré d'avoir eu la chance de bénéficier de sa présence bienveillante et de son accompagnement tout au long de ma vie. Je lui suis profondément reconnaissant pour les valeurs qu'il m'a inculquées et pour le modèle exemplaire qu'il représente. Je ne saurais jamais exprimer suffisamment ma gratitude envers lui. Merci, cher grand-père, pour tout ce que tu as fait pour moi.

En conclusion, je voudrais exprimer ma profonde reconnaissance envers toutes les personnes qui ont contribué, directement ou indirectement, à la réalisation de cette thèse. Leurs conseils, leur expertise, leur soutien et leur amitié ont été d'une valeur inestimable. Je suis honoré d'avoir eu l'opportunité de travailler avec de tels mentors, collègues et amis exceptionnels. Merci du fond du cœur à tous ceux qui ont joué un rôle dans cette aventure intellectuelle.



*"Le plus court chemin
d'une personne à une
autre... c'est un brin
de gentillesse."*

Merci à Tous

RÉSUMÉ

La présente thèse se concentre principalement sur le développement de méthodes de résolution de problèmes de programmation linéaire à plusieurs niveaux, en se basant sur le principe de la méthode adaptée. La résolution efficace de problèmes multi-niveaux reste un défi complexe. Cette difficulté se retrouve également dans la plupart des problèmes abordés dans ce travail.

L'objectif de cette thèse est de généraliser la méthode adaptée pour résoudre les problèmes de programmation linéaire multi-critère à plusieurs niveaux. Dans le chapitre 3, une étude géométrique de la stabilité des solutions d'un problème de programmation linéaire mono-critère est présentée. Les chapitres 5 et 6 sont consacrés à la construction de procédures basées sur le principe de la méthode adaptée pour résoudre les problèmes de programmation linéaire à plusieurs niveaux. Dans le chapitre 5, une construction technique avancée d'une application est établie pour définir une nouvelle estimation de sous-optimalité. Ces constructions sont ensuite utilisées pour développer deux versions d'un algorithme résolvant les problèmes de programmation linéaire multi-niveaux à objectif unique. Enfin, dans le chapitre 6, nous généralisons le concept de solutions non-dominées dans le contexte des problèmes linéaires multi-niveaux, et nous mettons en place une méthode pour les générer. Ce nouveau concept est utilisé pour décrire un algorithme qui applique la méthode adaptée afin d'atteindre un compromis unique.

Les résultats obtenus démontrent l'efficacité et la pertinence des méthodes développées pour résoudre les problèmes de programmation linéaire à plusieurs niveaux. Cette thèse apporte ainsi des contributions significatives à la résolution de problèmes complexes et ouvre de nouvelles perspectives pour la recherche future dans ce domaine.

Mots clés : Programmation linéaire, Programmation multi-objectifs, Programmation multi-niveaux, Analyse de sensibilité, Méthode adaptée, Estimation de sous-optimalité, Solutions non-dominées.



ABSTRACT

This thesis primarily focuses on the development of methods for solving multi-level linear programming problems based on the principle of the adaptive method. Efficiently solving multi-level problems remains a complex challenge, which is also evident in the majority of the problems addressed in this work.

The goal of this thesis is to generalize the adaptive method for solving multi-level multi-criteria linear programming problems. Chapter 3 presents a geometric study of solution stability in single-objective linear programming problems. Chapters 5 and 6 are dedicated to constructing procedures based on the principle of the adaptive method to solve multi-level linear programming problems. In Chapter 5, an advanced technical construction of an application is established to define a new sub-optimality estimate. These constructions are then utilized to develop two versions of an algorithm that solves single-objective multi-level linear programming problems. Lastly, in Chapter 6, we generalize the concept of non-dominated solutions in the context of multi-level linear problems and introduce a method to generate them. This new concept is employed to describe an algorithm that applies the adaptive method to achieve a unique compromise.

The obtained results demonstrate the effectiveness and relevance of the developed methods in solving multi-level linear programming problems. This thesis makes significant contributions to the resolution of complex problems and opens up new prospects for future research in this field.

Keywords : Linear programming, Multi-objective programming, Multi-level programming, Sensitivity analysis, Adaptive method, Sub-optimality estimate, Non-dominated solutions.



MULAKHAS

تركز هذه الأطروحة بشكل رئيسي على تطوير أساليب لحل مشاكل البرمجة الخطية متعددة المستويات، استنادًا إلى مبدأ الطريقة المتكيفة. يظل حل هذه المشاكل بفاعلية تحديًا معقدًا. تواجه هذه الصعوبة أيضًا في معظم المشاكل التي تمت مناقشتها في هذا العمل.

هدف هذه الأطروحة هو تعميم الطريقة المتكيفة لحل مشاكل البرمجة الخطية متعددة المعايير والمستويات. في الفصل 3، يتم تقديم دراسة هندسية لاستقرار الحلول لمشكلة البرمجة الخطية ذات المعيار الوحيد. تخصص الفصول 5 و 6 لبناء إجراءات استنادًا إلى مبدأ الطريقة المتكيفة لحل مشاكل البرمجة الخطية متعددة المستويات. في الفصل 5، يتم إنشاء تطبيق تقني متقدم لتحديد تقدير جديد للأداء الفرعي. يتم استخدام هذه البناءات لتطوير نسختين من خوارزمية لحل مشاكل البرمجة الخطية متعددة المستويات بهدف واحد. وأخيرًا، في الفصل 6، نعلم مفهوم الحلول غير المهيمنة في سياق المشاكل الخطية متعددة المستويات ونضع طريقة لتوليدها. يتم استخدام هذا المفهوم الجديد لوصف خوارزمية تطبق الطريقة المتكيفة لتحقيق توازن واحد. تؤكد النتائج المحصلة على فعالية وجدوى الأساليب المطورة في حل مشاكل البرمجة الخطية متعددة المستويات.

تعتبر هذه الأطروحة محاولة مهمة لتوسيع فهمنا لحلول مشاكل البرمجة الخطية متعددة المستويات وتطبيق الطريقة المتكيفة لتحقيق توازن مثلي. حيث تساهم في تطوير المنهجيات والتقنيات للتغلب على التحديات الصعبة في هذا المجال وتوسيع مجال البحث في هذا المجال، وتفتح آفاقًا جديدة للابتكار والتطوير المستقبلي.

الكلمات المفتاحية : البرمجة الخطية، البرمجة المتعددة الأهداف، البرمجة متعددة المستويات، تحليل الحساسية، الطريقة المتكيفة، تقدير العدم الأمثلية الجزئية، الحلول غير المهيمنة.



TABLE DES MATIÈRES

Remerciements	v
Résumé	vii
Abstract	viii
Mulakhas	ix
Table des matières	ix
Liste des figures	xiv
Liste des algorithmes	xiv
Liste des abréviations	xvi
Liste des symboles	xvii
Introduction générale	xix
I Optimisation Mono-Critère	1
1 Exploration Conceptuelle et Théorique : Fondements et Réflexions	3
1.1 Notations et Conventions pour les Matrices et les Vecteurs	4
1.1.1 Notations et Conventions pour les Vecteurs	4
1.1.2 Notations et Conventions pour les Matrices	5
1.2 Structures Fondamentales de \mathbb{R}^n : Exploration des Concepts	6
1.2.1 Géométrie Affine : Espace et Sous-Espaces Affines	7
1.2.2 Compréhension des Espaces Vectoriels Normés : Définitions et Illus- trations	12
1.2.2.1 Normes dans \mathbb{R}^n	12
1.2.2.2 Exemples de Normes sur \mathbb{R}^n	13



1.2.2.3	Espaces Vectoriels Normés et Norme Euclidienne	13
1.2.3	Exploration des Espaces Métriques : Définitions et Exemples	13
1.2.3.1	Définition d'un Espace Métrique	14
1.2.3.2	Exemples d'Espaces Métriques	14
1.2.4	Fondements des Espaces topologiques : Définitions et Exemplifications	15
1.3	Fondements de l'Analyse Convexe : Structures et Propriétés	18
1.3.1	Propriétés des Ensembles Convexes	18
1.3.2	Propriétés des Fonctions Convexes	22
1.4	Exploration des Polyèdres et des Polytopes	25
1.4.1	Polyèdres et Leurs Frontières : Une Brève Étude	25
1.4.2	Sous-ensemble Extrême et Sous-ensemble Exposé	30
2	Élégantes Approches pour Résoudre les Problèmes de Programmation Linéaire (PPLs)	34
2.1	Émergence dans le Monde de la Programmation Linéaire	35
2.1.1	Les Problèmes de Programmation Linéaire	35
2.1.2	Les Points Extrêmes et les Solutions Optimales	40
2.2	Méthodes de Recherche Linéaire dans les PPL	42
2.2.1	Propriétés des Méthodes de Recherche Linéaire	42
2.2.2	Comparaison avec d'Autres Méthodes d'Optimisation	42
2.2.2.1	Méthodes de Recherche Linéaire (y compris le Simplexe)	42
2.2.2.2	Méthodes de Gradient	43
2.2.2.3	Programmation Dynamique	43
2.2.2.4	Algorithmes d'Énumération	43
2.2.2.5	Méthodes Heuristiques	43
2.2.2.6	Métaheuristiques	43
2.2.3	La Notion de Métrique dans les Algorithmes d'Optimisation	44
2.2.3.1	Définition de la Métrique	44
2.2.3.2	La Métrique du Simplexe dans le Simplexe Classique	44
2.2.3.3	Utilisation de la Métrique dans l'Optimisation Linéaire à Variables Bornées	44
2.2.3.4	Métrique Adaptée du Simplexe dans la Méthode de Gabasov	44
2.3	Méthodes de Recherche Linéaire : Méthode du Simplexe et Méthode Adaptée	45
2.3.1	Transformation de la Forme Standard à la Forme Canonique d'un PPL	45
2.3.2	La Méthode du Simplexe	47
2.3.2.1	Le Fondement de l'Algorithme : Comprendre son Principe Fondamental	47
2.3.2.2	Exploration de l'Algorithme du Simplexe	47
2.3.2.3	Analyse de la Complexité du Simplexe et Perspectives d'Approches Alternatives	54
2.3.2.4	Exemple d'Application de la Méthode du Simplexe	54
2.3.3	La Méthode Adaptée	57
2.3.3.1	Estimation de Sous-optimalité	58
2.3.3.2	Le Fondement de la Méthode Adaptée	60
3	Une Approche Géométrique pour l'Analyse de Sensibilité	62



3.1	Introduction	63
3.2	Préliminaires	64
3.3	Formulation du Problème	66
3.4	Géométrie des Programmes Linéaires : Une Approche Visuelle	67
3.5	Analyse Géométrique de Sensibilité dans les Programmes Linéaires	70
3.6	Exemple d'Application	72
3.7	Conclusion	74
II Optimisation Multi-Critère Multi-Niveaux		75
4	Réflexions et Fondements Théoriques : Exploration des Concepts	77
4.1	Problèmes d'Optimisation Multi-Critère	78
4.1.1	Exploration des Terminologies	79
4.1.1.1	Les attributs	79
4.1.1.2	Les objectifs	79
4.1.1.3	Les buts	80
4.1.1.4	Les critères	80
4.1.2	À la Croisée des Objectifs : Une Exploration de la Programmation Mathématique Multi-Objectifs (PMMO) à travers le Cas d'un Achat de Voiture.	80
4.1.3	Approfondissement de la Programmation Linéaire Multi-Objectifs : Maximisation de Critères Multiples	81
4.1.4	Classification des Problèmes d'Optimisation Multi-Critère	81
4.1.5	Approche de Résolution des Problèmes de Programmation Linéaire Multi-Objectifs (PPLMOs)	85
4.1.6	La Méthode Adaptée pour la Recherche d'une Seule Solution Extrême Efficace d'un PPLMO à variable bornées	87
4.2	La Programmation Multi-Objectifs Multi-Niveaux	88
4.2.1	Approche de Résolution des Problèmes de Programmation Multi-Objectifs Multi-Niveaux (PPMO-MNs)	89
5	Une Approche Novatrice pour la Résolution d'un Problème de Programmation Linéaire MOno-Objectifs Multi-Niveaux (PPL(MO)O-MN)	91
5.1	Formulation du PPL(MO)O-MN	92
5.2	Formulation Linéaire du $p^{\text{ième}}$ Niveau	93
5.2.1	Construction de l'Application de Réduction des Intervalles	94
5.2.2	Estimation de Sous-Optimalité pour le $p^{\text{ième}}$ Niveau	100
5.3	Résolution du PPL(MO)O-MN (5.1) avec la Méthode Adaptée de Gabasov	102
5.3.1	Algorithme - Première Version	102
5.3.2	Illustration Numérique de la 1 ^{er} -Version de l'Algorithme	103
5.3.3	Algorithme - Deuxième Version	105
5.3.3.1	Explication de l'algorithme	106
5.3.4	Illustration Numérique de la 2 ^{ième} -Version de l'Algorithme	107
5.4	Conclusion	110



6	Exploration des Solutions de Compromis en Programmation Linéaire Multi-Objectifs Multi-Niveaux : La Méthode Adaptée de Gabasov	111
6.1	Génération Complète des Solutions de Compromis pour un Problème de Programmation Linéaire Multi-Objectifs Multi-Niveaux (PPLMO-MN) . . .	112
6.2	Formulation du PPLMO-MN	113
6.3	Préliminaires	114
6.4	Exploration des Solutions de Compromis dans un PPLMO-MN	116
6.4.1	Identification des Faces Non-dominées d'un PPLMO-MN	117
6.4.2	Élaboration de l'Algorithme : Processus Complet de Construction . . .	119
6.4.3	Exploration Exhaustive : L'Algorithme dans toute sa Splendeur . . .	122
6.4.4	Première Illustration Numérique : Mettre en Action l'Algorithme 6 . . .	125
6.5	Résolution des PPLMO-MNs via une Approche Novatrice et Constructive . .	128
6.5.1	Édification de l'Algorithme : De la Conception à la Réalisation . . .	128
6.5.2	Illustration Numérique du PPLMO à deux-niveaux (Section 6.4.4) : Étude de Cas à travers l'Algorithme 7	130
6.6	Dialogue Approfondi : Exploration et Analyses	131
6.6.1	La Méthode Adaptée vs La Méthode du Simplexe : Comparaison et Choix	131
6.6.2	Processus Étape par Étape pour Résoudre les PPLMO-MNs à Variables Bornées : La Méthode Adaptée	132
6.7	Conclusion	135
	Une Analyse Approfondie, des Conclusions Éclairées et des Perspectives Raffinées	136
1	Conclusions de l'Étude	136
2	Au cœur des débats : Une analyse approfondie et critique des résultats obtenus	138
3	Bilan et perspectives : Les conclusions clés de notre étude en optimisation multi-critère et les pistes pour de futures recherches	139
	Suppléments et Annexes	140
I	Génération de l'Ensemble N^{dex} : Méthode de P.L. Yu et M. Zeleny	140
II	Une Nouvelle Approche Géométrique pour les PPLMOs	142
II.1	Introduction	143
II.2	Formulation du Problème	143
II.3	Classification des problèmes de Programmation Linéaire Multi-Objectifs (PLMO)	145
II.4	Illustration Numérique d'un PPLMO en Utilisant les Calculs de l'Exemple 3.6 du Chapitre 3 : Classification des Formes Linéaires par Rapport à la Région Réalisable S définie par 28	146
II.5	Conclusion	147
III	Exemple pour Illustrer les Faces d'un Polyèdre dans \mathbb{R}^3	148
	Liste des publications	149
	Bibliographie	151



LISTE DES FIGURES

1	George Dantzig	xx
2	Rafail Gabasov	xxi
1.1	Hyperplans affine dans \mathbb{R}^2	12
1.2	Représentation des boules unités associées aux normes $\ \cdot\ _k$ dans \mathbb{R}^2	17
1.3	Ensembles convexes et non-convexes	20
1.4	Cône dans \mathbb{R}^3	20
1.5	Illustration du graphe, de l'hypographe et de l'épigraphe de la fonction $f(x, y) = x^2 + y^2$ dans \mathbb{R}^3 pour $\alpha = 3$	23
1.6	Josef Stoer	25
1.7	Christoph Witzgall	25
1.8	Hyperplan frontière	28
1.9	Points extrêmes d'un polytope dans \mathbb{R}^2	30
2.1	Directions admissibles	48
2.2	Résolution du problème (2.12) avec la méthode du simplexe	56
3.1	Projection d'un segment sur une droite.	66
3.2	Projection orthogonale sur les sous-espaces vectoriels de \mathbb{R}^3	67
3.3	Illustration graphique du lemme (3.4.2)	68
3.4	Les points extrêmes voisins à la solution optimale x^0	71
3.5	Illustration graphique de la proposition (3.5.3).	72
6.1	Illustration de l'ensemble \hat{N}^{dex} pour $P = 2$ et $S \subset \mathbb{R}^3$	118
6.2	Schéma de l'Algorithme 6	124
6.3	Ensemble des solutions du PPLMO (1)	126
6.4	Ensemble des solutions de PPLMO (2)	126
6.5	L'ensemble de tous les compromis possible du PPLMO à deux niveaux (6.11).	127
6	Face de dimension 0 (point extrême).	147
7	Face vide.	147
8	Face de dimension 1.	148
9	Face de dimension 2.	148



LISTE DES ALGORITHMES

1	Algorithme du simplexe mono-critère	53
2	Algorithme de la méthode adaptée mono-critère	61
3	La Méthode Adaptée de Programmation Linéaire pour la Résolution d'un PPLMO à Variables Bornées.	87
4	Résolution du PPL(MO)O-MN (5.1) - Première Version	102
5	Résolution du PPL(MO)O-MN (5.1)-Deuxième Version- Partie 1	105
6	Génération de l'Ensemble des solutions Non-dominées \hat{N}	123
7	Algorithme Complet : Génération d'un seul Compromis pour un PPLMO-MN .	130



LISTE DES ABRÉVIATIONS

PPLMO	Problème de Programmation Linéaire Multi-Objectifs
PPLMO-MN	Problème de Programmation Linéaire Multi-Objectifs Multi-Niveaux
PPL(MO)O-MN	Problème de Programmation Linéaire MOno-Objectifs Multi-Niveaux
PL	Programmation Linéaire
PPL	Problème de Programmation Linéaire
PMO	Programmation Multi-Objectif
PPMO	Problème de Programmation Multi-Objectif
PLMO	Programmation Linéaire Multi-Objectifs
OMC	Optimisation Multi-Critère
POMC	Problème d'Optimisation Multi-Critère
PM	Programmation Mathématique
PMMO	Programmation Mathématique Multi-Objectifs
PPMO-MN	Problème de Programmation Multi-Objectifs Multi-Niveaux
DCD	Décideur
DCD₁	Décideur du 1 ^{er} niveau
DCD₂	Décideur du 2 ^{ième} niveau
DCD₃	Décideur du 3 ^{ième} niveau
DCD₄	Décideur du 4 ^{ième} niveau
DCD₅	Décideur du 5 ^{ième} niveau
DCD_{p-1}	Décideur du $(p - 1)$ ^{ième} niveau
DCD_p	Décideur du p ^{ième} niveau
ARI	Application de Réduction des Intervalles



LISTE DES SYMBOLES

J_B	:	Ensemble des indices basiques.
J_N	:	Ensemble des indices non-basiques.
$ J_B $:	Le cardinal de l'ensemble J_B .
\mathbb{R}^n	:	Ensemble des vecteurs réels n -dimensionnels.
c^t	:	Transposée du vecteur c .
$\langle \cdot, \cdot \rangle$:	Un produit scalaire sur \mathbb{R}^n .
$\ \cdot\ $:	Une norme sur \mathbb{R}^n .
$ \alpha $:	Valeur absolue d'un élément α de \mathbb{R} .
$Aff(F)$:	Enveloppe affine d'un sous-ensemble F de \mathbb{R}^n .
$dim(F)$:	Dimension d'un sous-ensemble F de \mathbb{R}^n .
\mathcal{H}_κ	:	Hyperplan vectoriel de \mathbb{R}^n .
$\mathcal{H}_{\kappa r}$:	Hyperplan affine de \mathbb{R}^n .
$L(\mathbb{R}^n, \mathbb{R}^m)$:	Ensemble de toutes les applications linéaires de \mathbb{R}^n dans \mathbb{R}^m .
$\mathcal{L}(\mathbb{R}^n)$:	Ensemble de toutes les formes linéaires définies sur \mathbb{R}^n .
$\mathcal{P}(\mathbb{R}^n)$:	L'ensemble des parties de \mathbb{R}^n .
$\tau_{\mathbb{R}^n}$:	Une topologie sur \mathbb{R}^n .
$\overset{\circ}{E}$:	L'intérieur topologique d'un ensemble E .
\overline{E}	:	L'adhérence ou fermeture topologique d'un ensemble E .
$c\hat{o}nne(S)$:	L'enveloppe conique d'un sous-ensemble S de \mathbb{R}^n .
$ri(S)$:	L'intérieur relatif d'un sous-ensemble S de \mathbb{R}^n .
$Gf(f)$:	Le graphe de la fonction f .
$epi(f)$:	L'épigraphe de la fonction f .
$hyp(f)$:	L'hypographe de la fonction f .
$Vect(n_f)$:	Le sous-espace vectoriel de dimension 1 engendré par le vecteur n_f , formant ainsi une droite dans l'espace.



$Rot(\theta)$:	Matrice de rotation dans \mathbb{R}^2 d'angle θ .
T_v	:	Application de translation dans \mathbb{R}^2 dans la direction du vecteur v .
$arg\ max\ f$:	L'ensemble des valeurs de x pour lesquelles l'application f atteint son maximum.
$Pr_{(d)}$:	Une projection (ou projecteur) qui associe à tout élément x de \mathbb{R}^2 un autre élément $P_{(d)}(x)$ appartenant à la droite (d) .
$grad(f)$:	Le gradient de l'application f .
E^\perp	:	Le complément orthogonal du sous-espace vectoriel E .
$F(Q)$:	Une face d'une région réalisable bornée S d'un PPL.
$\partial F(Q)$:	Frontière de la face $F(Q)$.
$Card(Q)$:	Le cardinal de l'ensemble Q .
S^{dex}	:	Points extrêmes d'une région réalisable bornée S d'un PPLMO.
D_p	:	Ensemble de tous les points dominés du $p^{ième}$ PPLMO qui constitue le PPLMO-MN.
N_p	:	Ensemble de tous les points non-dominés du $p^{ième}$ PPLMO qui constitue le PPLMO-MN.
N_p^{dex}	:	Ensemble de tous les points extrêmes non-dominés du $p^{ième}$ PPLMO qui constitue le PPLMO-MN.
$N_p(Q)$:	Face non-dominée du $p^{ième}$ PPLMO qui constitue le PPLMO-MN.
\hat{N}	:	Ensemble de tous les compromis possibles (solutions réalisables non-dominées) d'un PPLMO-MN.
\hat{N}^{dex}	:	Ensemble de tous les points extrêmes non-dominés d'un PPLMO-MN.
$\hat{N}(Q)$:	Face non-dominée d'un PPLMO-MN.
\mathcal{SP}	:	Espace de tri d'un PPLMO-MN.
$\mathcal{H}(U)$:	L'enveloppe convexe des éléments de U .
\mathcal{R}	:	Une relation d'équivalence.
\mathcal{R}_S	:	Une relation d'équivalence qui dépend de l'ensemble S .

INTRODUCTION GÉNÉRALE

Divers domaines tels que la physique, l'industrie, la biologie, l'économie et d'autres, modélisent plusieurs problèmes de décision par le biais de la programmation mathématique. La coordination d'activités interreliées est requise dans des contextes tels que la production industrielle, le flux des ressources économiques, ou encore l'effort militaire en temps de guerre et la gestion financière. Ces problèmes nécessitent la conception d'un programme ou d'un calendrier qui, s'il était mis en œuvre, permettrait de faire évoluer le système depuis un état initial donné vers un objectif spécifique. Malgré les différences d'objectifs à atteindre, de processus spécifiques et d'ampleur de l'effort, ces systèmes, lorsqu'ils sont modélisés mathématiquement, présentent souvent une structure remarquablement similaire. La tâche consiste alors à concevoir un algorithme pour choisir le meilleur calendrier d'actions parmi les alternatives possibles, et les méthodes d'optimisation sont utilisées pour résoudre ces problèmes [Bra73; Bra74; Dan03; Ehr05; Ant16; Bel18].

Initialement, les problèmes étaient modélisés avec un seul décideur cherchant une solution optimale pour une seule fonction objectif [Wol59; Dan03; Gab78; Gab80; Dan63; Rad11b]. Par la suite, on a découvert des problèmes dans lesquels un seul décideur cherchait à optimiser plusieurs objectifs souvent en conflit, donnant ainsi lieu aux problèmes de programmation multi-objectifs. Il a donc été nécessaire de définir une notion de résolution pour développer des méthodes de solution appropriées [Yu 74; Yu 75; Yu74; Del18b; Rad14; Rad11a; Ant16; Ehr05]. D'autres problèmes de prise de décision surviennent lorsque plusieurs décideurs sont impliqués, chacun cherchant à résoudre un problème avec plusieurs objectifs. Ces types de problèmes sont modélisés par des problèmes de programmation multi-objectifs multi-niveaux [Kac22c; Pie11; Sin04; Bak10; Lu 16; Shi96; Yib19; Anz15; Abo10].

La programmation linéaire, découverte en 1947 par le célèbre mathématicien George B. Dantzig, est l'un des modèles phares de la recherche opérationnelle et constitue le domaine le plus étudié de la programmation mathématique [Dan97; Gab78; Kol95; Dan63]. Ce domaine concerne l'optimisation d'un programme mathématique où la fonction objectif et les fonctions



définissant les contraintes sont linéaires. Géométriquement, les problèmes de programmation linéaire sont des problèmes de programmation convexe, où les contraintes linéaires forment un polyèdre convexe. Ainsi, les résultats de la convexité sont exploités, et les sommets du polyèdre convexe correspondent aux solutions de base, parmi lesquelles se trouve éventuellement la solution optimale.



George Dantzig

La programmation linéaire et sa généralisation, la programmation mathématique, font partie d'un développement révolutionnaire qui a permis à l'humanité de formuler des objectifs généraux et de tracer un chemin de décisions détaillées pour atteindre de manière plus efficace ces objectifs, même dans des situations pratiques d'une grande complexité. Les outils nécessaires pour accomplir cela sont les modèles, qui expriment les problèmes du monde réel en termes mathématiques détaillés, les algorithmes, qui résolvent ces modèles, et les logiciels qui

exécutent ces algorithmes sur des ordinateurs, en s'appuyant sur la théorie mathématique.

Les premières applications industrielles de la programmation linéaire, et les plus fructueuses, ont été observées dans l'industrie pétrolière, notamment dans l'extraction, le raffinage, le mélange et la distribution du pétrole. Le secteur de la transformation des aliments est un autre utilisateur actif de la programmation linéaire, utilisée par exemple pour optimiser l'expédition du ketchup depuis quelques usines vers de nombreux entrepôts, ou pour déterminer le mélange d'ingrédients le plus économique pour la production de saucisses et d'aliments pour animaux.

L'analyse de sensibilité, également appelée analyse post-optimalité, est une composante essentielle de la programmation linéaire. Elle part de la solution optimale initiale et permet d'étudier les variations dans les données du problème et d'illustrer les marges des paramètres du programme linéaire pour lesquelles la solution du problème initial reste stable. L'analyse de sensibilité est également utile pour évaluer l'impact des erreurs dans les données du problème, évitant ainsi de recommencer la procédure de résolution à plusieurs reprises [Sha16; Dae78].

Les modèles de programmation linéaire sont utilisés pour résoudre divers problèmes concrets, tels que la maximisation des bénéfices d'une entreprise. Cependant, les conditions du marché peuvent évoluer, nécessitant ainsi des mises à jour des problèmes initiaux. C'est dans ce contexte que l'analyse de sensibilité joue un rôle crucial. Elle permet d'illustrer les marges de variation des paramètres du programme linéaire pour lesquelles la solution du problème initial demeure stable.

L'analyse de sensibilité offre également une perspective sur la stabilité d'une solution optimale face aux erreurs dans les données du problème. En identifiant les paramètres sensibles et en évaluant leurs marges de variation admissibles, il est possible de limiter les ajustements nécessaires et d'éviter de recommencer la procédure de résolution à plusieurs reprises.

Ainsi, la programmation linéaire, avec son analyse de sensibilité, constitue un outil puissant pour aborder les problèmes complexes de prise de décision dans divers domaines industriels

et économiques. Son utilisation permet de formuler les objectifs de manière mathématique, de trouver des solutions optimales et de prendre des décisions éclairées, contribuant ainsi à l'efficacité et à la performance des systèmes économiques et industriels.

L'optimisation multi-niveaux se révèle être un outil particulièrement approprié pour modéliser les problèmes de prise de décision impliquant plusieurs décideurs interagissant au sein d'une structure hiérarchique. Mathématiquement, dans un problème de programmation multi-niveaux, les variables au niveau supérieur sont considérées comme des paramètres dans les problèmes de programmation au niveau inférieur, tandis que les problèmes au niveau supérieur sont perçus comme des contraintes pour les problèmes au niveau inférieur.

Les problèmes de programmation multi-niveaux trouvent de nombreuses applications dans divers domaines tels que la gestion de la chaîne d'approvisionnement, la cybersécurité, la planification, la logistique, l'économie, les institutions gouvernementales et autonomes, l'agriculture, l'armée, la gestion d'établissements scolaires, hospitaliers et bancaires, pour n'en citer que quelques-uns. Bien que la plupart des recherches portant sur les programmes à niveaux multiples se soient concentrées sur des cas à deux niveaux (appelés programmes à deux niveaux), il existe de nombreux problèmes de programmation qui impliquent plus de deux niveaux.

Cette approche de l'optimisation multi-niveaux offre un cadre puissant pour analyser et résoudre les problèmes de prise de décision complexes dans lesquels les interactions entre les différents niveaux hiérarchiques sont essentielles. Elle permet de modéliser des structures organisationnelles complexes et de prendre en compte les relations de dépendance et les contraintes entre les décideurs.

La méthode adaptée de programmation linéaire, décrite par R. Gabasov, F.M. Kirillova et O.I. Kostyukova à la fin des années 1980, est une méthode numérique constructive qui a été généralisée pour développer de nombreuses approches dans divers domaines de la programmation linéaire [Gab95; Gab78]. Elle a été étendue pour traiter des problèmes de programmation linéaire par morceaux, de programmation quadratique [Rad07c; Rad07b; Rad98; Rad12; Bra10; Del18a; Kac21b], de programmation polynomiale, de contrôle optimal [Bib85; Bib96], de programmation linéaire multi-objectif [Rad11a; Rad14; Del18b; Rad04] et de programmation linéaire multi-niveau, comme c'est le cas dans cette étude.



Rafail Gabasov

Cette méthode adaptée de programmation linéaire représente une avancée significative dans le domaine des méthodes numériques et a permis de développer des approches efficaces pour résoudre des problèmes d'optimisation complexes. Elle offre des perspectives prometteuses pour la modélisation et la résolution de problèmes linéaires multi-objectifs et multi-niveaux, où les contraintes et les interactions entre les différents niveaux constituent des défis majeurs.

Depuis les travaux pionniers de Bracken et McGill [Bra73; Bra74], de nombreux chercheurs ont consacré leurs travaux à l'étude de l'optimisation à deux niveaux, en abordant les aspects



théoriques et méthodologiques de ce domaine. Des monographies et des revues de littérature ont été publiées, offrant une exploration approfondie des problèmes et des méthodes associées à l'optimisation multi-niveaux. Une variété d'approches ont été développées pour résoudre ces problèmes de programmation multi-niveaux.

Pour une bibliographie complète sur ces problèmes et leurs applications, il est recommandé de se référer à [Lu 16], où des travaux pertinents sont recensés et analysés de manière approfondie. Cette ressource fournit un aperçu complet des avancées dans le domaine de l'optimisation multi-niveaux, permettant aux chercheurs et aux praticiens d'accéder à des références importantes et de suivre les développements récents dans ce domaine.

Le principal objectif de cette thèse est d'appliquer le principe de la méthode adaptée pour résoudre des problèmes d'optimisation multi-critère à plusieurs niveaux. Nous commençons par l'application de cette méthode aux problèmes de programmation linéaire à plusieurs niveaux avec un seul objectif sur chaque niveau, en nous appuyant sur l'approche développée par S.B. Sinha et S. Sinha, et en construisant une application qui choisit les intervalles réduits des variables de décision. Cela nous permet de généraliser la méthode adaptée.

Nous étendons ensuite notre approche à des problèmes plus généraux, c'est-à-dire l'application de la méthode adaptée à la résolution de problèmes d'optimisation multi-critères à plusieurs niveaux. Pour ce faire, nous avons introduit un nouveau concept essentiel, à savoir l'ensemble de tous les compromis possibles. Nous décrivons un algorithme permettant de générer cet ensemble, qui joue le rôle d'une nouvelle région réalisable. À partir de cet ensemble, nous effectuons la recherche d'un compromis unique en utilisant la méthode adaptée.

En outre, nous développons une méthode pratique pour l'analyse de sensibilité en deux dimensions des variables de décision. Cette méthode peut être utilisée comme un test pour détecter la présence d'une hiérarchie dans les problèmes de programmation multi-niveaux, ainsi que pour identifier les conflits dans les problèmes de programmation multi-objectifs.

Pour cela, ce travail est divisé en deux volets. La première partie de la thèse présente une nouvelle approche géométrique pour l'analyse de sensibilité. Le chapitre 1 de cette thèse propose une compilation de notions et d'outils mathématiques qui interviennent de manière explicite ou implicite dans la programmation linéaire. Il aborde notamment la notion de convexité, les espaces affines, les espaces topologiques, ainsi que certaines propriétés des polyèdres. Cette partie fournit un cadre conceptuel solide pour comprendre les fondements géométriques de la programmation linéaire et de l'analyse de sensibilité. Elle établit les bases nécessaires pour développer nos approches novatrices.

Le chapitre 2 constitue une introduction à la programmation linéaire mono-objectif. Il fournit les préliminaires de base nécessaires pour établir les résultats présentés par la suite. Ce chapitre expose le principe de construction de deux méthodes numériques pour la résolution de problèmes de programmation linéaire avec une seule fonction objectif : la célèbre méthode du simplexe et la méthode adaptée.

La méthode du simplexe est largement connue et utilisée dans le domaine de la programmation



linéaire. Elle repose sur des techniques d'exploration des sommets d'un polyèdre convexe afin de trouver la solution optimale. Nous décrivons les principes fondamentaux de cette méthode et mettons en évidence ses avantages et ses limitations.

En parallèle, nous présentons la méthode adaptée qui constitue une alternative prometteuse pour la résolution des problèmes de programmation linéaire mono-objectif. Cette méthode, développée dans le cadre de cette recherche, repose sur des principes novateurs et offre des perspectives intéressantes pour améliorer l'efficacité de la résolution des problèmes de programmation linéaire.

Le chapitre 3 propose une analyse géométrique de la sensibilité des solutions d'un problème de programmation linéaire mono-objectif par rapport aux variations des données initiales du programme, plus précisément les variations des coefficients de la fonction objectif. Cette analyse est effectuée dans un espace à deux dimensions, où la région réalisable du problème est représentée.

Cette analyse de sensibilité offre des perspectives intéressantes pour détecter si un problème de programmation linéaire multi-objectifs multi-niveaux possède un compromis idéal ou non. En d'autres termes, elle permet de déterminer si un problème de programmation linéaire multi-objectifs présente un conflit ou si un problème de programmation linéaire multi-niveaux possède une hiérarchie entre les niveaux.

L'efficacité de cette méthode réside dans sa capacité à détecter l'absence de conflit dans un problème de programmation linéaire multi-objectifs, ainsi que la présence d'une hiérarchie dans un problème de programmation linéaire multi-niveaux. Cette analyse géométrique de sensibilité fournit donc des informations précieuses sur la structure du problème et permet de mieux comprendre les relations entre les objectifs et les niveaux hiérarchiques.

La seconde partie de cette thèse se concentre sur l'optimisation multi-critère. Dans le chapitre 4, nous commençons par présenter la définition générale des problèmes d'optimisation multi-critère, ainsi que leur classification et les principes de résolution associés. Nous soulignons notamment, dans la section 4.1.4, que résoudre de tels problèmes revient à faire un choix parmi différentes classes. En d'autres termes, il s'agit de sélectionner une fonction objectif, une région réalisable, une application-modèle et un ordre sur l'espace des critères.

Nous accordons une attention particulière à la présentation des problèmes multi-critères, en mettant en évidence leurs spécificités et en expliquant les différents concepts et méthodes associés à leur résolution. Nous examinons également les problèmes multi-critères à plusieurs niveaux, qui ajoutent une dimension supplémentaire en termes de hiérarchie et d'interactions entre les niveaux.

Cette partie de la thèse vise à offrir une vue d'ensemble approfondie des problèmes d'optimisation multi-critère, en fournissant des outils conceptuels et méthodologiques essentiels pour aborder ces problèmes complexes. Elle permet de comprendre les enjeux et les défis liés à la recherche de solutions optimales dans des contextes où plusieurs critères doivent être pris en compte.



Dans le chapitre 5, nous abordons les problèmes de programmation linéaire mono-objectifs multi-niveaux. Nous nous concentrons sur la construction d'une application de réduction des intervalles que nous avons établie. Cette application joue un rôle essentiel dans la définition d'une nouvelle estimation de sous-optimalité, qui est un élément indispensable caractérisant la méthode adaptée.

La construction de cette application de réduction des intervalles constitue la clé pour l'application réussie de la méthode adaptée à la résolution des problèmes de programmation linéaire mono-objectifs à plusieurs niveaux, même lorsque les variables ne sont pas bornées. Cette avancée permet d'étendre l'applicabilité de la méthode adaptée à des situations plus générales et ouvre de nouvelles perspectives pour la résolution efficace de problèmes complexes.

Dans le chapitre 6, nous entamons une construction méticuleuse de deux méthodes constructives. Tout d'abord, dans la section 6.4, nous adaptons la méthode du simplexe multi-critère généralisé développée par P.L. Yu et M. Zeleny pour générer l'ensemble de tous les compromis possibles dans le contexte d'un problème de programmation linéaire multi-objectifs à plusieurs niveaux. Cette adaptation nous permet de traiter efficacement ces problèmes complexes en identifiant les solutions optimales ainsi que les compromis entre les différents niveaux hiérarchiques. Ensuite, dans la section 6.5, nous construisons une extension de la méthode adaptée pour le cas des problèmes multi-niveaux. Nous nous appuyons sur la méthode précédente pour généraliser la méthode adaptée et ainsi résoudre efficacement les problèmes de programmation linéaire multi-objectifs multi-niveaux. Cette extension de la méthode adaptée nous permet de prendre en compte les interactions et les contraintes spécifiques liées à la structure hiérarchique des problèmes multi-niveaux.

Finalement, la thèse s'est clôturée par un chapitre de conclusion qui a récapitulé les principales contributions et résultats obtenus. Ce chapitre a également inclus des orientations de recherche futures, mettant en évidence les pistes à explorer pour approfondir les connaissances et développer de nouvelles avancées dans le domaine de l'optimisation multi-objectifs multi-niveaux.

Dans l'annexe, nous avons fourni une liste des différentes publications qui ont émergé de cette thèse. En particulier, nous avons mentionné les articles [Kac22a ; Kac22c] qui sont directement liés à l'analyse de sensibilité en programmation linéaire et en programmation linéaire multi-niveaux. Ces publications témoignent de la contribution scientifique de cette thèse et de son impact sur le domaine de la recherche opérationnelle.

PREMIÈRE PARTIE:

OPTIMISATION MONO-CRITÈRE

Au sein de cette partie, les deux premiers chapitres mettent en lumière les outils fondamentaux issus de la littérature traitant de la programmation linéaire mono-critère. Ces outils, incontournables pour la réalisation fructueuse des travaux menés dans le cadre de cette thèse, se dévoilent sous un jour privilégié. Parallèlement, le chapitre 3 révèle notre contribution novatrice à l'analyse de la sensibilité des solutions dans le contexte des programmes linéaires. Cette contribution apporte une nouvelle et substantielle perspective à ce domaine.

CHAPITRE 1

EXPLORATION CONCEPTUELLE ET THÉORIQUE : FONDEMENTS ET RÉFLEXIONS

Objectifs

Dans ce chapitre, nous abordons de manière concise les définitions de quelques notions et propriétés importantes. Nous commençons par établir certaines notations pour les vecteurs et les matrices. Ensuite, nous présentons quelques structures fondamentales de \mathbb{R}^n , qui seront utilisées de manière implicite et explicitée ultérieurement. Par la suite, nous explorons quelques concepts classiques de l'analyse convexe, tels que les ensembles et les fonctions convexes. Enfin, la dernière partie de ce chapitre est dédiée aux outils indispensables de la théorie des polyèdres, qui jouent un rôle essentiel dans notre étude.

Sommaire

1.1	Notations et Conventions pour les Matrices et les Vecteurs	4
1.1.1	Notations et Conventions pour les Vecteurs	4
1.1.2	Notations et Conventions pour les Matrices	5
1.2	Structures Fondamentales de \mathbb{R}^n : Exploration des Concepts	6
1.2.1	Géométrie Affine : Espace et Sous-Espaces Affines	7
1.2.2	Compréhension des Espaces Vectoriels Normés : Définitions et Illustrations . .	12
1.2.3	Exploration des Espaces Métriques : Définitions et Exemples	13
1.2.4	Fondements des Espaces topologiques : Définitions et Exemplifications	15
1.3	Fondements de l'Analyse Convexe : Structures et Propriétés	18
1.3.1	Propriétés des Ensembles Convexes	18



1.3.2	Propriétés des Fonctions Convexes	22
1.4	Exploration des Polyèdres et des Polytopes	25
1.4.1	Polyèdres et Leurs Frontières : Une Brève Étude	25
1.4.2	Sous-ensemble Extrême et Sous-ensemble Exposé	30

1.1 Notations et Conventions pour les Matrices et les Vecteurs

1.1.1 Notations et Conventions pour les Vecteurs

Nous notons par \mathbb{R}^n l'ensemble des vecteurs réels n -dimensionnels. Pour tout vecteur $x \in \mathbb{R}^n$, nous utilisons x_i pour représenter sa $i^{\text{ème}}$ coordonnée, également appelée $i^{\text{ème}}$ composante. Pour un entier $m \in \mathbb{N}_*$, nous définissons les ensembles d'indices suivants :

$$\begin{aligned} I &= \{1, 2, \dots, m\}, & J &= \{1, 2, \dots, n\}, & J &= J_B \cup J_N, \\ J_B \cap J_N &= \emptyset, & |J_B| &= m, & |J_N| &= n - m. \end{aligned}$$

Cela nous permet d'exprimer la partition de x comme suit :

$$\begin{aligned} x &= x(J) = (x_j, j \in J), & x &= (x_N, x_B), \\ x_N &= x(J_N) = (x_j, j \in J_N), & x_B &= x(J_B) = (x_j, j \in J_B). \end{aligned} \tag{1.1}$$

Les vecteurs de \mathbb{R}^n sont considérés comme des vecteurs colonne, sauf indication contraire explicite. Pour tout $x \in \mathbb{R}^n$, x^t représente la transposition de x , qui est un vecteur ligne n -dimensionnel.

Le produit scalaire de deux vecteurs $x, y \in \mathbb{R}^n$ est défini par :

$$\langle x^t, y \rangle = \sum_{i=1}^n x_i y_i.$$

Deux vecteurs $x, y \in \mathbb{R}^n$ tels que $\langle x, y \rangle = 0$ sont appelés orthogonaux. Pour un vecteur x dans \mathbb{R}^n , les notations $x > 0$ et $x \geq 0$ indiquent que toutes les composantes de x sont positives et non négatives, respectivement. La notation $x > y$ signifie que $x - y > 0$ pour deux vecteurs x et y . Les notations $x \geq y$, $x < y$, $x \leq y$, etc., sont interprétées de manière similaire.



1.1.2 Notations et Conventions pour les Matrices

Une matrice A d'ordre $m \times n$ à coefficients réels, notée $A = A(I, J)$, est représentée sous la forme suivante :

$$A = A(I, J) = (a_{ij}, i \in I, j \in J) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

où I et J désignent respectivement l'ensemble des indices des lignes et des colonnes de A , et a_{ij} représente le $ij^{\text{ème}}$ élément de la matrice. Une matrice est dite carrée si elle a le même nombre de lignes et de colonnes (c'est-à-dire $n = m$).

Pour faciliter les calculs pratiques, nous introduisons la notation suivante pour A :

$$A = (a_1, a_2, \dots, a_j, \dots, a_n) = \begin{pmatrix} A_1^t \\ A_2^t \\ \vdots \\ A_i^t \\ A_{i+1}^t \\ \vdots \\ A_m^t \end{pmatrix},$$

où

$$a_j = A(I, j) = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{ij} \\ \vdots \\ a_{mj} \end{pmatrix}$$

est un vecteur-colonne de dimension m et peut être considéré comme une matrice d'ordre $(m \times 1)$, et

$$A_i^t = (a_{i1} \ a_{i2} \ \dots \ a_{in}), i \in I$$

est un vecteur-ligne de dimension n et peut être considéré comme une matrice d'ordre $(1 \times n)$. Par conséquent, nous pouvons écrire la décomposition de la matrice A comme suit :

$$A = (A_N, A_B), \quad A_N = A(I, J_N), \quad A_B = A(I, J_B). \quad (1.2)$$



Propriété 1.1.1.

1. La transposée de A , notée A^t , est définie comme suit : $a_{ij}^t = a_{ji}$ selon la notation précédente : $A^t = A(J, I) = (a_{ji}, j \in J, i \in I)$.
2. Pour deux matrices A et B de dimensions compatiblesⁱ, la transposée du produit matriciel AB satisfait $(AB)^t = B^t A^t$.
3. Le rang de A est la dimension de l'espace colonneⁱⁱ de A , et il est égal au nombre maximal de colonnes linéairement indépendantesⁱⁱⁱ de A . Le rang est également égal au nombre maximal de lignes linéairement indépendantes de A .
4. La matrice A et sa transposée A^t ont le même rang.
5. On dit que A a un rang complet si son rang est égal à $\min\{m, n\}$. Cela est vrai si et seulement si toutes les lignes de A sont linéairement indépendantes ou si toutes les colonnes de A sont linéairement indépendantes.
6. Soit A une matrice carrée.
 - (a) On dit que A est symétrique si $A^t = A$.
 - (b) On dit que A est diagonale si $a_{ij} = 0$ pour tout $i \neq j$.
 - (c) La matrice identité est notée Id_A et représente une matrice diagonale dont les éléments diagonaux sont égaux à 1.
 - (d) Le déterminant de A est noté $\det(A)$.
 - (e) L'espace colonne de A , noté $R(A)$, est l'ensemble de tous les vecteurs $y \in \mathbb{R}^m$ tels que $y = Ax$ pour certains $x \in \mathbb{R}^n$.
 - (f) L'espace nul de A , noté $N(A)$, est l'ensemble de tous les vecteurs $x \in \mathbb{R}^n$ tels que $Ax = 0$.

1.2 Structures Fondamentales de \mathbb{R}^n : Exploration des Concepts

L'ensemble \mathbb{R}^n peut être doté de différentes structures, et l'utilisation de ses éléments dépend de la structure qui lui est associée. Chaque structure apporte de nouvelles propriétés qui enrichissent l'espace et permettent différentes exploitations. Parmi les structures couramment étudiées, on trouve l'espace affine, l'espace métrique, l'espace vectoriel, l'espace vectoriel normé (c'est-à-dire un espace vectoriel muni d'une norme) et l'espace topologique, pour n'en citer que quelques-unes.

i. Le nombre de colonnes de la matrice A égal au nombre de lignes de la matrice B .
ii. Le sous-espace vectoriel engendré par les vecteurs colonnes de A , voir la définition 1.2.1.
iii. Voir la définition 1.2.2



Dans la suite, nous mettons l'accent sur les structures qui nous intéressent particulièrement, à savoir la structure d'espace vectoriel normé et celle d'espace topologique. Ces structures fournissent des propriétés essentielles pour notre étude et ouvrent la voie à diverses applications.

1.2.1 Géométrie Affine : Espace et Sous-Espaces Affines

Dans l'espace \mathbb{R}^n , la manipulation géométrique des points est souvent perçue comme plus intuitive que la manipulation des vecteurs. Cependant, dans le cadre formel, nous adoptons une approche contraire où les vecteurs sont privilégiés en raison de leur linéarité, ce qui facilite les opérations (par exemple, l'existence d'un zéro absolu). Les points, quant à eux, sont considérés de manière plus abstraite.

Dans l'espace, nous traitons tous les points de manière égale, en omettant la spécificité du zéro (ou inversement, en considérant que tout point peut être le zéro). Dans les formules, nous ajoutons des termes constants aux combinaisons linéaires, correspondant ultérieurement à des translations dans l'espace.

Cette approche permet d'unifier le traitement des points et des vecteurs, offrant ainsi une cohérence mathématique et simplifiant les manipulations algébriques dans \mathbb{R}^n .

Définition 1.2.1.

1. Un sous-ensemble non vide F de \mathbb{R}^n contenant $0_{\mathbb{R}^n}$ est appelé un sous-espace vectoriel si $\alpha x + \beta y \in F$ pour tout $x, y \in F$ et tout $\alpha, \beta \in \mathbb{R}$.
2. Le sous-espace vectoriel engendré par une famille finie x^1, \dots, x^k d'éléments de \mathbb{R}^n est l'ensemble de tous les vecteurs y de la forme $y = \sum_{i=1}^k \alpha_i x^i$, où chaque α_i est un scalaire.
3. Un sous-espace affine dans \mathbb{R}^n est un sous-espace vectoriel translaté, c'est-à-dire un ensemble E de la forme $E = x + F = \{x + y : y \in F\}$, où x est un vecteur dans \mathbb{R}^n et F est un sous-espace vectoriel de \mathbb{R}^n . Le sous-espace vectoriel translaté est appelé la direction de F .



Il ne peut y avoir qu'un seul sous-espace vectoriel F associé à un sous-espace affine. Pour le voir, supposons que $E = x + F$ et $E = \hat{x} + \hat{F}$ soient deux représentations du même sous-espace affine E .

Ensuite, nous avons $x = \hat{x} + \acute{s}$ pour certains $\acute{s} \in \hat{F}$ (car $x \in E$), ce qui implique que $E = \hat{x} + \acute{s} + F$. Comme nous avons également $E = \hat{x} + \hat{F}$, il en découle que $F = \hat{F} - \acute{s} = \hat{F}$.

Ainsi, nous constatons qu'il n'y a qu'un seul sous-espace vectoriel F associé à un sous-espace affine donné.



Définition 1.2.2 (Indépendance linéaire). Les vecteurs $x^1, \dots, x^k \in \mathbb{R}^n$ sont dits linéairement indépendants s'il n'existe aucun ensemble de scalaires $\alpha_1, \dots, \alpha_k$, dont au moins l'un est non nul, tels que $\sum_{i=1}^k \alpha_i x^i = 0$.

Une définition équivalente est que les vecteurs x^i sont tous non nuls, et pour chaque $i = 1, \dots, k$, le vecteur x^i n'appartient pas au sous-espace vectoriel engendré par les vecteurs $x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^k \in \mathbb{R}^n$.

Définition 1.2.3 (Indépendance affine). Les points $x^1, \dots, x^k \in \mathbb{R}^n$ sont dits affinement indépendants si, pour tout choix de réels $\alpha_1, \dots, \alpha_k$ vérifiant $\sum_{i=1}^k \alpha_i = 0$, on a l'implication suivante :

Si $\sum_{i=1}^k \alpha_i x^i = 0$, alors nécessairement $\alpha_1 = \alpha_2 = \dots = \alpha_k = 0$.

En d'autres termes, les points x^1, \dots, x^k sont affinement indépendants si l'unique combinaison affine de ces points qui donne le point zéro est la combinaison triviale où tous les coefficients sont nuls.

Lemme 1.2.4 (Relation entre l'indépendance affine et l'indépendance linéaire). Les points $x^1, \dots, x^k \in \mathbb{R}^n$ sont affinement indépendants si et seulement si les vecteurs $x^2 - x^1, \dots, x^k - x^1$ sont linéairement indépendants.

En d'autres termes, l'indépendance affine des points est équivalente à l'indépendance linéaire des vecteurs qui relient ces points par rapport à un point de référence commun (ici, le point x^1).

Corolaire 1.2.5 (Dimension d'un sous-espace affine). Un sous-espace affine F est de dimension k si et seulement si le nombre maximal de points affinement indépendants qu'il contient est égal à $k + 1$.

En d'autres termes, la dimension d'un sous-espace affine est donnée par le nombre de points affinement indépendants nécessaires pour le définir, plus un.

Un sous-espace vectoriel F de \mathbb{R}^n , contenant au moins un vecteur non nul, possède une base. Une base est une famille de vecteurs qui sont linéairement indépendants et dont l'espace vectoriel qu'ils engendrent est égal à F . Chaque base d'un sous-espace vectoriel donné a le même nombre de vecteurs. Ce nombre est appelé la dimension de F . Par convention, le sous-espace vectoriel $\{0\}$ est de dimension zéro.

La dimension d'un sous-espace affine $x + F$ est égale à la dimension du sous-espace vectoriel correspondant à F .



Il est important de noter que chaque sous-espace vectoriel de dimension non nulle possède une base qui est orthogonale. Cela signifie que pour toute paire de vecteurs distincts de la base, ces vecteurs sont orthogonaux les uns aux autres.



Définition 1.2.6 (Orthogonalité d'un ensemble et Complément orthogonal).

1. Soit E un ensemble quelconque. L'ensemble des vecteurs qui sont orthogonaux à tous les éléments de E est appelé l'orthogonal de E et est noté E^\perp :

$$E^\perp = \{y : \langle y^t, x \rangle = 0, \forall x \in E\}.$$

En d'autres termes, E^\perp est l'ensemble des vecteurs qui sont perpendiculaires à tous les vecteurs de l'ensemble E .

2. E^\perp est un sous-espace vectoriel.

3. Si E est un sous-espace vectoriel, alors E^\perp est appelé le complément orthogonal de E . Tout vecteur x peut être décomposé de manière unique comme la somme d'un vecteur de E et d'un vecteur de E^\perp . De plus, nous avons $(E^\perp)^\perp = E$.

Cela signifie que pour tout vecteur x , il existe un unique vecteur $y \in E$ et un unique vecteur $z \in E^\perp$ tels que $x = y + z$. De plus, l'orthogonal du complément orthogonal de E est égal à E lui-même. En d'autres termes, les éléments de E sont orthogonaux à tous les vecteurs de E^\perp et les éléments de E^\perp sont orthogonaux à tous les vecteurs de E .

Définition 1.2.7 (Combinaison affine). Soient x^1, \dots, x^k un nombre fini de points dans \mathbb{R}^n , et $\alpha_1, \dots, \alpha_k$ des réels tels que

$$\sum_{i=1}^k \alpha_i = 1.$$

On dit que $x := \sum_{i=1}^k \alpha_i x^i$ est une combinaison affine des points x^1, \dots, x^k .



Plus généralement, si $F \in \mathbb{R}^n$ est un sous-ensemble quelconque, on dit que $x \in \mathbb{R}^n$ est une combinaison affine de points de F s'il existe un nombre fini de points de F dont x est une combinaison affine.

Proposition 1.2.8. Un sous-ensemble F de \mathbb{R}^n est un sous-espace affine si et seulement s'il contient toutes les combinaisons affines de points de F .



L'enveloppe affine d'un sous-ensemble F de \mathbb{R}^n est l'ensemble de toutes les combinaisons affines de points de F . On la note $Aff(F)$.



Proposition 1.2.9. L'enveloppe affine d'un sous-ensemble F de \mathbb{R}^n est un sous-espace affine, qui est le plus petit sous-espace affine contenant F .

Théorème 1.2.10 (Carathéodory). Si S est un sous-ensemble de \mathbb{R}^n , tout point de $Aff(S)$ peut s'écrire comme une combinaison affine de points de S affinement indépendants.

Démonstration. Soit x un point de $Aff(S)$, alors il existe x^1, \dots, x^k dans S , et des scalaires $\lambda_1, \dots, \lambda_k$ tels que

$$x = \sum_{i=1}^k \lambda_i x^i, \quad \text{avec} \quad \sum_{i=1}^k \lambda_i = 1. \quad (1.3)$$

Sans perte de généralité, on suppose que tous les λ_i sont non nuls. Si les x^i sont affinement dépendants, alors l'un d'entre eux (en supposant que ce soit x^k) peut s'écrire comme combinaison affine des $k - 1$ autres, donc il existe des scalaires ν_1, \dots, ν_{k-1} tels que

$$x^k = \sum_{i=1}^{k-1} \nu_i x^i, \quad \text{avec} \quad \sum_{i=1}^{k-1} \nu_i = 1. \quad (1.4)$$

En remplaçant x^k dans (1.3), nous obtenons :

$$\begin{aligned} x = \sum_{i=1}^k \lambda_i x^i &= \sum_{i=1}^{k-1} \lambda_i x^i + \lambda_k x^k \\ &= \sum_{i=1}^{k-1} \lambda_i x^i + \lambda_k \sum_{i=1}^{k-1} \nu_i x^i \\ &= \sum_{i=1}^{k-1} \lambda_i x^i + \sum_{i=1}^{k-1} \lambda_k \nu_i x^i \\ &= \sum_{i=1}^{k-1} (\lambda_i + \lambda_k \nu_i) x^i, \end{aligned}$$

où

$$\begin{aligned} \sum_{i=1}^{k-1} (\lambda_i + \lambda_k \nu_i) &= \sum_{i=1}^{k-1} \lambda_i + \sum_{i=1}^{k-1} \lambda_k \nu_i \\ &= \lambda_k \sum_{i=1}^{k-1} \nu_i + \sum_{i=1}^{k-1} \lambda_i \\ &= \lambda_k + \sum_{i=1}^{k-1} \lambda_i \\ &= \sum_{i=1}^k \lambda_i = 1, \end{aligned}$$

Ainsi, x peut s'écrire comme une combinaison affine d'au plus $k - 1$ points de S . Si les points x^1, \dots, x^{k-1} ne sont pas affinement indépendants, on répète la procédure précédente, et ainsi de suite. Après un nombre fini d'étapes (au plus $k - 1$ puisqu'une famille contenant un seul point est affinement indépendante), nous obtenons \hat{k} points $x^1, \dots, x^{\hat{k}}$ avec $1 \leq \hat{k} \leq k$, et des scalaires $\alpha_1, \dots, \alpha_{\hat{k}}$ tels que

$$x = \sum_{i=1}^{\hat{k}} \alpha_i x^i, \quad \text{avec} \quad \sum_{i=1}^{\hat{k}} \alpha_i = 1.$$

Cela conclut la démonstration. ■

La dimension d'un sous-ensemble non vide F de \mathbb{R}^n , notée $\dim(F)$, est équivalente à la dimension de son enveloppe affine. Une autre façon de définir un espace affine est de prendre un espace vectoriel et de supprimer la notion de direction. En d'autres termes, pour travailler dans un espace affine, on peut choisir n'importe quel point pour jouer le rôle de l'origine, tandis que dans un espace vectoriel, l'origine est toujours le point $(0, 0)$.

Définition 1.2.11 (Application affine). On appelle application affine entre les espaces affines \mathbb{R}^n et \mathbb{R}^m toute application de la forme :

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ x &\longmapsto Ax + b, \end{aligned} \quad (1.5)$$

où A est une matrice de taille $m \times n$ et b est un vecteur de \mathbb{R}^m .



Supposons que \mathbb{R}^n et \mathbb{R}^m sont des espaces vectoriels.

◇ Si $b = 0$, alors l'application f est dite linéaire. L'ensemble de toutes les applications linéaires de \mathbb{R}^n dans \mathbb{R}^m est noté $L(\mathbb{R}^n, \mathbb{R}^m)$.

◇ Si $m = 1$ et $b = 0$, alors l'application f est dite forme linéaire. L'ensemble de toutes les formes linéaires définies sur \mathbb{R}^n est noté $\mathcal{L}(\mathbb{R}^n)$.

Définition 1.2.12. Soit $\kappa \in \mathbb{R}^n$ et r un scalaire quelconque.

◇ **Hyperplan vectoriel** : Un hyperplan vectoriel de \mathbb{R}^n est un sous-espace vectoriel de dimension $n - 1$ de la forme

$$\mathcal{H}_\kappa = \{x \in \mathbb{R}^n : \langle \kappa^t, x \rangle = 0\}.$$

◇ **Hyperplan affine** : Un hyperplan affine de \mathbb{R}^n est un sous-espace affine de dimension $n - 1$ de la forme

$$\mathcal{H}_{\kappa r} = \{x \in \mathbb{R}^n : \kappa_1 x_1 + \dots + \kappa_n x_n = r\}.$$

Cela signifie qu'un hyperplan affine n'est rien d'autre qu'un hyperplan vectoriel translaté.

Exemple 1.2.13. Dans \mathbb{R}^2 , les hyperplans vectoriels sont les droites passant par le point $(0, 0)$, c'est-à-dire les droites d'équation $y = \alpha x$ avec $\alpha \geq 0$, tandis que les hyperplans affines sont les droites d'équation $y = \alpha x + \beta$ qui ne passent pas par le point $(0, 0)$, où



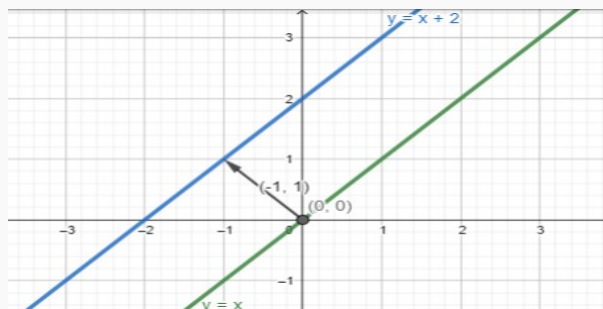


Figure 1.1 – Hyperplans affine dans \mathbb{R}^2

$\alpha \geq 0$ et $\beta > 0$. Par exemple, pour $\alpha = 1$ et $\beta = 2$, l'hyperplan affine d'équation $y = x + 2$ est la translation de l'hyperplan vectoriel d'équation $y = x$ dans la direction du vecteur $(-1, 1)$, comme illustré dans la figure (1.1).

1.2.2 Compréhension des Espaces Vectoriels Normés : Définitions et Illustrations

Dans le contexte de l'analyse fonctionnelle, nous introduisons une notion fondamentale : la norme. La norme est une fonction qui assigne à chaque élément d'un espace vectoriel une mesure de sa taille ou de son amplitude. Nous allons donner quelques exemples pour illustrer cette notion.

1.2.2.1 Normes dans \mathbb{R}^n

Une norme, notée $\|\cdot\|$, dans \mathbb{R}^n est une fonction qui associe à chaque élément x de \mathbb{R}^n un scalaire $\|x\|$, et qui satisfait les propriétés suivantes :

1. Condition d'homogénéité : $\|\alpha x\| = |\alpha| \cdot \|x\|$ pour tout $\alpha \in \mathbb{R}$ et tout $x \in \mathbb{R}^n$.
2. Condition de séparation : $\|x\| = 0$ si et seulement si $x = 0$, pour tout $x \in \mathbb{R}^n$.
3. Inégalité triangulaire : $\|x + y\| \leq \|x\| + \|y\|$ pour tout $x, y \in \mathbb{R}^n$.



Une remarque importante découlant de la notion de norme est la suivante : "Si $\|\cdot\|$ est une norme, alors pour tout élément x de l'espace vectoriel \mathbb{R}^n , $\|x\| \geq 0$ ". Cette propriété met en évidence le fait que la norme attribue une valeur non négative à chaque élément de l'espace vectoriel \mathbb{R}^n , reflétant ainsi la notion de taille ou d'amplitude du vecteur.

1.2.2.2 Exemples de Normes sur \mathbb{R}^n

Voici deux exemples courants de normes sur \mathbb{R}^n qui proviennent d'un produit scalaire :

◇ Dans \mathbb{R}^2 , la norme euclidienne est définie par :

$$\begin{aligned} \|\cdot\| : \mathbb{R}^2 &\longmapsto \mathbb{R}_+ \\ (x, y) &\longmapsto (x^2 + y^2)^{\frac{1}{2}}. \end{aligned}$$

◇ Dans \mathbb{R}^n , une autre norme fréquemment utilisée est définie comme suit :

$$\|x\| = \sqrt{\langle x^t, x \rangle} = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}. \quad (1.6)$$

où $x = (x_1, \dots, x_n)$ est un vecteur de \mathbb{R}^n .

Ces normes sont des mesures de la taille ou de l'amplitude des vecteurs dans \mathbb{R}^n et sont couramment utilisées dans diverses applications en mathématiques et en sciences.

1.2.2.3 Espaces Vectoriels Normés et Norme Euclidienne

Définition 1.2.14. L'espace vectoriel \mathbb{R}^n , muni de la norme $\|\cdot\|$, est appelé un espace vectoriel normé et est noté $(\mathbb{R}^n, \|\cdot\|)$.

Étant donné que nous ferons usage de la norme euclidienne dans une partie de cette étude, il est essentiel de rappeler deux résultats importants liés à cette norme.

Proposition 1.2.15 (Théorème de Pythagore). Pour tout couple de vecteurs x et y qui sont orthogonaux, nous avons :

$$\|x + y\| = \|x\| + \|y\|.$$

Proposition 1.2.16 (Inégalité de Cauchy-Schwarz). Pour tout couple de vecteurs x et y , nous avons :

$$|\langle x^t, y \rangle| \leq \|x\| \cdot \|y\|,$$

avec égalité si et seulement si x et y sont linéairement dépendants.

1.2.3 Exploration des Espaces Métriques : Définitions et Exemples

Cette section vise à introduire les notions fondamentales liées aux espaces métriques et à explorer divers exemples concrets. Un espace métrique est un espace mathématique dans lequel



on peut mesurer les distances entre les points. Cette mesure de distance est donnée par une fonction appelée métrique. Dans cette section, nous commencerons par définir formellement ce qu'est un espace métrique et examinerons les propriétés qui en découlent.

Ensuite, nous passerons en revue quelques exemples d'espaces métriques couramment rencontrés. Nous explorerons l'espace métrique euclidien, qui est l'espace de dimension finie dans lequel nous vivons, ainsi que d'autres exemples tels que l'espace métrique des nombres réels avec la distance standard, l'espace métrique discret, et l'espace métrique de fonctions.

1.2.3.1 Définition d'un Espace Métrique

Un espace métrique est un ensemble X muni d'une fonction appelée métrique $d : X \times X \rightarrow \mathbb{R}$ qui associe à chaque paire de points $x, y \in X$ une valeur réelle non négative $d(x, y)$, appelée distance entre x et y . Cette fonction doit satisfaire les propriétés suivantes :

1. Positivité : $d(x, y) \geq 0$ pour tout $x, y \in X$, et $d(x, y) = 0$ si et seulement si $x = y$.
2. Symétrie : $d(x, y) = d(y, x)$ pour tout $x, y \in X$.
3. Inégalité triangulaire : $d(x, y) + d(y, z) \geq d(x, z)$ pour tout $x, y, z \in X$.

Exemple 1.2.17. Dans \mathbb{R} , on définit la distance entre deux éléments α et β de \mathbb{R} par $d(\alpha, \beta) = |\alpha - \beta|$. Dans \mathbb{R}^n , on a deux types de distances :

◇ **Distance discrète** : On définit l'application de distance discrète sur \mathbb{R}^n comme suit :

$$d(x, y) = \begin{cases} 1 & \text{si } x \neq y, \\ 0 & \text{si } x = y. \end{cases}$$

◇ **Distance basée sur une norme** : Si \mathbb{R}^n est muni d'une norme $\|\cdot\|$, alors la distance entre deux éléments x et y de \mathbb{R}^n est définie par :

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

1.2.3.2 Exemples d'Espaces Métriques

Dans la suite de cette section, nous allons explorer quelques exemples d'espaces métriques couramment rencontrés :

- ◇ L'espace métrique euclidien \mathbb{R}^n , où la métrique est donnée par la distance euclidienne entre les points.
- ◇ L'espace métrique discret, où la distance entre deux points est 1 s'ils sont distincts et 0 s'ils sont égaux.
- ◇ L'espace métrique de fonctions, où la distance entre deux fonctions est définie en utilisant des intégrales ou des normes appropriées.



1.2.4 Fondements des Espaces topologiques : Définitions et Exemples

L'essence de cette section est de fournir une introduction succincte à la notion de topologie appliquée à \mathbb{R}^n . Nous amorcerons en établissant une topologie sur \mathbb{R}^n , formée par un ensemble d'ensembles respectant des caractéristiques spécifiques. L'incorporation d'une mesure de distance ou d'une norme contribuera à simplifier la compréhension ainsi que la manipulation des composants au sein de cette topologie.

Définition 1.2.18 (Topologie sur \mathbb{R}^n). Une topologie sur \mathbb{R}^n est la donnée d'un sous-ensemble $\tau_{\mathbb{R}^n}$ de $\mathcal{P}(\mathbb{R}^n)$, dont les éléments sont appelés les ouverts de la topologie, vérifiant les propriétés suivantes :

- ◇ L'ensemble vide \emptyset et \mathbb{R}^n sont des ouverts, c'est-à-dire qu'ils appartiennent à $\tau_{\mathbb{R}^n}$.
- ◇ La réunion quelconque d'ouverts est un ouvert.
- ◇ L'intersection finie d'ouverts est un ouvert.



Dans la suite de cette étude, nous focalisons notre attention sur l'espace vectoriel \mathbb{R}^n ainsi que sur ses sous-ensembles.



- ◇ Les fermés de la topologie $\tau_{\mathbb{R}^n}$ sont définis comme étant les complémentaires des ouverts de la topologie $\tau_{\mathbb{R}^n}$.
- ◇ Le couple $(\mathbb{R}^n, \tau_{\mathbb{R}^n})$ est appelé espace topologique.
- ◇ Si \mathbb{R}^n est muni d'une structure d'espace vectoriel en plus de la topologie, on dit que \mathbb{R}^n est un espace vectoriel topologique.

Définition 1.2.19 (Intérieur et adhérence). Soit E un sous-ensemble quelconque de \mathbb{R}^n .

- ◇ L'intérieur de E , noté $\overset{\circ}{E}$, est la réunion de tous les ouverts contenus dans E . C'est le plus grand ouvert contenu dans E . Si $x \in \overset{\circ}{E}$, on dit que x est un point intérieur à E .
- ◇ L'adhérence (ou fermeture) de E , notée \bar{E} , est l'intersection de tous les fermés contenant E . C'est le plus petit fermé contenant E .



Proposition 1.2.20 (Caractérisation des ouverts et des fermés). Un sous-ensemble quelconque de \mathbb{R}^n est :

- ◇ un ouvert si et seulement s'il est égal à son intérieur.
- ◇ un fermé si et seulement s'il est égal à son adhérence.

Exemple 1.2.21. La topologie usuelle $\tau_{\mathbb{R}}$ de \mathbb{R} est définie comme suit :

◇ Les ouverts de $\tau_{\mathbb{R}}$ sont les intervalles de la forme :

$$\mathbf{1)}]\alpha, \beta[, \quad \alpha \leq \beta, \quad \mathbf{2)}]\lambda, +\infty[, \quad \mathbf{3)}]-\infty, \lambda[.$$

◇ Les fermés de $\tau_{\mathbb{R}}$ sont les intervalles de la forme :

$$\mathbf{1)} [\alpha, \beta], \quad \alpha \leq \beta, \quad \mathbf{2)} [\lambda, +\infty[, \quad \mathbf{3)}]-\infty, \lambda],$$

où α, β, λ sont des nombres réels.



Si \mathbb{R}^n est muni de la distance (1.6), on peut définir la boule ouverte centrée en x et de rayon r par

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

et la boule fermée de centre x et de rayon r par

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

Exemple 1.2.22.

1. Dans \mathbb{R}^2 muni de la norme euclidienne, la boule ouverte de centre $(0, 0)$ et de rayon 1 est le disque unité^{iv} privé de sa frontière.

2. Dans \mathbb{R}^2 muni de la norme infinie, qui associe à tout $(\alpha, \beta) \in \mathbb{R}^2$ la valeur $\|(\alpha, \beta)\| = \max\{|\alpha|, |\beta|\}$, la boule ouverte unité est l'intérieur du carré de bord délimité par les points de coordonnées $(0, 1)$, $(0, -1)$, $(1, 0)$, $(-1, 0)$.

3. Dans \mathbb{R}^2 muni de la norme définie par $\|x\| = (x_1^k + x_2^k)^{\frac{1}{k}}$, notée également $\|\cdot\|_k$, pour $k \geq 1$ ou $k = +\infty$, la boule ouverte unité de centre 0 et de rayon 1 est l'intérieur de la courbe orange fermée tracée dans la figure (1.2).

Plus généralement, cet exemple peut être décrit de manière analogue dans l'espace \mathbb{R}^n en utilisant la norme $\|x\|_k = \left(\sum_{i=1}^n |x_i|^k\right)^{\frac{1}{k}}$.

iv. Disque de centre $(0, 0)$ et de rayon 1.



4. Ces exemples illustrent différentes formes de boules ouvertes en fonction de la norme utilisée, montrant ainsi la diversité des structures topologiques possibles sur \mathbb{R}^n en fonction du choix de la norme.

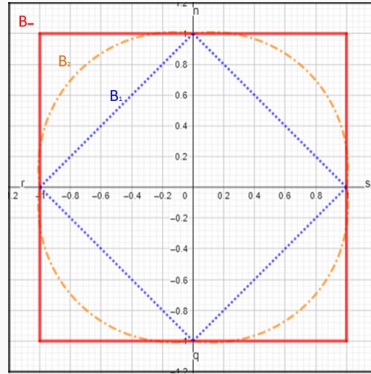


Figure 1.2 – Représentation des boules unités associées aux normes $\|\cdot\|_k$ dans \mathbb{R}^2

Proposition 1.2.23.

- ◇ Toute boule fermée est un fermé.
- ◇ Toute boule ouverte est un ouvert.
- ◇ Une partie de \mathbb{R}^n est bornée si et seulement si elle est contenue dans une boule.
- ◇ Un ouvert de \mathbb{R}^n est une union quelconque de boules ouvertes.

Proposition 1.2.24. Une norme sur \mathbb{R}^n induit une structure topologique.

Définition 1.2.25. Soit E une partie non vide de \mathbb{R}^n . On appelle topologie induite sur E (ou la topologie relative sur E) l'ensemble :

$$\tau_E := \{U \cap E : U \in \tau_{\mathbb{R}^n}\}.$$

Exemple 1.2.26. Quelques exemples dans $E = \mathbb{R}$:

- ◇ Considérons la topologie induite sur $S =]0, 2[\subset \mathbb{R}$ définie par :

$$\tau_S = \tau_S^1 \cup \tau_S^2 \cup \tau_S^3,$$

où

$$\begin{aligned} \tau_S^1 &= \{] \alpha, \beta[\cap]0, 2[: \alpha, \beta \in \mathbb{R}\}, \\ \tau_S^2 &= \{] \lambda, +\infty[\cap]0, 2[: \lambda \in \mathbb{R}\}, \\ \tau_S^3 &= \{]-\infty, \lambda[\cap]0, 2[: \lambda \in \mathbb{R}\}. \end{aligned}$$



Alors, par exemple, l'intervalle $[1, \frac{1}{2}[$ est un ouvert de S (même si ce n'est pas un ouvert de \mathbb{R}). En effet,

$$[0, 0.5[=]-1, 0.5[\cap [0, 2[.$$

D'autre part, $[1, 2[$ est fermé dans S (même si ce n'est pas un fermé de \mathbb{R}). En effet,

$$[1, 2[= [1, 4] \cap [0, 2[.$$

◊ Un autre exemple plus subtil est la topologie induite sur \mathbb{Z} par la topologie de \mathbb{R} . Pour tout $n \in \mathbb{Z}$, le singleton n est un ouvert dans la topologie de \mathbb{Z} . En effet,

$$\{n\} =]n - 1, n + 1[\cap \mathbb{Z}.$$

D'autre part, n est fermé dans \mathbb{Z} . En effet,

$$\{n\} = [n - 0.5, n + 0.5] \cap \mathbb{Z}.$$

Plus généralement, on peut montrer de la même façon que tous les sous-ensembles de \mathbb{Z} sont ouverts et fermés à la fois. En effet, puisqu'un singleton est ouvert et fermé, alors tous les sous-ensembles de \mathbb{Z} sont une réunion de singletons (ouverts et fermés de \mathbb{Z}).

1.3 Fondements de l'Analyse Convexe : Structures et Propriétés

Dans les problèmes d'optimisation, qu'ils soient avec ou sans contrainte, la notion de convexité joue un rôle central et essentiel [Roc70; Sun06; Rad98; Fli02; Nay20; Ye 89; Nem91]. En effet, la plupart des algorithmes d'optimisation garantissent la convergence vers un optimum global uniquement sous des hypothèses de convexité. Dans cette étude, nous considérons initialement \mathbb{R}^n comme un \mathbb{R} -espace vectoriel de dimension finie, sans structure additionnelle.

1.3.1 Propriétés des Ensembles Convexes

Pour introduire la notion d'ensemble convexe, il est nécessaire de considérer des combinaisons entre paires de points dans l'espace ambiant.

Définition 1.3.1 (*Segment*). Le segment d'extrémités x et y , noté $[x, y]$, est défini comme l'ensemble

$$[x, y] = \{\alpha x + (1 - \alpha)y : 0 \leq \alpha \leq 1\}.$$

Plus généralement, pour un nombre fini de points $x^1, \dots, x^k \in S$ avec $k \geq 2$, une combinaison convexe des points $x^1, \dots, x^k \in S$ est un point de la forme :

$$x = \sum_{i=1}^k \alpha_i x^i,$$

où

$$\sum_{i=1}^k \alpha_i = 1.$$

Définition 1.3.2 (*Ensemble convexe*).

1. Un ensemble $S \subset \mathbb{R}^n$ est dit convexe si, pour tout couple (x, y) de points de S , le segment $[x, y]$ est inclus dans S , c'est-à-dire,

$$\forall x, y \in S, \forall \alpha \in [0, 1], \text{ on a } \alpha x + (1 - \alpha)y \in S. \quad (1.7)$$

2. Plus généralement, un sous-ensemble $S \subset \mathbb{R}^n$ est dit convexe s'il contient toutes les combinaisons convexes d'éléments de S .

Définition 1.3.3.

1. **Combinaison conique** : Soit x^1, \dots, x^k un nombre fini de points de \mathbb{R}^n , et $\lambda_1, \dots, \lambda_k$ des réels positifs. On dit alors que

$$x := \sum_{i=1}^k \lambda_i x^i$$

est une combinaison conique des points x^1, \dots, x^k . Plus généralement, si $S \subset \mathbb{R}^n$ est un sous-ensemble quelconque, on dit que $x \in \mathbb{R}^n$ est une combinaison conique de points de S s'il existe un nombre fini de points de S dont x soit une combinaison conique.

2. **Enveloppe conique** : L'enveloppe conique d'un sous-ensemble S de \mathbb{R}^n est l'ensemble de toutes les combinaisons coniques de points de S . On la note $\text{cône}(S)$.

3. Un ensemble convexe S de points de \mathbb{R}^n est un cône^v si $x \in S$ implique $(\lambda x) \in S$ pour tout $\lambda \geq 0$. Un cône est dit pointé s'il possède un point extrême^{vi}.

v. À titre d'exemple dans \mathbb{R}^3 , veuillez vous référer à la figure 1.5.

vi. Voir la définition (1.4.11)



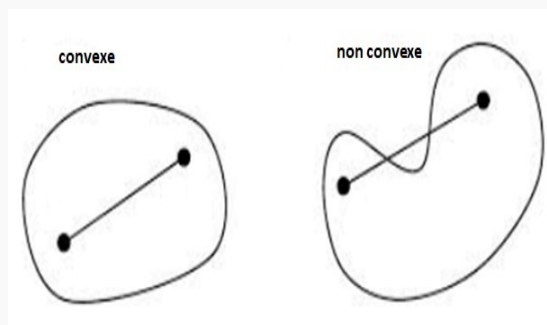


Figure 1.3 – Ensembles convexes et non-convexes

Proposition 1.3.4.

1. Un sous-ensemble S de \mathbb{R}^n est un cône convexe si et seulement s'il contient toutes les combinaisons coniques de points de S .
2. L'enveloppe conique d'un sous-ensemble S de \mathbb{R}^n est un cône convexe. C'est le plus petit cône convexe contenant S .

Théorème 1.3.5 (Théorème de Carathéodory pour les cônes). Soit S un sous-ensemble de \mathbb{R}^n . Tout point dans $\text{cône}(S)$ peut être représenté comme une combinaison conique de points de S qui sont linéairement indépendants.

Démonstration. La preuve de ce théorème est similaire à celle présentée dans la preuve (1.2.10). ■

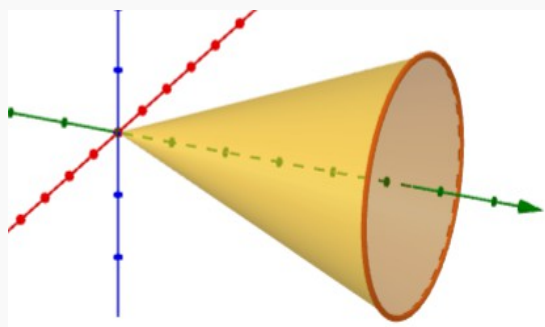


Figure 1.4 – Cône dans \mathbb{R}^3

Exemple 1.3.6. Voici quelques exemples d'ensembles convexes :

1. L'ensemble \mathbb{R}^n lui-même est convexe.
2. Tout espace affine et tout espace vectoriel sont convexes.
3. Soit \mathbb{R}^n muni de la norme euclidienne. Alors, pour tout $x \in \mathbb{R}^n$ et tout $r \geq 0$, la boule centrée en x et de rayon r (ouverte ou fermée) est convexe.

4. Soit d une direction non nulle et x_0 un point fixé dans \mathbb{R}^n . L'ensemble

$$S = \{x \in \mathbb{R}^n : x = x_0 + \alpha d, \alpha \geq 0\}$$

est convexe. En effet, pour tout $x^1, x^2 \in S$ et tout $\lambda \in [0, 1]$, nous avons

$$\begin{aligned} \lambda x^1 + (1 - \lambda)x^2 &= \lambda(x_0 + \alpha_1 d) + (1 - \lambda)(x_0 + \alpha_2 d) \\ &= x_0 + [\lambda\alpha_1 + (1 - \lambda)\alpha_2]d. \end{aligned}$$

Comme $\lambda\alpha_1 + (1 - \lambda)\alpha_2 \geq 0$, nous avons $\lambda x^1 + (1 - \lambda)x^2 \in S$.

Théorème 1.3.7. Soit $S \subset \mathbb{R}^n$ un ensemble convexe. Alors :

1. L'intérieur de S , noté $\overset{\circ}{S}$, est un ensemble convexe.
2. La fermeture de S , notée \overline{S} , est un ensemble convexe.

Démonstration.

1. Pour montrer que $\overset{\circ}{S}$ est convexe, nous devons vérifier que pour tout couple de points $x, y \in \overset{\circ}{S}$ et pour tout $\lambda \in [0, 1]$, le point $\lambda x + (1 - \lambda)y$ appartient à $\overset{\circ}{S}$. Comme x et y appartiennent à $\overset{\circ}{S}$, il existe des boules ouvertes $B(x, r_1)$ et $B(y, r_2)$ incluses dans S . Soit $r = \min(r_1, r_2)$. Alors, pour tout $\lambda \in [0, 1]$, le point $\lambda x + (1 - \lambda)y$ appartient à $B(x, r)$ et donc à S . Ainsi, $\overset{\circ}{S}$ est convexe.

2. Pour montrer que \overline{S} est convexe, nous devons vérifier que pour tout couple de points $x, y \in \overline{S}$ et pour tout $\lambda \in [0, 1]$, le point $\lambda x + (1 - \lambda)y$ appartient à \overline{S} . Comme x et y appartiennent à \overline{S} , cela signifie que pour toute boule ouverte $B(x, r_1)$ et $B(y, r_2)$, les boules intersectent S . Soit $r = \frac{1}{2}(\|x\| + \|y\|)$. Alors, pour tout $\lambda \in [0, 1]$, la boule ouverte $B(\lambda x + (1 - \lambda)y, r)$ intersecte S car elle contient des points de $B(x, r_1)$ et $B(y, r_2)$. Par conséquent, $\lambda x + (1 - \lambda)y$ appartient à \overline{S} . Ainsi, \overline{S} est convexe.

■

Théorème 1.3.8. Soit $(S_i)_{i \in I} \subset \mathbb{R}^n$ une famille quelconque d'ensembles convexes.

Alors :

- ◇ L'intersection quelconque $\bigcap_{i \in I} S_i$ est convexe.
- ◇ Le produit cartésien $S_i \times S_j$ de deux ensembles convexes est convexe.
- ◇ Pour $\alpha_1, \dots, \alpha_k$ des nombres réels, $\alpha_1 S_1, \dots, \alpha_k S_k$ est convexe.

De plus, l'image directe (resp. réciproque) d'un convexe par une application linéaire est convexe.





En analyse convexe, il est courant de rencontrer des ensembles convexes dont l'intérieur (au sens topologique) est vide. Cela se produit, par exemple, avec un segment dans \mathbb{R}^2 ou avec un carré dans \mathbb{R}^3 . Pour remédier à ce problème, on a introduit la notion d'intérieur relatif, qui permet de donner un sens à l'idée intuitive que l'intérieur d'un segment $[x, y] \in \mathbb{R}^2$ est l'intervalle ouvert $]x, y[$ et que l'intérieur d'un carré $[x, y] \times [x, y] \in \mathbb{R}^3$ est la boule ouverte $]x, y[\times]x, y[$ définie par la norme infinie (voir exemple 1.2.22).

Définition 1.3.9. Soit S une partie de l'espace vectoriel \mathbb{R}^n . L'intérieur relatif de S , noté $ri(S)$, est l'intérieur de S considéré comme sous-ensemble de l'espace métrique $(Aff(S), d)$, où d est une distance définie par une norme, voir (1.6). Autrement dit, $ri(S)$ est l'intérieur de S au sens de la définition (1.2.19) relativement au sous-espace affine engendré par S .



Nous présentons maintenant un lemme fondamental, crucial pour établir la convexité de l'ensemble des solutions dominées^{vii} dans un problème de programmation linéaire multi-objectifs^{viii}.

Lemme 1.3.10. Soit $S \subset \mathbb{R}^n$ un sous-ensemble convexe avec un intérieur non vide. Alors, pour tout $x \in \overset{\circ}{S}$ et $y \in \overline{S}$, on a

$$]x, y[= \{\alpha x + (1 - \alpha)y : 0 < \alpha \leq 1\} \subseteq \overset{\circ}{S}.$$

1.3.2 Propriétés des Fonctions Convexes

Définition 1.3.11. Soit S un sous-ensemble convexe de \mathbb{R}^n et f une fonction définie sur S à valeurs dans \mathbb{R} .

1. On dit que f est convexe si pour tout $x \in S, y \in S$ et $\lambda \in [0, 1]$, on a

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (1.8)$$

2. On dit que f est strictement convexe si l'inégalité (1.8) est stricte pour $x \neq y$ et $\lambda \in]0, 1[$.

3. On dit que f est concave si $(-f)$ est convexe.

viii. Voir la définition 4.1.11

viii. Voir le chapitre 4





Si x^1, \dots, x^k est un nombre fini de points de S et $f : S \mapsto \mathbb{R}$ est une fonction convexe, alors on peut montrer par récurrence que pour tout nombre fini de scalaires $\lambda_1, \dots, \lambda_k$, on a

$$f\left(\sum_{i=1}^k \lambda_i x^i\right) \leq \sum_{i=1}^k \lambda_i f(x^i).$$

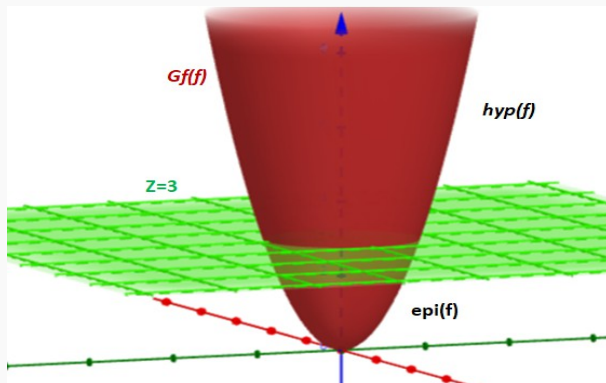


Figure 1.5 – Illustration du graphe, de l’hypographe et de l’épigraphe de la fonction $f(x, y) = x^2 + y^2$ dans \mathbb{R}^3 pour $\alpha = 3$.

Exemple 1.3.12.

1. Un exemple basique de fonction convexe est la fonction indicatrice. Soit $S \subset \mathbb{R}^n$ un sous-ensemble non vide. La fonction indicatrice $I_S : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ est définie par

$$I_S(x) = \begin{cases} 0, & \text{si } x \in S, \\ +\infty, & \text{ailleurs.} \end{cases}$$

La fonction indicatrice I_S est convexe si et seulement si S est convexe.

2. On définit la distance $d_S(x) = \inf\{\|y - x\| : y \in S\}$, où $S \subset \mathbb{R}^n$ est un ensemble convexe non vide et $\|\cdot\|$ est une norme quelconque dans \mathbb{R}^n . Alors d_S est une fonction convexe.

3. Soient $\alpha_1, \alpha_2, \alpha_3$ des nombres dans \mathbb{R} tels que $\alpha_1 < \alpha_3 < \alpha_2$ et on définit les applications

$$\begin{aligned} L : [\alpha_1, \alpha_2] &\longrightarrow [\alpha_1, \alpha_3] \\ x &\longrightarrow \frac{(\alpha_3 - \alpha_1)x + \alpha_1(\alpha_2 - \alpha_3)}{\alpha_2 - \alpha_1}, \end{aligned} \tag{1.9}$$

$$\begin{aligned} U : [\alpha_1, \alpha_2] &\longrightarrow [\alpha_3, \alpha_2] \\ x &\longrightarrow \frac{(\alpha_2 - \alpha_3)x + \alpha_2(\alpha_3 - \alpha_1)}{\alpha_2 - \alpha_1}. \end{aligned} \tag{1.10}$$



Alors L et U sont des bijections bicontinues et convexes qui réduisent l'intervalle $[\alpha_1, \alpha_2]$ sur $[\alpha_1, \alpha_3]$ (resp. $[\alpha_3, \alpha_2]$).

Définition 1.3.13. Soit $S \subset \mathbb{R}^n$ un ensemble non vide et $f : S \rightarrow \mathbb{R}$.

1. Le graphe de la fonction f , noté $Gf(f)$, est le sous-ensemble de \mathbb{R}^{n+1} , défini par :

$$Gf(f) = \{(x, f(x)) \in \mathbb{R}^{n+1} : x \in S\}.$$

2. L'épigraphe de la fonction f , noté $epi(f)$, est le sous-ensemble de \mathbb{R}^{n+1} défini par :

$$epi(f) = \{(x, \alpha) \in \mathbb{R}^{n+1} : f(x) \leq \alpha, x \in S, \alpha \in \mathbb{R}\}. \quad (1.11)$$

3. L'hypographe de la fonction f , noté $hyp(f)$, est le sous-ensemble de \mathbb{R}^{n+1} défini par :

$$hyp(f) = \{(x, \alpha) \in \mathbb{R}^{n+1} : f(x) \geq \alpha, x \in S, \alpha \in \mathbb{R}\}. \quad (1.12)$$



L'interprétation géométrique de la définition (1.3.11) de la convexité d'une fonction f sur un ensemble convexe S est que pour tout $x, y \in S$, le graphe de la fonction f est toujours situé en dessous du segment reliant les points $(x, f(x))$ et $(y, f(y))$. Cette propriété exprime la convexité de la fonction et illustre la notion de "courbure" de la fonction sur l'ensemble convexe.

Théorème 1.3.14. Soient $S \subset \mathbb{R}^n$ un ensemble convexe non vide et $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$. Alors, f est convexe si et seulement si l'épigraphe de f , c'est-à-dire $epi(f)$, est un ensemble convexe.

Démonstration. Supposons que f est convexe. Soient $x, y \in S$ et $(x, \alpha_1), (y, \alpha_2)$ dans $epi(f)$. D'après les définitions de la fonction convexe et de l'épigraphe, il s'ensuit que

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \leq \lambda \alpha_1 + (1 - \lambda)\alpha_2,$$

pour tout $\lambda \in [0, 1]$. Comme S est un ensemble convexe, on a $\lambda x + (1 - \lambda)y \in S$. Ainsi, $(\lambda x + (1 - \lambda)y, \lambda \alpha_1 + (1 - \lambda)\alpha_2) \in epi(f)$, ce qui implique que $epi(f)$ est convexe.

Supposons maintenant que $epi(f)$ est convexe, et soient $x, y \in S$. On a $(x, f(x)), (y, f(y)) \in epi(f)$. D'après la convexité de $epi(f)$, pour tout $\lambda \in [0, 1]$, on a

$$(\lambda x + (1 - \lambda)y, \lambda f(x) + (1 - \lambda)f(y)) \in epi(f), \text{ pour } \lambda \in [0, 1].$$



Cela implique que pour tout $\lambda \in [0, 1]$, on a

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y),$$

pour tout $\lambda \in [0, 1]$. Par conséquent, f est convexe. ■

Théorème 1.3.15 (Opérations usuelles sur les fonctions convexes). Soit S un sous-ensemble convexe de \mathbb{R}^n . Pour tout $\alpha_1, \alpha_2, \dots, \alpha_k \geq 0$, si f_1, f_2, \dots, f_k sont des fonctions convexes sur S à valeurs dans \mathbb{R} , alors $\sum_{i=1}^k \alpha_i f_i$ est aussi convexe sur S .



Un problème de programmation est convexe s'il vise à minimiser une fonction convexe (ou maximiser une fonction concave) sur un domaine convexe.

1.4 Exploration des Polyèdres et des Polytopes



Josef Stoer

Dans cette section, nous nous aventurons à explorer le contenu du 2^{ème} chapitre du livre de Josef Stoer et Christoph Witzgall [Sto70]. Notre objec-

tif est de présenter une brève étude sur quelques propriétés des polyèdres, des entités géométriques qui émergent naturellement lors de la modélisation des problèmes d'optimisation linéaire.



Christoph Witzgall

Il est à noter que lorsque les contraintes du problème sont exprimées linéairement, elles donnent lieu à ce que l'on désigne ultérieurement comme un

polytope. Une caractéristique intrigante réside dans le fait que la définition de ce polytope peut se révéler indépendante des contraintes qui lui ont donné naissance.

1.4.1 Polyèdres et Leurs Frontières : Une Brève Étude

Pour explorer les propriétés des polyèdres, considérons une matrice A de taille $m \times n$ et un vecteur b dans \mathbb{R}^m , où m et n sont des entiers finis. Nous définissons les systèmes d'équations et d'inéquations suivants :

Système d'équations linéaires

$$Ax = b. \quad (1.13)$$

Système d'inéquations linéaires

$$Ax \leq b. \quad (1.14)$$



Définition 1.4.1. Une inéquation est dite **singulière** si elle peut être satisfaite comme une équation, c'est-à-dire si elle vérifie les deux conditions suivantes :

$$Ax \leq b \quad \text{et} \quad -Ax \leq -b.$$

Une inéquation qui n'est pas singulière est appelée une **inéquation faible**.

Lemme 1.4.2. Soit D une matrice de taille $m \times n$. Alors, tout système du type

$$\begin{cases} Ax & \geq 0 \\ Dx & = 0 \end{cases}$$

admet une solution x_0 qui satisfait toutes les inégalités faibles $A_i^T x \geq 0$ en tant qu'inégalités strictes $A_i^T x > 0$.

Notations 1.4.3.

◇ $F = \{x \in \mathbb{R}^n : Ax = b\}$ est l'ensemble des solutions du système (1.13).

◇ $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ est l'ensemble des solutions du système (1.14).

Propriété 1.4.4. On distingue les trois cas suivants :

1. L'ensemble F peut se réduire à l'ensemble vide. Pour illustrer cela, prenons le cas où $n = m = 2$ et considérons les matrices suivantes :

$$A = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}; \quad b = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

Dans cette configuration, le système d'équations linéaires $Ax = b$ ne possède aucune solution, ce qui entraîne F à être un ensemble vide. Une illustration de ce scénario se produit lorsque les équations sont mutuellement contradictoires, comme démontré dans l'exemple susmentionné.

2. Pour $m = n$:

(a) Si la matrice A est inversible ($\det(A) \neq 0$), le système $Ax = b$ admet une unique solution x_{sol} , donc $F = \{x_{\text{sol}}\}$.

(b) Si la matrice A est singulière ($\det(A) = 0$), le système $Ax = b$ admet une infinité de solutions.

3. Pour $m \leq n$, c'est un cas particulier du cas précédent.





1. Pour tout $i = 1, \dots, m$, on a :

(a) L'ensemble F^i des solutions de l'équation $A_i^t x = b_i$ est l'hyperplan affine défini par

$$F^i := \mathcal{H}_{A_i^t} = \{x \in \mathbb{R}^n : \langle A_i^t, x \rangle = b_i\}.$$

(b) L'ensemble P^i des solutions de l'inéquation $A_i^t x \leq b_i$ est le demi-espace affine défini par $P^i := \mathcal{H}_{A_i^t}^{\leq} = \{x \in \mathbb{R}^n : \langle A_i^t, x \rangle \leq b_i\}$.

2. L'ensemble P des solutions du système d'inéquations (1.14) est obtenu par l'intersection d'un nombre fini de demi-espaces P^i dans \mathbb{R}^n , c'est-à-dire $P = \bigcap_{i=1}^m P^i$.

Définition 1.4.5.

1. L'ensemble P des solutions du système d'inéquations (1.14) est appelé un **polyèdre**.

2. Le système d'inéquations (1.14) est appelé une **représentation du polyèdre** P .

3. Le polyèdre P est dit à pleine dimension si $\dim(P) = n$.

4. Pour tout $i = 1, \dots, m$, l'hyperplan F^i est appelé une **hyperplan frontière** de P . Il est important de noter qu'une frontière d'un polyèdre ne rencontre pas nécessairement le polyèdre lui-même, comme illustré dans la figure (1.8).

5. Un hyperplan frontière contenant tout le polyèdre P est appelé un **hyperplan frontière singulier**.

6. L'intersection d'un hyperplan frontière avec le polyèdre P forme une face du polyèdre.

(a) L'intersection d'un hyperplan frontière avec le polyèdre P contenant exactement un seul point est appelée un **sommet** du polyèdre.

(b) Une face est elle-même un polyèdre.

(c) La fermeture \bar{F} d'une face F est également une face.

(d) Un polyèdre est considéré comme une face de lui-même.

(e) On appelle **face propre** toute face du polyèdre qui est différente du polyèdre lui-même.

Théorème 1.4.6. Un polyèdre est un ensemble convexe.



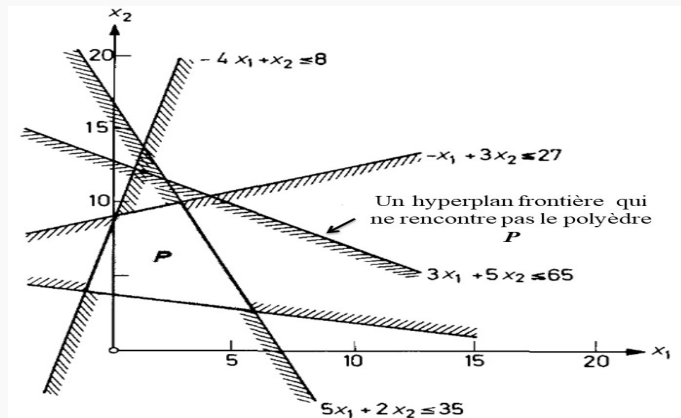


Figure 1.8 – Hyperplan frontière



Il est important de noter que différents systèmes d'inégalités qui décrivent le même polyèdre peuvent conduire à des ensembles différents d'hyperplans frontières. Par conséquent, la notion d'hyperplan frontière est liée aux systèmes d'inégalités plutôt qu'au polyèdre lui-même. La définition des faces est basée sur l'ensemble des hyperplans frontières. Cela signifie que la définition des faces dépend de la représentation du polyèdre par un système d'inégalités (1.13). Cependant, nous verrons plus tard que cette dépendance est en réalité fautive.

Définition 1.4.7 (Points frontières et points intérieurs). Les points frontières et les points intérieurs d'un polyèdre sont déterminés par leur appartenance ou non à une face propre. Soit x un point de P , alors :

1. x est un point frontière de P s'il appartient à une face propre de P . Cela signifie qu'il existe $i = 1, \dots, m$ tel que $x \in P^i \cap P$.
2. x est un point interne de P s'il satisfait chaque inégalité non singulière strictement. Autrement dit, pour chaque $i = 1, \dots, m$ tel que $Ax \leq b$, on a $Ax < b$.
3. x est un point intérieur de P si x est un point intérieur au sens de la définition (1.3.9), relativement à la topologie induite sur $Aff(P)$ par celle de \mathbb{R}^n . En termes topologiques, la notion de point interne et celle de point intérieur coïncident. Cela peut être caractérisé de la manière suivante :
 - (a) Il existe un réel ϵ strictement positif tel que la boule ouverte $B(x, \epsilon)$ soit entièrement contenue dans P .
 - (b) On peut déplacer légèrement x sans quitter le polyèdre P , comme décrit dans le lemme (1.4.9). En d'autres termes, pour toute direction $d \neq 0$ dans \mathbb{R}^n , il existe un réel ϵ strictement positif tel que $x + \epsilon d \in P$.

Lemme 1.4.8. Soit P un polyèdre, x est un point intérieur de P et y est un point arbitraire appartenant à l'intersection de tous les hyperplans frontières singuliers de P .



Alors :

1. Il existe $\epsilon > 0$ tel que $x + \epsilon(x - y) \in P$ et $x - \epsilon(x - y) \in P$.
2. Si de plus $y \notin P$, alors le segment ouvert $]x, y[$ contient un point frontière x_{fr} .

Démonstration.

1. Si pour tout $i = 1, \dots, m$, $A_i^t(x - y) = 0$, alors il n'y a rien à démontrer. Sinon, posons

$$\epsilon^+ := \min \left\{ \frac{b_i - A_i^t x}{|A_i^t(y - x)|} : A_i^t(y - x) > 0 \right\} \quad (1.15)$$

et

$$\epsilon^- := \min \left\{ -\frac{b_i - A_i^t x}{|A_i^t(y - x)|} : A_i^t(y - x) < 0 \right\}.$$

Définissons ϵ_{\max} par $\epsilon_{\max} := \min \{\epsilon^-, \epsilon^+\}$. Il suffit ensuite de prendre $0 < \epsilon \leq \epsilon_{\max}$. En effet, pour tout $i = 1, \dots, m$, nous avons

$$\epsilon \leq \frac{b_i - A_i^t x}{|A_i^t(y - x)|}.$$

En multipliant les deux côtés de l'équation, nous obtenons

$$\epsilon |A_i^t(y - x)| \leq b_i - A_i^t x.$$

En déplaçant le terme $A_i^t x$ de l'autre côté de l'équation, nous avons

$$A_i^t(x + \epsilon(y - x)) \leq b_i.$$

Le même calcul s'applique pour le point $x - \epsilon(y - x)$.

2. Il suffit de prendre $x_{fr} = x + \epsilon(y - x)$, où $\epsilon = \epsilon^+$ selon (1.15).

■

Lemme 1.4.9 (Lemme d'accessibilité pour les polyèdres). Soit P un polyèdre, x un point intérieur et y un point arbitraire de P . Alors $]x, y[\subset \overset{\circ}{P}$.

Définition 1.4.10. Un polyèdre borné, c'est-à-dire un polyèdre pour lequel il existe un nombre B tel que chaque point du polyèdre a des coordonnées comprises entre $-B$ et B , est appelé un polytope.



1.4.2 Sous-ensemble Extrême et Sous-ensemble Exposé

Dans cette section, nous introduisons une nouvelle définition du polyèdre P qui est indépendante de sa représentation par des inégalités et dépend uniquement de P lui-même. Pour ce faire, nous introduisons deux notions clés en analyse convexe : les sous-ensembles extrêmes et les sous-ensembles exposés d'un ensemble convexe. Nous montrerons plus tard que ces deux notions coïncident.

Définition 1.4.11. Soit $K \subset \mathbb{R}^n$ un ensemble convexe et W un sous-ensemble convexe de K . On dit que W est un sous-ensemble extrême de K si aucun de ses points n'est inclus dans un segment ouvert reliant deux points de K qui ne sont pas tous les deux dans W . Autrement dit, W est un sous-ensemble extrême de K si $W = \emptyset$ ou si W est un sous-ensemble convexe de K vérifiant la condition suivante :

$$\begin{array}{l} 1. x, y \in K. \\ 2. W \cap]x, y[\neq \emptyset. \end{array} \implies x, y \in W.$$

Un sous-ensemble extrême de K contenant un seul point est appelé un point extrême de K , comme illustré dans la figure (1.9).

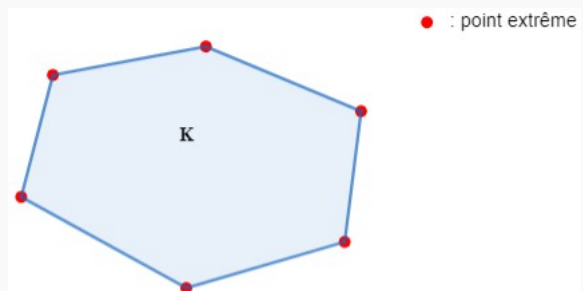


Figure 1.9 – Points extrêmes d'un polytope dans \mathbb{R}^2

Lemme 1.4.12. Soit K un ensemble convexe de \mathbb{R}^n et W un sous-ensemble convexe de K . Alors, $W = \text{Aff}(W) \cap K$.



Maintenant, considérons un polyèdre P représenté par le système (1.13). Selon le lemme (1.4.12), nous pouvons constater que tout sous-ensemble extrême de P est une face de P . Le lemme suivant établit la réciproque.

Lemme 1.4.13. Toute face F de P est un sous-ensemble extrême de P .

Démonstration. Soit F une face de P . Par définition de F (voir 1.4.5), il existe k hyperplans frontières F^1, \dots, F^k tels que $F = P \cap F^i$ pour tout $i = 1, \dots, k$.

Considérons $x, y \in P$ tels que $F \cap]x, y[\neq \emptyset$. Cela signifie qu'il existe des coefficients $\lambda, \nu > 0$ tels que $\lambda + \nu = 1$, permettant de définir un point $\lambda x + \nu y \in F$. Comme F est l'intersection de P avec chaque F^i , le point $\lambda x + \nu y$ appartient à $P \cap F^i$ pour tout i . Par conséquent, $\lambda x + \nu y$ appartient à la fois à P et à chaque hyperplan frontière F^i . Cela se traduit par l'équation linéaire $A_i^t(\lambda x + \nu y) = b_i$ pour chaque i , où A_i^t représente la i -ème ligne de A . Supposons maintenant, par contradiction, qu'il existe un i tel que ($A_i^t x < b_i$ et/ou $A_i^t y < b_i$). Cela conduirait à :

$$\lambda A_i^t x < \lambda b_i \text{ et/ou } \nu A_i^t y < \nu b_i \Rightarrow \lambda A_i^t x + \nu A_i^t y < (\lambda + \nu) b_i < b_i,$$

ce qui contredirait l'équation précédente $A_i^t(\lambda x + \nu y) = b_i$. Par conséquent, nous en concluons que $A_i^t x = b_i$ et $A_i^t y = b_i$ pour tous les i , ce qui signifie que x et y appartiennent à F .

Ainsi, nous montrons que x et y appartiennent à la face F , ce qui démontre que F est un sous-ensemble extrême de P . ■

Lemme 1.4.14. Si W est un sous-ensemble extrême du polyèdre P , alors le sous-espace affine $Aff(W)$ engendré par W est l'intersection de tous les hyperplans frontières F^i contenant W , c'est-à-dire

$$Aff(W) = \bigcap \{ F^i : W \subseteq F^i \}.$$



Une application du lemme (1.4.14) à l'ensemble $W = P$ permet de prouver le résultat suivant.

Théorème 1.4.15. Soit P un polyèdre dans \mathbb{R}^n . Les énoncés suivants sont vrais :

1. Les sous-ensembles extrêmes de P sont ses faces.
2. Le sous-espace affine engendré par P est l'intersection de ses hyperplans frontières singulières.
3. Le nombre de faces d'un polyèdre est fini et indépendant de la représentation du polyèdre par les inégalités.



Ce théorème affirme deux points très importants. Le premier montre que les faces d'un polyèdre P sont indépendantes de sa représentation, c'est-à-dire qu'une face de P est déterminée de manière unique comme l'intersection de tous les hyperplans frontières de P dans lesquels elle est contenue. Le deuxième point est crucial pour déterminer la dimension de P .



Définition 1.4.16. Soit P un polyèdre dans \mathbb{R}^n . Une face F de P est appelée une *facette* de P si $\dim(F) = \dim(P) - 1$.

Théorème 1.4.17. Un point x est un point extrême d'un polyèdre P si et seulement si $\{x\}$ est une face de P de dimension 0.

Définition 1.4.18 (Sous-ensemble exposé). Soient P un polyèdre dans \mathbb{R}^n et W un sous-ensemble de P . On dit que W est un sous-ensemble exposé de P s'il existe une forme linéaire f telle que

$$W = \{z \in P : f(z) \geq f(x), \text{ pour tous } x \in P\}.$$

En d'autres termes, W est un sous-ensemble exposé de P si et seulement si il existe une forme linéaire f telle que l'ensemble des points $x_0 \in P$ satisfaisant

$$f(x_0) = \max_{x \in P} f(x).$$

C'est-à-dire

$$f(x_0) \geq f(x) \text{ pour tout } x \in P \tag{1.16}$$

est précisément W . On dit que f a W comme domaine de maximalité.

Théorème 1.4.19. Soient P un polyèdre de \mathbb{R}^n et W un sous-ensemble de P . Alors W est un sous-ensemble exposé de P si et seulement si W est une face de P .

Démonstration. Soit W un sous-ensemble d'un polyèdre P .

1. (\Rightarrow) Supposons que W est un sous-ensemble exposé de P . Cela signifie qu'il existe une forme linéaire f vérifiant (1.16). Soit $z \in W$ et $x, y \in P$ tels que

$$z \in W \cap]x, y[\text{ et } z = \alpha x + \lambda y,$$

avec $\alpha, \lambda > 0$ et $\alpha + \lambda = 1$. Les relations (1.17) :

$$f(z) \geq f(x), \quad f(z) \geq f(y), \quad f(z) = \alpha f(x) + \lambda f(y) \tag{1.17}$$

sont toutes satisfaites si et seulement si $f(x) = f(y) = f(z)$.

2. (\Leftarrow) Supposons que W est une face de P . Cela signifie qu'il existe k hyperplans frontières

F^1, \dots, F^k tels que

$$W = P \cap \bigcap_{i \in \{1, \dots, k\}} \{F^i\}.$$

Soient w_1, \dots, w_k des réels strictement positifs, et posons $A_0 = \sum_{i=1}^k w_i A_i$, avec $A_0 \neq 0$. La forme linéaire définie par $f(x) := A_0^t x$ admet W comme domaine de maximalité. En effet, pour tout $z \in W$ et tout $x \in P$, on a

$$\begin{aligned} f(z) = A_0^t z &= \left(\sum_{i=1}^k w_i A_i \right)^t z = \sum_{i=1}^k w_i (A_i^t z) = \sum_{i=1}^k w_i b_i \\ &\geq \sum_{i=1}^k w_i (A_i^t x) \\ &\geq \left(\sum_{i=1}^k w_i A_i \right)^t x \\ &\geq A_0^t x = f(x). \end{aligned}$$

■

Corolaire 1.4.20. Soient P un polyèdre dans \mathbb{R}^n , f une forme linéaire et x_0 un point de P où f atteint sa valeur maximale sur P . Autrement dit, pour tout point $x \in P$, $f(x_0) \geq f(x)$. Alors cette valeur maximale est atteinte sur une face entière de P . En d'autres termes, il existe une face F de P telle que

$$f(x_0) = f(y) \text{ pour tous } y \in F.$$

Définition 1.4.21.

1. Le cône de récession du polyèdre P est le polyèdre $P_0 = \{x \in \mathbb{R}^n : Ax \leq 0\}$. Si P est un polytope, alors P_0 ne contient que le point 0, c'est-à-dire $\text{rang}(A) = n$.
2. Les éléments de P_0 sont appelés les rayons de P .
3. Un rayon r de P est appelé un rayon extrême de P s'il n'existe pas deux rayons distincts r_1 et r_2 de P tels que $r_1 \neq \lambda r_2$ pour tout $\lambda \geq 0$, et $r = \frac{1}{2}r_1 + \frac{1}{2}r_2$.



Une autre façon de caractériser les rayons extrêmes d'un polyèdre est donnée par le théorème suivant :

Théorème 1.4.22. Soit P un polyèdre non vide dans \mathbb{R}^n . Le vecteur r est un rayon extrême de P si et seulement si l'ensemble $\{\lambda r : \lambda \geq 0\}$ est une face de dimension 1 de P_0 .



CHAPITRE 2

ÉLÉGANTES APPROCHES POUR RÉSOLVRE LES PROBLÈMES DE PROGRAMMATION LINÉAIRES

Objectifs

À la Découverte des Mystères de la Programmation Linéaire : Une Exploration des Concepts et Techniques Fondamentaux [Ach84; Yil00; Rad07a; Kuc05]. Nous entamons notre périple en introduisant les Problèmes de Programmation Linéaire (PPLs) et en scrutant les particularités de leurs solutions. Par la suite, deux méthodes de résolution renommées pour les PPLs prennent place sous les feux des projecteurs : la méthode du simplexe, ainsi que la méthode adaptée aux cas où les variables du problème linéaire sont soumises à des bornes [Ben05; Bib98; Rad04; Del18a; Rad14; Rad11a; Kac21c; Kac23c]. Ces méthodes fournissent des outils puissants pour résoudre efficacement les problèmes de programmation linéaire, et leur utilisation s'est largement répandue dans divers domaines.

Sommaire

2.1	Émergence dans le Monde de la Programmation Linéaire	35
2.1.1	Les Problèmes de Programmation Linéaire	35
2.1.2	Les Points Extrêmes et les Solutions Optimales	40
2.2	Méthodes de Recherche Linéaire dans les PPL	42
2.2.1	Propriétés des Méthodes de Recherche Linéaire	42
2.2.2	Comparaison avec d'Autres Méthodes d'Optimisation	42
2.2.3	La Notion de Métrique dans les Algorithmes d'Optimisation	44
2.3	Méthodes de Recherche Linéaire : Méthode du Simplexe et Méthode Adaptée	45



2.3.1	Transformation de la Forme Standard à la Forme Canonique d'un PPL	45
2.3.2	La Méthode du Simplexe	47
2.3.3	La Méthode Adaptée	57

2.1 Émergence dans le Monde de la Programmation Linéaire

À la Quête des Solutions Optimums : Découvrons les problèmes récurrents dans divers domaines de la vie quotidienne, et comment les résoudre efficacement. Prenons, par exemple, le problème d'alimentation où une nutritionnisteⁱ souhaite concevoir un menu en utilisant deux aliments principaux **A** et **B**, en respectant les contraintes suivantes :

1. Chaque portion de 28 grammes d'aliment **A** contient 2 unités de matières grasses, 1 unité de glucides et 4 unités de protéines.
2. Chaque portion de 28 grammes d'aliment **B** contient 3 unités de matières grasses, 3 unités de glucides et 3 unités de protéines.
3. Le repas ne doit pas dépasser 38 unités de matières grasses, 32 unités de glucides et 44 unités de protéines.

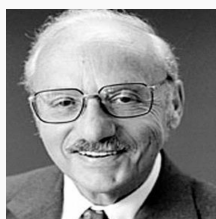
En tenant compte du coût, où une onceⁱⁱ d'aliment **A** coûte **20 DA** et une once d'aliment **B** coûte **25 DA**, la nutritionniste aspire à déterminer la meilleure répartition en onces de chaque aliment pour maximiser le coût du repas. Pour atteindre cet objectif, nous transformons ce défi en un problème mathématique pur et le résolvons à l'aide de techniques d'optimisation.



Comment devrions-nous procéder



2.1.1 Les Problèmes de Programmation Linéaire



Au milieu de l'année 1947, le mathématicien George Bernard Dantzig eut une idée révolutionnaire : modéliser le problème de planification des forces aériennes sous forme d'un système d'inéquations linéaires. Cela l'amena à développer une nouvelle théorie, qu'il consigna dans un document intitulé "Programmation dans une structure linéaire". En 1948, il décida de lui donner le nom que nous connaissons aujourd'hui : "Théorie de la Programmation Linéaire (PL)" [Dan97 ; Dan03 ; Dan63].

i. Spécialisé dans l'élaboration de plans alimentaires adaptés aux besoins de santé des individus.
ii. Mesure de la masse, dans le système impérial britannique, correspondant à 28 grammes environ.





En utilisant cette théorie, nous pouvons formuler le problème d'alimentation sous forme d'un programme linéaire, ce qui nous permettra de trouver la meilleure répartition en onces de chaque aliment en maximisant le coût du repas tout en respectant les contraintes nutritionnelles.

Définition 2.1.1 (Éclaircissement de la Programmation Linéaire). La programmation linéaire représente une discipline au sein de l'optimisation, axée sur la recherche d'optimums en maximisant ou minimisant une fonction linéaire tout en respectant des contraintes linéaires. La forme générale d'un PPL est la suivante :

$$\left\{ \begin{array}{l} \text{Trouver les valeurs de } x_1, x_2, \dots, x_n \text{ qui permettent de :} \\ \text{Maximiser (ou minimiser) } f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{sous les restrictions} \\ \left\{ \begin{array}{ll} a_{11}x_1 + \dots + a_{1n}x_n & (\leq, =, \geq) b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n & (\leq, =, \geq) b_2 \\ \vdots & \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n & (\leq, =, \geq) b_m. \end{array} \right. \end{array} \right. \quad (2.1)$$

Dans ce problème, f est la fonction objectif que nous cherchons à maximiser ou minimiser, et les inégalités $a_{ij}x_j \leq b_i$ représentent les contraintes linéaires. Les coefficients c_j , a_{ij} et b_i sont des constantes réelles. Les indices i et j sont des entiers naturels, avec $i = 1, \dots, m$ et $j = 1, \dots, n$, où n est le nombre de variables de décision et m est le nombre de contraintes du problème.



Dans chaque inégalité du problème (2.1), on utilise un et un seul des symboles suivants : \leq (inférieur ou égal), \geq (supérieur ou égal), ou $=$ (égal). Ces symboles définissent le type de contrainte linéaire associée à chaque inégalité.

Modélisation du cas de la nutritionniste sous forme d'un programme linéaire.

Soient $x_1 \geq 0$ et $x_2 \geq 0$ le nombre d'onces d'aliments **A** et **B**, respectivement, qui sont servis. Le nombre d'unités de matières grasses contenues dans le repas est $2x_1 + 3x_2$, de sorte que x_1 et x_2 doivent satisfaire l'inégalité $2x_1 + 3x_2 \leq 38$. De même, pour répondre aux besoins de la nutritionniste en glucides et en protéines, x_1 et x_2 doivent satisfaire les inégalités $x_1 + 3x_2 \leq 32$ et $4x_1 + 3x_2 \leq 44$.



Le coût du repas, qui doit être maximisé, est donné par $z = 20x_1 + 25x_2$. Alors, le modèle mathématique se présente comme suit : trouver les valeurs de x_1 et x_2 qui maximisent la fonction objectif $z = 20x_1 + 25x_2$ sous les contraintes suivantes :

$$\begin{cases} 2x_1 + 3x_2 \leq 38 \\ x_1 + 3x_2 \leq 32 \\ 4x_1 + 3x_2 \leq 44 \\ x_1, x_2 \geq 0. \end{cases} \quad (2.2)$$



On utilisera une méthode appelée la méthode du simplexe pour résoudre le problème (2.2). Si le problème est réalisable, c'est-à-dire s'il a une solution, nous pourrions fournir à la nutritionniste la solution optimale qui répond à ses exigences pour le menu. Sinon, elle devra ajuster ses exigences en conséquence.

Terminologies 2.1.2. Voici les terminologies importantes utilisées dans le contexte de la programmation linéaire :

- ◇ La fonction f est appelée la fonction objectif (ou fonction économique).
- ◇ Les égalités et/ou inégalités dans (2.1) sont désignées sous le nom de contraintes.
- ◇ Les variables x_1, \dots, x_n sont appelées variables de décision.
- ◇ Une contrainte est qualifiée de linéaire (ou non linéaire) si le côté gauche de l'inégalité ou de l'équation dans (2.1) est une fonction linéaire (ou non linéaire).



Notez que les contraintes du problème (2.2) sont linéaires.

- ◇ Si le problème de programmation contient au moins une contrainte non linéaire ou si la fonction objectif n'est pas linéaire, alors il est qualifié de problème de programmation non linéaire.
- ◇ Le PPL (2.1) peut être exprimé sous les deux formes suivantes :

1. La forme standard dans le cas d'un problème de maximisation :

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad \text{et} \quad x_j \geq 0, \quad \text{pour tout } i, j.$$

2. La forme canonique dans le cas d'un problème de maximisation :

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \quad \text{et} \quad x_j \geq 0, \quad \text{pour tout } i, j.$$





Chaque problème de minimisation peut être transformé en un problème de maximisation, et vice versa, en utilisant la propriété suivante :

$$\min_x \sum_{i=1}^n c_i x_i = -\max_x \left(-\sum_{i=1}^n c_i x_i \right).$$

Cela signifie que pour trouver le minimum de la fonction $\sum_{i=1}^n c_i x_i$, on peut rechercher le maximum de la fonction $-\sum_{i=1}^n c_i x_i$ et vice versa.

Chaque PPL comportant des variables de signe quelconque peut être résolu en reformulant un programme linéaire équivalent, où toutes les variables sont contraintes d'être non négatives. En outre, il est possible de représenter chaque PPL soit sous une forme standard correspondante, soit sous une forme canonique correspondante.

En résumé, pour tout PPL donné, il existe un problème de PL standard (ou un problème de PL canonique) dont la solution équivaut à celle du PPL. Pour approfondir ces notions, je vous recommande de consulter la référence [Kol95]. Cette propriété est cruciale car elle facilite la résolution et l'optimisation des problèmes de programmation linéaire dans des conditions normalisées.

Définition 2.1.3.

1. Un vecteur x dans \mathbb{R}^n satisfaisant les contraintes d'un problème de programmation linéaire est appelé une **solution réalisable**.
2. Une solution réalisable qui maximise ou minimise la fonction objectif d'un problème de programmation linéaire est appelée une **solution optimale**.
3. Supposons que le PPL est sous forme standard. Alors, l'ensemble

$$S = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$$

est appelé **région réalisable**.



Il est important de noter que l'ensemble de toutes les solutions possibles d'un problème général de PL forme un polyèdre convexe.

Le théorème suivant démontre que si un PPL possède au moins une solution optimale, alors l'une de ses solutions est un point extrême de la région réalisableⁱⁱⁱ. Cependant, une solution optimale peut être une solution réalisable qui n'est pas un point extrême. Pour élaborer une méthode de recherche de solutions, il suffit de considérer uniquement les points extrêmes. Heu-

iii. Dans \mathbb{R}^3 , un point extrême d'un ensemble est un véritable coin.



reusement, il n'y a qu'un nombre fini de points extrêmes, bien que ce nombre puisse être très important. Par la suite, nous expliquerons comment rechercher de manière méthodique la solution optimale en examinant les points extrêmes en un nombre relativement restreint d'étapes.

Théorème 2.1.4 (Points extrêmes). Soit S la région réalisable d'un PPL général.

1. Si S est un polytope non vide, alors au moins l'une des solutions optimales du problème est point extrême de S . Cela implique que la solution optimale est soit un sommet, soit un point situé sur une arête du polytope S .
2. Si S est un polyèdre non borné et qu'au moins une solution optimale du problème existe, alors au moins l'une des solutions optimales est un point extrême de S .
3. Si le problème n'admet pas de solution optimale, alors cela peut être dû à deux raisons : soit S est vide, soit S est non borné.



Le théorème 2.1.4 est essentiel pour comprendre les propriétés de S et déterminer les solutions optimales dans les problèmes de PL. Il fournit des indications importantes pour la recherche et l'analyse des solutions optimales possibles.

Notations 2.1.5. Considérons le problème de PL suivant sous sa forme standard :

$$\left\{ \begin{array}{l} \text{Maximiser } f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{sous les restrictions} \\ a_{i1}x_1 + \dots + a_{in}x_n \leq b_i \quad \text{et } x_j \geq 0, \quad \text{avec } \begin{array}{l} i = 1, \dots, m \\ j = 1, \dots, n \end{array} \end{array} \right. \quad (2.3)$$

Posons :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

Alors, le problème (2.3) peut être réécrit de manière plus concise comme suit :

$$\left\{ \begin{array}{l} \max_x f(x) = c^t x \\ Ax \leq b \\ x \geq 0. \end{array} \right. \quad (2.4)$$



2.1.2 Les Points Extrêmes et les Solutions Optimales

Les points extrêmes peuvent être difficiles à calculer géométriquement lorsque le problème comporte plus de trois variables. Afin de faciliter leur calcul, nous allons fournir une description algébrique des points extrêmes. Cette description repose sur le concept de solution de base d'un problème de programmation linéaire. Avant cela, nous présenterons deux théorèmes généraux très importants concernant les problèmes de programmation linéaire.

Considérons le problème de programmation linéaire sous forme canonique :

$$\begin{cases} \max_x f(x) = c^t x \\ Ax = b \\ x \geq 0, \end{cases} \quad (2.5)$$

où A est une matrice de taille $m \times n$, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, et $b \in \mathbb{R}^m$, nous faisons deux hypothèses sur la matrice de contraintes A . Premièrement, nous supposons que le nombre de lignes est inférieur ou égal au nombre de colonnes ($m \leq n$)^{iv}. Deuxièmement, nous supposons que A contient m vecteurs colonnes qui sont linéairement indépendants, c'est-à-dire que le rang de A est égal à m . Nous pouvons toujours renuméroter les colonnes de A (en réorganisant les composantes de x), de sorte que les dernières m colonnes de A soient linéairement indépendantes. Ces hypothèses sont toujours vérifiées pour un problème de programmation linéaire sous forme canonique qui découle d'un problème sous forme standard. Les deux théorèmes suivants établissent un lien direct entre les points extrêmes et la solution optimale d'un problème de programmation linéaire canonique.

Théorème 2.1.6.

1. Supposons que

- (a) Les m dernières colonnes a_1, \dots, a_m de A sont linéairement indépendantes.
- (b) Soit $y \in \mathbb{R}^m$, tel que $y_i \geq 0$ pour tout $i = 1, \dots, m$ vérifiant :

$$y_1 a_1 + \dots + y_m a_m = b.$$

Alors, le point $x = (0, 0, \dots, 0, y_1, y_2, \dots, y_m)$ est un point extrême de la région réalisable associée au problème (2.5).

2. Inversement, si $x = (x_1, x_2, \dots, x_n)$ est un point extrême de la région réalisable associée au problème (2.5), alors les colonnes de A qui correspondent à $x_j \geq 0$ forment un ensemble de vecteurs de \mathbb{R}^m linéairement indépendants.

Démonstration. Veuillez consulter le premier chapitre de [Kol95]. ■

iv. Afin d'assurer que la région réalisable n'est pas vide, voir les propriétés 1.4.4.



Théorème 2.1.7. Dans tout point extrême de la région réalisable associée au problème (2.5), au plus m composantes peuvent être positives, et les autres ($n - m$ composantes) sont égales à zéro.



En exploitant les décompositions (1.1) et (1.2), nous réécrivons le système d'équations $Ax = b$ de la manière suivante :

$$A_B x_B + A_N x_N = b. \quad (2.6)$$

Définition 2.1.8. Supposons que A_B est inversible et $x_N = 0$. Dans ce cas, le système (2.6) admet une unique solution donnée par (x_B, x_N) , où :

$$x_B = A_B^{-1} b.$$

On peut alors définir les termes suivants :

- ◇ Les composantes de x_B sont appelées **variables basiques**.
- ◇ Les composantes de x_N sont appelées **variables non basiques**.
- ◇ Le vecteur $x = (x_B, x_N)$ est appelé **solution basique**.
- ◇ Une solution basique et réalisable est appelée **solution de base réalisable**, c'est-à-dire une solution qui est à la fois réalisable et basique.



Une solution de base réalisable est particulièrement importante dans la résolution des problèmes de **PL** car elle représente un point extrême de la région réalisable. En effet, la combinaison des variables basiques fixées à leurs valeurs optimales (déterminées par $x_B = A_B^{-1} b$) et des variables non basiques fixées à zéro forme un point de la région réalisable qui correspond à une solution optimale.

Théorème 2.1.9. Le problème de programmation linéaire (2.5) admet un nombre fini de solutions basiques, égal au maximum à $\binom{n}{m} = \frac{n!}{m!(n-m)!}$.

De plus, il existe une correspondance bijective entre les solutions de base réalisables et les points extrêmes de la région réalisable. Cela signifie que chaque solution de base réalisable est un point extrême et réciproquement, chaque point extrême est une solution de base réalisable.



2.2 Méthodes de Recherche Linéaire dans les PPL

Les problèmes d'optimisation linéaire jouent un rôle essentiel dans de nombreux domaines tels que l'économie, l'ingénierie, la logistique, la finance et bien d'autres. Ces problèmes consistent à trouver la meilleure valeur d'une fonction linéaire, appelée fonction objectif, tout en respectant un ensemble de contraintes linéaires qui définissent la région réalisable du problème. Pour résoudre ces problèmes d'optimisation linéaire, on utilise des méthodes de recherche linéaire. Ces approches itératives explorent la région réalisable pour atteindre la solution optimale.

2.2.1 Propriétés des Méthodes de Recherche Linéaire

La recherche linéaire est une méthode itérative utilisée pour explorer une région convexe afin de trouver le point optimal qui maximise ou minimise la fonction objectif. Dans le cas des problèmes d'optimisation linéaire, la région réalisable est un polytope convexe défini par des contraintes linéaires. Les méthodes de recherche linéaire se déplacent le long des arêtes de ce polytope en utilisant des pivots pour atteindre la solution optimale. Elles présentent plusieurs propriétés et avantages. Ces méthodes garantissent la convergence vers une solution optimale pour les problèmes d'optimisation linéaire convexe. De plus, elles sont efficaces pour résoudre des problèmes de grande taille avec un nombre élevé de variables et de contraintes.

2.2.2 Comparaison avec d'Autres Méthodes d'Optimisation

Dans cette section, nous examinerons différentes méthodes d'optimisation, notamment les méthodes de recherche linéaire, et les comparerons avec d'autres approches utilisées pour résoudre des problèmes d'optimisation. Chaque méthode possède des avantages et des inconvénients, et leur choix dépend de la nature du problème à résoudre et des contraintes spécifiques.

2.2.2.1 Méthodes de Recherche Linéaire (y compris le Simplexe)

Les méthodes de recherche linéaire, dont le Simplexe est un exemple emblématique, sont spécialement conçues pour résoudre des problèmes d'optimisation linéaire où la fonction objectif et les contraintes sont linéaires. Leur efficacité réside dans leur capacité à explorer directement les régions convexes, permettant ainsi une convergence rapide vers la solution optimale. Le Simplexe est largement utilisé et très efficace pour les problèmes d'optimisation linéaire. Cependant, ces méthodes peuvent rencontrer des limitations lorsqu'elles sont appliquées à des problèmes non linéaires, car elles risquent de converger vers des solutions locales plutôt que la solution globale.

2.2.2.2 Méthodes de Gradient

Les méthodes de gradient sont principalement utilisées pour résoudre des problèmes d'optimisation non linéaire. Elles impliquent le calcul itératif des gradients pour déterminer la direction de recherche vers la solution optimale. Ces méthodes peuvent être très efficaces pour des problèmes avec des fonctions continues et dérivables, mais elles peuvent également présenter des problèmes de convergence vers des solutions locales. Le choix du point de départ est crucial pour obtenir la meilleure solution possible, et certaines variantes de ces méthodes permettent de surmonter ce problème.

2.2.2.3 Programmation Dynamique

La programmation dynamique est une approche puissante pour résoudre des problèmes d'optimisation en décomposant le problème en sous-problèmes plus petits et en construisant la solution globale à partir des résultats de ces sous-problèmes. Cette méthode est particulièrement utile pour les problèmes de grande taille avec des structures récurrentes.

2.2.2.4 Algorithmes d'Énumération

Les algorithmes d'énumération examinent de manière exhaustive toutes les combinaisons possibles des variables pour trouver la solution optimale. Ces méthodes sont particulièrement adaptées aux problèmes de petite taille où l'espace de recherche est limité. Cependant, pour les problèmes de grande dimension, ces méthodes deviennent rapidement inefficaces en raison de la complexité combinatoire, ce qui les rend moins appropriées pour ces cas.

2.2.2.5 Méthodes Heuristiques

Les méthodes heuristiques sont des approches pragmatiques qui visent à trouver des solutions satisfaisantes sans garantie de trouver la solution optimale. Elles sont souvent utilisées pour résoudre des problèmes complexes et de grande dimension où les méthodes classiques peuvent être inefficaces. Les méthodes heuristiques sont capables de converger rapidement vers une solution, mais le résultat obtenu peut varier en fonction des paramètres de la méthode et de la nature du problème.

2.2.2.6 Métaheuristiques

Les métaheuristiques sont des techniques d'optimisation basées sur des principes d'apprentissage, de recherche aléatoire ou d'évolution. Elles sont capables de surmonter les pièges locaux et offrent une approche plus globale pour résoudre des problèmes d'optimisation complexes. Les métaheuristiques sont particulièrement utiles lorsque les méthodes classiques ne donnent pas de résultats satisfaisants, mais elles nécessitent souvent un grand nombre d'itérations pour converger vers une solution acceptable.



2.2.3 La Notion de Métrique dans les Algorithmes d'Optimisation

Dans le domaine de l'optimisation, la métrique joue un rôle fondamental pour guider les algorithmes vers des solutions optimales en évaluant la distance ou le changement entre différents points dans l'espace des solutions. Elle permet de déterminer la direction de déplacement des variables lors des itérations de l'algorithme afin de converger vers une solution optimale.

2.2.3.1 Définition de la Métrique

La métrique, dans le contexte de l'optimisation, est une mesure ou une règle qui permet de quantifier la variation entre deux points dans l'espace des solutions. Elle s'exprime généralement sous forme de formule mathématique qui prend en compte les caractéristiques spécifiques du problème d'optimisation.

2.2.3.2 La Métrique du Simplexe dans le Simplexe Classique

Dans la méthode du simplexe classique, la métrique du simplexe est utilisée pour choisir une variable entrante et une variable sortante à chaque itération. Elle permet de se déplacer le long des arêtes du polyèdre défini par les contraintes du problème. La métrique du simplexe assure que le déplacement entre deux points dans l'espace des solutions se fait le long d'une direction admissible, respectant les contraintes.

2.2.3.3 Utilisation de la Métrique dans l'Optimisation Linéaire à Variables Bornées

Dans le cas de l'optimisation linéaire à variables bornées, la métrique est utilisée pour déterminer la direction d'amélioration à partir d'un plan support donné vers un nouveau plan support. Elle est essentielle pour guider l'algorithme vers des solutions réalisables qui respectent les contraintes et les bornes sur les variables.

2.2.3.4 Métrique Adaptée du Simplexe dans la Méthode de Gabasov

Dans la méthode adaptée de Gabasov, la métrique du simplexe est adaptée spécifiquement pour cette approche itérative. Contrairement au simplexe classique, cette métrique peut faire varier plusieurs indices simultanément lors des itérations. Elle est conçue pour permettre des variations adaptées des variables en fonction du plan support actuel. Cela rend l'algorithme plus efficace et capable de gérer des problèmes de grande taille avec un grand nombre de variables et de contraintes.

2.3 Méthodes de Recherche Linéaire : Méthode du Simplexe et Méthode Adaptée

L'approche fondamentale sur laquelle repose la méthode du simplexe et la méthode adaptée de R. Gabasov est la méthode de recherche linéaire pour trouver une distance optimale le long de la direction donnée tout en restant dans la région réalisable S . Cette méthode est couramment utilisée dans les algorithmes d'optimisation sous contraintes.

Les étapes générales pour effectuer une recherche linéaire dans la direction d donnée

1. Choisissez une direction $d \in \mathbb{R}^n$ dans laquelle vous souhaitez vous déplacer tout en améliorant la valeur de la fonction objectif.
2. Déterminez une distance maximale que vous pouvez parcourir le long de cette direction sans quitter la région réalisable S . Cela peut être obtenu en résolvant un problème d'optimisation sous contraintes, où la contrainte est de rester dans la région réalisable S .
3. Effectuez une recherche linéaire dans la direction d en utilisant la distance maximale calculée à l'étape précédente. L'idée est de trouver une distance optimale le long de la direction d qui permet d'augmenter la valeur de la fonction objectif sans violer les contraintes de la région réalisable S .
4. Mettez à jour la solution réalisable x en vous déplaçant le long de la direction d avec la distance optimale trouvée à l'étape précédente.
5. Répétez les étapes 2 à 4 jusqu'à ce que vous atteigniez un critère d'arrêt spécifié.



Attention : Toute cette section et ce travail se focalisent spécifiquement sur les critères d'optimalité et les méthodes conçus pour résoudre des problèmes de maximisation. Les approches pour les problèmes de minimisation peuvent présenter des légères différences, telles que le changement de signe des vecteurs d'estimation et l'inversion des inégalités. Cependant, les principes fondamentaux restent généralement similaires.

2.3.1 Transformation de la Forme Standard à la Forme Canonique d'un PPL

Le passage de la forme standard d'un programme linéaire à la forme canonique revêt une grande importance dans la résolution efficace de problèmes d'optimisation linéaire. La forme standard consiste à transformer les contraintes d'inégalités en égalités en introduisant des variables d'écart et des variables artificielles. Une fois que le programme linéaire est exprimé en



forme canonique, il devient plus facile de l'analyser et de l'utiliser avec des méthodes d'optimisation telles que le Simplexe ou d'autres algorithmes linéaires.

La forme canonique est particulièrement utile pour plusieurs raisons :

1. Les algorithmes d'optimisation linéaire, comme le Simplexe, sont spécialement conçus pour résoudre des problèmes exprimés sous forme canonique. Cette représentation facilite grandement la résolution du problème en utilisant directement ces algorithmes.
2. Stabilité numérique améliorée, réduisant les risques de divergence ou de comportement instable lors des calculs.
3. Interprétation et analyse simplifiées grâce aux variables d'écart et aux variables artificielles, permettant de mieux comprendre les contraintes et les conditions du problème. Cela facilite l'interprétation des résultats et l'analyse des solutions optimales.



Nous supposons que $b \geq 0$. Pour le cas général, où b peut avoir des valeurs négatives, on peut introduire des variables artificielles, comme décrit dans [Dan97].

Considérons maintenant le PPL sous forme standard suivant :

$$\begin{cases} \max_x f(x) = c^t x \\ \hat{A}x \leq b \\ x \geq 0, \end{cases} \quad (2.7)$$

où \hat{A} est une matrice de taille $m \times n$, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, et $b \in \mathbb{R}^m$. Pour obtenir la forme canonique du problème (2.7), il est possible d'introduire des variables d'écart e_1, \dots, e_m de la manière suivante :

$$\begin{cases} \max_x f(x) = c^t x \\ Ax = b \\ x \geq 0, \end{cases} \quad (2.8)$$

où $A = (\hat{A}, Id_{\mathbb{R}^m})$, dans ce cas est une matrice de taille $m \times (n + m)$, $c \in \mathbb{R}^{n+m}$, $x \in \mathbb{R}^{n+m}$, et $Id_{\mathbb{R}^m}$ est la matrice identité de taille $m \times m$, définie comme suit :

$$A = (\hat{A}, Id_{\mathbb{R}^m}) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix}.$$





Notons S la région réalisable du problème (2.8), définie par :

$$S := \{x \in \mathbb{R}^{n+m} : Ax = b, x \geq 0\}.$$

Définition 2.3.1. Deux points extrêmes distincts dans S sont dits adjacents s'ils ne diffèrent que d'une et une seule variable de base, en tant que solutions de base réalisables.

2.3.2 La Méthode du Simplexe

Dans cette section, nous explorerons la méthode du simplexe, une avancée développée par George Dantzig en 1947 pour résoudre les problèmes de programmation linéaire [Eva73; Wol59]. Les solutions de tels problèmes se trouvent aux points extrêmes de l'ensemble des solutions réalisables. La méthode consiste à systématiquement déterminer ces points extrêmes dans un ordre particulier, permettant ainsi de trouver une solution optimale en un minimum d'itérations. Nous nous concentrons sur les problèmes formulés sous forme canonique, car cette approche facilite la recherche d'un point de départ adéquat. Pour les problèmes de programmation linéaire en forme générale, nous renvoyons le lecteur à [Dan97].

2.3.2.1 Le Fondement de l'Algorithme : Comprendre son Principe Fondamental

L'algorithme du simplexe se déroule en deux étapes. Tout d'abord, il permet de vérifier si une solution de base réalisable donnée est optimale. Ensuite, il vise à obtenir une solution de base réalisable adjacente ayant une valeur égale ou supérieure pour la fonction objectif. Dans la pratique, la méthode du simplexe n'examine pas toutes les solutions de base possibles, ce qui serait généralement impraticable. Au lieu de cela, elle se concentre sur un nombre relativement restreint de solutions potentielles, rendant ainsi le processus de résolution plus efficace.

2.3.2.2 Exploration de l'Algorithme du Simplexe

L'approche sur laquelle le Simplexe est basé est la méthode de recherche linéaire en utilisant des pivots pour se déplacer le long des arêtes de la région réalisable d'un problème d'optimisation linéaire. Le Simplexe est un algorithme classique utilisé pour résoudre des problèmes d'optimisation linéaire où l'objectif est de maximiser ou minimiser une fonction linéaire sous des contraintes linéaires.



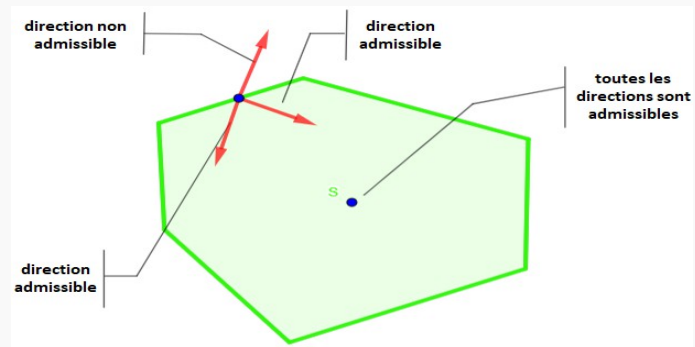


Figure 2.1 – Directions admissibles

Un aperçu de l'approche du Simplexe

1. Initialisation : Commencez avec une solution réalisable du problème. Le Simplexe débute généralement à un sommet de la région réalisable.
2. Recherche de la direction : Le Simplexe sélectionne une direction le long de laquelle il peut déplacer la solution vers un sommet voisin tout en améliorant la valeur de la fonction objectif.
3. Pivot : Le Simplexe utilise une règle de pivot pour déterminer quelle variable doit quitter la base et quelle variable doit y entrer afin de se déplacer vers le sommet voisin tout en restant dans la région réalisable. Cela garantit que les contraintes du problème sont toujours satisfaites.
4. Mise à jour de la solution : La solution est mise à jour en déplaçant le long de la direction choisie lors de l'étape de pivot.
5. Répétition : Les étapes 2 à 4 sont répétées jusqu'à ce qu'une solution optimale soit atteinte ou qu'une solution optimale soit déterminée comme non réalisable.

Le Simplexe itère à travers les sommets de la région réalisable jusqu'à atteindre une solution optimale. En raison de la stratégie de recherche linéaire et de la sélection judicieuse des pivots, le Simplexe converge généralement vers une solution optimale dans un nombre fini d'itérations pour les problèmes d'optimisation linéaire bien formulés.

D'autres approches modernes, comme la programmation linéaire en nombres entiers et mixtes, sont utilisées pour des problèmes plus complexes.

Définition 2.3.2 (Direction admissible). Soit x un élément du polyèdre S . On définit un vecteur $d \in \mathbb{R}^{n+m}$ comme étant une direction admissible en x s'il existe un scalaire positif θ tel que le point $x + \theta d$ soit également admissible, c'est-à-dire que $x + \theta d \in S$. En d'autres termes, le vecteur d représente une direction dans laquelle on peut se déplacer depuis x tout en restant dans le polyèdre S .

Définition 2.3.3 (Solution de base admissible non dégénérée). Une solution de base admissible non dégénérée est définie comme étant une solution de base réalisable dont toutes les variables basiques sont strictement positives, c'est-à-dire $x_B > 0$. Si au moins une variable basique est égale à zéro, la solution de base est considérée comme dégénérée.



- ◇ Variables de base x_B : Ce sont celles qui prennent des valeurs non nulles dans une solution de base admissible non dégénérée, déterminant les valeurs des contraintes.
- ◇ Variables hors base x_N : Ce sont celles qui ont des valeurs nulles dans une solution de base et peuvent être modifiées pour explorer différentes solutions ou atteindre l'optimalité en utilisant la méthode du Simplexe.

Navigation dans la région réalisable S : La $i_0^{\text{ème}}$ direction de base

Pour déterminer $x + \theta d$, on choisit d'abord une variable d'indice $i_0 \in J_N$ (hors base) qui vaut 0. Ensuite, on augmente sa valeur jusqu'à atteindre θ , tout en maintenant les autres variables hors base à zéro. Autrement dit, on affecte la valeur 1 à d_{i_0} , et pour tout indice de variable non basique i différent de i_0 , on lui attribue la valeur zéro. Ainsi, on définit d_N comme suit :

$$d_i = \begin{cases} 1, & \text{si } i = i_0 \\ 0, & \text{si } i \neq i_0 \end{cases}, \quad \text{pour tout } i \in J_N.$$

Pour rester confiné dans S , la solution $x + \theta d$ doit satisfaire les contraintes linéaires :

$$A(x + \theta d) = b \Leftrightarrow Ax + \theta Ad = b \Leftrightarrow b + \theta Ad = b \Leftrightarrow Ad = 0. \quad (2.9)$$

En utilisant la décomposition (1.1), la condition (2.9) devient

$$Ad = 0 \Leftrightarrow \sum_{i \in J_B} a_i d_i + \sum_{i \in J_N} a_i d_i = 0 \Leftrightarrow \sum_{i \in J_B} a_i d_i + a_{i_0} d_{i_0} = 0 \Leftrightarrow A_B d_B + a_{i_0} = 0,$$

d'où :

$$d_B = -A_B^{-1} a_{i_0}. \quad (2.10)$$

d_B est ce qu'on appelle la $i_0^{\text{ème}}$ direction de base.



La $i_0^{\text{ème}}$ direction de base : Contraintes satisfaites, variables non basiques positives

La $i_0^{\text{ème}}$ direction de base d_B garantit que les contraintes d'égalité seront satisfaites lorsque l'on s'éloigne de x le long de d , tout en assurant que les composantes non basiques $x_i + \theta d_i$ resteront positives ou nulles. x_{i_0} était nul et devient $x_{i_0} + \theta$, qui est un nombre positif. x_i pour $i \neq i_0$ restent à zéro car $d_i = 0$. Si de plus, x est une solution de base admissible non dégénérée ($x_B > 0$), alors lorsque θ est suffisamment petit, les variables basiques resteront positives, c'est-à-dire :

$$x_B + \theta d_B \geq 0.$$



Attention, en cas de dégénérescence d'une solution de base admissible, la direction d ne sera pas toujours admissible. Cela se produit lorsque au moins une des variables de base est égale à zéro et que la composante correspondante de la direction d est négative.

Proposition 2.3.4. Si x est une solution de base admissible non dégénérée, alors chaque direction de base correspondant à un indice de base $j \in J_B$ est admissible en x .

Effet du déplacement dans une direction admissible sur la fonction objectif

Examinons maintenant l'effet du déplacement le long d'une direction admissible d sur la fonction objectif. Lorsque nous quittons x et nous déplaçons dans la direction d sur la région réalisable S , la fonction objectif subit une modification donnée par :

$$c^t(x + \theta d) = c^t x + \theta c^t d.$$

Nous observons l'apparition d'un nouveau terme $\theta c^t d$ qui est ajouté à l'ancienne valeur de la fonction objectif en x . Ce terme représente le taux de modification de la valeur de la fonction objectif le long de d . En utilisant la décomposition (1.1) et la définition de d_N , on obtient :

$$c^t d = \sum_{i \in J_B} c_i d_i + \sum_{i \in J_N} c_i d_i = \sum_{i \in J_B} c_i d_i + c_{i_0} = c_B^t d_B + c_{i_0},$$

où i_0 est l'indice correspondant à une variable non basique choisie pour la direction d .



En substituant d_B donné par (2.10), nous obtenons :

$$c^t d = \begin{cases} c_{i_0} - c_B^t A_B^{-1} a_{i_0} & \text{si } i_0 \in J_N, \\ 0 & \text{si } i_0 \in J_B. \end{cases} \quad (2.11)$$

En effet, si i_0 est un indice correspondant à une variable basique, alors $A_B^{-1} a_{i_0} = e_{i_0}$, où e_{i_0} est la $i_0^{\text{ème}}$ colonne de la matrice identité de taille m . Cela donne :

$$c^t d = c_{i_0} - c_B^t A_B^{-1} a_{i_0} = c_{i_0} - c_B^t e_{i_0} = c_{i_0} - c_{i_0} = 0.$$



Cela signifie que si la variable choisie i_0 pour la direction d est une variable basique, alors le taux de modification de la fonction objectif sera nul, c'est-à-dire qu'elle ne change pas le long de cette direction. En revanche, si i_0 correspond à une variable non basique, le taux de modification de la fonction objectif sera donné par $c_{i_0} - c_B^t A_B^{-1} a_{i_0}$.

Définition 2.3.5 (Taux d'accroissement de la fonction objectif). On appelle **taux d'accroissement de la fonction objectif** le vecteur défini dans (2.11), noté par "**E**".

Théorème 2.3.6 (Conditions d'optimalité). Soient x une solution de base admissible, A_B la matrice de base associée, et "**E**" le taux d'accroissement de la fonction objectif correspondant. Alors :

1. Si $\mathbf{E} \leq 0^v$, alors x est une solution optimale.
2. Inversement, si x est une solution optimale et non dégénérée, alors $\mathbf{E} \leq 0$.

v. Il est intéressant de souligner le fait que la condition $\mathbf{E} \leq 0$ pour une solution optimale devient $\mathbf{E} \geq 0$ dans le cas de minimisation.

Vers l'optimalité : Sélection de la direction admissible avec le simplexe

Supposons qu'aucune solution de base admissible ne soit dégénérée. Étant donné une solution de base admissible x , on calcule les taux d'accroissement \mathbf{E}_i pour les variables hors base. Si tous les taux d'accroissement sont négatifs, alors x est une solution optimale et l'algorithme s'arrête.

Sinon, il existe une variable hors base x_{i_0} dont le taux d'accroissement \mathbf{E}_{i_0} est positif. Dans ce cas, la $i_0^{\text{ème}}$ direction de base est une direction admissible le long de laquelle le taux d'accroissement \mathbf{E}_{i_0} augmente, ce qui signifie que x_{i_0} entre dans la base.



Cependant, pour rester dans la région réalisable S tout en allant le plus loin possible dans la direction d , nous devons trouver un θ_0 tel que :

$$\theta_0 = \max \{ \theta \geq 0 : x + \theta d \in S \}.$$

Si $d \geq 0$, alors nous sommes toujours dans S (car $x + \theta d \geq 0$), mais dans ce cas, S n'est pas borné, et l'algorithme s'arrête.

Sinon, d a une composante négative d'indice i , et la contrainte $x_i + \theta d_i \geq 0$ devient :

$$\theta \leq -\frac{x_i}{d_i},$$

donc θ_0 doit vérifier toutes les contraintes :

$$\theta_0 = \min_{\{i: d_i < 0\}} \left(-\frac{x_i}{d_i} \right).$$



Il est important de noter que si x_i est une variable hors base, alors $d_i \geq 0$ (car $d_i = 0$ ou $d_i = 1$), et donc, il suffit de considérer les variables de base :

$$\theta_0 = \min_{\{d_i < 0: i \in J_B\}} \left(-\frac{x_i}{d_i} \right).$$

Comme x est non dégénéré ($x_B > 0$), cela garantit que $\theta_0 > 0$ pour tout $i \in J_B$.

Finalement, si la $i_0^{\text{ème}}$ direction de base d et j_0 est un indice tel que :

$$\theta_0 = -\frac{x_{j_0}}{d_{j_0}} = \min_{\{j \in J_B: d_j < 0\}} \left(-\frac{x_j}{d_j} \right),$$

alors on dit que x_{i_0} entre dans la base et x_{j_0} sort de la base. Cela nous permet de réaliser un pivot pour passer à la prochaine solution de base admissible et continuer l'algorithme jusqu'à atteindre une solution optimale.

L'algorithme 1 est exécuté pour obtenir une séquence de solutions de base admissibles jusqu'à ce qu'une solution optimale soit atteinte. Cependant, dans certaines situations, la méthode peut ne pas converger ou rencontrer des cas de dégénérescence, ce qui peut poser des défis supplémentaires. Pour pallier ces problèmes, des techniques spéciales peuvent être mises en œuvre. Par exemple, pour les problèmes de convergence, il est possible de modifier les règles de sélection des variables entrantes ou sortantes, d'ajuster les valeurs initiales ou d'explorer des méthodes alternatives pour trouver une solution réalisable.

Algorithme 1 Algorithme du simplexe mono-critère**Entrée :** Matrice A , Vecteur c , Vecteur b |**Sortie :** Solution optimale x |**Étape 1 :** Choisir une base J_B (ensemble d'indices basiques) et calculer la solution de base admissible x associée à A_B |**Étape 2 :** Calculer les taux d'accroissements \mathbf{E}_i pour chaque indice $i \in J_N$ hors base : $\mathbf{E}_i = c_i - c^t A_B^{-1} a_i$ |**tant que** *il existe un indice $i \in J_N$ tel que $\mathbf{E}_i > 0$* **faire**

1. Choisir i_0 tel que $\mathbf{E}_{i_0} > 0$ et calculer la $i_0^{\text{ème}}$ direction de base :

$$d_B = A_B^{-1} a_{i_0} \quad |$$

si aucune composante de d_B n'est négative **alors**

Le taux d'accroissement est optimal et infini |

Le problème n'a pas de solution optimale |

Renvoyer x |**fin**

2. Calculer le pas d'optimisation θ_0 :

$$\theta_0 = -\frac{x_{j_0}}{d_{j_0}} = \min \left\{ -\frac{x_j}{d_j} : d_j < 0, \text{ pour tous } j \in J_B \right\} \quad |$$

3. Former une nouvelle base en remplaçant a_{i_0} par a_{j_0} |

4. Si y est la nouvelle solution de base admissible, les valeurs des nouvelles variables basiques sont :

$$\diamond y_{j_0} = \theta_0.$$

$$\diamond y_{i_0} = x_{i_0} + \theta_0 d_{i_0} \quad |$$

fin

Oltre la méthode que nous venons de présenter, il existe plusieurs procédures plus avancées et polyvalentes. Le simplexe du grand \mathcal{M} est l'une de ces approches, exploitant le concept de variables artificielles^{vi} [Ko195]. Une autre méthode est celle du simplexe multi-critère, particulièrement utile pour résoudre des problèmes de programmation multi-critère^{vii}.

vi. Ces variables sont introduites dans le programme initial pour surmonter certaines limitations de la méthode standard.

vii. Ce type de programmation gère des objectifs multiples, comme expliqué dans le chapitre 4.





Bien que nous n'examinerons pas ces méthodes en détail, elles seront utilisées implicitement pour résoudre certains exemples complexes.



La métrique dans cet algorithme est implémentée de façon implicite à travers les taux d'accroissements E_i calculés à l'étape 2. Ces taux mesurent la variation de la fonction objectif $c^t x$ lorsque les variables x_i changent d'une petite quantité.

2.3.2.3 Analyse de la Complexité du Simplexe et Perspectives d'Approches Alternatives

L'approche du Simplexe est basée sur une méthode de recherche linéaire utilisant des pivots pour se déplacer le long des arêtes de la région réalisable d'un problème d'optimisation linéaire. Cet algorithme classique est utilisé pour résoudre des problèmes où l'objectif est de maximiser ou minimiser une fonction linéaire sous des contraintes linéaires.

Il est important de noter que le comportement du Simplexe peut varier selon les problèmes spécifiques. En moyenne, pour certains problèmes bien formulés, le Simplexe peut avoir une complexité linéaire, ce qui signifie que le nombre d'itérations nécessaires pour atteindre la solution optimale croît linéairement avec la taille du problème. Cependant, dans le pire des cas, la complexité peut être exponentielle, ce qui signifie que le nombre d'itérations peut croître de manière exponentielle avec la taille du problème.

Cependant, malgré son comportement en moyenne linéaire, le Simplexe peut rencontrer des cas où le nombre d'itérations devient prohibitif pour les problèmes de grande taille, ce qui le rend moins efficace que certaines autres approches pour les problèmes particulièrement difficiles.

2.3.2.4 Exemple d'Application de la Méthode du Simplexe

Considérons le problème (2.2) formulé par la nutritionniste. Avant d'appliquer la méthode du simplexe, nous devons d'abord le transformer en une forme canonique en introduisant les variables d'écart. Ainsi, nous obtenons :

$$\begin{cases} \max_x f(x) = c^t x \\ Ax = b \\ x \geq 0, \end{cases} \quad (2.12)$$

où

$$A = \begin{pmatrix} 2 & 3 & 1 & 0 & 0 \\ 1 & 3 & 0 & 1 & 0 \\ 4 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 38 \\ 32 \\ 44 \end{pmatrix}, \quad c = \begin{pmatrix} 20 \\ 25 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix}.$$



Phase 1 : (Test du premier point extrême)

L'ensemble des indices de base et la matrice de base associée sont :

$$J_N = \{1, 2\}, \quad J_B = \{3, 4, 5\}, \quad A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

La solution réalisable initiale x^0 est :

$$x_B = (A_B^{-1}b) = b, \quad x^0 = \begin{pmatrix} 0 \\ 0 \\ 38 \\ 32 \\ 44 \end{pmatrix}.$$

Les composantes non basiques du taux d'accroissement de f sont : $t_1^0 = 20 \geq 0$, $t_2^0 = 25 \geq 0$. Alors, x^0 n'est pas une solution optimale.

Phase 2 : (Début des itérations du simplexe)**Itération 1**

◇ La 1^{ère} direction de base admissible est :

$$d_N = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad d_B = -A_B^{-1}a_1 = \begin{pmatrix} -2 \\ -1 \\ -4 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ 0 \\ -2 \\ -1 \\ -4 \end{pmatrix}.$$

◇ Le taux d'accroissement maximale θ_0 est : $\theta_0 = \min \{14, 32, 11\} = 11$, $j_0 = 5$.

◇ La nouvelle solution réalisable x^1 est :

$$x^1 = x^0 + \theta_0 d = \begin{pmatrix} 0 \\ 0 \\ 38 \\ 32 \\ 44 \end{pmatrix} + 11 \begin{pmatrix} 1 \\ 0 \\ -2 \\ -1 \\ -4 \end{pmatrix} = \begin{pmatrix} 11 \\ 0 \\ 16 \\ 21 \\ 0 \end{pmatrix}.$$

◇ Les composantes non basiques du taux d'accroissement de f sont : $t_1^1 = 10 \geq 0$, $t_2^1 = -5 \leq 0$, $i_0 = 1$. Alors, x_1 n'est pas une solution optimale.



◇ Le nouvel ensemble des indices de base et la matrice de base associée sont :

$$\begin{aligned} J_N &= J_N \setminus \{i_0\} \cup \{j_0\} = \{2, 5\} \\ J_B &= J_B \setminus \{j_0\} \cup \{i_0\} = \{1, 3, 4\} \end{aligned}, \quad A_B = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix}.$$

Itération 2

◇ La 2^{ème} direction de base admissible est :

$$d_N = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad d_B = \begin{pmatrix} -\frac{3}{4} \\ -\frac{3}{2} \\ -\frac{9}{4} \end{pmatrix}, \quad d = \begin{pmatrix} -\frac{3}{4} \\ 1 \\ -\frac{3}{2} \\ -\frac{9}{4} \\ 0 \end{pmatrix}.$$

◇ Le taux d'accroissement maximal θ_1 est : $\theta_1 = \min \left\{ \frac{44}{3}, 4, \frac{42}{5} \right\} = 4, j_1 = 3.$

◇ La nouvelle solution réalisable x^2 est :

$$x^2 = x^1 + \theta_1 d = \begin{pmatrix} 11 \\ 0 \\ 6 \\ 21 \\ 0 \end{pmatrix} + 4 \begin{pmatrix} -\frac{3}{4} \\ 1 \\ -\frac{3}{2} \\ -\frac{9}{4} \\ 0 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 10 \\ 12 \\ 0 \end{pmatrix}.$$

◇ Les composantes non basiques du taux d'accroissement de f sont : $t_1^2 = -\frac{20}{3} \leq 0, t_2^2 = -\frac{5}{3} \leq 0.$ D'où, x^2 est une solution optimale.

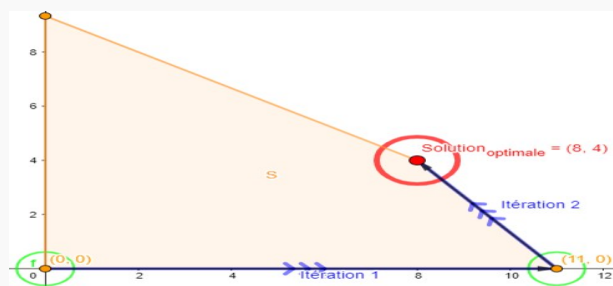


Figure 2.2 – Résolution du problème (2.12) avec la méthode du simplexe

Répartition idéale des aliments pour une solution équilibrée (2.2)

La nutritionniste devra servir 8 onces d'aliments **A** et 4 onces d'aliments **B**.

2.3.3 La Méthode Adaptée



Dans cette section, nous aurons le privilège de découvrir une méthode singulière et efficace, spécialement conçue pour résoudre avec brio les problèmes de programmation linéaire à variables bornées. Cette approche numérique, fruit des travaux éminents de R. Gabasov, F.M. Kirillova et O. I. Kostyukova à la fin des années 1980 [Gab95 ; Gab78], s'est révélée être une véritable pépite de polyvalence. Au fil des ans, elle a transcendé les frontières de la discipline pour aborder avec succès un éventail diversifié de domaines, notamment la programmation linéaire, la programmation quadratique [Eav71 ; Sya17], le contrôle optimal [Gab78 ; Gab80], et même la Programmation Linéaire Multi-Objectifs (PLMO) [Del18b ; Rad11b].

Ainsi, au fil de cette section, nous explorerons les contours délicats de cette méthode éminente, en présentant de manière synthétique l'algorithme sous-jacent. Toutefois, pour les esprits curieux en quête de fondements théoriques plus approfondis, nous ne saurions trop recommander la lecture attentive de la référence bibliographique de R. Gabasov [Gab95].

Sous la lumière éclairante de cette méthode innovante, nous examinerons désormais un problème de programmation linéaire à variables bornées, dont la formulation se présente sous la forme canonique suivante :

$$\begin{cases} \max_x f(x) = c^t x \\ Ax = b \\ l \leq x \leq u. \end{cases} \quad (2.13)$$

Les vecteurs c et b sont de dimensions n et m respectivement. La matrice A , de taille $m \times n$, présente une particularité intéressante : son rang est égal à m et satisfait $m < n$. Quant aux vecteurs l et u , appartenant à l'espace réel \mathbb{R}^n , avec la condition que $\|l\| < \infty$ et $\|u\| < \infty$.

Définition 2.3.7.

1. Un vecteur x qui satisfait les contraintes du problème (2.13) est désigné sous le nom de **solution réalisable** ou simplement **plan**. L'ensemble de toutes les solutions réalisables est appelé **région réalisable**, défini comme suit :

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

2. Une solution réalisable x^0 est qualifiée d'**optimale** si elle satisfait l'équation suivante :

$$f(x^0) = \max_{x \in S} f(x).$$



3. Une solution réalisable x_ϵ est dite ϵ -optimale ou sous-optimale si elle vérifie l'inégalité suivante : $f(x^0) - f(x_\epsilon) = c^t x^0 - c^t x_\epsilon \leq \epsilon$, où x^0 est une solution optimale pour le problème (2.13) et ϵ est un nombre strictement positif préalablement choisi.

4. Soit J_B un ensemble d'indices, tel que $|J_B| = m$, alors J_B est qualifié de support si la matrice A_B est inversible.

5. Le couple $\{x, J_B\}$, composé d'un plan x et d'un support J_B , est appelé plan support. Un plan support est considéré comme non-dégénéré si les conditions suivantes sont satisfaites :

$$l_j < x_j < u_j, \quad j \in J_B.$$

2.3.3.1 Estimation de Sous-optimalité

Considérons $\{x, J_B\}$ comme un plan support du problème (2.13) et x^0 comme un plan quelconque tel que $x^0 = x + \Delta x$. Alors, l'accroissement de la fonction objectif peut être exprimé comme suit :

$$\Delta f(x) = f(x^0) - f(x) = c^t x^0 - c^t x = c^t (x^0 - x) = c^t \Delta x. \quad (2.14)$$

De plus, on peut observer que :

$$A \Delta x = A (x^0 - x) = A x^0 - A x = b - b = 0.$$

Ce qui signifie que :

$$A_B \Delta x_B + A_N \Delta x_N = 0.$$

En conséquence, on peut exprimer Δx_B en fonction de Δx_N :

$$\Delta x_B = -A_B^{-1} A_N \Delta x_N. \quad (2.15)$$

En remplaçant Δx_B dans l'équation (2.14), on obtient :

$$\begin{aligned} \Delta f(x) = c^t \Delta x &= c_B^t \Delta x_B + c_N^t \Delta x_N \\ &= -c_B^t A_B^{-1} A_N \Delta x_N + c_N^t \Delta x_N. \end{aligned}$$

Ce qui nous permet de réécrire l'accroissement de la fonction objectif sous la forme suivante :

$$\Delta f(x) = -\left(c_B^t A_B^{-1} A_N - c_N^t\right) \Delta x_N. \quad (2.16)$$

Pour simplifier davantage notre analyse, nous introduisons le vecteur des potentiels K et le vecteur des estimations E définis comme suit :

$$K_B^t = c_B^t A_B^{-1}, \quad E^t = K^t A - c^t = (E_N^t, E_B^t),$$

où

$$\begin{aligned} E_N^t &= c_B^t A_B^{-1} A_N - c_N^t, \\ E_B^t &= c_B^t A_B^{-1} A_B - c_B^t = 0. \end{aligned}$$

Finalement, en utilisant ces notations, la formule (2.16) peut être réécrite comme suit :

$$\Delta f(x) = -E^t \Delta x_N = \sum_{j \in J_N} E_j (x_j - x_j^0). \quad (2.17)$$

Cette dernière expression nous permet de quantifier l'accroissement de la fonction objectif en fonction des variations des composantes du plan support non-basiques, fournissant ainsi un outil essentiel pour notre méthode d'optimisation linéaire à variables bornées.

Théorème 2.3.8 (Critère d'optimalité [Gab95]). Soit $\{x, J_B\}$ un plan support du problème (2.13). Alors, les relations suivantes sont suffisantes pour garantir l'optimalité du plan support $\{x, J_B\}$:

$$\begin{cases} E_j \geq 0 & \text{si } x_j = l_j \\ E_j \leq 0 & \text{si } x_j = u_j \\ E_j = 0 & \text{si } l_j < x_j < u_j \end{cases}, \quad j \in J_N. \quad (2.18)$$

De plus, ces relations sont également nécessaires lorsque le plan support $\{x, J_B\}$ est non-dégénéré. En d'autres termes, si le plan support est non-dégénéré, alors ces relations sont obligatoires pour assurer l'optimalité.

En utilisant le théorème (2.3.8), nous avons l'inégalité suivante : $x_j - u_j \leq x_j - x_j^0 \leq x_j - l_j$, $j \in J_N$. Cela nous permet d'établir les majorations suivantes :

$$\begin{aligned} E_j (x_j - x_j^0) &\leq E_j (x_j - l_j), \quad \text{si } E_j > 0 \\ E_j (x_j - x_j^0) &\leq E_j (x_j - u_j), \quad \text{si } E_j < 0. \end{aligned}$$

Par conséquent, nous obtenons la majoration suivante :

$$\begin{aligned} f(x^0) - f(x) &= \sum_{j \in J_N} E_j (x_j - x_j^0) \\ &= \sum_{E_j > 0, j \in J_N} E_j (x_j - x_j^0) + \sum_{E_j < 0, j \in J_N} E_j (x_j - x_j^0) \\ &\leq \sum_{E_j > 0, j \in J_N} E_j (x_j - l_j) + \sum_{E_j < 0, j \in J_N} E_j (x_j - u_j). \end{aligned}$$



Cette inégalité nous permet d'évaluer l'écart entre la valeur de la fonction objectif au plan optimal x^0 et tout autre plan de support x , en termes de contributions positives ou négatives des variables non-basiques.

Définition 2.3.9. Nous définissons la quantité suivante comme l'estimation de sous-optimalité :

$$\beta(x, J_B) = \sum_{E_j > 0, j \in J_N} E_j (x_j - l_j) + \sum_{E_j < 0, j \in J_N} E_j (x_j - u_j). \quad (2.19)$$

Théorème 2.3.10 (Condition suffisante d'optimalité). Soient $\{x, J_B\}$ un plan support du problème (2.13) et ϵ un nombre strictement positif. Alors, le plan x est une solution ϵ -optimale si et seulement si $\beta(x, J_B) < \epsilon$.

2.3.3.2 Le Fondement de la Méthode Adaptée

La méthode adaptée de Gabasov est une approche itérative et efficace pour résoudre des problèmes d'optimisation linéaire à variables bornées, qui s'appuie sur la métrique du simplexe. Son objectif est de trouver la meilleure solution possible pour un problème donné en respectant les contraintes et les bornes sur les variables.

Le principe de la méthode adaptée de Gabasov est le suivant : nous commençons avec un plan support initial $\{x, J_B\}$, qui est une solution candidate satisfaisant les contraintes du problème. Ensuite, nous itérons pour atteindre un nouveau plan support $\{\bar{x}, \bar{J}_B\}$ tout en améliorant la valeur de la fonction objectif, c'est-à-dire, en cherchant à satisfaire l'inégalité $f(\bar{x}) \geq f(x)$.

Contrairement à certaines méthodes d'optimisation, la méthode adaptée de Gabasov peut simultanément faire varier plusieurs indices j lors de chaque itération, en fonction des valeurs des bornes l_j, u_j et des variables x_j . La caractéristique clé de la méthode adaptée est que la métrique utilisée pour déterminer la direction d'amélioration dépend du plan actuel, ce qui peut entraîner des changements dans plusieurs variables en même temps. La métrique adaptée est définie de la manière suivante : $l_j - x_j \leq d_j \leq u_j - x_j$. Cette métrique permet de déterminer les variations possibles pour chaque variable x_j tout en respectant les bornes l_j et u_j .



L'algorithme 2 ne contient pas explicitement de métrique. Cependant, la notion de métrique est implémentée indirectement dans les étapes 7, 8 et 9, où on détermine la direction d'amélioration d et le pas d'optimisation θ^0 .



Dans le chapitre 5, vous trouverez gracieusement un exemple éloquent illustrant la méthode adaptée de Gabasov.

Algorithme 2 Algorithme de la méthode adaptée mono-critère

Entrée : Plan support initial $\{x, J_B\}$, nombre ϵ .

Sortie : Plan support optimal $\{x, J_B\}$ ou ϵ -optimal.

Étape 1 : Calculer $K_B^t = c_B^t A_B^{-1}$, $E_j^t = K^t a_j - c_j^t$, $j \in J_N$.

Étape 2 : Calculer $\beta(x, J_B)$ en utilisant la formule (2.19).

Étape 3 : Si $\beta(x, J_B) = 0$, arrêtez avec le plan support optimal $\{x, J_B\}$.

Étape 4 : Si $\beta(x, J_B) \leq \epsilon$, arrêtez avec le plan support $\{x, J_B\}$ ϵ -optimal.

Étape 5 : Déterminer l'ensemble d'indices non-optimal :

$$J_{NNO} = \{j \in J_N : [E_j < 0, x_j < u_j] \text{ ou } [E_j > 0, x_j > l_j]\}.$$

Étape 6 : Choisir un indice j_0 de J_{NNO} tel que $|E_{j_0}| = \max_{j \in J_{NNO}} |E_{j_0}|$.

Étape 7 : Calculer la direction d'amélioration admissible d en utilisant les relations :

$$\begin{cases} d_{j_0} = -\text{sign}(E_{j_0}) \\ d_j = 0 & \text{si } j \neq j_0, j \in J_N \\ d_B = -A_B^{-1} A_{j_0} d_{j_0} \end{cases} \quad (2.20)$$

Étape 8 : Calculer $\theta_{j_1} = \min_{j \in J_B} \theta_j$, où θ_j est déterminé par la formule :

$$\theta_j = \begin{cases} \frac{u_j - x_j}{d_j}, & \text{si } d_j > 0 \\ \frac{l_j - x_j}{d_j}, & \text{si } d_j < 0 \\ \infty, & \text{si } d_j = 0. \end{cases} \quad (2.21)$$

Étape 9 : Calculer θ_{j_0} en utilisant la formule :

$$\theta_{j_0} = \begin{cases} x_{j_0} - l_{j_0} & \text{si } E_{j_0} > 0 \\ u_{j_0} - x_{j_0} & \text{si } E_{j_0} < 0. \end{cases} \quad (2.22)$$

Étape 10 : Calculer $\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}\}$.

Étape 11 : Calculer $\bar{x} = x + \theta^0 d$.

Étape 12 : Calculer $\beta(\bar{x}, J_B) = \beta(x, J_B) - \theta^0 |E_{j_0}|$.

Étape 13 : Si $\beta(\bar{x}, J_B) = 0$, arrêtez avec le plan support optimal $\{x, J_B\}$.

Étape 14 : Si $\beta(\bar{x}, J_B) \leq \epsilon$, arrêtez avec le plan support $\{x, J_B\}$ ϵ -optimal.

Étape 15 : Si $\theta^0 = \theta_{j_0}$, alors on pose $\bar{J}_B = J_B$.

Étape 16 : Si $\theta^0 = \theta_{j_1}$, alors on pose $\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}$.

Étape 17 : Poser $x = \bar{x}$ et $J_B = \bar{J}_B$, et aller à l'étape 1.



CHAPITRE 3

UNE APPROCHE GÉOMÉTRIQUE POUR L'ANALYSE DE SENSIBILITÉ

Objectifs

Dans ce chapitre, nous présentons une approche géométrique pour étudier la stabilité de la solution optimale d'un problème de programmation linéaire mono-objectif en deux dimensions de variables, tout en permettant la variation simultanée des deux coefficients de la fonction objectif sans altérer le domaine réalisable. Cette approche nous permet d'effectuer une étude de stabilité en prenant en compte deux paramètres de manière simultanée [Kac21a; Kac22a; Kac21b; War90].

Nous exploitons la norme euclidienne et la linéarité des fonctions de contrainte pour formuler un problème équivalent au problème d'analyse de sensibilité, qui se résout facilement par un calcul simple. Après avoir fixé une solution optimale d'un programme linéaire donné, qui représente un point extrême de la région réalisable, nous déterminons toutes les formes linéaires qui atteignent leur valeur optimale en ce point. Cette étude géométrique consiste à identifier les hyperplans dans \mathbb{R}^3 , où la variation des coefficients de la fonction objectif correspond à la rotation du plan. Ainsi, notre approche nous permet de considérer la stabilité de manière plus complète en présence de deux paramètres.

Sommaire

3.1	Introduction	63
3.2	Préliminaires	64
3.3	Formulation du Problème	66
3.4	Géométrie des Programmes Linéaires : Une Approche Visuelle	67
3.5	Analyse Géométrique de Sensibilité dans les Programmes Linéaires	70
3.6	Exemple d'Application	72



Résumé

Dans cette contribution ⁱ, nous présentons une approche géométrique innovante pour l'analyse de sensibilité en programmation linéaire. Cette approche offre une solution computationnellement pratique pour examiner le comportement de la solution optimale du PPL ⁱⁱ face à des variations dans les données du problème. Dans un premier temps, nous délimitons le domaine réalisable en établissant les contraintes linéaires. Ensuite, nous formulons géométriquement le PPL. Nous introduisons par la suite une nouvelle formulation géométrique équivalente du problème d'analyse de sensibilité, en utilisant les concepts de géométrie affine. Pour ce faire, nous exprimons le vecteur des coefficients de la fonction objectif en coordonnées polaires, nous permettant ainsi d'identifier les angles auxquels la solution demeure inchangée. Enfin, nous développons cette approche en détail et l'illustrons à travers un exemple numérique.

i. Cet article a été publié en 2022 dans le journal "Mathematica Applicanda" sous le titre "A new geometric approach for sensitivity analysis in linear programming", référencé sous [Kac22a].

ii. Problème de Programmation Linéaire

Mots clés

Programmation linéaire, analyse de sensibilité, approche géométrique.

3.1 Introduction

L'analyse de sensibilité occupe une place essentielle dans la programmation linéaire, car elle permet d'évaluer de manière systématique la réaction d'une solution optimale face aux changements des données d'entrée [Sha16; Ste79; Had91]. Cette étape cruciale de l'évaluation de la solution optimale est d'une grande valeur et doit être systématiquement effectuée sur toutes les données d'entrée critiques.

L'analyse de sensibilité en programmation linéaire revêt une importance capitale en raison de ses multiples applications. Tout d'abord, elle permet de déterminer l'exactitude requise pour les données d'entrée du modèle, évitant ainsi la répétition de la procédure de résolution à maintes reprises. Ensuite, elle joue un rôle crucial dans l'établissement d'intervalles de contrôle pour les changements des paramètres d'entrée et des constantes, garantissant que la solution optimale actuelle reste pratiquement optimale. Cela est particulièrement utile pour des cas concrets tels que la maximisation des bénéfices d'une entreprise, où les fluctuations des données de marché nécessitent des mises à jour sans remettre en question les problèmes initiaux. Enfin, l'analyse



de sensibilité permet d'évaluer la valeur marginale des ressources rares à des fins de planification, en établissant l'impact des changements dans les contraintes sur les variables de décision. Ainsi, cette approche offre des perspectives essentielles pour une meilleure compréhension et optimisation des problèmes linéaires.

3.2 Préliminaires

Définition 3.2.1 (Isométries de \mathbb{R}^2). Nous définissons une isométrie de \mathbb{R}^2 comme étant une application linéaire $u \in \mathcal{L}(\mathbb{R}^2, \mathbb{R}^2)$ qui satisfait l'une des propriétés suivantes :

1. u préserve la norme : pour tout $x \in \mathbb{R}^2$, $\|u(x)\| = \|x\|$.
2. u préserve le produit scalaire : pour tout $x, y \in \mathbb{R}^2$, $\langle u(x), u(y) \rangle = \langle x, y \rangle$.
3. u transforme une base orthonormale en une base orthonormaleⁱⁱⁱ.

Cette application u est également appelée automorphisme orthogonal.^{iv}

iii. C'est un ensemble de vecteurs linéairement indépendants dont la norme de chaque vecteur est 1, et les vecteurs sont mutuellement orthogonaux.

iv. L'espace vectoriel de toutes les isométries (automorphismes orthogonaux) de \mathbb{R}^2 est noté $\mathcal{O}(\mathbb{R}^2)$.

Propriété 3.2.2. Soient u_1 et u_2 des isométries de $\mathcal{O}(\mathbb{R}^2)$. Alors :

1. u_1 et u_2 sont des applications bijectives.
2. $u_1 \circ u_2$ et u_1^{-1} sont des isométries.
3. La translation et la rotation de \mathbb{R}^2 sont des isométries.

En résumé, les isométries de $\mathcal{O}(\mathbb{R}^2)$ sont des applications bijectives qui préservent la structure géométrique de l'espace vectoriel \mathbb{R}^2 .

Exemple 3.2.3.

1. Une translation de \mathbb{R}^2 dans la direction du vecteur v est une application bijective notée T_v , qui associe à chaque élément x de \mathbb{R}^2 un autre élément $T_v(x) = x + v$ de \mathbb{R}^2 .
2. Soit θ un angle compris entre 0 et 2π . Une rotation de \mathbb{R}^2 d'angle θ est une application bijective R_θ définie par :

$$R_\theta : \quad \mathbb{R}^2 \quad \longrightarrow \quad \mathbb{R}^2$$

$$(x_1, x_2) \quad \longrightarrow \quad R_\theta(x_1, x_2) = \text{Rot}(\theta)(x_1, x_2)^t,$$



où $\text{Rot}(\theta)$ est la matrice associée à la rotation définie par :

$$\text{Rot}(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$



Il est important de noter que pour tout angle θ , la matrice de rotation $\text{Rot}(\theta)$ satisfait à la propriété suivante :

$$\text{Rot}(\theta)^t \text{Rot}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = Id_2.$$

Définition 3.2.4. Dans \mathbb{R}^n , considérons deux sous-espaces vectoriels F_1 et F_2 qui sont supplémentaires, c'est-à-dire que la somme de leurs dimensions est égale à la dimension de l'espace ambiant, soit $\dim(F_1) + \dim(F_2) = \dim(\mathbb{R}^n)$, et leur intersection est réduite à l'origine, c'est-à-dire $F_1 \cap F_2 = \{0\}$.

Dans ce contexte, nous définissons la projection sur F_1 parallèlement à F_2 , notée Pr_{F_1} , comme l'application qui associe à chaque élément x de \mathbb{R}^n l'élément unique $Pr_{F_1}(x)$ de F_1 tel que $x = Pr_{F_1}(x) + z$, où z est un élément de F_2 . Autrement dit, la projection $Pr_{F_1}(x)$ est l'unique point du sous espace F_1 qui est le plus proche de x parmi tous les points accessibles en ajoutant un élément de F_2 . Cette propriété est illustrée dans la figure (3.1).



Une telle application Pr_{F_1} est communément appelée un projecteur, car elle permet de projeter tout point de \mathbb{R}^n sur le sous-espace F_1 tout en conservant un certain parallélisme avec le sous-espace F_2 .

Propriété 3.2.5 (Équation reliant les segments et l'angle). Considérons deux droites (d) et (\acute{d}) dans \mathbb{R}^n qui forment un angle $\acute{\theta}$. Soient M et N deux points appartenant à la droite (d) . Les projections orthogonales respectives de ces points sont notées $\acute{M} = Pr_{(\acute{d})}(M)$ et $\acute{N} = Pr_{(\acute{d})}(N)$, comme illustré dans la figure (3.1).

Dans ce contexte, nous pouvons établir la relation suivante :

$$\acute{M}\acute{N} = MN \cos(\acute{\theta}).$$



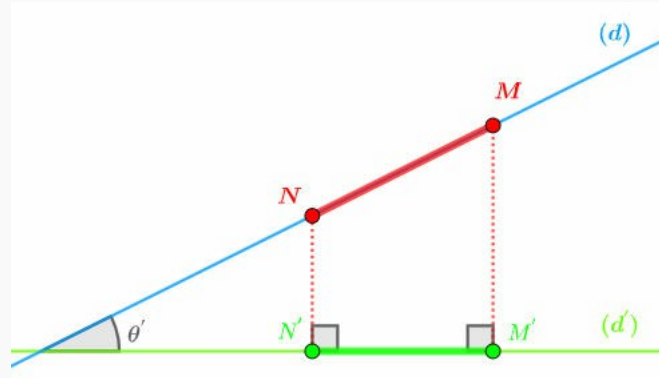


Figure 3.1 – Projection d'un segment sur une droite.

3.3 Formulation du Problème

Considérons le problème de programmation linéaire suivant sous forme standard :

$$\begin{cases} \max_x f(x) = c_0^t x \\ Ax \leq b \\ x \geq 0, \end{cases} \quad (3.1)$$

avec les matrices et vecteurs donnés par :

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad c_0 = \begin{pmatrix} c_1^0 \\ c_2^0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$



Nous supposons que c_1^0 et c_2^0 sont des nombres positifs ou nuls. Cette hypothèse simplifie la présentation, comme expliqué dans la remarque (3.4.3).

Notations 3.3.1. On fixe les notations suivantes :

- ◇ La région réalisable du problème (3.1) : $S := \{x \in \mathbb{R}^2 : Ax \leq b, x \geq 0\}$.
- ◇ L'espace vectoriel : $\mathcal{L}(\mathbb{R}^2) := \{g : \mathbb{R}^2 \rightarrow \mathbb{R} : g(x_1, x_2) = c_1 x_1 + c_2 x_2, c_1, c_2 \in \mathbb{R}\}$.

Problème 3.3.2. Soit $x^0 = (x_1^0, x_2^0)^t$ une solution optimale du problème (3.1). Un problème d'analyse de sensibilité consiste à trouver toutes les formes linéaires g différentes de f , vérifiant :

$$\arg \max_{x \in S} f(x) = \arg \max_{x \in S} g(x), \quad g \in \mathcal{L}(\mathbb{R}^2). \quad (3.2)$$

3.4 Géométrie des Programmes Linéaires : Une Approche Visuelle

Dans cette section, nous allons établir un problème équivalent au problème de programmation linéaire (3.1). En considérant \mathbb{R}^3 , nous remarquons que la région réalisable S est un polytope entièrement contenu dans l'hyperplan vectoriel d'équation $z = 0$. De plus, le graphe de f correspond à un hyperplan vectoriel (sous-espace vectoriel de dimension deux).

Pour transformer l'expression de la forme linéaire f , nous effectuons la modification suivante :

$$\begin{aligned} f(x_1, x_2) = c_1^0 x_1 + c_2^0 x_2 &\Leftrightarrow c_1^0 x_1 + c_2^0 x_2 - f(x_1, x_2) = 0 \\ &\Leftrightarrow \langle (c_1^0, c_2^0, -1), (x_1, x_2, f(x_1, x_2)) \rangle = 0 \\ &\Leftrightarrow \langle n_f, (x_1, x_2, z) \rangle = 0, \end{aligned}$$

où $x_1, x_2 \in \mathbb{R}$, $z = f(x_1, x_2)$ et $n_f = (c_1^0, c_2^0, -1)$. Ainsi, le graphe de la forme linéaire f , défini par :

$$\begin{aligned} Gf(f) &= \{(x_1, x_2, z) \in \mathbb{R}^3 : z = f(x_1, x_2)\} \\ &= \{(x_1, x_2, z) \in \mathbb{R}^3 : \langle n_f, (x_1, x_2, z) \rangle = 0\} \end{aligned}$$

est orthogonal à la droite vectorielle engendrée par le vecteur n_f . En conséquence, tout vecteur de \mathbb{R}^3 peut s'exprimer comme la somme d'un vecteur de $Gf(f)$ et d'un autre vecteur appartenant à $\text{vect}(n_f)$.

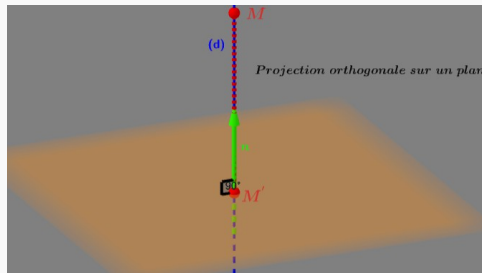


Figure 3.2 – Projection orthogonale sur les sous-espaces vectoriels de \mathbb{R}^3

Maintenant, nous nous plaçons dans \mathbb{R}^2 , c'est-à-dire le plan $z = 0$, et nous considérons la droite d_0 définie par l'intersection de $Gf(f)$ avec le plan $z = 0$:

$$(d_0) : c_1^0 x_1 + c_2^0 x_2 = 0. \quad (3.3)$$

La droite d_0 est un sous-espace vectoriel de \mathbb{R}^2 de dimension 1, dirigée par le vecteur :

$$v_0 = (-c_2^0, c_1^0).$$

v. L'ensemble des valeurs de x pour lesquelles l'application f atteint son maximum.



Définition 3.4.1.

1. Le graphe de la droite d_0 est défini par l'ensemble :

$$D_0 = \{(x_1, x_2) \in \mathbb{R}^2 : c_1^0 x_1 + c_2^0 x_2 = 0\}.$$

2. Le demi-espace supérieur de \mathbb{R}^2 privé de d_0 est défini par :

$$D_0^+ = \{(x_1, x_2) \in \mathbb{R}^2 : c_1^0 x_1 + c_2^0 x_2 > 0\}.$$

3. Le demi-espace inférieur de \mathbb{R}^2 privé de d_0 est défini par :

$$D_0^- = \{(x_1, x_2) \in \mathbb{R}^2 : c_1^0 x_1 + c_2^0 x_2 < 0\}.$$

Lemme 3.4.2. Pour tout $x^1, x^2 \in D_0^+$ (respectivement, $y^1, y^2 \in D_0^-$), on a la propriété suivante :

$$\frac{f(x^1)}{\|x^1 - Pr_{(d_0)}(x^1)\|} = \frac{f(x^2)}{\|x^2 - Pr_{(d_0)}(x^2)\|} \tag{3.4}$$

et

$$\frac{f(y^1)}{\|y^1 - Pr_{(d_0)}(y^1)\|} = \frac{f(y^2)}{\|y^2 - Pr_{(d_0)}(y^2)\|}. \tag{3.5}$$

Démonstration. L'intersection de $Gf(f)$ avec le plan d'équation $z = 0$ forme un angle constant $0 \leq \lambda \leq \pi$, comme illustré dans la figure (3.3). Alors, pour tout $x^1, x^2 \in D_0^+$ et $y^1, y^2 \in D_0^-$, on a les relations suivantes :

$$\begin{cases} \tan(\lambda) &= \frac{f(x^1)}{\|x^1 - Pr_{(d_0)}(x^1)\|} = \frac{f(x^2)}{\|x^2 - Pr_{(d_0)}(x^2)\|}, & x^1, x^2 \in D_0^+, \\ \tan(\pi - \lambda) &= \frac{f(y^1)}{\|y^1 - Pr_{(d_0)}(y^1)\|} = \frac{f(y^2)}{\|y^2 - Pr_{(d_0)}(y^2)\|}, & y^1, y^2 \in D_0^-. \end{cases}$$

■

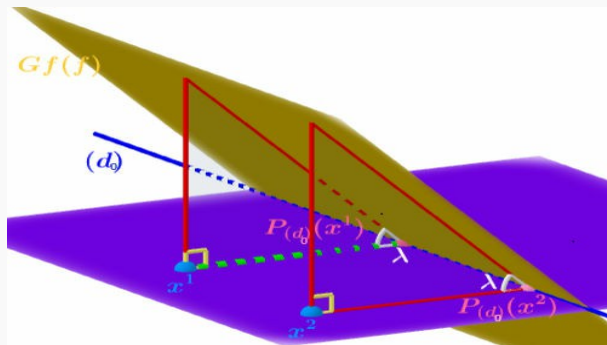


Figure 3.3 – Illustration graphique du lemme (3.4.2)



Remarques 3.4.3 (Simplification du problème en rendant les coefficients de c_0 positifs ou nuls.). Si c_1^0 et c_2^0 sont des nombres réels quelconques, on peut se ramener aux hypothèses du problème (3.1) en effectuant les changements suivants :

1. On applique la rotation R_θ sur la région réalisable S et le vecteur c_0 simultanément, et on obtient le problème équivalent :

$$\max_{x \in S} (R_\theta(c_0))^t R_\theta(x), \quad (3.6)$$

où $R_\theta(c_0)$ est le vecteur obtenu en appliquant la rotation à c_0 . Ce problème est équivalent au problème initial car la rotation conserve la valeur de la fonction objectif. Ainsi, toute solution optimale x^0 du problème (3.1) est également une solution optimale du problème (3.6).

2. Ensuite, on pose $\text{Rot}(\theta^0) c_0 = (|c_1^0|, |c_2^0|) \geq 0$, où θ^0 est la solution du système suivant :

$$\begin{cases} \cos(\theta)c_1^0 - \sin(\theta)c_2^0 = |c_1^0|, \\ \sin(\theta)c_1^0 + \cos(\theta)c_2^0 = |c_2^0|. \end{cases}$$

Cette étape permet de garantir que c_1^0 et c_2^0 deviennent tous deux positifs, simplifiant ainsi la suite de la présentation.

3. Enfin, on applique une translation T_{v_0} de S dans la direction de $\text{Rot}(\theta^0) (c_1^0, c_2^0)^t$ car la fonction objectif est constante dans cette direction. En d'autres termes, pour tout $y \in T_{v_0}(S)$, on a :

$$f(y) = f(x + v_0) = f(x) + f(v_0) \leq f(x^0) + f(v_0) \leq f(x^0 + v_0) \leq f(y^0),$$

où $y^0 = x^0 + v_0$ et $v_0 = \alpha \text{Rot}(\theta^0) (c_1^0, c_2^0)^t$, avec α étant un nombre strictement positif suffisamment grand.

Proposition 3.4.4. Soit c_1^0 et c_2^0 des nombres positifs. Alors, le problème (3.1) est équivalent à :

$$\max_{x \in S} \|x - \text{Pr}_{(d_0)}(x)\|. \quad (3.7)$$



La formulation 3.7 utilise une distance géométrique pour simplifier le problème.



Démonstration. La preuve repose sur la comparaison des solutions optimales entre les problèmes (3.1) et (3.7). Soit x^0 une solution optimale du problème (3.1). En utilisant le lemme (3.4.2) et l'inégalité $f(x^0) \geq f(x), \forall x \in S$, nous montrons que pour tout $x \in S$,

$$\|x^0 - Pr_{(d_0)}(x^0)\| \geq \|x - Pr_{(d_0)}(x)\|,$$

ce qui montre que x^0 est aussi une solution optimale du problème (3.7).

Inversement, si x^0 est une solution optimale du problème (3.7), alors en utilisant l'inégalité

$$\|x^0 - Pr_{(d_0)}(x^0)\| \geq \|x - Pr_{(d_0)}(x)\|,$$

on passe à l'inverse et obtenons

$$\frac{1}{\|x - Pr_{(d_0)}(x)\|} \geq \frac{1}{\|x^0 - Pr_{(d_0)}(x^0)\|}, \quad \forall x \in S.$$

En multipliant par $f(x)$ et en appliquant le lemme (3.4.2), on montre que x^0 est aussi une solution optimale du problème (3.1). ■

3.5 Analyse Géométrique de Sensibilité dans les Programmes Linéaires

L'analyse géométrique que nous présentons ici consiste à écrire un programme linéaire en coordonnées polaires, puis à déterminer une relation qui relie les coefficients de la fonction objectif f . Nous savons déjà qu'à toute forme linéaire g dans $\mathcal{L}(\mathbb{R}^2)$, on peut associer un unique vecteur (gradient de g) dans \mathbb{R}^2 et vice versa. Un autre fait important est que nous pouvons obtenir tous les sommets de la région réalisable S en résolvant un nombre fini de systèmes linéaires à deux équations et deux inconnues. Cela nous permet de reformuler le problème (3.3.2) de la manière suivante :

Problème 3.5.1. Soient x^1, x^0, x^2 des sommets successifs de S , voir la figure (3.4), où x^0 est la solution du problème (3.7). Trouvons toutes les droites vectorielles (d) qui satisfont simultanément les deux inégalités suivantes :

$$\|x^0 - Pr_{(d)}(x^0)\| \geq \|x^1 - Pr_{(d)}(x^1)\| \tag{3.8}$$

et

$$\|x^0 - Pr_{(d)}(x^0)\| \geq \|x^2 - Pr_{(d)}(x^2)\|. \tag{3.9}$$

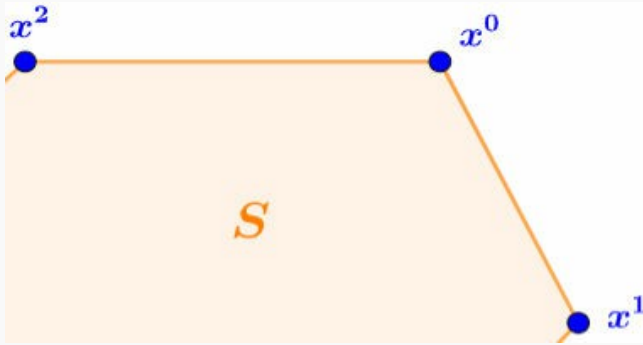


Figure 3.4 – Les points extrêmes voisins à la solution optimale x^0

Proposition 3.5.2. Le problème (3.3.2) et (3.5.1) sont équivalents.

Démonstration. La preuve découle naturellement de la proposition (3.4.4) et de la convexité de S . ■



Le problème (3.5.1) hérite de la solution triviale de l'autre problème (3.7) à partir duquel il est construit. Ainsi, la droite vectorielle (d_0) qui est associée à la solution optimale de (3.7), constitue également une solution pour le problème (3.5.1). Cette propriété découle directement de la construction du problème (3.5.1) et de la présence de la solution optimale déjà connue du problème (3.7).

Proposition 3.5.3. Soit $\theta_1, \theta_2 \in [0, \pi]$, $\phi \in [0, \frac{\pi}{2}[$ et $r, r_1, r_2 \geq 0$. Considérons les vecteurs suivants exprimés dans le système de coordonnées polaires :

$$x^{10} := x^0 - x^1 = r_1(\cos(\theta_1), \sin(\theta_1))$$

$$x^{20} := x^2 - x^0 = r_2(\cos(\theta_2), \sin(\theta_2))$$

$$c = (c_1, c_2) = r(\cos(\phi), \sin(\phi)).$$

Alors, les solutions du problème (3.5.1) sont les droites définies par l'équation suivante :

$$(d) : r \cos(\phi)x_1 + r \sin(\phi)x_2 = 0, \quad \theta_1 < \phi + \frac{\pi}{2} < \theta_2. \quad (3.10)$$



La proposition 3.5.3 offre une caractérisation explicite des solutions du problème (3.5.1), représentées par les droites vectorielles (d) qui satisfont aux inéquations (3.8) et (3.9), définies par la formule 3.10.

Démonstration. Supposons que x^0 est une solution du problème (3.7). En utilisant la



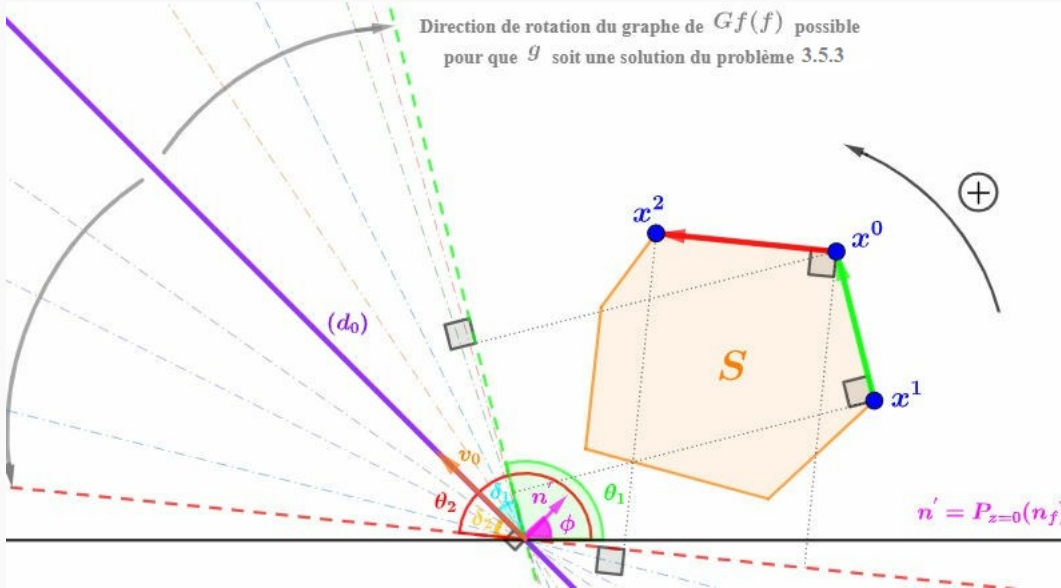


FIGURE 3.5 – Illustration graphique de la proposition (3.5.3).

propriété (3.2.5), pour tout $i = 1, 2$, nous avons :

$$\|Pr_{(d)}(x^i) - Pr_{(d)}(x^0)\| = \|x^i - x^0\| \cos(\delta_i),$$

où δ_i est l'angle formé par le vecteur x^{i0} avec v_0 . Comme le montre la figure (3.5), nous avons $|\delta_i| = |\phi + \frac{\pi}{2} - \theta_i| < \frac{\pi}{2}$. Ainsi, pour tout $i = 1, 2$, nous obtenons les inégalités suivantes :

$$\|Pr_{(d)}(x^i) - Pr_{(d)}(x^0)\| < \|x^i - x^0\|.$$

Puisque les quadrilatères $x^0x^iPr_{(d)}(x^i)Pr_{(d)}(x^0)$ sont irréguliers et que les segments $Pr_{(d)}(x^0)x^0$ et $Pr_{(d)}(x^i)x^i$ sont parallèles, pour tout $i = 1, 2$, nous obtenons finalement :

$$\|Pr_{(d)}(x^i) - x^i\| < \|Pr_{(d)}(x^0) - x^0\|.$$

■

3.6 Exemple d'Application

Considérons le problème de programmation linéaire suivant :

$$\begin{cases} \max_{x,y} f(x_1, x_2) = 2x_1 + 3x_2 \\ A(x_1, x_2)^t \leq b \\ x_1 \geq 0; x_2 \geq 0, \end{cases} \quad (3.11)$$

où

$$A = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} \\ \frac{2}{5} & \frac{1}{5} \\ 0 & \frac{4}{5} \end{pmatrix}, \quad b = \begin{pmatrix} 40 \\ 40 \\ 40 \end{pmatrix}, \quad c_0 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad x^0 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

On commence par résoudre le problème (3.11), ce qui nous donne une solution optimale $x^0 = (80, 40)$. Ensuite, on résout le problème suivant :

$$\arg \max_{x \in S} f(x) = \arg \max_{x \in S} g(x). \quad (3.12)$$

Pour ce faire, nous considérons les coins successifs du domaine réalisable, x^1, x^0, x^2 , tels que : $x^1 = (100, 0)$ et $x^2 = (60, 50)$.

En utilisant les coordonnées polaires, nous calculons les vecteurs x^{10} et x^{20} :

$$\begin{aligned} x^{10} &= (-20, 40) = 20\sqrt{5} \left(-\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}} \right), \\ x^{20} &= (-20, 10) = 10\sqrt{5} \left(-\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}} \right). \end{aligned}$$

Nous déterminons ensuite les angles θ_1 et θ_2 :

$$\theta_1 = \arccos \left(-\frac{1}{\sqrt{5}} \right) \approx 116.5650^\circ$$

et

$$\theta_2 = \arccos \left(-\frac{2}{\sqrt{5}} \right) \approx 153.4349^\circ.$$

Ainsi, pour tout $r > 0$ et $\phi \in]26.57^\circ, 63.43^\circ[$, la forme linéaire $g(\cdot) = r \langle (\cos(\phi), \sin(\phi)), \cdot \rangle$ est une solution du problème (3.12).

Enfin, pour l'analyse de sensibilité, nous exprimons les coefficients de la fonction objectif f en coordonnées polaires :

$$c_0 = r_f (\cos(\phi_f), \sin(\phi_f)),$$

où $\phi_f = \arccos \left(\frac{2}{\sqrt{13}} \right) \approx 56.31^\circ$ et $r_f = \sqrt{13}$. En considérant une solution g qui rend x^0 optimal, nous évaluons sa valeur en x^0 , comme suit :

$$\begin{aligned} g(x^0) = g(80, 40) &\approx 40\sqrt{5}r \langle (\cos(\phi), \sin(\phi)), (\cos(26.5650^\circ), \sin(26.5650^\circ)) \rangle \\ &\approx 40\sqrt{5}r \cos(\phi - 26.5650^\circ) \\ &\approx 40\sqrt{5}r \cos(\phi_f + \nu - 26.5650^\circ) \\ &\approx 40\sqrt{5}r \cos(56.31^\circ + \nu - 26.5650^\circ) \\ &\approx 40\sqrt{5}r \cos(\nu + 29.745^\circ), \end{aligned}$$



où $\nu \in]-3.18^\circ, 33.87^\circ[$ correspond à l'angle de rotation entre $\text{grad}(f)$ et $\text{grad}(g)$. Si $\nu \in]-3.18^\circ, 0^\circ[$, la valeur optimale de f augmente, c'est-à-dire que la valeur optimale de g en x^0 est supérieure à la valeur optimale de f en x^0 . Si $\nu \in]0^\circ, 33.87^\circ[$, la valeur optimale de f diminue.

3.7 Conclusion

La formulation du problème (3.5.1) pour l'analyse de sensibilité nous a offert une approche élégante pour étudier la stabilité d'une solution optimale d'un problème de programmation linéaire en deux dimensions. Cette méthode simplifie la recherche en réduisant celle-ci à un seul intervalle, sans nécessiter l'utilisation d'une machine. De plus, elle nous permet de déterminer toutes les fonctions objectifs qui atteignent leur valeur optimale en un point commun. En comparaison, l'approche classique consiste à étudier la stabilité en variant un seul coefficient [Kol11; Fil11]. Cependant, si l'on souhaite varier les deux coefficients en même temps, cela entraîne la création de deux intervalles dépendants l'un de l'autre, rendant ainsi l'analyse plus complexe. L'approche que nous avons présentée offre donc une meilleure compréhension et une plus grande simplicité dans l'analyse de sensibilité des problèmes de programmation linéaire en deux dimensions.

DEUXIÈME PARTIE:

OPTIMISATION MULTI-CRITÈRE MULTI-NIVEAUX

Cette partie propose une synthèse des fondements de la programmation linéaire multi-critère à plusieurs niveaux, exposés en détail dans le chapitre 4. Ces fondements revêtent une importance cruciale pour la progression des travaux entrepris au sein de cette thèse. Par la suite, le chapitre 5 met en avant une contribution innovante à la résolution des problèmes de programmation linéaire mono-objectif à plusieurs niveaux. Enfin, le chapitre 6 insère deux avancées majeures inédites dans la résolution des problèmes de programmation linéaire multi-objectifs multi-niveaux, amplifiant ainsi la portée et la valeur des résultats obtenus.

CHAPITRE 4

RÉFLEXIONS ET FONDEMENTS THÉORIQUES : EXPLORATION DES CONCEPTS

Objectifs

Ce chapitre a pour ambition de présenter une introduction approfondie aux problèmes d'optimisation multi-critèreⁱ, en mettant en lumière la programmation linéaire multi-objectifs ainsi que les problèmes de programmation multi-objectifs multi-niveaux. Nous débuterons en établissant les bases conceptuelles en définissant les termes clés liés aux problèmes d'optimisation multi-critère. Ensuite, nous accorderons une attention particulière aux Problèmes de Programmation Linéaire Multi-Objectifs (PPLMO), en explorant leurs définitions, caractéristiques et les méthodes de résolution appropriées. Une classification des problèmes d'optimisation multi-critère sera proposée pour une meilleure appréhension de leurs particularités. Par la suite, nous approfondirons le principe de résolution d'un PPLMO, en mettant en exergue une méthode efficace pour surmonter ces défis. Enfin, une exploration approfondie des PPLMO-MN nous permettra de fournir une vue globale et complète des enjeux associés à ces problèmes d'optimisation sophistiqués.

i. Termes interchangeable : "critère" et "objectif" dans les problèmes d'optimisation multi-critère.

Sommaire

4.1	Problèmes d'Optimisation Multi-Critère	78
4.1.1	Exploration des Terminologies	79



4.1.2	À la Croisée des Objectifs : Une Exploration de la PMMO à travers le Cas d'un Achat de Voiture.	80
4.1.3	Approfondissement de la Programmation Linéaire Multi-Objectifs : Maximisation de Critères Multiples	81
4.1.4	Classification des Problèmes d'Optimisation Multi-Critère	81
4.1.5	Approche de Résolution des PPLMOs	85
4.1.6	La Méthode Adaptée pour la Recherche d'une Seule Solution Extrême Efficace d'un PPLMO à variable bornées	87
4.2	La Programmation Multi-Objectifs Multi-Niveaux	88
4.2.1	Approche de Résolution des PPMO-MNs	89



Dans ce chapitre et tout au long de la suite, lorsque nous évoquerons des solutions de compromis, des solutions non dominées ou efficaces, nous entendrons généralement les **meilleurs compromis**.

4.1 Problèmes d'Optimisation Multi-Critère

La Programmation Multi-Objectif (PMO) constitue une extension de la programmation classique à un seul objectif (mono-objectif) vers des cas où plusieurs fonctions objectifs sont expressément prises en compte dans les modèles d'optimisation mathématique [De 03; Vin89; Rad20; Col02; Noc99]. Cependant, lorsque ces fonctions sont contradictoires, cela entraîne un changement de paradigme. Le concept de solution optimale perd sa signification générale, car en règle générale, il n'existe pas de solution réalisable qui optimise simultanément toutes les fonctions objectifs. Dans la programmation à objectif unique, le paradigme de l'optimalité prévaut, assurant une comparabilité complète entre les paires d'alternatives possibles avec application de la transitivité. Ce problème est mathématiquement bien formulé, car nous disposons d'outils mathématiques suffisants pour résoudre les trois questions fondamentales de l'analyse : l'existence, l'unicité et la construction de la solution. Toutefois, lorsque plusieurs fonctions objectifs sont considérées, la notion d'ordre total dans l'espace des critères est perdue, invalidant ainsi les trois questions fondamentales qui sous-tendent tout processus de résolution [Ant16; Ehr05; Zim78; Zim76; Moh97; Kac23a].

Les racines de la Programmation Mathématique (PM) multi-objectifs plongent dans la théorie économique. En 1906, Pareto a défini un concept fondamental pour la PM avec des objectifs multiples, connu sous le nom de solution optimale de Pareto (aussi appelée solution efficace, non dominée ou non-inférieure). Une solution n'est pas dominée lorsqu'il n'existe pas d'autres solutions réalisables qui améliorent simultanément toutes les valeurs de la fonction objectif. En d'autres termes, l'amélioration d'un objectif entraîne la détérioration d'au moins une des autres valeurs de la fonction objectif [Gal77; Ise77; Luc87].

En 1982, M. Zeleny a mis en évidence de manière très expressive les différences essentielles



entre les modèles d'optimisation à objectif unique et multiples. Il a utilisé l'exemple d'un sac d'oranges pour illustrer son raisonnement, où l'objectif était de sélectionner d'abord la plus grande orange, puis simultanément la plus grande et la plus douce. Bien que cela ne soit pas un exemple de problème de PMO, son analyse reste très suggestive et les conclusions sont également applicables à la PMO. Le choix de la plus grande orange relève d'un problème d'optimisation purement technique, impliquant la mesure et l'ordonnement des oranges. Ce même processus s'applique à tout problème d'optimisation lié aux solutions possibles. Dans le second cas, si la plus grosse orange n'est pas la plus douce, il n'y a pas de solution optimale au problème. La sélection d'une orange nécessite un compromis subjectif entre les deux objectifs, à savoir la taille et la douceur, ce qui démontre que la résolution de problèmes ne se limite pas à des questions purement techniques. Toutefois, il existe un sous-ensemble d'oranges pour lesquelles la solution de compromis doit se trouver, à savoir la solution non-dominée, constituée d'oranges pour lesquelles il n'existe pas d'autre orange qui soit simultanément plus grande et plus sucrée [Zel74 ; Gar94 ; Phi72 ; Sad15 ; Bou92 ; Iqb10].

4.1.1 Exploration des Terminologies

Au sein des documents décisionnels portant sur de multiples critères, les termes les plus fréquemment utilisés sont les "attributs", les "objectifs", les "buts" et les "critères". Ces termes ne possèdent pas de définition universelle, et certains auteurs font des distinctions dans leur utilisation, tandis que d'autres les considèrent de manière interchangeable. Toutefois, nous ferons quelques distinctions entre ces termes pour clarifier leur usage.

4.1.1.1 Les attributs

Les attributs correspondent aux caractéristiques, aux qualités ou aux paramètres de performance associés aux différentes solutions de rechange. Par exemple, imaginons qu'une commission municipale doive décider de l'orientation de l'expansion future d'une ville. Les alternatives possibles incluent trois directions : est, sud-est et ouest. La commission doit évaluer les alternatives en fonction d'attributs tels que le coût de l'expansion des services, l'impact sur la vitalité du centre-ville, le risque d'inondation, la préservation de l'équilibre des quartiers, la proximité des installations récréatives existantes, et bien d'autres.

4.1.1.2 Les objectifs

Les objectifs sont les orientations que le décideur cherche à améliorer et reflètent donc ses désirs. Ils indiquent la direction dans laquelle l'organisation doit progresser. Dans le contexte des Problème de Programmation Multi-Objectif (PPMO), les solutions de rechange sont conçues pour optimiser ou satisfaire au mieux ces objectifs. Par exemple, dans l'élaboration de plans de développement pour le gouvernement d'un pays en développement, les objectifs pourraient in-



clure la maximisation du bien-être national, la réduction de la dépendance à l'égard de l'aide étrangère, la minimisation du taux de chômage, etc.

4.1.1.3 Les buts

Les buts sont les objectifs spécifiques que le décideur souhaite atteindre dans l'espace et le temps. Ils définissent un niveau souhaité (ou cible) à atteindre, tandis que les objectifs indiquent la direction désirée. Cependant, dans la littérature, la distinction entre ces deux termes est parfois floue, et ils sont souvent utilisés de manière interchangeable.

4.1.1.4 Les critères

Les critères correspondent aux normes de jugement ou aux règles permettant de vérifier l'acceptabilité des solutions. En ce sens, les critères incluent les attributs et/ou les objectifs. Un problème de prise de décision multi-critère peut donc impliquer la considération de multiples attributs et/ou objectifs. Cette terminologie est utilisée pour englober la classe générale des problèmes auxquels nous nous intéressons ici.

4.1.2 À la Croisée des Objectifs : Une Exploration de la PMMO à travers le Cas d'un Achat de Voiture.

Pour introduire les concepts fondamentaux de la PMMO, prenons l'exemple d'un processus de décision pour l'achat d'une nouvelle voiture parmi quatre modèles : une Mercedes GLC, une Seat Arona, une Ford Focus et une Volkswagen Caddy. Notre décision sera basée sur trois critères essentiels : le prix, la consommation d'essence et la puissance. Notre préférence va vers une voiture abordable, puissante et économe en carburant. Dans cette situation, nous sommes confrontés à un problème de décision comportant quatre alternatives et trois critères. Le défi réside dans le choix de la meilleure alternative parmi ces voitures, car la plus puissante est également celle qui consomme le plus d'essence, ce qui rend difficile l'achat d'une voiture à la fois puissante et économe en carburant à un prix abordable.

Modélisation du Problème Multi-Objectifs : Espace de Décision et Critères.

Dans ce contexte, nous définissons $S = \{\text{Mercedes}, \text{Seat}, \text{Ford}, \text{Volkswagen}\}$ comme l'ensemble réalisable ou l'ensemble des solutions de rechange pour ce problème de décision. Si nous notons le prix par f_1 et la consommation d'essence par f_2 , alors les fonctions $f_i : S \mapsto \mathbb{R}$ représentent les critères ou les fonctions objectifs. Le problème d'optimisation peut être formulé mathématiquement comme suit :

$$\min_{x \in S} (f_1(x), f_2(x)). \quad (4.1)$$

Dans cette formulation, nous cherchons à minimiser conjointement les valeurs des fonctions objectifs f_1 et f_2 pour trouver la meilleure alternative parmi les quatre modèles de voitures. La **PMO** permettra d'explorer les compromis possibles entre ces critères conflictuels et d'identifier les solutions non-dominées, c'est-à-dire les voitures qui ne peuvent pas être améliorées simultanément sur tous les critères. Ainsi, la **PMO** offre une approche puissante pour résoudre les problèmes de décision complexes avec plusieurs objectifs contradictoires.

4.1.3 Approfondissement de la Programmation Linéaire Multi-Objectifs : Maximisation de Critères Multiples

Dans cette section, nous nous concentrons exclusivement sur les modèles de **PPLMO** où toutes les fonctions objectifs et contraintes sont linéaires. Nous considérons des problèmes qui se présentent sous la forme de maximisation (utilisant l'opérateur "max"), où nous cherchons à optimiser plusieurs critères simultanément. Pour mieux appréhender ce cadre, nous commençons par présenter la forme standard du **PPLMO** :

$$\max_{x \in S} f(x) = c^t x, \quad (4.2)$$

où c^t est une matrice de constantes représentant les coefficients des fonctions objectifs, et $f(x)$ est le vecteur résultant des fonctions objectifs appliquées à l'alternative x . Ensuite, nous détaillons les contraintes qui définissent l'ensemble réalisable S , noté :

$$S = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0, b \in \mathbb{R}^m\},$$

Ici, S représente la région réalisable convexe, et m est le nombre de contraintes linéaires imposées sur les variables de décision x . Nous utilisons une matrice A de taille $(m \times n)$ pour décrire ces contraintes, et b est un vecteur de taille m .

Nous abordons également la notion d'espace critère, qui est l'espace à partir duquel les valeurs des critères sont prises. L'image de l'ensemble réalisable S par l'application $f = (f_1, \dots, f_k)$ est notée $Z := f(S)$, et représente l'ensemble réalisable dans l'espace des critères.



Cependant, définir le sens de « max » pour les vecteurs de fonctions objectifs $(f_1(x), \dots, f_k(x))$ devient délicat en raison de l'absence d'un ordre canonique sur \mathbb{R}^k par rapport à \mathbb{R} . En conséquence, nous devons recourir à des définitions plus faibles des ordres pour surmonter ce défi.

4.1.4 Classification des Problèmes d'Optimisation Multi-Critère

Dans cette section, nous amorcerons notre exploration en introduisant les relations binaires et en examinant certaines de leurs propriétés essentielles, ce qui nous permettra de définir diffé-



rentes classes d'ordres. Soit S un ensemble arbitraire. Une relation binaire sur S est représentée par un sous-ensemble \mathcal{R} de $S \times S$.

Définition 4.1.1 (Propriétés des Relations Binaires sur S).

- ◇ **Réflexive** : $\forall s \in S, (s, s) \in \mathcal{R}$.
- ◇ **Irréflexive** : $\forall s \in S, (s, s) \notin \mathcal{R}$.
- ◇ **Symétrique** : $\forall s^1, s^2 \in S, (s^1, s^2) \in \mathcal{R} \Rightarrow (s^2, s^1) \in \mathcal{R}$.
- ◇ **Asymétrique** : $\forall s^1, s^2 \in S, (s^1, s^2) \in \mathcal{R} \Rightarrow (s^2, s^1) \notin \mathcal{R}$.
- ◇ **Antisymétrique** : $\forall s^1, s^2 \in S, (s^1, s^2) \in \mathcal{R} \text{ et } (s^2, s^1) \in \mathcal{R} \Rightarrow s^1 = s^2$.
- ◇ **Transitive** : $\forall s^1, s^2, s^3 \in S, (s^1, s^2) \in \mathcal{R} \text{ et } (s^2, s^3) \in \mathcal{R} \Rightarrow (s^1, s^3) \in \mathcal{R}$.
- ◇ **Négativement Transitive** : $\forall s^1, s^2, s^3 \in S, (s^1, s^2) \notin \mathcal{R} \text{ et } (s^2, s^3) \notin \mathcal{R} \Rightarrow (s^1, s^3) \notin \mathcal{R}$.
- ◇ **Connectée** : $\forall s^1, s^2 \in S, (s^1, s^2) \in \mathcal{R} \text{ ou } (s^2, s^1) \in \mathcal{R} \text{ avec } s^1 \neq s^2$.
- ◇ **Fortement Connectée (ou Totale)** : $\forall s^1, s^2 \in S, (s^1, s^2) \in \mathcal{R} \text{ ou } (s^2, s^1) \in \mathcal{R}$.



Les propriétés de la définition 4.1.1 nous aideront à comprendre et à caractériser différents types de relations binaires dans les PPLMOs.

Définition 4.1.2. Une relation binaire \mathcal{R} sur l'ensemble S est définie comme suit :

- ◇ \mathcal{R} est une **relation d'équivalence** si elle est réflexive, symétrique et transitive.
- ◇ \mathcal{R} est un **pré-ordre** (ou **quasi-ordre**) si elle est réflexive et transitive.

Notations 4.1.3.

- ◇ Au lieu d'écrire $(s^1, s^2) \in \mathcal{R}$, nous utiliserons également la notation $s^1 \mathcal{R} s^2$.
- ◇ Lorsque \mathcal{R} est un pré-ordre, la paire (S, \mathcal{R}) est appelée un ensemble pré-ordonné.
- ◇ Dans le contexte des (pré)ordres, une autre notation pour la relation \mathcal{R} est pratique. Nous écrirons $s^1 \succeq s^2$ comme raccourci pour $(s^1, s^2) \in \mathcal{R}$, et $s^1 \not\succeq s^2$ pour $(s^1, s^2) \notin \mathcal{R}$. Nous utiliserons indifféremment la notation \mathcal{R} ou \succeq pour faire référence à cette relation. Cette notation peut être interprétée comme signifiant "préférée à".

Définition 4.1.4. Soit \succeq un pré-ordre. On définit deux autres relations étroitement associées à \succeq comme suit :

$$s^1 \prec s^2 \quad \text{si et seulement si} \quad s^1 \succeq s^2 \text{ et } s^2 \not\succeq s^1 \quad (4.3)$$

$$s^1 \sim s^2 \quad \text{si et seulement si} \quad s^1 \succeq s^2 \text{ et } s^2 \succeq s^1 \quad (4.4)$$

Proposition 4.1.5. Soit \succeq un pré-ordre sur S . Alors :

1. La relation définie en (4.3) est irréflexive et transitive, et la relation définie en (4.4) est une relation d'équivalence.
2. Toute relation binaire asymétrique est irréflexive.
3. Toute relation binaire transitive et irréflexive est asymétrique.

Définition 4.1.6. Soit (S, \succeq) un ensemble ordonné. La relation \succeq est appelée une **relation d'ordre total** si pour tout $x, y \in S$, on a soit $x \succeq y$, soit $y \succeq x$. En d'autres termes, cela signifie que tous les éléments de S sont toujours comparables deux à deux. Dans le cas contraire, l'ordre est dit **partiel**.



Par le choix d'un ordre sur \mathbb{R}^k , nous pouvons enfin définir le sens de « max » dans la formulation du problème (4.2). En effet, les vecteurs de la fonction objectif sont envoyés dans un espace ordonné (\mathbb{R}^k, \succeq) par une application Θ , où les comparaisons sont effectuées en utilisant la relation d'ordre \succeq . L'application Θ est appelée **Application modèle**.

Définition 4.1.7 (Composants d'un Problème d'Optimisation Multi-critère). Un problème d'optimisation multi-critère est composé de :

- ◇ La région réalisable S .
- ◇ Le vecteur de la fonction objectif $f = (f_1, \dots, f_k) : S \rightarrow \mathbb{R}^k$.
- ◇ L'espace des objectifs \mathbb{R}^k .
- ◇ L'ensemble ordonné (\mathbb{R}^k, \succeq) .
- ◇ L'application modèle Θ .

Exemple 4.1.8. Le problème (4.2) peut être reformulé de la manière suivante :

$$\max_{x \in S} \max_{i=1, \dots, k} f_i(x).$$



Nous devons donc choisir l'application modèle suivante :

$$\begin{aligned} \Theta : \mathbb{R}^k &\longrightarrow (\mathbb{R}, \succeq) \\ y = f(x) &\longrightarrow \Theta(f(x)) = \max_{i=1, \dots, k} f_i(x), \end{aligned}$$

où \succeq est l'ordre canonique de \mathbb{R} .

Les données du problème de **PMO** sont le vecteur de la fonction objectif f et l'espace des objectifs. L'application modèle fournit le lien entre l'espace des objectifs et l'ensemble ordonné dans lequel la signification de la maximisation est définie. Ainsi, avec ces trois principaux aspects donnés, la classification suivante :

$$(S, f, \mathbb{R}^n) / \Theta / (\mathbb{R}^k, \succeq)$$

décrit complètement un Problème d'Optimisation Multi-Critère (**POMC**). Maintenant, nous sommes prêts à donner une définition formelle des solutions optimales et des valeurs optimales des problèmes d'optimisation multi-critère.

Définition 4.1.9.

1. Une solution réalisable $x^* \in S$ est une solution optimale d'un POMC $(S, f, \mathbb{R}^n) / \Theta / (\mathbb{R}^k, \succeq)$, s'il n'existe aucune autre solution réalisable $x \neq x^*$ vérifiant :

$$\Theta(f(x)) \succeq \Theta(f(x^*)).$$

2. Pour une solution optimale x^* , $\Theta(f(x^*))$ est appelée valeur optimale du problème **PMO**.



◇ Étant donné que nous avons souvent affaire à des ordres qui ne sont pas totaux, une définition de l'optimalité comme :

$$\Theta(f(x^*)) \succeq \Theta(f(x)), \quad \text{pour tout } x \in S,$$

n'est pas possible en général.

◇ Pour des choix spécifiques de Θ et (\mathbb{R}^k, \succeq) , des termes spécifiques pour des solutions et des valeurs optimales sont couramment utilisés, tels que des solutions efficaces (voir la définition 4.1.11).

Notations 4.1.10 (Classes d'Optimisation Multi-Critère (OMC)). Une classe d'OMC est l'ensemble de tous les PPMO avec la même application modèle et le même ensemble ordonné, et est notée par :

$$\bullet/\Theta/(\mathbb{R}^k, \succeq).$$

En particulier, la classe " $\bullet/\Theta/(\mathbb{R}^k, \geq)$ " désignera la classe de toutes les PMO où l'optimalité est comprise dans un sens que l'on appelle l'efficacité.

Définition 4.1.11.

- ◇ Pour $z^1, z^2 \in Z$, on dit que z^1 domine z^2 si $z^1 \geq z^2$.
- ◇ Pour deux solutions réalisables $x^1, x^2 \in S$, on dit que x^1 domine x^2 si $f(x^1) \geq f(x^2)$.
- ◇ Une solution réalisable $x \in S$ est non dominée si elle n'est dominée par aucune autre solution réalisable de S . Autrement dit, il n'existe aucune solution réalisable $\hat{x} \neq x$ dans S , telle que :

$$f(\hat{x}) \geq f(x).$$

Dans ce cas, x est appelée une solution efficace ou une solution de Pareto.



Il existe d'autres types d'efficacité pour une solution optimale d'un problème de PMO. À titre d'exemple, on cite les solutions faiblement efficaces, les solutions fortement efficaces, les solutions proprement efficaces (ou propres), voir [Ehr05].

Exploration de la Classe de OMC Étudiée dans cette Thèse

Dans ce travail, nous nous concentrons exclusivement sur les problèmes d'optimisation multi-critère de la classe $\bullet/Id_{\mathbb{R}^k}/(\mathbb{R}^k, \geq)$, où " \geq " est une relation d'ordre qui compare deux vecteurs quelconques composante par composante. En d'autres termes, pour tout $z, \hat{z} \in \mathbb{R}^k$, on a $z \geq \hat{z}$ si et seulement si $z_i \geq \hat{z}_i$ pour tous les $i = 1, \dots, k$.

4.1.5 Approche de Résolution des PPLMOs

Dans cette partie, nous présentons brièvement une classe de méthodes appelées "méthodes de la somme pondérée", qui permettent d'obtenir les solutions efficaces d'un PPLMO en résolvant un problème à objectif unique de la forme :



$$\max_{x \in S} \sum_{i=1}^k \lambda_i f_i(x), \text{ avec } \lambda \in \mathbb{R}^k, \quad (4.5)$$

qui, en termes de classification, est écrit comme :

$$(S, f, \mathbb{R}^k) / \langle \lambda, \cdot \rangle / (\mathbb{R}, \geq),$$

où $\langle \lambda, \cdot \rangle$ est le produit scalaire dans \mathbb{R}^k , voir [Ben81; Wen77; Eck78a; Eck78b; Arm91; Eck80]. Nous appelons le problème d'optimisation à objectif unique (ou scalaire) (4.5) une scalarisation de somme pondérée du PPLMO (4.2). Cependant, pour établir un lien entre les solutions des deux problèmes, nous devons d'abord procéder à une analyse approfondie.

Notations 4.1.12.

- ◇ L'ensemble des vecteurs à composantes toutes positives ou nulles est noté par \mathbb{R}_{\geq}^k , c'est-à-dire : $\mathbb{R}_{\geq}^k := \{ \lambda \in \mathbb{R}^k : \lambda_i \geq 0, \forall i = 1, \dots, k \}$.
- ◇ On note par S_E et Z_E l'ensemble des solutions efficaces et des points efficaces respectivement, où $Z_E = f(S_E)$.

Définition 4.1.13.

1. L'ensemble des solutions du problème (4.5), par rapport à un vecteur λ fixé dans \mathbb{R}^k , est défini comme suit :

$$SOL(\lambda, S) := \left\{ x^* \in S : \langle \lambda, f(x^*) \rangle = \max_{x \in S} \langle \lambda, f(x) \rangle \right\}.$$

2. L'ensemble de toutes les solutions du problème (4.5) est alors défini par :

$$SOL := \bigcup_{\lambda \in \Lambda} SOL(\lambda, S),$$

où

$$\Lambda = \left\{ \lambda \in \mathbb{R}_{\geq}^k : \sum_{i=1}^k \lambda_i = 1 \right\}.$$

Théorème 4.1.14. $x^* \in S$ est une solution efficace si et seulement si x^* est une solution du problème (4.5) pour un certain $\lambda \in \Lambda$. En d'autres termes : $S_E = SOL$.

4.1.6 La Méthode Adaptée pour la Recherche d'une Seule Solution Extrême Efficace d'un PPLMO à variable bornées

Considérons le programme linéaire multi-objectifs à variables bornées, exprimé sous sa forme canonique :

$$\begin{cases} \max_x c^t x \\ Ax = b \\ l \leq x \leq u, \end{cases} \quad (4.6)$$

où l et u sont des vecteurs $(n + m)$ -dimensionnels bornés tels que $|l| < \infty$ et $|u| < \infty$.

Algorithme 3 La Méthode Adaptée de Programmation Linéaire pour la Résolution d'un PPLMO à Variables Bornées.

Entrée : Les vecteurs c , b , l et u , ainsi que la matrice A .

Étape 1 : Obtenir une solution réalisable x^0 du problème (4.6). Si x^0 existe, passer à l'étape 2. Sinon, arrêter, car le problème est irréalisable [Nyi21].

Étape 2 : Obtenir la solution (y^0, r^0, v^0, w^0) du PPL mono-objectif auxiliaire suivant :

$$\begin{cases} y^t b - r^t c^t x^0 - v^t l + w^t u \rightarrow \min \\ y^t A + r^t c - v^t + w^t = s^t \\ y \in \mathbb{R}^m, r \geq 0, w \geq 0, v \geq 0, \end{cases} \quad (4.7)$$

où $s^t = (1, \dots, 1) \in \mathbb{R}^k$.

Étape 3 : Obtenir le vecteur $\lambda = (r^0 + s)$, et passer à l'étape 4.

Étape 4 : Formuler le PPL mono-objectif suivant :

$$\begin{cases} \max_x \lambda^t c^t x \\ Ax = b \\ l \leq x \leq u. \end{cases} \quad (4.8)$$

Étape 5 : Appliquer l'algorithme 2 mono-objectif de la méthode adaptée, décrit au chapitre 2, pour résoudre le PPL (4.8) et obtenir la solution x^* .

Sortie : La solution efficace x^* du problème (4.6).



Pour plus de détails, voir [Gab95; Gab78; Del18b; Rad11b; Bra10; Rad07c; Rad12; Kac22b; Rad07b].



4.2 La Programmation Multi-Objectifs Multi-Niveaux

L'optimisation à plusieurs niveaux s'avère être un puissant outil pour modéliser les problèmes de prise de décision impliquant plusieurs Décideurs (DCDs) interagissant dans une structure hiérarchique. Mathématiquement, dans un Problème de Programmation Multi-Objectifs Multi-Niveaux (PPMO-MN), les variables du niveau supérieur sont considérées comme des paramètres dans les problèmes de niveau inférieur, et les problèmes du niveau supérieur sont considérés comme des contraintes dans les problèmes de niveau inférieur. Bien que la plupart des recherches sur les programmes à niveaux multiples se soient concentrées sur les cas comprenant uniquement deux niveaux [Wen92; Yib19; Pie11; Abo10; Anz15], il existe de nombreux PPMO-MNs impliquant plus de deux niveaux, comme mentionné dans [Lu 16; Shi96; Bak10; Kac22c; Sei79].

Problème 4.2.1. Soient $x_l, x_s \in \mathbb{R}$ et S une partie convexe de \mathbb{R}^2 . Les fonctions f et g sont à valeurs dans \mathbb{R}^3 et \mathbb{R}^4 , respectivement.

Imaginons que nous avons deux décideurs, le premier étant le *leader* et le deuxième le *suiveur*. Le leader doit prendre une décision en premier et cherche à maximiser la fonction f dans l'ensemble d'alternatives S par rapport à la seule variable x_l . Cela conduit au problème suivant : $\max_{x_l} f(x_l, x_s)$. La solution optimale de ce problème est notée (x_l^*, x_s^*) , où x_l^* représente la solution satisfaisante idéale du leader.

Comme les variables de décision sont liées par le vecteur (x_l, x_s) , le leader doit imposer de nouvelles contraintes au suiveur afin que sa solution ne s'éloigne pas trop de la solution optimale (x_l^*, x_s^*) . Par exemple, le leader peut désigner un sous-ensemble $S^* \subset S$ contenant (x_l^*, x_s^*) . Ensuite, le suiveur doit maximiser sa fonction g en respectant cette nouvelle contrainte. Autrement dit, le suiveur doit résoudre le problème suivant : $\max_{x_s \in S^{**}} g(x_l, x_s)$, où $S^{**} = \{x_s \in \mathbb{R} : (x_l, x_s) \in S^*\}$. Le suiveur obtient ainsi un nouveau vecteur de solutions (x_l^{**}, x_s^{**}) .

Formulation du problème 4.2.1

Niveau 1 : $\text{Max}_{x_l} f(x_l, x_s)$,
où x_s résout

Niveau 2 : $\text{Max}_{x_s} g(x_l, x_s)$,
(4.9)

soumis à $(x_l, x_s) \in S$.



- ◊ Lorsque les variables ne sont pas liées ou que l'ensemble des alternatives est de dimension 1, le suiveur peut considérer S comme son domaine de recherche de solution.
- ◊ La liaison des variables de décision x_l et x_s reflète la notion hiérarchique.

Plus généralement, si on dispose de plus de deux décideurs, disons $P \geq 2$ décideurs, notons le Décideur du $p^{\text{ième}}$ niveau (DCD_p) qui a le contrôle sur les variables de décision $\bar{x}^p = x_{p1}, \dots, x_{pn_p} \in \mathbb{R}^{n_p}$, $p = 1, \dots, P$, où $x = (\bar{x}^1, \dots, \bar{x}^P)^t$, $n = n_1 + \dots + n_P$, $k = k_1 + \dots + k_P$ tel que k_p est le nombre de fonctions objectifs à chaque niveau, et

$$F_p : \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_P} \mapsto \mathbb{R}^{k_p} .$$

Formulation d'un PPMO à P niveaux

$$\text{Niveau 1 : } \begin{aligned} & \text{Max}_{\bar{x}^1} F_1(x), \\ & \text{tel que } \bar{x}^2, \dots, \bar{x}^P \text{ résout} \end{aligned}$$

$$\text{Niveau 2 : } \begin{aligned} & \text{Max}_{\bar{x}^2} F_2(x), \\ & \vdots \\ & \text{tel que } \bar{x}^P \text{ résout} \end{aligned}$$

$$\text{Niveau } P : \text{Max}_{\bar{x}^P} F_P(x), \tag{4.10}$$

soumis à $x \in S \subset \mathbb{R}^n$.

Notations 4.2.2.

- ◇ Si S est défini par des contraintes linéaires et les fonctions F_p sont également linéaires, alors le problème (4.10) est qualifié de Problème de Programmation Linéaire Multi-Objectifs Multi-Niveaux (PPLMO-MN).
- ◇ Un problème de programmation linéaire multi-niveaux contenant une seule fonction objectif dans chaque niveau est appelé Problème de Programmation Linéaire MOno-Objectifs Multi-Niveaux (PPL(MO)O-MN).

4.2.1 Approche de Résolution des PPMO-MNs

Les PPMO-MNs sont caractérisés par la présence d'une hiérarchie de décideurs, où chaque niveau de décideur prend en compte les décisions du niveau précédent pour résoudre son propre problème multi-objectifs. Voici une approche pratique pour résoudre ces problèmes :

1. Définition des Niveaux de Décision : Considérons un problème multi-niveau avec $P \geq 2$ niveaux de décision. Le premier niveau est le décideur supérieur, et les niveaux suivants sont les décideurs subordonnés. Pour ce type de problème, les variables x_l et x_s doivent respecter des contraintes globales qui définissent la région réalisable S du problème multi-niveau.

2. Résolution du Problème du Premier Décideur (Niveau Supérieur) : Le décideur du premier niveau résout son problème multi-objectifs sans contrainte des autres niveaux. Soit F_1 la fonction objectif du premier niveau, le problème peut être formulé comme suit :

$$\max_{x_l^1} F_1(x_l^1, x_s^1), \tag{4.11}$$

soumis à $(x_l^1, x_s^1) \in S$. Où x_l^1 représente les variables de décision du premier niveau et x_s^1 les



variables de décision des niveaux suivants.

3. Imposition de Contraintes aux Niveaux Subordonnés : Après avoir obtenu sa solution optimale x_l^{1*} , le premier niveau utilise cette solution pour imposer de nouvelles contraintes aux niveaux subordonnés, restreignant ainsi leur espace de recherche à un sous-ensemble $S^* \subset S$ de l'ensemble des solutions réalisables.

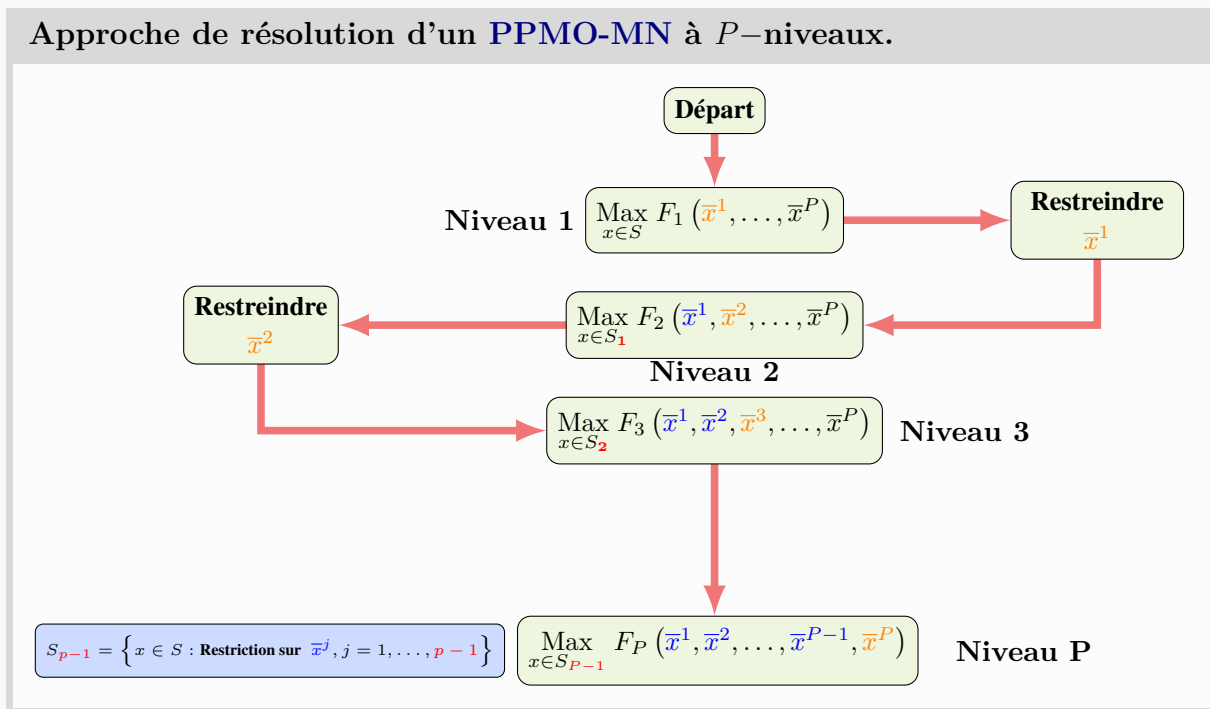
4. Résolution des Problèmes des Décideurs Subordonnés : Les décideurs subordonnés résolvent ensuite leur propre problème multi-objectifs en tenant compte des nouvelles contraintes imposées par le niveau supérieur. Le décideur du $p^{\text{ième}}$ niveau résout son problème avec S^* comme l'espace de recherche pour ses variables de décision. Le problème du $p^{\text{ième}}$ décideur peut être formulé comme suit :

$$\max_{x_s^p} F_p(x_l^p, x_s^p), \tag{4.12}$$

soumis à $(x_l^p, x_s^p) \in S^*$.

5. Itération du Processus : Le processus continue en itérant les étapes 3 et 4 pour chaque niveau subordonné jusqu'à ce que le dernier niveau soit atteint.

6. Obtention du Compromis : Une fois que tous les niveaux ont résolu leurs problèmes en améliorant progressivement la solution du niveau supérieur. Le compromis final est atteint au dernier niveau, où toutes les décisions sont prises en compte de manière équilibrée, en prenant en considération les contraintes imposées par les niveaux supérieurs. Ainsi, le compromis final est la solution du dernier niveau, qui reflète les préférences de chaque niveau de décision et aboutit à une solution efficace et équilibrée pour les problèmes de programmation multi-objectifs multi-niveaux.



CHAPITRE 5

UNE APPROCHE NOVATRICE POUR LA RÉSOLUTION D'UN PROBLÈME DE PROGRAMMATION LINÉAIRE MONO-OBJECTIF MULTI-NIVEAUX PPL(MO)O-MN

Objectifs

Dans ce chapitre, nous proposons une amélioration de l'approche développée par S.B. Sinha et S. Sinha dans [Sin04], en utilisant l'algorithme 2 de la méthode adaptée de Gabasov.

Notre approche repose sur la construction d'une application que nous appelons **Application de réduction des intervalles**, qui permet de réduire les intervalles des variables de décision contrôlées par le niveau précédent. Nous utilisons ensuite cette application pour définir une nouvelle estimation de sous-optimalité.

Nous développons deux versions d'un algorithme pour résoudre un **PPL(MO)O-MN**ⁱ. Les deux versions de l'algorithme sont divisées en trois parties. Tout d'abord, nous calculons la solution de tous les objectifs indépendamment, puis nous déterminons les bornes de toutes les variables de décision. Enfin, nous entamons la recherche d'un compromis satisfaisant en résolvant $P - 1$ problèmes de programmation linéaire mono-objectif à variables bornées en utilisant l'algorithme 2 de la méthode adaptée mono-objectif.



Sommaire

5.1	Formulation du PPL(MO)O-MN	92
5.2	Formulation Linéaire du $p^{\text{ième}}$ Niveau	93
5.2.1	Construction de l'Application de Réduction des Intervalles	94
5.2.2	Estimation de Sous-Optimalité pour le $p^{\text{ième}}$ Niveau	100
5.3	Résolution du PPL(MO)O-MN (5.1) avec la Méthode Adaptée de Gabasov	102
5.3.1	Algorithme - Première Version	102
5.3.2	Illustration Numérique de la 1 ^{er} -Version de l'Algorithme	103
5.3.3	Algorithme - Deuxième Version	105
5.3.4	Illustration Numérique de la 2 ^{ième} -Version de l'Algorithme	107
5.4	Conclusion	110

Résumé

Dans cette contribution ⁱⁱ, nous proposons un algorithme novateur pour résoudre un PPL(MO)O-MN, en utilisant la méthode adaptée constructive de programmation linéaire. Cette approche est une version modifiée de l'approche de programmation linéaire développée par S.B. Sinha et S. Sinha.

Notre démarche commence par la construction d'une application appelée Application de Réduction des Intervalles (ARI), qui réduit les intervalles des variables de décision contrôlées par le Décideur (DCD) du niveau précédent. Cette application joue un rôle clé dans la définition d'une nouvelle estimation de sous-optimalité pour le problème étudié. Toutes les étapes de construction de cette approche sont soigneusement vérifiées et illustrées par deux exemples numériques.

ii. Cet article est actuellement en révision dans une revue spécialisée.

Mots clés

Programmation linéaire multi-niveau, programmation linéaire, méthode adaptée, estimation de sous-optimalité, application affine.

5.1 Formulation du PPL(MO)O-MN

Pour chaque $p = 1, \dots, P$ et $j = 1, \dots, n_p$, la fonction objectif $f_p(\bar{x}^1, \dots, \bar{x}^P) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \dots \times \mathbb{R}^{n_P} \rightarrow \mathbb{R}$ du $p^{\text{ième}}$ niveaux du PPL(MO)O-MN est définie comme suit :

$$f_p(x) = c_p^{1j}\bar{x}^1 + c_p^{2j}\bar{x}^2 + \dots + c_p^{Pj}\bar{x}^P = c_p^{11}x_{11} + \dots + c_p^{1n_1}x_{1n_1} + c_p^{21}x_{21} + \dots + c_p^{2n_2}x_{2n_2} + c_p^{31}x_{31} + \dots + \dots + c_p^{P1}x_{P1} + \dots + c_p^{Pn_P}x_{Pn_P}.$$

i. *Problème de Programmation Linéaire Mono-Objectifs Multi-Niveaux*



Formulation d'un Problème de Programmation Linéaire à P niveaux Avec un Seul Objectif à Chaque Niveau

$$\begin{aligned}
\text{Niveau 1 : } & \underset{\bar{x}^1}{\text{Max}} f_1(x) = \underset{\bar{x}^1}{\text{Max}} f_1(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^P) = \underset{\bar{x}^1}{\text{Max}} c_1^t x, \\
& \text{tel que } \bar{x}^2, \dots, \bar{x}^P \text{ résout :} \\
\text{Niveau 2 : } & \underset{\bar{x}^2}{\text{Max}} f_2(x) = \underset{\bar{x}^2}{\text{Max}} f_2(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^P) = \underset{\bar{x}^2}{\text{Max}} c_2^t x, \\
& \vdots \\
& \vdots \\
& \vdots \\
& \text{tel que } \bar{x}^P \text{ résout :} \\
\text{Niveau } P : & \underset{\bar{x}^P}{\text{Max}} f_P(x) = \underset{\bar{x}^P}{\text{Max}} f_P(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^P) = \underset{\bar{x}^P}{\text{Max}} c_P^t x,
\end{aligned} \tag{5.1}$$

soumis à :

$$x \in S = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0, b \in \mathbb{R}^m\}.$$

Où, $S \neq \emptyset$ est la région réalisable convexe, m est le nombre des contraintes, $c_p^{i,j}$ pour $i, p = 1, \dots, P, j = 1, \dots, n_i$ sont des constantes et A est une matrice de taille $m \times n$.

Notations 5.1.1.

1. $\overset{o}{x}^p$ est la solution optimale du DCD_p et $\overset{o}{x}_{i,j}^p$ en est sa $i, j^{\text{ième}}$ composante.
2. $l^{(p-1)}$ et $u^{(p-1)}$ représentent les bornes, inférieure et supérieure respectivement, des variables de décision au $p^{\text{ième}}$ niveau (ce sont les nouvelles bornes des variables de décision après réduction de leurs intervalles par le Décideur du $(p-1)^{\text{ième}}$ niveau (DCD_{p-1}). Les notations $l_{i,j}^{(p-1)}$ et $u_{i,j}^{(p-1)}$ désignent leurs $i, j^{\text{ième}}$ composantes.
3. L'indice dans la partie supérieure droite p indique que nous sommes dans le $p^{\text{ième}}$ niveau, et l'indice en bas à droite i, j désigne la $i, j^{\text{ième}}$ composante.
4. $LW_{p-1,j}$ et $UP_{p-1,j}$ sont les intervalles inférieur et supérieur respectivement de la $(p-1, j)^{\text{ième}}$ composante du vecteur des variables de décision.

5.2 Formulation Linéaire du $p^{\text{ième}}$ Niveau

Dans cette section, nous nous inspirons de l'approche de programmation linéaire développée par S.B. Sinha et S. Sinha dans [Sin04] pour formuler le $p^{\text{ème}}$ niveau du problème (5.1) sous forme d'un problème de programmation linéaire mono-objectif à variables bornées. Cela nous permettra ensuite d'appliquer l'algorithme 2 de la méthode adaptée.



5.2.1 Construction de l'Application de Réduction des Intervalles

L'application ξ_{p-1} que nous développerons permettra de fournir l'intervalle réduit des variables de décision contrôlées par le DCD_{p-1} , en suivant précisément les étapes (3)-(9) de l'algorithme proposé par S.B. Sinha et S. Sinha.



L'indice $i = p - 1$ indique que nous traitons les variables de décision qui sont sous le contrôle du $(p - 1)^{i\text{ème}}$ décideur.

Considérons le problème de programmation linéaire à P niveaux (5.1), où $P \geq 2$, comprenant m contraintes linéaires, n variables de décision, et une seule fonction objectif à chaque niveau. Pour $p = 1, \dots, P$, nous notons par $\overset{o}{x}^p$ la solution optimale du PPL suivant :

$$\begin{cases} \max_x f_p(x) = c_p^t x \\ Ax \leq b \\ x \geq 0. \end{cases} \quad (5.2)$$

Ensuite, nous trouvons les bornes de chacune des n variables de décision comme suit :

$$l_{i,j}^{(0)} = \min_{q \in \{1, \dots, P\}} \{ \overset{o}{x}_{i,j}^q \}, \quad u_{i,j}^{(0)} = \max_{q \in \{1, \dots, P\}} \{ \overset{o}{x}_{i,j}^q \}, \quad i = 1, \dots, P, j = 1, \dots, n_i. \quad (5.3)$$

On pose $l_{i,j}^{(1)} \leftarrow l_{i,j}^{(0)}$ et $u_{i,j}^{(1)} \leftarrow u_{i,j}^{(0)}$. Après, on considère les $2n$ intervalles idéaux suivants :

$$l_{i,j}^{(1)} \leq x_{i,j} \leq u_{i,j}^{(1)}, \quad i = 1, \dots, P, j = 1, \dots, n_i. \quad (5.4)$$

Contrôle de DCD_p : Les différentes possibilités.

Sous le contrôle de DCD_p , chaque variable de décision, peut se situer dans un intervalle idéal ou atteindre l'une de ses bornes inférieures ou supérieures. Voici les cas distingués :

$$\overset{o}{x}_{p-1,j}^{(p-1)} = l_{p-1,j}^{(p-1)}$$

3^{ième} Cas

$$l_{p-1,j}^{(p-1)} < \overset{o}{x}_{p-1,j}^{(p-1)} < u_{p-1,j}^{(p-1)}$$

1^{er} Cas

$$\overset{o}{x}_{p-1,j}^{(p-1)} = u_{p-1,j}^{(p-1)}$$

2^{ième} Cas

Cas 1 : Le $(p - 1)^{i\text{ème}}$ décideur choisit les intervalles réduits de ses variables de décision en vérifiant le signe avant la variable de décision $x_{p-1,j}$, $j = 1, \dots, n_{p-1}$, dans $f_{p-1}(x)$.

Si le signe est négatif, il choisit l'intervalle inférieur $LW_{p-1,j} = \left[l_{p-1,j}^{(p-1)}, \overset{o}{x}_{p-1,j}^{(p-1)} \right]$, en utilisant

$$\begin{aligned} LW_{p-1,j} : \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] &\rightarrow LW_{p-1,j} \\ x_{p-1,j} &\rightarrow \frac{\left(\overset{o}{x}_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)} \right) x_{p-1,j} + l_{p-1,j}^{(p-1)} \left(u_{p-1,j}^{(p-1)} - \overset{o}{x}_{p-1,j}^{(p-1)} \right)}{u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}}. \end{aligned} \quad (5.5)$$



Si le signe est positif, il choisit l'intervalle supérieur $UP_{p-1,j} = \left[x_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right]$, en utilisant l'application :

$$\begin{aligned} U_{p-1,j} : \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] &\rightarrow UP_{p-1,j} \\ x_{p-1,j} &\rightarrow \frac{\left(u_{p-1,j}^{(p-1)} - x_{p-1,j}^{(p-1)} \right) x_{p-1,j} + u_{p-1,j}^{(p-1)} \left(x_{p-1,j} - l_{p-1,j}^{(p-1)} \right)}{u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}}. \end{aligned} \quad (5.6)$$



Les applications $L_{p-1,j}$ et $U_{p-1,j}$ sont définies de manière similaire à celles présentées dans (1.9) et (1.10), avec les paramètres suivants :

$$\alpha_1 = l_{p-1,j}^{(p-1)}, \quad \alpha_2 = u_{p-1,j}^{(p-1)}, \quad \alpha_3 = x_{p-1,j}^{(p-1)}.$$

Ces applications agissent spécifiquement sur la $(p-1, j)^{\text{ième}}$ composante.

Définition 5.2.1 (L'application Signe). L'application "sign" attribue la valeur +1 aux nombres positifs, -1 aux nombres négatifs et 0 aux nombres nuls.

Proposition 5.2.2 (Définition). Pour une variable de décision donnée $x_{p-1,j}$, où $x_{p-1,j} \in \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right]$, nous définissons les quantités suivantes :

$$T_{p-1,j}^- = 1 - \text{sign} \left(c_{p-1}^{p-1,j} \right) \quad \text{et} \quad T_{p-1,j}^+ = 1 + \text{sign} \left(c_{p-1}^{p-1,j} \right).$$

Ensuite, nous définissons la fonction $\psi_{p-1,j} (x_{p-1,j})$ comme :

$$\psi_{p-1,j} (x_{p-1,j}) = \frac{T_{p-1,j}^- L_{p-1,j} (x_{p-1,j}) + T_{p-1,j}^+ U_{p-1,j} (x_{p-1,j})}{2}.$$

Lorsque $c_{p-1}^{p-1,j} \neq 0$, on a :

$$\psi_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right) = \begin{cases} LW_{p-1,j} & \text{si } c_{p-1}^{p-1,j} < 0, \\ UP_{p-1,j} & \text{si } c_{p-1}^{p-1,j} > 0. \end{cases}$$

Démonstration. Puisque $c_{p-1}^{p-1,j} \neq 0$, cela signifie que $(c_{p-1}^{p-1,j} > 0 \text{ ou } c_{p-1}^{p-1,j} < 0)$. Par conséquent, nous avons deux cas à considérer :

1. Lorsque $c_{p-1}^{p-1,j} > 0$, on a : $T_{p-1,j}^+ = 2$ et $T_{p-1,j}^- = 0$. Alors, la fonction $\psi_{p-1,j}$ devient :

$$\psi_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right) = U_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right) = UP_{p-1,j}.$$



2. Lorsque $c_{p-1}^{p-1,j} < 0$, on a :

$$T_{p-1,j}^+ = 0 \quad \text{et} \quad T_{p-1,j}^- = 2.$$

La fonction $\psi_{p-1,j}$ devient :

$$\psi_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right) = L_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right) = LW_{p-1,j}.$$

Le cas où $c_{p-1}^{p-1,j} = 0$ est couvert par la proposition 5.2.3. ■



Nous pouvons remarquer que si $\text{sign} \left(c_{p-1}^{p-1,j} \right) \neq 0$, alors $\psi_{p-1,j}$ envoie l'intervalle $\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right]$ soit sur $LW_{p-1,j}$, soit sur $UP_{p-1,j}$. Cependant, si $\text{sign} \left(c_{p-1}^{p-1,j} \right) = 0$, cela signifie que la composante $x_{p-1,j}$ n'apparaît pas dans la fonction f_{p-1} , et donc elle n'influence pas la maximisation de cette fonction. Dans ce cas, nous n'effectuons pas de réduction de l'intervalle $\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right]$, car cela n'aurait aucun effet sur la maximisation de la fonction. Nous appliquons alors simplement une identité pour garder l'intervalle inchangé. Il est important de noter que ces mises à jour ne seront appliquées que si $c_p^{p-1,j} \neq 0$. Sinon, le décideur du niveau inférieur DCD_p ne les utilisera pas, car la variable concernée n'apparaît pas dans la fonction f_p .

Toutes ces considérations sont prises en compte par l'application $\nu_{p-1,j}$, qui gère le cas où $c_p^{p-1,j} = 0$, comme indiqué dans la proposition 5.2.3.

Proposition 5.2.3 (Définition). Pour une variable de décision donnée $x_{p-1,j}$, où $x_{p-1,j} \in \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right]$, nous définissons les quantités suivantes :

$$S_{p-1}^{p-1,j} = \text{sign} \left(c_{p-1}^{p-1,j} \right) \quad \text{et} \quad S_p^{p-1,j} = \text{sign} \left(c_p^{p-1,j} \right).$$

En utilisant la notation de la définition 5.2.2, nous définissons alors :

$$\nu_{p-1,j} \left(x_{p-1,j} \right) = \left(S_p^{p-1,j} S_{p-1}^{p-1,j} \right)^2 \psi_{p-1,j} \left(x_{p-1,j} \right) + \left(1 - \left(S_p^{p-1,j} S_{p-1}^{p-1,j} \right)^2 \right) x_{p-1,j}.$$

Ainsi, si à la fois $c_{p-1}^{p-1,j} \neq 0$ et $c_p^{p-1,j} \neq 0$, nous avons :

$$\nu_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right) = \psi_{p-1,j} \left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right] \right).$$

Sinon, si $c_{p-1}^{p-1,j} = 0$ ou $c_p^{p-1,j} = 0$, alors nous appliquons simplement l'application identité, c'est-à-dire : $\nu_{p-1,j} = Id \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} \right]$.

Démonstration.

1. Lorsque à la fois $c_{p-1}^{p-1,j} \neq 0$ et $c_p^{p-1,j} \neq 0$, nous avons :

$$(S_p^{p-1,j} S_{p-1}^{p-1,j})^2 = 1, \quad 1 - (S_p^{p-1,j} S_{p-1}^{p-1,j})^2 = 0$$

et

$$\nu_{p-1,j} \left([l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] \right) = \psi_{p-1,j} \left([l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] \right).$$

2. Sinon, si $c_{p-1}^{p-1,j} = 0$ ou $c_p^{p-1,j} = 0$, alors nous appliquons simplement l'application identité, c'est-à-dire :

$$(S_p^{p-1,j} S_{p-1}^{p-1,j})^2 = 0 \quad \text{et} \quad 1 - (S_p^{p-1,j} S_{p-1}^{p-1,j})^2 = 1$$

et

$$\nu_{p-1,j} \left([l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] \right) = [l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] = Id_{[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}]}.$$

■

Cas 2 : Si $x_{p-1,j}^{(p-1)} = u_{p-1,j}^{(p-1)}$, alors le choix de réduction d'intervalle sera le suivant :

$$LW_{p-1,j}^{\alpha_{p-1,j}} = [l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} - \alpha_{p-1,j}],$$

en utilisant l'application

$$\begin{aligned} L_{p-1,j}^{\alpha_{p-1,j}} : [l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] &\rightarrow LW_{p-1,j}^{\alpha_{p-1,j}} \\ x_{p-1,j} &\rightarrow \frac{(u_{p-1,j}^{(p-1)} - (l_{p-1,j}^{(p-1)} + \alpha_{p-1,j})) x_{p-1,j} + \alpha_{p-1,j} l_{p-1,j}^{(p-1)}}{u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}}. \end{aligned} \quad (5.7)$$

Cas 3 : Si $x_{p-1}^{(p-1)} = l_{p-1,j}^{(p-1)}$, alors le choix de réduction d'intervalle va être comme suit :

$$UP_{p-1,j}^{\alpha_{p-1,j}} = [l_{p-1,j}^{(p-1)} + \alpha_{p-1,j}, u_{p-1,j}^{(p-1)}],$$

en utilisant l'application :

$$\begin{aligned} U_{p-1,j}^{\alpha_{p-1,j}} : [l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] &\rightarrow UP_{p-1,j}^{\alpha_{p-1,j}} \\ x_{p-1,j} &\rightarrow \frac{(u_{p-1,j}^{(p-1)} - (l_{p-1,j}^{(p-1)} + \alpha_{p-1,j})) x_{p-1,j} + \alpha_{p-1,j} u_{p-1,j}^{(p-1)}}{u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}}, \end{aligned} \quad (5.8)$$

où $\alpha_{p-1,j} \in \mathbb{R}$ est un nombre arbitraire choisi par le DCD $_{p-1}$, tel que :

$$\alpha_{p-1,j} < u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}. \quad (5.9)$$



Proposition 5.2.4 (Definition). Pour chaque composante $x_{p-1,j}^{o(p-1)}$, définissons

$$A_{p-1,j} = \text{sign} \left(u_{p-1,j}^{(p-1)} - x_{p-1,j}^{o(p-1)} \right) \quad \text{et} \quad B_{p-1,j} = \text{sign} \left(x_{p-1,j}^{o(p-1)} - l_{p-1,j}^{(p-1)} \right).$$

Ensuite, nous définissons l'application $\hat{\psi}_{p-1,j}$ comme suit :

$$\hat{\psi}_{p-1,j}(x_{p-1,j}) = B_{p-1,j} L_{p-1,j}^{\alpha_{p-1,j}}(x_{p-1,j}) + A_{p-1,j} U_{p-1,j}^{\alpha_{p-1,j}}(x_{p-1,j}).$$

Alors, pour tout $x_{p-1,j} \in [l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}]$, nous avons les cas suivants :

- ◇ Si $A_{p-1,j} = 0$, alors $\hat{\psi}_{p-1,j}(x_{p-1,j}) = L_{p-1,j}^{\alpha_{p-1,j}}(x_{p-1,j})$.
- ◇ Si $B_{p-1,j} = 0$, alors $\hat{\psi}_{p-1,j}(x_{p-1,j}) = U_{p-1,j}^{\alpha_{p-1,j}}(x_{p-1,j})$.

Démonstration. Cette proposition est immédiate. ■

Proposition 5.2.5 (Definition). Soit $x_{p-1,j} \in [l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}]$, en utilisant les notations de la proposition 5.2.3, nous définissons

$$\hat{v}_{p-1,j}(x_{p-1,j}) = \left(S_p^{p-1,j} S_{p-1}^{p-1,j} \right)^2 \hat{\psi}_{p-1,j}(x_{p-1,j}) + \left(1 - \left(S_p^{p-1,j} S_{p-1}^{p-1,j} \right)^2 \right) x_{p-1,j}.$$

Alors,

$$\hat{v}_{p-1,j} \left([l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] \right) = \begin{cases} \hat{\psi}_{p-1,j} \left([l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}] \right) & \text{si } c_{p-1}^{p-1,j} \neq 0 \text{ et } c_p^{p-1,j} \neq 0, \\ \hat{v}_{p-1,j} = Id_{[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}]} & \text{sinon.} \end{cases}$$

Démonstration. La preuve est semblable à celle de la proposition 5.2.3. ■

Notations 5.2.6. En utilisant les notations de la proposition 5.2.4, notons

1. $A_{p-1,j}^+ = 1 + A_{p-1,j}$, 2. $A_{p-1,j}^- = 1 - A_{p-1,j}$, 3. $B_{p-1,j}^- = 1 - B_{p-1,j}$, 4. $B_{p-1,j}^+ = 1 + B_{p-1,j}$,
5. $H_{p-1,j}^1 = A_{p-1,j}^+ A_{p-1,j}^- + B_{p-1,j}^+ B_{p-1,j}^-$, 6. $H_{p-1,j}^2 = -A_{p-1,j} B_{p-1,j}$.



Dans le premier niveau, le premier DCD résout son problème par rapport aux contraintes initiales uniquement, sans aucune variable réduite. Nous définissons l'ARI dans ce niveau comme suit : $\xi_0(x_{i,j}) = x_{i,j}$.

Définition 5.2.7 (Application de Réduction des Intervalles). Soit $i = 2, \dots, P$, $j = 1, \dots, n_i$ et $x_{i,j} \in [l_{i,j}^{(p-1)}, u_{i,j}^{(p-1)}]$. Définissons

$$\xi_{p-1}(x_{i,j}) = \begin{cases} H_{p-1,j}^1 \hat{\psi}_{i,j}(x_{i,j}) + H_{p-1,j}^2 \hat{\nu}_{i,j}(x_{i,j}) & \text{si } i = p-1 \\ x_{i,j} & \text{si } i \neq p-1. \end{cases} \quad (5.10)$$

Proposition 5.2.8. L'application ξ_{p-1} ajustera les intervalles de la variable de décision $x_{i,j}$ en prenant en compte toutes les modifications effectuées précédemment.

Démonstration. Soit $i = 2, \dots, P$, $j = 1, \dots, n_i$ et $x_{i,j} \in [l_{i,j}^{(p-1)}, u_{i,j}^{(p-1)}]$, alors

Preuve du premier cas : Lorsque $l_{p-1,j}^{(p-1)} < x_{p-1,j}^{(p-1)} < u_{p-1,j}^{(p-1)}$, on aura

$$\begin{cases} A_{p-1,j} = 1, \\ B_{p-1,j} = -1 \end{cases} \Rightarrow \begin{cases} A_{p-1,j}^+ = 1 + A_{p-1,j} = 2, & A_{p-1,j}^- = 1 - A_{p-1,j} = 0, \\ B_{p-1,j}^+ = 1 + B_{p-1,j} = 0, & B_{p-1,j}^- = 1 - B_{p-1,j} = 2, \\ H_{p-1,j}^1 = A_{p-1,j}^+ A_{p-1,j}^- + B_{p-1,j}^+ B_{p-1,j}^- = 2 \times 0 + 0 \times 2 = 0, \\ H_{p-1,j}^2 = -A_{p-1,j} B_{p-1,j} = -1 \times -1 = 1. \end{cases}$$

Alors, nous obtenons

$$\xi_{p-1}(x_{i,j}) = \begin{cases} 0 \times \hat{\psi}_{i,j}(x_{i,j}) + \hat{\nu}_{i,j}(x_{i,j}) = \hat{\nu}_{i,j}(x_{i,j}) & \text{si } i = p-1, \\ x_{i,j} & \text{si } i \neq p-1. \end{cases}$$

Preuve du second cas : Lorsque $x_{p-1,j}^{(p-1)} = u_{p-1,j}^{(p-1)}$, on aura

$$\begin{cases} A_{p-1,j} = 0, \\ B_{p-1,j} = 1 \end{cases} \Rightarrow \begin{cases} A_{p-1,j}^+ = 1 + A_{p-1,j} = 1, & A_{p-1,j}^- = 1 - A_{p-1,j} = 1, \\ B_{p-1,j}^+ = 1 + B_{p-1,j} = 2, & B_{p-1,j}^- = 1 - B_{p-1,j} = 0, \\ H_{p-1,j}^1 = A_{p-1,j}^+ A_{p-1,j}^- + B_{p-1,j}^+ B_{p-1,j}^- = 1 \times 1 + 2 \times 0 = 1, \\ H_{p-1,j}^2 = -A_{p-1,j} B_{p-1,j} = -0 \times 1 = 0. \end{cases}$$

Alors, nous obtenons

$$\xi_{p-1}(x_{i,j}) = \begin{cases} \hat{\psi}_{i,j}(x_{i,j}) + 0 \times \hat{\nu}_{i,j}(x_{i,j}) = \hat{\psi}_{i,j}(x_{i,j}) & \text{si } i = p-1, \\ x_{i,j} & \text{si } i \neq p-1. \end{cases}$$



Preuve du troisième cas : Lorsque $\hat{x}_{p-1,j}^{(p-1)} = l_{p-1,j}^{(p-1)}$, on aura

$$\boxed{\begin{matrix} A_{p-1,j} = 1, \\ B_{p-1,j} = 0 \end{matrix}} \Rightarrow \begin{cases} A_{p-1,j}^+ = 1 + A_{p-1,j} = 2, & A_{p-1,j}^- = 1 - A_{p-1,j} = 0, \\ B_{p-1,j}^+ = 1 + B_{p-1,j} = 1, & B_{p-1,j}^- = 1 - B_{p-1,j} = 1, \\ H_{p-1,j}^1 = A_{p-1,j}^+ A_{p-1,j}^- + B_{p-1,j}^+ B_{p-1,j}^- = 2 \times 0 + 1 \times 1 = 1, \\ H_{p-1,j}^2 = -A_{p-1,j} B_{p-1,j} = -1 \times 0 = 0. \end{cases}$$

Alors, nous obtenons

$$\xi_{p-1}(x_{i,j}) = \begin{cases} \hat{\psi}_{i,j}(x_{i,j}) + 0 \times \hat{\nu}_{i,j}(x_{i,j}) = \hat{\psi}_{i,j}(x_{i,j}) & \text{si } i = p - 1, \\ x_{i,j} & \text{si } i \neq p - 1. \end{cases}$$

■



L'application ξ_{p-1} réduira les intervalles des composantes $x_{p-1,j}$ pour tout $j = 1, \dots, n_p$ uniquement.

Notations 5.2.9. On note par $l^{(p)}$ et $u^{(p)}$ les vecteurs suivants :

$$l^{(p)} := \xi_{p-1}(l^{(p-1)}) = \left(\xi_{p-1} \left(l_{i,j}^{(p-1)} \right), i = 1, \dots, P, j = 1, \dots, n_i \right), \quad (5.11)$$

$$u^{(p)} := \xi_{p-1}(u^{(p-1)}) = \left(\xi_{p-1} \left(u_{i,j}^{(p-1)} \right), i = 1, \dots, P, j = 1, \dots, n_i \right). \quad (5.12)$$

Définition 5.2.10. La formulation linéaire du $p^{\text{ième}}$ -niveau est donné comme suit :

$$\begin{cases} \max_x f_p(x) = c_p^t x \\ Ax \leq b \\ \xi_{p-1} \left(l^{(p-1)} \right) \leq x \leq \xi_{p-1} \left(u^{(p-1)} \right). \end{cases} \quad (5.13)$$

5.2.2 Estimation de Sous-Optimalité pour le $p^{\text{ième}}$ Niveau

Pour passer d'une indexation par deux indices à une indexation par un seul indice, nous introduisons la définition suivante :

Définition 5.2.11 (Application de Changement d'Indices). Soit $I = 1, \dots, n$ et $I_1 = 1, \dots, n_1$, ainsi que $i = 2, \dots, P$. Nous définissons les ensembles I_i comme suit :

$$I_i = \left\{ \sum_{t=1}^{i-1} n_t + j \quad : \quad j = 1, \dots, n_i \right\}.$$

Ensuite, pour tout $i = 1, \dots, P$, nous introduisons l'application $e : I_i \rightarrow I$ telle que :

$$e : I_i \rightarrow I$$

$$j \rightarrow e_{i,j} := e(i, j) = \begin{cases} \sum_{t=1}^{i-1} n_t + j, & \text{si } i > 1, \\ j, & \text{si } i = 1. \end{cases}$$

Définition 5.2.12. Soit $\{x^p, J_B^p\}$ une solution réalisable du problème (5.13), et définissons :

$$\begin{aligned} \beta_p^\xi(x^p, J_B^p) = & \sum_{E_{e_{i,j}}^p > 0, e_{i,j} \in J_N^p} E_{e_{i,j}}^p \left(\xi_{p-1}(x_{e_{i,j}}^p) - \xi_{p-1}(l_{e_{i,j}}^{(p-1)}) \right) \\ & + \sum_{E_{e_{i,j}}^p < 0, e_{i,j} \in J_N^p} E_{e_{i,j}}^p \left(\xi_{p-1}(x_{e_{i,j}}^p) - \xi_{p-1}(u_{e_{i,j}}^{(p-1)}) \right). \end{aligned} \quad (5.14)$$

$\beta_p^\xi(x^p, J_B^p)$ est l'estimation de sous-optimalité pour le $p^{\text{ième}}$ -niveau.

Proposition 5.2.13 (Condition Suffisante de Sous-optimalité pour le $p^{\text{ième}}$ -Niveau). Soit $\{x^p, J_B^p\}$ une solution de base réalisable du problème (5.13) et ϵ_p un nombre positif arbitraire. Si $\beta_p^\xi(x^p, J_B^p) < \epsilon_p$, alors la solution réalisable x^p est ϵ_p -optimale.

Démonstration. Voir [Gab95]. ■

Définition 5.2.14. La solution optimale \hat{x}^P du problème (5.13) est appelée compromis satisfaisant du programme linéaire à plusieurs niveaux (5.1).

Proposition 5.2.15 (Condition Suffisante pour un Compromis Satisfaisant d'un PPL à Plusieurs Niveaux). Soit $\{\hat{x}^P, J_B^P\}$ une solution de base réalisable du problème (5.13) et ϵ_P un nombre positif arbitraire. Si $\beta_P^\xi(\hat{x}^P, J_B^P) < \epsilon_P$, alors, la solution réalisable \hat{x}^P est un compromis ϵ_P -optimal pour le problème multi-niveaux (5.1).

Démonstration. Le compromis satisfaisant du problème (5.1) est obtenu comme solution du problème de programmation linéaire à variables bornées (5.13) en utilisant la méthode adaptée à la $(P - 1)^{\text{ième}}$ itération, voir [Gab95]. ■



5.3 Résolution du PPL(MO)O-MN (5.1) avec la Méthode Adaptée de Gabasov

Le déroulement de l'algorithme s'articule selon le schéma détaillée ci-dessous :

1. Calculer la solution de tous les objectifs indépendamment.
2. Déterminer les bornes de toutes les variables de décision.
3. Commencer la recherche d'un compromis en résolvant $P - 1$ problèmes de programmation linéaire standard à variables bornées à l'aide de l'algorithme 2 de la méthode adaptée.



La différence entre les deux versions réside dans l'utilisation de la nouvelle estimation de sous-optimalité (5.14) dans la seconde version, contrairement à la première.

Considérons le problème de programmation linéaire suivant :

$$\begin{cases} \max_x f_p(x) = c_p^t x \\ Ax \leq b \\ l^{(p)} \leq x \leq u^{(p)}. \end{cases} \quad (5.15)$$

5.3.1 Algorithme - Première Version

Algorithme 4 Résolution du PPL(MO)O-MN (5.1) - Première Version

Étape 1 : Poser $p = 1$ |

tant que $p \leq P$ **faire**

1. Expliciter le modèle (5.2) |

2. Obtenir la solution \hat{x}^p du modèle (5.2) "solution du $p^{\text{ième}}$ -niveau" |

3. Poser $p \leftarrow p + 1$ |

Étape 2 : Obtenir toutes les bornes $l_{e_{i,j}}^{(1)}$ et $u_{e_{i,j}}^{(1)}$ en utilisant la formule (5.3) |

Étape 3 : Poser $p \leftarrow 2$ |

tant que $p \leq P$ **faire**

1. Expliciter le modèle (5.15) |

2. Résoudre le modèle (5.15) en utilisant l'algorithme 2 de la méthode adaptée mono-critère pour obtenir la solution optimale \hat{x}^p |

3. Poser $p \leftarrow p + 1$ |

Sortie : La solution satisfaisante \hat{x}^P pour le problème multi-niveaux (5.1) |

5.3.2 Illustration Numérique de la 1^{er}-Version de l'Algorithme

Énoncé

Considérons le problème de programmation linéaire à trois niveaux suivant :

$$\begin{aligned}
 \text{Niveau 1 : } & \underset{x_1, x_2}{\text{Max}} f_1(x) = -5x_1 + x_2 + 2x_3 + 3x_4 \\
 & \text{tel que } x_3, x_4 \text{ résolvent} \\
 \text{Niveau 2 : } & \underset{x_3}{\text{Max}} f_2(x) = 6x_1 + 2x_2 - 3x_3 + x_4 \quad (5.16) \\
 & \text{tel que } x_4 \text{ résout} \\
 \text{Niveau 3 : } & \underset{x_4}{\text{Max}} f_3(x) = -x_2 + 2x_3 + 3x_4,
 \end{aligned}$$

soumis à :

$$x \in S = \{x \in \mathbb{R}^4 : Ax \leq b, x \geq 0, b \in \mathbb{R}^7\}.$$

Où

$$A = \begin{pmatrix} 3 & 2 & 1 & 2 \\ 1 & 3 & -2 & -1 \\ -2 & -5 & -2 & 0 \\ -1 & 4 & 1 & 0 \\ 1 & -1 & 1 & 2 \\ 1 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 6 \\ 3 \\ -2 \\ 2 \\ 5 \\ 4 \\ 2 \end{pmatrix}.$$

◇ Nous obtenons les solutions optimales suivantes :

$$f_1 = 6 \quad \text{en} \quad \overset{o^1}{x} = \left(0 \ 0 \ 2 \ \frac{2}{3} \right)^t,$$

$$f_2 = 12 \quad \text{en} \quad \overset{o^2}{x} = \left(2 \ 0 \ 0 \ 0 \right)^t,$$

$$f_3 = 6 \quad \text{en} \quad \overset{o^3}{x} = \left(1 \ 0 \ 3 \ 0 \right)^t.$$

◇ Ensuite, les bornes des variables de décision sont :

$$l_{11}^{(1)} = 0, \quad u_{11}^{(1)} = 2, \quad l_{31}^{(1)} = 0,$$

$$l_{12}^{(1)} = 0, \quad u_{12}^{(1)} = 0, \quad u_{31}^{(1)} = \frac{2}{3},$$

$$l_{21}^{(1)} = 0, \quad u_{21}^{(1)} = 3.$$

Itération 1 : Nous choisissons les valeurs suivantes : $\alpha_{11} = 0.25, \alpha_{12} = \alpha_{21} = \alpha_{31} = 0$ et nous avons :

$$\xi_1 \left(l_{11}^{(1)} \right) = U_{11}^{0.25} \left(l_{11}^{(1)} \right) = 0.25, \quad \xi_1 \left(u_{11}^{(1)} \right) = Id_{[0,2]} \left(u_{11}^{(1)} \right) = 2,$$

$$\xi_1 \left(l_{12}^{(1)} \right) = Id_{\{0\}} \left(l_{12}^{(1)} \right) = 0, \quad \xi_1 \left(u_{12}^{(1)} \right) = Id_{\{0\}} \left(u_{12}^{(1)} \right) = 0,$$

$$\xi_1 \left(l_{21}^{(1)} \right) = Id_{[0,3]} \left(l_{21}^{(1)} \right) = 0, \quad \xi_1 \left(u_{21}^{(1)} \right) = Id_{[0,3]} \left(u_{21}^{(1)} \right) = 3,$$

$$\xi_1 \left(l_{31}^{(1)} \right) = Id_{\left[0, \frac{2}{3}\right]} \left(l_{31}^{(1)} \right) = 0, \quad \xi_1 \left(u_{31}^{(1)} \right) = Id_{\left[0, \frac{2}{3}\right]} \left(u_{31}^{(1)} \right) = \frac{2}{3}.$$



Nous définissons ensuite les bornes pour cette itération :

$$l^{(2)} = \left(0.25 \ 0 \ 0 \ 0 \right)^t \quad \text{et} \quad u^{(2)} = \left(2 \ 0 \ 3 \ \frac{2}{3} \right)^t.$$

Ensuite, nous considérons le problème de programmation linéaire suivant :

$$\begin{cases} \max_x f_2(x) = 6x_1 + 2x_2 - 3x_3 + x_4 \\ Ax \leq b \\ l^{(2)} \leq x \leq u^{(2)}. \end{cases} \quad (5.17)$$

En utilisant l'algorithme de la méthode adaptée, nous obtenons la solution optimale du problème linéaire (5.17) :

$$\bar{x}^2 = \left(2 \ 0 \ 0 \ 0 \right)^t.$$

Itération 2 : Nous choisissons les valeurs suivantes : $\alpha_{21} = 0.25$, $\alpha_{11} = \alpha_{12} = \alpha_{31} = 0$, et nous avons :

$$\begin{aligned} l_{11}^{(3)} &= \xi_2 \left(l_{11}^{(2)} \right) = Id_{[0.25,2]} \left(l_{11}^{(2)} \right) = 0.25, & u_{11}^{(3)} &= \xi_2 \left(u_{11}^{(2)} \right) = Id_{[0.25,2]} \left(u_{11}^{(2)} \right) = 2, \\ l_{12}^{(3)} &= \xi_2 \left(l_{12}^{(2)} \right) = Id_{\{0\}} \left(l_{12}^{(2)} \right) = 0, & u_{12}^{(3)} &= \xi_2 \left(u_{12}^{(2)} \right) = Id_{\{0\}} \left(u_{12}^{(2)} \right) = 0, \\ l_{21}^{(3)} &= \xi_2 \left(l_{21}^{(2)} \right) = U_{21}^{0.25} \left(l_{21}^{(2)} \right) = 0.25, & u_{21}^{(3)} &= \xi_2 \left(u_{21}^{(2)} \right) = Id_{[0,3]} \left(u_{21}^{(2)} \right) = 3, \\ l_{31}^{(3)} &= \xi_2 \left(l_{31}^{(2)} \right) = Id_{[0, \frac{2}{3}]} \left(l_{31}^{(2)} \right) = 0 & u_{31}^{(3)} &= \xi_2 \left(u_{31}^{(2)} \right) = Id_{[0, \frac{2}{3}]} \left(u_{31}^{(2)} \right) = \frac{2}{3}. \end{aligned}$$

Nous définissons ensuite les bornes pour cette itération :

$$l^{(3)} = \left(0.25 \ 0 \ 0.25 \ 0 \right)^t \quad \text{et} \quad u^{(3)} = \left(2 \ 0 \ 3 \ \frac{2}{3} \right)^t,$$

◇ Ensuite, nous considérons le problème de programmation linéaire suivant :

$$\begin{cases} \max_x f_3(x) = -x_2 + 2x_3 + 3x_4 \\ Ax \leq b \\ l^{(3)} \leq x \leq u^{(3)}. \end{cases} \quad (5.18)$$

Finalement, en appliquant la méthode adaptée de Gabasov 2, nous parvenons à obtenir le compromis satisfaisant du PPL à trois niveaux (5.16) en tant que solution du problème (5.18), avec la valeur

$$\bar{x}^3 = \left(0.2500 \ 0 \ 2.7671 \ 0 \right)^t.$$

Alors, on obtient :

$$f_1 \left(\bar{x}^3 \right) = 4.2841, \quad f_2 \left(\bar{x}^3 \right) = -6.8012, \quad f_3 \left(\bar{x}^3 \right) = 5.5341.$$

5.3.3 Algorithme - Deuxième Version

Algorithme 5 Résolution du PPL(MO)O-MN (5.1)-Deuxième Version-**Partie 1**

Étape 1 : Poser $p \leftarrow 1$ |

tant que $p \leq P$ **faire**

1. Expliciter le modèle (5.2) |
2. Obtenir la solution \hat{x}^p du modèle (5.2) "solution de $p^{\text{ième}}$ -niveau" |
3. Poser $p \leftarrow p + 1$ |

Étape 2 : Obtenir toutes les bornes $l_{e_{i,j}}^{(1)}$ et $u_{e_{i,j}}^{(1)}$ en utilisant la formule (5.3) |

Étape 3 : Poser $p \leftarrow 2$ |

tant que $p \leq P$ **faire**

1. Calculer $l^{(p)} = \xi_{p-1}(l^{(p-1)})$ et $u^{(p)} = \xi_{p-1}(u^{(p-1)})$ |
2. Expliciter le modèle (5.15) |
3. Résoudre le modèle (5.15) avec la méthode adaptée mono-critère suivante :
 - (a) Choisir un réel $\epsilon_p > 0$ |
 - (b) Obtenir une solution de base réalisable $\{x^p, J_B^p\}$ |
 - (c) Calculer $K_B^{pt} = c_{pB}^t A_B^{-1}$; $E_{e_{i,j}}^{pt} = K^{pt} a_{e_{i,j}} - c_p^{e_{i,j}}$, $e_{i,j} \in J_N^p$ |
 - (d) Calculer $\beta_p^\xi(x^p, J_B^p)$ avec la formule (5.14) |
 - (e) Si $\beta_p^\xi(x^p, J_B^p) = 0$. Alors, arrêter la procédure avec la solution optimale $\{x^p, J_B^p\}$ et aller à l'étape 4. |
 - (f) Si $\beta_p^\xi(x^p, J_B^p) \leq \epsilon_p$. Alors, arrêter la procédure avec la solution ϵ -optimale $\{x^p, J_B^p\}$ et aller à l'étape 4. |

(g) Poser

$$J_{NNO_1}^p = \{e_{i,j} \in J_N^p : [E_{e_{i,j}}^p < 0, \xi_{p-1}(x_{e_{i,j}}^p) < u_{e_{i,j}}^{(p)}]\},$$

$$J_{NNO_2}^p = \{e_{i,j} \in J_N^p : [E_{e_{i,j}}^p > 0, \xi_{p-1}(x_{e_{i,j}}^p) > l_{e_{i,j}}^{(p)}]\}$$

(h) Obtenir l'ensemble des indices non-optimaux : $J_{NNO}^p = J_{NNO_1}^p \cup J_{NNO_2}^p$ |

(i) Choisir un indice $e_{i_0j_0}$ de J_{NNO}^p tel que : $|E_{e_{i_0j_0}}^p| = \max_{e_{i,j} \in J_{NNO}^p} |E_{e_{i,j}}^p|$ |

(j) Calculer la direction d'amélioration d^p en utilisant les relations :

$$\begin{cases} d_{e_{i_0j_0}}^p &= -\text{sign}(E_{e_{i_0j_0}}^p) \\ d_{e_{i,j}}^p &= 0 & \text{si } e_{i,j} \neq e_{i_0j_0}, e_{i,j} \in J_N^p. \\ d^p(J_B^p) &= -A_B^{-1} A_N d^p(J_N^p) \end{cases} \quad (5.19)$$



Suite de l'algorithme 5 - Partie 2

(k) Calculer $\theta_{e_{i_1j_1}}^p = \min_{e_{i,j} \in J_B^p} \theta_{e_{i,j}}^p$, où $\theta_{e_{i,j}}^p$ est déterminé par la formule :

$$\theta_{e_{i,j}}^p = \begin{cases} \frac{u_{e_{i,j}}^{(p)} - \xi_{p-1}(x_{e_{i,j}}^p)}{d_{e_{i,j}}^p}, & \text{si } d_{e_{i,j}}^p > 0 \\ \frac{l_{e_{i,j}}^{(p)} - \xi_{p-1}(x_{e_{i,j}}^p)}{d_{e_{i,j}}^p}, & \text{si } d_{e_{i,j}}^p < 0 \\ \infty, & \text{si } d_{e_{i,j}}^p = 0. \end{cases} \quad (5.20)$$

(l) Calculer $\theta_{e_{i_0j_0}}^p$ en utilisant la formule :

$$\theta_{e_{i_0j_0}}^p = \begin{cases} \xi_{p-1}(x_{e_{i_0j_0}}^p) - l_{e_{i_0j_0}}^{(p)}, & \text{si } E_{e_{i_0j_0}}^p > 0 \\ u_{e_{i_0j_0}}^{(p)} - \xi_{p-1}(x_{e_{i_0j_0}}^p), & \text{si } E_{e_{i_0j_0}}^p < 0. \end{cases} \quad (5.21)$$

(m) Calculer $\theta_0^p = \min \{ \theta_{e_{i_1j_1}}^p, \theta_{e_{i_0j_0}}^p \}$ |

(n) Calculer $\bar{x}^p = x^p + \theta_0^p d^p$ |

(o) Calculer $\beta_p^\xi(\bar{x}^p, J_B^p) = \beta_p^\xi(x^p, J_B^p) - \theta_0^p |E_{e_{i_0j_0}}^p|$ |

(p) Si $\beta_p^\xi(\bar{x}^p, J_B^p) = 0$. Alors, arrêter la procédure avec la solution satisfaisante $\{\bar{x}^p, J_B^p\}$ et aller à l'étape 4. |

(q) Si $\beta_p^\xi(\bar{x}^p, J_B^p) \leq \epsilon_p$. Alors, arrêter la procédure avec la solution ϵ_p -optimale $\{\bar{x}^p, J_B^p\}$ et aller à l'étape 4. |

(r) Si $\theta_0^p = \theta_{e_{i_0j_0}}^p$, alors poser $\bar{J}_B^p = J_B^p$ |

(s) Si $\theta_0^p = \theta_{e_{i_1j_1}}^p$, alors poser $\bar{J}_B^p = (J_B^p \setminus \{e_{i_1j_1}\}) \cup \{e_{i_0j_0}\}$ |

(t) Poser $x^p = \bar{x}^p$, et $J_B^p = \bar{J}_B^p$. Ensuite aller à l'étape (c) |

4 Poser $\overset{c}{x} = x^p$ |

5 Poser $p \leftarrow p + 1$ |

Sortie Retourner le compromis $\overset{c}{x}^P$ du PPL(MO)O-MN (5.1) |

5.3.3.1 Explication de l'algorithme

L'algorithme peut être décrit en plusieurs étapes clés, mettant en œuvre une approche itérative pour obtenir le compromis $\overset{c}{x}^P$ du PPL(MO)O-MN :

Étape 1 : Pour chaque niveau p , calculer les solutions optimales des fonctions objectives individuelles du problème 5.1, de manière indépendante.

Étape 2 : Visualiser les bornes initiales du vecteur des variables de décision.

Étape 3 : L'algorithme itère jusqu'à atteindre une solution satisfaisante \bar{x}^P en $P - 1$ étapes. À chaque itération, le DCD_p (le décideur du niveau p dans le PPL(MO)O-MN) sélectionne les intervalles réduits pour les variables de décision sous son contrôle en utilisant l'ARI. Ensuite, le DCD_p résout son propre PPL mono-objectif en utilisant l'algorithme 2 de la méthode adaptée exposée dans le chapitre 4 et décrite en détail dans [Gab95].



La distinction entre les deux versions de l'algorithme réside dans l'étape 3. de la deuxième boucle "tant que". Dans la première version, l'opérateur ξ_{p-1} est appliqué uniquement aux bornes des variables de décision $l^{(p-1)}$ et $u^{(p-1)}$. Par la suite, l'algorithme de la méthode adaptée est exécuté en utilisant l'ancienne estimation de sous-optimalité. En contraste, dans la deuxième version, la nouvelle estimation de sous-optimalité est utilisée. Par conséquent, l'étape 3. de la deuxième boucle "tant que" englobe l'intégralité de l'algorithme 2 de la méthode adaptée, afin de démontrer comment la nouvelle estimation est exploitée.

5.3.4 Illustration Numérique de la 2^{ème}-Version de l'Algorithme

Énoncé

Considérons le problème de programmation linéaire à trois niveaux suivant :

$$\begin{aligned}
 \text{Niveau 1 : } & \underset{x_1}{\text{Max}} f_1(x) = -7x_1 + 2x_2 - 4x_3 + x_4 \\
 & \text{tel que } x_2, x_3, x_4 \text{ résolvent} \\
 \text{Niveau 2 : } & \underset{x_2, x_3}{\text{Max}} f_2(x) = -4x_1 - x_2 + 2x_3 + 3x_4 \quad (5.22) \\
 & \text{tel que } x_4 \text{ résout} \\
 \text{Niveau 3 : } & \underset{x_4}{\text{Max}} f_3(x) = -10x_1 + x_2 + 2x_3 - 3x_4,
 \end{aligned}$$

soumis à :

$$x \in S = \{x \in \mathbb{R}^4 : Ax = b, x \geq 0\} \neq \emptyset.$$

Où

$$A = \begin{pmatrix} 6 & 4 & 2 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & -4 & -2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & -10 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 8 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & -2 & 2 & 4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 2 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 13 \\ 8 \\ -5 \\ 4 \\ 6 \\ 7 \\ 9 \end{pmatrix}.$$



Nous obtenons les solutions optimales :

$$\begin{aligned} f_1 = -15.1669 & \text{ en } \overset{o}{x}^1 = \left(2.16 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 3.66 \ 3.66 \ 8.33 \ 1.66 \ 2.66 \ 9 \right)^t, \\ f_2 = 5.5 & \text{ en } \overset{o}{x}^2 = \left(0 \ 0 \ 2 \ 0.5 \ 7 \ 17 \ 3 \ 0 \ 0 \ 0 \ 7 \right)^t, \\ f_3 = 4 & \text{ en } \overset{o}{x}^3 = \left(0 \ 0 \ 2 \ 0 \ 9 \ 16 \ 3 \ 0 \ 2 \ 3 \ 9 \right)^t. \end{aligned}$$

Soit $M = 1000$. Alors, les bornes des variables de décision sont définies comme suit : $l_{4,j}^{(1)} = 0$, et $u_{4,j}^{(1)} = M$ pour tous $j = 1, \dots, 7$. Quant aux autres composantes, les bornes sont données par :

$$\begin{aligned} l_{11}^{(1)} = 0, \quad l_{21}^{(1)} = 0, \quad l_{22}^{(1)} = 0, \quad l_{31}^{(1)} = 0, \\ u_{11}^{(1)} = 2.1667, \quad u_{21}^{(1)} = 0, \quad u_{22}^{(1)} = 2, \quad u_{31}^{(1)} = 0.5. \end{aligned}$$

Première Itération de l'algorithme 5 : On prend $\alpha_{11} = 0.25$, $\alpha_{21} = \alpha_{22} = \alpha_{31} = \alpha_{4j} = 0$ pour tous $j = 1, \dots, 7$. Ensuite, pour $j = 1, \dots, 7$, on calcule les bornes réduites pour chaque composante des vecteurs $l^{(1)}$ et $u^{(1)}$ en utilisant l'ARI ξ_1 :

$$\begin{aligned} l_{11}^{(2)} = \xi_1 \left(l_{11}^{(1)} \right) = Id_{[0,2.1667]} \left(l_{11}^{(1)} \right) = 0, \quad l_{22}^{(2)} = \xi_1 \left(l_{22}^{(1)} \right) = Id_{[0,2]} \left(l_{22}^{(1)} \right) = 0, \\ u_{11}^{(2)} = \xi_1 \left(u_{11}^{(1)} \right) = L_{11}^{0.25} \left(u_{11}^{(1)} \right) = 1.9167, \quad u_{22}^{(2)} = \xi_1 \left(u_{22}^{(1)} \right) = Id_{[0,2]} \left(u_{22}^{(1)} \right) = 2, \\ l_{21}^{(2)} = \xi_1 \left(l_{21}^{(1)} \right) = Id_{\{0\}} \left(l_{21}^{(1)} \right) = 0, \quad l_{31}^{(2)} = \xi_1 \left(l_{31}^{(1)} \right) = Id_{[0,0.5]} \left(l_{31}^{(1)} \right) = 0, \\ u_{21}^{(2)} = \xi_1 \left(u_{21}^{(1)} \right) = Id_{\{0\}} \left(u_{21}^{(1)} \right) = 0, \quad u_{31}^{(2)} = \xi_1 \left(u_{31}^{(1)} \right) = Id_{[0,0.5]} \left(u_{31}^{(1)} \right) = 0.5, \\ l_{4j}^{(2)} = \xi_1 \left(l_{4j}^{(1)} \right) = Id_{[0,1000]} \left(l_{4j}^{(1)} \right) = 0, \quad u_{4j}^{(2)} = \xi_1 \left(u_{4j}^{(1)} \right) = Id_{[0,1000]} \left(u_{4j}^{(1)} \right) = 1000. \end{aligned}$$

Nous obtenons les bornes suivantes :

$$\begin{aligned} l^{(2)} &= \left(0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^t, \\ u^{(2)} &= \left(1.9167 \ 0 \ 2 \ 0.5 \ M \ M \ M \ M \ M \ M \ M \right)^t. \end{aligned}$$

Ensuite, nous considérons le problème de programmation linéaire suivant :

$$\begin{cases} \max_x f_2(x) = -4x_1 - x_2 + 2x_3 + 3x_4 \\ Ax = b \\ l^{(2)} \leq x \leq u^{(2)}. \end{cases} \quad (5.23)$$

Alors, l'algorithme de la méthode adaptée peut être appliqué pour résoudre le problème (5.23) en utilisant l'estimation de la sous-optimalité (5.14).

Première Itération de la Méthode Adaptée :

- ◇ La solution réalisable initiale : $x^2 = \left(1.25 \ 0 \ 0 \ 0 \ 5.5 \ 5.5 \ 0 \ 6.5 \ 3.5 \ 4.5 \ 9 \right)^t$.
- ◇ Le support initial : $J_B^2 = \{5, 6, 7, 8, 9, 10, 11\}$ et $J_N^2 = \{1, 2, 3, 4\}$.
- ◇ Le vecteur des multiplicateurs : $E^2 = \left(4 \ 1 \ -2 \ -3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^t$.
- ◇ L'estimation de sous-optimalité :

$$\begin{aligned} \beta_2^\xi(x_2, J_B^2) = & E_{11}^2 \left(\xi_1 \left(u_{11}^{(1)} \right) - \xi_1 \left(x_{11}^2 \right) \right) + E_{21}^2 \left(\xi_1 \left(u_{21}^{(1)} \right) - \xi_1 \left(x_{21}^2 \right) \right) \\ & + E_{22}^2 \left(\xi_1 \left(l_{22}^{(1)} \right) - \xi_1 \left(x_{22}^2 \right) \right) + E_{31}^2 \left(\xi_1 \left(l_{31}^{(1)} \right) - \xi_1 \left(x_{31}^2 \right) \right) = 10.5 > 0. \end{aligned}$$

Alors, La solution réalisable initiale x^2 n'est pas optimale.

- ◇ La nouvelle direction d'amélioration :

$$d^2 = \left(-1.25 \ 0 \ 2 \ 0.5 \ 1.5 \ 11.5 \ 3 \ -6.5 \ -3.5 \ -4.5 \ -2 \right)^t.$$

- ◇ Le pas optimale : $\theta_0^2 = 1$. Alors, $\hat{x}^2 = x^2 + \theta_0^2 d^2 = \left(0 \ 0 \ 2 \ 0.5 \ 7 \ 17 \ 3 \ 0 \ 0 \ 0 \ 7 \right)^t$ est la solution optimal du problème (5.23).

Deuxième Itération de l'Algorithme 5 : On prend $\alpha_{22} = 0.25$, $\alpha_{11} = \alpha_{21} = \alpha_{31} = \alpha_{4j} = 0$. Ensuite, pour $j = 1, \dots, 7$, on calcule les bornes réduites pour chaque composante des vecteurs $l^{(2)}$ et $u^{(2)}$ en utilisant l'ARI ξ_2 :

$$\begin{aligned} l_{11}^{(3)} = \xi_2 \left(l_{11}^{(2)} \right) &= Id_{[0, 2.1667]} \left(l_{11}^{(2)} \right) = 0, & l_{22}^{(3)} = \xi_2 \left(l_{22}^{(2)} \right) &= Id_{[0, 2]} \left(l_{22}^{(2)} \right) = 0, \\ u_{11}^{(3)} = \xi_2 \left(u_{11}^{(2)} \right) &= L_{11}^{0.25} \left(u_{11}^{(2)} \right) = 1.9167, & u_{22}^{(3)} = \xi_2 \left(u_{22}^{(2)} \right) &= Id_{[0, 2]} \left(u_{22}^{(2)} \right) = 1.75, \\ l_{21}^{(3)} = \xi_2 \left(l_{21}^{(2)} \right) &= Id_{\{0\}} \left(l_{21}^{(2)} \right) = 0, & l_{31}^{(3)} = \xi_2 \left(l_{31}^{(2)} \right) &= Id_{[0, 0.5]} \left(l_{31}^{(2)} \right) = 0, \\ u_{21}^{(3)} = \xi_2 \left(u_{21}^{(2)} \right) &= Id_{\{0\}} \left(u_{21}^{(2)} \right) = 0, & u_{31}^{(3)} = \xi_2 \left(u_{31}^{(2)} \right) &= Id_{[0, 0.5]} \left(u_{31}^{(2)} \right) = 0.5, \\ l_{4j}^{(3)} = \xi_2 \left(l_{4j}^{(2)} \right) &= Id_{[0, 1000]} \left(l_{4j}^{(2)} \right) = 0, & u_{4j}^{(3)} = \xi_2 \left(u_{4j}^{(2)} \right) &= Id_{[0, 1000]} \left(u_{4j}^{(2)} \right) = 1000. \end{aligned}$$

Nous obtenons les bornes suivantes :

$$\begin{aligned} l^{(3)} &= \left(0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^t, \\ u^{(3)} &= \left(1.9167 \ 0, 1.75 \ 0.5 \ M \ M \ M \ M \ M \ M \ M \right)^t. \end{aligned}$$

Ensuite, nous considérons le problème de programmation linéaire suivant :

$$\begin{cases} \max_x f_3(x) = -10x_1 + x_2 + 2x_3 - 3x_4 \\ Ax = b \\ l^{(3)} \leq x \leq u^{(3)}. \end{cases} \quad (5.24)$$

Alors, l'algorithme de la méthode adaptée peut être appliqué pour résoudre le problème (5.24) en utilisant l'estimation de la sous-optimalité (5.14).



Première Itération de la Méthode Adaptée :

- ◊ La solution réalisable initiale : $x^3 = \left(1.25 \ 0 \ 0 \ 0 \ 5.5 \ 5.5 \ 0 \ 6.5 \ 3.5 \ 4.5 \ 9 \right)^t$.
- ◊ Le support initial est $J_B^3 = \{5, 6, 7, 8, 9, 10, 11\}$, $J_N^3 = \{1, 2, 3, 4\}$.
- ◊ Le vecteur des multiplicateurs : $E^3 = \left(10 \ -1 \ -2 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^t$.
- ◊ L'estimation de sous-optimalité :

$$\begin{aligned} \beta_3^\xi(x^3, J_B^3) &= E_{11}^3 \left(\xi_2 \left(u_{11}^{(2)} \right) - \xi_2 \left(x_{11}^3 \right) \right) + E_{21}^3 \left(\xi_2 \left(u_{21}^{(2)} \right) - \xi_2 \left(x_{21}^3 \right) \right) \\ &+ E_{22}^3 \left(\xi_2 \left(l_{22}^{(2)} \right) - \xi_2 \left(x_{22}^3 \right) \right) + E_{31}^3 \left(\xi_2 \left(l_{31}^{(2)} \right) - \xi_2 \left(x_{31}^3 \right) \right) = 10.5 > 0. \end{aligned}$$

Alors, la solution réalisable initiale x^3 n'est pas optimale.

- ◊ La direction d'amélioration : $d^3 = \left(-1.25 \ 0 \ 1.75 \ 0 \ 4 \ 9.5 \ 2 \ -6 \ -1 \ -1 \ 0 \right)^t$.
- ◊ Le pas optimal : $\theta_0^3 = 1$. Alors, $\bar{x}^3 = x^3 + \theta_0^3 d^3 = \left(0 \ 0 \ 1.75 \ 0 \ 9.5 \ 15 \ 2 \ 0.5 \ 2.5 \ 3.5 \ 9 \right)^t$ est la solution optimale au problème (5.24).

Enfin, le compromis satisfaisant du PPL à trois niveaux (5.22) : $\bar{x}^3 = \left(0 \ 0 \ 1.75 \ 0 \right)$ Alors, $f_1 \left(\bar{x}^3 \right) = -7$, $f_2 \left(\bar{x}^3 \right) = 3.5$, $f_3 \left(\bar{x}^3 \right) = 3.5$.

5.4 Conclusion

La méthode adaptée se distingue comme une méthode numérique de choix pour résoudre efficacement les problèmes de programmation linéaire mono-objectif à variables bornées. Cette méthode a démontré sa supériorité par rapport à la méthode du simplexe [Bel18].

Dans cette étude, deux objectifs ont été pleinement réalisés. Tout d'abord, nous avons pionnièrement appliqué l'algorithme de la méthode adaptée à la résolution de problèmes de programmation linéaire à plusieurs niveaux. Deuxièmement, nous avons sensiblement amélioré l'algorithme de S.B. Sinha et S. Sinha [Sin04] en introduisant la méthode adaptée à l'étape 3, en remplacement de la méthode du simplexe. De plus, nous avons réussi à consolider les étapes (3)-(9) en utilisant l'ARI ξ , ce qui nous a permis de les réunir en une seule étape (étape 3 dans notre algorithme).

Notre contribution repose sur l'exploitation ingénieuse des principes de la méthode adaptée pour élaborer et exposer en détail deux versions d'un algorithme résolvant les problèmes de programmation linéaire à plusieurs niveaux. En développant l'ARI, nous avons pu définir une nouvelle estimation sous-optimalité spécifique au PPL(MO)O-MN, ce qui nous a permis d'intégrer l'algorithme de la méthode adaptée dans notre algorithme. Enfin, nous avons illustré les deux versions de notre algorithme par le biais d'exemples numériques.



CHAPITRE 6

EXPLORATION DES SOLUTIONS DE COMPROMIS EN PROGRAMMATION LINÉAIRE MULTI-OBJECTIFS MULTI-NIVEAUX : LA MÉTHODE ADAPTÉE DE GABASOV

Objectifs

Dans ce chapitre, nous présentons deux contributions majeures dans le domaine de la PPMO-MN. La première de ces contributions, se matérialise par le biais d'une adaptation élaborée de la méthode du simplexe multi-critère généralisé. Cette adaptation soigneusement élaborée vise à explorer de manière exhaustive l'ensemble des solutions de compromis envisageables pour les Problèmes de Programmation Linéaire Multi-Objectifs Multi-Niveaux (PPLMO-MNs). En appréhendant ces compromis à travers le prisme du simplexe multi-critère généralisée, nous élevons l'analyse à un niveau supérieur de finesse, offrant ainsi une perspective enrichissante pour la prise de décisions éclairées. Dans notre seconde incursion, nous propulsons la méthode adaptée de Gabasov vers des hauteurs encore plus impressionnantes. En élargissant son spectre, nous sommes en mesure de pénétrer les problèmes multi-niveaux avec confiance et ingéniosité, ouvrant ainsi une porte prometteuse vers la résolution de défis mathématiques encore plus complexes.

Sommaire

6.1	Génération Complète des Solutions de Compromis pour un PPLMO-MN	112
-----	---	-----



6.2	Formulation du PPLMO-MN	113
6.3	Préliminaires	114
6.4	Exploration des Solutions de Compromis dans un PPLMO-MN	116
6.4.1	Identification des Faces Non-dominées d'un PPLMO-MN	117
6.4.2	Élaboration de l'Algorithme : Processus Complet de Construction	119
6.4.3	Exploration Exhaustive : L'Algorithme dans toute sa Splendeur	122
6.4.4	Première Illustration Numérique : Mettre en Action l'Algorithme 6	125
6.5	Résolution des PPLMO-MNs via une Approche Novatrice et Constructive	128
6.5.1	Édification de l'Algorithme : De la Conception à la Réalisation	128
6.5.2	Illustration Numérique du PPLMO à deux-niveaux (Section 6.4.4) : Étude de Cas à travers l'Algorithme 7	130
6.6	Dialogue Approfondi : Exploration et Analyses	131
6.6.1	La Méthode Adaptée vs La Méthode du Simplexe : Comparaison et Choix . .	131
6.6.2	Processus Étape par Étape pour Résoudre les PPLMO-MNs à Variables Bor- nées : La Méthode Adaptée	132
6.7	Conclusion	135

6.1 Génération Complète des Solutions de Compromis pour un PPLMO-MN

Résumé

Dans notre œuvre inauguraleⁱ, nous établissons l'ensemble de toutes les solutions réalisables non-dominées (ensemble de tous les compromis possibles) pour un PPLMO-MN. En démontrant sa décomposition en faces non-dominées, nous offrons la preuve que ce domaine peut être exploré avec finesse. À cet effet, nous élaborons une stratégie algorithmique afin d'engendrer l'ensemble des solutions réalisables non-dominées. Le mécanisme s'appuie sur la méthode énoncée par P.L. Yu et Zeleny, comme documenté dans [Yu 74; Yu 75]. Par la suite, nous définissons un ensemble de tri comme les faces constituant l'ensemble de solutions réalisables non-dominées, comme détaillé dans [Kac22c].

i. Cet article a été consigné au sein des pages du journal 'Croatian Operational Research Review', où il a été intitulé 'L'ensemble de tous les compromis possibles d'un problème linéaire multi-objectifs multi-niveaux'. Cette parution a vu le jour en 2022, et vous pouvez obtenir de plus amples informations en vous référant à [Kac22c].

Mots clés

Programmation linéaire multi-objectifs, décideur, point non-dominé, face non-dominée, point extrême non-dominé.



6.2 Formulation du PPLMO-MN

Soit $p = 1, \dots, P$, $n = n_1 + \dots + n_P$, $k = k_1 + \dots + k_P$. Considérons la fonction objectif suivante : $F_p(x) = c_p x : \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_P} \mapsto \mathbb{R}^{k_p}$. Où,

$$c_p = \begin{pmatrix} c_{p1} \\ \vdots \\ c_{pk_p} \end{pmatrix} = \begin{pmatrix} c_{p1}^{11} & \dots & c_{p1}^{1n_1} & \dots & c_{p1}^{P1} & \dots & c_{p1}^{Pn_P} \\ \vdots & & \vdots & & \vdots & & \vdots \\ c_{pk_p}^{11} & \dots & c_{pk_p}^{1n_1} & \dots & c_{pk_p}^{P1} & \dots & c_{pk_p}^{Pn_P} \end{pmatrix},$$

et

$$\begin{aligned} c_{pq}x &= c_{pq}^{1j}\bar{x}^1 + c_{pq}^{2j}\bar{x}^2 + \dots + c_{pq}^{Pj}\bar{x}^P = c_{pq}^{11}x_{11} + \dots + c_{pq}^{1n_1}x_{1n_1} + c_{pq}^{21}x_{21} + \dots + c_{pq}^{2n_2}x_{2n_2} \\ &+ c_{pq}^{31}x_{31} + \dots + \dots + c_{pq}^{P1}x_{P1} + \dots + c_{pq}^{Pn_P}x_{Pn_P}. \end{aligned}$$

Formulation d'un PPLMO à P niveaux

$$\begin{aligned} \text{Niveau 1 : } \quad \text{Max}_{\bar{x}^1} F_1(x) &= \text{Max}_{\bar{x}^1} F_1(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^P) = \text{Max}_{\bar{x}^1} \begin{pmatrix} c_{11}x \\ \vdots \\ c_{1k_1}x \end{pmatrix}, \\ &\text{tel que } \bar{x}^2, \dots, \bar{x}^P \text{ résout :} \\ \text{Niveau 2 : } \quad \text{Max}_{\bar{x}^2} F_2(x) &= \text{Max}_{\bar{x}^2} F_2(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^P) = \text{Max}_{\bar{x}^2} \begin{pmatrix} c_{21}x \\ \vdots \\ c_{2k_2}x \end{pmatrix}, \\ &\vdots \\ &\text{tel que } \bar{x}^P \text{ résout :} \\ \text{Niveau } P : \quad \text{Max}_{\bar{x}^P} F_P(x) &= \text{Max}_{\bar{x}^P} F_P(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^P) = \text{Max}_{\bar{x}^P} \begin{pmatrix} c_{P1}x \\ \vdots \\ c_{Pk_P}x \end{pmatrix}, \end{aligned} \quad (6.1)$$

soumis à : $S = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0, b \in \mathbb{R}^m\}$.

Où, $S \neq \emptyset$ est la région réalisable, m est le nombre de contraintes, k_p est le nombre de fonctions objectifs de DCD_p , n_p est le nombre de variables de décision de DCD_p , $c_{pq}^{pj} = c_{pq}^{p1}, c_{pq}^{p2}, \dots, c_{pq}^{pn_p}$, $p = 1, \dots, P$, $q = 1, \dots, k_p$, c_p est une matrice de taille $k_p \times n$, A est une matrice de taille $(m \times n)$ et b est un m -vecteur. On suppose que $m < n$ et que S est un polyèdre compact. C'est-à-dire un polyèdre borné.



6.3 Préliminaires

Par la désignation \tilde{A} , nous référons à une matrice de dimensions $((n + m) \times n)$, où les m premières lignes correspondent aux lignes de la matrice A , tandis que les n lignes suivantes représentent les contraintes de non-négativité ($x_{ij} \geq 0, i = 1, \dots, P, j = 1, \dots, n_i$). Le vecteur \tilde{b} , de taille $(n + m)$, satisfait $\tilde{b}_i = b_i$ pour $i = 1, \dots, m$ et $\tilde{b}_i = 0$ pour $i = m + 1, \dots, m + n$. Cela se traduit par

$$\tilde{A} = \begin{pmatrix} A \\ -Id_n \end{pmatrix}, \quad \tilde{b} = \begin{pmatrix} b \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

où Id_n représente la matrice identité d'ordre n .



Faisons l'introduction de M , défini comme l'ensemble des indices de lignes de la matrice $\tilde{A} : M = \{1, \dots, m + n\}$.

Pour un sous-ensemble $Q \subseteq M$, \tilde{A}_Q désigne la matrice dérivée de \tilde{A} en supprimant les lignes dont les indices ne se trouvent pas dans Q , et de manière similaire, \tilde{b}_Q est dérivé. Ainsi, nous attribuons le terme "face" de S , tel que présenté dans l'appendice III, pour l'ensemble défini par :

$$F(Q) = \{x \in S : \tilde{A}_Q x = \tilde{b}_Q\}.$$

Notations 6.3.1.

- ◇ L'ensemble N_p englobe l'ensemble de tous les points non-dominés du $p^{\text{ème}}$ niveau, c'est-à-dire ceux qui émergent du $p^{\text{ème}}$ PPLMO, un composant du problème 6.1. Parallèlement, D_p englobe les points dominés du même niveau.
- ◇ La compacité du polyèdre de la région réalisable S garantit que celle-ci possède un nombre fini de sommets, également qualifiés de points extrêmes. Dans ce contexte, S^{dex} représente l'ensemble complet de ces points extrêmes.
- ◇ Pour chaque niveau p , l'ensemble N_p^{dex} se matérialise comme l'intersection entre S^{dex} et N_p , représentant ainsi l'ensemble des points extrêmes non-dominés du $p^{\text{ème}}$ niveau.
- ◇ Pour référence ultérieure, notons que N_p^{dx} désigne un point spécifique appartenant à l'ensemble N_p^{dex} .



Lemme 6.3.2. Soient $x^1, x^2 \in S$ et $x^1 \in D_p$. Alors, $[x^1, x^2) \subset D_p$, où

$$[x^1, x^2) = \{\alpha x^1 + (1 - \alpha)x^2 : 0 < \alpha \leq 1\}$$

est le segment délimité par x^1 et x^2 . En particulier, si $x^1 \in \overset{\circ}{S}$, alors $\overset{\circ}{S} \subseteq D_p$.

Théorème 6.3.3. Soit X un sous-ensemble de S , supposons que $x^1 \in \overset{\circ}{X}$ et $x^1 \in N_p$, alors $\bar{X} \subset N_p$.

En règle générale, les objectifs au sein d'un PPLMO entrent en conflit, ce qui implique que $N_p \neq \emptyset$. En conséquence, d'après le Théorème 6.3.3 et le Lemme 6.3.2, nous déduisons que $\overset{\circ}{S} \subseteq D_p$, ce qui indique que les points réalisables non-dominés résident au sein des faces de la région réalisable S . Par conséquent, la quête de points non-dominés se concentre sur la frontière de la région réalisable (c'est-à-dire S privé de $\overset{\circ}{S}$).

Définition 6.3.4. Soit $Q \subseteq M$, alors l'ensemble $N_p(Q) = N_p \cap F(Q)$ est appelé face non-dominée du $p^{\text{ième}}$ -niveau.

Théorème 6.3.5 (Décomposition de N_p en faces non-dominées). Soient $Q \subset M$, $l = 0, \dots, m+n$ et définissons $\mathcal{F}^l = \{Q \subset M : \text{Card}(Q) = l\}$. Alors, on a :

$$N_p = \bigcup_{l=0}^{m+n} \bigcap_{Q \in \mathcal{F}^l} N_p(Q). \quad (6.2)$$

Définition 6.3.6. $F(Q)$ est appelée face maximale si les conditions suivantes sont vérifiées :

◇ Le sous-espace engendré par l'ensemble

$$\tilde{F}(Q) = \{x \in \mathbb{R}^n : \tilde{A}_Q x = \tilde{b}_Q\}$$

est équivalent à l'espace engendré par $F(Q)$.

◇ Si $F(\hat{Q})$ et $F(Q)$ représentent la même face, alors cela implique que $\hat{Q} \subset Q$.

Théorème 6.3.7. Considérons une face maximale $F(Q)$ telle que $N_p^{\text{dex}}(Q) = N_p^{\text{dex}} \cap F(Q) \neq \emptyset$. Dans ce contexte, il en découle que $F(Q)$ est inclus dans N_p seulement si $N_p^{\text{dex}}(Q)$ contient un minimum de $n - \text{Card}(Q) + 1$ éléments.



Théorème 6.3.8. Considérons une solution de base réalisable $x^0 = (x_{J_B}, x_{J_N})$ avec un ensemble d'indices J_B , et examinons le problème linéaire suivant :

$$w_p = \max \sum_{i=1}^{k_p} v_{pi}, \quad (6.3)$$

soumis à $\tilde{S}_p = \{(x, v_p) : x \in S, c_p^t x - v_p \geq c_p^t x^0, v_p \geq 0\}$, où $v_p = (v_{p1}, \dots, v_{pk_p})$. Dans ce contexte, nous énonçons les éléments suivants de manière organisée :

- ◇ x^0 est un point non-dominé si et seulement si $w_p = 0$.
- ◇ x^0 est un point dominé si et seulement si $w_p > 0$.

6.4 Exploration des Solutions de Compromis dans un PPLMO-MN

Dans le contexte des PPLMO-MNs, l'ensemble global \hat{N} peut être construit de la manière suivante : nous commençons par générer les ensembles N_p pour chaque problème multi-objectif individuel qui contribue au problème global. Ensuite, nous prenons leur intersection. Cette opération peut être réalisée en utilisant la méthode de P.L. Yu et Zeleny décrite dans [Yu 75], étant donné que chaque ensemble N_p est en réalité une réunion de faces. Ainsi, l'intersection des ensembles N_p peut être obtenue de manière relativement directe.

Cependant, il est important de noter que l'approche proposée par P.L. Yu et Zeleny se compose de deux phases majeures. Dans la première phase, l'ensemble de tous les points extrêmes non-dominés est identifié. La seconde phase permet d'obtenir l'ensemble de tous les points non-dominés. De plus, cette procédure doit être répétée P fois pour obtenir les ensembles N_p , ce qui peut entraîner une consommation importante de temps et de mémoire. C'est pourquoi nous proposons une autre méthode pour obtenir l'ensemble \hat{N} .

Dans un PPLMO-MN, chaque DCD_p recherche une solution non-dominée réalisable dans l'ensemble des solutions non-dominées N_p . Ceci est réalisé en maximisant sa propre fonction objectif F_p par rapport aux variables de décision \bar{x}^p qui sont sous son contrôle. En conséquence, tout compromis potentiel \hat{x} doit nécessairement se trouver dans l'intersection des ensembles de solutions non-dominées des différents problèmes multi-objectifs qui forment les niveaux individuels. En fait, si \hat{x} n'appartient pas à l'intersection des ensembles N_p pour $p = 1, \dots, P$, cela implique que \hat{x} serait une solution dominée pour au moins l'un des problèmes de l'un des décideurs, ce qui contredirait la nature même du compromis. Par conséquent, nous pouvons naturellement définir l'ensemble de tous les compromis possibles (c'est-à-dire les solutions non-dominées réalisables) pour le PPLMO-MN comme suit :

$$\hat{N} = \bigcap_{p=1}^P N_p. \quad (6.4)$$

De manière similaire, nous pouvons définir l'ensemble de tous les points extrêmes non-dominés du PPLMO-MN comme étant l'ensemble des points extrêmes non-dominés qui sont communs à tous les ensembles N_p , c'est-à-dire :

$$\hat{N}^{dex} = \bigcap_{p=1}^P N_p^{dex}. \quad (6.5)$$

Cette définition de l'ensemble \hat{N}^{dex} est cohérente et garantit que aucun point extrême non-dominé n'est omis. En effet, puisque pour tout $p = 1, \dots, P$, l'ensemble N_p de points non-dominés est construit en tant qu'union de faces non-dominées de la région réalisable S (voir [Yu 75]), il s'ensuit que, pour tout p_1 et p_2 , l'intersection $N_{p_1} \cap N_{p_2}$ ne peut pas introduire de nouveaux points extrêmes non-dominés qui ne sont pas déjà présents dans les ensembles N_{p_1} et N_{p_2} . Autrement dit, tous les points extrêmes du polyèdre $N_{p_1} \cap N_{p_2}$ sont déjà des points extrêmes pour les ensembles N_{p_1} et N_{p_2} . Cette propriété est valable car les ensembles N_{p_1} et N_{p_2} sont des polyèdres inclus dans la frontière d'un polyèdre plus grand, à savoir S . Cela diffère du cas général, comme illustré dans les figures 6.1.



Étant donné que les ensembles N_p , pour $p = 1, \dots, P$, ne sont pas nécessairement convexes, il en découle que l'ensemble \hat{N} n'est pas nécessairement convexe de manière générale. Dans le cas où $\hat{N} = \emptyset$, cela indique que le problème (6.1) est insoluble.

6.4.1 Identification des Faces Non-dominées d'un PPLMO-MN

Supposons que $\hat{N} \neq \emptyset$ et définissons l'ensemble \hat{N} comme l'ensemble de tous les sous-ensembles Q de M correspondant aux faces non vides $F(Q)$ de la région réalisable S incluses dans \hat{N} :

$$\hat{N} = \{Q \subset M : F(Q) \neq \emptyset \text{ et } F(Q) \subseteq \hat{N}\}.$$

Notations 6.4.1. La face associée à un élément Q de \hat{N} est désignée comme une face non-dominée d'un PPLMO-MN et est symbolisée par :

$$\hat{N}(Q) = \bigcap_{p=1}^P N_p(Q).$$

Il est évident que pour deux sous-ensembles Q_1 et Q_2 de \hat{N} tels que $Q_1 \subset Q_2$, on a $F(Q_2) \subseteq F(Q_1)$. Afin d'exclure les faces $F(Q_2)$ qui sont entièrement contenues dans une autre face plus



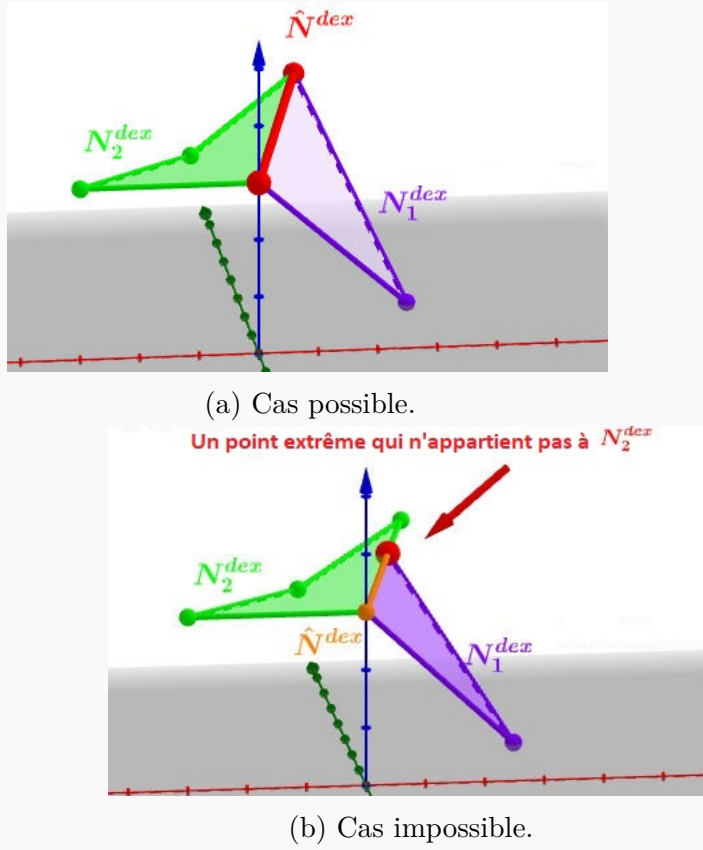


FIGURE 6.1 – Illustration de l'ensemble \hat{N}^{dex} pour $P = 2$ et $S \subset \mathbb{R}^3$.

grande $F(Q_1)$, nous définissons un ensemble distinct \tilde{N} contenant tous les éléments de \hat{N} ne contenant aucun autre élément de \hat{N} , comme suit :

$$\tilde{N} = \{Q \in \hat{N} : \text{pour tout } \hat{Q} \in \hat{N}, Q \neq \hat{Q} \text{ et } \hat{Q} \not\subset Q\}.$$

Théorème 6.4.2. La décomposition suivante de \hat{N} convient : $\hat{N} = \bigcup_{Q \in \tilde{N}} \hat{N}(Q)$.

Démonstration. Puisque pour tout $p = 1, \dots, P$, on a

$$N_p = \bigcup_{l=0}^{n+m} \bigcap_{Q \in \mathcal{F}^l} N_p(Q).$$

Alors, on obtient

$$\begin{aligned} \hat{N} &= \bigcap_{p=1}^P N_p = \bigcap_{p=1}^P \left(\bigcup_{l=0}^{n+m} \bigcap_{Q \in \mathcal{F}^l} N_p(Q) \right) \\ &= \bigcup_{l=0}^{n+m} \bigcap_{Q \in \mathcal{F}^l} \left(\bigcap_{p=1}^P N_p(Q) \right) = \bigcup_{l=0}^{n+m} \bigcap_{Q \in \mathcal{F}^l} \hat{N}(Q). \end{aligned}$$

■

Définition 6.4.3. Une face $F(Q)$, où $Q \in \tilde{\mathcal{N}}$, est désignée comme un *espace de tri* pour le PPLMO-MN, et nous le notons par \mathcal{SP} .

6.4.2 Élaboration de l'Algorithme : Processus Complet de Construction

1. Pour entamer le processus, nous commençons par obtenir l'ensemble complet \hat{N}^{dex} , qui regroupe tous les points extrêmes non-dominés provenant des Problèmes de Programmation Linéaire Multi-Objectifs (PPLMOs) englobant le problème multi-niveaux. Pour réaliser cela, nous adoptons la procédure énoncée par P.L. Yu et Zeleny dans leur travail [Yu 75], comme détaillé dans l'appendice I. En pratique, cela signifie que nous utilisons l'algorithme développé par P.L. Yu et Zeleny pour chaque PPLMO individuel, c'est-à-dire pour chaque $p = 1, \dots, P$, afin de générer l'ensemble N_p^{dex} . Par la suite, nous rassemblons ces ensembles N_p^{dex} pour dériver l'ensemble global \hat{N}^{dex} contenant les points extrêmes non-dominés pour le PPLMO-MN.

2. Ensuite, notre démarche consiste à identifier l'ensemble des faces qui possèdent la propriété de non-dominance pour l'ensemble des niveaux simultanément. Il est important de noter que si nous partons du principe que \hat{N} n'est pas vide (c'est-à-dire que des solutions non-dominées existent dans le problème), alors il est toujours possible de trouver au moins une face non-dominée pour un PPLMO-MN donné. Même dans le scénario le plus pessimiste, nous obtenons des faces de dimension minimale, à savoir des points (éléments de \hat{N}^{dex}), voire des segments (faces de dimension 1). Cela souligne que notre méthode est capable de détecter des solutions non-dominées, quel que soit le degré de complexité du problème multi-niveaux sous-jacent.

3. Il est essentiel de noter que notre recherche se concentre spécifiquement sur les faces qui possèdent deux caractéristiques essentielles : d'une part, tous leurs points extrêmes se trouvent dans \hat{N}^{dex} , et d'autre part, ces faces ont une dimension maximale de $n - 1$. Ces propriétés sont cruciales car elles garantissent que les faces que nous considérons sont pertinentes pour le test de non-dominance que nous allons établir dans les étapes suivantes. Ainsi, pour identifier et examiner ces faces, nous procédons comme suit :

(a) Tout d'abord, nous déterminons le cardinal de \hat{N}^{dex} , noté \hat{d} , et nous désignons les éléments de \hat{N}^{dex} par \hat{N}^{dex_i} pour $i = 1, \dots, \hat{d}$.

(b) Ensuite, pour chaque g allant de 1 à \hat{d} , nous définissons l'ensemble \mathcal{U}^g qui contient tous les sous-ensembles de l'ensemble $\{1, \dots, \hat{d}\}$ ayant exactement g éléments :

$$\mathcal{U}^g = \{U \subseteq \{1, \dots, \hat{d}\} : \text{Card}(U) = g\}. \quad (6.6)$$

(c) Pour chaque g et pour chaque ensemble $U \in \mathcal{U}^g$, nous associons l'enveloppe convexe des



points \hat{N}^{dx_i} correspondant aux indices de U . Cela définit l'ensemble $\mathcal{H}(U)$ comme suit :

$$\mathcal{H}(U) = \left(\left\{ \hat{N}^{dx_i} : i \in U \right\} \right). \quad (6.7)$$



Il est important de noter que l'ensemble $\mathcal{H}(U)$ peut ne pas être une face de S dans tous les cas.

(d) Parmi tous les ensembles $\mathcal{H}(U)$ ainsi définis, nous cherchons ceux qui sont des faces de la région réalisable S . En d'autres termes, nous cherchons un ensemble d'indices $Q \subseteq M$ tel que la face $F(Q)$ soit égale à $\mathcal{H}(U)$, c'est-à-dire $F(Q) = \mathcal{H}(U)$. Pour ce faire, nous substituons les points \hat{N}^{dx_j} pour $j \in U$ dans le système d'équations linéaires $\tilde{A}x = \tilde{b}$, puis nous construisons l'ensemble Q contenant les indices $i \in M$ pour lesquels les équations sont satisfaites, c'est-à-dire :

$$\tilde{A}_i \hat{N}^{dx_j} = \tilde{b}_i, \quad \text{pour tout } j \in U.$$

Si l'ensemble Q ainsi obtenu est vide, cela signifie que $\mathcal{H}(U)$ n'est pas une face de S .

(e) Pour chaque valeur de g de 1 à \hat{d} , et pour chaque ensemble U appartenant à \mathcal{U}^g , nous définissons l'ensemble \mathcal{G}^g qui contient uniquement les faces de la région réalisable S pouvant être définies comme l'enveloppe convexe de g points extrêmes non-dominés de \hat{N}^{dex} . Formellement :

$$\mathcal{G}^g = \{F(Q) : F(Q) = \mathcal{H}(U)\}. \quad (6.8)$$

(f) Puisque nous cherchons à identifier les espaces de tri comme enveloppe convexe de g points extrêmes non-dominés de \hat{N}^{dex} , où $Q \in \mathcal{F}^l$ (c'est-à-dire les ensembles de faces non-dominées de dimension $n - l$), nous devons nous assurer que g est suffisamment grand pour que ces faces aient une chance de passer le test de non-dominance. Selon le théorème 6.3.7, nous devons avoir $g \geq n - l + 1$.

(g) En conséquence, nous pouvons créer un ensemble $\hat{\mathcal{G}}^g$ qui contient uniquement les faces de la région réalisable S qui peuvent être définies comme l'enveloppe convexe d'au moins g points extrêmes non-dominés de \hat{N}^{dex} et qui doivent également satisfaire la condition $g \geq n - l + 1$ pour chaque l possible :

$$\hat{\mathcal{G}}^g = \bigcup_{l \geq n-g+1} \{F(Q) : Q \in \mathcal{F}^l, F(Q) = \mathcal{H}(U)\}. \quad (6.9)$$

4. En fin de compte, nous obtenons l'ensemble $\hat{\mathcal{G}}$ qui regroupe toutes les faces de la région réalisable S qui peuvent être définies comme l'enveloppe convexe d'un certain nombre de points extrêmes non-dominés de \hat{N}^{dex} et qui satisfont les conditions de dimension pour le test de non-



dominance :

$$\hat{\mathcal{G}} = \bigcup_{g=1}^d \hat{\mathcal{G}}^g. \quad (6.10)$$



Il est important de noter que $\hat{\mathcal{G}}^1$ est égal à \hat{N}^{dex} , qui est l'ensemble des points extrêmes non-dominés.

5. Pour tester la non-dominance des éléments de l'ensemble $\hat{\mathcal{G}}$, nous utilisons les critères suivants :

Proposition 6.4.4 (Le critère de non-dominance pour un point). Soit $\mathbf{P}^{dx} \in S$ et soient w_p les valeurs optimales du problème (6.3) soumis à \tilde{S}_p , tel que défini précédemment dans le théorème 6.3.8. Définissons la quantité R comme la somme des valeurs optimales w_p pour tous les niveaux p . Alors, \mathbf{P}^{dx} est un point non-dominé pour le PPLMO-MN si et seulement si $R = 0$.



Il est important de souligner que le critère de non-dominance pour un point, tel qu'énoncé dans la Proposition 6.4.4, possède une signification profonde. Il équivaut en réalité au test de non-dominance de P.L. Yu et Zeleny, appliqué aux sous-problèmes de PLMO qui composent le PPLMO-MN. Ainsi, ce critère offre une manière explicite de déterminer si un point appartient à l'ensemble \hat{N} , en généralisant le test de non-dominance classique au contexte multi-niveaux.

Théorème 6.4.5 (Le critère de non-dominance pour une face). Soit $g \geq 2$ et considérons une face $F(Q)$ appartenant à l'ensemble $\hat{\mathcal{G}}^g$. Supposons que le point \mathbf{P}^{dx} appartient à $F(Q) \setminus \partial F(Q)$ et n'appartient pas à \hat{N}^{dex} . Alors, le point \mathbf{P}^{dx} appartient à l'ensemble \hat{N} si et seulement si la face $F(Q)$ est entièrement incluse dans l'ensemble \hat{N} .



Dans la preuve du Théorème 6.4.5 suivante, la fermeture $\overline{F(Q)}$ de $F(Q)$ est prise par rapport à la topologie induite par celle de \mathbb{R}^n sur le plus petit hyperplan contenant $F(Q)$.

Démonstration.

◇ (\Rightarrow) Supposons que $\mathbf{P}^{dx} \in F(Q) \setminus \partial F(Q) \cap (\hat{N} - \hat{N}^{dex})$. Cela signifie que \mathbf{P}^{dx} est à la fois dans la face $F(Q) \setminus \partial F(Q)$ et en dehors de l'ensemble \hat{N}^{dex} . Par conséquent, \mathbf{P}^{dx} appartient



à l'intérieur relatif de la face $F(Q)$, c'est-à-dire $\mathbf{P}^{dx} \in ri(F(Q))$. Selon le Théorème 6.3.3, nous savons que $\overline{F(Q)} = F(Q) \subseteq \hat{N}$, ce qui conduit à $F(Q) \subseteq \hat{N}$.

◇ (\Leftarrow) Supposons que $F(Q) \subseteq \hat{N}$. Cela signifie que tous les points de la face $F(Q)$ sont contenus dans l'ensemble \hat{N} .

■



Le Théorème (6.4.5) offre une perspective intéressante : si l'intérieur relatif de la face $F(Q)$ contient un point non-dominé \mathbf{P}^{dx} , alors $F(Q)$ est elle-même une face non-dominée pour le PPLMO-MN. L'application pratique de ce théorème repose sur la sélection de g coefficients $\alpha_1, \dots, \alpha_g$ tels que $\sum_{t=1}^g \alpha_t = 1$ et $0 < \alpha_t < 1$. Ensuite, en testant la non-dominance du point $\mathbf{P}^{dx} = \sum_{t=1}^g \alpha_t N^{dx_t}$ à l'aide de la Proposition 6.4.4, on peut déterminer si la face $F(Q)$ est non-dominée.

En somme, une conclusion importante émerge : une face est non-dominée pour le PPLMO-MN si et seulement si elle renferme au moins un point non-dominé dans son intérieur relatif. Cette observation simplifie la recherche des faces non-dominées et facilite leur identification dans le contexte du problème multi-niveaux.

6. Pendant l'exécution de la procédure, dès qu'une face non-dominée $F(Q)$ est identifiée, il devient clair que toutes les faces contenues à l'intérieur de cette dernière sont également non-dominées. Pour gagner en efficacité, nous adoptons la stratégie suivante :

- (a) Pour une face non-dominée $F(Q) \in \hat{G}^g$, nous définissons l'ensemble $\mathcal{V}_{F(Q)}$ qui regroupe toutes les faces strictement contenues dans $F(Q)$.
- (b) Nous introduisons $\hat{\mathcal{S}}\mathcal{P}$, l'ensemble de tous les espaces de tri du PPLMO-MN. Pour chaque $F(Q) \in \hat{\mathcal{S}}\mathcal{P}$, nous définissons \mathcal{W} comme l'ensemble des faces qui ne nécessitent pas d'être vérifiées pour le test de non-dominance. Formellement :

$$\mathcal{W} = \bigcup_{F(Q) \in \hat{\mathcal{S}}\mathcal{P}} \mathcal{V}_{F(Q)}.$$



Cette approche nous permet de réduire considérablement le nombre de tests de non-dominance nécessaires, en évitant de réexaminer les faces qui ont déjà été prouvées non-dominées en raison de leur inclusion dans une face non-dominée plus grande.

6.4.3 Exploration Exhaustive : L'Algorithme dans toute sa Splendeur

Dans cette section, nous exploiterons la construction que nous avons développée précédemment pour décrire une méthode permettant de générer l'ensemble \hat{N} , qui représente toutes les

solutions non-dominées d'un PPLMO-MN. En résumé, notre démarche commence par la création de l'ensemble N^{dex} , contenant les points extrêmes non-dominés partagés entre tous les niveaux. Ensuite, nous procédons au test de non-dominance pour toutes les faces de la région réalisable S qui possèdent exclusivement des points extrêmes issus de N^{dex} .

Algorithme 6 Génération de l'Ensemble des solutions Non-dominées \hat{N} .

Étape 1 : Obtenir \hat{N}^{dex} comme décrit dans la section précédente |

si $\hat{d} = 1$ alors

└ arrêter avec $\hat{N} = \hat{N}^{dex}$ |

Étape 2 : Poser $\hat{\mathcal{S}}\mathcal{P} = \emptyset$, $\mathcal{W} = \emptyset$ et $d = \hat{d}$ |

tant que $d > 1$ **faire**

1. Illustrer l'ensemble $\hat{\mathcal{G}}^d$ |

2. Poser $\hat{\mathcal{G}}^d = \hat{\mathcal{G}}^d - \mathcal{W}$ |

3. **tant que** $\hat{\mathcal{G}}^d \neq \emptyset$ **faire**

(a) Prendre $K \in \hat{\mathcal{G}}^d$ |

(b) Choisir $\mathbf{P}^{dx} \in ri(K)$ |

(c) Poser $p \leftarrow 1$ et $R = 0$ |

(d) **tant que** $p \leq P$ **faire**

i. Résoudre le problème (6.3) soumis à \tilde{S}_p |

ii. poser $R \leftarrow R + w_p$ |

iii. poser $p \leftarrow p + 1$ |

si $R = 0$ **alors**

└ obtenir \mathcal{V}_K et poser : $\hat{\mathcal{S}}\mathcal{P} = \hat{\mathcal{S}}\mathcal{P} \cup K$ et $\mathcal{W} = \mathcal{W} \cup \mathcal{V}_K$ |

(e) Poser $\hat{\mathcal{G}}^d = \hat{\mathcal{G}}^d - K$ |

4. Poser $d = d - 1$ |

Étape 3 : Obtenir l'ensemble \hat{N} comme suit :

$$\hat{N} = \begin{cases} \cup \{K : K \in \hat{\mathcal{S}}\mathcal{P}\} \cup \hat{N}^{dex}, & \text{si } \mathcal{W} \neq \emptyset, \\ \hat{N}^{dex}, & \text{si } \mathcal{W} = \emptyset \end{cases}$$

Sortie : L'ensemble \hat{N} .



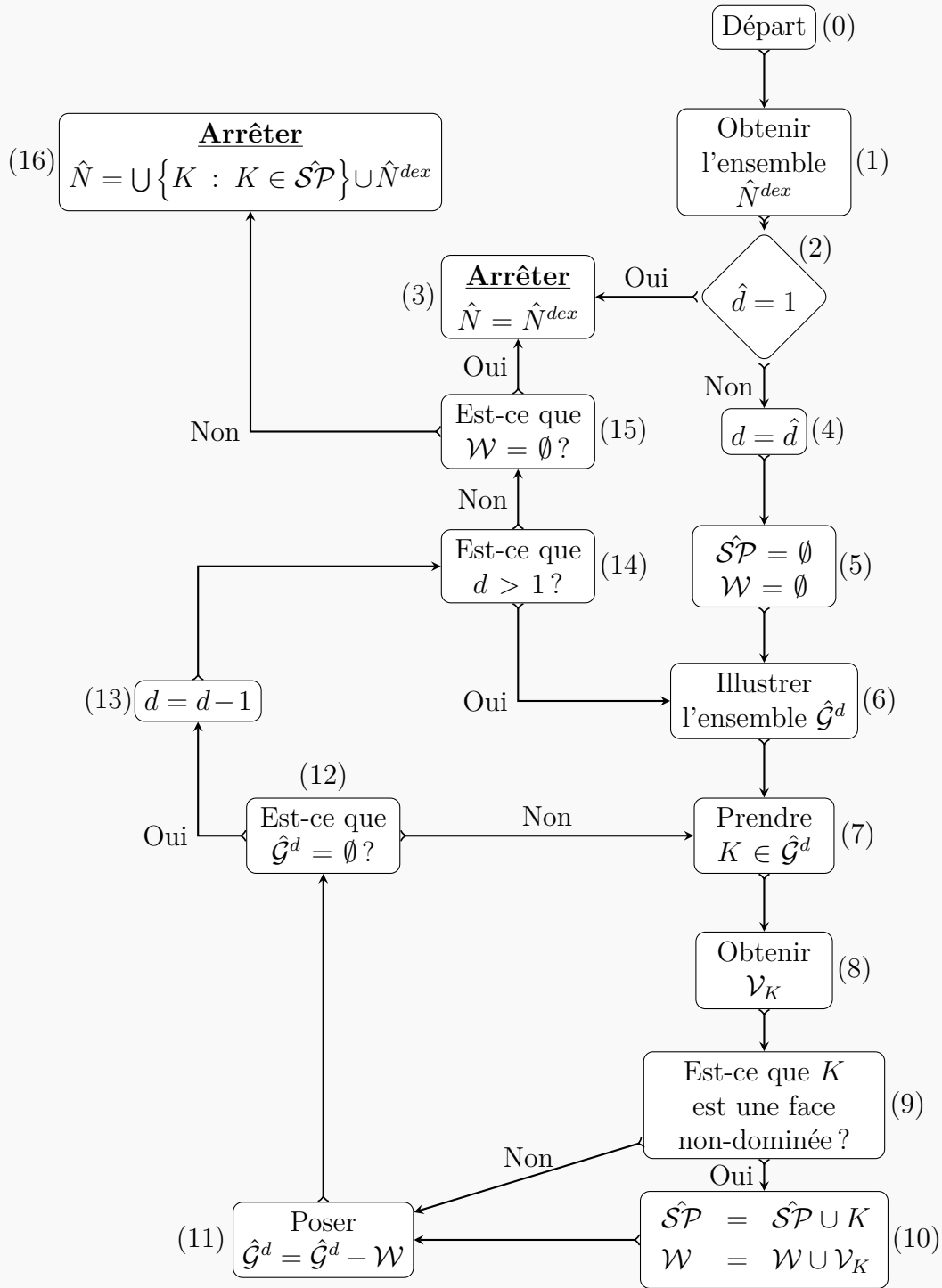


FIGURE 6.2 – Schéma de l’Algorithme 6

Dévoilement du Schéma Algorithmique : Décryptage et Détails

Case 1 : Générer l'ensemble \hat{N}^{dex} des points extrêmes non-dominés en appliquant la méthode du simplexe multi-critère généralisée, comme décrite dans [Yu 74] pour les P PPLMOs constituant le problème multi-niveaux. Cela donne les P ensembles $N_1^{dex}, \dots, N_P^{dex}$, que l'on intersecte pour obtenir \hat{N}^{dex} .

Case 6 : Illustrer l'ensemble $\hat{\mathcal{G}}^d$ en localisant toutes les faces de la région réalisable S qui contiennent exactement d points extrêmes non-dominés appartenant à \hat{N}^{dex} , comme décrit dans la section précédente.

Cases (7)-(12) : Tester tous les éléments de $\hat{\mathcal{G}}^d$ en utilisant le critère défini dans le théorème 6.4.5. Dans l'étape (10), identifier celles qui sont non-dominées et les ajouter à l'ensemble $\hat{\mathcal{S}}\mathcal{P}$. De plus, ajouter leurs sous-faces non-dominées à l'ensemble \mathcal{W} pour éviter de les tester à nouveau.

6.4.4 Première Illustration Numérique : Mettre en Action l'Algorithme 6

Énoncé

Considérons le PPLMO à deux niveaux suivant :

$$\begin{aligned}
 \text{Niveau 1 : } & \underset{x_1}{\text{Max}} \left(f_{11}(x) = 2x_1 + 2x_2, f_{12}(x) = -\frac{1}{2}x_1 + \frac{7}{25}x_2, f_{13}(x) = -\frac{1}{5}x_1 + \frac{1}{2}x_2 \right), \\
 & \text{tel que } x_2 \text{ résolvent} \\
 \text{Niveau 2 : } & \underset{x_2}{\text{Max}} \left(f_{21}(x) = x_1 + 3x_2, f_{22}(x) = -2x_1 - x_2, f_{23}(x) = x_2 \right),
 \end{aligned} \tag{6.11}$$

soumis à :

$$x \in S = \left\{ x \in \mathbb{R}^2 : Ax \leq b, x \geq 0, b \in \mathbb{R}^7 \right\}.$$

Où

$$A = \begin{pmatrix} -2 & 1 \\ -1 & 2 \\ 0 & 1 \\ 1 & 0 \\ -1 & -2 \\ 3 & -4 \\ 1 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 9 \\ 6 \\ 6 \\ -9 \\ 7 \\ 2 \end{pmatrix}.$$



Considérons les deux PPLMOs suivants :

$$(1) \begin{cases} \max c_1 x \\ Ax \leq b \\ x \geq 0 \end{cases}, \quad (2) \begin{cases} \max c_2 x \\ Ax \leq b \\ x \geq 0. \end{cases} \quad (6.12)$$

Avec

$$c_1 = \begin{pmatrix} 2 & 2 \\ -\frac{1}{2} & \frac{7}{25} \\ -\frac{1}{5} & \frac{1}{2} \end{pmatrix}, \quad c_2 = \begin{pmatrix} 1 & 3 \\ -2 & -1 \\ 0 & 1 \end{pmatrix}.$$

Étape 1 : Nous appliquons l’algorithme décrit dans l’appendice I pour obtenir les deux ensembles de points extrêmes non-dominés suivants :

◇ PPLMO (1) : $N_1^{dex} = \{N^{dx_1} = (6, 6), N^{dx_2} = (3, 6), N^{dx_3} = (1, 5)\}$.

◇ PPLMO (2) : $N_2^{dex} = \{N^{dx_1}, N^{dx_2}, N^{dx_3}, N^{dx_4} = (0, 3), N^{dx_5} = (0, 0)\}$.

Alors, l’ensemble \hat{N}^{dex} du problème 6.11 est donné par : $\hat{N}^{dex} = \{N^{dx_1}, N^{dx_2}, N^{dx_3}\}$.

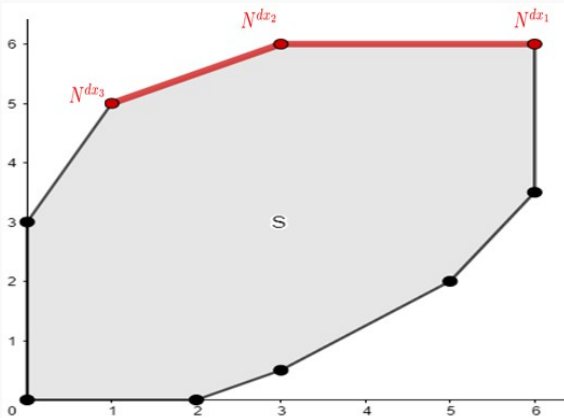


FIGURE 6.3 – Ensemble des solutions du PPLMO (1)

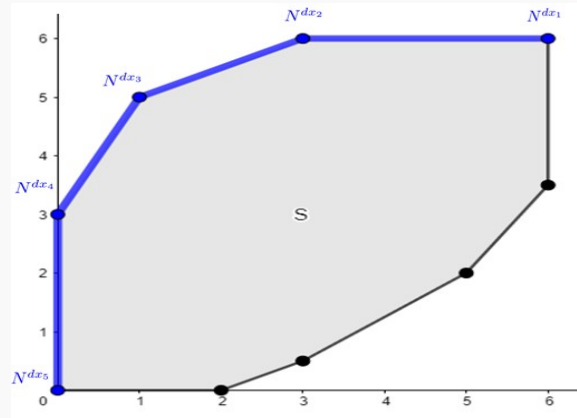


FIGURE 6.4 – Ensemble des solutions de PPLMO (2)

Étape 2 : Poser $\hat{\mathcal{S}}\mathcal{P} = \emptyset$, $\mathcal{W} = \emptyset$ et $\hat{d} = \hat{d} - 1 = 2$.

Itération 1 : Nous choisissons $\mathbf{P}^{dx} = \frac{1}{3} (N^{dx_1} + N^{dx_2} + N^{dx_3})$. Ensuite, nous résolvons le PPL (6.3) pour $p = 1$ et $p = 2$, et nous obtenons $R = w_1 + w_2 \neq 0$. Cela implique que $\mathcal{H}(\{N^{dx_1}, N^{dx_2}, N^{dx_3}\})$ n’est pas une face non-dominée du PPLMO à deux niveaux (6.11).

Itération 2 : Nous diminuons \hat{d} de 1 : $\hat{d} = \hat{d} - 1 = 2$, puis nous calculons l’ensemble $\hat{\mathcal{G}}^2$ comme suit :

$$\hat{\mathcal{G}}^2 = \left\{ \mathcal{H}(\{N^{dx_1}, N^{dx_2}\}), \mathcal{H}(\{N^{dx_2}, N^{dx_3}\}) \right\},$$

comme illustré dans la figure 6.5. Maintenant, nous avons seulement deux faces à tester.

Test de la face $K_1 = \mathcal{H}(\{N^{dx_1}, N^{dx_2}\})$: Nous choisissons $\mathbf{P}^{dx} = \frac{1}{2} (N^{dx_1} + N^{dx_2}) \in ri(K_1)$. Ensuite, nous résolvons le PPL (6.3) pour $p = 1$ et $p = 2$, et nous obtenons $R = w_1 + w_2 = 0$. Cela implique que $K_1 = \mathcal{H}(\{N^{dx_1}, N^{dx_2}\})$ est une face non-dominée du PPLMO à deux niveaux (6.11).

Test de la face $K_2 = \mathcal{H}(\{N^{dx_2}, N^{dx_3}\})$: Nous choisissons $\mathbf{P}^{dx} = \frac{1}{2} (N^{dx_2} + N^{dx_3}) \in ri(K_2)$. Ensuite, nous résolvons le PPL (6.3) pour $p = 1$ et $p = 2$, et nous obtenons $R =$

$w_1 + w_2 = 0$. Cela implique que $K_2 = \mathcal{H}(\{N^{dx_2}, N^{dx_3}\})$ est une face non-dominée du PPLMO à deux niveaux (6.11).

Étape 3 Pour obtenir l'ensemble \hat{N} de toutes les solutions non-dominées du PPLMO à deux niveaux (6.11), nous effectuons l'union des deux faces non-dominées identifiées, comme illustré dans la figure 6.5, soit :

$$\hat{N} = \mathcal{H}(N^{dx_1}, N^{dx_2}) \cup \mathcal{H}(N^{dx_2}, N^{dx_3}).$$



Pour mieux appréhender cette notion, examinons d'abord l'ensemble des solutions du PPLMO (1), mis en évidence dans la figure 6.3 :

$$N_1 = \mathcal{H}(N^{dx_1}, N^{dx_2}) \cup \mathcal{H}(N^{dx_2}, N^{dx_3})$$

De manière analogue, considérons l'ensemble des solutions du PPLMO (2), présenté dans la figure 6.4 :

$$N_2 = \mathcal{H}(N^{dx_1}, N^{dx_2}) \cup \mathcal{H}(N^{dx_2}, N^{dx_3}) \cup \mathcal{H}(N^{dx_3}, N^{dx_4}) \cup \mathcal{H}(N^{dx_4}, N^{dx_5}).$$

Dès lors, il est essentiel de noter que l'ensemble global \hat{N} de tous les compromis possibles du PPLMO à deux niveaux (6.11) intègre simultanément les solutions non-dominées tant pour les problèmes (1) que (2). Cette situation est clairement illustrée dans la figure 6.5, mettant en évidence comment les compromis sont atteints.

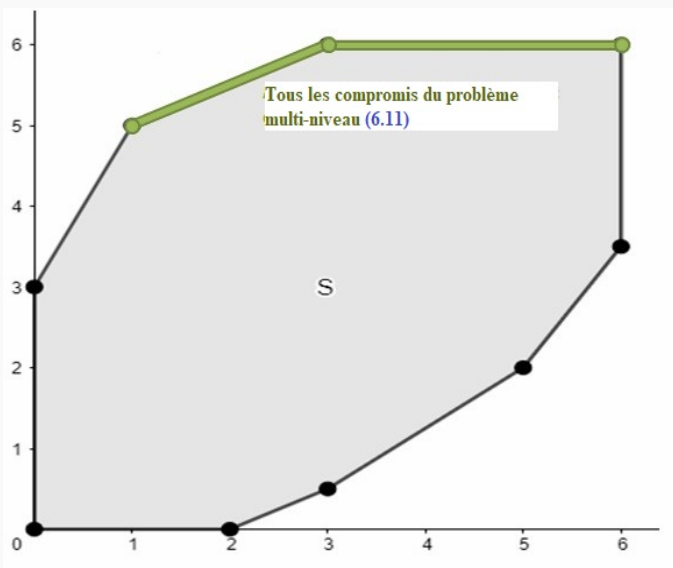


Figure 6.5 – L'ensemble de tous les compromis possible du PPLMO à deux niveaux (6.11).



6.5 Résolution des PPLMO-MNs via une Approche Novatrice et Constructive

Résumé

Notre contribution principaleⁱⁱ, réside dans l'exploitation de l'ensemble de tri comme nouvelle région réalisable du problème. Nous avons élaboré un second algorithme qui étend les concepts de la méthode adaptée de Gabasov afin de résoudre un PPLMO-MN [Kac23c]. Dans ce contexte, nous choisissons un seul compromis parmi ceux disponibles.

Notre algorithme se déploie en deux phases distinctes. Dans un premier temps, nous générons l'ensemble de tous les points extrêmes non-dominés issus de chacun des PPLMOs qui constituent le PPLMO-MN. Cette étape est accomplie en utilisant l'algorithme 6, développé dans la première contribution de ce chapitre, et détaillé dans la référence [Kac22c]. L'objectif de cette phase est d'obtenir l'ensemble complet des solutions réalisables non-dominées \hat{N} , en vue de sélectionner par la suite l'ensemble de tri.

Dans la seconde phase, nous entamons le processus de recherche d'un compromis satisfaisant au sein de l'ensemble de tri que nous avons obtenu. Pour ce faire, nous définissons les bornes des variables de décision et choisissons un ensemble de tri spécifique. Ensuite, en résolvant $P - 1$ PPLMOs standard avec des contraintes sur les variables, nous entamons la quête du compromis idéal. Pour cette étape, nous utilisons l'algorithme 3 de méthode adaptée multi-critère que nous avons décrit dans la section 4.1.6, comme également référencée dans [Rad11b].

ii. *Cet article a été accepté pour publication dans le 33^{ième} volume, 3^{ième} numéro de la revue "Operations Research and Decisions". Vous pouvez également trouver la version préimpression de l'article sur arXiv en suivant la référence [Kac23c].*

Mots clés

Programmation linéaire multi-niveau, programmation linéaire, méthode adaptée, estimation de sous-optimalité, application affine.

6.5.1 Édification de l'Algorithme : De la Conception à la Réalisation

Dans cette section, nous décrivons un algorithme qui généralise la méthode adaptée de résolution des PPLMOs décrite dans la section 4.1.6, afin de résoudre un PPLMO-MN en choisissant une seule solution satisfaisante à partir d'un ensemble de tri \mathcal{SP} choisi préalablement.

Construction de l'Algorithme

Première phase

Générer l'ensemble de tous les compromis possibles \hat{N} d'un PPLMO-MN en utilisant la méthode décrite dans [Kac22c]. Ensuite, choisir un ensemble de tri $\mathcal{SP} \subseteq \hat{N}$ et supposer que $|\hat{N}^{dex}| \geq 2$, $\mathcal{SP} \neq \hat{N}^{dex}$ et $\mathcal{SP} \neq \emptyset$. Définir l'ensemble des points extrêmes de l'ensemble \mathcal{SP} comme suit : $\mathcal{SP}^{dex} := \hat{N}^{dex} \cap \mathcal{SP} = N^{dx_1}, \dots, N^{dx_s}$, où s est le cardinal de \mathcal{SP}^{dex} .

Deuxième phase

1. Soit $M > 0$ un nombre assez grand. Fixer les bornes de chacune des n variables de décision et des variables d'écart comme suit :

$$\begin{aligned} l_{ij}^{(1)} &= \min_{t \in \{1, \dots, s\}} \{N_{ij}^{dx_t}\}, & u_{ij}^{(1)} &= \max_{t \in \{1, \dots, s\}} \{N_{ij}^{dx_t}\}, & i &= 1, \dots, P, & j &= 1, \dots, n_i, \\ l_{(P+1)j}^{(1)} &= 0, & u_{(P+1)j}^{(1)} &= M, & & & j &= n+1, \dots, n+m. \end{aligned} \quad (6.13)$$

2. Poser $n_{P+1} = m$ et considérer les $n+m$ bornes suivantes des variables de décision et des variables d'écart :

$$l_{ij}^{(1)} \leq x_{ij} \leq u_{ij}^{(1)}, \quad i = 1, \dots, P+1, \quad j = 1, \dots, n_i. \quad (6.14)$$

3. Poser $p \leftarrow 1$.

4. Définir la nouvelle région réalisable : $\mathcal{S}_p = \{x \in \mathcal{SP} : l_{ij}^{(p)} \leq x_{ij} \leq u_{ij}^{(p)}\}$. Ensuite, résoudre le PPLMO à variables bornées :

$$\max_{x \in \mathcal{S}_p} c_p x. \quad (6.15)$$

Notons $\overset{c_p}{x}$ la solution du PPLMO (6.15).

5. Le DCD_p choisit $2n_p$ paramètres positifs \mathbf{l}_{pj} et \mathbf{r}_{pj} pour $j = 1, \dots, n_p$. Ensuite, définit les nouvelles bornes des variables de décision qui sont sous son contrôle comme suit :

$$l_{pj}^{(p)} = \overset{c_p}{x}_{pj} - \mathbf{l}_{pj} \quad \text{et} \quad u_{pj}^{(p)} = \overset{c_p}{x}_{pj} + \mathbf{r}_{pj}, \quad (6.16)$$

de sorte que

$$\overset{c_p}{x}_{pj} - \mathbf{l}_{pj} \geq l_{pj}^{(1)} \quad \text{et} \quad \overset{c_p}{x}_{pj} + \mathbf{r}_{pj} \leq u_{pj}^{(1)}. \quad (6.17)$$

6. Poser $p \leftarrow p + 1$.

7. Si $p > P$, arrêter avec $\overset{c_P}{x}$ comme compromis satisfaisant du PPLMO-MN. Sinon, aller à l'étape 4.



Algorithme 7 Algorithme Complet : Génération d'un seul Compromis pour un PPLMO-MN

Première phase

1. Appliquer la procédure décrite par M. Kaci et S. Radjef dans [Kac22c] pour générer l'ensemble complet des compromis \hat{N} du PPLMO-MN |
2. Choisir un ensemble de tri $\mathcal{SP} \subseteq \hat{N}$ et établir l'ensemble \mathcal{SP}^{dex} .

Deuxième phase

Étape 1 : Poser $p \leftarrow 1$ |

Étape 2 : Formuler les contraintes (6.14) |

tant que $p \leq P$ **faire**

1. Formuler le modèle (6.15) |
2. Résoudre le modèle (6.15) en utilisant l'algorithme 3 de méthode adaptée décrit dans la section 4.1.6 |
3. Soit \hat{x}^p la solution du problème de programmation linéaire (6.15) |
- si** $p < P$ **alors**
 - a. Choisir l_{p1}, \dots, l_{pn_p} et r_{p1}, \dots, r_{pn_p} de telle manière que les propriétés (6.17) soient vérifiées à l'étape b. |
 - b. Fixer les $2n_p$ nouvelles contraintes (6.16) |
4. Poser $p \leftarrow p + 1$ |

Sortie : \hat{x}^P est le compromis satisfaisant du PPLMO-MN |

6.5.2 Illustration Numérique du PPLMO à deux-niveaux (Section 6.4.4) : Étude de Cas à travers l'Algorithme 7

Première phase : Pour des détails, veuillez vous référer à l'exemple 6.4.4

Nous avons : $\hat{N} = \mathcal{H}(N^{dx_1}, N^{dx_2}) \cup \mathcal{H}(N^{dx_2}, N^{dx_3})$. Donc, nous choisissons l'espace de tri : $\mathcal{SP} = \mathcal{H}(N^{dx_2}, N^{dx_3})$ et $\mathcal{SP}^{dex} = \{N^{dx_2} = (3, 6), N^{dx_3} = (1, 5)\}$.



Deuxième phase

On initialise p à 1 et amorçons les itérations de l'algorithme de la méthode adaptée multi-critère de la manière suivante :

Itération 1 : Soit $M = 1000$, les contraintes (6.14) s'expriment comme suit :

$$\begin{aligned} l^{(1)} &= \left(1 \ 5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^t, \\ u^{(1)} &= \left(3 \ 6 \ M \ M \ M \ M \ M \ M \ M \right)^t. \end{aligned}$$

En appliquant l'algorithme 3 de la méthode adaptée, nous résolvons le PPLMO suivant :

$$\begin{cases} \max_x c_1 x \\ Ax \leq b \\ l^{(1)} \leq x \leq u^{(1)}, \end{cases}$$

en partant de la solution réalisable initiale $x^0 = \left(2, \frac{11}{2} \right)$, pour obtenir le compromis $\bar{x}^1 = \left(2, \frac{11}{2} \right)$ au premier niveau. Ensuite, nous choisissons $\mathbf{l}_{11} = \mathbf{r}_{11} = 0.5$.

Itération 2 : Nous mettons à jour les bornes des variables de décision comme suit :

$$\begin{aligned} l^{(2)} &= \left(\frac{3}{2} \ 5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^t, \\ u^{(2)} &= \left(\frac{5}{2} \ 6 \ M \ M \ M \ M \ M \ M \ M \right)^t. \end{aligned}$$

En partant de la solution réalisable initiale $x^0 = \left(\frac{5}{2}, \frac{23}{4} \right)$, nous résolvons le problème :

$$\begin{cases} \max_x c_2 x \\ Ax \leq b \\ l^{(2)} \leq x \leq u^{(2)}, \end{cases}$$

afin d'obtenir un compromis satisfaisant du PPLMO à deux niveaux (6.11) : $\bar{x}^2 = \left(\frac{5}{2}, \frac{23}{4} \right)$.

6.6 Dialogue Approfondi : Exploration et Analyses

6.6.1 La Méthode Adaptée vs La Méthode du Simplexe : Comparaison et Choix

La méthode du simplexe est une méthode numérique développée en 1947 pour résoudre des PPL, où la région réalisable S est construite à partir de contraintes linéaires de la forme " $Ax = b$ " et de contraintes de non-négativité " $x \geq 0$ ". Lorsque le programme linéaire inclut des



contraintes additionnelles sous la forme :

$$l \leq x \leq u, \quad (6.18)$$

nous qualifions le PPL de "problème de programmation linéaire à variables bornées". Pour résoudre ce type de problème, il est nécessaire de transformer les contraintes (6.18) en contraintes linéaires, ce qui entraîne généralement une augmentation de la taille de la matrice A . Cependant, R. Gabasov a développé la méthode adaptée pour résoudre ce genre de problèmes en traitant directement les contraintes (6.18) sans modifier la matrice A . Cette caractéristique a conféré à la méthode adaptée une efficacité supérieure à celle de la méthode du simplexe.

6.6.2 Processus Étape par Étape pour Résoudre les PPLMO-MNs à Variables Bornées : La Méthode Adaptée

Les problèmes de programmation à plusieurs niveaux se caractérisent par une structure hiérarchique, impliquant plusieurs décideurs cherchant un compromis. Ces décideurs sont classés par ordre de priorité, du Décideur du 1^{er} niveau (DCD_1) au P ^{ième}-décideur (le dernier).

Chaque décideur résout son propre problème multi-objectif. En fonction de la solution qu'il obtient, il impose des contraintes pour guider les étapes suivantes. Le DCD_1 délimite une région réalisable dans laquelle les solutions des décideurs suivants devront se trouver. Ensuite, il communique ces nouvelles contraintes au Décideur du 2^{ième} niveau (DCD_2), qui résout son propre problème et détermine une nouvelle région (sous-ensemble de celle définie par le décideur précédent). Ce processus se répète successivement jusqu'à ce qu'une solution soit trouvée. Ainsi, le processus de résolution se déroule comme suit : Pour commencer, nous transformons le problème (6.1) en un PPLMO-MN à variables bornées en définissant les bornes $l^{(1)}$ et $u^{(1)}$ des variables de

décision comme indiqué par la formule (6.13). Notons les composantes positives des vecteurs $l^{(1)}$ et $u^{(1)}$ comme suit :

$$l_{11}^{(1)}, \dots, l_{1n_1}^{(1)}, l_{21}^{(1)}, \dots, l_{2n_2}^{(1)}, \dots, l_{P1}^{(1)}, \dots, l_{Pn_P}^{(1)}, \\ u_{11}^{(1)}, \dots, u_{1n_1}^{(1)}, \dots, u_{2n_2}^{(1)}, \dots, u_{P1}^{(1)}, \dots, u_{Pn_P}^{(1)}.$$

Itération 1 : Le premier décideur entame le processus en optimisant sa fonction objectif (la première fonction objectif F_1) par rapport à la région réalisable S et aux contraintes (6.18). En d'autres termes, DCD_1 résout le problème multi-objectifs suivant :

$$\begin{cases} \max_x F_1(x) = c_1x \\ Ax \leq b \\ l^{(1)} \leq x \leq u^{(1)}, \end{cases} \quad (6.19)$$

pour obtenir sa solution satisfaisante notée par

$$\bar{x} = \left(\bar{x}_{11}^1, \bar{x}_{12}^1, \dots, \bar{x}_{1n_1}^1, \bar{x}_{21}^1, \dots, \bar{x}_{Pn_P}^1 \right).$$

Ensuite, DCD_1 impose de nouvelles contraintes aux variables de décision qui sont sous son contrôle :

$$\overset{c^1}{x}_{11}, \overset{c^1}{x}_{12}, \dots, \overset{c^1}{x}_{1n_1},$$

en choisissant les $2n_1$ nombres réels $l_{11}^{(2)}, l_{12}^{(2)}, \dots, l_{1n_1}^{(2)}, u_{11}^{(2)}, u_{12}^{(2)}, \dots, u_{1n_1}^{(2)}$ qui vérifient

$$\begin{aligned} l_{11}^{(1)} &\leq l_{11}^{(2)} < u_{11}^{(2)} \leq u_{11}^{(1)}, \\ l_{12}^{(1)} &\leq l_{12}^{(2)} < u_{12}^{(2)} \leq u_{12}^{(1)}, \\ &\vdots & \vdots & \vdots & \vdots & \vdots \\ l_{1n_1}^{(1)} &\leq l_{1n_1}^{(2)} < u_{1n_1}^{(2)} \leq u_{1n_1}^{(1)}. \end{aligned}$$

Par la suite, nous définissons les nouvelles contraintes de la manière suivante :

$$\begin{aligned} l_{11}^{(2)} &\leq x_{11} \leq u_{11}^{(2)}, \\ l_{12}^{(2)} &\leq x_{12} \leq u_{12}^{(2)}, \\ &\vdots & \vdots & \vdots & \vdots \\ l_{1n_1}^{(2)} &\leq x_{1n_1} \leq u_{1n_1}^{(2)}. \end{aligned} \quad (6.20)$$

Itération 2 : Le 2^{ième} décideur (DCD_2) poursuit le processus en optimisant sa fonction objectif (la deuxième fonction objectif F_2) par rapport à la région réalisable S et aux contraintes (6.18 et 6.20). Autrement dit, DCD_1 résout le problème multi-objectifs suivant :

$$\left\{ \begin{array}{l} \max_x F_2(x) = c_2x, \\ Ax \leq b, \\ l_{11}^{(2)} \leq x_{11} \leq u_{11}^{(2)}, \\ \vdots \\ l_{1n_1}^{(2)} \leq x_{1n_1} \leq u_{1n_1}^{(2)}, \\ l_{21}^{(1)} \leq x_{21} \leq u_{21}^{(1)}, \\ \vdots \\ l_{2n_2}^{(1)} \leq x_{2n_2} \leq u_{2n_2}^{(1)}, \\ \vdots \\ l_{Pn_P}^{(1)} \leq x_{Pn_P} \leq u_{Pn_P}^{(1)}, \end{array} \right. \quad (6.21)$$

pour obtenir sa solution satisfaisante :

$$\overset{c^2}{x} = \left(\overset{c^2}{x}_{11}, \overset{c^2}{x}_{12}, \dots, \overset{c^2}{x}_{1n_1}, \overset{c^2}{x}_{21}, \dots, \overset{c^2}{x}_{Pn_P} \right).$$

Ensuite, DCD_2 impose de nouvelles contraintes aux variables de décision qui sont sous son contrôle : $\overset{c^2}{x}_{21}, \overset{c^2}{x}_{22}, \dots, \overset{c^2}{x}_{2n_2}$, en choisissant les $2n_2$ nombres réels $l_{21}^{(3)}, \dots, l_{2n_2}^{(3)}, u_{21}^{(3)}, \dots, u_{2n_2}^{(3)}$ qui vérifient

$$\begin{aligned} l_{21}^{(1)} &\leq l_{21}^{(3)} < u_{21}^{(3)} \leq u_{21}^{(1)}, \\ &\vdots & \vdots & \vdots & \vdots & \vdots \\ l_{2n_2}^{(1)} &\leq l_{2n_2}^{(3)} < u_{2n_2}^{(3)} \leq u_{2n_2}^{(1)}. \end{aligned}$$

Par la suite, nous définissons les nouvelles contraintes de la manière suivante :

$$\begin{aligned} l_{21}^{(3)} &\leq x_{21} \leq u_{21}^{(3)}, \\ &\vdots & \vdots & \vdots & \vdots \\ l_{2n_2}^{(3)} &\leq x_{2n_2} \leq u_{2n_2}^{(3)}. \end{aligned} \quad (6.22)$$

Itération 3 : Le Décideur du 3^{ième} niveau (DCD_3) continue le processus en optimisant sa fonction objectif (la troisième fonction objectif F_3) par rapport à la région réalisable S et aux contraintes (6.18, 6.20 et 6.22). Autrement dit, DCD_3 résout le problème multi-objectifs suivant :

$$\left\{ \begin{array}{l} \max_x F_3(x) = c_3x, \\ Ax \leq b, \\ l_{11}^{(2)} \leq x_{11} \leq u_{11}^{(2)}, \\ \vdots \\ l_{1n_1}^{(2)} \leq x_{1n_1} \leq u_{1n_1}^{(2)}, \\ l_{21}^{(3)} \leq x_{21} \leq u_{21}^{(3)}, \\ \vdots \\ l_{2n_2}^{(3)} \leq x_{2n_2} \leq u_{2n_2}^{(3)}, \\ l_{31}^{(1)} \leq x_{31} \leq u_{31}^{(1)}, \\ \vdots \\ l_{Pn_P}^{(1)} \leq x_{Pn_P} \leq u_{Pn_P}^{(1)}, \end{array} \right. \quad (6.23)$$



pour obtenir sa solution satisfaisante :

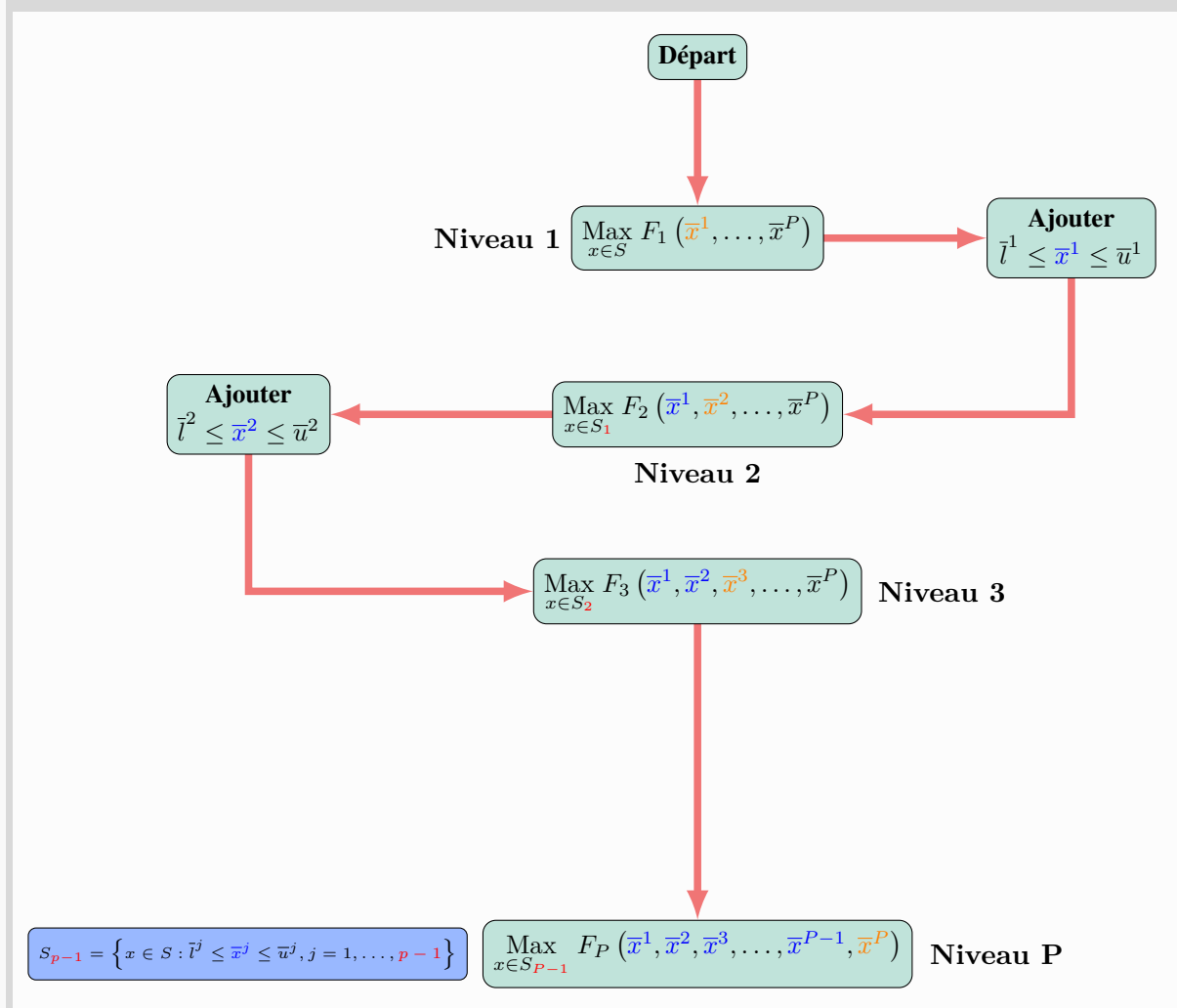
$$\bar{x}^3 = \left(c_{11}^3, c_{12}^3, \dots, c_{1n_1}^3, c_{21}^3, \dots, c_{Pn_P}^3 \right).$$

Ensuite, le DCD_3 impose de nouvelles contraintes aux variables de décision qui sont sous son contrôle. Ces contraintes seront prises en compte par les prochains décideurs : Décideur du 4^{ième} niveau (DCD_4), Décideur du 5^{ième} niveau (DCD_5), ..., P ^{ième} décideur.

Itération 4 $\rightarrow P$: Ce processus de résolution se poursuit jusqu'à ce qu'un compromis satisfaisant soit trouvé dans le P ^{ième}-niveau.

Enfin, la méthode adaptée continue de démontrer sa pertinence dans le domaine de la programmation linéaire en abordant une variété de problèmes. Cette méthode s'est avérée efficace pour résoudre des problèmes de programmation à plusieurs niveaux, en particulier les PPLMO-MN qui étaient l'objectif principal de cette étude. Il est à noter que contrairement aux approches antérieures, l'exigence de variables bornées dans le problème initial n'est plus nécessaire, car elles sont intégrées dans la construction de la méthode elle-même.

Approche de résolution d'un PPMO-MN à P -niveaux.



6.7 Conclusion

Ce chapitre se déploie en deux volets majeurs, chacun apportant une contribution significative. La première partie propose une approche novatrice pour la génération exhaustive de tous les points extrêmes non-dominés d'un PPLMO-MN. Par distinction avec l'approche de P.L. Yu et Zeleny décrite dans [Yu 75], notre méthode repose sur une localisation des faces de la région réalisable par leurs points extrêmes plutôt que par l'ensemble des équations qui les définissent. L'amorce de notre démarche consiste en la construction de l'ensemble complet de toutes les faces ayant leurs points extrêmes dans l'ensemble des points extrêmes non-dominés. Ensuite, nous développons un critère de non-dominance pour évaluer ces faces, en combinant habilement la proposition 6.4.4 et le théorème 6.4.5. Cette étape pave la voie à un algorithme dédié à la localisation précise de l'ensemble des faces non-dominées. La méthodologie est illustrée à travers un exemple de problème de programmation linéaire multi-objectifs à deux niveaux, concrétisant ainsi son application concrète.

Dans la deuxième partie, la méthode adaptée pour résoudre les PPLMO est étendue pour aborder le cas plus complexe des PPLMO-MN. À cet effet, nous exposons un algorithme spécifique qui cible la recherche d'un unique compromis satisfaisant, au sein d'un sous-ensemble convexe de \hat{N} que nous baptisons "ensemble de tri". Cette dénomination s'explique par le fait que l'ensemble de tous les compromis envisageables peut se subdiviser en plusieurs ensembles de tri distincts. En conclusion, nous illustrons notre approche au moyen de l'exemple préalablement évoqué. Il est important de noter que le calcul des points extrêmes est réalisé manuellement, tandis que la résolution des programmes linéaires mono-objectifs est accomplie à l'aide de MATLAB R2007b, avec la mise en œuvre tant de la méthode du simplexe que de la méthode adaptée.



UNE ANALYSE APPROFONDIE, DES CONCLUSIONS ÉCLAIRÉES ET DES PERSPECTIVES RAFFINÉES

1 Conclusions de l'Étude

Dans cette thèse, nous avons présenté une approche améliorée pour résoudre des problèmes linéaires multi-niveaux à l'aide d'une méthode purement numérique. De plus, nous avons introduit une nouvelle notion importante, à savoir l'ensemble des compromis potentiels dans un contexte multi-niveaux. Nous avons mis en lumière cette notion dans la section 6.4 en détaillant une méthode permettant de générer cet ensemble.

Nous avons développé une approche géométrique novatrice pour analyser la stabilité de la solution optimale d'un programme linéaire mono-objectif face à des variations des coefficients de deux variables de décision dans la fonction objectif. Cette approche nous a permis d'étudier simultanément la stabilité en modifiant deux paramètres. Nous avons formulé un problème équivalent qui se prête à une analyse de sensibilité, lequel est résolu à l'aide de calculs simples. En fixant un coin de l'espace réalisable, nous avons identifié toutes les formes linéaires dont les valeurs optimales sont atteintes en ce point. Cette démarche géométrique revient à déterminer des hyperplans dans l'espace tridimensionnel \mathbb{R}^3 , où les changements des coefficients de la fonction objectif sont assimilables à une rotation du plan.

Pour aborder ces concepts, nous avons tout d'abord examiné un problème de programmation linéaire, notamment la maximisation d'une forme linéaire (3.1). Ensuite, nous avons présenté une formulation géométrique originale pour ce problème, exploitée pour établir une équivalence avec l'analyse de sensibilité. Cette dernière vise à définir un intervalle angulaire au sein duquel



le vecteur des coefficients de la fonction objectif peut varier. Cela permet de déterminer toutes les formes linéaires qui atteignent leurs valeurs optimales sur la solution optimale initiale. Nous avons également envisagé la fonction objectif sous l'angle d'un plan dans \mathbb{R}^3 , tandis que les contraintes linéaires ont été envisagées comme un sous-ensemble convexe de cet espace tridimensionnel.

Nous avons amélioré l'approche proposée par S.B. Sinha et S. Sinha pour la résolution d'un PPLMO-MNⁱⁱⁱ, en utilisant le principe de la méthode adaptée.

Initialement, nous avons développé une application appelée Application de Réduction des Intervalles, contrôlée par le niveau précédent, qui vise à restreindre les intervalles des variables de décision. Ensuite, nous avons utilisé cette application pour établir une nouvelle estimation de sous-optimalité spécifique au problème en question. En conséquence, deux versions distinctes d'un algorithme ont été élaborées pour résoudre des problèmes de programmation linéaire mono-objectif à niveaux multiples.

La première version de l'algorithme met en évidence l'impact de la nouvelle estimation de sous-optimalité, tandis que la deuxième version démontre l'utilisation exclusive de l'application de réduction des intervalles. Ces deux versions de l'algorithme sont divisées en trois étapes. Tout d'abord, nous calculons séparément les solutions pour chaque objectif de manière indépendante. Ensuite, nous déterminons les bornes des variables de décision. Enfin, nous entamons la recherche d'un compromis satisfaisant en résolvant $P - 1$ problèmes de programmation linéaire mono-objectif à variables bornées. Cela est réalisé en appliquant l'algorithme 2 de la méthode adaptée mono-critère.

Nous avons conçu une nouvelle approche pour résoudre les PPLMO-MNs. Dans cette optique, nous avons exploité la version généralisée de l'algorithme de la méthode du simplexe multi-critère développée par P.L. Yu et M. Zeleny [Yu 74; Yu74]. Cela nous a permis de créer un nouvel algorithme capable de générer l'ensemble de tous les compromis envisageables pour un PPLMO-MN donné.

Par la suite, en utilisant cet ensemble de compromis comme nouvelle région réalisable, nous avons élaboré un second algorithme. Celui-ci s'appuie sur le principe de la méthode adaptée pour résoudre un PPLMO-MN en sélectionnant un compromis unique parmi ces possibilités.

Cette thèse met en évidence l'importance continue de la méthode adaptée en programmation linéaire, illustrée à travers sa pertinence dans une diversité de problèmes, notamment les PPLMO-MNs, qui ont constitué l'objectif principal de ce travail de recherche. Dans ce contexte, la méthode adaptée se distingue par sa capacité à traiter ces problèmes, même lorsque la présence de variables bornées n'est plus une contrainte obligatoire, mais plutôt un élément intégré

iii. Problème de Programmation Linéaire Multi-Objectif Multi-Niveaux.



dans la structure de notre méthode.

Par ailleurs, notre étude démontre clairement que l'adoption d'une approche purement géométrique dans l'analyse des problèmes de programmation linéaire ouvre de nouvelles perspectives. Elle permet le développement de méthodes novatrices, susceptibles de fournir des solutions globales et d'aboutir à des échappatoires inédites dans la résolution de ces problèmes complexes.

2 Au cœur des débats : Une analyse approfondie et critique des résultats obtenus

Les résultats de notre étude ont permis de mettre en évidence plusieurs aspects clés des méthodes proposées et de leurs applications dans le contexte de la programmation linéaire multi-objectifs à niveaux multiples.

Tout d'abord, le Résultat [Kac22c], exposé dans le chapitre 6, met en lumière un inconvénient majeur de la méthode en termes de temps de calcul et de mémoire. En générant toutes les solutions possibles, elle peut rencontrer des limitations en matière d'efficacité pour les problèmes de grande taille ou complexes. Par conséquent, il est essentiel de prendre en compte ces contraintes lors de l'application de la méthode à des cas spécifiques.

Dans le même chapitre, nous avons détaillé le Résultat [Kac23c], qui démontre comment notre méthode répond à notre besoin d'explorer l'ensemble des solutions en utilisant la méthode adaptée pour trouver un seul compromis satisfaisant. Cette approche nous permet d'atteindre une solution optimale tout en évitant les coûts computationnels excessifs associés à la génération de multiples solutions. En adoptant cette stratégie, nous parvenons à concilier efficacité et précision dans la résolution des problèmes complexes de programmation linéaire multi-objectif. Les Résultats [Kac22a], présenté dans le chapitre 3, et [Kac23a], présenté dans l'appendice II, démontrent une caractéristique intéressante de notre approche. Elle est capable de détecter la présence d'une solution idéale dans certains cas, évitant ainsi la nécessité de recourir à d'autres méthodes plus complexes ou gourmandes en ressources.

Enfin, le Résultat [Kac22b], présenté dans le chapitre 5, affirme la validité de notre approche simple, en particulier dans le cas des PPL(MO)O-MN. En fournissant un seul compromis optimal, notre méthode s'avère être une alternative efficace aux méthodes mentionnées précédemment [Kac22c ; Kac23c], tout en conservant une simplicité conceptuelle et une applicabilité pratique.

En les reliant, ces résultats démontrent la faisabilité d'une approche unifiée. En combinant ces éléments dans un seul algorithme, nous pourrions obtenir une méthode plus robuste et puissante pour résoudre un large éventail de problèmes de programmation linéaire à plusieurs objectifs et niveaux, tout en maintenant un équilibre entre précision et efficacité computationnelle.

3 Bilan et perspectives : Les conclusions clés de notre étude en optimisation multi-critère et les pistes pour de futures recherches

Le développement de méthodes rapides et efficaces pour la résolution des **PPLMO-MNs** demeure un défi majeur. La plupart des travaux de recherche existants se concentrent principalement sur les **PPMOs** à deux niveaux, laissant une marge d'amélioration pour les niveaux supérieurs. Cependant, afin que les méthodes développées dans cette étude deviennent des outils encore plus robustes et fiables, des améliorations continues et des tests approfondis sont indispensables. La communauté scientifique bénéficierait grandement de la validation rigoureuse de ces méthodes sur diverses classes de problèmes, ce qui contribuerait à renforcer leur validité et à identifier les éventuelles limites. Ces améliorations sont essentielles pour garantir que ces méthodes peuvent être largement adoptées et utilisées dans des situations réelles.

Les différentes versions de la méthode adaptée de Gabasov (mono-critère, multi-critère, multi-niveau, développées dans le cadre de cette étude, etc.) continuent de jouer un rôle essentiel dans la résolution de divers problèmes. Notre étude a démontré leur applicabilité à une variété de situations, en particulier aux **PPLMO-MN**, qui étaient au cœur de notre travail. Une réalisation significative est que la contrainte de présence de variables bornées, qui était courante dans les approches précédentes, n'est plus une exigence. Cela élargit considérablement la portée d'application de ces méthodes, les rendant encore plus polyvalentes et adaptées à un éventail plus large de problématiques réelles.

D'autre part, l'approche géométrique adoptée dans notre analyse a été révélatrice de nouvelles perspectives dans la résolution des problèmes linéaires. En mettant en évidence l'importance de considérer les aspects géométriques des contraintes et des objectifs, nous avons ouvert de nouvelles possibilités pour le développement de méthodes générales et globales. Cette approche a permis une compréhension plus profonde des interactions entre les différentes composantes du problème, conduisant à des solutions novatrices et potentiellement plus performantes.

En outre, une perspective passionnante pour l'avenir serait la généralisation de notre approche géométrique d'analyse de sensibilité à des problèmes de dimensions supérieures. Cette extension pourrait permettre d'aborder des problèmes plus complexes tout en préservant la simplicité conceptuelle qui caractérise notre méthode. En développant des méthodes qui puissent traiter des espaces de décision de plus grande dimension, nous pourrions ouvrir la voie à des solutions innovantes pour des problèmes de plus grande envergure et plus complexes, apportant ainsi une contribution significative à la programmation linéaire et à ses applications pratiques.

En somme, bien que cette étude ait réalisé des avancées significatives, il reste encore beaucoup à explorer et à perfectionner dans le domaine des **PPLMO-MN**. Les perspectives futures sont vastes et offrent des opportunités pour des améliorations et des développements novateurs.



SUPLÉMENTS ET ANNEXES

Sommaire

I	Génération de l'Ensemble N^{dex} : Méthode de P.L. Yu et M. Zeleny	140
II	Une Nouvelle Approche Géométrique pour les PPLMOs	142
II.1	Introduction	143
II.2	Formulation du Problème	143
II.3	Classification des problèmes de PLMO	145
II.4	Illustration Numérique d'un PPLMO en Utilisant les Calculs de l'Exemple 3.6 du Chapitre 3 : Classification des Formes Linéaires par Rapport à la Région Réalizable S définie par 28	146
II.5	Conclusion	147
III	Exemple pour Illustrer les Faces d'un Polyèdre dans \mathbb{R}^3	148

I Génération de l'Ensemble N^{dex} : Méthode de P.L. Yu et M. Zeleny

Nous présentons la méthode de P.L. Yu et M. Zeleny décrite dans [Yu 74] pour générer l'ensemble de tous les points extrêmes non-dominés N^{dex} d'un PPLMO, afin de faciliter sa construction.

Définition I.1. Soit J_B^1 une base. Nous créons une nouvelle base en insérant la $j^{ième}$ colonne de J_N^1 dans J_B^1 , comme décrit dans l'algorithme de la méthode du simplexe. Cela crée une nouvelle base $J_B^2 = J_B^1 - \{i\} \cup \{j\}$. Les bases J_B^1 et J_B^2 sont nommées bases adjacentes, et les solutions de base réalisables qui en résultent, si distinctes, sont appelées deux points extrêmes adjacents de S .

Pour toute base J_B , $\mathcal{D}(J_B)$ représente l'ensemble des bases dominées adjacentes, et $\mathcal{A}(J_B)$ est l'ensemble des bases adjacentes à J_B mais excluant celles de $\mathcal{D}(J_B)$.



Objectif de la Méthode : Générer l'ensemble N^{dex} .

Les étapes détaillées de la méthode de P.L. Yu et M. Zeleny [Yu 74]

Étape 1 - Initialisation :

- ◇ Trouver un point extrême non-dominé N^{dex} du PPLMO.
- ◇ Sélectionner une base de support initiale J_B^1 correspondant à ce point extrême.
- ◇ Calculer l'ensemble $\mathcal{D}(J_B^1)$, qui contient toutes les bases dominées adjacents à J_B^1 .
- ◇ Calculer $\mathcal{A}(J_B^1)$, qui contient toutes les bases adjacentes à J_B^1 qui ne sont pas dominées.

Étape 2 - Initialisation des ensembles :

- ◇ Initialiser l'ensemble H_1 avec la base J_B^1 .
- ◇ Initialiser l'ensemble D_1 avec $\mathcal{D}(J_B^1)$.
- ◇ Initialiser l'ensemble W_1 avec $\mathcal{A}(J_B^1)$.

Étape 3 - Itérations :

- ◇ Initialiser $i \leftarrow 1$.
- ◇ Répéter les étapes suivantes jusqu'à ce que l'ensemble W_i soit vide.

Étape 4 - Vérification de l'arrêt :

- ◇ Si W_i est vide, arrêter le processus et retourner l'ensemble $N^{dex} = H_1$.
- ◇ Sinon, passer à l'étape suivante.

Étape 5 - Sélection de la prochaine base : Sélectionner une base $K \in W_i$.

Étape 6 - Vérification si K est un N^{dex} -base :

- ◇ Vérifier si la base K est elle-même une N^{dex} -base (c'est-à-dire une base qui définit un point extrême non-dominé). Si c'est le cas, mettre à jour les ensembles D_i et W_i en conséquence, puis retourner à l'étape 4.
- ◇ Si K n'est pas une N^{dex} -base, passer à l'étape suivante.

Étape 7 - Calcul de $\mathcal{D}(K)$ et $\mathcal{A}(K)$:

- ◇ Calculer $\mathcal{D}(K)$ qui contient les bases dominées adjacents à K .
- ◇ Calculer $\mathcal{A}(K)$ qui contient les bases adjacentes à K mais qui ne sont pas dominées.

Étape 8 - Mise à jour des ensembles :

- ◇ Mettre à jour D_{i+1} en ajoutant les bases de $\mathcal{D}(K)$.
- ◇ Mettre à jour W_{i+1} en ajoutant les bases de $\mathcal{A}(K)$ qui ne sont pas déjà dans D_{i+1} .
- ◇ Ajouter la base K à l'ensemble H_i .

Étape 9 - Passage à l'itération suivante : Poser $i \leftarrow i + 1$ et retourner à l'étape 4.





Ce processus itératif se poursuit jusqu'à ce que l'ensemble W_i soit vide, à ce moment-là, l'ensemble complet des points extrêmes non-dominés N^{dex} aura été généré. Chaque itération explore différentes bases et leurs relations, en ajoutant progressivement des bases dominées et adjacentes pour construire l'ensemble complet des points extrêmes non-dominés.

II Une Nouvelle Approche Géométrique pour les PPL-MOs

Résumé

Dans cet article ^{iv}, nous présentons une méthode novatrice pour résoudre les PPL-MOs qui évite le besoin de calculer la valeur optimale de chaque fonction objectif. Cette méthode fait suite à notre travail précédent sur l'analyse de sensibilité, où nous avons développé une nouvelle approche géométrique. La première étape de notre démarche consiste à diviser l'espace des formes linéaires en un nombre fini d'ensembles basés sur un sous-ensemble polygonal convexe fixe de \mathbb{R}^2 . Ceci est accompli à l'aide d'une relation d'équivalence, qui garantit que tous les éléments d'une classe d'équivalence donnée ont la même solution optimale.

Nous caractérisons ensuite les classes d'équivalence de l'ensemble quotient à l'aide d'une approche géométrique de l'analyse de sensibilité. Cette étape est cruciale pour identifier la solution idéale au PPLMO. Grâce à cette approche, nous pouvons déterminer si un PPLMO donné a une solution idéale sans avoir besoin de calculer la valeur optimale de chaque fonction objectif. Cela constitue une amélioration significative par rapport aux méthodes existantes, car cela réduit considérablement la complexité computationnelle et le temps requis pour résolution. Pour illustrer notre méthode, nous présentons un exemple numérique qui en démontre l'efficacité.

iv. Cet article a été accepté pour publication dans le 51^{ième} volume, 1^{ième} numéro de la revue "Mathematica Applicanda". Vous pouvez également trouver la version préimpression de l'article sur arXiv en suivant la référence [Kac23b].

Mots clés

Programmation linéaire , analyse sensitive, géométrie affine, programmation multiobjectifs.



Cela fait suite à notre travail antérieur sur l'analyse de sensibilité, où nous avons élaboré une nouvelle approche géométrique. Pour des détails plus approfondis et pour les préliminaires nécessaires, veuillez vous référer à [Kac22a] ou consulter le chapitre 3.



II.1 Introduction

La PL consiste à optimiser des programmes mathématiques avec des fonctions objectifs et des contraintes linéaires [Dan63; Dan63; Kac22c; Sto70; Kuc05]. Cependant, les modifications des données du marché peuvent nécessiter des mises à jour des coefficients initiaux du modèle, ce qui fait de l'analyse de sensibilité une partie essentielle de la PL. L'analyse de sensibilité est utilisée pour illustrer l'intervalle des paramètres du programme linéaire pour lesquels la solution du problème initial reste stable [Dar21; Fil11; Kol11; Zim76; War90].

Des techniques de la PMO ont été développées pour aider le DCD à trouver un compromis acceptable [Sha16; Man17; Siv19]. Cependant, rien n'empêche que tous les objectifs soient en accord (atteignant une valeur optimale dans une solution commune réalisable). Pour cela, nous développerons une approche pour détecter la présence du conflit. Ensuite, nous les classerons en fonction de leur solution, la classification dépendra de la région réalisable.

Ce travail propose une approche permettant de déterminer si un PPLMO spécifique est en conflit ou non. En d'autres termes, nous développons une méthode indiquant si toutes les fonctions objectifs qui composent un PPLMO atteignent leur maximum dans un même point (admettant une solution idéale). Cette approche s'appuie sur l'analyse de sensibilité décrite dans [Kac22a] et peut être considérée comme une application de cette dernière.

Nous commençons par définir une relation d'équivalence dans l'espace des formes linéaires, que nous divisons ensuite en un nombre fini de classes d'équivalence où tous les éléments atteignent leur maximum au même point.

En résumé, nous présentons un nouveau résultat pour classer les PPLMOs dans une région réalisable fixée.

II.2 Formulation du Problème

Formulation du PPLMO initial

Considérons le problème de programmation linéaire multi-objectif initial suivant :

$$\text{Max}_{x \in S} F^0(x) = \text{Max}_{x \in S} (f_1^0(x), f_2^0(x), \dots, f_K^0(x)), \quad K \geq 2. \quad (24)$$

Sous les contraintes suivantes : $x \in S := \{x \in \mathbb{R}^2 : Ax \leq b, x \geq 0\}$.

Où

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad c_{0k} = \begin{pmatrix} c_1^{0k} \\ c_2^{0k} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$



Le problème d'optimisation donné a K fonctions objectifs, notées $f_k(x) = c_{0k}^T x$ pour $k = 1, \dots, K$. Les coefficients de la fonction objectif pour la k -ème fonction objectif sont représentés par le vecteur colonne c_{0k} , et les constantes associées aux première et deuxième variables de décision sont c_1^{0k} et c_2^{0k} respectivement. Les variables de décision sont représentées par un vecteur x de taille 2×1 . Les contraintes sont représentées par une matrice A de taille $m \times 2$ avec des coefficients constants et un vecteur b de taille $m \times 1$ avec des valeurs constantes. Le nombre de contraintes est représenté par m .

Définition II.1 (Solution idéale d'un PPLMO). Une solution idéale x^0 d'un PPLMO est un point qui appartient à la région réalisable S et qui satisfait un ensemble de conditions. Plus précisément, pour chaque fonction objectif du PPLMO, x^0 doit représenter la valeur maximale réalisable dans la région réalisable. En d'autres termes, une solution idéale représente le meilleur résultat possible pour toutes les fonctions objectifs simultanément. Le concept de solution idéale est important en optimisation multi-objectif car il fournit une référence pour évaluer la qualité d'autres solutions réalisables. Autrement dit, x^0 est une solution idéale du PPLMO si et seulement si

$$f_k^0(x^0) \geq f_k^0(x) \quad \forall x \in S, \quad \forall k = 1, \dots, K.$$

Problème II.2. Soit x^0 une solution optimale qui maximise la fonction objectif $f_{k_0}^0$ pour un certain $k_0 = 1, \dots, K$. Le problème est de déterminer toutes les applications linéaires $F \in \mathcal{L}(\mathbb{R}^2)^K$, qui est le produit de l'espace des applications linéaires de \mathbb{R}^2 vers lui-même, de telle manière que la valeur maximale de $f_k(x)$ soit atteinte en x^0 pour tout $k = 1, \dots, K$. Autrement dit, x^0 est la solution optimale qui maximise simultanément toutes les fonctions objectifs. Cela peut être réalisé si et seulement si le vecteur $F(x)$, composé des fonctions objectifs $f_1(x), f_2(x), \dots, f_K(x)$, est également maximisé en x^0 . Plus précisément, le problème est de déterminer toutes les applications linéaires :

$$F \in \mathcal{L}(\mathbb{R}^2)^K := \underbrace{\mathcal{L}(\mathbb{R}^2) \times \mathcal{L}(\mathbb{R}^2) \times \dots \times \mathcal{L}(\mathbb{R}^2)}_{K \text{ fois}}.$$

Telle que

$$F(x) = (f_1(x), f_2(x), \dots, f_K(x)),$$

et

$$x^0 = \arg \max_{x \in S} f_k(x), \quad \text{pour tous } k = 1, \dots, K. \quad (25)$$

II.3 Classification des problèmes de PLMO

Soit $g, h \in \mathcal{L}(\mathbb{R}^2)$. Définissons une relation d'équivalence sur $\mathcal{L}(\mathbb{R}^2)$, comme suit :

$$g \mathcal{R}_S h \Leftrightarrow \arg \max_{x \in S} g(x) = \arg \max_{x \in S} h(x). \quad (26)$$

Lemme II.3. La relation \mathcal{R}_S est une relation d'équivalence dans $\mathcal{L}(\mathbb{R}^2)$.

Démonstration. Pour tout $g, h, h_1, h_2 \in \mathcal{L}(\mathbb{R}^2)$, on a :

$$\diamond \mathcal{R}_S \text{ est réflexive, en effet : } g \mathcal{R}_S g \Leftrightarrow \arg \max_{x \in S} g(x) = \arg \max_{x \in S} g(x).$$

$\diamond \mathcal{R}_S$ est symétrique, en effet :

$$g \mathcal{R}_S h \Leftrightarrow \arg \max_{x \in S} g(x) = \arg \max_{x \in S} h(x) \Leftrightarrow \arg \max_{x \in S} h(x) = \arg \max_{x \in S} g(x) \Leftrightarrow h \mathcal{R}_S g.$$

$\diamond \mathcal{R}_S$ est transitive. En d'autres termes, si $g \mathcal{R}_S h_1$ et $h_1 \mathcal{R}_S h_2$, nous aurons :

$$\arg \max_{x \in S} g(x) = \arg \max_{x \in S} h_1(x) = \arg \max_{x \in S} h_2(x).$$

Ce qui implique

$$\arg \max_{x \in S} g(x) = \arg \max_{x \in S} h_2(x) \Leftrightarrow g \mathcal{R}_S h_2.$$

■

Définition II.4 (Caractérisation des classes d'équivalence).

1. La classe d'équivalence de $g \in \mathcal{L}(\mathbb{R}^2)$: $\bar{g} = \{h \in \mathcal{L}(\mathbb{R}^2, \mathbb{R}) : g \mathcal{R}_S h\}$.

2. L'ensemble quotient de \mathcal{R}_S : $\bar{\mathcal{L}}(\mathbb{R}^2) := \mathcal{L}(\mathbb{R}^2) / \mathcal{R}_S = \{\bar{g} : g \in \mathcal{L}(\mathbb{R}^2)\}$.

3. Soit x^1, \dots, x^I et F^1, \dots, F^J les coins, faces de S respectivement. Définissons

$$\bar{\mathcal{L}}_{faces}(\mathbb{R}^2) = \left\{ \bar{g} \in \bar{\mathcal{L}}(\mathbb{R}^2) : \forall h \in \bar{g}, \exists j \in \{1, \dots, J\}, \text{ telle que } F_j = \arg \max_{x \in S} h(x) \right\}$$

et

$$\bar{\mathcal{L}}_{coins}(\mathbb{R}^2) = \left\{ \bar{g} \in \bar{\mathcal{L}}(\mathbb{R}^2) : \forall h \in \bar{g}, \exists i \in \{1, \dots, I\}, \text{ telle que } x^i = \arg \max_{x \in S} h(x) \right\}.$$

Où, I et J sont le nombre de coins et de faces de S respectivement.

Propriété II.5. Le cardinal de $\bar{\mathcal{L}}_{coins}(\mathbb{R}^2)$, $\bar{\mathcal{L}}_{faces}(\mathbb{R}^2)$ et $\bar{\mathcal{L}}(\mathbb{R}^2)$ est égal à I , J et $I + J$, respectivement.



Théorème II.6.

1. Le problème (II.2) possède une solution idéale si et seulement s'il exist $\bar{g} \in \overline{\mathcal{L}}_{coins}(\mathbb{R}^2)$, avec $f_k \in \bar{g}$, pour tous les indices $k = 1, \dots, K$.
2. Plus généralement, soit \mathcal{B}_S l'ensemble comprenant tous les coins et faces de la région réalisable S , défini comme suit :

$$\mathcal{B}_S = \{x^1, \dots, x^I, F^1, \dots, F^J\}.$$

Alors, pour tout $X \in \mathcal{B}_S$ il existe un unique $\bar{g} \in \overline{\mathcal{L}}(\mathbb{R}^2)$ tel que :

$$X = \arg \max_{x \in S} h(x), \quad \text{pour tous } h \in \bar{g}.$$

Démonstration. Immédiates. ■

II.4 Illustration Numérique d'un PPLMO en Utilisant les Calculs de l'Exemple 3.6 du Chapitre 3 : Classification des Formes Linéaires par Rapport à la Région Réalisable S définie par 28

Énoncé

Considérons le problème de PLMO initial suivant :

$$\text{Max}_{x \in S} F^0(x) = \max_{x \in S} (f_1^0(x), f_2^0(x), \dots, f_K^0(x)), \quad K \geq 2. \quad (27)$$

Sous les contraintes suivantes :

$$S := \{x \in \mathbb{R}^2 : Ax \leq b, x \geq 0\}. \quad (28)$$

Où

$$f_1^0(x_1, x_2) = 2x_1 + 3x_2 \quad \text{et} \quad f_2^0, f_3^0, \dots, f_K^0 \in \mathcal{L}(\mathbb{R}^2)$$

et

$$A = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} \\ \frac{2}{5} & \frac{1}{5} \\ 0 & \frac{4}{5} \end{pmatrix}, \quad b = \begin{pmatrix} 40 \\ 40 \\ 40 \end{pmatrix}, \quad c_0 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Premièrement, nous résolvons le problème de programmation linéaire suivant :



$$\begin{cases} \max_{x_1, x_2} f_1^0(x_1, x_2) = 2x_1 + 3x_2 \\ A(x_1, x_2)^t \leq b \\ x_1 \geq 0; x_2 \geq 0, \end{cases} \quad (29)$$

en utilisant la méthode simplexe, nous obtenons $x^0 = (80, 40)$. Ensuite, on résout le problème suivant :

$$x^0 = \arg \max_{x \in S} f(x) = \arg \max_{x \in S} g(x), \quad (30)$$

en utilisant l'approche de l'analyse de sensibilité. on obtient les solutions suivantes :

$$\bar{g} = \left\{ h \in \mathcal{L}(\mathbb{R}^2) : h(\cdot) = r \langle (\cos(\phi), \sin(\phi)), \cdot \rangle, r > 0, \phi \in]26.5650^\circ, 63.4349^\circ[\right\}.$$

Pour plus de détails sur le calcul, voir [Kac22a].

Enfin, en appliquant le théorème II.6, nous montrons que le problème de PLMO (27) admet x^0 comme une solution idéale, si et seulement si $f_k^0 \in \bar{g}$, pour tous $k \in \{1, \dots, K\}$.

II.5 Conclusion

Dans cette étude, une approche novatrice a été développée pour déterminer une solution idéale pour un PPLMO. Initialement, une relation d'équivalence a été établie au sein de l'espace des formes linéaires $\mathcal{L}(\mathbb{R}^2)$, conduisant à la partition de cet espace en un ensemble fini de classes. Chacune de ces classes regroupe des éléments ayant atteint leurs valeurs maximales en un même point, offrant ainsi une structure de regroupement pertinente. Par la suite, des résultats significatifs ont été démontrés, révélant que si toutes les fonctions objectifs d'un PPLMO appartiennent à une même classe, alors la détermination de l'optimum se fait naturellement, exempt de calculs supplémentaires. Pour illustrer cette méthode novatrice, un exemple numérique concret a été présenté.

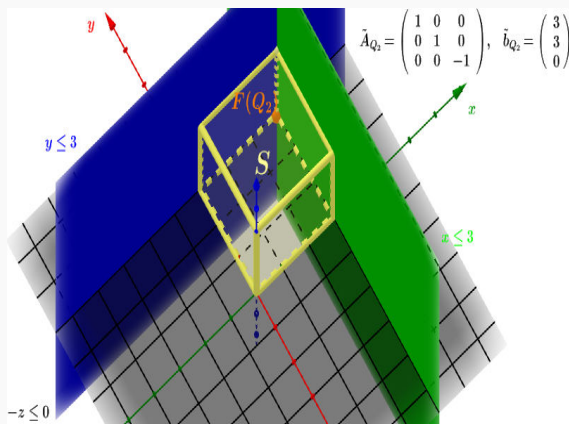


FIGURE 6 – Face de dimension 0 (point extrême).

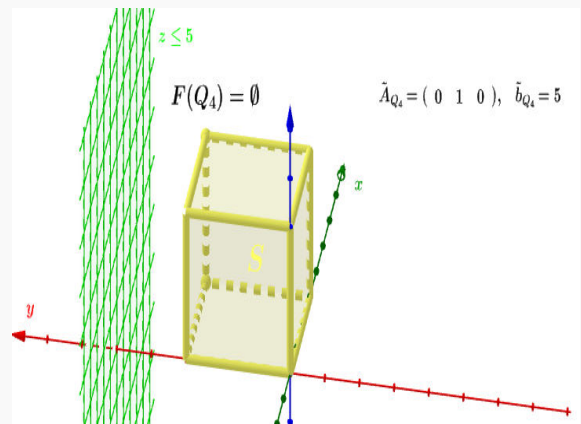


FIGURE 7 – Face vide.

III Exemple pour Illustrer les Faces d'un Polyèdre dans \mathbb{R}^3

Énoncé

Pour illustrer la définition et la notation des faces utilisées dans ce travail, considérons la région réalisable S définie par le système de contraintes (1.14), avec les matrices et vecteurs suivants :

$$\tilde{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \tilde{b} = \begin{pmatrix} 3 \\ 3 \\ 3 \\ 5 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

ainsi que les ensembles $Q_1 = 1, 2$, $Q_2 = 1, 2, 7$, $Q_3 = 2$ et $Q_4 = 4$.

À présent, nous pouvons visualiser les faces $F(Q_1)$, $F(Q_2)$, $F(Q_3)$ et $F(Q_4)$ à l'aide des figures 6, 7, 8, et 9 respectivement. Ces illustrations fournissent une meilleure compréhension de la notion de faces et de leur relation avec les ensembles Q_i dans le contexte de notre étude.

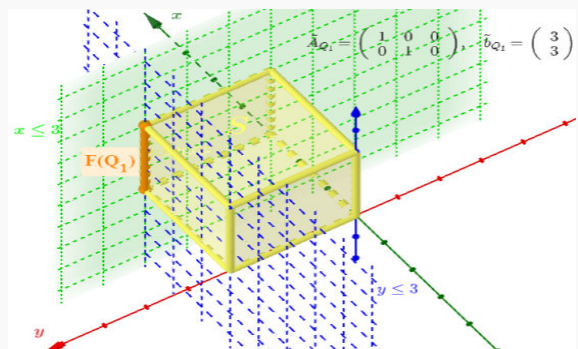


Figure 8 – Face de dimension 1.

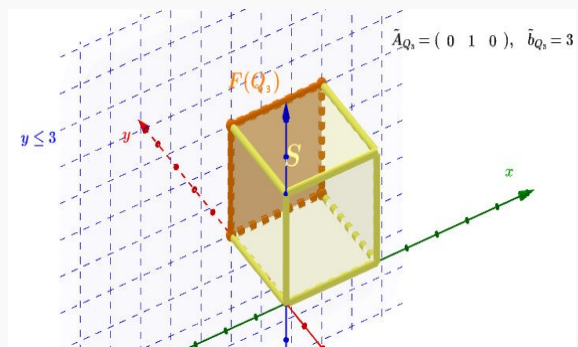


Figure 9 – Face de dimension 2.

LISTE DES PUBLICATIONS

Articles à comité de lecture (Mustapha Kaci)

Sonia Radjef. "A new geometric approach for sensitivity analysis in linear programming".
Mathematica Applicanda.

DOI: [10.14708/ma.v49i2.7112](https://doi.org/10.14708/ma.v49i2.7112)

Sonia Radjef. "The set of all the possible compromises of a multi-level multi-objective linear programming problem". Croatian Operational Research Review.

DOI: [10.17535/corr.2022.0002](https://doi.org/10.17535/corr.2022.0002)

À Paraître : Mes Articles Acceptés (Mustapha Kaci)

Sonia Radjef. "An adaptive method to solve multilevel multiobjective linear programming problems". Accepté pour le Vol. 33, No. 3 de la revue 'Operations Research and Decisions'. Préimpression disponible sur arXiv via le DOI..

DOI: [10.48550/arXiv.2210.14302](https://doi.org/10.48550/arXiv.2210.14302)

Sonia Radjef. "A new geometric approach to multiobjective linear programming problems". Accepté pour le Vol. 33, No. 3 de la revue 'Mathematica Applicanda'. Préimpression disponible sur arXiv via le DOI..

DOI: [10.48550/arXiv.2210.10829](https://doi.org/10.48550/arXiv.2210.10829)

Articles en préparation (Mustapha Kaci)

Sonia Radjef. "Solving a multilevel linear programming problems through a new constructive approach". Préimpression disponible sur arXiv via le DOI..

DOI: [10.48550/arXiv.2210.14269](https://doi.org/10.48550/arXiv.2210.14269)



Communications internationales (Mustapha Kaci)

Sonia Radjef. "*Sensitive analysis in linear programming and affine geometry in dimension 2*". International Seminar on Mathematics and Computer Science, Oran university (USTOMB), Algeria, 2021.

Sonia Radjef. "*The set of all non-dominated solutions for multi-level multi-objective linear programming problems and the adaptive method*". 14th International Conference on Multiple Objective Programming and Goal Programming (MOPGP), SKEMA Business School, France, 2021.

Sonia Radjef. "*An adaptive method to solve multilevel multiobjective linear Programming problems*". The 9th Maltepe University International Student Congress (MUISC-ENSAD), Maltepe University, Turkey, 2022.

Sonia Radjef. "*The adaptive method and multilevel multiobjective linear programming problems*". Colloque international MOAD'22 Méthodes Et Outils D'aide À la Décision, Béjaïa University , Algeria, 2022.

Communications nationales (Mustapha Kaci)

Sonia Radjef. "*Solving multi-level multi-objective linear programming problems through the adaptive method*". First national conference on mathematics and its applications, Laghouat university, Algeria, 2021.

Sonia Radjef. "*Multi-objective linear programming problems from a geometric point of view*". First national conference on pure and applied mathematics, Bourdj Bou Arréridj university, Algeria, 2021.

BIBLIOGRAPHIE

- [Abo10] M.A. ABO-SINNA ET I.A. BAKY. « Fuzzy Goal Programming Procedure to Bilevel Multiobjective Linear Fractional Programming Problems ». *International Journal of Mathematics and Mathematical Sciences* 2010 (2010), p. 1-15. DOI : [10.1155/2010/148975](https://doi.org/10.1155/2010/148975) (cf. p. [xix](#), [88](#)).
- [Ach84] S. ACHMANOV. *Programmation linéaire*. Mir, Moscou, 1984 (cf. p. [34](#)).
- [Ant16] C.H. ANTUNES ET M.J. ALVES ET J. CLÍMACO. *Multiobjective Linear and Integer Programming*. First. Springer, 2016. DOI : [10.1007/978-3-319-28746-1](https://doi.org/10.1007/978-3-319-28746-1) (cf. p. [xix](#), [78](#)).
- [Anz15] A. ANZI ET M.S. RADJEF. « A DC Algorithm for Solving Quadratic-linear Bilevel Optimization Problems ». *Modelling, Computation and Optimization in Information Systems and Management Sciences*. Springer International Publishing, 2015, p. 119-129 (cf. p. [xix](#), [88](#)).
- [Arm91] P. ARMAND ET C. MALIVERT. « Determination of the efficient set in multiobjective linear programming ». *Journal of optimization theory and applications* 70.3 (1991), p. 467-490. DOI : [10.1007/BF00941298](https://doi.org/10.1007/BF00941298) (cf. p. [86](#)).
- [Bak10] I.A. BAKY. « Solving multi-level multi-objective linear programming problems through fuzzy goal programming approach ». *Applied Mathematical Modelling* 34.9 (2010), p. 2377-2387. DOI : [10.1016/j.apm.2009.11.004](https://doi.org/10.1016/j.apm.2009.11.004) (cf. p. [xix](#), [88](#)).
- [Bel18] S. BELAHCENE ET P. MARTHON ET M. AIDENE. « The Pivot Adaptive Method for Solving Linear Programming Problems ». *American Journal of Operations Research* 8 (2018), p. 92-111. DOI : [10.4236/ajor.2018.82008](https://doi.org/10.4236/ajor.2018.82008) (cf. p. [xix](#), [110](#)).
- [Ben81] H.P. BENSON. « Finding an Initial Efficient Extreme Point for a Linear Multiple Objective Program ». *The Journal of the Operational Research Society* 32.6 (1981), p. 495-498. DOI : [10.1057/jors.1981.100](https://doi.org/10.1057/jors.1981.100) (cf. p. [86](#)).
- [Ben05] M. BENTOBACHE. *Nouvelle méthode pour la résolution des problèmes de programmation linéaire sous forme canonique et à variables bornées*. Mémoire de Magistère. Université de Béjaïa, 2005 (cf. p. [34](#)).
- [Bib85] M.O. BIBI. *Methods for solving linear-quadratic problems of optimal control*. PhD thesis. University of Minsk, 1985 (cf. p. [xxi](#)).
- [Bib96] M.O. BIBI. « Support method for solving a linear-quadratic problem with polyhedral with polyhedral constraints on control ». *Optimization* 37.2 (1996), p. 139-147. DOI : [10.1080/02331939608844205](https://doi.org/10.1080/02331939608844205) (cf. p. [xxi](#)).
- [Bib98] M.O. BIBI. *Cours de Post-graduation sur la méthode adaptée en programmation linéaire*. Université de Béjaïa, 1998 (cf. p. [34](#)).
- [Bou92] O. BOUARAB. *Une contribution sur la programmation linéaire multi-objectifs floue*. Thèse de master. Université de Tizi Ouzou, 1992 (cf. p. [79](#)).



- [Bra73] J. BRACKEN ET J.T. MCGILL. « Mathematical programs with optimization problems in the constraints ». *Operations Research* 21 (1973), p. 37-44. DOI : [10.1287/opre.21.1.37](https://doi.org/10.1287/opre.21.1.37) (cf. p. xix, xxi).
- [Bra74] J. BRACKEN ET J.T. MCGILL. « Technical Note—A Method for Solving Mathematical Programs with Nonlinear Programs in the Constraints ». *Operations Research* 22.5 (1974), p. 917-1133. DOI : [10.1287/opre.22.5.1097](https://doi.org/10.1287/opre.22.5.1097) (cf. p. xix, xxi).
- [Bra10] B. BRAHMI ET M.O. BIBI. « Dual support method for solving convex quadratic programs ». *Optimization* 59.6 (2010), p. 851-872. DOI : [10.1080/02331930902878341](https://doi.org/10.1080/02331930902878341) (cf. p. xxi, 87).
- [Col02] Y. COLLETTE ET P. SIARRY. *Optimisation multiobjectif*. Editions Eyrolles, Paris, 2002 (cf. p. 78).
- [Dae78] H.G. DAELLENBACH ET J.A. GEORGE. *Introduction to operations research technique*. Allyn et Bacon, Inc, Boston, MA. U.S.A, 1978 (cf. p. xx).
- [Dan63] G.B. DANTZIG. *Linear Programming and Extensions*. T. 10. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1963 (cf. p. xix, 35, 143).
- [Dan03] G.B. DANTZIG ET C. RICHARD. *The Basic*. Stanford University Press, 2003 (cf. p. xix, 35).
- [Dan97] G.B. DANTZIG ET M.N. THAPA. *Linear programming. Introduction*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 1997. DOI : [10.1007/b97672](https://doi.org/10.1007/b97672) (cf. p. xix, 35, 46, 47).
- [Dar21] D. DARVISHI, F. POUROFOGHI, J. Y. FORREST. « Sensitivity analysis of grey linear programming for optimization problems ». *Operations Research and Decisions* 31.4 (2021), p. 35-52. DOI : [10.37190/ord210402](https://doi.org/10.37190/ord210402) (cf. p. 143).
- [De 03] D. DE WERRA ET T.M. LIEBLING ET J.-F HËCHE. *Recherche opérationnelle pour ingénieur I*. Presses polytechniques et universitaires romandes, 2003, p. 402 (cf. p. 78).
- [Del18a] S.Z. DELHOUM. *La minimisation des polynômes et la méthode adaptée du support*. Thèse de Doctorat. Université d'Oran 1 Ahmed Ben-Bella, 2018, p. 156 (cf. p. xxi, 34).
- [Del18b] S.Z. DELHOUM ET S. RADJEF ET F. BOUDAUD. « Generation of efficient and epsilon-efficient solutions in multiple objective linear programming ». *Turkish Journal of Mathematics* 42 (2018), p. 1031-1048. DOI : [10.3906/mat-1702-12](https://doi.org/10.3906/mat-1702-12) (cf. p. xix, xxi, 57, 87).
- [Eav71] B.C. EAVES. « On Quadratic Programming ». *Management Science* 17.11 (1971), p. 698-711. DOI : [10.1287/mnsc.17.11.698](https://doi.org/10.1287/mnsc.17.11.698) (cf. p. 57).
- [Eck78a] J.G. ECKER ET I.A. KOUADA. « Finding efficient points for linear multiobjective problem ». *Mathematical Programming* 14 (1978), p. 249-261. DOI : [10.1007/BF01588968](https://doi.org/10.1007/BF01588968) (cf. p. 86).
- [Eck78b] J.G. ECKER ET N.S. HEGNER. « On computing an initial efficient extreme point ». *Journal of Operational Research Society* 29.10 (1978). DOI : [10.1057/jors.1978.214](https://doi.org/10.1057/jors.1978.214) (cf. p. 86).
- [Eck80] J.G. ECKER ET N.S. HEGNER. « Generating the efficient set for multiple objective linear programs ». *Journal of Optimization Theory and Applications* 30.3 (1980), p. 353-381. DOI : [10.1007/BF00935493](https://doi.org/10.1007/BF00935493) (cf. p. 86).
- [Ehr05] M. EHRGOTT. *Multicriteria Optimization*. Deuxième édition. Springer, 2005. DOI : [10.1007/3-540-27659-9](https://doi.org/10.1007/3-540-27659-9) (cf. p. xix, 78, 85).
- [Eva73] J.P. EVANS ET R.E. STEUER. « A revised simplex method for linear multiple objective programs ». *Mathematical Programming* 5 (1973), p. 54-72. DOI : [10.1007/BF01580111](https://doi.org/10.1007/BF01580111) (cf. p. 47).
- [Fil11] C. FILIPPI. *Sensitivity Analysis in Linear Programming*. John Wiley et Sons, Ltd, 2011. DOI : [10.1002/9780470400531.eorms0760](https://doi.org/10.1002/9780470400531.eorms0760) (cf. p. 74, 143).
- [Fli02] J. FLIEGE ET A. HESELEER. « Constructing approximations to the efficient set of convex quadratic multiobjective problems ». *Optimization online. University Dortmund* (2002) (cf. p. 18).
- [Gab78] R. GABASOV ET F.M. KIRILLOVA. *Linear Programming methods*. House, Minsk, 1978 (cf. p. xix, xxi, 57, 87).
- [Gab80] R. GABASOV ET F.M. KIRILLOVA ET O.I. KOSTJUKOVA. *Approximate Methods of Solving General Linear Programming Problem*. Reports of the BSSR Acad. of Sci, 1980, p. 680-683 (cf. p. xix, 57).



- [Gab95] R. GABASOV ET F.M. KIRILLOVA ET S.V. PRISCHEPOVA. *Optimal Feedback Control*. Lecture Notes in Control and Information Sciences. Springer, Berlin, Heidelberg, 1995. DOI : [10.1007/BFb0030998](https://doi.org/10.1007/BFb0030998) (cf. p. [xxi](#), [57](#), [59](#), [87](#), [101](#), [107](#)).
- [Gal77] T. GAL. « A general method for determining the set of all efficient solutions to a linear vectormaximum problem ». *European Journal of Operational Research* 1.5 (1977), p. 307-322. DOI : [10.1016/0377-2217\(77\)90063-7](https://doi.org/10.1016/0377-2217(77)90063-7) (cf. p. [78](#)).
- [Gar94] L.R. GARDINER ET R.E. STEUER. « Unified interactive multiple objective programming ». *European Journal of Operational Research* 74.3 (1994), p. 391-406. DOI : [10.1016/0377-2217\(94\)90219-4](https://doi.org/10.1016/0377-2217(94)90219-4) (cf. p. [79](#)).
- [Had91] A.G. HADIGHEH ET O. RAMANKO ET T. TERLAKY. « Sensitivity analysis in convex quadratic optimization ». *Advanced optimization laboratory* (1991) (cf. p. [63](#)).
- [Iqb10] H. IQBAL ET R.G. MATTOO. « Multiobjective Duality in Variational Problems with Higher Order Derivatives ». *Communications and Network* 2.2 (2010), p. 138-144. DOI : [10.4236/cn.2010.22021](https://doi.org/10.4236/cn.2010.22021) (cf. p. [79](#)).
- [Ise77] H. ISERMANN. « The Enumeration of the Set of All Efficient Solutions for a Linear Multiple Objective Program ». *Journal of the Operational Research Society* 28.3 (1977), p. 711-725. DOI : [10.1057/jors.1977.147](https://doi.org/10.1057/jors.1977.147) (cf. p. [78](#)).
- [Kac21a] M. KACI ET S. RADJEF. *L'analyse sensitive en programmation linéaire et la géométrie affine en dimension 2*. Algerian Journal of Engineering Architecture et Urbanism (AJEAU), Algérie, 2021 (cf. p. [62](#)).
- [Kac21b] M. KACI ET S. RADJEF. *Multi-objective linear programming problems from a geometric point of view*. Université Amar Telidji Laghouat, Algérie, 2021 (cf. p. [xxi](#), [62](#)).
- [Kac21c] M. KACI ET S. RADJEF. *Solving multi-level multi-objective linear programming problems through the adaptive method*. Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj, Algérie, 2021 (cf. p. [34](#)).
- [Kac22a] M. KACI ET S. RADJEF. « A new geometric approach for sensitivity analysis in linear programming ». *Mathematica Applicanda*. Annales Societatis Mathematicae Polonae, Series III 49.2 (2022), p. 145-157. DOI : [10.14708/ma.v49i2.7112](https://doi.org/10.14708/ma.v49i2.7112) (cf. p. [xxiv](#), [62](#), [63](#), [138](#), [142](#), [143](#), [147](#)).
- [Kac22b] M. KACI ET S. RADJEF. « Solving multilevel linear programming problems through new constructive approach ». *En cours d'examen par une revue* (2022). DOI : [10.48550/arXiv.2210.14269](https://doi.org/10.48550/arXiv.2210.14269) (cf. p. [87](#), [138](#)).
- [Kac22c] M. KACI ET S. RADJEF. « The set of all the possible compromises of a multi-level multi-objective linear programming problem ». *Croatian Operational Research Review* 13.1 (2022), p. 13-30. DOI : [10.17535/croorr.2022.0002](https://doi.org/10.17535/croorr.2022.0002) (cf. p. [xix](#), [xxiv](#), [88](#), [112](#), [128-130](#), [138](#), [143](#)).
- [Kac23a] M. KACI ET S. RADJEF. « A new geometric approach to multiobjective linear programming problems ». *Ce travail sera bientôt publié dans le journal "Mathematica Applicanda", au sein du Volume 51, Numéro 1* (2023). DOI : [10.48550/arXiv.2210.10829](https://doi.org/10.48550/arXiv.2210.10829) (cf. p. [78](#), [138](#)).
- [Kac23b] M. KACI ET S. RADJEF. « A new geometric approach to multiobjective linear programming problems ». *Ce travail sera bientôt publié dans le journal "Mathematica Applicanda", au sein du Volume 51, Numéro 1* (2023). DOI : [10.48550/arXiv.2210.10829](https://doi.org/10.48550/arXiv.2210.10829) (cf. p. [142](#)).
- [Kac23c] M. KACI ET S. RADJEF. « An adaptive method to solve multilevel multiobjective linear programming problems ». *Ce travail sera bientôt publié dans le journal "Operations Research and Decisions", au sein du Volume 33, Numéro 3* (2023). DOI : [10.48550/arXiv.2210.14302](https://doi.org/10.48550/arXiv.2210.14302) (cf. p. [34](#), [128](#), [138](#)).
- [Kol95] B. KOLMAN ET R.E. BECK. *Elementary Linear Programming with Applications*. Deuxième édition. Elsevier Science and Technology Books, 1995. DOI : [10.1016/B978-0-12-417910-3.X5000-1](https://doi.org/10.1016/B978-0-12-417910-3.X5000-1) (cf. p. [xix](#), [38](#), [40](#), [53](#)).
- [Kol11] T. KOLTAI ET V. TATAY. « A practical approach to sensitivity analysis in linear programming under degeneracy for management decision making ». *International Journal of Production Economics* 131.1 (2011), p. 392-398. DOI : <https://doi.org/10.1016/j.ijpe.2010.04.037> (cf. p. [74](#), [143](#)).



- [Kuc05] D. KUČHTA. « Fuzzy solution of the linear programming problem with interval coefficients in the constraints ». *Operations Research and Decisions* 15.4 (2005), p. 35-42 (cf. p. 34, 143).
- [Lu 16] J. LU ET J. HAN ET Y. HU ET G. ZHANG. « Multilevel decision-making: A survey ». *Information Sciences* 346-347 (2016), p. 463-487. DOI : [10.1016/j.ins.2016.01.084](https://doi.org/10.1016/j.ins.2016.01.084) (cf. p. xix, xxii, 88).
- [Luc87] D.T. LUC. « Connectedness of the efficient point sets in quasiconcave vector maximization ». *Journal of Mathematical Analysis and Applications* 122.2 (1987), p. 346-354. DOI : [10.1016/0022-247X\(87\)90264-2](https://doi.org/10.1016/0022-247X(87)90264-2) (cf. p. 78).
- [Man17] W.A. MANDAL, S. ISLAM. « Multiobjective geometric programming problem under uncertainty ». *Operations Research and Decisions* 27.4 (2017), p. 85-109. DOI : [10.5277/ord170405](https://doi.org/10.5277/ord170405) (cf. p. 143).
- [Moh97] R.H. MOHAMED. « The relationship between goal programming and fuzzy programming ». *Fuzzy Sets and Systems* 89.2 (1997), p. 215-222. DOI : [10.1016/S0165-0114\(96\)00100-5](https://doi.org/10.1016/S0165-0114(96)00100-5) (cf. p. 78).
- [Nay20] S. NAYAK. *Fundamentals of Optimization Techniques with Algorithms*. Première édition. 2020 (cf. p. 18).
- [Nem91] A. NEMIROVSKII ET K. SCHEINBERG. « Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems ». *Mathematical Programming* 72 (1991), 273—289. DOI : [10.1007/BF02592093](https://doi.org/10.1007/BF02592093) (cf. p. 18).
- [Noc99] J. NOCEDAL ET S.J. WRIGHT. *Numerical Optimization*. Springer, 1999. DOI : [10.1007/978-0-387-40065-5](https://doi.org/10.1007/978-0-387-40065-5) (cf. p. 78).
- [Nyi21] P.B. NYIAM ET A. SALHI. « A Comparison of Benson's Outer Approximation Algorithm with an Extended Version of Multiobjective Simplex Algorithm ». *Advances in Operations Research* 2021 (2021), p. 1-11. DOI : [10.1155/2021/1857030](https://doi.org/10.1155/2021/1857030) (cf. p. 87).
- [Phi72] J. PHILIP. « Algorithms for the vector maximization problem ». *Mathematical Programming* 2 (1972), p. 207-229. DOI : [10.1007/BF01584543](https://doi.org/10.1007/BF01584543) (cf. p. 79).
- [Pie11] C.O. PIEUME ET P. MARCOTTE ET L.P. FOTSO ET P. SIARRY. « Solving Bilevel Linear Multiobjective Programming Problems ». *American Journal of Operations Research* 1.4 (2011), p. 214-219. DOI : [10.4236/ajor.2011.14024](https://doi.org/10.4236/ajor.2011.14024) (cf. p. xix, 88).
- [Rad11a] S. RADJEF. *Application de la méthode adaptée aux problèmes multicritères*. Thèse de Doctorat. Université de Béjaïa, 2011 (cf. p. xix, xxi, 34).
- [Rad20] S. RADJEF. *Optimisation Cours Et Exercices*. Thèses algerie, 2020 (cf. p. 78).
- [Rad98] S. RADJEF ET M. HAMGA. *Etude comparative entre deux méthodes de résolution d'un programme quadratique convexe*. Mémoire d'ingénieur. Université de Bejaïa, 1998 (cf. p. xxi, 18).
- [Rad07a] S. RADJEF ET M.O. BIBI. *A new algorithm for linear programming problems*. Proceedings of ORP 3 Meeting, Guimaraes, Portugal, 2007 (cf. p. 34).
- [Rad07b] S. RADJEF ET M.O. BIBI. *Solution of a convex quadratic program via the direct support method*. Proceedings COSI 2007. Université USTO, Oran, 2007 (cf. p. xxi, 87).
- [Rad07c] S. RADJEF ET M.O. BIBI. *Solution of a convex quadratic program with mixed variables by the direct support method*. Proceedings of the international conference MOAD'2007, 2007 (cf. p. xxi, 87).
- [Rad11b] S. RADJEF ET M.O. BIBI. « An Effective Generalization of the Direct Support Method ». *Mathematical Problems in Engineering* (2011), p. 1-18. DOI : [10.1155/2011/374390](https://doi.org/10.1155/2011/374390) (cf. p. xix, 57, 87, 128).
- [Rad12] S. RADJEF ET M.O. BIBI. « An effective generalization of the direct support method in quadratic convex programming ». *Applied Mathematical Sciences* 6.29-32 (2012), p. 1563-1569 (cf. p. xxi, 87).
- [Rad14] S. RADJEF ET M.O. BIBI. « A new algorithm for linear multiobjective programming problems with bounded variables ». *Arabian Journal of Mathematics* 3.1 (2014), p. 79-92. DOI : [10.1007/s40065-013-0094-x](https://doi.org/10.1007/s40065-013-0094-x) (cf. p. xix, xxi, 34).
- [Rad04] S. RADJEF ET M.O. BIBI ET M.S. RADJEF. *Résolution d'un programme linéaire multiobjectifs à variables bornées par la méthode directe du support*. Proceedings COSI 2007. Université de Tizi Ouzou, 2004 (cf. p. xxi, 34).



- [Roc70] R.T. ROCKAFELLAR. *Convex Analysis*. Princeton University Press, 1970 (cf. p. 18).
- [Sad15] F. SADI. *Algorithmes exacts et approchés pour les problèmes d'ordonnement multi-agent à machines parallèles*. Thèse de Doctorat. Université François Rabelais de Tours, 2015 (cf. p. 79).
- [Sei79] L. SEIFORD ET P.L. YU. « Potential solutions of linear systems: The multi-criteria multiple constraint levels program ». *Journal of Mathematical Analysis and Applications* 69.2 (1979), p. 283-303. DOI : [10.1016/0022-247X\(79\)90143-4](https://doi.org/10.1016/0022-247X(79)90143-4) (cf. p. 88).
- [Sha16] A. SHAHIN ET P. HANAFIZADEH ET M. HLADÍK. « Sensitivity analysis of linear programming in the presence of correlation among right-hand side parameters or objective function coefficients ». *Central European Journal of Operations Research* 24 (2016), p. 563-593. DOI : [10.1007/s10100-014-0353-8](https://doi.org/10.1007/s10100-014-0353-8) (cf. p. xx, 63, 143).
- [Shi96] H.-S. SHIH ET Y.-J. LAI ET E.S. LEE. « Fuzzy approach for multi-level programming problems ». *Computers et Operations Research* 23.1 (1996), p. 73-91. DOI : [10.1016/0305-0548\(95\)00007-9](https://doi.org/10.1016/0305-0548(95)00007-9) (cf. p. xix, 88).
- [Sin04] S.B. SINHA ET S. SINHA. « A Linear Programming Approach for Linear Multi-Level Programming Problems ». *The Journal of the Operational Research Society* 55.3 (2004), p. 312-316. DOI : [10.1057/palgrave.jors.2601701](https://doi.org/10.1057/palgrave.jors.2601701) (cf. p. xix, 91, 93, 110).
- [Siv19] M. SIVRI ET H. G. KOCKEN ET I. ALBAYRAK ET S. AKIN. « Generating a set of compromise solutions of a multi objective linear programming problem through game theory ». *Operations Research and Decisions* 29.2 (2019), p. 77-88. DOI : [10.37190/ord190205](https://doi.org/10.37190/ord190205) (cf. p. 143).
- [Ste79] R.E. STEUER. « Goal programming sensitivity analysis using interval penalty weights ». *Mathematical Programming* 17 (1979), p. 16-31. DOI : [10.1007/BF01588223](https://doi.org/10.1007/BF01588223) (cf. p. 63).
- [Sto70] J. STOER ET C. WITZGALL. *Convexity and Optimization in Finite Dimensions I*. Die Grundlehren der mathematischen Wissenschaften. Springer, Berlin, Heidelberg, 1970. DOI : [10.1007/978-3-642-46216-0](https://doi.org/10.1007/978-3-642-46216-0) (cf. p. 25, 143).
- [Sun06] W. SUN ET Y.-X. YUAN. *Optimization Theory and methods. Nonlinear programming*. Springer, 2006 (cf. p. 18).
- [Sya17] SYARIPUDDIN ET H. SUPRAJITNO ET FATMAWATI. « Extension of Wolfe Method for Solving Quadratic Programming with Interval Coefficients ». *Journal of Applied Mathematics* 2017 (2017), p. 1-6. DOI : [10.1155/2017/9037857](https://doi.org/10.1155/2017/9037857) (cf. p. 57).
- [Vin89] P. VINCKE. *L'aide multicritères à la décision*. Université de Bruxelles. Ellipses, 1989 (cf. p. 78).
- [War90] J.E. WARD ET R.E. WENDELL. « Approaches to sensitivity analysis in linear programming ». *Annals of Operations Research* 27 (1990), p. 3-38. DOI : [10.1007/BF02055188](https://doi.org/10.1007/BF02055188) (cf. p. 62, 143).
- [Wen92] U.-P. WEN ET S.-T. HSU. « Efficient solutions for the linear bilevel programming problem ». *European Journal of Operational Research* 62.3 (1992), p. 354-362. DOI : [10.1016/0377-2217\(92\)90124-R](https://doi.org/10.1016/0377-2217(92)90124-R) (cf. p. 88).
- [Wen77] R.E. WENDELL ET D.N. LEE. « Efficiency in multiple objective optimization problems ». *Mathematical Programming* 12 (1977), p. 406-414. DOI : [10.1007/BF01593807](https://doi.org/10.1007/BF01593807) (cf. p. 86).
- [Wol59] P. WOLFE. « The Simplex Method for Quadratic Programming ». *Econometrica* 27.3 (1959), p. 382-398 (cf. p. xix, 47).
- [Ye 89] Y. YE ET E. TSE. « An extension of Karmarkar's projective algorithm for convex quadratic programming ». *Mathematical Programming* 44 (1989), p. 157-179. DOI : [10.1007/BF01587086](https://doi.org/10.1007/BF01587086) (cf. p. 18).
- [Yib19] L.V. YIBING ET Z. WAN. « Linear bilevel multiobjective optimization problem: Penalty approach ». *Journal of Industrial and Management Optimization* 15.3 (2019), p. 1213-1223. DOI : [10.3934/jimo.2018092](https://doi.org/10.3934/jimo.2018092) (cf. p. xix, 88).
- [Yil00] E.A. YILDIRIM ET S.J. WRIGHT. « Warm-start strategies in interior-point methods for linear programming. Technical report 1258 ». *School of Operations research and industrial engineering, Cornell University* (2000) (cf. p. 34).



- [Yu74] P.L. YU. « Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives ». *Journal of Optimization Theory and Applications* 14 (1974), p. 319-377. DOI : [10.1007/BF00932614](https://doi.org/10.1007/BF00932614) (cf. p. [xix](#), [137](#)).
- [Yu 74] P.L. YU ET M. ZELENY. « The techniques of linear multiobjective programming ». *Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Série Verte* 8.3 (1974), p. 51-71 (cf. p. [xix](#), [112](#), [125](#), [137](#), [140](#), [141](#)).
- [Yu 75] P.L. YU ET M. ZELENY. « The set of all nondominated solutions in linear cases and a multicriteria simplex method ». *Journal of Mathematical Analysis and Applications* 49 (1975), p. 430-468. DOI : [10.1016/0022-247X\(75\)90189-4](https://doi.org/10.1016/0022-247X(75)90189-4) (cf. p. [xix](#), [112](#), [116](#), [117](#), [119](#), [135](#)).
- [Zel74] M. ZELENY. *Linear Multiobjective Programming*. Springer, Berlin, Heidelberg, 1974, p. 223. DOI : [10.1007/978-3-642-80808-1](https://doi.org/10.1007/978-3-642-80808-1) (cf. p. [79](#)).
- [Zim76] H.-J. ZIMMERMANN. « Description and optimization fuzzy systems ». *International Journal of General Systems* 2.4 (1976), p. 209-215. DOI : [10.1080/03081077608547470](https://doi.org/10.1080/03081077608547470) (cf. p. [78](#), [143](#)).
- [Zim78] H.-J. ZIMMERMANN. « Fuzzy programming and linear programming with several objective functions ». *Fuzzy Sets and Systems* 1.1 (1978), p. 45-55. DOI : [10.1016/0165-0114\(78\)90031-3](https://doi.org/10.1016/0165-0114(78)90031-3) (cf. p. [78](#)).

تركز هذه الأطروحة بشكل رئيسي على تطوير أساليب لحل مشاكل البرمجة الخطية متعددة المستويات، استنادًا إلى مبدأ الطريقة المتكيفة. يظل حل هذه المشاكل بفاعلية تحديًا معقدًا. تواجه هذه الصعوبة أيضًا في معظم المشاكل التي تمت مناقشتها في هذا العمل. هدف هذه الأطروحة هو تعميم الطريقة المتكيفة لحل مشاكل البرمجة الخطية متعددة المعايير والمستويات. في الفصل 3، يتم تقديم دراسة هندسية لاستقرار الحلول لمشكلة البرمجة الخطية ذات المعيار الوحيد. تخصص الفصول 5 و 6 لبناء إجراءات استنادًا إلى مبدأ الطريقة المتكيفة لحل مشاكل البرمجة الخطية متعددة المستويات. في الفصل 5، يتم إنشاء تطبيق تقني متقدم لتحديد تقدير جديد للأداء الفرعي. يتم استخدام هذه البناءات لتطوير نسختين من خوارزمية لحل مشاكل البرمجة الخطية متعددة المستويات بهدف واحد. وأخيرًا، في الفصل 6، نعلم مفهوم الحلول غير المهيمنة في سياق المشاكل الخطية متعددة المستويات ونضع طريقة لتوليدها. يتم استخدام هذا المفهوم الجديد لوصف خوارزمية تطبق الطريقة المتكيفة لتحقيق توازن واحد. تؤكد النتائج المحصلة على فعالية وجدوى الأساليب المطورة في حل مشاكل البرمجة الخطية متعددة المستويات. تعتبر هذه الأطروحة محاولة مهمة لتوسيع فهمنا لحلول مشاكل البرمجة الخطية متعددة المستويات وتطبيق الطريقة المتكيفة لتحقيق توازن مثلي. حيث تساهم في تطوير النهجيات والتقنيات للتغلب على التحديات الصعبة في هذا المجال وتوسيع مجال البحث في هذا المجال، وتفتح آفاقًا جديدة للابتكار والتطوير المستقبلي.

الكلمات المفتاحية

البرمجة الخطية، البرمجة المتعددة الأهداف، البرمجة متعددة المستويات، تحليل الحساسية، الطريقة المتكيفة، تقدير العدم الأمثلية الجزئية، الحلول غير المهيمنة.

RÉSUMÉ

La présente thèse se concentre principalement sur le développement de méthodes de résolution de problèmes de programmation linéaire à plusieurs niveaux, en se basant sur le principe de la méthode adaptée. La résolution efficace de problèmes multi-niveaux reste un défi complexe. Cette difficulté se retrouve également dans la plupart des problèmes abordés dans ce travail.

L'objectif de cette thèse est de généraliser la méthode adaptée pour résoudre les problèmes de programmation linéaire multi-critère à plusieurs niveaux. Dans le chapitre 3, une étude géométrique de la stabilité des solutions d'un problème de programmation linéaire mono-critère est présentée. Les chapitres 5 et 6 sont consacrés à la construction de procédures basées sur le principe de la méthode adaptée pour résoudre les problèmes de programmation linéaire à plusieurs niveaux. Dans le chapitre 5, une construction technique avancée d'une application est établie pour définir une nouvelle estimation de sous-optimalité. Ces constructions sont ensuite utilisées pour développer deux versions d'un algorithme résolvant les problèmes de programmation linéaire multi-niveaux à objectif unique. Enfin, dans le chapitre 6, nous généralisons le concept de solutions non-dominées dans le contexte des problèmes linéaires multi-niveaux, et nous mettons en place une méthode pour les générer. Ce nouveau concept est utilisé pour décrire un algorithme qui applique la méthode adaptée afin d'atteindre un compromis unique.

Les résultats obtenus démontrent l'efficacité et la pertinence des méthodes développées pour résoudre les problèmes de programmation linéaire à plusieurs niveaux. Cette thèse apporte ainsi des contributions significatives à la résolution de problèmes complexes et ouvre de nouvelles perspectives pour la recherche future dans ce domaine.

MOTS CLÉS

Programmation linéaire, Programmation multi-objectifs, Programmation multi-niveaux, Analyse de sensibilité, Méthode adaptée, Estimation de sous-optimalité, Solutions non-dominées.

ABSTRACT

This thesis primarily focuses on the development of methods for solving multi-level linear programming problems based on the principle of the adaptive method. Efficiently solving multi-level problems remains a complex challenge, which is also evident in the majority of the problems addressed in this work.

The goal of this thesis is to generalize the adaptive method for solving multi-level multi-criteria linear programming problems. Chapter 3 presents a geometric study of solution stability in single-objective linear programming problems. Chapters 5 and 6 are dedicated to constructing procedures based on the principle of the adaptive method to solve multi-level linear programming problems. In Chapter 5, an advanced technical construction of an application is established to define a new sub-optimality estimate. These constructions are then utilized to develop two versions of an algorithm that solves single-objective multi-level linear programming problems. Lastly, in Chapter 6, we generalize the concept of non-dominated solutions in the context of multi-level linear problems and introduce a method to generate them. This new concept is employed to describe an algorithm that applies the adaptive method to achieve a unique compromise.

The obtained results demonstrate the effectiveness and relevance of the developed methods in solving multi-level linear programming problems. This thesis makes significant contributions to the resolution of complex problems and opens up new prospects for future research in this field.

KEYWORDS

Linear programming, Multi-objective programming, Multi-level programming, Sensitivity analysis, Adaptive method, Sub-optimality estimate, Non-dominated solutions.

