

UNIVERSITÉ DE NANTES
UFR DES SCIENCES ET TECHNIQUES

SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DE MATHÉMATIQUES

Année 2013

N° attribué par la bibliothèque

| | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|

Caractérisation des solutions efficaces et
algorithmes d'énumération exacts pour
l'optimisation multiobjectif en variables mixtes
binaires

THÈSE DE DOCTORAT
Discipline : INFORMATIQUE
Spécialité : RECHERCHE OPÉRATIONNELLE

*Présentée
et soutenue publiquement par*

Thomas VINCENT

le 30 octobre 2013 au LINA, devant le jury ci-dessous

| | | | |
|-------------|---|-------------------------------------|----------------------------|
| Président | : | Pr. Marc SEVAUX, Professeur | Université de Bretagne Sud |
| Rapporteurs | : | Pr. Daniel VANDERPOOTEN, Professeur | Université Paris-Dauphine |
| | | Pr. Jacques TEGHEM, Professeur | Université de Mons |
| | | Pr. Matthias EHRGOTT, Professeur | Lancaster University |
| Examineurs | : | Pr. Stefan RUZIKA, Professeur | Universität Koblenz-Landau |

Directeur de thèse : Pr. Xavier GANDIBLEUX

Co-encadrant de thèse : Dr. Anthony PRZYBYLSKI

Laboratoire : LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE.
2, rue de la Houssinière, BP 92 208 – 44 322 Nantes, CEDEX 3.

**CARACTÉRISATION DES SOLUTIONS EFFICACES ET
ALGORITHMES D'ÉNUMÉRATION EXACTS POUR
L'OPTIMISATION MULTIOBJECTIF EN VARIABLES MIXTES
BINAIRES**

*Characterization of efficient solutions and enumeration algorithms
for multiple objectives optimization with mixed binary variables*

Thomas VINCENT



favet neptunus eunti

Université de Nantes

Thomas VINCENT

*Caractérisation des solutions efficaces et algorithmes d'énumération exacts pour
l'optimisation multiobjectif en variables mixtes binaires*

x+129 p.

Résumé

Dans ce travail, nous nous intéressons à la résolution exacte de problèmes d'optimisation multiobjectif en variables mixtes binaires. La nature mixte des variables introduit d'importantes différences par rapport aux contextes purement discrets ou continus. Nous proposons donc de prendre en compte ces différences grâce à une représentation appropriée des ensembles de solutions ainsi qu'une procédure de mise à jour dédiée.

Ces propositions nous permettent, dans le contexte biobjectif, d'adapter deux méthodes de résolution usuellement appliquées aux problèmes combinatoires : la procédure de Branch & Bound et la méthode en deux phases. Nous proposons de nombreux affinements pour ces méthodes, comme de nouveaux ensembles bornant ou des stratégies de cheminement. À partir de nos observations sur leurs performances, nous proposons une nouvelle routine pour la seconde phase de la méthode en deux phases, reprenant les points forts des méthodes étudiées.

Dans le contexte triobjectif, nous étendons notre représentation des ensembles de solutions en procédant par analogie avec le cas biobjectif. Les méthodes de résolution sont également adaptées à ce contexte et étudiées. En particulier, la décomposition de la zone de recherche lors de la seconde phase est décrite en détail.

La solution logicielle proposée a été appliquée sur un problème réel : l'évaluation d'une politique de choix de véhicules. Les choix concernés vont de véhicules conventionnels aux véhicules électriques, eux-mêmes alimentés par une source d'électricité classique ou par panneaux solaires.

Mots-clés : programmation linéaire multiobjectif en variables mixtes, branch & bound, méthode en deux phases, résolution exacte.

Abstract

The purpose of this work is the exact solution of multiple objective binary mixed integer linear programmes. The mixed nature of the variables implies significant differences with purely continuous or purely discrete programmes. Thus, we propose to take these differences into account using a proper representation of the solution sets and a dedicated update procedure.

These propositions allow us to adapt for the biobjective case two solution methods commonly used for combinatorial problems: the Branch & Bound algorithm and the two phase method. Several improvements are proposed, such as bound sets or visiting strategies. We introduce a new routine for the second phase of the two phase method that takes advantage of all the relevant features of the previously studied methods.

In the 3-objective context, the solution sets representation is extended by analogy with the biobjective case. Solutions methods are extended and studied as well. In particular, the decomposition of the search area during the second phase is thoroughly described.

The proposed software solution has been applied on a real world problem: evaluation of a vehicle choice policy. The possible choices range from classical to electric vehicles that are powered by grid or solar power.

Keywords: multiple objective mixed integer linear programmes, branch & bound, two-phases method, exact solution.

Remerciements

Ce manuscrit de thèse serait bien différent (voire, ne serait pas) sans l'apport et la contribution de nombreuses personnes. Faire une liste complète triplerait certainement la taille de ce manuscrit et je vais, malgré l'impossibilité de la tâche, tenter d'en faire un bilan succinct qui rende justice à tout le monde.

Bien évidemment, je tiens tout d'abord à remercier chaleureusement Xavier Gandibleux et Anthony Przybylski, mes encadrants, qui m'ont transmis leur passion de la RO en général et du multiobjectif en particulier, qui m'ont fait confiance en me proposant ce sujet de thèse, puis m'ont progressivement laissé prendre la barre de mon propre navire en ces mers inconnues. Je remercie également Marc Sevaux, Daniel Vanderpooten, Jacques Teghem, Matthias Ehrgott et Stefan Ruzika qui m'ont fait l'immense honneur de constituer mon jury et dont les rapports et remarques alimenteront et inspireront mes recherches futures.

Mes remerciements vont également aux nombreuses personnes qui ont contribué à la qualité scientifique de cette thèse. Il y a tout d'abord Stefan Ruzika et Florian Seipp, avec qui j'ai eu la chance de travailler au début de ma thèse et qui m'ont aidé à formaliser mes idées. Stefan, encore lui, et Matthias Ehrgott, ont suivi tout mon travail dans le cadre du comité de suivi de thèse et m'ont aidé à garder un cap cohérent. Leurs précieux conseils et avis se ressentent dans toutes les parties de cette thèse. Le chapitre 5 de cette thèse n'existerait pas sans Ellen de Schepper qui, en plus d'être une personne adorable et d'avoir très bon goût en matière de parapluies, m'a servi sur un plateau une application passionnante permettant à mon travail de prendre une toute autre dimension. Je tiens également à remercier tout le groupe de l'ANR Guépard, avec qui j'ai eu de nombreux échanges incroyablement enrichissants, tant sur le plan scientifique qu'humain.

Évidemment, je remercie aussi mes nombreux collègues de bureaux qui ont, chacun à leur façon, égayé mon quotidien : Benoît, qui m'a également appris à écrire *multiobjectif* correctement ; Gilles, pour ses spams mémorables ; Marie, pour sa bonne humeur, ses oursons à la guimauve et son expertise en latex et tikz ; Aurélien, pour son humour de bon goût ; Laurianne, pour son accent qui chante et sa préoccupation pour mes rides ; et enfin Benjamin et Audrey, même s'ils ont mauvais goût en matière de thé, parce qu'ils m'ont énormément soutenu pendant les derniers mois. Merci également à Fabien pour nos soirées BSG mais aussi pour son soutien et ses conseils. Plus généralement, je remercie tous les (ex) doctorants, notamment Ophélie, Liza, Matthieu, Adrien, Julien, Bruno, Laurent, Nagham, Khaled, Mohamed, Rima, Hafedh. . .

Par ailleurs, s'il est bien une catégorie de personnes qui sont trop peu souvent remerciées, c'est bien les personnels administratifs et techniques, sans qui le quotidien des chercheurs serait nettement moins enviable. Pour contribuer à la réparation de cette injustice, je remercie donc Anne-Françoise, les Annie, Isabelle, Floriane, Élodie, Jean-Paul, Christine, Marie, Sabine, Dimitri, Wallid et Jean-Yves. Grâce à vous, ma thèse s'est déroulée dans les meilleures conditions possibles.

Merci à mes proches et à ma famille, qui m'ont soutenu au cours de cette longue traversée de quatre ans.

Enfin, je souhaite remercier Aurélia qui m'a accompagné, soutenu, motivé pendant toute cette thèse. Tu as été mon inspiration, mon modèle, ma boussole.
Merci.

Sommaire

| | |
|---|------------|
| Introduction | 1 |
| Notations | 5 |
| 1 Optimisation multiobjectif | 7 |
| 1.1 Optimisation mono-objectif | 9 |
| 1.2 Optimisation multiobjectif | 11 |
| 1.3 Méthodes de résolution des programmes linéaires multiobjectifs | 15 |
| 1.4 Méthodes de résolution programmes linéaires multiobjectifs en variables entières et combinatoires | 18 |
| 1.5 Discussion | 37 |
| 2 Algorithme de Branch & Bound pour les problèmes biobjectifs | 39 |
| 2.1 Défauts de l'algorithme de Mavrotas et Diakoulaki | 41 |
| 2.2 Caractérisation correcte des solutions efficaces | 42 |
| 2.3 Méthode de mise à jour des ensembles non dominés | 45 |
| 2.4 Améliorations de l'algorithme de Branch & Bound | 53 |
| 2.5 Résultats expérimentaux | 57 |
| 2.6 Conclusion | 63 |
| 3 Méthodes en deux phases pour les problèmes biobjectifs | 65 |
| 3.1 Méthode en deux phases avec Branch & Bound | 67 |
| 3.2 Méthode en deux phases avec ranking | 68 |
| 3.3 Proposition d'une nouvelle méthode pour la phase 2 | 73 |
| 3.4 Résultats expérimentaux | 76 |
| 3.5 Conclusion | 79 |
| 4 Résolution de problèmes à trois objectifs | 81 |
| 4.1 Difficultés attendues | 83 |
| 4.2 Caractérisation des points dominés et non dominés | 86 |
| 4.3 Méthode de mise à jour des ensembles non dominés | 87 |
| 4.4 Adaptation des composants du Branch & Bound | 88 |
| 4.5 Méthode en deux phases | 89 |
| 4.6 Résultats expérimentaux | 99 |
| 4.7 Conclusion | 101 |
| 5 Solutions informatiques et application | 103 |
| 5.1 Implémentation logicielle | 105 |
| 5.2 Application à un problème réel | 106 |
| 5.3 Conclusion | 114 |
| Conclusions et perspectives | 115 |
| Bibliographie | 119 |
| Liste des tableaux | 123 |

| | |
|------------------------------------|------------|
| Liste des figures | 125 |
| Liste des algorithmes | 126 |

Introduction

La programmation linéaire en variables mixtes permet de modéliser de nombreux problèmes rencontrés dans des situations réelles. C'est donc sans surprise que l'on constate qu'un effort important a été consenti sur la résolution de ce type de problèmes. Parallèlement, force est de constater que le paradigme multiobjectif s'invite naturellement dans ce cadre. Les problèmes de localisation de services où il convient d'optimiser simultanément des objectifs économiques et environnementaux en sont une illustration parmi tant d'autres. L'étude de la conjonction de ces deux domaines de l'optimisation est donc pertinente. Cependant, bien que l'optimisation multiobjectif ait bénéficié d'un grand intérêt au cours des vingt dernières années, la communauté se concentre essentiellement sur des programmes linéaires en variables exclusivement continues (depuis les années 1970) ou discrètes (surtout depuis le milieu des années 1990). En effet, les difficultés rencontrées sur ces classes de problèmes ont été suffisantes pour reporter jusqu'à récemment l'étude spécifique des problèmes en variables mixtes. Si des problèmes de programmation linéaire multiobjectif en variables mixtes ont bien été considérés implicitement dans la littérature, leur résolution fut quasiment exclusivement abordée à l'aide d'algorithmes d'approximation sans garantie de qualité (heuristiques ou métaheuristiques) et en ignorant le caractère continu d'une partie des variables de ces problèmes. Le manque de techniques adaptées résultant de l'étude approfondie de ces problèmes spécifiques est évidemment en cause.

Nos travaux s'intéressent donc à la résolution exacte des programmes linéaires multiobjectifs en variables mixtes. En particulier, nous nous intéressons aux différences, par rapport au domaine combinatoire, induites par l'introduction de variables continues, comme la représentation de l'ensemble des solutions. Nous abordons successivement les cadres biobjectif et triobjectif.

Les seules contributions notables dans notre domaine d'intérêt sont, à notre connaissance, les articles de [[Mavrotas et Diakoulaki, 1998](#), [Mavrotas et Diakoulaki, 2005](#)] qui ont proposé un algorithme de Branch & Bound pour les résoudre. Il s'agit de l'adaptation au contexte mixte d'un algorithme couramment utilisé en optimisation combinatoire multiobjectif. Le principe de cet algorithme repose sur l'énumération implicite de toutes les combinaisons possibles des variables binaires et ne traite les variables continues que lors de l'évaluation des nœuds de l'arbre. De nombreuses énumérations sont évitées grâce à l'utilisation de bornes qui permettent de déterminer des ensembles de combinaisons inintéressantes. Hélas, l'algorithme proposé comporte une erreur de conception au niveau de la représentation des solutions pouvant conduire à des résultats erronés ou incohérents, comme des solutions qui sont en fait dominées et l'omission de certaines solutions efficaces.

Nos travaux se sont donc attachés à la correction de cette erreur dans le contexte biobjectif au moyen d'une représentation appropriée de l'ensemble des solutions efficaces. Cette nouvelle représentation tient compte de la nature continue de certaines variables, ce qui engendre des ensembles convexes de solutions. En particulier, il est donc nécessaire de représenter des arêtes constituées de points non dominés ainsi que de définir une méthode de mise à jour adaptée pour ces nouveaux objets. En plus de rendre l'algorithme de Branch & Bound correct et complet, la représentation proposée constitue également une amélioration de l'ensemble bornant supérieurement utilisé dans l'algorithme. Nous proposons par ailleurs différentes améliorations significatives sur d'autres composants de l'algorithme.

Une autre méthodologie usitée pour la résolution exacte de problèmes d'optimisation combinatoire multiobjectif gagne en popularité, du fait de ses performances remarquables rapportées dans la littérature scientifique : la méthode en deux phases. Comme l'indique son nom, cette méthode décompose en deux phases le calcul de l'ensemble des solutions efficaces. La première phase est dédiée au calcul des solutions dites supportées tandis que la seconde s'attache à la détermination des solutions efficaces restantes, dites non supportées. La réussite de cette méthode tient dans le fait que les solutions issues de la première phase sont algorithmiquement généralement plus simples à obtenir et permettent de restreindre l'espace de recherche dans lequel va opérer la seconde phase.

L'extension aux problèmes triobjectifs, bien que récemment proposée par [Przybylski *et al.*, 2010] dans le cas des problèmes d'optimisation combinatoire, nécessite d'être revue dans le cadre des problèmes en variables mixtes. En effet, des difficultés se manifestent immédiatement du fait des variables continues. En particulier, l'ensemble non dominé est dorénavant décrit en utilisant des objets ayant une dimension allant jusqu'à 2. Dès lors, de nouvelles problématiques apparaissent quant à la comparaison et la mise à jour de ces faces. Par ailleurs, même si la plupart des changements induits par le passage à trois objectifs sont maintenant bien connus, quelques autres difficultés persistent, comme la représentation de la zone de recherche utilisée par la méthode en deux phases.

Nos principales contributions à la programmation linéaire multiobjectif en variables mixtes binaires sont rapportées dans le présent manuscrit, lequel est structuré en cinq chapitres. Le premier chapitre introduit les propriétés principales des programmes linéaires multiobjectifs en variables continues et en variables entières. Pour chaque catégorie de problèmes, les méthodes de résolution sur lesquelles nous nous basons sont présentées.

Dans le second chapitre, des erreurs sont mises en évidence dans l'algorithme de Branch & Bound de [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005]. Ces erreurs sont corrigées dans le contexte biobjectif pour donner une méthode de mise à jour de l'ensemble non dominé basée sur une représentation appropriée des éléments non dominés. Par la suite, des améliorations sont proposées pour chaque composant de l'algorithme. Ces propositions sont évaluées et comparées.

Le troisième chapitre s'inscrit dans la continuité du précédent en s'intéressant cette fois à la méthode en deux phases pour les problèmes biobjectifs. Deux méthodes sont évaluées en tant que secondes phases. Tout d'abord, l'algorithme de Branch & Bound du chapitre 2 est réutilisé localement pour visiter les zones de recherches. Des modifications y sont proposées pour permettre de restreindre la recherche à une zone donnée, mais aussi pour tirer parti de cette restriction, en adaptant certaines idées utilisées pour la version globale de l'algorithme. La seconde méthode étudiée est une adaptation de celle du ranking. Le contexte mixte exige en effet un certain nombre de modifications par rapport à la version classique (discrète) de l'algorithme. Les expérimentations qui suivent permettent de tirer des conclusions sur les forces et faiblesses de chacune des deux méthodes. Ces observations mènent à l'élaboration d'une nouvelle méthode égalant ou surpassant les deux précédentes.

Dans le quatrième chapitre se pose la question d'une troisième fonction objectif. Les difficultés attendues sont décrites et distinguées en deux catégories : les difficultés communes à tous les problèmes triobjectifs et celles qui sont propres au contexte mixte. À partir de ces observations, la représentation biobjectif de l'ensemble non dominé est étendue, de même que la procédure de mise à jour associée. L'extension des méthodes de résolution est ensuite étudiée. Si celle du Branch & Bound global est immédiate, il en va autrement pour la méthode en deux phases. En

effet, bien que première et deuxième phases ne posent pas de difficultés particulières à proprement parler, la description de la zone de recherche prenant place entre les deux phases est nettement plus ardue. Le calcul de cette zone de recherche fait donc l'objet d'une description détaillée. Les deux méthodes sont par la suite évaluées et comparées.

Enfin, le cinquième chapitre aborde deux sujets transversaux à ceux des précédents. La première partie est consacrée au développement logiciel mis en œuvre pour tester, valider et comparer tous les algorithmes proposés par cette thèse. En particulier, le cas de deux méthodes de résolutions de programmes linéaires biobjectif et triobjectif est détaillé. La deuxième partie est dédiée à l'application de nos travaux dans le cadre d'un problème biobjectif étudié dans le secteur de l'économie de l'environnement. L'étude de ce problème est menée à l'université de Hasselt en Belgique, la résolution étant conduite dans le cadre d'une coopération de recherche avec ce partenaire. Ce problème concerne le choix d'une politique énergétique et de transports pour une entreprise souhaitant minimiser à la fois ses coûts et son impact environnemental. Une formulation du problème est proposée et l'analyse des solutions obtenues à l'issue de la résolution est décrite et discutée.

Notations

| Notation | Signification |
|-----------------------------------|---|
| n | nombre de variables d'un problème d'optimisation |
| m | nombre de contraintes d'un problème d'optimisation |
| p | nombre de fonctions objectif d'un problème d'optimisation multiobjectif |
| $x = (x_1, \dots, x_n)$ | solution |
| $y = (y_1, \dots, y_p)$ | point |
| $z = (z_1, \dots, z_p)$ | vecteur de fonctions objectifs |
| X | ensemble admissible d'un problème d'optimisation |
| $Y := z(X)$ | ensemble réalisable dans l'espace des objectifs |
| y^I | point idéal |
| y^N | point nadir |
| y^U | point utopique |
| y^{ILP} | point idéal de la relaxation linéaire |
| $y^1 < y^2$ | $y_k^1 < y_k^2$ pour $k = 1, \dots, p$ |
| $y^1 \leq y^2$ | $y_k^1 \leq y_k^2$ pour $k = 1, \dots, p$ |
| $y^1 \leq y^2$ | $y^1 \leq y^2$ et $y^1 \neq y^2$ |
| $\mathbb{R}_{>}^p$ | $\{y \in \mathbb{R}^p : y > 0\}$ |
| \mathbb{R}_{\geq}^p | $\{y \in \mathbb{R}^p : y \geq 0\}$ |
| \mathbb{R}_{\leq}^p | $\{y \in \mathbb{R}^p : y \leq 0\}$ |
| X_E | ensemble complet |
| X_{Em} | ensemble complet minimal |
| X_{EM} | ensemble complet maximal |
| X_{SE} | ensemble complet de solutions supportées |
| X_{SEm} | ensemble complet minimum de solutions supportées |
| X_{SEM} | ensemble complet maximum de solutions supportées |
| X_{SE1} | ensemble complet de solutions supportées extrêmes |
| X_{SE1m} | ensemble complet minimum de solutions supportées extrêmes |
| X_{SE1M} | ensemble complet maximum de solutions supportées extrêmes |
| X_{SE2} | ensemble complet de solutions supportées non extrêmes |
| X_{SE2m} | ensemble complet minimum de solutions supportées non extrêmes |
| X_{SE2M} | ensemble complet maximum de solutions supportées non extrêmes |
| Y_N | ensemble des points non dominés |
| Y_{SN} | ensemble des points supportés |
| Y_{SN1} | ensemble des points supportés extrêmes |
| Y_{SN2} | ensemble des points supportés non extrêmes |
| Y_{NN} | ensembles des points non supportés |
| S_N où $S \subset \mathbb{R}^p$ | $S_N := \{s \in S : (s - \mathbb{R}_{\geq}^p) \cap S = \{s\}\}$ |
| $\text{conv } S$ | enveloppe convexe de S |
| $cl(S)$ | fermeture de S |

CHAPITRE 1

Optimisation multiobjectif

Dans ce premier chapitre, nous présentons l'optimisation multiobjectif en exposant les principales définitions et propriétés. Notre travail se trouvant à l'intersection de l'optimisation continue et de l'optimisation discrète, nous présenterons pour chacune les méthodes de résolutions dédiées dans un contexte a posteriori.

| | | |
|------------|--|-----------|
| 1.1 | Optimisation mono-objectif | 9 |
| 1.1.1 | Programmation mathématique | 9 |
| 1.1.2 | Programmation linéaire | 9 |
| 1.1.3 | Programmation linéaire en variables entières et optimisation combinatoire | 9 |
| 1.1.4 | Programmation linéaire en variables mixtes | 10 |
| 1.2 | Optimisation multiobjectif | 11 |
| 1.2.1 | Définitions | 11 |
| 1.2.2 | Propriétés | 11 |
| 1.3 | Méthodes de résolution des programmes linéaires multiobjectifs | 15 |
| 1.3.1 | Simplexe multiobjectif | 15 |
| 1.3.2 | Méthode d'approximation externe de Benson | 18 |
| 1.4 | Méthodes de résolution programmes linéaires multiobjectifs en variables entières et combinatoires | 18 |
| 1.4.1 | Ensembles bornant | 20 |
| 1.4.2 | Branch and bound | 22 |
| 1.4.3 | Méthode en deux phases (2 et 3 objectifs) | 29 |
| 1.5 | Discussion | 37 |

1.1 Optimisation mono-objectif

1.1.1 Programmation mathématique

L'optimisation mono-objectif vise à produire, quand elle existe, une solution optimale d'un problème au regard d'une fonction objectif tout en respectant un système de contraintes formulées sous la forme d'(in)égalités.

Un problème d'optimisation mono-objectif se formule sous la forme suivante :

$$\begin{array}{ll} \text{opt} & z(x) \\ \text{s.c.} & a_j(x) \diamond_j b_j \quad j = 1, \dots, m \\ & \diamond_j \in \{\leq, =, \geq\} \quad j = 1, \dots, m \\ & x_i \in \mathcal{D}_i \quad i = 1, \dots, n \end{array}$$

La fonction $z : \mathbb{R}^n \rightarrow \mathbb{R}$ est à optimiser, c'est-à-dire qu'on recherche (quand il existe) soit son minimum, soit son maximum. Ces deux cas étant symétriques, nous nous plaçons dans la suite, sans perte de généralité, dans un contexte de minimisation. Les vecteurs $x \in \mathcal{D}$ sont appelés *solutions* et \mathcal{D}_i est le domaine de chaque variable x_i . Enfin, les solutions satisfaisant le système de contraintes sont dites *admissibles* et l'ensemble de ces solutions, appelé *ensemble admissible*, est noté X .

1.1.2 Programmation linéaire

La programmation linéaire (PL ou LP : *Linear Programming*) est un cas particulier de la programmation mathématique dans lequel la fonction objectif et les contraintes du problème sont linéaires et les variables sont non négatives. La fonction à minimiser s'écrit donc sous la forme

d'une somme $z = \sum_{i=1}^n c_i x_i$ dont une forme équivalente est $z = c^T x$ où c^T est la transposée du vecteur de coûts $c = (c_1, \dots, c_n)^T$. Du fait de la linéarité des contraintes, les coefficients des variables se représentent usuellement par une *matrice des contraintes* $A \in \mathbb{R}^{m \times n}$, ce qui permet d'écrire un système de contraintes $Ax \geq b$, où $b \in \mathbb{R}^m$ est le vecteur des membres de droite. Enfin, on suppose que les variables sont positives ou nulles et peuvent être bornées. On obtient donc la formulation suivante :

$$\begin{array}{ll} \min & c^T x \\ \text{s.c.} & Ax \geq b \\ & x_i \in \mathbb{R}^+ \quad i = 1, \dots, n \end{array}$$

L'ensemble admissible X d'un programme linéaire est un polytope, un polyèdre non borné ou l'ensemble vide et est en particulier convexe.

Les programmes linéaires peuvent être résolus au moyen de la *méthode du simplexe* ou des *méthodes de point intérieur* [Chvatal, 1983].

1.1.3 Programmation linéaire en variables entières et optimisation combinatoire

En considérant un programme linéaire pour lequel on ajoute une contrainte d'intégrité qui restreint les valeurs des variables aux seuls nombres entiers, on obtient un problème de programmation linéaire en variables entières (PLVE ou ILP : *Integer Linear Programming*). Un tel problème se modélise donc comme suit :

$$\begin{aligned} \min \quad & c^T x \\ \text{s.c.} \quad & Ax \geq b \\ & x_i \in \mathbb{N} \quad i = 1, \dots, n \end{aligned}$$

À cause de ce changement de nature des variables, l'ensemble admissible X n'est plus convexe et les programmes linéaires en variables entières nécessitent donc d'autres méthodes de résolution que celles des programmes linéaires en variables continues.

Pour ce type de problèmes, les méthodes de résolution les plus couramment utilisées sont la programmation dynamique, les méthodes polyédrales ainsi que les méthodes de séparation et évaluation (*Branch & Bound*) [Wolsey, 1998].

Optimisation combinatoire

Les problèmes d'optimisation combinatoire (CO : *Combinatorial Optimization*) peuvent être considérés comme un sous-ensemble de la programmation linéaire en variables binaires ($x_i \in \{0, 1\}$). Ces problèmes se distinguent par leurs contraintes conférant au problème une structure particulière. Parmi les problèmes classiques d'optimisation combinatoire, on peut citer les problèmes d'affectation, de sac à dos, de plus court chemin, de voyageur de commerce, etc. [Teghem, 2012]

Malgré leur difficulté, ces problèmes à structure particulière sont très étudiés dans la littérature et connaissent des méthodes de résolution qui permettent de résoudre des instances de grande taille pour certains problèmes. En particulier, leur structure a pu être exploitée pour donner lieu à des méthodes dédiées.

1.1.4 Programmation linéaire en variables mixtes

Comme leur nom le laisse entendre, les problèmes de programmation linéaire en variables mixtes (PLVM ou MILP : *Mixed Integer Linear Programming*) sont une généralisation des deux classes de problèmes présentées précédemment. Les MILP possèdent en effet à la fois des variables continues et des variables entières, ce qui donne la formulation suivante :

$$\begin{aligned} \min \quad & c^T x \\ \text{s.c.} \quad & Ax \geq b \\ & x_i \in \mathbb{R}_+ \quad i = 1, \dots, n_1 \\ & x_j \in \mathbb{N} \quad j = n_1 + 1, \dots, n \end{aligned}$$

De même que pour les programmes linéaires en nombres entiers, la présence de variables discrètes rend l'ensemble réalisable non convexe en général. La difficulté de résolution provient donc des variables discrètes. D'un point de vue méthodologique, peu de choses différencient la résolution des ILP de celles des MILP. On peut toutefois noter que les variables continues peuvent être avantageusement exploitées, par exemple dans le calcul de coupes [Gomory, 1960].

1.2 Optimisation multiobjectif

1.2.1 Définitions

Un problème d'optimisation multiobjectif ne considère plus une unique fonction à optimiser mais repose sur p fonctions objectifs, c'est-à-dire un vecteur de fonctions objectif. On a donc le modèle :

$$\begin{array}{ll} \text{“min”} & z(x) = (z_1(x), \dots, z_p(x)) \\ \text{s.c.} & x \in X \end{array}$$

où $x = (x_1, \dots, x_n)$ et pour lequel le sens de «min» reste à définir.

Lesendants des classes de problèmes mono-objectifs déjà présentés sont la programmation linéaire multiobjectif (MOLP : *Multiple Objective Linear Programming*), la programmation linéaire multiobjectif en variables entières (MOILP : *Multiple Objective Integer Linear Programming*), l'optimisation combinatoire multiobjectif (MOCO : *Multiple Objective Combinatorial Optimization*) et la programmation linéaire multiobjectif en variables mixtes (MOMILP : *Multiple Objective Mixed Integer Linear Programming*). Nous les aborderons tous dans la suite de cette section.

Définition 1.2.1 (Espace des décisions et espace des objectifs). On appelle *espace des décisions* \mathbb{R}^n , $X \subset \mathbb{R}^n$ et *espace des objectifs* \mathbb{R}^p , $Y = \{z(x) : x \in X\} \subset \mathbb{R}^p$. Y , l'image de X par le vecteur de fonctions objectif, est appelé *ensemble réalisable*. Les vecteurs de \mathbb{R}^p et Y sont respectivement appelés *points* et *points réalisables*.

1.2.2 Propriétés

Nous présentons ici les propriétés liées au contexte multiobjectif.

Efficacité et non dominance

Les fonctions objectif sont supposées conflictuelles. De ce fait, il n'existe pas de solution admissible optimisant tous les objectifs simultanément. La notion mono-objectif de solution optimale ne s'applique donc plus. Il est alors nécessaire de définir un contexte de résolution afin de donner un sens à «min».

Définition 1.2.2 (Dominance). Soient deux points $y^1, y^2 \in \mathbb{R}^p$. On dit que :

- y^1 domine faiblement y^2 si $y_i^1 \leq y_i^2, \forall i = 1, \dots, p$. On écrit alors $y^1 \leq y^2$.
- y^1 domine strictement y^2 si $y_i^1 < y_i^2, \forall i = 1, \dots, p$. On écrit alors $y^1 < y^2$.
- y^1 domine y^2 si $y^1 \leq y^2$ et $y^1 \neq y^2$. On écrit alors $y^1 \leq y^2$.

Définition 1.2.3 (Solution efficace). Soit $x^* \in X$.

- x^* est dite *efficace* s'il n'existe aucune autre solution admissible $x \in X$ telle que $z(x) \leq z(x^*)$. On dit alors que $y^* = z(x^*)$ est un *point non dominé*.
- x^* est dite *faiblement efficace* s'il n'existe aucune autre solution admissible $x \in X$ telle que $z(x) < z(x^*)$. On dit alors que $y^* = z(x^*)$ est un *point faiblement non dominé*.

Définition 1.2.4 (Ensembles efficace et non dominé). L'ensemble des solutions efficaces est noté X_E et l'ensemble des points non dominés est Y_N .

Y_N peut alternativement être défini par $Y_N := \{y \in Y : (y - \mathbb{R}_{\geq}^p) \cap Y = \{y\}\}$.

Plus généralement, pour $S \subset \mathbb{R}^p$, nous posons $S_N := \{s \in S : (s - \mathbb{R}_{\geq}^p) \cap S = \{s\}\}$, l'ensemble des points de S qui ne sont pas dominés par d'autres points de S .

Remarque 1.2.1 – Dans le cas biobjectif, il existe un ordre *naturel* des solutions efficaces. Il est en effet possible de trier ces solutions par valeur croissante d'un objectif et simultanément par valeur décroissante sur le second objectif. Comme nous le verrons par la suite, cette particularité rend les problèmes biobjectifs bien plus simples à traiter et en fait une classe à part dans le domaine de l'optimisation multiobjectif.

Définition 1.2.5 (Équivalence, ensemble complet [Hansen, 1980]). Deux solutions admissibles x et $x' \in X$ sont dites *équivalentes* si $z(x) = z(x')$.

Un ensemble complet X_E est un ensemble de solutions efficaces tel que toute solution $x \in X \setminus X_E$ est soit équivalente, soit dominée par une solution $x' \in X_E$.

Un ensemble complet minimal X_{E_m} est un ensemble complet sans solution équivalente.

L'ensemble complet maximal X_{E_M} est l'ensemble complet contenant toutes les solutions efficaces.

Le sens à donner à «min» dépend de l'ensemble calculé :

- l'ensemble non dominé complet ;
- l'ensemble des solutions lexicographiquement optimales ;
- l'ensemble de compromis, qui fait appel aux préférences du décideur.

Dans le cas des préférences, on parle usuellement de trois contextes :

- le contexte *a priori*, dans lequel on connaît à l'avance les préférences des décideurs ;
- le contexte *a posteriori*, dans lequel aucune supposition n'est faite sur les préférences du décideur, ce qui impose de calculer la totalité de l'ensemble non dominé ;
- le contexte *interactif*, dans lequel le décideur peut affiner ses préférences au cours de la résolution afin d'aboutir à une solution efficace le satisfaisant.

Dans nos travaux, nous cherchons à calculer un ensemble non dominé complet, ce qui s'apparente au contexte *a posteriori*.

Classification des solutions efficaces et des points non dominés

Une méthode bien connue pour calculer des solutions efficaces consiste à résoudre des problèmes dont la fonction objectif est une somme pondérée des p objectifs initiaux.

Théorème 1.2.1 (Somme pondérée [Geoffrion, 1968]). Soient P un MOP ayant p fonctions objectifs et $\lambda \in \mathbb{R}_{\geq}^p$. Le problème pondéré P_λ est défini par :

$$\begin{aligned} \min \quad & z^\lambda = \sum_{k=1}^p \lambda_k z_k(x) \\ \text{s.c.} \quad & x \in X \end{aligned} \tag{P_\lambda}$$

Supposons que x^* soit optimale pour P_λ . On a alors :

- (i) si $\lambda \in \mathbb{R}_{>}^p$, alors x^* est faiblement efficace pour P ;
- (ii) si $\lambda \in \mathbb{R}_{>}^p$, alors x^* est efficace pour P ;

(iii) si $\lambda \in \mathbb{R}_{\geq}^p$ et x^* est l'unique solution optimale de P_λ , alors x^* est efficace pour P .

Définition 1.2.6. Les vecteurs $\lambda \in \mathbb{R}_{\geq}^p$ sont appelés *poids*.

Cette méthode de calcul, populaire grâce à sa simplicité de mise en œuvre, a conduit à la distinction des solutions efficaces et des points non dominés en deux catégories :

- les points non dominés *supportés*, se trouvant sur la frontière de l'enveloppe convexe de Y_N . Les ensembles de points non dominés supportés et des solutions efficaces supportées correspondantes sont notés Y_{SN} et X_{SE} . Ces points sont obtenus par la résolution de sommes pondérées.
- les points non dominés *non supportés*, constitués des points de Y_N situés à l'intérieur de son enveloppe convexe. Les ensembles de points non dominés non supportés et des solutions efficaces non supportées correspondantes sont notés Y_{NN} et X_{NE} .

Par ailleurs, on peut encore distinguer deux types de solutions parmi X_{SE} :

- les solutions efficaces supportées *extrêmes* dont l'image se situe sur un sommet de $\text{conv}(Y)$. Ces solutions forment X_{SE_1} et $Y_{SN_1} = z(X_{SE_1})$ est l'ensemble des points non dominés supportés extrêmes ;
- les solutions efficaces supportées *non extrêmes* dont l'image n'est pas sur un sommet de $\text{conv}(Y)$. Ces solutions forment X_{SE_2} et $Y_{SN_2} = z(X_{SE_2})$ est l'ensemble des points non dominés supportés non extrêmes.

Remarque 1.2.2 – La classification de la définition 1.2.1 peut également s'appliquer aux ensembles X_{NE} , X_{SE} , X_{SE_1} et X_{SE_2} . On peut donc considérer les ensembles X_{NE_m} , X_{SE_m} , $X_{SE_{1m}}$, $X_{SE_{2m}}$, X_{NE_M} , X_{SE_M} , $X_{SE_{1M}}$ et $X_{SE_{2M}}$

La principale difficulté réside dans la détermination des solutions non supportées puisqu'on ne dispose pas de méthode algorithmiquement simple pour les calculer, comme c'est le cas pour les solutions supportées.

Exemple 1.2.1 – Soit le programme linéaire biobjectif suivant :

$$\begin{aligned}
 \min z_1 &= -x_1 - 4x_2 \\
 \min z_2 &= -2x_1 + 2x_2 \\
 \text{s.c.} \quad & -x_1 + 2x_2 \leq 9 \\
 & 6x_1 - 2x_2 \leq 30 \\
 & x_1 + 2x_2 \leq 12 \\
 & x_i \geq 0 \quad i = 1, 2
 \end{aligned}
 \tag{MOPL1}$$

Les ensembles admissible et réalisable de **MOPL1** sont représentés sur les figures 1.1(a) et 1.1(b). En particulier, le polytope X est délimité par cinq solutions extrêmes :

- $x^1 = (0, 0)$ dont l'image est $y^1 = (0, 0)$;
- $x^2 = (0, 4.5)$ dont l'image est $y^2 = (-18, 9)$;
- $x^3 = (1.5, 5.25)$ dont l'image est $y^3 = (-22.5, 7.5)$;
- $x^4 = (6, 3)$ dont l'image est $y^4 = (-18, -6)$;
- $x^5 = (5, 0)$ dont l'image est $y^5 = (-5, -10)$.

Ici, une infinité de solutions sont efficaces : les segments $[x^3x^4]$ et $[x^4x^5]$. On peut remarquer que ces solutions peuvent être décrites en utilisant uniquement les trois solutions x^3 , x^4 ,

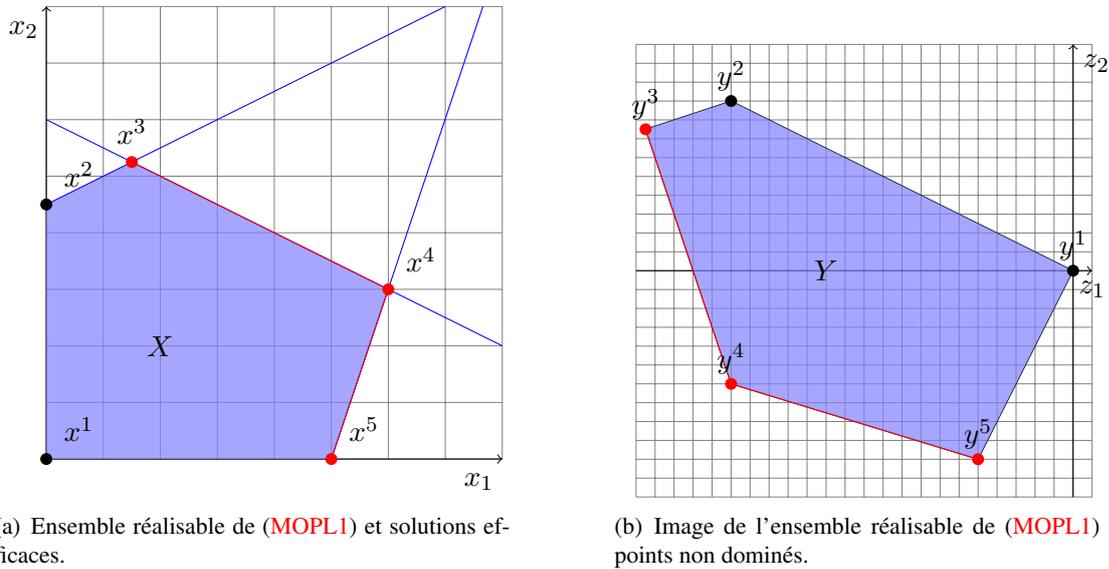


Figure 1.1 – Ensemble réalisable du problème MOPL1 dans l'espace des décisions et l'espace des objectifs

x^5 . On dit que ces solutions sont *efficaces extrêmes*. Généralement, elles sont utilisées pour décrire implicitement l'ensemble efficace et l'ensemble non dominé, et leur nombre peut augmenter exponentiellement avec la taille du problème ([Ehrgott, 2005]).

Théorème 1.2.2 ([Isermann, 1974]). *Toutes les solutions efficaces d'un MOLP sont supportées.*

Dans l'exemple 1.2.1, on a donc $X_E = X_{SE} = [x^3x^4] \cup [x^4x^5]$, $X_{SE_1} = \{x^3, x^4, x^5\}$ et $X_{SE_2} =]x^3x^4[\cup]x^4x^5[$. Afin d'illustrer les autres ensembles introduits, considérons un exemple de programme discret.

Exemple 1.2.2 – Soit le programme linéaire biobjectif suivant :

$$\begin{aligned}
 \min z_1 &= -x_1 - 4x_2 \\
 \min z_2 &= -2x_1 + 2x_2 \\
 \text{s.c.} \quad &-x_1 + 2x_2 \leq 9 \\
 &6x_1 - 2x_2 \leq 30 \\
 &x_1 + 2x_2 \leq 12 \\
 &x_i \in \mathbb{N} \quad i = 1, 2
 \end{aligned}
 \tag{MOIPL1}$$

Les ensembles admissible et réalisable de MOIPL1 sont représentés sur les figures 1.2(a) et 1.2(b).

Dans cet exemple :

- $X_E = \{(2, 5), (4, 4), (6, 3), (5, 0), (5, 1)\}$ et $Y_N = \{(-22, 6), (-20, 0), (-18, -6), (-5, -10), (-9, -8)\}$
- $X_{SE} = \{(2, 5), (4, 4), (6, 3), (5, 0)\}$ et $Y_{SN} = \{(-22, 6), (-20, 0), (-18, -6), (-5, -10)\}$
- $X_{NE} = \{(5, 1)\}$ et $Y_{NN} = \{(-9, -8)\}$
- $X_{SE_1} = \{(2, 5), (6, 3), (5, 0)\}$ et $Y_{SN_1} = \{(-22, 6), (-18, -6), (-5, -10)\}$
- $X_{SE_2} = \{(4, 4)\}$ et $Y_{SN_2} = \{(-20, 0)\}$

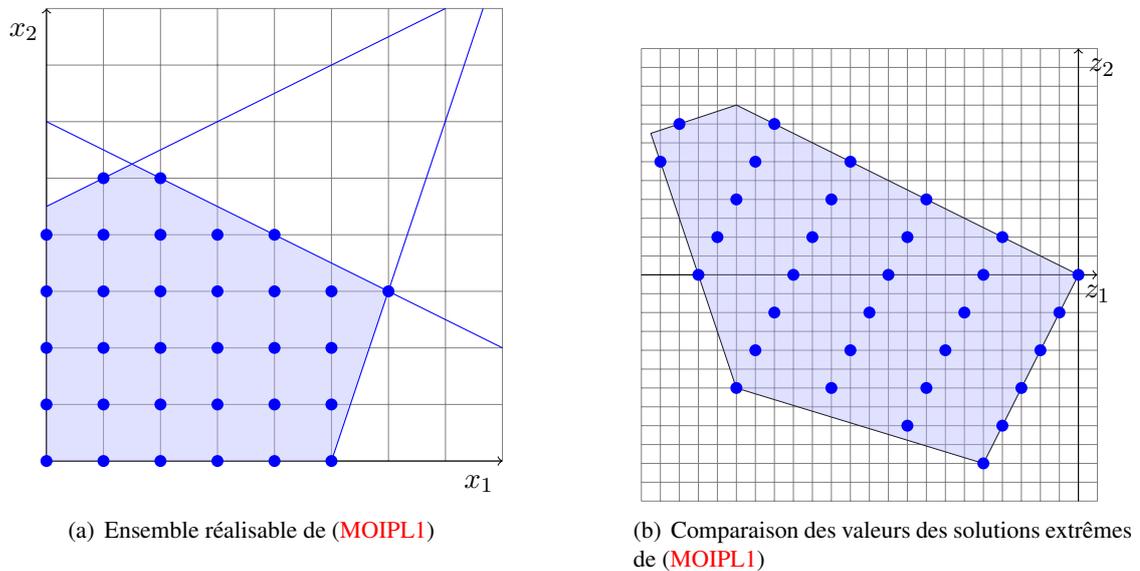


Figure 1.2 – Ensemble réalisable du problème MOIPL1 dans l'espace des décisions et l'espace des objectifs

Il existe donc des solutions non supportées dans les MOILP.

Maintenant que les caractéristiques principales de l'optimisation multiobjectif ont été abordées, la suite de ce chapitre présente les différentes méthodes de résolution.

1.3 Méthodes de résolution des programmes linéaires multiobjectifs

De nombreuses extensions de l'algorithme du simplexe ont été proposées pour résoudre les problèmes multiobjectifs [Philip, 1972, Zeleny, 1974, Gal, 1977, Isermann, 1977, Philip, 1977, Ecker et Kouada, 1978, Steuer, 1986, Armand et Malivert, 1991, Strijbosch *et al.*, 1991]. Tous ces algorithmes ont pour principe d'énumérer toutes les solutions de base efficaces. Un de leur principaux défauts provient de la nécessité d'énumérer plusieurs fois une même solution si le problème est dégénéré, ce qui est un phénomène courant.

D'autres méthodes s'appuient sur le fait que l'ensemble réalisable Y est souvent bien plus petit que l'ensemble admissible X en terme de dimensions et accomplissent donc leur recherche dans l'espace des objectifs [Dauer et Liu, 1990, Dauer et Saleh, 1990, Dauer, 1993, Dauer et Gallagher, 1996, Benson, 1998a, Benson, 1998b, Benson, 1998c]. En plus de parcourir un espace plus petit (en terme de dimensions), ces méthodes ne sont pas affectées par les problèmes de dégénérescence ou d'énumération de solutions équivalentes.

Seules les méthodes de résolution utilisées par la suite sont présentées ici.

1.3.1 Simplexe multiobjectif

Une première méthode de résolution des MOLP consiste en une extension de l'algorithme du simplexe servant à résoudre les problèmes mono-objectifs.

Elle repose sur des notions d'adjacence entre les solutions. Soit $G = (N, A)$ le graphe d'adjacence d'un MOLP dont les sommets sont les solutions de base efficaces et les arêtes représentent les pivots permettant de passer d'une solution de base à une autre. [Isermann, 1977] a montré que ce graphe est connexe, ce qui signifie qu'il est possible de passer d'une solution efficace à toute autre solution efficace en n'effectuant que des pivots passant par d'autres solutions efficaces.

Cette méthode de simplexe multiobjectif est décrite dans [Ehrgott, 2005] et permet d'obtenir toutes les solutions efficaces extrêmes et, de façon implicite, toutes les faces efficaces de \mathbb{R}^n et toutes les faces non dominées de \mathbb{R}^p . Il est couramment admis que l'ensemble des solutions efficaces extrêmes suffit à décrire X_E et Y_N , mais nous verrons plus tard dans cette thèse que cela a conduit à des erreurs dans le cadre de la résolution des MOMILP.

Nous présentons maintenant les grandes lignes d'un algorithme spécifique aux problèmes biobjectifs avant de considérer le cas multiobjectif.

Simplexe biobjectif

Comme déjà mentionné, le cas des problèmes biobjectifs est souvent sensiblement plus simple à traiter que celui des problèmes à trois objectifs et plus. La résolution des MOLP biobjectifs (BOLP) ne déroge pas à cette règle et un algorithme de simplexe simplifié permet le traitement de ces problèmes.

Considérons le BOLP suivant :

$$\begin{array}{ll} \min & (z_1(x), z_2(x)) \\ \text{s.c.} & x \in X \end{array} \quad (\text{BOLP1})$$

D'après le théorème 1.2.2, nous savons qu'il est équivalent de calculer les solutions efficaces de BOLP1 et de résoudre le PL

$$\begin{array}{ll} \min & \lambda_1(c^1)^T x + \lambda_2(c^2)^T x \\ \text{s.c.} & x \in X \end{array} \quad (1.1)$$

pour tout $\lambda \in \mathbb{R}_+^2$. En normalisant, on pose $(\lambda_1, \lambda_2) = (\lambda, 1 - \lambda)$ où $\lambda \geq 0$, ce qui nous permet d'obtenir la fonction objectif suivante

$$c(\lambda) := \lambda c^1 + (1 - \lambda)c^2. \quad (1.2)$$

Ainsi, il nous faut donc résoudre :

$$\begin{array}{ll} \min & c(\lambda)^T x \\ \text{s.c.} & x \in X \end{array} \quad (1.3)$$

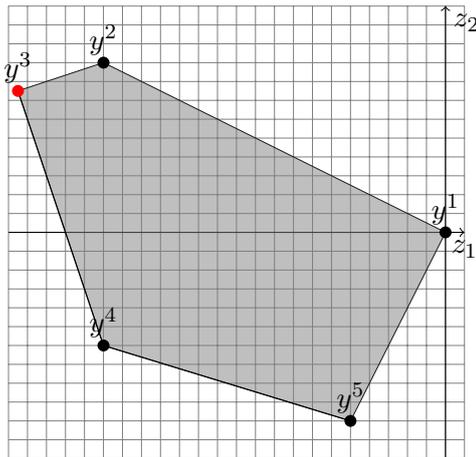
Il s'agit d'un problème paramétrique dont il faut déterminer les valeurs de $\lambda \in [0, 1]$ à calculer. Puisque les solutions efficaces sont naturellement ordonnées dans le cas de l'optimisation biobjectif, il suffit donc de résoudre le problème pour $\lambda = 1$ puis de faire progressivement diminuer λ jusqu'à 0 pour déterminer X_E . Cela est réalisé en calculant les poids critiques tels que la solution efficace considérée reste optimale pour le problème considéré (P_λ).

Exemple 1.3.1 – On cherche à résoudre le problème MOPL1 présenté en page 13. La première étape consiste à résoudre le problème MOPL1₍₁₎. Le point y^3 est obtenu (figure 1.3(a)). Ensuite, le premier pivot permet d'obtenir y^4 avec $\lambda = \frac{3}{4}$ (figure 1.3(b)). Puis y^5 est obtenu au pivot suivant avec $\lambda = \frac{6}{19}$ (figure 1.3(c)). Il n'existe plus de pivot possible, donc l'algorithme termine.

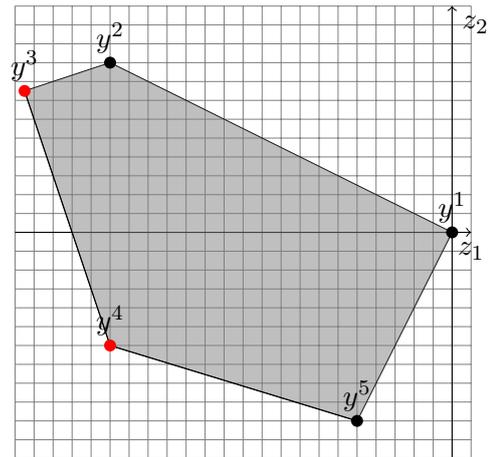
Les solutions efficaces extrêmes sont donc :

- x^3 pour $\lambda \in [\frac{3}{4}, 1]$
- x^4 pour $\lambda \in [\frac{6}{19}, \frac{3}{4}]$
- x^5 pour $\lambda \in [0, \frac{6}{19}]$

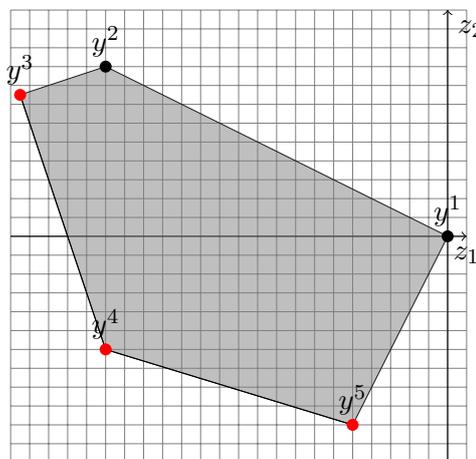
Cette résolution est représentée par la figure 1.3.



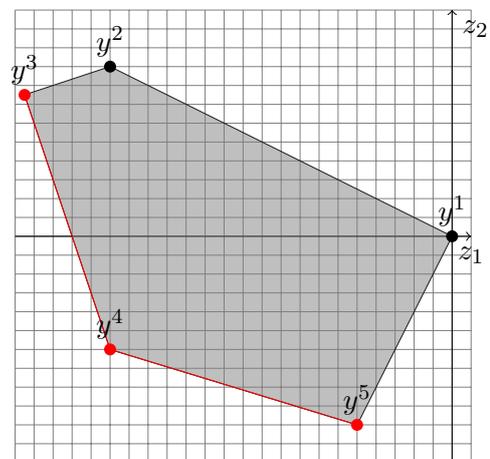
(a) y^3 est le premier point obtenu.



(b) y^4 est obtenu après le premier pivot.



(c) y^5 est obtenu après le deuxième et dernier pivot.



(d) Les trois points obtenus décrivent l'ensemble des points non dominés.

Figure 1.3 – Résolution du problème MOPL1 par l'algorithme du simplexe biobjectif paramétrique.

Simplexe multiobjectif

Le principe du simplexe multiobjectif est similaire, mais le nombre d'objectifs fait qu'il faut maintenant gérer au moins deux paramètres dans la fonction objectif. Les solutions efficaces étant

connectées [Isermann, 1977], on peut à nouveau toutes les obtenir en partant de l'une d'entre elles et en effectuant des pivots.

En revanche, puisque plusieurs paramètres varient désormais, il n'est plus possible de calculer les solutions efficaces aussi directement que dans le cadre biobjectif. En effet, si une solution efficace est connectée à plusieurs autres solutions efficaces pas encore obtenues, alors chacune d'entre elles doit être considérée ainsi que les solutions auxquelles elle est connectée, puis il faut revenir en arrière pour explorer les autres solutions. Puisqu'un pivot de simplexe est nécessaire à chaque déplacement, ces potentiels allers-retours peuvent devenir coûteux.

1.3.2 Méthode d'approximation externe de Benson

Des méthodes ont été proposées pour résoudre les MOLP en se déplaçant dans l'espace des objectifs au lieu de l'espace des décisions. C'est le cas de la méthode d'approximation externe de Benson [Benson, 1998c].

La méthode de Benson, malgré son nom d'approximation externe (*outer approximation*), est bien une méthode exacte. Son nom provient du fait qu'il s'agit d'une approximation de l'ensemble non dominé qui est petit à petit raffinée jusqu'à aboutir à l'ensemble exact. Elle consiste en effet en la détermination, dans un premier temps, d'un ensemble

$$Y' = \{y \in \mathbb{R}^p : Cx \leq y \leq \hat{y} \text{ pour } x \in X\}.$$

où \hat{y} est un point strictement dominé par $y^{AI} := \max\{y_k : y \in Y\}$. Un point $\hat{p} \in \text{int}(Y')$ et un simplexe S à p dimensions englobant Y' sont également déterminés.

Ensuite, pour chaque sommet s du simplexe n'appartenant pas à Y' , on détermine le point du segment $[s\hat{p}]$ se trouvant sur la frontière de Y' . À partir de ce point, on calcule l'hyperplan h contenant la face de Y à laquelle le point appartient. On supprime ensuite de S le demi espace délimité par l'hyperplan h et contenant s .

En répétant ce processus jusqu'à ce que tous les points de S soient dans Y' , on *sculpte* donc S petit à petit pour que S_N prenne la forme de $Y'_N = Y_N$.

La figure 1.4 montre l'exécution de cet algorithme quand il est appliqué au problème MOPL1 de la page 13.

Plusieurs améliorations ont été apportées à cette méthode dans [Shao et Ehrgott, 2008], notamment lors de la détermination du point de l'arête $[s\hat{p}]$ et pour le calcul des connexions entre les points de l'ensemble résultant après une coupe.

1.4 Méthodes de résolution programmes linéaires multiobjectifs en variables entières et combinatoires

De la même façon que dans le contexte mono-objectif, les programmes multiobjectifs en variables entières ou mixtes nécessitent d'énumérer des solutions admissibles. Le cadre de cette thèse étant les programmes en variables binaires, nous précisons dès maintenant notre propos aux problèmes dont les variables discrètes sont binaires.

Dans cette section, nous nous étendons plus particulièrement sur les deux méthodes de résolution que sont l'algorithme multiobjectif de *Branch and Bound* et la méthode en deux phases car ces méthodes sont centrales dans les propositions algorithmiques de la thèse. Les autres méthodes de résolution, comme la programmation dynamique ou l' ϵ -contrainte, ne seront pas abordées.

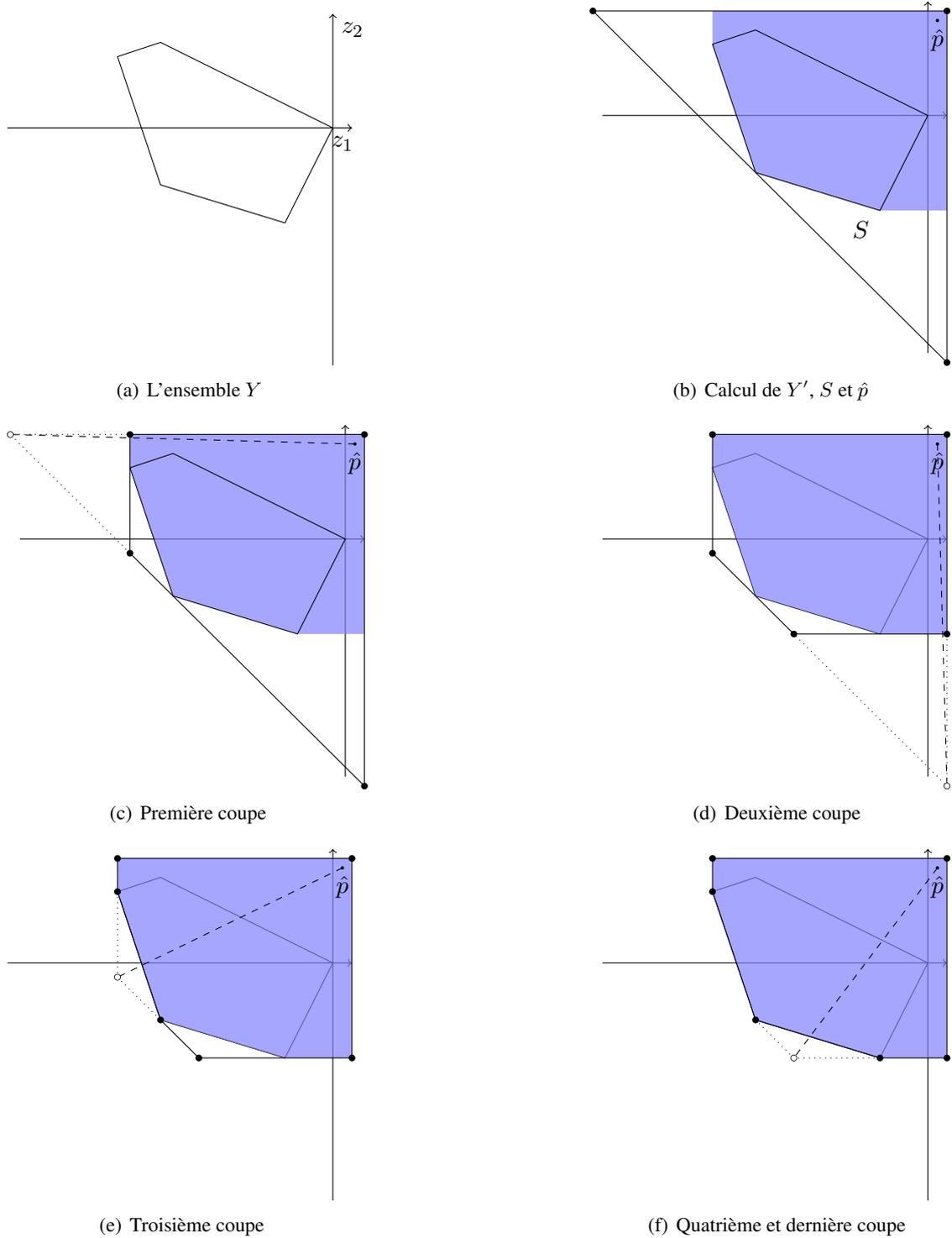


Figure 1.4 – Illustration de la méthode d'approximation externe de Benson.

1.4.1 Ensembles bornant

Bornes dans le contexte mono-objectif

Pour restreindre efficacement leur espace de recherche, les méthodes de résolution de problèmes mono-objectifs en variables entières ont recours à des bornes permettant d'estimer le plus précisément possible la valeur de la solution optimale. Le principe consiste donc à encadrer celle-ci au moyen de deux bornes : une borne *inférieure* et une borne *supérieure*.

La borne inférieure l est généralement fournie par une résolution du problème courant relâché (relaxation linéaire, lagrangienne, surrogate, etc.). Cette borne représente une valeur inférieure ou égale à la valeur optimale et n'est généralement pas une solution admissible.

La borne supérieure u d'un problème est obtenue en déterminant des solutions admissibles de ce problème. Cette borne est généralement calculée au moyen d'heuristiques en début de résolution puis est resserrée au fur et à mesure que de nouvelles solutions admissibles sont trouvées.

L'objectif est donc de serrer le plus possible ces deux bornes, c'est-à-dire d'avoir la borne inférieure la plus grande possible et la borne supérieure la plus petite possible. Ceci permet d'éviter des énumérations inutiles et donc d'aboutir aussi rapidement que possible à la solution optimale du problème.

Remarque 1.4.1 – En maximisation, les rôles des bornes sont inversés.

Ensembles bornant dans le contexte multiobjectif

De même que dans le cadre mono-objectif, l'efficacité des algorithmes de résolution dépend grandement de celle des bornes utilisées. Il est donc nécessaire d'étudier celles-ci dans un contexte multiobjectif. Avec le passage à plusieurs objectifs, le but est maintenant de borner l'ensemble non dominé. Une proposition naturelle lors du passage à plusieurs objectifs est de transposer les bornes inférieures et supérieures en utilisant respectivement le point idéal y^I et le point nadir y^N , définis comme suit :

$$y_k^I = \min_{x \in X} c^k x; \quad k = 1, \dots, p$$

$$y_k^N = \max_{x \in X_E} c^k x; \quad k = 1, \dots, p$$

En définissant des bornes par un unique point, y^I et y^N sont les bornes les plus serrées possibles. De fait, on a par définition $y^I \leq y \leq y^N$ pour tout point $y \in Y_N$ et serrer davantage ces points reviendrait à ne plus encadrer la totalité des points non dominés. y^I et y^N souffrent néanmoins de défauts jouant en défaveur de ces bornes. Tout d'abord, il est très improbable de parvenir à une situation d'égalité entre elles du fait de la nature conflictuelle des objectifs. Ce certificat d'optimalité utilisé en optimisation mono-objectif ne serait donc plus applicable dans le présent contexte. D'autre part, il a été constaté que ces bornes sont de plus relativement éloignées de l'ensemble non dominé (voir par exemple [Ehrgott et Gandibleux, 2007]). Ceci est également imputable aux objectifs conflictuels et n'augure pas d'une résolution efficace.

Pour pallier cette relative inefficacité des bornes y^I et y^N , il a par conséquent été proposé de recourir à des ensembles de points pour encadrer Y_N . Différentes définitions formelles des ensembles bornant ont été données dans la littérature ([Villarreal et Karwan, 1981,

[Ehrgott et Gandibleux, 2001, Ehrgott et Gandibleux, 2007]). Nous présentons brièvement ces définitions dans le but de poser un cadre dans lequel se place chaque ensemble bornant possible. On remarquera que pour chacune d'entre elles, $\{y^I\}$ et $\{y^N\}$ sont des ensembles bornant valides.

La première définition a été donnée par [Villarreal et Karwan, 1981], bien que le terme d'ensemble bornant n'ait pas été utilisé à l'époque.

Définition 1.4.1 (Ensembles bornant d'après [Villarreal et Karwan, 1981]). Un ensemble bornant inférieurement (EBI) L pour la solution d'un MOILP est un ensemble de points satisfaisant les conditions suivantes :

- (i) chaque point non dominé du problème est un élément de L ou est dominé par un élément de L ;
- (ii) chaque élément de L est non dominé ou domine au moins un des points non dominés du problème.

Un ensemble bornant supérieurement (EBS) U pour la solution d'un MOILP est un ensemble de points satisfaisant la condition suivante :

- (iii) chaque élément de U est non dominé ou est dominé par au moins un point non dominé du problème.

On peut remarquer que cette définition permet d'avoir l'égalité $L = U = Y_N$. Cependant, l'ensemble L utilisé dans un cadre de programmation dynamique par [Villarreal et Karwan, 1981] est le point idéal du problème ou d'une de ses relaxations, ce qui fait que les auteurs utilisent en pratique un unique point en guise d'ensemble bornant inférieurement.

En ce qui concerne U , les auteurs proposent d'utiliser l'ensemble des solutions réalisables connues filtrées par dominance. Nous verrons que cet ensemble bornant est très couramment utilisé.

La définition de [Ehrgott et Gandibleux, 2001] a été la première à parler explicitement d'ensemble bornant.

Définition 1.4.2 (Ensembles bornant d'après [Ehrgott et Gandibleux, 2001]). Un ensemble bornant inférieurement $L \subset \mathbb{R}^p$ pour \bar{Y} est un ensemble tel que :

- (i) pour tout $y \in \bar{Y}$, il existe $l = (l^1, \dots, l^p) \in L$ tel que $l \leq y$;
- (ii) il n'existe pas de points $y \in \bar{Y}$ et $l \in L$ tels que $y \leq l$.

Un ensemble bornant supérieurement $U \subset \mathbb{R}^p$ pour \bar{Y} est un ensemble tel que :

- (iii) pour tout $y \in \bar{Y}$, il existe $u = (u^1, \dots, u^p) \in U$ tel que $y \leq u$;
- (iv) il n'existe pas de points $y \in \bar{Y}$ et $u \in U$ tels que $u \leq y$.

Cette définition est plus générale que la précédente puisqu'elle vise à borner un sous ensemble $\bar{Y} \subset Y$ de points réalisables. En pratique, cela permet d'appliquer les bornes à un ensemble \bar{Y} obtenu en particulier lors de la résolution d'un sous-problème.

À la différence de la précédente, cette définition permet de considérer des EBI L tels que $(L + \mathbb{R}_{\geq}^p)$ soit convexe, comme $L = \text{conv } Y_{SN}$ ou la relaxation continue. Cependant, un sous-ensemble de points réalisables filtrés par dominance ne définit pas ici un EBS.

Un ensemble $S \subset \mathbb{R}_{\geq}^p$ est dit \mathbb{R}_{\geq}^p -fermé si l'ensemble $S + \mathbb{R}_{\geq}^p$ est fermé, et \mathbb{R}_{\geq}^p -borné si l'ensemble $S + \mathbb{R}_{\geq}^p$ est borné. $cl(S)$ est la fermeture de S .

Définition 1.4.3 (Ensembles bornant d'après [Ehrgott et Gandibleux, 2007]). Un ensemble bornant inférieurement L pour \bar{Y} est un ensemble $L \subset \mathbb{R}^p$, \mathbb{R}_{\geq}^p -fermé et \mathbb{R}_{\geq}^p -borné, tel que $\bar{Y} \subset L + \mathbb{R}_{\geq}^p$ et $L \subset (L + \mathbb{R}_{\geq}^p)_N$.

Un ensemble bornant supérieurement U pour \bar{Y} est un ensemble $U \subset \mathbb{R}^p$, \mathbb{R}_{\geq}^p -fermé et \mathbb{R}_{\geq}^p -borné, tel que $\bar{Y} \subset cl[(U + \mathbb{R}_{\geq}^p)^c]$ et $U \subset (U + \mathbb{R}_{\geq}^p)_N$.

La définition de L se justifie tout d'abord par le fait qu'on ne souhaite pas que les points de cet ensemble puissent se dominer les uns les autres. On a donc $L \subset (L + \mathbb{R}_{\geq}^p)_N$. Par ailleurs, puisque L se doit de borner \bar{Y} en étant *au-dessous*, il est nécessaire qu'il existe pour chaque point de \bar{Y} un point de L le dominant, ce qui implique $\bar{Y} \subset L + \mathbb{R}_{\geq}^p$.

Une partie de la définition de U est similaire à celle de L . En effet, on souhaite également que les points de U ne se dominent pas les uns les autres, ce qui nous donne $U \subset (U + \mathbb{R}_{\geq}^p)_N$. Enfin, aucun point de \bar{Y} ne peut être dominé par un point de U , donc $\bar{Y} \not\subset (U + \mathbb{R}_{\geq}^p)$. Autrement dit, \bar{Y} doit être dans $(U + \mathbb{R}_{\geq}^p)^c$ et $cl[(U + \mathbb{R}_{\geq}^p)^c]$ est utilisé pour obtenir un ensemble \mathbb{R}_{\geq}^p -fermé. En particulier, tout ensemble de points réalisables filtrés par dominance peut être considéré comme un ensemble bornant supérieurement valide.

Plusieurs ensembles bornant peuvent être considérés. Tout d'abord, on peut noter que y^I et y^N sont respectivement des ensembles bornant inférieurement et supérieurement valides. De plus, un ensemble bornant supérieurement couramment utilisé est un sous-ensemble des points non dominés ou de points non filtrés par dominance. Quant aux ensembles bornant inférieurement, on peut mentionner la relaxation convexe ou la relaxation continue.

Dans cette thèse, nous nous basons sur la définition de [Ehrgott et Gandibleux, 2007] dans un contexte de minimisation. Nous aborderons par la suite quelques-unes des propositions de la littérature.

1.4.2 Branch and bound

La procédure de séparation et évaluation (*Branch & Bound*) consiste en une énumération implicite des solutions, c'est-à-dire que toutes les combinaisons possibles de valuation des variables binaires sont évaluées, bien qu'elles ne soient pas explorées explicitement.

De même que pour le Branch & Bound mono-objectif, cet arbre est construit durant la résolution afin de représenter toutes les combinaisons de valeurs possibles pour les variables binaires (voir figure 1.5). Chaque nœud de l'arbre représente donc un sous-problème pour lequel certaines variables binaires sont fixées. Les variables dont la valeur n'est pas encore fixée sont appelées *variables libres*. Le nœud racine représente le problème original à résoudre sans variable binaire fixée et les feuilles de l'arbre correspondent aux sous-problèmes pour lesquels toutes les variables binaires sont fixées.

Nous présentons dans cette section comment ses principaux composants sont étendus pour former un algorithme multiobjectif. De plus, quelques algorithmes de Branch & Bound de la littérature sont passés en revue au regard des différents composants. Bien que cette liste ne puisse être exhaustive, nous l'estimons néanmoins représentative de ce qui a été proposé et des évolutions des méthodes au cours des 30 dernières années.

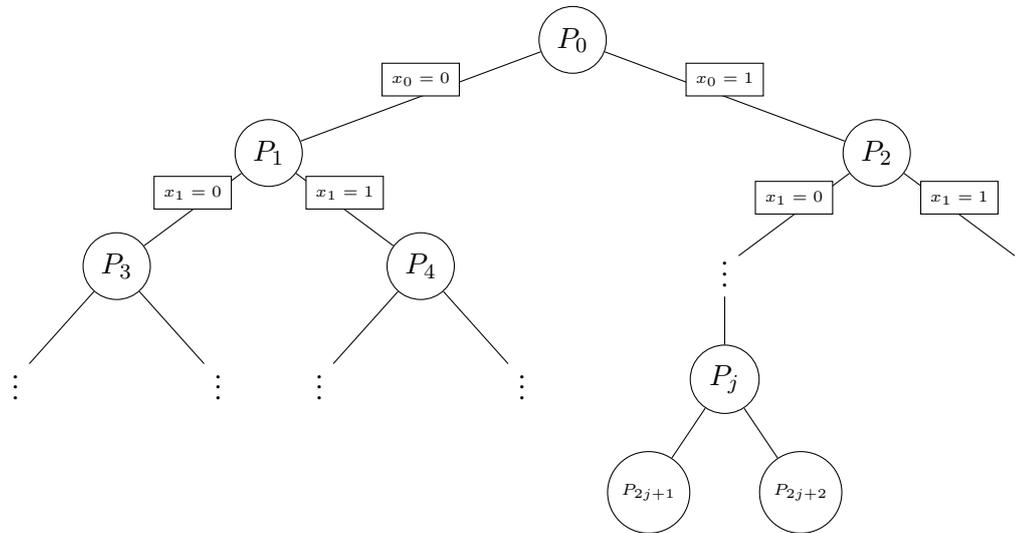


Figure 1.5 – Exemple d'arbre binaire.

Problèmes traités

Parmi les algorithmes de Branch & Bound de la littérature, la plupart sont dédiés à un problème particulier. On trouve par exemple le problème de sac à dos, qu'il soit unidimensionnel biobjectif ([Ulungu et Teghem, 1997, Visée *et al.*, 1998]), unidimensionnel triobjectif ([Jorge, 2010]) ou multidimensionnel multiobjectif ([Florios *et al.*, 2010]), mais aussi le problème d'arbre couvrant de poids minimum biobjectif ([Ramos *et al.*, 1998, Sourd et Spanjaard, 2008]) et le problème d'affectation biobjectif ([Delort, 2011]). Cependant, d'autres articles s'attachent à résoudre des problèmes génériques en variables binaires ([Kiziltan et Yucaoglu, 1983]). Néanmoins, toutes les méthodes considérées sont réduites à l'utilisation de variables binaires (à l'exception de [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005] qui traitent des problèmes en variables continues et binaires).

Adaptation des composants du *Branch and Bound* multiobjectif

Par souci d'homogénéité et pour faciliter les comparaisons, nous parlerons des ensembles bornant dans le contexte de minimisation.

Ensemble bornant supérieurement L'ensemble bornant supérieurement U s'applique au problème global et borne donc Y_N . Au début de l'algorithme, il peut être initialisé avec le point (∞, \dots, ∞) ou de préférence avec des points réalisables filtrés par dominance obtenus au moyen d'une heuristique. U est potentiellement mis à jour dès que de nouveaux points réalisables sont obtenus et est égal à Y_N à la fin de l'algorithme. Pour toutes les méthodes présentées, l'ensemble bornant utilisé est l'ensemble des points potentiellement non dominés. Il s'agit en effet de points réalisables déjà obtenus et pour lesquels on n'a pas (encore) trouvé de points réalisables les dominant. Ce choix s'impose naturellement et il est donc important d'obtenir de bonnes solutions dès que possible afin d'améliorer le sondage par dominance.

Ensemble bornant inférieurement Un ensemble bornant inférieurement L est calculé pour chaque nœud considéré. L borne l'ensemble non dominé \bar{Y}_N du sous-problème correspondant et est utilisé pour évaluer si le nœud courant peut être sondé par optimalité ou par dominance. Nous nous intéressons ici aux ensembles bornant inférieurement en minimisation (supérieurement en maximisation). On constate que la principale évolution concerne le passage d'une borne réduite à un point à un ensemble bornant.

En effet, une des premières propositions a été d'utiliser un point utopique y^U dont chaque coordonnée est calculée indépendamment. Dans [Kiziltan et Yucaoglu, 1983], les coordonnées sont calculées en assignant à chaque variable la valeur la plus avantageuse pour l'objectif concerné, sans tenir compte des contraintes. Plus tard, dans [Ulungu et Teghem, 1997], [Visée et al., 1998] et [Jorge, 2010], y^U est calculé en utilisant des bornes mono-objectifs dédiées au problème de sac à dos. Bien qu'il s'agisse toujours d'un point utopique, la borne est néanmoins plus serrée. y^{ILP} , le point idéal de la relaxation linéaire du problème courant, est également utilisé ([Florios et al., 2010]). Il s'agit toujours d'un point utopique.

Enfin, un ensemble bornant consistant en une relaxation convexe a été utilisé plus récemment ([Sourd et Spanjaard, 2008], [Jorge, 2010] et [Delort, 2011]). La relaxation convexe L est la borne la plus serrée telle que $(L + \mathbb{R}_{\geq}^p)$ soit convexe. Bien que plus coûteux, cet ensemble est l'ensemble bornant convexe le plus serré possible et fournit de plus des points réalisables qui permettront de renforcer l'ensemble bornant supérieurement. Il n'est néanmoins utilisable que lorsqu'il est algorithmiquement possible de résoudre rapidement le problème mono-objectif correspondant.

Sondage par dominance Sonder un nœud par dominance requiert de comparer son ensemble bornant inférieurement L à l'actuel ensemble bornant supérieurement U du problème global. Si, pour tout $l \in L$, il existe un $u \in U$ tel que $u \leq l$, alors tous les points réalisables du sous-problème considéré sont faiblement dominés. Ainsi, aucun nouveau point non dominé ne pourra être obtenu de la branche courante et son exploration peut être interrompue (Figure 1.6(a)).

En revanche, il suffit qu'un point de l'ensemble bornant inférieurement ne soit pas faiblement dominé pour que le nœud actif soit considéré intéressant et ne soit pas sondé par dominance (Figure 1.6(b)).

Sondage par optimalité Dans le cadre mono-objectif, le sondage par optimalité consiste simplement en un test de réalisabilité de la borne inférieure du sous-problème courant. Malheureusement, ce sondage n'est plus utilisable dans le contexte multiobjectif. En effet, obtenir une solution (ou un ensemble de solutions) réalisable(s) permet d'affirmer que ces solutions sont efficaces pour le sous-problème considéré, mais ne permet pas d'affirmer que les solutions définissent l'ensemble non dominé de ce sous-problème. La seule exception à cela serait que l'ensemble bornant inférieurement obtenu soit réduit à un point (idéal) réalisable. Il est donc plus rare de sonder des sommets par optimalité que dans le cas mono-objectif.

Sondage par inadmissibilité Enfin, le sondage par inadmissibilité n'est pas affecté par l'ajout d'objectifs puisqu'il ne concerne que l'espace des décisions. Celui-ci consiste en effet à sonder les nœuds dont les sous-problèmes ne sont pas réalisables.

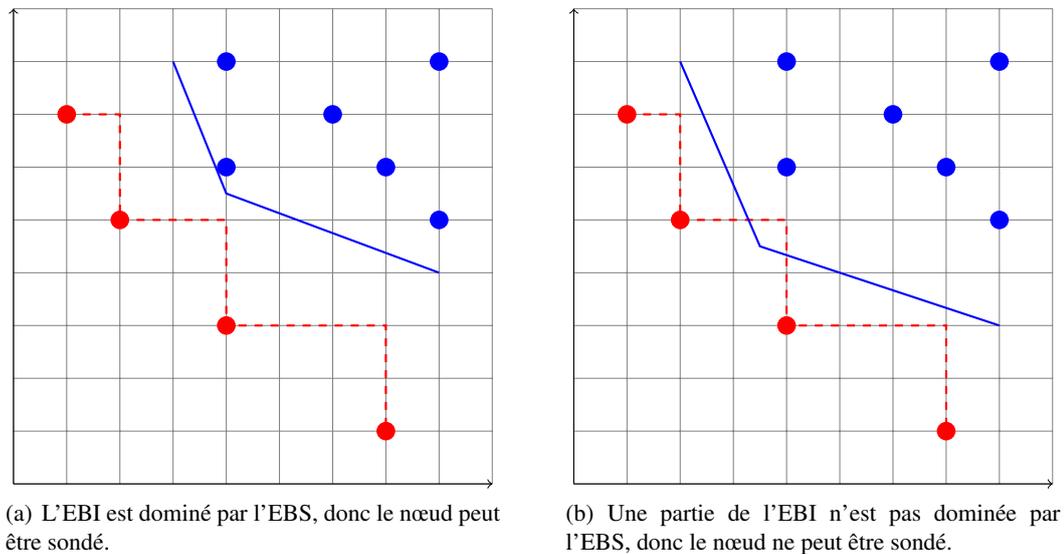


Figure 1.6 – Comparaison de l'EBS d'un nœud (en trait plein bleu) avec l'EBS (en trait pointillé rouge) du problème global pour déterminer si un nœud peut être sondé ou non.

Obtention de nouveaux points Obtenir de nouveaux points réalisables (et potentiellement non dominés) permet de renforcer l'ensemble bornant supérieurement et donc de rapprocher la résolution de sa fin. Ce sujet constitue donc un composant à ne pas négliger.

La façon la plus simple d'obtenir de nouveaux points est d'atteindre les feuilles de l'arbre. En effet, toutes les variables binaires y sont fixées et la solution correspondante est donc immédiatement disponible.

Cependant, il est également possible d'obtenir des points réalisables plus tôt. Une initialisation est parfois faite afin de débiter l'algorithme avec de bonnes solutions et ainsi commencer avec un bon ensemble bornant supérieurement. Cette initialisation peut se faire en calculant Y_{SN} ([Ramos *et al.*, 1998]) ou au moyen d'une heuristique ([Delort, 2011]).

Par la suite, des heuristiques pour générer des points réalisables dans les nœuds intermédiaires peuvent être utilisées pour améliorer l'ensemble non dominé. Enfin, des solutions peuvent être obtenues en calculant l'ensemble bornant inférieurement du sous-problème considéré ([Sourd et Spanjaard, 2008, Jorge, 2010, Delort, 2011]).

Procédure de séparation La procédure de séparation, dont le but est de partitionner l'ensemble admissible X , ne subit pas de modification fondamentale par rapport à sa version mono-objectif. En effet, le nombre d'objectifs à optimiser simultanément n'a aucun impact sur l'espace des décisions, ce qui implique également que le sondage par inadmissibilité se fasse de la même manière. Les variables binaires n'ayant pas encore été assignées à une valeur sont appelées *variables libres*.

Dans le cas des programmes en variables binaires, la procédure de séparation systématiquement proposée dans la littérature consiste simplement à créer deux sous-problèmes du problème courant en fixant une variable libre x_i à 0 dans un cas et 1 dans l'autre. Fixer une variable binaire est simple et la principale difficulté est donc de déterminer quelle variable libre fixer. Pour cela, une stratégie de séparation peut être déterminée statiquement (avant la résolution) ou bien dynamiquement (au cours de la résolution).

Choix du nœud actif Bien que le choix du nœud actif de l'arbre de recherche s'applique à l'espace des décisions, il peut dépendre des résultats obtenus dans l'espace des objectifs dans le cas d'une exploration dynamique. Par conséquent, il paraît nécessaire d'adapter également cette procédure au cadre multiobjectif. Néanmoins, les méthodes proposées dans la littérature utilisent presque exclusivement un parcours statique et plus particulièrement un parcours en profondeur de l'arbre de recherche. En effet, il est plus difficile que dans le contexte mono-objectif d'évaluer et de comparer les nœuds candidats afin d'en déterminer le meilleur. Les seules exceptions à cela sont les méthodes proposées par [Jorge, 2010].

Récapitulatif Les composants choisis par les différentes méthodes abordées sont récapitulés dans le tableau 1.1.

Nous détaillons maintenant la méthode sur laquelle se base la suite de nos travaux.

Méthode de Mavrotas et Diakoulaki (1998,2005)

Cette méthode [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005] étant celle sur laquelle se base la suite de nos travaux sur la méthode de Branch & Bound, nous la présentons plus en détail.

La méthode proposée résout les MOM01LP, c'est-à-dire les programmes linéaires multiobjectifs en variables mixtes continu/binaire. Elle utilise pour cela un parcours en profondeur.

À chaque étape de l'algorithme, l'ensemble bornant supérieurement est tenu à jour. Cette liste sert à filtrer les solutions dominées tout en étant composée des solutions qui ne sont pas dominées pour le moment. À la fin de l'algorithme, cette liste doit donc correspondre à Y_N ou tout au moins contenir un sous ensemble de Y_N qui permet sa description complète.

La liste de solutions potentiellement non dominées de Mavrotas et Diakoulaki est notée D_{ex} et est vide à l'origine. Quand une feuille de l'arbre est atteinte, la liste D_{ex} est mise à jour. Le sous-problème correspondant à une feuille est un MOLP puisque toutes les variables binaires y ont été fixées. L'ensemble des points non dominés du MOLP est donné implicitement par ses points extrêmes formant un ensemble E . C'est cet ensemble qui est utilisé pour mettre à jour D_{ex} au moyen d'une comparaison paire à paire des points de E et D_{ex} .

Intéressons-nous maintenant au sondage des nœuds. Chaque nœud P_j définit un sous-problème pour lequel certaines variables binaires sont fixées et sa relaxation linéaire est considérée. En cas d'inadmissibilité, le nœud est sondé. Autrement, le point idéal de la relaxation linéaire est calculé et comparé à D_{ex} . Si un point de D_{ex} domine ce point idéal, alors le nœud est sondé par dominance. Si ce n'est pas le cas, alors les nœuds fils P_{2j+1} et P_{2j+2} , dans lesquels une variable binaire supplémentaire est fixée, sont créés pour que les sous-problèmes correspondants soient explorés.

L'article [Mavrotas et Diakoulaki, 2005] reprend le même algorithme et propose des améliorations permettant d'accélérer la résolution. Parmi ces propositions, l'accent est principalement mis sur l'amélioration des performances dans la résolution des MOLP (au moyen de l'algorithme du simplexe révisé, en particulier) et sur l'ajout d'un test permettant de filtrer des solutions dominées. En effet, les auteurs ont constaté que la comparaison paire à paire classique entre les points non dominés extrêmes obtenus dans les feuilles de l'arbre de recherche n'était pas suffisant pour écarter toutes les solutions dominées, ce qui pouvait donc mener à les conserver. Ce problème est décrit dans l'exemple 1.4.1.

| Auteurs | type de pb | opt | EBI | EBS | nouveaux points | nœud actif | séparation |
|--------------------------------|------------|-----|------------|------------|--------------------------|------------|------------|
| [Kiziltan et Yucaoglu, 1983] | MO01LP | max | liste | y^U | variables libres à 0 | profondeur | 1 puis 0 |
| [Ulungu et Teghem, 1997] | B01KP | max | liste | y^U | variables libres à 0 | profondeur | 1 puis 0 |
| [Visée <i>et al.</i> , 1998] | B01KP | max | liste | y^U | variables libres à 0 | profondeur | 1 puis 0 |
| [Ramos <i>et al.</i> , 1998] | BST | min | y^{LLP} | liste | init+feuilles | profondeur | 1 puis 0 |
| [Mavrotas et Diakoulaki, 1998] | MOM01LP | max | liste | y^{LLP} | MOLP aux feuilles | profondeur | 0 puis 1 |
| [Mavrotas et Diakoulaki, 2005] | MOM01LP | max | liste | y^{LLP} | MOLP aux feuilles | profondeur | 0 puis 1 |
| [Sourd et Spanjaard, 2008] | BST | min | relax conv | liste | relax conv à chaque nœud | profondeur | 1 puis 0 |
| [Florios <i>et al.</i> , 2010] | MOMDKP | max | liste | y^{LLP} | init arrondie + feuilles | profondeur | 1 puis 0 |
| [Jorge, 2010] | 3OKP | max | liste | y^U | calculEBS+feuilles | profondeur | 1 puis 0 |
| [Jorge, 2010] | 3OKP | max | liste | relax conv | calculEBS du nœud | dynamique | 1 puis 0 |
| [Delort, 2011] | 2AP | min | liste | relax conv | meta+relax conv | profondeur | 1 puis 0 |

Table 1.1 – Récapitulatif des choix de composants pour les méthodes abordées.

Exemple 1.4.1 – Considérons trois points y^1 , y^2 et y^3 obtenus lors de la résolution d'un MOMILP grâce à la procédure de séparation et évaluation de [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005]. Ces points sont représentés sur la figure 1.7. Les trois points ne se dominent pas entre eux et aucun n'est donc supprimé dans une comparaison paire à paire. En revanche, si y^1 et y^2 ont été obtenus dans la même feuille de l'arbre, alors l'arête $[y^1 y^2] = \text{conv}(y^1, y^2)$ est réalisable puisque l'ensemble du MOLP dont ils proviennent est convexe. Dans cette arête, certains points dominent y^3 (qui ne provient pas de la même feuille de l'arbre). C'est par exemple le cas de y' . y^3 est donc dominé et doit être supprimé de D_{ex} .

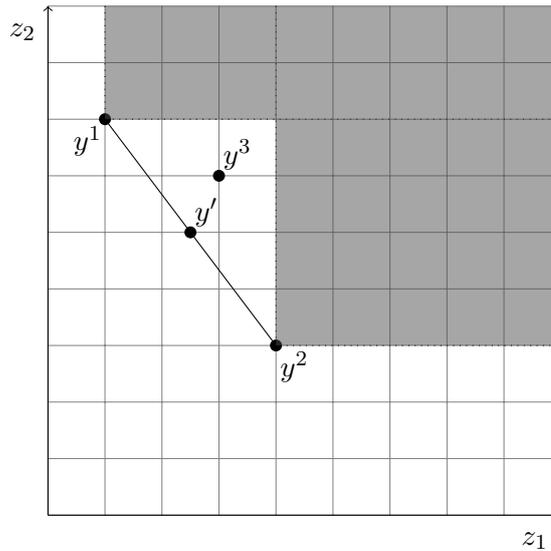


Figure 1.7 – y^3 n'est dominé ni par y^1 ni par y^2 mais par certaines de leurs combinaisons convexes comme y' .

Le moyen proposé pour remédier à ce problème consiste en un test de filtrage effectué a posteriori de l'énumération. Tout au long du parcours de l'arbre de recherche, les combinaisons de variables binaires correspondant aux points conservés dans D_{ex} sont listées. À la fin de la recherche, les combinaisons figurant toujours dans D_{ex} sont appelées *combinaisons efficaces*. Chaque point $y = (y_1, y_2, \dots, y_p) \in D_{ex}$ est alors testé par rapport aux points de chaque autre combinaison efficace. Cette combinaison efficace représente k points y^1, y^2, \dots, y^k qui sont utilisés pour former le programme linéaire suivant :

$$\begin{aligned}
 \max z = & \quad \epsilon_1 + \quad \epsilon_2 + \quad \dots + \quad \epsilon_p \\
 \text{s.c.} \quad & \lambda_1 y_1^1 + \lambda_2 y_1^2 + \dots + \lambda_k y_1^k - \epsilon_1 \geq y_1 \\
 & \dots \\
 & \lambda_1 y_p^1 + \lambda_2 y_p^2 + \dots + \lambda_k y_p^k - \epsilon_p \geq y_p \\
 & \lambda_1 + \lambda_2 + \dots + \lambda_k = 1 \\
 & \lambda_i \geq 0 \quad \forall i \in \{1, \dots, p\} \\
 & \epsilon_i \geq 0 \quad \forall i \in \{1, \dots, p\}
 \end{aligned} \tag{1.4}$$

dans lequel les λ_i et ϵ_j sont les variables de décision. Si le problème 1.4 n'est pas réalisable, alors aucune combinaison linéaire des points y^1, y^2, \dots, y^k ne domine le point testé y , qui peut donc

être conservé dans D_{ex} . Par contre, si le problème admet une solution admissible avec $z > 0$, alors il existe une telle combinaison dominant y et ce point doit par conséquent être supprimé. Pour compléter ce test, chaque solution présente dans D_{ex} doit être testée avec chaque combinaison efficace.

Pour résumer, nous regroupons succinctement les composants choisis dans cette méthode.

- EBS : point idéal de la relaxation linéaire ;
- EBI : liste des solutions potentiellement non dominées ;
- Choix du nœud actif : parcours en profondeur ;
- Procédure de séparation : une variable est d’abord fixée à 0 puis à 1. L’ordre des variables est sélectionné statiquement et suit l’indice donné aux variables ;
- Obtention de nouveaux points : lorsqu’une feuille est atteinte, résolution du MOLP correspondant et mise à jour de l’ensemble non dominé.

1.4.3 Méthode en deux phases (2 et 3 objectifs)

La méthode en deux phases est un principe général de résolution pour les MOCO proposé par [Ulungu et Teghem, 1995] dans le cadre biobjectif. La méthode propose d’utiliser de façon intensive les méthodes efficaces et dédiées à la résolution de la version mono-objectif d’un problème donné. Ces algorithmes dédiés reposant sur la structure particulière des problèmes d’optimisation combinatoire, il n’est pas possible de modifier cette structure en ajoutant par exemple des contraintes. En particulier, on ne peut donc pas restreindre explicitement l’espace de recherche au moyen de contraintes bornant la valeur des objectifs.

La première phase consiste à déterminer un ensemble de solutions supportées (l’ensemble $X_{SE_{1m}}$ au minimum et X_{SE_M} au maximum). Ensuite, les autres solutions efficaces sont calculées durant la phase 2. Cette phase tire parti des solutions obtenues lors de la phase 1 pour déterminer des zones de recherche dans lesquelles des points non dominés peuvent être trouvés. Ces zones de recherche sont ensuite visitées au moyen d’une stratégie énumérative.

Les algorithmes utilisés pour les problèmes multiobjectifs diffèrent grandement de ceux mis en œuvre pour résoudre les problèmes biobjectifs comme le montre [Przybylski, 2006]. En effet, dans le cadre biobjectif, les stratégies s’appuient fortement sur l’ordre naturel des solutions efficaces, ce qui n’est plus possible au-delà de deux objectifs. Nous présentons donc séparément les méthodes dédiées aux programmes biobjectifs et celles s’appliquant dans le cadre multiobjectif.

Deux objectifs

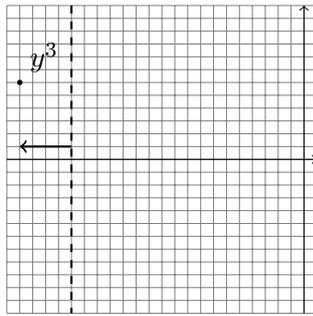
Pour les problèmes à deux objectifs, la phase 1 est en général accomplie grâce à l’algorithme d’énumération par dichotomie de [Aneja et Nair, 1979]. Cette méthode permet de calculer un ensemble $X_{SE_{1m}}$ et quelques solutions supplémentaires.

Lors de l’étape d’initialisation, les solutions lexicographiquement optimales x^1 et x^2 sont déterminées. Par la suite, on suppose que les points obtenus sont triés par première coordonnée croissante. À chaque itération de l’algorithme, on considère deux points consécutifs y^r et y^s et on résout le problème (P_λ) dont la pondération est $\lambda = (y_2^r - y_2^s, y_1^s - y_1^r)$. Un point y^t est obtenu lors de la résolution de (P_λ) et deux cas sont alors à considérer :

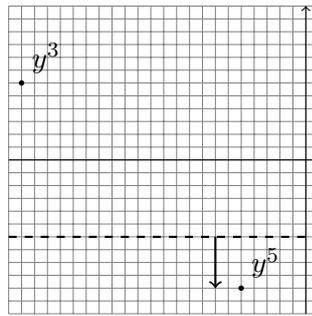
- y^t est un nouveau point tel que $\sum_{i=1}^2 \lambda_i y_i^t < \sum_{i=1}^2 \lambda_i y_i^r$. Dans ce cas, il est nécessaire de résoudre deux nouveaux problèmes en définissant λ grâce à y^r et y^t et grâce à y^t et y^s .

– y^t est tel que $\sum_{i=1}^2 \lambda_i y_i^t = \sum_{i=1}^2 \lambda_i y_i^r$ et la recherche s'arrête.

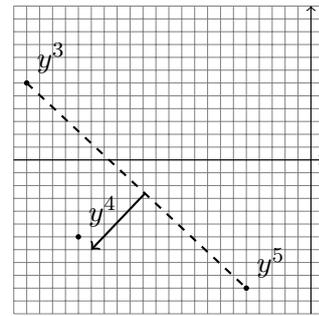
À la première étape, la recherche est amorcée grâce à y^1 et y^2 et il résulte de l'algorithme un ensemble X_{SE1_m} . Un exemple d'exécution de la première phase est donné par la figure 1.8.



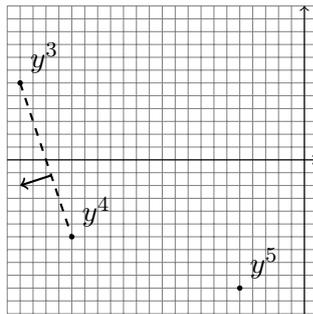
(a) Résolution avec la pondération $\lambda = (1, 0)$. Obtention de y^3 .



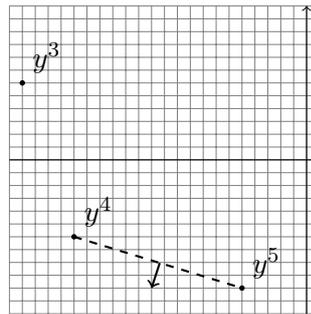
(b) Résolution avec la pondération $\lambda = (0, 1)$. Obtention de y^5 .



(c) y^3 et y^5 sont utilisés pour obtenir la pondération $\lambda = (16, 17)$. Obtention de y^4 .



(d) y^3 et y^4 sont utilisés pour obtenir la pondération $\lambda = (12, 4)$. Pas de nouveau point obtenu.



(e) y^4 et y^5 sont utilisés pour obtenir la pondération $\lambda = (4, 13)$. Pas de nouveau point obtenu.

Figure 1.8 – Exemple d'exécution de la première phase appliqué au problème **MOIPL1**.

On peut remarquer que cette méthode est fortement liée à l'ordre naturel des solutions non dominées et que sa généralisation à plus de deux objectifs n'est pas triviale.

Lors de la phase 2, les points issus de la première phase sont utilisés pour délimiter les zones dans lesquelles rechercher des solutions efficaces non supportées ou supportées non extrêmes. Pour cela, chaque couple (y^r, y^s) de points consécutifs est utilisé pour définir un triangle rectangle dont ils forment l'hypoténuse. L'autre sommet de ce triangle est (y_1^s, y_2^r) et est appelé *point nadir local* de y^r et y^s . Ensuite, chaque triangle est visité un à un à la recherche de nouvelles solutions. L'exploration de chaque triangle se fait au moyen d'un algorithme d'énumération, le Branch & Bound peut donc être utilisé. Une autre méthode d'énumération des k meilleurs solutions, le *ranking*, a connu un certain succès récemment pour accomplir cette tâche.

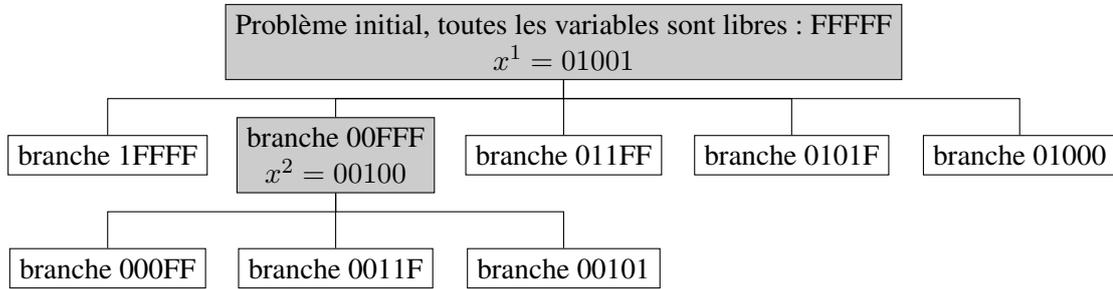


Figure 1.9 – Exemple de la méthode de partitionnement de [Lawler, 1972]. x^1 et x^2 sont connues et x^3 doit être déterminé parmi les branches ouvertes.

Ranking La méthode de ranking a pour but de calculer les k meilleures solutions d'un problème mono-objectif. L'objectif est donc d'obtenir k solutions x^1, x^2, \dots, x^k telles que

$$z(x^1) \leq z(x^2) \leq \dots \leq z(x^k) \leq z(x), \forall x \in X \setminus \{x^1, \dots, x^k\}$$

. Ces solutions sont obtenues au moyen d'un algorithme de partitionnement comme celui de [Lawler, 1972] appliqué aux problèmes combinatoires. Nous détaillons maintenant cette méthode.

La meilleure solution $x^1 = (x_1^1, x_1^1, \dots, x_n^1)$ est obtenue grâce à une résolution classique du problème. À partir de cette solution, $X \setminus \{x^1\}$ est partitionné en $\mathcal{P}_1 = \mathcal{P}(X \setminus \{x^1\}) = \{x \in X : x_1 \neq x_1^1\}, \{x \in X : x_1 = x_1^1, x_2 \neq x_2^1\}, \dots, \{x \in X : x_i = x_i^1, \forall i \in \{1, \dots, n-1\}, x_n \neq x_n^1\}$. Dans chacune des parties, le problème est résolu à nouveau et, puisque x^1 a été exclue, la meilleure des solutions obtenues sera x^2 , la deuxième meilleure solution. À partir de la partie $X_i \in \mathcal{P}_1$ de laquelle provient x^2 , on dérive $\mathcal{P}_2 = \mathcal{P}(X_i \setminus \{x^2\})$ et chaque partie de \mathcal{P}_2 est résolue séparément. Ainsi, la troisième meilleure solution x^3 sera la meilleure des solutions obtenues à partir de $(\mathcal{P}_1 \setminus X_i) \cup \mathcal{P}_2$. La figure 1.9 illustre une partie de ce processus.

Cette méthode a été utilisée avec succès dans la deuxième phase pour explorer les solutions non supportées. $z^\lambda(x) = \lambda_1 x_1 + \lambda_2 x_2$ y est utilisée pour que les solutions soient obtenues par leur distance croissante par rapport à l'hypoténuse du triangle $\Delta(y^r, y^s)$ considéré. Ainsi, les solutions obtenues qui ne sont pas dominées par l'ensemble des solutions déjà obtenues (l'ensemble bornant supérieurement) sont ajoutées à cet ensemble.

Remarque 1.4.2 – Toute solution x non dominée au moment de son obtention le sera définitivement. En effet, s'il existe une autre solution x' telle que $z(x') \leq z(x)$, alors nécessairement $z^\lambda(x') < z^\lambda(x)$, ce qui implique que x' a déjà été obtenue.

Néanmoins, si les premières solutions obtenues au moyen de la méthode de ranking sont susceptibles d'être intéressantes, il reste à déterminer quand arrêter l'exploration. En effet, on ne souhaite pas énumérer toutes les solutions du triangle et le nombre de solutions non supportées n'est pas connu à l'avance. Pour cela, la borne $\beta = \max\{z^\lambda(x) : x = (x_1^b, x_2^a) : x^a$ et x^b sont deux points consécutifs de $\Delta\}$ est utilisée. β est la borne sur la somme pondérée des objectifs telle que tout point ayant une valeur pondérée supérieure à β est dominée. Quand $z^\lambda(x^k) \geq \beta$, l'exploration s'arrête donc. Ce processus est illustré par la figure 1.10.

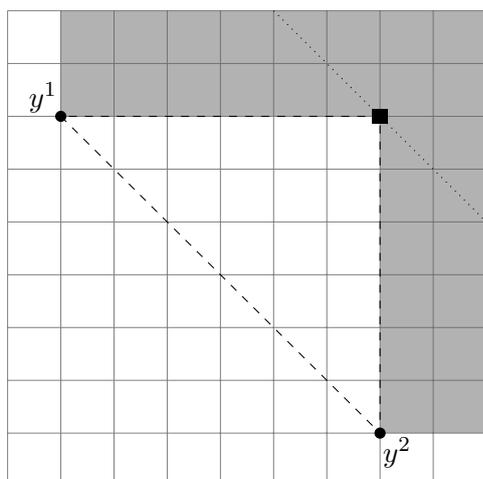
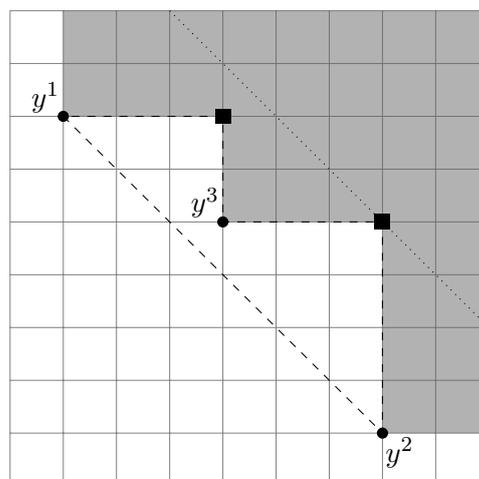
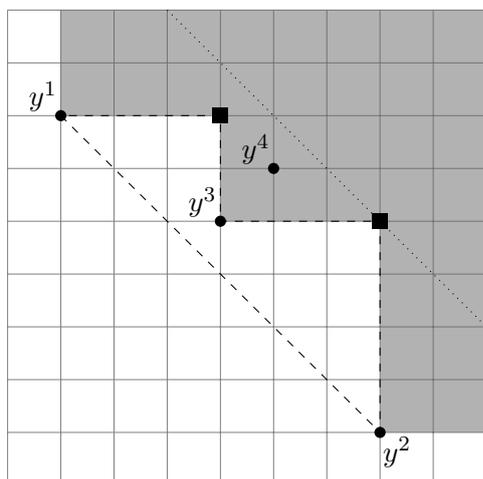
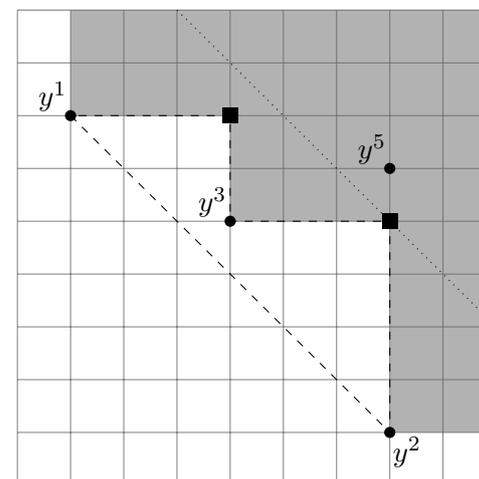
(a) y^1 et y^2 sont déjà connus. Initialisation de β .(b) y^3 n'est pas dominé. Mise à jour de β .(c) y^4 est dominé mais sous la borne. β ne change pas.(d) y^5 est dominé et au-dessus de β . Fin.

Figure 1.10 – Exemple d'exécution de la méthode du ranking dans un triangle $\Delta(y^1 y^2)$ dans le cadre de la seconde phase.

Remarque 1.4.3 – La borne β peut être affinée quand le problème traité est exclusivement discret. Puisque nous utiliserons cette borne dans un contexte en nombres mixtes, nous ne détaillerons pas ces améliorations.

Plus de deux objectifs

Comme nous venons de le voir, les algorithmes présentés pour mettre en œuvre la méthode en deux phases reposent sur l'ordre naturel des solutions efficaces dans le cadre biobjectif. Malheureusement, cet ordre n'existe plus quand plus de deux fonctions objectif sont considérées. Pour

résoudre des problèmes à trois objectifs et plus, il est donc nécessaire de généraliser les deux phases de cette méthode.

Première phase Contrairement au cas biobjectif, aucune méthode permettant d’obtenir l’ensemble des solutions supportées efficaces n’était proposée jusqu’en 2010 [Przybylski *et al.*, 2010, Özpeynirci et Köksalan, 2010]. Pourtant, il peut sembler intuitivement simple de passer de deux à trois objectifs (ou plus) quand il s’agit de calculer l’ensemble Y_{SN} . Il s’agit en effet d’étendre la dichotomie afin de calculer de nouvelles solutions au moyen d’un λ dérivé de points consécutifs connus. Néanmoins, perte de l’ordre naturel oblige, la notion de points non dominés consécutifs en dimension 3 n’est pas définie. De plus, en partant des points lexicographiquement optimaux, il n’est pas toujours possible d’obtenir tous les points supportés, comme le montre l’exemple 1.4.2.

Exemple 1.4.2 – Soit le programme linéaire triobjectif suivant :

$$\begin{array}{rcll}
 \min z_1 = & -x_1 & -4x_2 & \\
 \min z_2 = & -2x_1 & +2x_2 & \\
 \min z_3 = & & +x_2 & +x_3 \\
 \text{s.c.} & -x_1 & +2x_2 & \leq 9 \\
 & 6x_1 & -2x_2 & \leq 30 \\
 & x_1 & +2x_2 & \leq 12 \\
 & & & x_3 \leq 5 \\
 & 10x_1 & +6x_2 & -x_3 \leq 50 \\
 & x_1 & +x_2 & -7x_3 \leq 0 \\
 & & & x_i \in \mathbb{N} \quad i = 1, 2, 3
 \end{array} \tag{3O IPL}$$

Essayons d’appliquer la dichotomie à ce problème. 3O IPL admet trois solutions lexicographiquement optimales :

- $x^1 = (0, 0, 0)$ dont le point correspondant est $y^1 = (0, 0, 0)$ et minimise le troisième objectif ;
- $x^2 = (5, 0, 1)$ dont le point correspondant est $y^2 = (-5, -10, 1)$ et minimise le deuxième objectif ;
- $x^3 = (2, 5, 1)$ dont le point correspondant est $y^3 = (-22, 6, 6)$ et minimise le premier objectif.

Puisque nous n’avons que trois points, un seul choix est possible pour le poids initial. En calculant la normale au plan défini par y^1 , y^2 et y^3 , nous obtenons le poids $\lambda = (33, -4, 125)$ ou $\lambda = (-33, 4, -125)$. Dans les deux cas, λ a au moins une composante négative, ce qui fait que l’hypothèse $\lambda \in \mathbb{R}_{>}^p$ du théorème 1.2.1 n’est pas vérifiée. Par conséquent, la résolution du problème $(3O IPL)_\lambda$ ne garantit pas d’obtenir une solution efficace. Néanmoins, nous pouvons poursuivre la dichotomie en essayant les deux possibilités :

- en résolvant $(3O IPL)_{(33,-4,125)}$, nous obtenons à nouveau le point $(0, 0, 0) = y^1$;
- en résolvant $(3O IPL)_{(-33,4,-125)}$, nous obtenons le point $(0, 0, 5)$ qui est dominé.

À partir de ces deux poids, on ne peut donc pas obtenir de nouvelle solution supportée. Pourtant, il existe par exemple $x^4 = (4, 2, 2)$ dont le point correspondant est $y^4 = (-12, -4, 4)$ et qui peut être obtenu par la résolution du problème $3O IPL_{(4,4,1)}$.

Ainsi, il n'est pas simple de trouver les poids permettant de déterminer les solutions supportées efficaces, mais de récents travaux ont permis de passer outre ces difficultés. Nous abordons pour commencer la méthode proposée par [Przybylski *et al.*, 2010] permettant d'effectuer ce calcul.

[Przybylski *et al.*, 2010] propose de travailler dans l'espace des poids W^0 plutôt que l'espace des objectifs.

Définition 1.4.4 (Espace des poids [Przybylski *et al.*, 2010]). L'ensemble W^0 est défini par

$$W^0 = \left\{ \lambda \in \mathbb{R}^p : \lambda_p = 1 - \sum_{i=1}^{p-1} \lambda_i \right\}$$

Les auteurs montrent en effet qu'il y a correspondance entre les points non dominés extrêmes et les polytopes convexes de W^0 . Ils proposent alors une méthode calculant la décomposition de l'espace de poids et, simultanément, les points non dominés extrêmes. Les nouveaux poids à considérer dans cette méthode étant obtenus en décomposant l'espace de poids, tous les poids considérés ont des composantes strictement positives.

[Özpeynirci et Köksalan, 2010] considèrent également une décomposition de l'espace de poids pour montrer que leur méthode est correcte, mais celle-ci est basée sur l'identification des facettes non dominées de l'enveloppe convexe des points non dominés extrêmes. Puisque les faces non dominées de dimension maximale de ce polyèdre ne sont pas nécessairement des facettes, les auteurs introduisent des points artificiels dans le but d'en modifier la structure. Dans cette méthode, des poids ayant des composantes négatives peuvent être considérés mais sont immédiatement écartés.

[Schandl *et al.*, 2002] ont quant à eux proposé des algorithmes d'approximation dédiés aux problèmes multiobjectifs convexes continus. Même s'il ne s'agit pas de leur but premier, ces algorithmes peuvent être utilisés pour générer tous les points non dominés extrêmes d'un MOILP. Les poids à considérer sont obtenus à partir de l'enveloppe convexe des points déjà trouvés. Les poids ayant des composantes négatives doivent être considérés et ne sont pas ignorés. Des points non dominés sont ainsi générés et modifient l'enveloppe convexe calculée, ce qui permet de déduire de nouveaux poids et de poursuivre l'énumération.

Finalement, [Przybylski *et al.*, 2012] ont proposé une méthode s'inspirant des précédentes qui permet de ne considérer que les poids ayant des composantes strictement positives. Celle-ci est synthétisée dans l'algorithme 1.1.

Sortie Ensemble S

- 1: Initialiser S avec les points non dominés extrêmes des p problèmes à $p - 1$ objectifs
- 2: **répéter**
- 3: Mettre à jour $\text{conv } S$
- 4: Choisir une facette non explorée ayant un poids $\lambda \in \mathbb{R}_{>}^p$
- 5: Résoudre le problème de somme pondérée
- 6: **jusqu'à** Toutes les facettes non dominées de $\text{conv } S$ soient explorées

Algorithme 1.1: Algorithme computeYSN

Deuxième phase Les méthodes présentées pour la deuxième phase biobjectif ne nécessitent pas de modification particulière dans le cas multiobjectif. En effet, la procédure de Branch & Bound

peut fonctionner telle quelle et la méthode du ranking, puisqu'elle ne résout que des problèmes mono-objectifs (issus de pondérations des objectifs), n'est pas affectée non plus.

En revanche, pour que ces méthodes puissent être utilisées efficacement, il est nécessaire de connaître la zone de recherche à explorer. Malheureusement, la détermination de ces zones à visiter pose beaucoup plus de problèmes. La notion de solutions consécutives n'a effectivement plus lieu d'être dans le cadre multiobjectif et on ne peut donc plus définir de triangles à visiter aussi aisément.

Pour pallier ce problème, [Przybylski *et al.*, 2010] propose de remplacer la notion de point nadir local par celle de *point délimitant*.

Définition 1.4.5 ([Przybylski *et al.*, 2010]). Soit U un ensemble de points ne se dominant pas les uns les autres. d est un point délimitant pour U ssi

- (i) Pour tout $\epsilon \in \mathbb{R}_{>0}^p$, $d - \epsilon$ n'est pas dominé par un point de U
- (ii) Il n'existe pas de point v vérifiant la condition (i) avec $d \leq v$

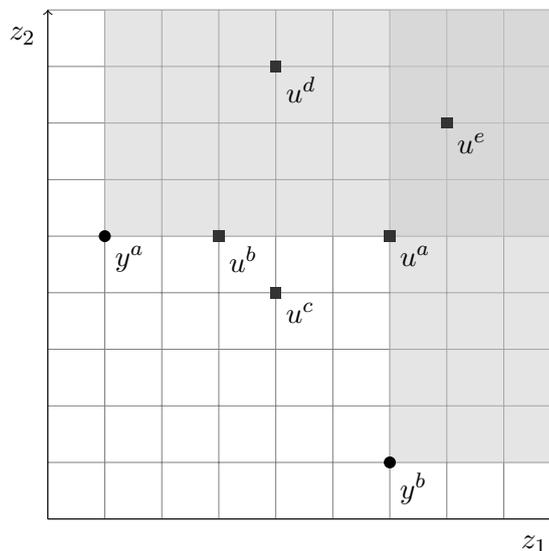


Figure 1.11 – Illustration des points délimitant

Exemple 1.4.3 – Quelques exemples de potentiels points délimitants sont donnés dans la figure 1.11. L'ensemble U est ici constitué de deux points y^a et y^b . Parmi les points candidats, on note que :

- u^d et u^e sont strictement dominés et ne vérifient donc pas la propriété 1.4.5-(i) ;
- u^a , u^b et u^c vérifient la propriété 1.4.5-(i) ;
- u^b et u^c dominent u^a qui vérifie la propriété 1.4.5-(i), ils ne vérifient donc pas la propriété 1.4.5-(ii) ;
- u^a vérifie la propriété 1.4.5-(ii).

u^a est donc un point délimitant pour l'ensemble $U = \{y^a, y^b\}$.

L'ensemble des points délimitants est donc une généralisation de l'ensemble des points nadir locaux qui était utilisé dans les triangles en optimisation combinatoire biobjectif. Ils déterminent la zone dans laquelle peuvent se trouver des points non dominés et qu'il faut donc explorer.

Quand un nouveau point non dominé y est trouvé, l'ensemble des points délimitants doit être mis à jour en conséquence. Pour cela, il est nécessaire de déterminer tous les points délimitants u strictement dominés par y . Quand un tel point existe, celui-ci est supprimé et remplacé par de nouveaux points permettant de délimiter la même région que précédemment à l'exception de la partie dominée par y . La figure 1.12 illustre cette situation. Dans cette figure, le point u est remplacé par trois nouveaux points délimitants u^1 , u^2 et u^3 . Il faut cependant noter que ces trois points ne seront ajoutés que s'ils ne dominent pas d'autres points délimitants.

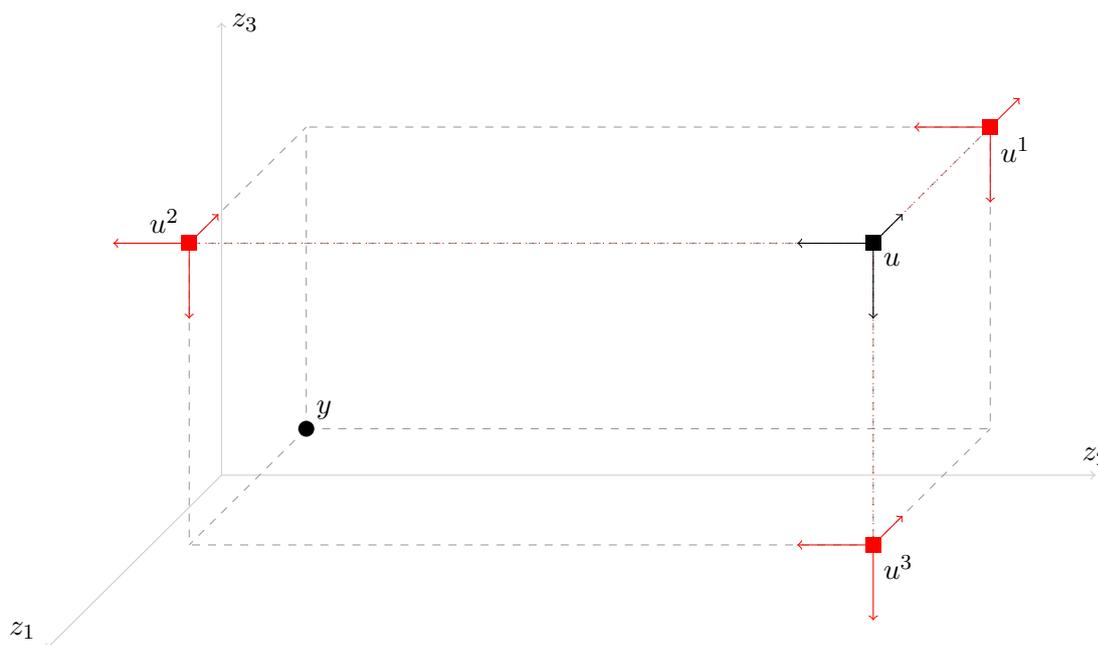


Figure 1.12 – Mise à jour d'un point délimitant dans le contexte discret.

La procédure de mise à jour des points délimitants est donnée dans l'algorithme 1.2.

Entrée Ensemble de points réalisables U , point nadir y^N , point idéal y^I

Sortie Ensemble $D(U)$

```

1: // Initialisation
2: // À chaque étape,  $Q$  est un sous ensemble de  $U$ 
3:  $Q \leftarrow \emptyset$ 
4:  $D(Q) \leftarrow \{y^N\}$ 
5: // Détermination de  $D(U)$  en considérant les points de  $U$  un
   par un
6: pour chaque  $y \in U$  faire
7:   updateUBS( $y \downarrow, D(Q) \downarrow, D(Q \cup \{y\}) \uparrow$ )
8:    $Q \leftarrow Q \cup \{y\}$ 
9: fin pour

```

Algorithme 1.2: Algorithme computeUBS [Przybylski *et al.*, 2010]

Entrée Point réalisable $y \in U \setminus Q$, ensemble $D(Q)$
Sortie Ensemble $D(Q \cup \{y\})$

```

1: //  $W$  est l'ensemble des points de  $D(Q) \setminus D(Q \cup \{y\})$ 
2: //  $N$  est l'ensemble des points de  $D(Q \cup \{y\}) \setminus D(Q)$ 
3:  $W \leftarrow \emptyset$ 
4:  $N \leftarrow \emptyset$ 
5: // Comparaison de  $y$  avec chaque point  $u \in D(Q)$ 
6: pour chaque  $u \in D(Q)$  faire
7:   si  $y < u$  alors
8:      $W \leftarrow W \cup \{u\}$ 
9:     // Par la suite,  $u^i = (u_1, \dots, u_{i-1}, y_i, u_{i+1}, \dots, u_p)$ 
10:    pour chaque  $i \in \{1, \dots, p\}$  faire
11:      si  $(u_i = y_i^l)$  ou (il existe  $v \neq u \in D(Q)$  tel que  $u^i \leq v$ ) alors
12:        // Rien à faire ici
13:      sinon
14:         $N \leftarrow N \cup \{u^i\}$ 
15:      fin si
16:    fin pour
17:  fin si
18: fin pour

```

Algorithme 1.3: Algorithme updateUBS [Przybylski *et al.*, 2010]

Après avoir déterminé les points délimitants lors de la première phase, il reste à explorer les régions qu'ils délimitent, de la même façon qu'il faut visiter les triangles dans le cas biobjectif. Puisque plusieurs points délimitants peuvent définir des régions dont l'intersection n'est pas nulle, il n'est donc plus possible de facilement explorer séparément chacune des partitions de l'espace.

Puisque plusieurs points délimitants peuvent définir des zones de recherches communes, [Przybylski *et al.*, 2010] propose d'organiser l'exploration de ces zones comme suit. Parmi les hyperplans générés par les faces de $(\text{conv } Y_{SN})_N$ (obtenu à l'issue de la première phase), l'hyperplan $h(u)$ le plus proche du point u est déterminé, pour chaque point $u \in F$ où F est l'ensemble des points qui délimitent la zone qu'il reste à explorer. Les $h(u)$ forment H_p , l'ensemble des hyperplans les plus proches d'au moins un point délimitant. Cet ensemble n'est pas nécessairement égal à l'ensemble H de tous les hyperplans. De plus, on associe à chaque hyperplan h l'ensemble $U(h) := \{u \in F : h(u) = h\}$ et $Val(h) := \max_{u \in U(h)} \text{dist}(u, h)$.

À chaque étape, on recherche h^* tel que $Val(h^*) = \min_{h \in H_p} Val(h)$ afin d'explorer $\bigcup_{u \in U(h^*)} (u - \mathbb{R}_{\geq}^p)$. Cette exploration se fait en utilisant la bande définie par l'hyperplan h^* et l'hyperplan qui y est parallèle et passe par le point de $U(h^*)$ le plus éloigné de h^* . Lors de cette exploration, les nouveaux points trouvés sont utilisés pour mettre à jour l'ensemble F et ce processus se répète jusqu'à obtenir $F = \emptyset$.

1.5 Discussion

Ce chapitre a présenté le domaine de l'optimisation multiobjectif et plus particulièrement les programmes linéaires multiobjectifs en variables continues, discrètes ou mixtes. Des méthodes de

résolution habituellement utilisées pour ces problèmes ont également fait l'objet d'une présentation.

Les problèmes en variables continues peuvent être résolus au moyen d'extensions multiobjectifs de l'algorithme du simplexe ou bien d'algorithmes basés dans l'espace des objectifs, comme la méthode d'approximation externe de Benson. Ces méthodes permettent de calculer l'ensemble des points non dominés extrêmes à partir duquel tout l'ensemble non dominé peut être déterminé.

Dans le cas des problèmes en variables entières, et plus particulièrement celui des problèmes combinatoires, de nombreuses méthodes existent également. Parmi elles, nous avons détaillé l'algorithme de Branch & Bound, qui se base sur une recherche arborescente, et la méthode en deux phases, qui décompose l'espace de recherche.

En revanche, très peu de publications concernent les problèmes en variables mixtes. Nous nous sommes donc intéressés à [[Mavrotas et Diakoulaki, 1998](#), [Mavrotas et Diakoulaki, 2005](#)] qui proposent un algorithme de Branch & Bound adapté au contexte mixte. Néanmoins, des erreurs existent dans le premier article qui peuvent mener à l'obtention de solutions dominées et à la perte de solutions efficaces. Le second article s'attache à les corriger. Dans le prochain chapitre, nous commençons par montrer que des erreurs subsistent [[Mavrotas et Diakoulaki, 2005](#)] à cause d'une mauvaise représentation des solutions efficaces. Nous proposons donc de corriger ce problème grâce à une représentation appropriée de ces solutions puis introduirons différentes améliorations sur les composants de l'algorithme.

Algorithme de Branch & Bound pour les problèmes biobjectifs

Dans ce chapitre, nous revenons sur la méthode proposée par [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005]. Comme nous l'avons vu, cet algorithme de séparation et évaluation adapté pour traiter les problèmes multiobjectifs en variables mixtes a fait l'objet d'une correction dans un second article qui permet d'éliminer certaines solutions dominées. Nous allons maintenant détailler ce problème et montrer que la correction proposée n'est pas suffisante. Puis, en nous appuyant sur une représentation correcte de l'ensemble des points non dominés, nous proposerons une procédure de mise à jour garantissant un filtrage par dominance correct. Par la suite, nous proposerons des améliorations portant sur les différents composants de l'algorithme de Branch & Bound. Enfin, des résultats expérimentaux nous permettront de juger de la pertinence de ces améliorations. Les éléments principaux du présent chapitre sont publiés dans [Vincent et al., 2013d].

| | | |
|------------|---|-----------|
| 2.1 | Défauts de l'algorithme de Mavrotas et Diakoulaki | 41 |
| 2.2 | Caractérisation correcte des solutions efficaces | 42 |
| 2.3 | Méthode de mise à jour des ensembles non dominés | 45 |
| 2.3.1 | Comparaison d'un point avec une arête dont l'intérieur relatif est non vide | 46 |
| 2.3.2 | Comparaison de deux arêtes dont l'intérieur relatif est non vide | 48 |
| 2.3.3 | Formulation de la procédure de mise à jour | 51 |
| 2.4 | Améliorations de l'algorithme de Branch & Bound | 53 |
| 2.4.1 | Ensemble bornant supérieurement | 53 |
| 2.4.2 | Ensemble bornant inférieurement | 54 |
| 2.4.3 | Procédure de branchement | 55 |
| 2.5 | Résultats expérimentaux | 57 |
| 2.5.1 | Comparaison des ensembles bornant inférieurement | 58 |
| 2.5.2 | Apports de la procédure de branchement proposée | 61 |
| 2.5.3 | Apports de l'initialisation de l'ensemble bornant supérieurement | 61 |
| 2.5.4 | Tests supplémentaires sur de plus grandes instances | 62 |
| 2.6 | Conclusion | 63 |

2.1 Défauts de l’algorithme de Mavrotas et Diakoulaki

Comme nous l’avons vu dans le chapitre précédent (section 1.4.2), la méthode de résolution proposée par [Mavrotas et Diakoulaki, 1998] a fait l’objet d’une seconde publication [Mavrotas et Diakoulaki, 2005] dans laquelle était proposé un test de filtrage supplémentaire en plus d’améliorations générales des performances. Ce test, effectué à la toute fin de la procédure de recherche, permet d’éliminer, parmi les solutions contenues dans l’ensemble des points non dominés noté D_{ex} , les points qui sont dominés par des combinaisons efficaces non considérées jusque là.

Hélas, malgré ce filtrage supplémentaire, il peut rester des points dominés dans l’ensemble résultant. Pire encore, des points non dominés sont ignorés à tort. Ce problème est dû au fait que le test considérant les combinaisons convexes de points extrêmes (c’est-à-dire les faces des MOLP) a lieu après l’énumération. Il en résulte que des informations peuvent être perdues, notamment concernant des faces dont des points extrêmes sont dominés. Pour illustrer ce problème, considérons l’exemple suivant.

Exemple 2.1.1 – La figure 2.1 représente (en noir) y^1 , y^2 , y^3 et y^4 , quatre points extrêmes (de MOLP) conservés jusqu’à la fin de l’énumération, ils sont donc considérés non dominés. En gris figurent deux points t et u filtrés par dominance lors de l’énumération. Reposant uniquement sur la prise en compte de points extrêmes des MOLP, la procédure de Mavrotas et Diakoulaki supprime à juste titre t et u puisque ces points sont effectivement dominés par y^1 et y^4 respectivement.

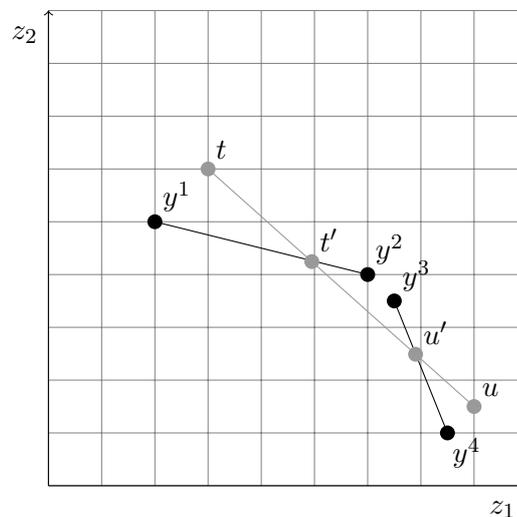


Figure 2.1 – y^2 et y^3 devraient être éliminés puisqu’ils sont dominés par des points de $[tu]$.

En revanche, ces deux points supprimés avaient été obtenus dans la même feuille de l’arbre de recherche. Ils ont donc la même combinaison de valeurs pour les variables binaires et appartiennent au même MOLP. Par conséquent, t et u forment une arête réalisable $[tu]$ dont une sous arête $[t'u']$ n’est pas dominée.

Puisque t et u ne font plus partie de l’ensemble des points non dominés (ce qui est correct, rappelons-le), l’information quant à l’arête $[tu]$ est perdue. En particulier, il n’y a donc plus de moyen de déterminer que $[t'u']$ est composée de points non dominés. De plus, puisque $[t'u']$ n’est

pas prise en compte, y^2 et y^3 sont considérés non dominés à tort alors que certains points de $[t'u']$ les dominent.

En conclusion, cette perte d'information conduit à ignorer des points non dominés (les points de l'arête $[t'u']$) et à maintenir des points dominés (les points des arêtes $(t'y^2]$ et $[y^3u')$).

Pour remédier à ce problème, il pourrait être envisagé de conserver tous les points extrêmes des MOLP résolus, même si ces points ont été reconnus dominés. Ainsi, toutes les informations nécessaires au test final seraient disponibles et toutes les solutions dominées pourraient être filtrées.

Cependant, recourir à une telle solution serait beaucoup trop coûteux en mémoire (le nombre de points extrêmes multiplié par le nombre de MOLP) et le problème des points non dominés ignorés persisterait. En effet, déterminer que les points y^2 et y^3 de l'exemple 2.1.1 sont dominés serait possible, mais rien ne permettrait de caractériser l'arête $[t'u']$ à conserver.

Ces constatations nous ont incités à proposer une nouvelle caractérisation des solutions efficaces. Dans le reste de ce chapitre, nos travaux se placent dans le cadre biobjectif.

2.2 Caractérisation correcte des solutions efficaces

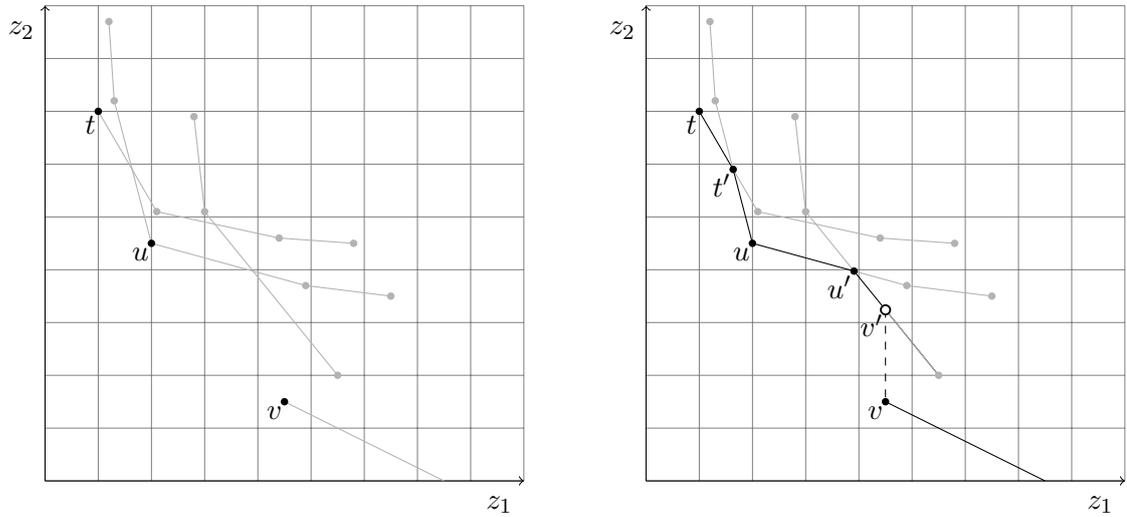
Le besoin d'une caractérisation correcte des solutions efficaces provient du fait qu'il n'est pas raisonnable d'envisager de conserver toutes les solutions extrêmes énumérées au cours de la résolution. Il est donc nécessaire de garder à jour l'état *exact* et *complet* de l'ensemble des points potentiellement non dominés, c'est-à-dire les points extrêmes de MOLP mais aussi les arêtes qui ne sont pas dominées.

Remarque 2.2.1 – Nous notons $[y^1y^2]$ l'arête *fermée potentiellement non dominée* dont les extrémités sont y^1 et y^2 , c'est-à-dire l'ensemble $\text{conv}(y^1, y^2)$ où y^1 et y^2 ne se dominent pas l'un l'autre. De même que pour les notations des intervalles dans \mathbb{R} , il y a également des arêtes potentiellement non dominées *semi-ouvertes* (ouvertes à gauche ou à droite) et *ouvertes* respectivement notées $]y^1y^2]$, $[y^1y^2[$ et $]y^1y^2[$. Enfin, quand $y^1 = y^2$, l'ensemble $[y^1y^2]$ est réduit à un singleton.

Dans la suite de ce manuscrit, nous appellerons Z l'ensemble potentiellement non dominé. Ainsi, à la fin de l'algorithme, nous devrions donc aboutir à $Z = Y_N$. Dans l'exemple 2.1.1, Z est obtenu par l'union des arêtes fermées $[y^1t']$, $[t'u']$ et $[u'y^4]$. On constate que t' et u' sont nécessaires à la description correcte et complète de Z . Ces deux points sont pourtant des points *intermédiaires*, c'est-à-dire qu'ils ne sont pas les points extrêmes d'un MOLP.

Contrairement à l'ensemble non dominé d'un MOLP biobjectif, les courbes de Z ou Y_N ne sont généralement pas convexes, mais elles restent décroissantes. La figure 2.2(b) est un autre exemple montrant que l'ensemble Z peut être non convexe et qu'il peut contenir des arêtes semi-ouvertes (arête $[u'v'[,$ dans laquelle le point v' est faiblement non dominé). Dans ces deux exemples, il n'est pas possible de déterminer la représentation exacte des points potentiellement non dominés à partir de l'ensemble ne contenant que des points extrêmes de MOLP, même lorsque ces points sont non dominés (figures 2.2(a) et 2.2(b)).

Notre but est donc de caractériser l'ensemble potentiellement non dominé Z tout en utilisant un ensemble discret de points. Cet ensemble est mis à jour au moyen d'un test de dominance à chaque fois qu'une feuille de l'arbre est visitée. Évidemment, tous les points et arêtes ajoutés à Z doivent être potentiellement non dominés. C'est pourquoi nous omettons le terme *potentiellement non dominé* par la suite, afin d'alléger les notations.



(a) Représentation de Z avec des points non dominés extrêmes.

(b) Représentation de Z avec des points non dominés extrêmes et non extrêmes.

Figure 2.2 – Comparaison entre les deux représentations de Z .

Nous allons montrer que Z peut être décrit par un ensemble d'arêtes. Pour cela, nous allons d'abord montrer que la partie non dominée d'une arête est nécessairement un ensemble d'arêtes. Quand ce n'est pas précisé, nous considérons que les arêtes peuvent être de n'importe quel type présenté dans la remarque 2.2.1. Pour un ensemble $I \subseteq \mathbb{R}^2$, nous notons \bar{I} la fermeture de I .

Lemme 2.2.1. *Soit $I \subseteq \mathbb{R}^2$ une arête telle que $\bar{I} = [y^1 y^2]$ et soit $J \subseteq I$ un ensemble connexe non vide, alors J est une arête.*

Démonstration.

Si $y^1 = y^2$, l'affirmation est évidente. Nous supposons donc que y^1 et y^2 sont distincts et nous considérons $f_{\bar{I}}$, la projection de \bar{I} sur la première composante, c'est-à-dire la fonction définie par

$$\begin{aligned} f_{\bar{I}}: [y^1 y^2] \subseteq \mathbb{R}^2 &\rightarrow [y_1^1, y_1^2] \subseteq \mathbb{R} \\ y = (y_1, y_2) &\mapsto y_1. \end{aligned}$$

Il est clair que $f_{\bar{I}}$ est une bijection. Pour tout ensemble connexe $J \subseteq I$, $f_{\bar{I}}(J)$ est l'image d'un ensemble connexe par une fonction continue, donc $f_{\bar{I}}(J)$ est connexe. Puisque tous les ensembles connexes de \mathbb{R} sont des intervalles, $f_{\bar{I}}(J)$ est un intervalle. On en déduit donc que J est une arête.

□

Le corollaire 2.2.2 est une application directe du lemme 2.2.1.

Corollaire 2.2.2. *Si $I \subseteq \mathbb{R}^2$ est un ensemble connexe non vide tel que $\bar{I} = [y^1 y^2]$, alors I est une arête.*

Théorème 2.2.3. *Soient I_1 et I_2 des arêtes et posons $I := I_1 \cap (I_2 + \mathbb{R}_{\geq}^2)$, le sous-ensemble de I_1 dominé par des points de I_2 . Alors I est soit une arête, soit l'ensemble vide. De plus, $I_1 \setminus I$, le*

sous-ensemble de I_1 n'étant pas dominé par I_2 , peut être caractérisé par

$$I_1 \setminus I = \bigcup_{i=1}^k I_{1_i}$$

où $k \in \{0, 1, 2\}$ et les I_{1_i} sont des arêtes.

Démonstration.

L'affirmation est triviale pour $I = \emptyset$, donc nous supposons dans la suite que $I \neq \emptyset$.

Le cas particulier dans lequel $I_1 = \{y^1\}$ est un singleton implique que $I = I_1$, ce qui rend à nouveau l'affirmation triviale.

Il nous reste donc à considérer le cas dans lequel $\bar{I}_1 = [y^1 y^2]$ avec $y^1 \neq y^2$.

Si $I = \{\tilde{y}\}$ est un singleton, alors $\bar{I}_1 \setminus I = \bar{I}_1 \setminus \{\tilde{y}\} = [y^1 \tilde{y} \cup \tilde{y} y^2]$, et l'affirmation découle du corollaire 2.2.2. Dans ce cas, on peut remarquer qu'on a nécessairement $\tilde{y} \in \{y^1, y^2\}$, car I_2 ne peut pas dominer un point isolé appartenant à l'intérieur relatif d' I_1 . Une des arêtes semi-ouvertes est donc en fait vide.

Supposons maintenant que I contienne au moins deux points distincts. I est alors un ensemble convexe (donc connexe) puisqu'il s'agit de l'intersection des ensembles convexes I_1 et $(I_2 + \mathbb{R}_{\geq}^2)$. En particulier, le lemme 2.2.1 implique que I est également une arête dont la fermeture peut être notée $\bar{I} = [e^1 e^2]$. On peut donc écrire $\bar{I}_1 \setminus \bar{I}$ sous la forme

$$\bar{I}_1 \setminus \bar{I} = [y^1 e^1 \cup e^2 y^2].$$

En fonction de l'appartenance ou non de e^1 et e^2 à I , de l'appartenance ou non de y^1 et y^2 à I_1 et de la coïncidence ou non de y^1 et e^1 ou y^2 et e^2 , plusieurs résultats sont possibles pour $I_1 \setminus I$. Il est facile de voir que

$$I_1 \setminus I = I_{1_1} \cup I_{1_2}$$

où I_{1_1} et I_{1_2} sont des arêtes (fermées, semi-ouvertes, ouvertes ou réduites à un point) ou des ensembles vides. Ceci complète la preuve.

□

Le théorème 2.2.3 nous donne le principe d'une procédure de mise à jour correcte de l'ensemble potentiellement non dominé. Étant données une arête I_1 de E , l'ensemble non dominé du MOLP d'une feuille, et une arête I_2 de Z , le théorème 2.2.3 peut être utilisé à deux reprises pour supprimer la partie de I_1 dominée par des points de I_2 et la partie de I_2 dominée par des points de I_1 (lignes 3 à 6 de l'algorithme 2.1). En répétant ce test de dominance pour toutes les paires d'arêtes (I_1, I_2) avec $I_1 \in E$ et $I_2 \in Z$, nous obtenons la procédure de mise à jour décrite par l'algorithme 2.1.

Il n'est pas immédiat d'appliquer en pratique le test de dominance des lignes 3 à 6 de l'algorithme. Néanmoins, cette procédure de mise à jour est suffisante pour caractériser de l'ensemble Z .

Théorème 2.2.4. *À chaque étape de l'algorithme, Z est composé d'arêtes, qu'elles soient fermées, semi-ouvertes, ouvertes ou réduites à un point.*

Démonstration.

Entrée Ensemble d'arêtes E donné par un MOLP, ensemble potentiellement non dominé Z

Sortie Ensemble potentiellement non dominé Z mis à jour

```

1: pour chaque  $I_1 \in E$  faire
2:   pour chaque  $I_2 \in Z$  faire
3:      $I \leftarrow I_1 \cap (I_2 + \mathbb{R}_{\geq}^2)$ 
4:      $I_1 \leftarrow I_1 \setminus I$ 
5:      $I \leftarrow I_2 \cap (I_1 + \mathbb{R}_{\geq}^2)$ 
6:      $I_2 \leftarrow I_2 \setminus I$ 
7:   fin pour
8: fin pour
9:  $Z \leftarrow (\bigcup_{I_1 \in E} I_1) \cup (\bigcup_{I_2 \in Z} I_2)$ 

```

Algorithme 2.1: procédure `metaUpdate`

Z est évidemment vide avant le début de l'énumération puis, après la première étape, contient l'ensemble E des arêtes fermées obtenues à partir du MOLP considéré dans la première feuille de l'arbre de recherche. Par la suite, l'algorithme 2.1 est utilisé pour mettre Z à jour dès qu'une nouvelle feuille de l'arbre de recherche est visitée. L'application répétée du théorème 2.2.3 implique qu'une union finie d'arêtes (pouvant être fermées, semi-ouvertes, ouvertes ou réduites à un point) est obtenue à la fin de la mise à jour.

□

2.3 Méthode de mise à jour des ensembles non dominés

Dans cette section, nous détaillons comment la procédure de mise à jour venant d'être présentée peut être appliquée en pratique. Pour cela, les étapes élémentaires des lignes 3 à 6 de l'algorithme 2.1, consistant à comparer deux arêtes, doivent être détaillées. Nous distinguons donc trois cas possibles :

- chaque arête est réduite à un point isolé ;
- une arête est réduite à un point tandis que l'intérieur relatif de l'autre n'est pas vide ;
- les deux arêtes ont un intérieur relatif non vide.

Le premier cas correspond à la comparaison classique de points dans l'espace des objectifs (cf. figure 2.5). Les deuxième et troisième cas nécessitent en revanche une description plus détaillée. En particulier, nous allons présenter les modifications requises pour mener à bien cette mise à jour.

Grâce au théorème 2.2.4, on peut remarquer que l'ensemble Z peut être défini comme un ensemble discret, contenant les points qui forment les extrémités de ses arêtes, ainsi qu'une information supplémentaire précisant comment ces points sont connectés. Cet ensemble de points (qui sont des points non dominés extrêmes ou intermédiaires de certains MOLP) est noté Z_{ex} .

Remarque 2.3.1 – Z_{ex} est ordonné de façon à ce que ses points soient triés par valeur croissante de leur première composante, puis par valeur décroissante de leur seconde composante. L'ordre concernant la seconde composante se justifie par le fait que Z_{ex} peut contenir des extrémités dominées, comme c'est le cas de v' dans la figure 2.2(b). En effet, puisque nous souhaitons décrire l'arête $[u'v']$ puis l'arête commençant en v , il nous faut placer v' avant v dans la liste des points de Z_{ex} .

Pour chacun de ces points, il est donc nécessaire de savoir s'il est connecté à son prédécesseur et à son successeur dans Z_{ex} afin de pouvoir construire Z à partir de Z_{ex} . Puisque les arêtes peuvent être fermées, semi-ouvertes ou ouvertes, il est également nécessaire d'indiquer la nature de chaque point de Z_{ex} , c'est-à-dire s'il est dominé ou non. Par exemple, une arête $I = [y^1 y^2[$ est représentée par deux points consécutifs dans Z_{ex} , y^1 et y^2 , accompagnés de l'indication que ces deux points sont connectés et que y^2 est dominé. À la suite de chaque résultat de la section, une courte explication sera donnée sur l'implémentation de l'algorithme.

Remarque 2.3.2 – Il est suffisant de ne considérer que les arêtes fermées dans les tests de dominance à venir. Cela vient du fait que, puisqu'elles font partie de l'ensemble dominé d'un MOLP, toutes les arêtes considérées sont à la base fermées. Les arêtes ouvertes ou semi-ouvertes ne sont que le résultat d'un test de dominance. Ainsi, toute extrémité exclue d'une arête reste un point réalisable pour le problème et peut donc toujours être utilisée pour montrer qu'une arête ou un point est dominé(e).

2.3.1 Comparaison d'un point avec une arête dont l'intérieur relatif est non vide

Lorsqu'un point est comparé à un autre point ou à une arête, il peut être dominé ou non. En revanche, une arête peut être partiellement dominée par un point, comme nous l'avons déjà vu dans la preuve du théorème 2.2.3. Les sous arêtes qui ne sont pas dominées sont maintenant décrites. Tous les cas possibles sont pris en compte par la proposition 2.3.3.

Lemme 2.3.1. *Si les deux extrémités d'une arête sont dominés par le même point, alors toute l'arête est dominée.*

Démonstration.

Soit I l'arête donnée et considérons sa fermeture $\bar{I} = [y^1 y^2]$. Soit y le point dominant à la fois y^1 et y^2 , les deux extrémités de I . Nous avons en particulier $y_1 \leq y_1^1$ et $y_2 \leq y_2^2$. Pour tout point \hat{y} de l'intérieur de I , nous avons nécessairement $y_1^1 < \hat{y}_1$ et $y_2^2 < \hat{y}_2$ et donc $y_1 < \hat{y}_1$ et $y_2 < \hat{y}_2$. Ceci conclut la preuve.

□

Remarque 2.3.3 – Pour un point donné $y \in \mathbb{R}^2$ et une arête fermée non vide $I = [z^1 z^2]$, nous notons $y^{\leftrightarrow} := \text{proj}_1^I(y)$ et $y^\uparrow := \text{proj}_2^I(y)$ les projections, si elles existent, de y sur I respectivement le long de l'axe des abscisses et des ordonnées. On peut par exemple remarquer que y^\uparrow n'existe que si $z_1^1 \leq y_1 \leq z_1^2$ (cf. figures 2.3 et 2.4). De même, y^{\leftrightarrow} n'existe que si $z_2^1 \geq y_2 \geq z_2^2$.

Proposition 2.3.2. *Soient y un point isolé et $I = [z^1 z^2]$ une arête telle que $z^1 \neq z^2$. Il n'y a aucune relation de dominance entre y et n'importe quel point de I dans les cas suivants :*

- (i) $y \in I$;
- (ii) $y_1 < z_1^1$ et $y_2 > z_2^1$ (région A_1 de la figure 2.6) ;
- (iii) $y_1 > z_1^2$ et $y_2 < z_2^2$ (région A_2).

Démonstration.

L'affirmation est triviale quand $y \in I$. Dans les autres cas, pour tout point $\bar{z} \in I$, nous avons soit $y_1 < \bar{z}_1$ et $y_2 > \bar{z}_2$, soit $y_1 > \bar{z}_1$ et $y_2 < \bar{z}_2$. Ainsi, il ne peut y avoir de relation de dominance entre y et un point de I .

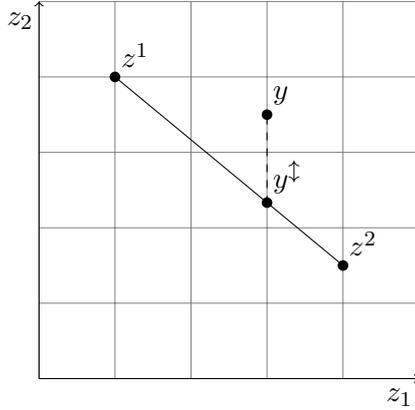


Figure 2.3 – y est dominé par sa projection y^\uparrow qui appartient à $[z^1 z^2]$.

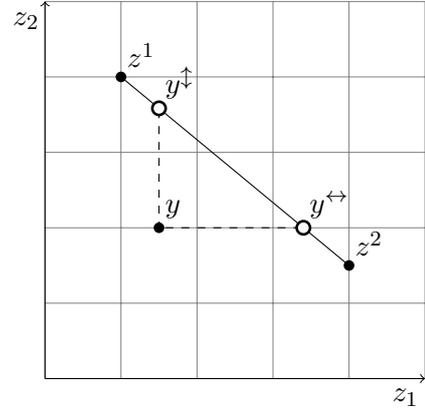


Figure 2.4 – y domine sa projection y^\uparrow et n'est donc dominé par aucun point de $[z^1 z^2]$.

□

Proposition 2.3.3. Soient y un point isolé et $I = [z^1 z^2]$ une arête telle que $z^1 \neq z^2$. Supposons qu'aucune des trois conditions de la proposition 2.3.2 ne soit remplie. Alors il existe une relation de dominance entre y et des points de I :

- (i) le point y est dominé par des points de I si et seulement s'il est dominé par y^\uparrow (s'il existe, cf. la figure 2.3 et les régions E_1 et E_2 de la figure 2.6) ou par z^2 (régions B et E_3);
- (ii) si y n'est dominé par aucun point de I , alors la partie de I dominée par y est soit $[y^\uparrow y^{\leftrightarrow}]$ (les deux projections existent, cf. la figure 2.4 et la région D_2), $[z^1 y^{\leftrightarrow}]$ (seule y^{\leftrightarrow} existe, voir la région D_1), $[y^\uparrow z^2]$ (seule y^\uparrow existe, cf. région D_3) ou $I = [z^1 z^2]$ entière (aucune projection n'existe, cf. région C).

Démonstration.

- (i) Notons tout d'abord que y^\uparrow existe si et seulement si $z_1^1 \leq y_1 \leq z_1^2$. Si $y_1 < z_1^1$, alors aucun point de I ne domine y . Si $y_1 > z_1^2$, alors y est dominé par des points de I si et seulement si $y_2 \geq z_2^2$. (c'est-à-dire qu'on a $z^2 \leq y$). Autrement, y^\uparrow existe et est un point réalisable pouvant être comparé à y . Si y^\uparrow ne domine pas y , alors aucun autre point $\bar{z} \in I$ ne peut le dominer puisque pour $\bar{z}_1 < y_1^\uparrow = y_1$ nous avons nécessairement $\bar{z}_2 > y_2^\uparrow > y_2$.
- (ii) Supposons d'abord l'existence de y^\uparrow et y^{\leftrightarrow} . Les deux projections sont dominées par y puisque celui-ci n'est dominé par aucun point de I et n'appartient pas à I . Grâce au lemme 2.3.1, nous déduisons que la sous arête $[y^\uparrow y^{\leftrightarrow}]$ est dominée par y . Si y^\uparrow n'existe pas, alors le fait que y ne soit pas dominé par z^2 implique que y domine z^1 . De même, si y^{\leftrightarrow} n'existe pas, on peut en déduire que y domine z^2 . En fonction de l'existence de y^\uparrow et y^{\leftrightarrow} , on peut déduire du lemme 2.3.1 que $[z^1 y^{\leftrightarrow}]$, $[y^\uparrow z^2]$ ou $[z^1 z^2]$ sont dominés par y comme annoncé. Dans tous les cas, l'application de la proposition 2.3.2 implique qu'il n'y a pas de relation de dominance entre y et tout autre point de I .

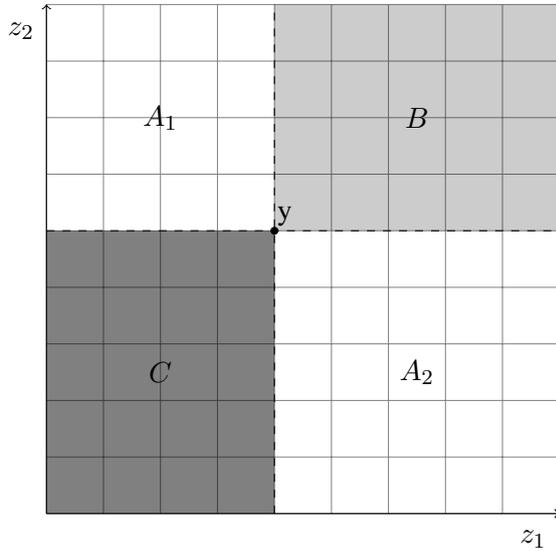


Figure 2.5 – Tout point situé dans les régions A_1 ou A_2 n'a aucune relation de dominance avec y . y domine tout point dans B et est dominé par tout point dans C .

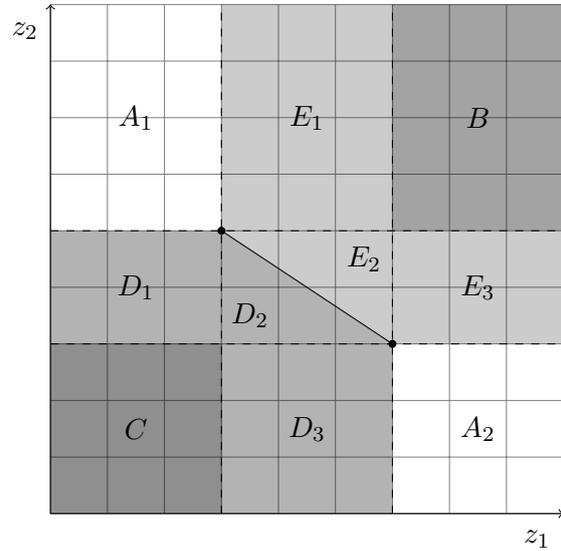


Figure 2.6 – Tout point situé dans les régions A_i n'a aucune relation de dominance avec l'arête ; tout point de l'arête domine tout point appartenant à B tandis que seule une sous-partie de l'arête domine les points situés dans les régions E_i . C représente la région dans laquelle tout point domine l'intégralité de l'arête et les régions D_i regroupent les points ne dominant qu'une partie de l'arête.

□

Comme mentionné précédemment, l'arête $I = [z^1 z^2]$ telle que considérée dans la proposition 2.3.3 est initialement représentée par deux points consécutifs z^1 et z^2 dans Z_{ex} . Des informations supplémentaires indiquent que ces deux points sont connectés ainsi que leur nature (dominés ou non, voir remarque 2.3.2). Si la proposition 2.3.3 permet de conclure que :

- $[z^1 z^2]$ est dominé, alors nous supprimons simplement les deux points de Z_{ex} ;
- $[y^\uparrow z^2]$ est dominé (sans dominance de $[z^1 y^\uparrow]$), alors le point z^2 est remplacé par y^\uparrow dans Z_{ex} avec l'information supplémentaire que y^\uparrow est dominé ;
- $[z^1 y^\leftrightarrow]$ est dominé (sans dominance de $[y^\leftrightarrow z^2]$), alors le point z^1 est remplacé par y^\leftrightarrow dans Z_{ex} avec l'information supplémentaire que y^\leftrightarrow est dominé ;
- $[y^\uparrow y^\leftrightarrow]$ est dominé (sans dominance de $[z^1 y^\uparrow]$ ni de $[y^\leftrightarrow z^2]$), alors y^\uparrow et y^\leftrightarrow sont ajoutés à Z_{ex} entre z^1 et z^2 avec l'information supplémentaire que z^1 et y^\uparrow sont connectés, y^\leftrightarrow et z^2 sont connectés et y^\uparrow et y^\leftrightarrow sont dominés.

2.3.2 Comparaison de deux arêtes dont l'intérieur relatif est non vide

Deux cas sont à considérer lors de la comparaison de deux arêtes dont l'intérieur relatif est non vide : soit les arêtes se coupent en leur intérieur, soit elles ne le font pas. Tous les cas possibles

d'intersection non vide sont décrits par la proposition 2.3.4. Pour deux points distincts y^1 et y^2 , nous notons (y^1y^2) la droite passant par ces deux points.

Proposition 2.3.4. Soient $I_1 = [y^1y^2]$ et $I_2 = [z^1z^2]$ deux arêtes se coupant en leur intérieur relatif, c'est-à-dire que $]y^1y^2[\cap]z^1z^2[\neq \emptyset$. Les affirmations suivantes sont vérifiées :

- (i) si (y^1y^2) et (z^1z^2) sont parallèles (i.e. confondues), alors $I_1 \cap I_2$ est une sous arête commune à I_1 et I_2 et il n'y a aucune dominance entre elles. $I_1 \cup I_2$ est toujours une arête ;
- (ii) si (y^1y^2) et (z^1z^2) ne sont pas parallèles, alors $I_1 \cap I_2$ est réduit à un point noté p . Nous considérons les sous arêtes $[y^1p]$ et $[py^2]$ de I_1 et les sous arêtes $[z^1p]$ et $[pz^2]$ de I_2 . Il n'y a pas de dominance entre $[y^1p]$ et $[pz^2]$ ni entre $[z^1p]$ et $[py^2]$ (cf. Figure 2.7).

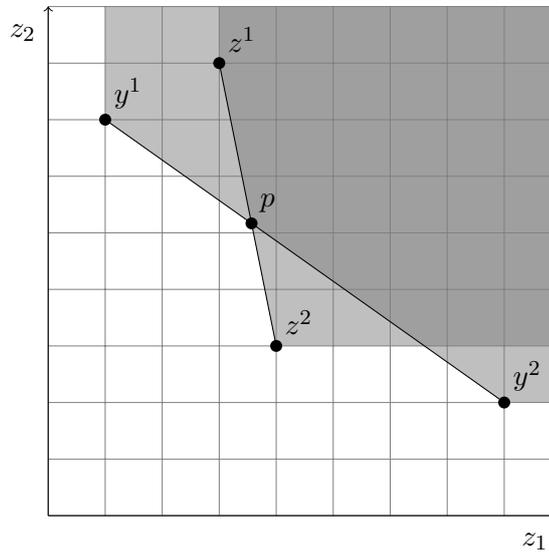


Figure 2.7 – Illustration de la proposition 2.3.4 ii). Aucun point de $[y^1p]$ ne peut dominer $[pz^2]$ et aucun point de $[pz^2]$ ne peut dominer $[y^1p]$.

Démonstration.

- (i) Il n'y a rien à montrer.
- (ii) Trivialement, $[y^1y^2]$ et $[z^1z^2]$ ne se coupent qu'en un unique point p , autrement les deux arêtes seraient parallèles. Pour tout couple de points $t \in [y^1p] \cup [z^1p]$ et $u \in [py^2] \cup [pz^2]$, nous avons $t_1 \leq p_1 \leq u_1$ et $t_2 \geq p_2 \geq u_2$. L'égalité entre t_i et u_i , pour $i \in \{1, 2\}$, implique immédiatement que $t = p = u$. Il n'existe donc pas de dominance entre $[y^1p]$ et $[pz^2]$ ni entre $[z^1p]$ et $[py^2]$.

□

Dans le cas (i) de la proposition 2.3.4, l'union de I_1 et I_2 est algorithmiquement réalisée en remplaçant ces arêtes dans Z_{ex} par une nouvelle arête constituée des extrémités de I_1 et I_2 qui minimisent et maximisent, respectivement, le premier objectif. Dans le cas (ii), l'arête $I_1 = [y^1y^2]$ est remplacée par ses deux sous arêtes $I_{11} = [y^1p]$ et $I_{12} = [py^2]$. De même, $I_2 = [z^1z^2]$ est

remplacée par $I_{21} = [z^1 p]$ et $I_{22} = [p z^2]$. Grâce à la proposition 2.3.4 (ii), la comparaison entre I_1 et I_2 peut être remplacée par deux comparaisons, une entre I_{11} et I_{21} et l'autre entre I_{12} et I_{22} .

Il reste maintenant à expliquer comment comparer deux arêtes ne se coupant pas en leur intérieur relatif. Dans un premier temps, la proposition 2.3.5 dénombre les cas pour lesquels aucune relation de dominance ne peut avoir lieu entre deux arêtes. Ensuite, la proposition 2.3.6 décrit la comparaison et la mise à jour des arêtes dans les cas restant.

Proposition 2.3.5. Soient $I_1 = [y^1 y^2]$ et $I_2 = [z^1 z^2]$ deux arêtes telles que $y^1 \neq y^2$ et $z^1 \neq z^2$ et n'ayant pas d'intersection en leur intérieur relatif. Il n'y a pas de relation de dominance entre I_1 et I_2 dans les cas suivants :

- (i) $y^2 = z^1$ ou $z^2 = y^1$;
- (ii) $y_1^2 < z_1^1$ et $y_2^2 > z_2^1$;
- (iii) $z_1^2 < y_1^1$ et $z_2^2 > y_2^1$.

Démonstration.

Une preuve de l'affirmation (i) est donnée par la proposition 2.3.4 (ii). Pour les affirmations restantes, nous supposons, sans perte de généralité, que nous sommes dans la situation (ii). Tout couple de points $\bar{y} \in [y^1 y^2]$ et $\bar{z} \in [z^1 z^2]$ satisfait les inégalités $\bar{y}_1 \leq y_1^2 < z_1^1 \leq \bar{z}_1$ et $\bar{y}_2 \geq y_2^2 > z_2^1 \geq \bar{z}_2$. L'affirmation en découle directement.

□

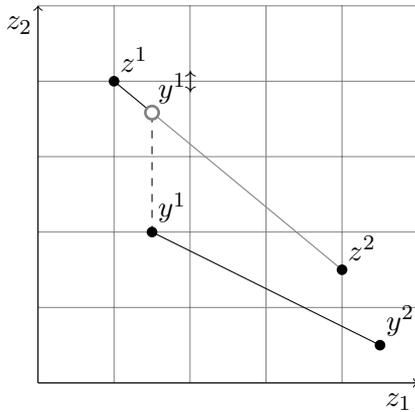


Figure 2.8 – y^{\dagger} constitue l'extrémité droite ouverte d'une arête potentiellement non dominée.

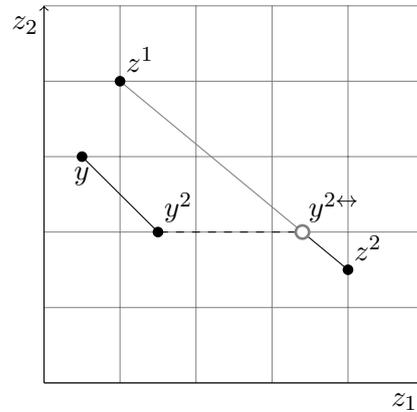


Figure 2.9 – y^{\leftrightarrow} constitue l'extrémité gauche ouverte d'une arête potentiellement non dominée.

Proposition 2.3.6. Soient $I_1 = [y^1 y^2]$ et $I_2 = [z^1 z^2]$ deux arêtes ne se coupant pas dans leur intérieur relatif. Supposons qu'aucune des trois conditions énumérées dans la proposition 2.3.5 ne soit vérifiée. Alors une des deux arêtes (supposons sans perte de généralité qu'il s'agisse de I_1) reste potentiellement non dominée tandis que des parties de l'autre arête sont dominées. Soit y^{\dagger} la projection de y^1 sur \bar{I}_2 suivant l'axe des ordonnées et soit y^{\leftrightarrow} la projection de y^2 sur \bar{I}_2 suivant l'axe des abscisses (en supposant que ces projections existent).

- (i) si les deux projections existent, alors $[y^{1\downarrow}y^{2\leftrightarrow}]$ est dominée et $[z^1y^{1\downarrow}]$ ainsi que $]y^{2\leftrightarrow}z^2]$ restent potentiellement non dominées ;
- (ii) si seule $y^{1\downarrow}$ existe, alors $[y^{1\downarrow}z^2]$ est dominée et $[z^1y^{1\downarrow}]$ reste potentiellement non dominée (voir figure 2.8) ;
- (iii) si seule $y^{2\leftrightarrow}$ existe, alors $[z^1y^{2\leftrightarrow}]$ est dominée et $]y^{2\leftrightarrow}z^2]$ reste potentiellement non dominée (voir figure 2.9) ;
- (iv) si aucune projection n'existe, alors l'arête $[z^1z^2]$ est entièrement dominée.

Démonstration.

Appliquer la proposition 2.3.3 pour comparer une arête aux extrémités de l'autre arête (et vice-versa) implique qu'au moins une des quatre extrémités domine une partie de l'autre arête. Sans perte de généralité, soit I_2 l'arête contenant des points dominés. Puisque que les arêtes ne se coupent pas, aucun point de I_1 ne peut être dominé par I_2 .

Supposons tout d'abord que les deux projections $y^{1\downarrow}$ et $y^{2\leftrightarrow}$ existent, et soit $\bar{z} \in [y^{1\downarrow}y^{2\leftrightarrow}]$. Alors, il existe $\alpha \in [0, 1]$ tel que $\bar{z} = \alpha y^{1\downarrow} + (1 - \alpha)y^{2\leftrightarrow}$. Définissons $\bar{y} := \alpha y^1 + (1 - \alpha)y^2$, alors $\bar{y} \leq \bar{z}$ puisque $y^1 \leq y^{1\downarrow}$ et $y^2 \leq y^{2\leftrightarrow}$. Par définition des projections $y^{1\downarrow}$ et $y^{2\leftrightarrow}$, aucune autre partie de I_2 n'est dominée. Cela prouve (i).

Si seule $y^{1\downarrow}$ existe, alors nous considérons le point $\tilde{y} \in I_1$ tel que $\tilde{y}^{\leftrightarrow} = z^2$, s'il existe. Si c'est le cas, alors la même preuve que pour le cas (i) peut être appliquée en utilisant la sous arête $[y^1\tilde{y}]$ au lieu de $[y^1y^2]$ et $[y^{1\downarrow}z^2]$ au lieu de $[y^{1\downarrow}y^2]$. Autrement, y^1 domine à la fois $y^{1\downarrow}$ et z^2 . L'affirmation (ii) est donc vérifiée par le lemme 2.3.1.

Le cas (iii) est symétrique au cas (ii) et combiner les arguments pour les cas (ii) et (iii) permet de prouver le cas (iv).

□

La mise à jour de Z_{ex} et des informations additionnelles par l'application de la proposition 2.3.6 se fait de façon similaire à la mise à jour par l'application de la proposition 2.3.3.

2.3.3 Formulation de la procédure de mise à jour

Les lignes 3 à 6 de l'algorithme 2.1 (page 45) peuvent être détaillées au moyen des propositions précédentes. L'algorithme 2.2 qui en résulte est ainsi basé sur la comparaison paire à paire d'arêtes I_1 et I_2 par rapport à la dominance. Tout d'abord, le cas de la comparaison de deux points est considéré comme un cas particulier (ligne 2). Les propositions 2.3.2 et 2.3.3 gèrent le cas où une arête est réduite à un point (ligne 5). Enfin, il reste à considérer le cas où les deux arêtes I_1 et I_2 ont un intérieur relatif non vide. Si ces arêtes ne se coupent pas, alors il suffit d'appliquer les propositions 2.3.5 et 2.3.6 (ligne 17). Autrement, les arêtes sont fusionnées si leurs droites respectives sont confondues (ligne 11) ou bien elles sont coupées et leurs sous arêtes sont traitées séparément d'après la proposition 2.3.4.

Remarque 2.3.4 – Il est à noter que l'implémentation de l'algorithme vise à éviter l'exécution de cette procédure dès que c'est possible. C'est par exemple le cas quand la dominance mutuelle entre arêtes peut être évitée immédiatement. Pour observer cela, supposons que l'arête I_1 soit donnée. Toute autre arête I_2 située dans les zones A_1 ou A_2 décrites par la figure 2.6 peuvent être ignorées puisqu'il n'y aura pas de relation de dominance entre I_1 et I_2 . Si les arêtes de E et Z sont stockées et considérées dans l'ordre donné par la remarque 2.3.1, alors tester toutes les arêtes suivant une telle arête I_2 peut être évité.

Entrée deux arêtes I_1 et I_2

```
1: si  $I_1$  et  $I_2$  sont des points alors
2:   Faire la comparaison paire à paire classique
3: sinon
4:   si  $I_1$  ou  $I_2$  est un point alors
5:     Faire la comparaison définie par les propositions 2.3.2 et 2.3.3
6:   sinon
7:     //  $I_1$  et  $I_2$  sont des arêtes
8:     si  $I_1$  et  $I_2$  se coupent dans leur intérieur relatif alors
9:       // Application de la proposition 2.3.4
10:      si  $I_1$  et  $I_2$  sont parallèles alors
11:        Calculer l'union des deux arêtes
12:      sinon
13:        Séparer  $I_1$  et  $I_2$  en les sous arêtes  $I_{11}$  et  $I_{12}$ ,  $I_{21}$  et  $I_{22}$ 
14:        Appliquer la proposition 2.3.6 pour comparer  $I_{11}$  et  $I_{21}$ 
15:        Appliquer la proposition 2.3.6 pour comparer  $I_{12}$  et  $I_{22}$ 
16:      fin si
17:    sinon
18:      Appliquer les propositions 2.3.5 et 2.3.6 pour comparer  $I_1$  et  $I_2$ 
19:    fin si
20:  fin si
21: fin si
```

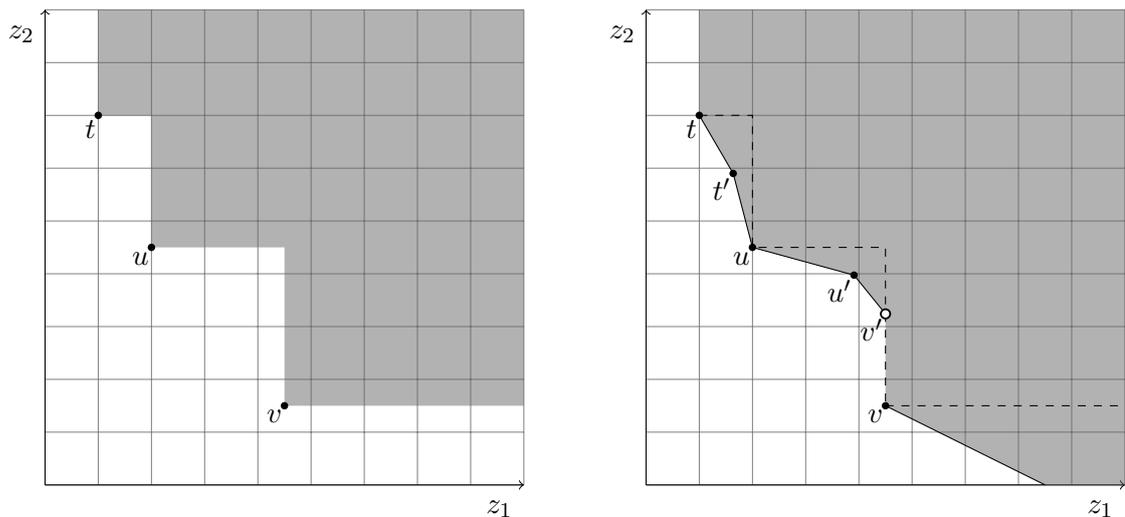
Algorithme 2.2: Procédure update

2.4 Améliorations de l’algorithme de Branch & Bound

Dans cette section, nous proposons plusieurs affinements afin d’améliorer l’efficacité de l’algorithme corrigé. Ces améliorations s’appliquent aux différents composants présentés dans le premier chapitre (page 22).

2.4.1 Ensemble bornant supérieurement

L’ensemble bornant supérieurement U est amélioré de deux façons. Tout d’abord, la nouvelle représentation des solutions potentiellement non dominées améliore naturellement l’ensemble U . En effet, [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005] utilisent D_{ex} , l’ensemble des points extrêmes potentiellement non dominés, en guise d’ensemble bornant supérieurement (voir figure 2.10(a)). Or, contrairement à D_{ex} , Z ne contient aucun point dominé et représente surtout l’ensemble exact des points potentiellement non dominés connus à chaque étape de l’algorithme. Puisque cet ensemble contient également les informations représentant les arêtes contenant les points non dominés non extrêmes, l’ensemble bornant correspondant est par conséquent aussi serré que possible (figure 2.10(b)).



(a) Région bornée par l’ensemble de la figure 2.2(a).

(b) Région bornée par l’ensemble de la figure 2.2(b) comme utilisé par [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005].

Figure 2.10 – Comparaison entre les ensembles bornant induits par les deux représentations de Z comme proposé dans notre amélioration.

Par ailleurs, Mavrotas et Diakoulaki démarrent l’algorithme avec un ensemble U vide, ce qui implique que le sondage par dominance ne sera pas efficace avant la visite d’un nombre potentiellement élevé de feuilles de l’arbre de recherche. Or, nous avons vu dans le premier chapitre que d’autres méthodes proposent d’initialiser l’ensemble bornant supérieurement avec de bonnes solutions dans le but de fournir dès le départ un filtrage et un sondage efficaces. Nous proposons donc de faire de même en initialisant U avec les points de l’ensemble Y_{SN} . En effet, ces points présentent l’avantage d’être non dominés et donc de rester dans Z_{ex} jusqu’à la fin de l’exécution. De

ce fait, ils engendrent un ensemble bornant supérieurement efficace puisque ces points ne peuvent être remis en question.

Les relations entre ces ensembles bornant supérieurement sont représentées par la figure 2.12.

Remarque 2.4.1 – Il est à noter que nous supposons la résolution des MILP correspondants raisonnable en terme de coût. Néanmoins, ce calcul représente un investissement intéressant puisque les solutions obtenues seront conservées.

Enfin, nous proposons d’améliorer encore l’ensemble U initial en tirant parti de la mixité des variables. En effet, chaque point non dominé supporté appartient à une combinaison efficace de points de Y_N . Ces combinaisons sont intéressantes puisqu’elles correspondent à des feuilles particulières de l’arbre dont les sous problèmes associés sont des MOLP. En résolvant ces MOLP, nous obtenons de nouvelles solutions que nous ajoutons à U afin de resserrer l’ensemble bornant tout en élargissant sa portée.

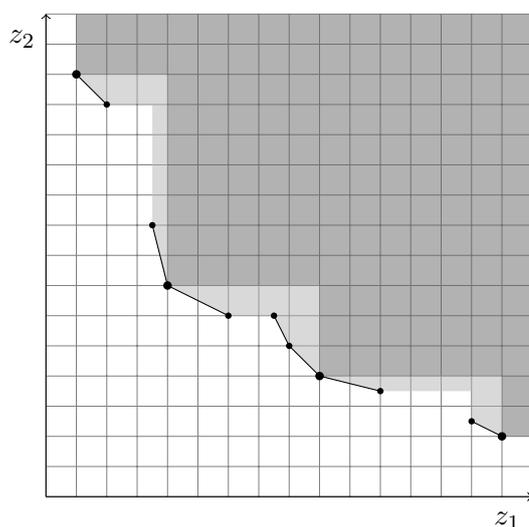


Figure 2.11 – Comparaison de la zone bornée par les points de Y_{SN} (en gris foncé) et de celle bornée par les points Y_{SN} et leurs MOLP (la partie gris clair s’ajoute).

2.4.2 Ensemble bornant inférieurement

Lors de la visite d’un nœud donné, plusieurs ensembles bornant inférieurement peuvent être considérés pour borner l’ensemble \bar{Y} du sous problème \bar{P} courant. Le point idéal y^{ILLP} de la relaxation linéaire de \bar{P} était proposé par [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005] tandis que d’autres auteurs ([Sourd et Spanjaard, 2008, Jorge, 2010, Delort, 2011]) proposaient, pour d’autres problèmes, d’utiliser $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^p)_N$. Calculer y^{ILLP} consiste à résoudre deux LP, ce qui rend cet ensemble très rapide à calculer mais probablement éloigné de \bar{Y} . $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^p)_N$ est quant à lui bien plus précis puisqu’il contient des points réalisables, mais le coût de son calcul risque d’être prohibitif car plusieurs MILP doivent être résolus.

Ces deux ensembles bornant sont donc bien évidemment à considérer, mais il est également possible d’envisager des intermédiaires. En particulier, deux ensembles intermédiaires peuvent

être envisagés. Le premier est le point idéal y^I de \bar{Y} . Celui-ci se situe en effet entre $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^P)_N$ et y^{ILP} en terme de borne et de coût (deux résolutions de MILP sont nécessaires). Le second est l’ensemble de solutions $(\text{conv } E_L + \mathbb{R}_{\geq}^P)_N$ de la relaxation linéaire \bar{Y} dont y^{ILP} est le point idéal. Cet ensemble nécessite de résoudre un MOLP, c’est-à-dire que son coût est équivalent à celui de l’évaluation d’une feuille de l’arbre de recherche.

Les différents ensembles bornant considérés sont hiérarchisés dans la figure 2.13.

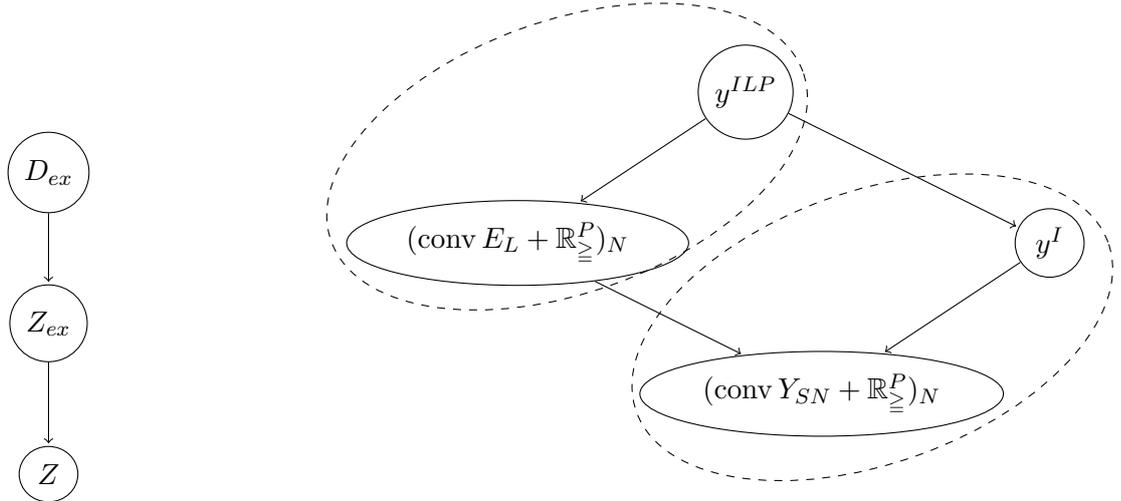


Figure 2.12 – Hiérarchie des ensembles bornant supérieurement, du plus faible au plus fort.

Figure 2.13 – Hiérarchie des ensembles bornant inférieurement, du plus faible au plus fort. On constate qu’il n’existe pas d’ordre total car on ne peut pas comparer $(\text{conv } E_L + \mathbb{R}_{\geq}^P)_N$ et y^I .

2.4.3 Procédure de branchement

La procédure de branchement utilisée dans les articles [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005] considère, lors du parcours en profondeur, les variables binaires sans ordre particulier et les fixe à 0 d’abord puis à 1. Du fait des problèmes considérés, toutes les autres méthodes fixent ces variables à 1 puis 0. On peut en revanche noter dans le cas de [Kiziltan et Yucaoglu, 1983] que lors de la construction d’un point utopique, les auteurs font dépendre l’affectation de chaque variable binaire en fonction de la valeur de ses coefficients dans les fonctions objectifs. La variable est en effet fixée à 1 si elle contribue à améliorer l’objectif considéré et à 0 sinon.

Nous proposons de nous inspirer de cette idée pour faire varier la valeur de la première affectation des variables binaires. Pour cela, nous considérons pour chaque variable binaire x_k le *coefficient d’efficacité*

$$e(k) := (c_k^1 - \mu_1) + (c_k^2 - \mu_2)$$

où μ_1 et μ_2 sont respectivement les moyennes des coefficients du premier et du second objectifs. Si $e(k) \geq 0$ (coin supérieur droit de la figure 2.14), alors fixer la variable x_k à 1 dégrade en moyenne les objectifs, elle doit donc être fixée à 0 d'abord. Si au contraire $e(k) < 0$ (coin inférieur gauche de la figure 2.14), alors x_k améliore en moyenne les objectifs en étant fixée à 1, c'est donc cette valeur qui privilégiée en premier.

Enfin, puisque nous adoptons un parcours en profondeur, il est nécessaire de déterminer un ordre dans lequel considérer les variables binaires. Pour cela, nous pouvons à nouveau utiliser le coefficient d'efficacité $e(k)$. Si $e(k_1) < e(k_2) < 0$ pour deux variables x_{k_1} et x_{k_2} , les deux seront fixées à 1 d'abord, mais il semble également plus intéressant de fixer x_{k_1} avant x_{k_2} . En effet, x_{k_1} contribuera en moyenne davantage à améliorer les deux objectifs que x_{k_2} . Un raisonnement similaire s'applique entre deux variables ayant des coefficients d'efficacité positifs. Ainsi, nous proposons d'ordonner les variables binaires par valeur absolue décroissante de leur coefficient d'efficacité et de fixer les variables suivant cet ordre.

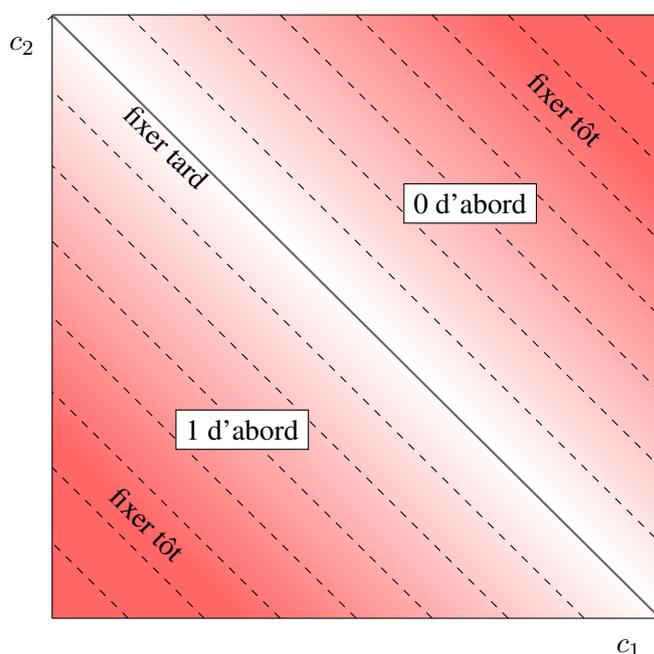


Figure 2.14 – La valeur des deux coefficients objectifs c_1 et c_2 de chaque variable binaire permet de déterminer la valeur à lui attribuer en premier ainsi que le moment où elle doit être fixée.

Les figures 2.15 et 2.16 présentent l'apport attendu d'une stratégie de branchement adéquate sur la taille de l'arbre de recherche. Sur la figure 2.15, les nœuds feuilles dont les MOLP contribuent à l'ensemble non dominé final sont *dispersés* et nécessitent de visiter une grande part de l'arbre de recherche pour être trouvés. De ce fait, de nombreux nœuds inutiles doivent être explorés. Sur la figure 2.16, la stratégie de branchement permet de réunir tous les nœuds feuilles intéressants dans un sous-arbre réduit. L'obtention précoce des meilleurs points de l'ensemble non dominé permet par la suite de sonder plus efficacement le reste de l'arbre de recherche. L'exploration est donc fortement réduite, ce qui a un impact direct sur le temps de résolution.

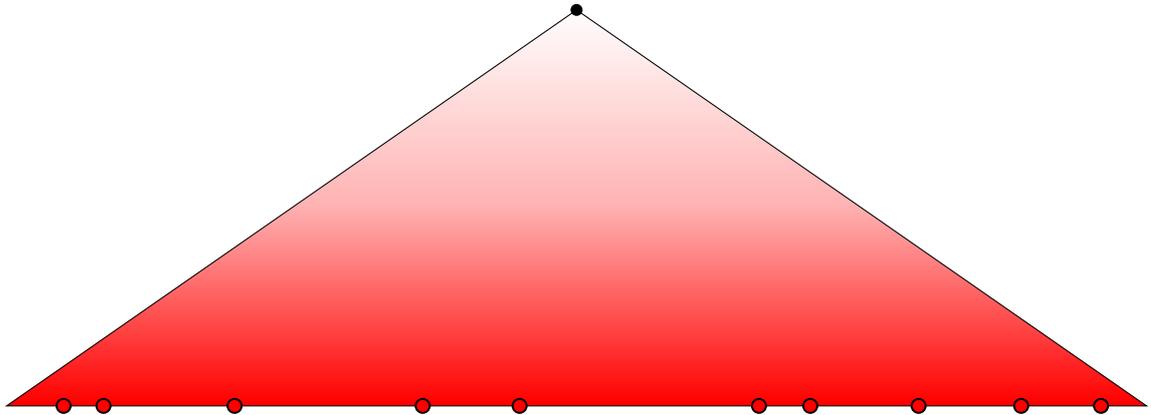


Figure 2.15 – Sans stratégie de branchement particulière, un grand arbre binaire doit être exploré pour trouver les BOLP qui fourniront des solutions efficaces (en rouge).

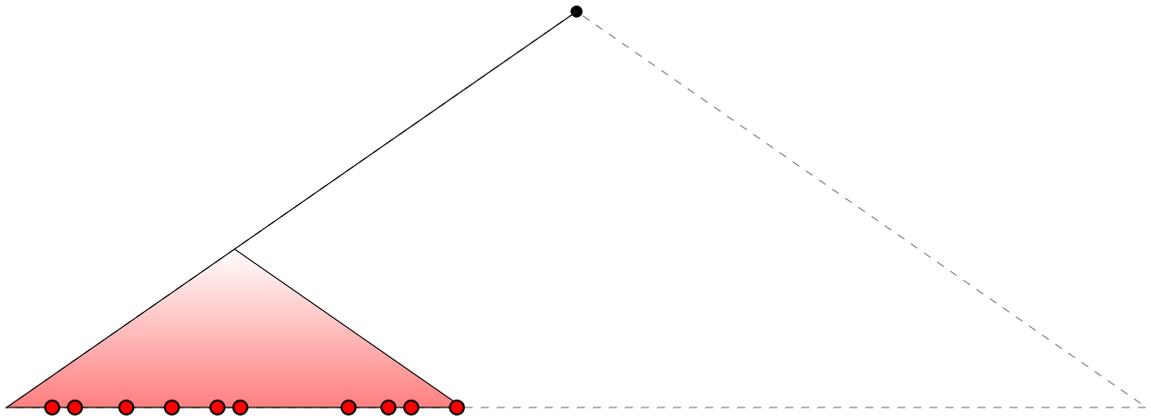


Figure 2.16 – Avec un branchement approprié, la taille de la partie à explorer peut être fortement réduite, ce qui permet de sonder rapidement le reste de l’arbre binaire.

2.5 Résultats expérimentaux

Les différentes améliorations proposées ont été testées et les résultats de ces expérimentations sont présentés dans cette section. Nous comparons séparément l’apport de chacune de nos propositions avant de nous intéresser à leurs interactions. Les expérimentations ont été menées sur un ordinateur portable équipé d’un processeur Core2Quad à 1.8GHz et de 4Go de RAM. Le système d’exploitation utilisé est Debian GNU/Linux 6.0 en version 32 bits.

Les instances de problèmes étudiées ont été générées aléatoirement en respectant les intervalles de valeurs indiqués dans [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005]. Ces intervalles de coefficients sont les suivants :

- $[-10, 10]$ pour les coefficients des variables continues dans les fonctions objectifs ;
- $[-200, 200]$ pour les coefficients des variables binaires dans les fonctions objectifs ;
- $[-100, 50]$ pour les parties droites des contraintes ;
- $[-20, 1]$ pour les coefficients des contraintes.

La densité de la matrice des contraintes est de 25%.

Six classes de problèmes différant par leur taille ont été étudiées. Ces classes sont définies par $m \times n(n_b)$, représentant les problèmes à m contraintes et n variables, dont n_b sont binaires. Pour chaque classe, dix instances ont été générées aléatoirement suivant une loi uniforme.

Les tableaux 2.1 et 2.2 présentent le temps d'exécution moyen et le nombre moyen de nœuds visités pour chaque classe d'instance. Chaque classe est résolue avec les différents ensembles bornant supérieurement, avec et sans la stratégie d'exploration proposée. Enfin, les deux tableaux présentent les résultats pour les mêmes instances, mais avec et sans initialisation de l'ensemble bornant supérieurement. Par la suite, nous considérerons la ligne 1A du tableau 2.1 comme référence pour nos comparaisons. Il s'agit en effet des résolutions comportant nos corrections mais dont les composants sont tels que proposés par [Mavrotas et Diakoulaki, 2005], c'est-à-dire sans les améliorations proposées dans la section 2.4.

2.5.1 Comparaison des ensembles bornant inférieurement

Les quatre ensembles bornant inférieurement présentés dans la section 2.4.2 ont été testés et comparés. Pour comparer leur impact sur la résolution, les lignes pertinentes à considérer sont 1A, 1C, 1E et 1G, dans le tableau 2.1.

Lorsqu'on observe le nombre moyen de nœuds visités par résolution, les résultats correspondent à la qualité des ensembles bornant :

- y^{ILP} est nettement moins efficace que les autres ensembles bornant inférieurement (253 992 nœuds visités dans la classe des plus grandes instances) ;
- y^I est meilleur puisqu'il permet de sonder davantage de nœuds (178 137 nœuds visités) ;
- $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ est encore plus efficace (16 870 nœuds visités) ;
- enfin, $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^2)_N$ est l'ensemble bornant inférieurement le plus efficace puisqu'il permet de ne visiter que 10 808 nœuds dans l'arbre de recherche.

Ces résultats sont cohérents avec ce qui était attendu dans la figure 2.13.

Néanmoins, les résultats en terme de temps de résolution ne suivent pas la même tendance. En effet, même si les ensembles bornant correspondants sont plus serrés, il est plus coûteux de résoudre un problème P que sa version relâchée \bar{P} . Les temps de résolution en fonction de l'ensemble bornant inférieurement sont ordonnés comme suit :

- la résolution la plus lente se fait avec y^I (1 172 s dans la classe des plus grandes instances) ;
- l'utilisation de y^{ILP} est légèrement plus rapide (849 s) ;
- la résolution moyenne en utilisant $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^2)_N$ prend 584 s ;
- la résolution la plus rapide se fait avec $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ en 66 s en moyenne.

Au vu de ces derniers résultats, $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ semble être l'ensemble bornant inférieurement le plus intéressant pour accélérer le temps de résolution. Le facteur d'amélioration par rapport au temps de référence est en effet de 12,8 en moyenne pour la classe des plus grandes instances.

Néanmoins, il convient de rappeler que des programmes linéaires biobjectifs *génériques* sont résolus, ce qui est en défaveur de $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^2)_N$. En effet, il n'est pas possible d'utiliser un solveur dédié pour les résolutions de MILP nécessaires à son calcul, ce qui rend le calcul de l'ensemble bornant très consommateur en temps. Dans le cadre d'un problème à structure, il serait possible de tirer parti d'un tel solveur pour accélérer grandement le calcul de l'ensemble bornant, ce qui rendrait ce dernier bien plus intéressant.

| | EBI | Branchement | 10 × 10(5) | 20 × 20(10) | 30 × 30(15) | 40 × 40(20) | 50 × 50(25) | 60 × 60(30) | |
|----|--|-------------|-----------------|-------------|--------------|----------------|-----------------|------------------|--------------------|
| 1A | y^{LLP} | inactif | temps #nœuds | 0 38,2 | 0,3 426,8 | 8,1 5225,8 | 25,6 12492,2 | 252 85623,2 | 849,8 253992,4 |
| 1B | | actif | temps #nœuds | 0 29,8 | 0,4 276 | 3,7 2433,6 | 24 11632,8 | 189 65898,8 | 1289 378988,8 |
| 1C | $(\text{conv } E_L + \mathbb{R}_{\Xi}^2)_N$ | inactif | temps #nœuds | 0 31,6 | 0,5 224 | 2,4 1372 | 4,7 2061,8 | 17,8 5471,8 | 66,4 16870,8 |
| 1D | | actif | temps #nœuds | 0 17,6 | 0,1 60 | 0,2 193 | 0,7 267,8 | 2,1 652,6 | 10,7 2509 |
| 1E | y^I | inactif | temps #nœuds | 0 36 | 0,5 426,8 | 11,8 5085,8 | 40,4 12073,8 | 336,8 66286,6 | 1172,3 178137,2 |
| 1F | | actif | temps #nœuds | 0 27,6 | 0,4 275,8 | 5,8 2364 | 37,2 10775,8 | 270 49656,8 | 1482,8 214421,2 |
| 1G | $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\Xi}^2)_N$ | inactif | temps #nœuds | 0 29,6 | 0,9 207,6 | 10,6 1235,6 | 25,7 1881 | 127,2 4045 | 584,6 10808 |
| 1H | | actif | temps #nœuds | 0 16,6 | 0,1 56,6 | 2 169,2 | 4,8 221,6 | 19,8 392 | 71,9 783,4 |

Table 2.1 – Temps de résolution et nombre de nœuds visités en fonction de l'ensemble bornant inférieurment et de la stratégie de branchement utilisés.

| | EBI | Branchement | $10 \times 10(5)$ $20 \times 20(10)$ $30 \times 30(15)$ $40 \times 40(20)$ $50 \times 50(25)$ $60 \times 60(30)$ | | | | | | | | | | | |
|----|---|-------------|--|---------|-------|---------|-------|---------|-------|---------|-------|---------|--------|----------|
| | | | temps | #noeuds | temps | #noeuds | temps | #noeuds | temps | #noeuds | temps | #noeuds | temps | #noeuds |
| 2A | y_{ILLP} | inactif | 0,1 | 31,8 | 0,3 | 297,4 | 4 | 2568,6 | 15,6 | 7418,6 | 133,1 | 49403,2 | 418,9 | 120979,6 |
| | | actif | 0,1 | 0,1 | 0,2 | 276 | 4 | 2433,4 | 25,2 | 11631,8 | 176,9 | 65868,8 | 1291,1 | 378375,8 |
| 2C | $(\text{conv } E_L + \mathbb{R}_{\leq}^2)_N$ | inactif | 0 | 23,8 | 0,1 | 103 | 0,5 | 377,8 | 1,3 | 553,8 | 5,1 | 1533,6 | 18,3 | 4188,4 |
| | | actif | 0 | 18,4 | 0,1 | 58,4 | 0,3 | 188,2 | 0,6 | 253,4 | 1,9 | 585,6 | 10,5 | 2189,4 |
| 2E | y^I | inactif | 0 | 29,6 | 0,6 | 297,4 | 5,9 | 2481,8 | 24,3 | 7090,4 | 175,8 | 35389,2 | 559,9 | 73000 |
| | | actif | 0 | 0 | 0,6 | 275,8 | 5,4 | 2363,8 | 36,7 | 10775 | 243,9 | 49630,4 | 1484,4 | 214073,8 |
| 2G | $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\leq}^2)_N$ | inactif | 0,1 | 21,6 | 0,2 | 88,4 | 3 | 265,4 | 8,1 | 429 | 33,5 | 762 | 131 | 1506,6 |
| | | actif | 0 | 0 | 0,3 | 55 | 2,2 | 205,6 | 5,1 | 20 | 71,3 | 354,2 | 673,2 | |

Table 2.2 – Temps de résolution et nombre de noeuds visités en fonction de l'ensemble bornant inférieurement et de la stratégie de branchement utilisés lorsque l'ensemble bornant supérieurement est initialisé.

2.5.2 Apports de la procédure de branchement proposée

La procédure de branchement proposée dans la section 2.4.3 vise à tirer profit des informations fournies par les fonctions objectif. Le but est en effet d'inférer dans quel ordre il est préférable de fixer les variables et quelle valeur leur donner en premier.

L'apport de cette procédure de cheminement peut être observé en comparant les ligne 1A et 1B du tableau 2.1. Nous pouvons constater que les temps de résolution sont globalement meilleurs quand notre procédure de cheminement est utilisée, mais cela ne s'applique qu'aux instances de petite taille. En effet, le temps moyen pour les instances de taille $60 \times 60(30)$ est nettement plus important avec cette procédure (1289 s contre 849 s pour le temps de référence).

Notre procédure de cheminement semble donc bénéfique jusqu'à une certaine taille puis, dès que cette taille est franchie, elle semble ralentir l'exécution de l'algorithme de résolution. Pour en comprendre la raison, observons également l'impact de cette procédure quand les différents ensembles bornant inférieurement sont utilisés. Pour cela, nous comparons les lignes 1C et 1D (utilisation de $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$), 1E et 1F (utilisation de y^I), 1G et 1H (utilisation de $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^2)_N$).

À partir de ces comparaisons, on peut constater que le même effet se produit quand y^I est utilisé : la stratégie de branchement est profitable jusqu'à l'avant dernière taille d'instance, puis devient nettement défavorable pour les instances les plus grandes. En revanche, ce phénomène n'apparaît pas avec les deux autres ensembles bornant supérieurement, pour lesquels l'utilisation de la procédure de cheminement est au contraire très profitable : on passe de 66 s à 10 s pour $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ et de 584 s à 71 s pour $(\text{conv } \bar{Y}_{SN} + \mathbb{R}_{\geq}^2)_N$.

Une explication possible est que l'inefficacité des ensembles bornant y^{LLP} et y^I ne permet pas de tirer parti de l'ordre de branchement proposé.

La combinaison la plus intéressante est donc avec l'ensemble $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$, qui est l'ensemble le plus intéressant d'après la section précédente. Dans cette configuration (sans initialisation, c'est-à-dire en se restreignant au tableau 2.1), le temps de résolution est en moyenne 79 fois plus rapide que le temps de référence pour les instances de plus grande taille.

2.5.3 Apports de l'initialisation de l'ensemble bornant supérieurement

Le tableau 2.1 présente les résultats d'expérimentation sans initialisation de l'ensemble bornant supérieurement avec de bonnes solutions, tandis que le tableau 2.2 présente les résultats quand l'initialisation proposée en section 2.4.1 est utilisée.

Comparer les temps de la ligne 2A à ceux de la ligne 1A permet d'observer l'apport de l'initialisation. Le temps de résolution passe alors de 849 s à 418 s, et le nombre de nœuds visités de 253 992 à 120 979.

Initialiser avec de bonnes solutions est donc, comme attendu, un facteur d'amélioration pour le temps de résolution, mais on constate que son effet dépend grandement de l'utilisation de la procédure de cheminement. En effet, combinée avec notre procédure de cheminement, l'initialisation de l'ensemble bornant n'apporte pas d'amélioration en terme de temps de résolution. On peut supposer que cela est dû au rôle similaire qu'ont ces deux améliorations, puisqu'elles permettent toutes deux d'obtenir de bonnes solutions qui améliorent U à un stade précoce de la résolution. De ce fait, les bénéfices des deux améliorations sont confondus quand on les applique en même temps, mais c'est l'utilisation de $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ qui, séparément, a le plus d'impact (10,7 s avec cet ensemble bornant contre 18,3 s avec l'initialisation seule).

Enfin, l'utilisation combinée de la stratégie de branchement, de l'initialisation et de l'ensemble bornant inférieurement ($\text{conv } E_L + \mathbb{R}_{\geq}^2$) $_N$ permet de faire descendre le temps moyen de résolution à 10,5 s pour la plus grande classe d'instances.

2.5.4 Tests supplémentaires sur de plus grandes instances

Afin de confirmer nos observations et de mieux cerner l'intérêt de combiner les trois améliorations proposées, nous avons étendu nos tests à une taille plus grande d'instances, générées comme précédemment. Celles-ci sont de type $70 \times 70(35)$ et leurs résultats sont présentés dans le tableau 2.3.

| | EBI | Branchement | Initialisation | $70 \times 70(35)$ | |
|-----|--|-------------|----------------|--------------------|-----------|
| 1A' | y^{ILP} | inactif | non | temps | 7261 |
| | | | | #nœuds | 1790058,8 |
| 1B' | y^{ILP} | actif | oui | temps | 8459,4 |
| | | | | #nœuds | 2062992,8 |
| 1C' | $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ | inactif | non | temps | 180,5 |
| | | | | #nœuds | 36834,8 |
| 1D' | $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ | actif | non | temps | 21,1 |
| | | | | #nœuds | 4242,5 |
| 2A' | y^{ILP} | inactif | oui | temps | 3738,7 |
| | | | | #nœuds | 902701,2 |
| 2D' | $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ | actif | oui | temps | 18 |
| | | | | #nœuds | 3324,6 |

Table 2.3 – Temps de résolution et nombre de nœuds visités en fonction des améliorations proposées pour une taille d'instance plus grande (70 contraintes, 35 variables continues, 35 variables binaires).

Nous avons restreint nos tests aux configurations les plus intéressantes :

- la configuration de référence, c'est-à-dire sans aucune des améliorations proposées (extension de la ligne 1A du tableau 2.1) ;
- les configurations appliquant séparément chaque proposition d'amélioration (lignes 1B, 1C et 2A) ;
- la configuration appliquant toutes ces propositions simultanément (ligne 2D) ;
- la configuration combinant la procédure de branchement à l'ensemble bornant qui lui convient le mieux (ligne 1D).

Ces résultats confirment nos observations précédentes. Utiliser l'ensemble bornant inférieurement ($\text{conv } E_L + \mathbb{R}_{\geq}^2$) $_N$ (180 s, ligne 1C') ou initialiser l'ensemble supérieurement (3738 s, ligne 2A') représentent des améliorations par rapport à l'algorithme de référence (7261 s, ligne 1A'). La

procédure de cheminement semble toujours être un mauvais choix lorsque l'ensemble bornant inférieurement utilisé est y^{ILLP} (8549 s, ligne 1B') mais, combinée à $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$, elle apporte une importante amélioration (21 s, ligne 1D'). Enfin, il s'avère qu'utiliser simultanément les trois propositions constitue la meilleure solution (18 s, ligne 2D').

En comparant ce dernier temps au temps de référence, on observe que la résolution est accélérée par un facteur 403.

2.6 Conclusion

Dans ce chapitre, nous avons corrigé et amélioré l'algorithme de Branch & Bound proposé par [Mavrotas et Diakoulaki, 1998, Mavrotas et Diakoulaki, 2005] dans le cadre biobjectif. Pour cela, nous avons dans un premier temps proposé une représentation de l'ensemble non dominé des programmes linéaires biobjectifs en variables mixtes ainsi qu'une procédure permettant sa mise à jour. Puis, nous avons introduit des améliorations apportant des gains calculatoires significatifs grâce à l'utilisation d'ensembles bornant inférieurement représentant le meilleur compromis entre coût de calcul et efficacité. D'autres améliorations sont basées sur l'observation des coefficients des variables binaires dans les fonctions objectifs, qui nous ont permis de déterminer un prétraitement indiquant l'ordre dans lequel fixer les variables et quelle valeur leur donner en premier. Enfin, la dernière amélioration a consisté en l'initialisation de l'ensemble bornant supérieurement grâce à la détermination des points non dominés supportés.

Cette initialisation de l'ensemble bornant supérieurement pose une nouvelle question. En effet, elle correspond à une extension de la première phase de la méthode en deux phases, couramment utilisées pour traiter les problèmes combinatoires. Il est donc naturel de considérer l'exploitation complète de cette première phase, qui consisterait donc à restreindre l'exploration à la zone de recherche réduite obtenue à partir de l'ensemble bornant supérieurement. Cette réduction de la zone de recherche devrait permettre d'accélérer la résolution globale. Le prochain chapitre s'intéresse donc à l'utilisation de la méthode en deux phases pour résoudre les BOMILP.

Méthodes en deux phases pour les problèmes biobjectifs

Dans ce chapitre, nous nous intéressons à évaluer l'intérêt des méthodes en deux phases pour les problèmes biobjectifs. Une des améliorations proposées dans le chapitre précédent, l'initialisation de l'ensemble des solutions, consiste en la détermination des points non dominés supportés. Il s'agit de la première phase de la méthode en deux phases, et il est donc naturel de s'intéresser à l'apport d'une deuxième phase. Nous nous intéressons donc à deux méthodes déjà utilisées dans la littérature pour accomplir cette deuxième phase, puis nous proposons une nouvelle méthode fondée sur le rapprochement des idées fortes des deux précédentes. Les travaux du présent chapitre ont fait l'objet de communications en conférences internationales [Vincent et al., 2011, Vincent et al., 2013a] et seront prochainement soumis en revue scientifique.

| | | |
|------------|---|-----------|
| 3.1 | Méthode en deux phases avec Branch & Bound | 67 |
| 3.1.1 | Restriction de l'espace de recherche | 67 |
| 3.1.2 | Adaptation des composants du Branch & Bound | 67 |
| 3.2 | Méthode en deux phases avec ranking | 68 |
| 3.2.1 | Adaptation aux programmes en variables mixtes | 69 |
| 3.2.2 | Calcul de la borne β | 69 |
| 3.2.3 | Particularités du contexte mixte | 72 |
| 3.3 | Proposition d'une nouvelle méthode pour la phase 2 | 73 |
| 3.3.1 | Problèmes potentiels de la méthode de ranking | 73 |
| 3.3.2 | Principe de fonctionnement de rank&bound | 75 |
| 3.4 | Résultats expérimentaux | 76 |
| 3.4.1 | Remarques générales | 76 |
| 3.4.2 | Résultats de la méthode de Branch&Bound globale | 77 |
| 3.4.3 | Résultats de la méthode en deux phases avec Branch&Bound | 78 |
| 3.4.4 | Résultats de la méthode en deux phases avec Ranking | 78 |
| 3.4.5 | Résultats de la méthode en deux phases avec Rank&Bound | 79 |
| 3.5 | Conclusion | 79 |

3.1 Méthode en deux phases avec Branch & Bound

La méthode en deux phases, présentée dans la section 1.4.3 (page 29), est directement applicable à partir des composants présentés dans la section précédente. En effet, la détermination des points supportés extrêmes, qui correspond à la première phase, a été proposée pour initialiser l'ensemble bornant supérieurement. À partir de ces points, des zones de recherche peuvent être déterminées et notre algorithme de Branch & Bound peut y être appliqué localement. En réduisant ainsi l'espace de recherche, il est attendu que le nombre de combinaisons à énumérer soit significativement réduit et que la résolution des BOMILP s'en retrouve accélérée.

Nous proposons de réutiliser l'algorithme de Branch & Bound développé dans le chapitre précédent en reprenant les composants ayant donné les meilleurs résultats d'après nos expérimentations, c'est-à-dire :

- l'ensemble bornant inférieurement $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$;
- l'ensemble bornant supérieurement Z_{ex} , initialisé grâce à Y_{SN} ;
- la stratégie de branchement basée sur les coefficients d'efficacité $e(k)$.

Néanmoins, l'algorithme que nous avons présenté est conçu pour accomplir une exploration globale de l'espace de recherche et les stratégies mises en place sont également globales. Nous présentons donc dans cette section un moyen de restreindre l'exécution du Branch & Bound à un triangle donné avant d'adapter certains de ses composants de façon à tirer parti des informations fournies par le triangle visité.

3.1.1 Restriction de l'espace de recherche

Un des principes fondamentaux de la méthode en deux phases est que la structure du problème résolu ne doit pas être modifiée. En effet, la méthode est prévue pour se baser sur des algorithmes mono-objectifs dédiés et toute altération de la structure d'un problème empêcherait d'utiliser ces algorithmes.

Pourtant, il est évident qu'ajouter des contraintes sur les valeurs des objectifs permettraient de délimiter la zone de recherche. Ces contraintes additionnelles auraient en fait un impact sur le sondage par inadmissibilité puisque tout sous-problème n'ayant pas de point admissible dans le triangle visité deviendrait inadmissible. Par ailleurs, les points considérés sont restreints aux triangles visités. Le sondage par dominance peut donc être appliqué localement et s'en retrouve renforcé.

Par conséquent, nous proposons de modifier le sondage par inadmissibilité lui-même : sera considéré inadmissible tout sous problème n'ayant pas de point admissible situé dans la zone de recherche.

Pour ce faire, nous avons recours aux contraintes additionnelles, mais celles-ci ne sont appliquées que lors du calcul d'un ensemble bornant inférieurement. Puisque l'ensemble bornant utilisé est $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$, c'est donc un MOLP que nous modifions et nous n'altérons donc pas sa structure : il s'agit toujours d'un MOLP, résolu de la même façon.

3.1.2 Adaptation des composants du Branch & Bound

La stratégie de branchement peut être améliorée dans chaque triangle visité. Le coefficient d'efficacité présenté dans la section 2.4.3 est en effet généraliste puisqu'il considère équitablement les deux fonctions objectifs. Il oriente donc l'exploration de façon équilibrée, tentant de réduire

autant z_1 que z_2 et ce, quel que soit le triangle (figure 3.1). En fonction de la forme du triangle visité, et en particulier de la pente de son hypoténuse, il est possible d'affiner ce coefficient afin de privilégier les variables qui poussent l'exploration dans le sens souhaité (figure 3.2).

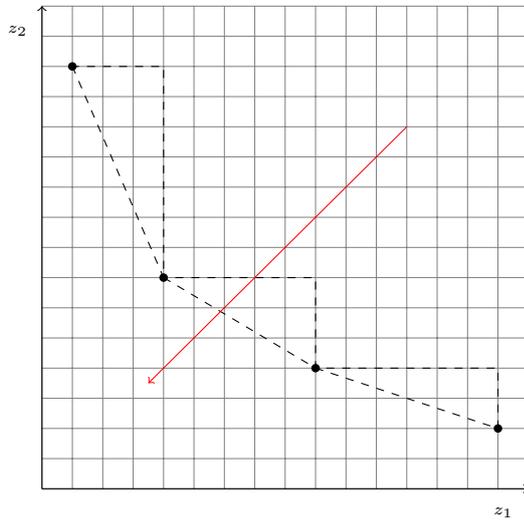


Figure 3.1 – La stratégie de cheminement globale suit une direction générale considérant équitablement les deux fonctions objectif.

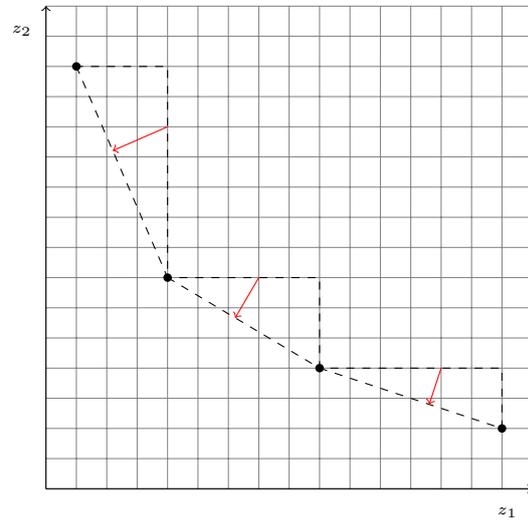


Figure 3.2 – Les stratégies de cheminement locales tirent partie des informations provenant de chaque triangle pour favoriser la recherche en direction de l'hypoténuse des triangles.

Pour cela, nous notons y^r et y^s les deux points définissant, avec leur point nadir local, le triangle courant. Nous utilisons ces deux points pour calculer les poids $\lambda_1 = y_2^r - y_2^s$ et $\lambda_2 = y_1^s - y_1^r$. Ces poids servent à définir le coefficient d'efficacité pondéré

$$e_\lambda(k) = \lambda_1(c_k^1 - \mu_1) + \lambda_2(c_k^2 - \mu_2)$$

pour chaque variable binaire x_k . Grâce à ce coefficient d'efficacité pondéré, la procédure de cheminement suivra une direction locale au triangle courant qui sera donc plus adaptée que la recherche globale.

3.2 Méthode en deux phases avec ranking

La méthode de ranking est une autre possibilité pour accomplir la deuxième phase. Néanmoins, la méthode de ranking, telle que présentée dans la section 1.4.3, ne gère que les variables binaires et il est donc nécessaire de l'adapter au cadre des programmes en variables mixtes.

Pour rappel, la méthode de ranking, étant donnée une pondération λ des fonctions objectif, permet de calculer les k meilleures solutions du problème pondéré. Ainsi, les points réalisables sont obtenus du plus proche de l'hypoténuse du triangle visité au plus éloigné. L'énumération prend fin lorsqu'une borne β est atteinte. La valeur de cette borne correspond au poids du point nadir local le plus éloigné de l'hypoténuse. β symbolise donc la limite au-delà de laquelle aucun point non dominé ne peut être obtenu.

3.2.1 Adaptation aux programmes en variables mixtes

Le problème induit par la présence de variables mixtes provient du fait que des arêtes réalisables existent dans l'espace des objectifs. Dès lors, une infinité de solutions peuvent être énumérées (et classées), ce qui est clairement impossible - et inutile - pour un ranking. Par conséquent, nous proposons de traiter différemment les deux types de variables, à la manière de ce qui a été fait pour le branch & bound. La méthode de ranking peut donc être appliquée de façon classique sur les variables binaires, tandis que les variables continues sont traitées séparément.

Pour chaque combinaison de variables binaires obtenues, les variables continues restantes représentent un MOLP devant être résolu, comme c'est le cas dans un nœud feuille dans l'arbre de recherche du branch & bound.

Ainsi, au lieu d'obtenir un classement des solutions isolées comme dans le cas discret, ce sont ici les MOLP qui sont triés en fonction de leur meilleure solution étant donné le poids λ . Quand un nouveau MOLP est obtenu, il ne reste donc qu'à fusionner l'ensemble de points correspondants à l'ensemble global avant de poursuivre la procédure de ranking. Ce processus est illustré en figure 3.3.

3.2.2 Calcul de la borne β

Grâce à l'ajout de nouveaux points, les points délimitants et la borne β peuvent être abaissés. En revanche, la présence d'arêtes rend la détermination des points délimitants plus complexe puisqu'il ne sert à rien de calculer le nadir local de deux points consécutifs appartenant à une même arête (figure 3.6). Par conséquent, nous proposons d'utiliser la notion de point délimitant présentée dans la définition 1.4.5 qui n'était jusque-là utilisée que dans le contexte multiobjectif. Néanmoins, dans le cadre biobjectif, il nous est possible de savoir quels points non dominés ont contribué à un point délimitant donné, de la même façon que cette information est disponible pour les nadir locaux. Lorsqu'un triangle est visité, nous pouvons grâce à cela ne considérer que les points situés à l'intérieur du triangle pour calculer ses points délimitants.

À partir de la définition 1.4.5, il nous est maintenant possible de déterminer correctement les points délimitants d'un triangle donné.

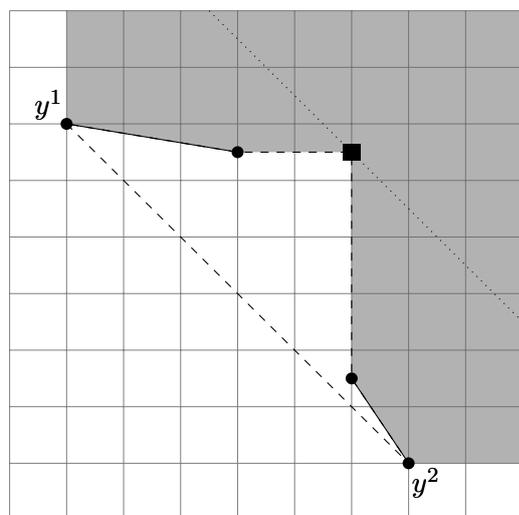
Proposition 3.2.1. *Soient y^a et y^b deux points consécutifs de Z_{ex} appartenant à un triangle donné. En fonction de leur configuration, des ensembles de points délimitants peuvent être définis :*

- (i) *si y^a et y^b ne sont pas connectés (n'appartiennent pas à la même arête) et qu'aucune relation de dominance n'existe entre eux, alors il y a un point $d = (y_1^b, y_2^a)$ (figure 3.4) ;*
- (ii) *si y^a et y^b ne sont pas connectés mais qu'un des points est faiblement non dominé par l'autre point, alors ce point est un point délimitant (figure 3.5) ;*
- (iii) *si y^a et y^b appartiennent à la même arête, alors tout point de $]y^a y^b[$ est un point délimitant (figure 3.6) ;*
- (iv) *de plus, dans le cas (iii), y^a (resp. y^b) est également un point délimitant s'il est connecté à son prédécesseur (resp. successeur).*

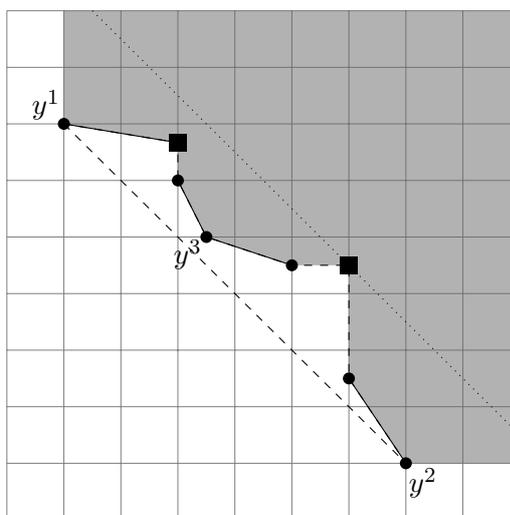
Démonstration.

Pour montrer qu'un point donné est un point délimitant, nous devons montrer qu'il respecte les conditions (i) et (ii) de la définition 1.4.5.

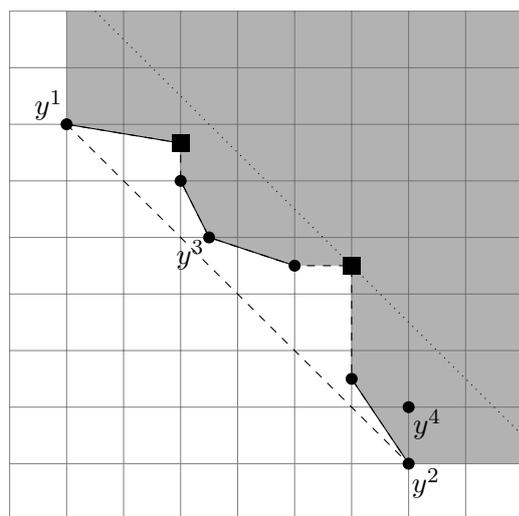
- (i) Ce cas correspond à celui des problèmes discrets.



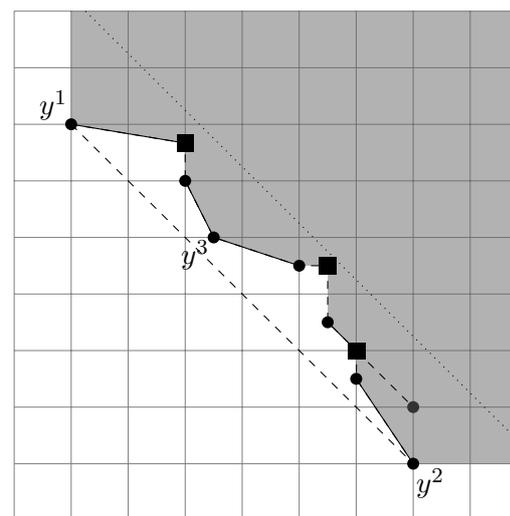
(a) y^1 et y^2 sont déjà connus, ainsi que les MOLP auxquels ils appartiennent. Initialisation de β .



(b) y^3 et son MOLP sont calculés. Mise à jour de β .



(c) y^4 est dominé. Cependant, il se trouve sous β et son MOLP est donc calculé.



(d) En effet, son MOLP permet d'obtenir des points non dominés.

Figure 3.3 – Exemple du processus de la méthode de ranking pour les problèmes en variables mixtes

- (ii) Sans perte de généralité, supposons que y^a soit faiblement non dominé tel que $y_1^a = y_1^b$ et $y_2^a > y_2^b$. Montrons que y^a est un point délimitant. Puisque y^a appartient à Z_{ex} , le fait qu'il ne soit que faiblement non dominé indique qu'il s'agit de l'extrémité droite d'une arête $[y^c y^a]$. De plus, ce point n'est pas strictement dominé et vérifie donc la condition (i) de la définition 1.4.5. Considérons maintenant un point v tel que $y^a \leq v$. Deux cas peuvent se produire :
- $y^a < v$ donc v est strictement dominé.

- $y^a \leq v$ mais $y^a < v$ est faux. À nouveau, deux situations sont possibles :
 - $y_1^a = v_1$, alors soit $v^{\leftrightarrow} \in]y^c y^a[$ tel que $v_2^{\leftrightarrow} = \min(y_2^c, v_2)$. Nécessairement, $\frac{y_1^a + v_1^{\leftrightarrow}}{2} < v$.
 - $y_2^a = v_2$, alors $y^b < v$.

v ne vérifie donc jamais la condition (i) de la définition 1.4.5. En conclusion, y^a est un point délimitant.

- (iii) Montrons que tout point de $]y^a y^b[$ est un point délimitant. Tout d’abord, puisque y^a et y^b appartiennent à une arête non dominée, alors les points de $]y^a y^b[$ sont nécessairement non dominés. Soit $d = \lambda y^a + (1 - \lambda)y^b$ avec $0 < \lambda < 1$. Puisque d est par définition non dominé, alors $d - \epsilon$ l’est également, ce qui vérifie la condition (i) de la définition 1.4.5.

Considérons maintenant un point v tel que $d \leq v$. Deux cas peuvent se produire :

- $d < v$ donc v est strictement dominé.
- $d \leq v$ mais $d < v$ est faux. Nous supposons sans perte de généralité que $d_1 = v_1$ et $d_2 < v_2$. À nouveau, deux cas sont envisageables :
 - si $y_2^a < v_2$, alors $y^a < v$, donc v est strictement dominé.
 - si $v_2 < y_2^a$, alors nous considérons le point $v^{\leftrightarrow} \in]y^a d[$ tel que $v_2^{\leftrightarrow} = v_2$. Dans ce cas, $\frac{d_1 + v_1^{\leftrightarrow}}{2} < v$.

Par conséquent, v est nécessairement strictement dominé, donc d vérifie la condition (ii) de la définition 1.4.5.

- (iv) Il reste à montrer que y^a et y^b sont également des points délimitants s’ils sont connectés à leur prédécesseur et successeur. Sans perte de généralité, considérons le cas de y^a . Par hypothèse, ce point n’est pas strictement dominé et vérifie donc la condition (i) de la définition 1.4.5. Concernant la condition (ii) de la même définition, l’argument est le même que pour les points de $]y^a y^b[$ du cas (iii) en choisissant la projection de v dans la bonne arête.

□

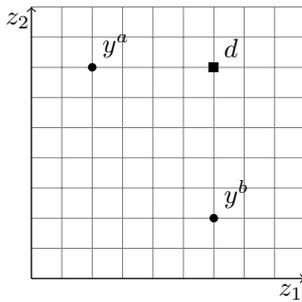


Figure 3.4 – y^a et y^b ne sont pas connectés et ne se dominent pas.

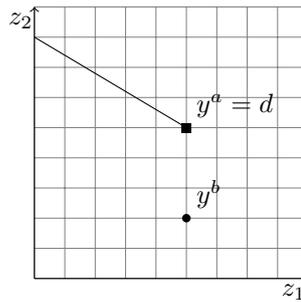


Figure 3.5 – y^a est faiblement dominé par y^b et ils ne sont pas connectés.

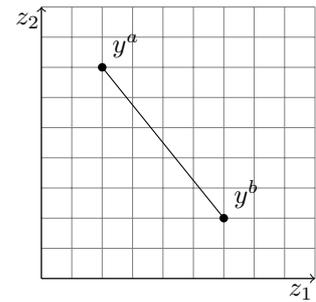


Figure 3.6 – y^a et y^b sont connectés et ne se dominent pas.

Les points délimitants sont utilisés pour calculer et mettre à jour la borne β qui est utilisée comme critère d’arrêt de la méthode de ranking. En effet, la valeur de cette borne est déterminée au moyen du point délimitant ayant la plus grande valeur pondérée. Puisqu’il peut exister une infinité de points délimitants dans la condition (iii) de la proposition 3.2.1, il est important de pouvoir écarter ce cas critique lors du calcul de β .

Proposition 3.2.2. *Les points délimitant décrits par la condition (iii) de la proposition 3.2.1 n'interviennent pas dans le calcul de β .*

Démonstration.

Montrons que ces points délimitants ne servent pas au calcul de β . Pour montrer qu'ils ne sont pas nécessaires, il suffit de montrer qu'aucun d'eux ne peut être le point délimitant ayant la plus grande valeur pondérée au sein du triangle $\delta(y^r, y^s)$. Supposons donc que d soit un point délimitant appartenant à l'arête $]y^a y^b[$ et corresponde à la description faite dans la condition (iii) de la proposition 3.2.1. Deux cas sont possibles concernant $]y^a y^b[$:

– $]y^a y^b[$ n'est pas parallèle à l'hypoténuse du triangle courant. On a donc $z^\lambda(y^a) > z^\lambda(d) > z^\lambda(y^b)$ ou $z^\lambda(y^a) < z^\lambda(d) < z^\lambda(y^b)$. Sans perte de généralité, considérons que nous sommes dans le premier cas.

Dans cette situation, il est suffisant de considérer $z^\lambda(y^a)$ si ce point est délimitant. Autrement, y^a définit nécessairement un autre point délimitant d^2 tel que $z^\lambda(d^2) > z^\lambda(y^a) > z^\lambda(d)$ et il faut donc considérer d^2 .

– $]y^a y^b[$ est parallèle à l'hypoténuse du triangle et on a donc $z^\lambda(y^a) = z^\lambda(d) = z^\lambda(y^b)$. Il est donc suffisant de considérer $z^\lambda(y^a)$ ou $z^\lambda(y^b)$.

En conclusion, il existe nécessairement un autre point délimitant que d ayant une plus grande valeur pondérée, ou bien un point ayant la même valeur mais étant déjà connu (une extrémité d'arête). Par conséquent, il est inutile de tenir compte des points délimitant correspondant au cas (iii) de la proposition 3.2.1 lors du calcul de β .

□

3.2.3 Particularités du contexte mixte

Une autre différence est due au fait que les points obtenus lors de cette deuxième phase peuvent être supprimés. En effet, contrairement au cas des problèmes discrets, il n'est pas garanti que les points ajoutés à l'ensemble non dominé soient effectivement non dominés. Ces points sont obtenus lors de la résolution d'un MOLP et certains d'entre eux seront nécessairement dominés lorsque de nouveaux MOLP seront résolus. Cela se produit par exemple sur la figure 3.3, dans laquelle le MOLP obtenu à partir de y^1 est partiellement dominé et supprimé après avoir été ajouté à l'ensemble des solutions. De plus, il n'est pas garanti qu'un MOLP énuméré soit conservé même partiellement, comme le montre l'exemple suivant.

Exemple 3.2.1 – Un exemple d'exécution du ranking est donné par la figure 3.7. Lorsque la troisième meilleure solution x^3 est obtenue, le point correspondant y^3 et tout le MOLP associé sont calculés et utilisés pour mettre à jour l'ensemble non dominé. Dans la figure 3.7(a), bien que y^3 soit dominé, une partie du MOLP auquel il appartient est ajoutée à l'ensemble non dominé. La figure 3.7(b) correspond à l'étape suivante au cours de laquelle y^4 et son MOLP sont ajoutés. Le MOLP de y^4 domine toute la partie restante du MOLP de y^3 . Cette partie est donc supprimée de l'ensemble non dominé.

Grâce à l'exemple précédent, on comprend que le MOLP auquel appartient y n'est pas conservé car il est dominé de deux façons :

- une partie comprenant y^3 , le point ayant la valeur pondérée la plus faible, est dominé dès l'énumération du MOLP ;

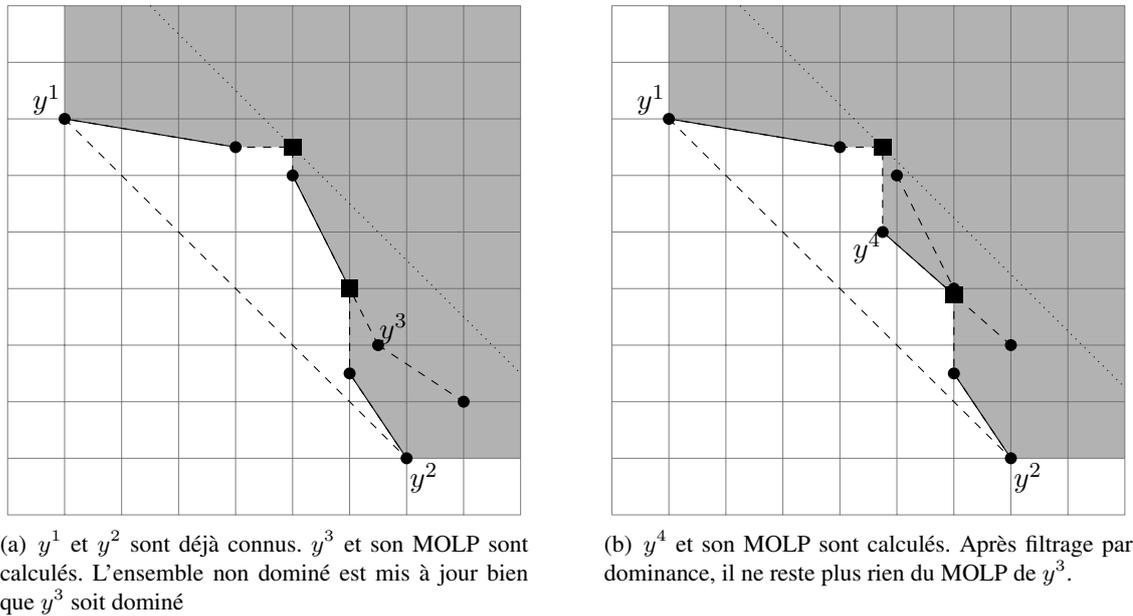


Figure 3.7 – Tous les MOP énumérés par la méthode de ranking ne sont pas nécessairement conservés, même partiellement.

– le reste est dominé par les MOP énumérés par la suite.

Par conséquent, le seul point garanti d'être non dominé est y^k quand il n'est pas dominé dès l'énumération de son MOP.

3.3 Proposition d'une nouvelle méthode pour la phase 2

La méthode de ranking est très utilisée pour visiter les triangles lors de la deuxième phase. Son intérêt réside dans le fait qu'elle fournit des points, du plus proche de l'hypoténuse au plus éloigné, jusqu'à ce qu'une borne soit atteinte. Il est en effet attendu que les points *proches* de l'hypoténuse soient plus probablement non dominés que les points qui en sont *éloignés*. Cette observation est d'ailleurs vraie suffisamment souvent pour rendre la méthode compétitive. Malheureusement, la méthode du ranking souffre de certains écueils connus.

3.3.1 Problèmes potentiels de la méthode de ranking

Les résolutions mono-objectif sur lesquelles reposent le ranking sont coûteuses et nombreuses. Il faut en effet résoudre un MILP par partition considérée. Puisque les problèmes étudiés sont sans structure particulière, il n'est pas envisageable d'accélérer la résolution des MILP mono-objectifs. Le seul levier d'amélioration est donc le nombre de résolutions à effectuer. Pourtant, certaines situations peuvent provoquer un nombre important de résolutions dans le cas du ranking.

Tout d'abord, il est possible que le ranking énumère beaucoup de solutions dominées bien qu'elles soient proches de l'hypoténuse (figure 3.8). Dans cette situation, les solutions obtenues ne permettent pas de mettre à jour les points délimitants. Cela peut par exemple se produire quand

tous les points issus d'une partition \mathcal{P}_k sont dominés mais restent intéressants au regard de la valeur de leur z^λ . Le ranking va alors énumérer de nombreux points dominés provenant de \mathcal{P}_k et de ses partitions avant de poursuivre son exploration dans d'autres partitions moins prometteuses mais contenant pourtant des points non dominés.

La seconde situation pouvant entraîner des problèmes vient du fait que la rapidité de l'énumération dépend de la largeur de la bande visitée. En effet, le ranking risque d'énumérer des points en dehors du triangle et une bande plus large augmente potentiellement le nombre de points externes énumérés (figure 3.9). La méthode souffre donc d'explorations inutiles en dehors du triangle considéré.

En comparaison, le branch&bound est moins susceptible de tomber dans ce type de piège grâce au test de sondage effectué sur chacun des nœuds qu'il visite dans son arbre de recherche. Ainsi, cette méthode évite plus efficacement les solutions dominées (en supposant que les ensembles bornant utilisés soient suffisamment bons) et nous avons vu qu'il est possible de d'utiliser des contraintes lors des sondages pour limiter l'exploration au triangle d'intérêt.

Par conséquent, il semble intéressant de combiner les deux méthodes afin de tirer parti de leurs avantages respectifs :

- d'un côté, la méthode de ranking accomplit une recherche très agressive des points non dominés, ce qui est généralement payant ;
- de l'autre côté, la méthode de branch&bound est capable d'éviter les situations qui peuvent considérablement ralentir le ranking.

Cette combinaison est appelée *Rank & Bound* dans la suite de cette thèse.

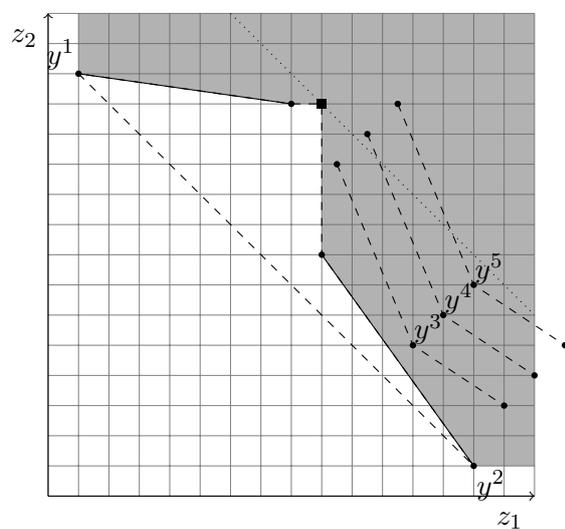


Figure 3.8 – La méthode de ranking ne dispose pas de mécanisme permettant de ne pas considérer les partitions qui ne mènent qu'à des points dominés.

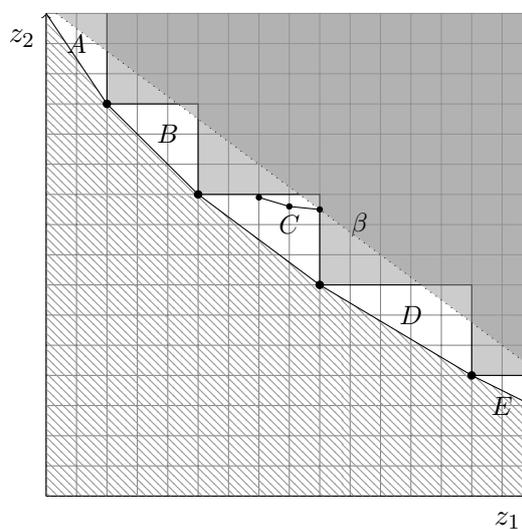


Figure 3.9 – Lors de l'exploration du triangle C , toute la région se trouvant sous la bande délimitée par β est visitée. Si cette bande reste élevée, de nombreux autres points risquent d'être énumérés.

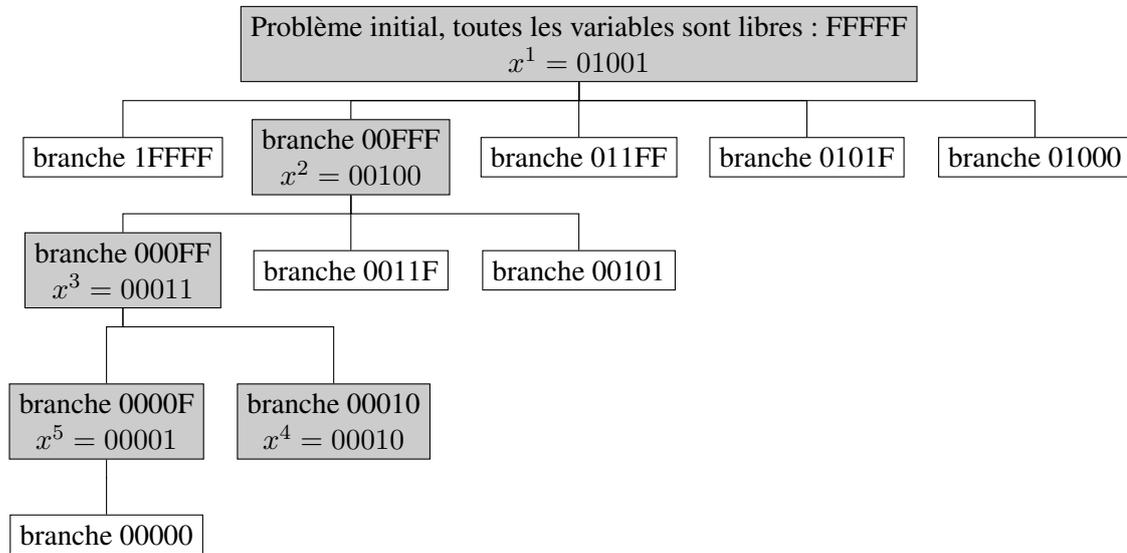


Figure 3.10 – Exemple d’arbre de partitionnement correspondant à l’exemple de la figure 3.8.

3.3.2 Principe de fonctionnement de rank&bound

L’objectif que nous nous proposons donc de remplir est de conserver la méthode de ranking en y ajoutant un mécanisme, inspiré du sondage et des contraintes additionnelles utilisés par le branch&bound, qui lui permettra d’éviter les pièges évoqués dans la section précédente.

Le sondage par dominance est déjà utilisé lorsque la k -ème meilleure solution (combinaison de variables binaires) x_k a été trouvée et qu’il faut mettre à jour l’ensemble non dominé :

- l’ensemble bornant supérieurement est l’ensemble des points déjà connus, comme pour le branch&bound ;
- l’ensemble bornant inférieurement, quant à lui, est tiré du MOLP correspondant aux variables binaires de x_k .

Le problème représenté par la solution x_k correspond en effet à la feuille d’un arbre de recherche de branch&bound puisque toutes les variables binaires y sont fixées, ce qui donne bien un MOLP.

En revanche, quand une solution x_k mène à un MOLP entièrement dominé, la partition \mathcal{P} dont x_k est issue est testée afin de déterminer si tous les MOLP qu’elle représente seront également dominés. Pour cela, nous considérons le sous-problème représenté par la partition \mathcal{P} , c’est-à-dire le sous-problème obtenu en fixant les variables telles qu’elles sont dans \mathcal{P} . Nous appliquons ensuite le test de sondage par dominance à ce sous-problème, qui correspond donc à un nœud intermédiaire d’un arbre de recherche de Branch&Bound. Si la relaxation du sous-problème courant est dominée (ou en dehors du triangle), alors tout point obtenu à partir de la partition \mathcal{P} sera inintéressant. Quelle que soit la qualité des points au regard de leur valeur pondérée, ils pourront donc être sondés et ne seront donc plus considérés par la suite. De même, si tous les points du sous-problème sont en dehors du triangle courant, les contraintes ajoutées au sondage par dominance permettront de le détecter et d’éviter des énumérations inutiles.

3.4 Résultats expérimentaux

Pour tester et comparer les différentes méthodes en deux phases, les instances utilisées ont été générées suivant la même procédure que celle présentée dans la section 2.5. Par ailleurs, afin de tester des instances plus difficiles, nous avons également généré des instances dont les coefficients des fonctions objectif sont négativement corrélés. Ces instances sont donc plus réalistes puisque les objectifs y sont pleinement conflictuels.

Le tableau 3.1 présente les temps moyens de résolutions pour les problèmes dont les objectifs ont été générés indépendamment. Le tableau 3.2 présente quant à lui les temps pour les problèmes dont les objectifs sont corrélés.

Dans chaque tableau figurent les temps pour cinq méthodes de résolution :

- le Branch & Bound présenté dans la section 2.4 et bénéficiant de toutes les améliorations proposées. Cette méthode servira de référence (noté B&B) ;
- la méthode en deux phases utilisant le Branch & Bound avec la procédure de branchement $e(k)$ dans sa deuxième phase (noté 2P B&B($e(k)$)) ;
- la méthode en deux phases utilisant le Branch & Bound avec la procédure de branchement $e_\lambda(k)$ dans sa deuxième phase (noté 2P B&B($e'_\lambda(k)$)) ;
- la méthode en deux phases utilisant le ranking dans sa deuxième phase (noté 2P Ranking) ;
- la méthode en deux phases utilisant le Rank & Bound dans sa deuxième phase (noté 2P R&B).

| Taille | B&B | 2P B&B($e(k)$) | 2P B&B($e_\lambda(k)$) | 2P Ranking | 2P R&B |
|---------------|-------|------------------|--------------------------|------------|--------|
| 30 × 30(15) | 0,1 | 0,6 | 0,5 | 0,5 | 0,5 |
| 40 × 40(20) | 0,3 | 1,0 | 0,9 | 1,0 | 0,8 |
| 50 × 50(25) | 1,4 | 2,8 | 1,6 | 2,5 | 1,8 |
| 60 × 60(30) | 6,9 | 11,3 | 6,2 | 7,1 | 5,2 |
| 70 × 70(35) | 12,3 | 20,1 | 12,1 | 14,0 | 11,3 |
| 80 × 80(40) | 47,8 | 76,9 | 37,4 | 40,4 | 31,1 |
| 90 × 90(45) | 70,6 | 118,6 | 55,8 | 65,5 | 52,9 |
| 100 × 100(50) | 185,2 | 351,4 | 163,3 | 189,5 | 165,4 |
| 110 × 110(55) | - | - | 277,9 | 266,5 | 235,6 |

Table 3.1 – Comparaison des temps moyens de résolution pour les instances non corrélées (en secondes) de l’algorithme de Branch & Bound global et de la méthode en deux phases utilisant localement le Branch & Bound, le ranking et la méthode Rank & Bound.

3.4.1 Remarques générales

Avant de comparer les différentes méthodes entre elles, il est possible de faire plusieurs remarques d’ordre général.

| Taille | B&B | 2P B&B($e(k)$) | 2P B&B($e_\lambda(k)$) | 2P Ranking | 2P R&B |
|---------------|-------|------------------|--------------------------|------------|--------|
| 30 × 30(15) | 0,8 | 1,4 | 0,9 | 0,9 | 1 |
| 40 × 40(20) | 2,1 | 3,7 | 1,7 | 2,3 | 1,7 |
| 50 × 50(25) | 6,0 | 8,7 | 4,2 | 5,8 | 4,2 |
| 60 × 60(30) | 25,0 | 31,8 | 10,3 | 13,6 | 11 |
| 70 × 70(35) | 102,8 | 139,0 | 38,0 | 37,1 | 31,9 |
| 80 × 80(40) | 191,9 | 270,8 | 96,3 | 75,6 | 66,8 |
| 90 × 90(45) | - | - | 248,6 | 192,6 | 176,2 |
| 100 × 100(50) | - | - | 351,3 | 378,4 | 336,4 |

Table 3.2 – Comparaison des temps moyens de résolution pour les instances corrélées (en secondes) de l’algorithme de Branch & Bound global et de la méthode en deux phases utilisant localement le Branch & Bound, le ranking et la méthode Rank & Bound.

Tout d’abord, on remarque que les temps de résolution par la méthode de branch&bound global sont sensiblement plus courts (réduits d’environ 30 %) que dans les résultats de la section 2.5. Cela est dû au passage de de la version 32 bits du système d’exploitation à la version 64 bits sur la machine utilisée pour mener les tests. La version 32 bits étant générique et prenant en charge un grand nombre de processeurs (parfois anciens), les bibliothèques utilisées ne tirent pas parti des instructions les plus évoluées (et efficaces), ce qui n’est pas le cas pour la version 64 bits.

Ensuite, on peut remarquer, comme cela était attendu, que les temps de résolution sont nettement plus élevés lorsque les problèmes traités ont des fonctions objectifs négativement corrélés. Pour les instances de taille 100 × 100(50), les différentes méthodes mettent en effet environ deux fois plus de temps pour terminer leur résolution, et cet écart augmente avec la taille des instances.

3.4.2 Résultats de la méthode de Branch&Bound globale

Cette méthode de résolution ayant déjà fait l’objet d’expérimentations dans le chapitre précédent, nous nous contentons ici de comparer ses performances en fonction du type d’instance résolu. Comme cela a déjà été indiqué, la vitesse de résolution se dégrade fortement quand on passe d’objectifs indépendants à objectifs corrélés négativement. En effet, le temps moyen de résolution pour les instances de taille 80 × 80(40) est de 47,8 s dans le tableau 3.1 et de 191,9 s dans le tableau 3.2, soit un facteur de plus de 4.

Cette importante différence peut s’expliquer par le fait que la procédure de cheminement proposée dans la section 2.4.3 devient beaucoup moins efficace dans un contexte où les coefficients des objectifs sont corrélés. Par conséquent, la visite de l’arbre de recherche est moins bien guidée, ce qui ralentit la résolution.

Les cases vides (marquées d’un «-») correspondent à des tailles d’instances pour lesquelles la méthode n’est pas parvenue à résoudre les dix instances. Cela est dû à des instabilités numériques causées par le détournement de bibliothèques de PL lors de la résolution des BOLP.

3.4.3 Résultats de la méthode en deux phases avec Branch&Bound

Deux méthodes utilisant le branch&bound dans la deuxième phase ont été comparées : l'une utilisant la stratégie de branchement basée sur le coefficient d'efficacité globale $e(k)$, l'autre utilisant la stratégie basée sur $e_\lambda(k)$, le coefficient d'efficacité locale. La comparaison entre les deux méthodes est clairement en la défaveur de la méthode se basant sur $e(k)$. En effet, les temps de résolution de cette méthode sont toujours les plus lents, toutes méthodes confondues, avec des écarts parfois importants (du simple au double entre 2P B&B($e_\lambda(k)$) et 2P B&B($e(k)$) puisqu'on passe de 163,3 s à 351,4 s avec les instances non corrélées de taille $100 \times 100(50)$). Ces résultats confirment donc l'importance d'utiliser un coefficient d'efficacité local s'appuyant sur les spécificités de chaque triangle visité.

Nous abordons maintenant l'efficacité de la méthode 2P B&B($e_\lambda(k)$).

En comparant dans le tableau 3.1 les temps de résolution de la méthode en deux phases utilisant le Branch&Bound et du Branch&Bound global, on constate que le Branch&Bound global est plus rapide sur les petites instances (de taille inférieure à $50 \times 50(25)$) mais la tendance s'inverse quand les instances grandissent. En effet, sur les instances de petite taille, la méthode en deux phases est ralentie par des opérations redondantes. Elle doit effectivement explorer un arbre de recherche par triangle, et certains nœuds équivalents risquent d'être visités dans plusieurs arbres différents, si ce n'est tous. De son côté, le branch&bound global ne visite qu'un seul arbre de recherche, ce qui s'avère moins coûteux.

En revanche, lorsque les instances sont de plus grande taille, la méthode en deux phases réagit mieux puisque l'exploration de l'arbre de chaque triangle se fait de manière spécialisée. Bien que le nombre de triangles à visiter soit également plus important, les arbres correspondants, du fait de l'utilisation d'une procédure de cheminement locale, se différencient davantage. Il y a alors moins de nœuds équivalents visités et cette spécialisation devient rentable pour la méthode en deux phases.

Dans le cas des instances négativement corrélées du tableau 3.2, le même phénomène se produit mais le basculement a lieu bien plus tôt, dès les instances de taille $30 \times 30(15)$. Par ailleurs, les temps de résolutions croient toujours exponentiellement, mais la version globale de la méthode de Branch&Bound est la plus affectée à cause de sa procédure de séparation moins efficace. La méthode en deux phases tire quant à elle parti des espaces de recherches restreints.

3.4.4 Résultats de la méthode en deux phases avec Ranking

D'après le tableau 3.1, la méthode en deux phases avec ranking est du même ordre que le Branch&Bound global quand les instances à résoudre ont des objectifs générés indépendamment. Cette méthode s'avère donc plus lente que l'autre méthode en deux phases utilisant un Branch&Bound local.

Deux éléments peuvent expliquer cette relative contreperformance. Tout d'abord, des sources potentielles de ralentissement ont été évoquées dans la section 3.3.1 : certaines situations peuvent conduire à énumérer beaucoup de solutions dominées. Par ailleurs, un autre facteur explicatif est l'utilisation d'un solveur MILP générique pour résoudre les différents problèmes de somme pondérée. Le solveur utilisé ne peut exploiter de structure particulière pour accélérer la résolution. Or, puisque de nombreux MILP doivent être résolus dans chaque triangle, le temps global de résolution s'en retrouve affecté, ce qui réduit l'intérêt du ranking pour ce genre de problème.

En revanche, cette méthode réagit beaucoup mieux aux instances ayant des objectifs corrélés. À partir de la taille $70 \times 70(35)$ jusqu'à la taille $100 \times 100(50)$, la méthode en deux phases avec ranking devient en effet plus rapide que les deux autres méthodes précédemment analysées.

3.4.5 Résultats de la méthode en deux phases avec Rank&Bound

Mis à part pour les instances de petite taille ($30 \times 30(15)$ exclusivement), l'utilisation de Rank&Bound s'avère plus efficace que celle du ranking classique. En revanche, cette amélioration n'augmente que très peu avec la taille des instances et son ratio semble tendre vers 1.14. Ce ratio constant pose question puisque les bénéfices du sondage renforcé ne sont a priori pas fixes.

Parmi toutes les méthodes testées, celle-ci est la plus rapide à partir de la taille $40 \times 40(20)$, quelle que soit la nature des fonctions objectif.

On peut donc en déduire que combiner le meilleur du Branch&Bound et du ranking est une stratégie payante.

Dans le but d'améliorer encore cette nouvelle méthode, il pourrait être intéressant d'étudier l'impact des sondages en faisant varier leur fréquence. Pour le moment, le sondage est effectivement utilisé dès qu'un MOLP dominé ou en dehors du triangle est obtenu, mais on pourrait envisager de ne l'appliquer que quand la partition considérée est d'une taille suffisamment petite.

Enfin, il serait possible de s'inspirer du coefficient d'efficacité $e(k)$ pour réduire le nombre de partitions générées et évaluées. En effet, puisqu'à chaque étape de l'algorithme, la partition d'où est originaire l'actuelle meilleure solution est redécoupée en autant de partitions qu'il reste de variables libres, il serait intéressant de placer ces solutions dans les partitions ayant le moins possibles de variables libres. De cette façon, le nombre de partitions à évaluer, et donc de MILP à résoudre, serait réduit. Cette proposition est bien sûr également valable pour la méthode de ranking.

3.5 Conclusion

Dans ce chapitre, nous avons étudié l'application de la méthode en deux phases aux programmes biobjectifs en variables mixtes. Déjà réputée dans le contexte des MOCO, cette méthode a également montré son intérêt dans le cadre qui nous intéresse.

Pour pouvoir l'appliquer, nous avons dû généraliser certaines méthodes au contexte mixte. Tout d'abord, nous avons vu dans le chapitre précédent que la première phase pouvait être améliorée en utilisant l'information fournie par les points de Y_{SN} pour résoudre certains MOLP de façon préliminaire et ainsi renforcer l'ensemble bornant supérieurement initial.

Ensuite, deux stratégies connues ont été appliquées dans la seconde phase. La première méthode consistait simplement en l'utilisation locale de l'algorithme de Branch & Bound présenté dans le chapitre précédent et qu'il a fallu adapter afin de restreindre les zones de recherches. La seconde méthode est celle du ranking, généralement très efficace dans le domaine des MOCO. Pour pouvoir appliquer cette dernière, il nous a fallu utiliser la définition des points délimitants afin de pouvoir calculer des bornes serrées. Le cadre mixte a néanmoins nécessité d'introduire une caractérisation de ces points délimitants pour couvrir les cas particuliers présents.

Enfin, nos observations sur les forces et faiblesses des deux méthodes utilisées dans la seconde phase nous ont inspiré une troisième méthode, originale, visant à combiner les points forts de chacune. D'après nos résultats expérimentaux, cette dernière méthode s'avère la plus efficace.

Une question naturelle à cette étape concerne les problèmes à plus de deux objectifs. La transition de deux à trois fonctions objectifs est en effet connue pour être ardue sur plusieurs aspects. Le chapitre suivant s'attache à cette problématique en proposant pour la première fois un algorithme permettant de résoudre un MOMILP ayant trois objectifs.

Résolution de problèmes à trois objectifs

Dans ce chapitre, nous abordons la résolution de problèmes à trois objectifs. Nous allons dans un premier temps détailler les difficultés inhérentes à ce contexte, qu'elles proviennent des problèmes à trois objectifs en général ou du cadre mixte auquel nous nous intéressons. Par la suite, nous établirons une caractérisation des éléments dominés et non dominés en procédant par analogie avec le contexte biobjectif. Cela nous permettra de proposer une procédure de mise à jour de l'ensemble non dominé, puis d'adapter certaines méthodes de résolution étudiées dans les chapitres précédents. En particulier, un grand détail sera apporté sur la description de la zone de recherche lors de la seconde phase de la méthode en deux phases. Enfin, les différentes méthodes seront comparées expérimentalement. Les résultats nous permettront de conclure et de proposer des perspectives d'amélioration. Ces travaux ont fait l'objet de communications en conférences internationales [Vincent et al., 2013b, Vincent et al., 2013c] et seront prochainement soumis dans une revue scientifique.

| | | |
|------------|--|------------|
| 4.1 | Difficultés attendues | 83 |
| 4.1.1 | Difficultés inhérentes au contexte mixte | 83 |
| 4.1.2 | Réduction à un seul type de facette | 84 |
| 4.2 | Caractérisation des points dominés et non dominés | 86 |
| 4.3 | Méthode de mise à jour des ensembles non dominés | 87 |
| 4.4 | Adaptation des composants du Branch & Bound | 88 |
| 4.4.1 | Ensembles bornant | 88 |
| 4.4.2 | Procédure de cheminement | 88 |
| 4.5 | Méthode en deux phases | 89 |
| 4.5.1 | Adaptation de la première phase au contexte mixte | 89 |
| 4.5.2 | Deuxième phase | 89 |
| 4.6 | Résultats expérimentaux | 99 |
| 4.6.1 | Observations générales | 100 |
| 4.6.2 | Comparaison des temps de résolutions | 100 |
| 4.6.3 | Potentiels facteurs d'amélioration | 100 |
| 4.7 | Conclusion | 101 |

4.1 Difficultés attendues

Plusieurs difficultés sont à attendre lors de la résolution de problèmes multiobjectifs en variables mixtes. Nous les distinguons en deux catégories.

Tout d’abord, certaines difficultés sont propres au contexte multiobjectif dans le sens où elles sont communes à tous les problèmes multiobjectifs. Celles-ci ont notamment déjà été abordées dans la section 1.4.3, où les différences entre la méthode en deux phases à deux et trois objectifs ont été discutées. La difficulté la plus importante est la perte de l’ordre naturel des points non dominés, ce qui rend caduque la plupart des économies dans les comparaisons paire à paire d’éléments, d’exploration de l’espace de recherche, etc. En particulier, une optimisation importante dans la comparaison des points devient impossible. La comparaison paire à paire peut en effet être fortement restreinte dans le contexte biobjectif, comme évoqué dans la remarque 2.3.4. Or, à partir de trois fonctions objectif, ce raccourci n’est plus accessible, ce qui impose de réaliser toutes les comparaisons possibles entre deux ensembles de facettes.

Ensuite, d’autres particularités n’apparaissent que dans le contexte des problèmes en variables mixtes. Nous abordons maintenant plus en détail ces difficultés.

4.1.1 Difficultés inhérentes au contexte mixte

Nous savons par la proposition 2.2.4 que l’ensemble non dominé d’un BOMILP est composé de points et d’arêtes. Intuitivement, cela peut s’expliquer par le fait qu’il s’agit d’une union de MOLP filtrés par dominance. Puisque l’ensemble non dominé des MOLP est composé d’arêtes dans le contexte biobjectif, ces arêtes se retrouvent naturellement dans l’ensemble non dominé lors de la résolution des BOMILP.

Plus précisément, lorsque le problème compte p objectifs, les ensembles non dominés des MOLP sont composés de faces de dimension inférieure ou égale à $p - 1$. Pour $p = 2$, on obtient donc des facettes de dimension $p - 1 = 1$, c’est-à-dire les arêtes que nous avons observées. En revanche, pour $p = 3$, ces facettes ont une dimension $p - 1 = 2$ mais on peut avoir des faces de dimension 1 et 2. On aura donc des points, des arêtes et des facettes.

Puisque l’utilisation explicite d’arêtes nécessitait une procédure de mise à jour appropriée dans le contexte biobjectif, il en sera de même pour les arêtes et les facettes dans le contexte triobjectif. De plus, l’apparition d’éléments de dimension 2 engendre de nouvelles difficultés.

Tout d’abord, ces facettes peuvent être de plusieurs formes et contenir différents nombres de points extrêmes. Là où les arêtes n’étaient représentées que par leurs deux points extrêmes, chaque facette doit être représentée par ses $k \geq 3$ points extrêmes. Bien que ce nombre indéterminé de points extrêmes ne soit pas un problème en théorie, il peut s’avérer complexe à gérer en pratique. Par ailleurs, bien que les facettes soient convexes, il se peut que, par le fait du filtrage par dominance, leur partie non dominée soit non convexe. Cela peut par exemple se produire quand seule une partie de l’intérieur relatif d’une facette est dominée. Il convient donc de déterminer comment représenter ces parties non dominées non convexes.

Exemple 4.1.1 – Soit F une facette et y un point dominant une partie de F , tels que représentés par la figure 4.1. y ne domine qu’une partie de l’intérieur relatif de F .

Ces particularités rendent a priori plus difficiles la gestion et la mise à jour des ensembles non dominés. Nous commençons donc par proposer une importante simplification visant à uniformiser la caractérisation des facettes non dominées.

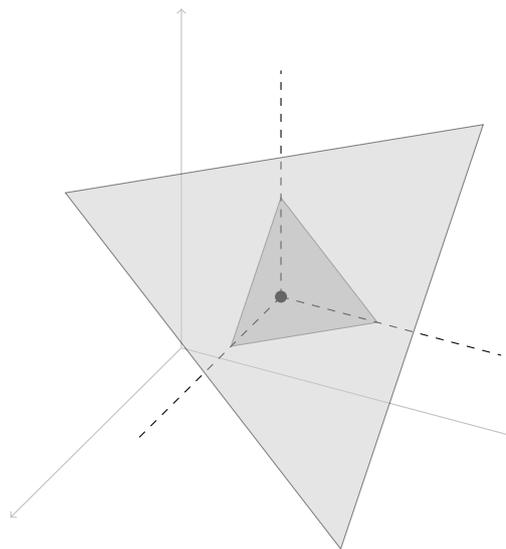


Figure 4.1 – Dominance problématique, seule une partie intérieure de la facette est dominée.

4.1.2 Réduction à un seul type de facette

Afin d'éviter les problèmes de représentation des facettes que nous venons d'aborder, nous proposons de recourir à la triangulation.

Définition 4.1.1 (Triangulation [De Loera *et al.*, 2010]). Soit A un ensemble fini de points. Une triangulation T de A est un ensemble de simplexes dont les sommets sont dans A et vérifiant les propriétés suivantes :

- (i) fermeture : toutes les faces des simplexes de T sont dans T ;
- (ii) intersection : l'intersection de deux simplexes de T est une face (potentiellement vide) des deux ;
- (iii) union : l'union de tous ces simplexes est égale à $\text{conv } A$.

La définition de la triangulation peut être renforcée pour définir la triangulation complète.

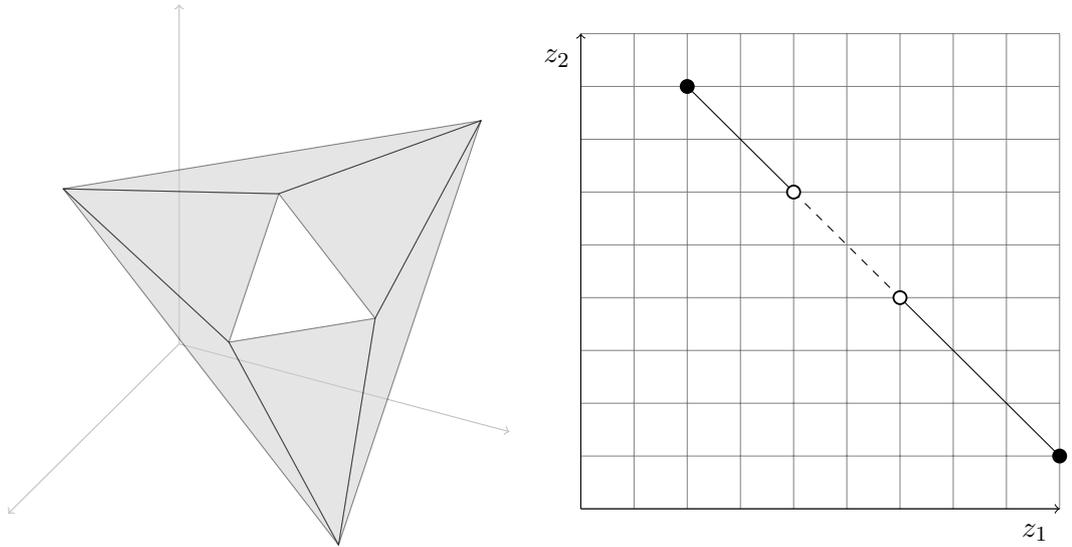
Définition 4.1.2 (Triangulation complète [De Loera *et al.*, 2010]). Soit A un ensemble fini de points. Une triangulation T de A est dite *complète* si elle utilise tous les points de A .

Dans la suite de ce manuscrit, nous n'utiliserons que la triangulation complète et ne ferons donc pas de distinction entre les deux termes.

Théorème 4.1.1 ([De Loera *et al.*, 2010]). *Toute configuration finie de points admet des triangulations.*

Grâce au théorème 4.1.1, nous savons que toute facette considérée peut être remplacée par la triangulation de ses points extrêmes. Il nous est donc possible de ramener tout type de facette à un ensemble de triangles.

En plus de simplifier les facettes ayant plus de trois points extrêmes, cela nous permet de caractériser les facettes problématiques ayant une partie non dominée non convexe par un ensemble



(a) Dans le contexte triobjectif, une facette dominée en son intérieur relatif peut être divisée en facettes triangulaires. (b) Dans le contexte biobjectif, une arête dominée en son intérieur relatif est naturellement divisée en deux sous-arêtes.

Figure 4.2 – La triangulation dans le contexte triobjectif est comparable au découpage des arêtes dans le contexte biobjectif.

de triangles, c'est-à-dire des éléments convexes. En effet, il suffit dans ce cas de retirer de la triangulation les triangles appartenant à la partie dominée de la facette.

Cette simplification est à rapprocher de celle qui est implicitement faite dans le cadre biobjectif. Les arêtes dominées en leur intérieur y sont en effet séparées en deux sous-arêtes non dominées alors qu'il s'agit d'une arête dont la partie non dominée est non convexe. Ainsi, le recours à la triangulation (figure 4.2(a)) peut être considéré comme l'équivalent de la séparation des sous-arêtes (figure 4.2(b)).

Ainsi, sans perte de généralité, nous supposons par la suite que toutes les facettes considérées sont des triangles.

En contrepartie de la simplification qu'elle permet dans la description et le traitement des facettes, cette réduction en triangles a également des répercussions non désirées. En effet, la triangulation engendre la création d'un plus grand nombre de facettes que l'on peut déterminer.

Théorème 4.1.2 (Nombre de triangles [De Loera *et al.*, 2010]). *Soit T une triangulation complète d'un ensemble de points $A \subset \mathbb{R}^2$. Soient n le nombre de points de A et n_b le nombre de points de A appartenant à la frontière de $\text{conv } A$. Alors, T a exactement $2n - n_b - 2$ triangles.*

On peut en déduire le nombre de triangles obtenus à partir d'une facette pleine ainsi que d'une facette à trous. En reprenant les mêmes notations que dans le théorème 4.1.2 :

- la triangulation compte $n - 2$ triangles quand A représente une facette convexe sans trou ;
- la triangulation compte n triangles quand A représente une facette à un trou (avec n_b sommets externes et $n - n_b$ sommets internes).

Cela implique donc qu'à chaque étape de mise à jour, le nombre d'éléments à comparer augmentera avec chaque triangulation. Or, si ce phénomène n'était pas sensible dans le contexte biobjectif, il est à attendre qu'il le soit beaucoup plus ici, du fait de l'impossibilité d'éviter les compa-

raisons inutiles que nous évoquions dans la sous-section précédente. Par conséquent, il nous faudra observer attentivement les effets de cette uniformisation pour qu'elle ne s'avère pas un handicap.

4.2 Caractérisation des points dominés et non dominés

De même que dans le cas biobjectif, il nous est nécessaire de maintenir l'état complet et exact de l'ensemble des points potentiellement non dominés. Pour cela, nous avons donc besoin de caractériser les points dominés et non dominés de chaque facette.

Remarque 4.2.1 – Dans la suite de cette section, nous considérons que toutes les facettes sont réduites à des triangles. Chaque triangle peut être fermé, ouvert, partiellement ouvert, mais aussi réduit à une arête (qu'elle soit ouverte, semi-ouverte ou fermé) ou à un point.

Pour caractériser la partie non dominée d'une facette, nous proposons d'étendre le théorème 2.2.3.

Théorème 4.2.1. *Soient I_1 et I_2 des facettes et posons $I := I_1 \cap (I_2 + \mathbb{R}_{\geq}^3)$, le sous-ensemble de I_1 dominé par des points de I_2 . Alors I est soit une facette, soit l'ensemble vide. De plus, $I_1 \setminus I$, le sous-ensemble de I_1 n'étant pas dominé par I_2 , peut être caractérisé par*

$$I_1 \setminus I = \bigcup_{i=1}^k I_{1_i}$$

où $k \geq 0$ et les I_{1_i} sont des facettes.

Démonstration.

Plusieurs cas sont possibles pour $I_1 \setminus I$:

- I fait partie de la frontière de I_1 , alors $I_1 \setminus I$ doit toujours être décrit par les points extrêmes de I_1 avec l'information additionnelle qu'une partie de sa frontière est dominée. Par conséquent, $I_1 \setminus I$ est une facette.
- $I = I_1$, alors $I_1 \setminus I = \emptyset$ et l'affirmation est vraie avec $k = 0$.
- I représente une partie de la frontière de I_1 et une partie de son intérieur relatif. Dans ce cas $I_1 \setminus I$ est un polygone simple qui peut donc être triangulé en k facettes.
- I est une partie de l'intérieur relatif de I_1 . Dans ce cas, $I_1 \setminus I$ est un polygone à trou qui peut également être triangulé en k facettes.

□

À partir de la caractérisation que nous venons de donner, il est immédiat d'adapter l'algorithme 2.1 afin de prendre en compte les problèmes triobjectifs. Il suffit de remplacer les arêtes par des facettes. Étant données une facette I_1 de E et une facette I_2 de Z , nous pouvons appliquer le théorème 4.2.1 à deux reprises pour supprimer la partie de I_1 dominée par I_2 et celle de I_2 dominée par I_1 (lignes 3 à 6 de l'algorithme 4.1). En répétant ce test de dominance pour toutes les paires de facettes de E et Z , nous obtenons l'algorithme 4.1.

Nous pouvons maintenant étendre le théorème 2.2.4 pour caractériser l'ensemble Z dans le contexte triobjectif.

Théorème 4.2.2. *À chaque étape de l'algorithme, Z est composé de facettes, qu'elles soient fermées, partiellement ouvertes, ouvertes ou réduites à une arête ou un point.*

Entrée Ensemble de facettes E donné par un MOLP, ensemble potentiellement non dominé Z

Sortie Ensemble potentiellement non dominé Z mis à jour

```

1: pour chaque  $I_1 \in E$  faire
2:   pour chaque  $I_2 \in Z$  faire
3:      $I \leftarrow I_1 \cap (I_2 + \mathbb{R}_{\geq}^3)$ 
4:      $I_1 \leftarrow I_1 \setminus I$ 
5:      $I \leftarrow I_2 \cap (I_1 + \mathbb{R}_{\geq}^3)$ 
6:      $I_2 \leftarrow I_2 \setminus I$ 
7:   fin pour
8: fin pour
9:  $Z \leftarrow (\bigcup_{I_1 \in E} I_1) \cup (\bigcup_{I_2 \in Z} I_2)$ 

```

Algorithme 4.1: procédure `metaUpdateTri`

Démonstration.

La preuve est similaire à celle du théorème 2.2.4. L'ensemble Z est vide au départ, puis contient les facettes fermées issues du premier MOLP résolu. L'algorithme 4.1 est ensuite utilisé à chaque visite d'une feuille de l'arbre de recherche. En appliquant à chaque étape le théorème 4.2.1, il en découle que Z est composé d'une union finie de facettes à la fin de la mise à jour.

□

Il reste maintenant à déterminer comment appliquer en pratique la comparaison des facettes I_1 et I_2 décrites par les lignes 3 à 6 de l'algorithme 4.1.

4.3 Méthode de mise à jour des ensembles non dominés

Nous détaillons dans cette section comment appliquer la mise à jour de l'ensemble Z , en particulier la comparaison entre deux facettes.

Supposons que deux facettes A et B soient à comparer. La première facette A a y^1 , y^2 et y^3 pour points extrêmes et la seconde B est définie par z^1 , z^2 et z^3 . En s'inspirant du programme linéaire 1.4 proposé par [Mavrotas et Diakoulaki, 2005], nous pouvons définir la partie de B dominée par A au moyen de l'ensemble de contraintes suivant :

$$\begin{aligned}
 \lambda_1 y^1 + \lambda_2 y^2 + \lambda_3 y^3 &\leq \mu_1 z^1 + \mu_2 z^2 + \mu_3 z^3 \\
 \lambda_1 + \lambda_2 + \lambda_3 &= 1 \\
 \mu_1 + \mu_2 + \mu_3 &= 1 \\
 \lambda_i &\geq 0 \quad \forall i \\
 \mu_j &\geq 0 \quad \forall j
 \end{aligned} \tag{4.1}$$

La première contrainte assure qu'un point du plan de A domine faiblement un point du plan de B , tandis que les contraintes restantes garantissent que ces points appartiennent bien à leurs facettes respectives.

Toute relation de dominance d'un point de B par un point de A est représentée par une solution admissible d'un programme linéaire contenant ces contraintes. Par conséquent, en parcourant la frontière de la région admissible de ce programme linéaire, on obtient la frontière de la partie de

B faiblement dominée par A . Quand la région dominée de B est obtenue, il suffit de trianguler la partie non dominée résultante pour obtenir un ensemble de facettes qui remplacera B .

Pour comparer une arête ou un point à une facette, il suffit simplement d'utiliser un modèle similaire ayant moins de variables de décision.

Remarque 4.3.1 – Comme pour le programme linéaire 1.4, cette méthode peut être utilisée quel que soit le nombre d'objectifs. On peut par ailleurs remarquer que le nombre de sommets des facettes comparées importe peu, tant que celles-ci sont convexes.

Nous avons donc étendu la mise à jour de l'ensemble non dominé au cadre triobjectif. Néanmoins, cette mise à jour nécessite de très nombreuses comparaisons du fait de l'absence d'ordre des points non dominés. Là où les comparaisons pouvaient être facilement restreintes dans le cas biobjectif, ces comparaisons sont obligatoires à partir de trois objectifs. À cause du grand nombre de facettes générés par l'utilisation de la triangulation, le nombre de comparaisons risque de devenir un facteur déterminant de l'efficacité des méthodes de résolution.

4.4 Adaptation des composants du Branch & Bound

4.4.1 Ensembles bornant

Pour l'ensemble bornant supérieurement, nous utilisons à nouveau l'ensemble des points potentiellement non dominés muni des informations sur les facettes, à savoir (pour chaque facette) :

- la liste de ses points extrêmes ;
- pour chaque arête extrême, l'information indiquant si elle est dominée ou non, ou les sous-arêtes non dominées.

Pour l'ensemble bornant inférieurement, nous employons à nouveau $(\text{conv } E_L + \mathbb{R}_{\geq}^2)_N$ qui s'était avéré le meilleur compromis dans le cadre biobjectif. Cet ensemble bornant est calculé au moyen de la méthode d'approximation externe de Benson, présentée dans la section 1.3.2.

4.4.2 Procédure de cheminement

Nous étendons le coefficient d'efficacité pour gérer trois objectifs :

$$e(k) := (c_k^1 - \mu_1) + (c_k^2 - \mu_2) + (c_k^3 - \mu_3)$$

où μ_i est la moyenne des coefficients du i ème objectif.

$e(k)$ est utilisé de façon similaire que par rapport au contexte biobjectif, à la fois pour déterminer l'ordre dans lequel fixer les variables binaires et pour indiquer quelle valeur est prioritaire. Néanmoins, on peut s'attendre à ce que l'utilisation de ce coefficient d'efficacité ait un impact moindre dans le cadre triobjectif puisqu'il sera plus rare d'avoir des variables x_k dont les trois coefficients sont du même côté de la moyenne. Les valeurs des $e(k)$ seront donc plus proches les unes des autres, ce qui permettra moins de tirer un ordre clair entre les variables.

4.5 Méthode en deux phases

4.5.1 Adaptation de la première phase au contexte mixte

Pour calculer l'ensemble Y_{SN} lors de la première phase, nous utilisons l'algorithme 1.1 proposé par [Przybylski *et al.*, 2012]. De la même façon que dans le contexte biobjectif, il est possible d'étendre l'ensemble bornant supérieurement obtenu en résolvant le MOLP associé à la combinaison de variables binaires de chacun des points non dominés supportés.

4.5.2 Deuxième phase

Pour le calcul des points non dominés non supportés, nous proposons d'utiliser la procédure de Branch & Bound triobjectif que nous venons de présenter. Pour cela, nous appliquons cette procédure localement à chaque zone définie par un point délimitant. À nouveau, nous restreignons la zone visitée par l'algorithme en procédant de la façon décrite dans la section 3.1.1 pour le cadre biobjectif.

Néanmoins, toute la difficulté réside ici dans la détermination des zones à visiter. Nous nous attachons donc à décrire ces zones de recherche dans la prochaine section.

Description de la zone de recherche

Dans le contexte triobjectif combinatoire, chaque point délimitant définit une zone à explorer dont l'union (non disjointe) constitue la zone de recherche de la deuxième phase. Ces points délimitants sont calculés et mis à jour à partir des points de l'ensemble non dominé.

Aucune méthode n'a pour le moment été proposée pour calculer les points délimitants dans le contexte mixte. Néanmoins, il semble naturel d'appliquer les méthodes dédiées aux problèmes combinatoires afin d'observer d'éventuelles différences dues à la présence d'arêtes et de facettes. C'est ce que nous nous proposons de faire à partir des exemples suivants.

Exemple 4.5.1 – Soient un point délimitant u et une arête réalisable $I = [y^a y^b]$ tels que représentés dans la figure 4.3. Dans cette situation, u est dominé par I . Par conséquent, u n'est pas un point délimitant et il faut donc remplacer ce point par d'autres (les u^i , en rouge), calculés grâce à la méthode introduite par [Przybylski *et al.*, 2010].

Néanmoins, deux différences peuvent être observées :

- puisque I n'est pas réduite à un seul point, la coordonnée de chaque u^i provenant de I ne vient pas nécessairement du même point. Par exemple, u^1 et u^2 prennent une coordonnée de y^a alors que u^3 prend une coordonnée de y^b . En fait, chaque u^i tire simplement sa i ème coordonnée du point de I ayant la plus petite coordonnée sur z^i .
- les u^i ne suffisent pas pour décrire toutes les zones qu'il reste à explorer. Il est donc nécessaire d'ajouter de nouveaux points délimitants (en bleu) : les y^i . Pour ces points, la i ème coordonnée provient de u et les autres proviennent de l'arête I .

Avant de réunir ces observations dans le lemme qui suit, nous définissons les points utilisés, pour $i \in \{1, 2, 3\}$:

- u^i est le point tel que $u_i^i = \min(y_i^a, y_i^b)$ et $u_j^i = u_j$ pour $j \neq i$;
- y^{a_i} est le point tel que $y_i^{a_i} = u_i$ et $y_j^{a_i} = y_j^a$ pour $j \neq i$ (y^{b_i} est défini de la même façon).

De plus, on peut observer que les arêtes $]y^{a_1} y^{b_1}[$ et $]y^{a_2} y^{b_2}[$ sont entièrement composées de points délimitants. Nous introduisons donc la terme d'arête délimitante.

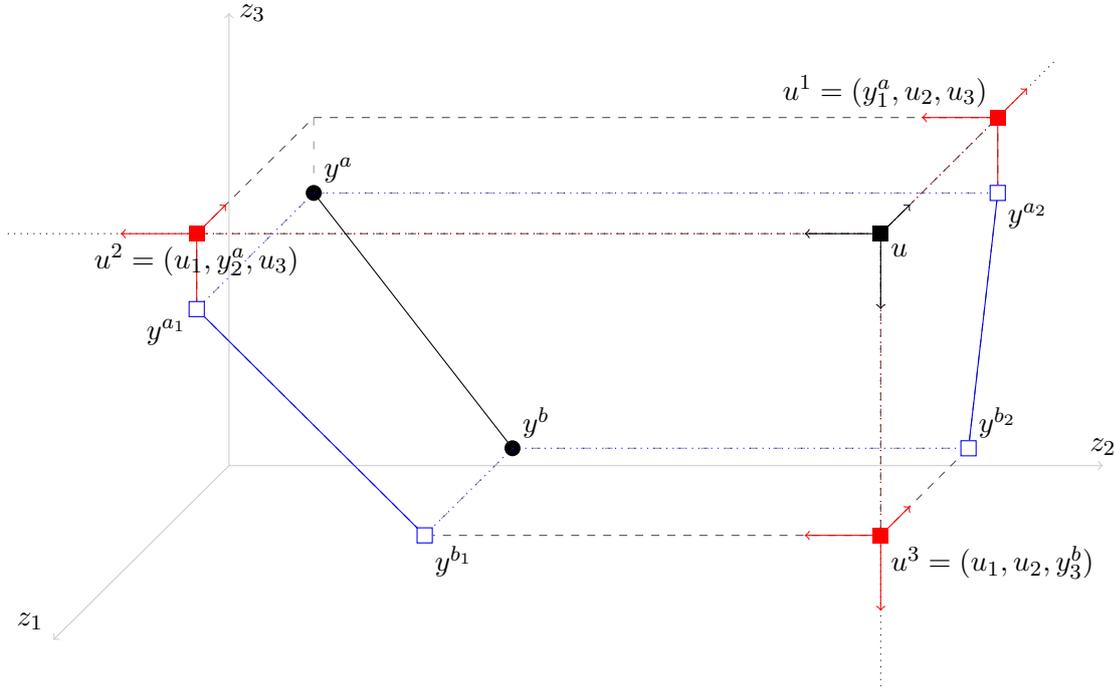


Figure 4.3 – Mise à jour du point délimitant u grâce à l'arête réalisable $[y^a y^b]$.

Définition 4.5.1 (Arête délimitante). Une arête $[y^a y^b]$ est une *arête délimitante* si tout point de son intérieur relatif est un point délimitant.

Lemme 4.5.1. Soient $Q \subset V$, $I = [y^a y^b] \subset V \setminus Q$ et $u \in D(Q \setminus I)$ tels que tout point de I domine strictement u .

- (i) les u^i vérifient la condition (i) de la définition 1.4.5 ;
- (ii) nous considérons les arêtes $[y^{a_1} y^{b_1}]$, $[y^{a_2} y^{b_2}]$ et $[y^{a_3} y^{b_3}]$. Nécessairement, une de ces arêtes ne vérifie pas la condition (i) de la définition 1.4.5, à l'exception d'une de ses extrémités. Les deux autres arêtes vérifient la condition (i) de la définition 1.4.5 ;
- (iii) aucune extrémité des arêtes $[y^{a_i} y^{b_i}]$ ne vérifie la condition (ii) de la définition 1.4.5 ;
- (iv) il n'existe aucun point $v \in (u - \mathbb{R}_{\geq}^3)$ vérifiant la condition (i) de la définition 1.4.5 et dominé par un des u^i ou les arêtes $[y^{a_i} y^{b_i}]$ mentionnées dans (i) et (ii).

Démonstration.

- (i) Par définition, $\forall i \in \{1, 2, 3\}$, $u^i = \min(y_i^a, y_i^b)$ donc pour $\epsilon \in \mathbb{R}_{>}$ et $\forall y \in [y^a y^b]$, $u^i - \epsilon < y_i$. Alors, $u^i - \epsilon$ n'est dominé par aucun point de $[y^a y^b]$.
- (ii) Étant donné que y^a et y^b ne se dominent pas mutuellement, il existe $i \in \{1, 2, 3\}$ tel que soit $y_i^a > y_i^b$ et $y_j^a < y_j^b$ pour $j \neq i$, soit $y_i^a < y_i^b$ et $y_j^a > y_j^b$ pour $j \neq i$. Sans perte de généralité, nous considérons que $i = 3$ et que nous sommes dans le cas où $y_1^a < y_1^b$, $y_2^a < y_2^b$ et $y_3^a > y_3^b$ (tel que sur la figure 4.3).
 - Soit $y \in (y^{a_3} y^{b_3})$, alors $y_j^a < y_j^b$ pour $j \in \{1, 2\}$. De plus, $y_3^a < y_3 = u_3$, donc $y^a < y$ et y ne vérifie pas la condition (i) de la définition 1.4.5. Par conséquent, $(y^{a_3} y^{b_3})$ ne vérifie pas la condition (i) de la définition 1.4.5.

- Nous savons que $y_2^{a3} = y_2^a$, donc y^a ne domine pas strictement y^{a3} . De plus, $y_2^{a3} < y_2$ pour tout point $y \in (y^a y^b]$, donc y ne domine pas strictement y^{a3} . Par conséquent, y^{a3} vérifie la condition (i).
 - Soient $y^{c1} \in [y^{a1} y^{b1}]$ et $\epsilon \in \mathbb{R}_{>}^3$, nous montrons que $y - \epsilon$ n'est dominé par aucun point de $[y^a y^b]$. Soit $y^c \in [y^a y^b]$ tel que $y_2^c = y_2^{c1}$, $y_3^c = y_3^{c1}$ et $y_1^c < y_1^{c1}$. Autrement dit, y^{c1} est une projection de y^c le long de z_1 et $y^c \leq y^{c1}$. Soit également $y \in [y^a y^b] \setminus \{y^c\}$, alors nous avons soit $y_2 > y_2^{c1} (= y_2^c)$, soit $y_3 > y_3^{c1} (= y_3^c)$. Par conséquent, le seul point de $[y^a y^b]$ dominant y^{c1} est y^c et la dominance n'est pas stricte. Ainsi, tout point de $[y^{a1} y^{b1}]$ vérifie la condition (i) de la définition 1.4.5. Une preuve similaire montre que tout point de $[y^{a2} y^{b2}]$ vérifie la condition (i) de la définition 1.4.5.
- (iii) Nous considérons y^{a1} qui vérifie la condition (i) de la définition 1.4.5 d'après le point (ii) et nous utilisons les mêmes hypothèses que précédemment à propos de $[y^a y^b]$. Par définition, $y_2^{a1} = y_2^a$, $y_3^{a1} = y_3^a$ et $y_1^{a1} = u_1 > y_1^a$. Néanmoins, u^2 est tel que $u_2^2 = y_2^a$, $u_3^2 = u_3$, $u_1^2 = u_1$. Alors, $y^{a1} \leq u^2$ et y^{a1} ne vérifient pas la condition (ii) de la définition 1.4.5. Similairement, nous pouvons montrer qu'aucun point y^{a_i} ou y^{b_i} ne peut vérifier la condition (ii) de la définition 1.4.5.
- (iv) Nous prouvons l'affirmation de façon séparée pour les points u^i et pour les arêtes $(y_i^a y_i^b)$.
- Considérons $v^j = u^i + \alpha e_j$ avec $\alpha \in \mathbb{R}_{>}$ et $i, j \in \{1, \dots, 3\}$.
 - Si $i \neq j$, alors $v_i^j > u_i$, donc $v^j \notin (u - \mathbb{R}_{\geq}^3)$.
 - Si $i = j$, alors nous considérons $y \in \{y^a, y^b\}$ tel que $y_i = \min(y_i^a, y_i^b)$. Alors $y < v^j$.
 - Soit $y^{c_i} \in (y^{a_i} y^{b_i})$ et considérons $v^j = y^{c_i} + \alpha e_j$ avec $\alpha \in \mathbb{R}_{>}$ et $j \in \{1, \dots, 3\}$.
 - Si $i = j$, alors $v_i^j > u_i$ donc $v^j \notin (u - \mathbb{R}_{\geq}^3)$.
 - Si $i \neq j$, alors nous considérons $y \in (y^a y^b)$ tel que $y_j = v_j^j$. Alors $\frac{y^c + y}{2} < v^j$.
- Par conséquent, aucun point de $(u - \mathbb{R}_{\geq}^3)$ ne peut vérifier la condition (i) de la définition 1.4.5 tout en étant dominé par un point u^i ou une arête $(y^{a_i} y^{b_i})$.

□

Remarque 4.5.1 – Le lemme 4.5.1 impose dans ses hypothèses que tout point de I domine strictement u . Dans le cas où toute l'arête I ne domine pas strictement u , il suffit de se restreindre à la sous-arête de I dominant u strictement.

Ce type de dominance partielle est naturellement détecté dès que certains points y obtenus à partir du système de contraintes 4.1 ne correspondent pas aux points extrêmes de la facette dominant.

Grâce au lemme 4.5.1, nous savons que $D(U)$ peut comprendre des arêtes délimitantes. Dans chacune de ces arêtes, tous les points ont une coordonnée commune, nous dirons donc que ces arêtes sont *plates*.

Puisqu'un nouveau type d'objet délimitant a été introduit, il est donc nécessaire de comparer une arête délimitante et une arête réalisable.

Exemple 4.5.2 – Soient deux arêtes $I_1 = [y^a y^b]$ et $I_2 = [u^c u^d]$ telles que représentées dans la figure 4.4. I_1 est une arête réalisable et I_2 est une arête délimitante, telle que définie par le lemme 4.5.1. Le calcul des u^i et des y^i se fait de manière similaire à l'exemple précédent, mais on observe tout de même une différence. En effet, il y a à présent une arête délimitante non plate, $[y^{a2} u^{d1}]$.

(v) Aucun point $v \in (u - \mathbb{R}_{\geq}^3)$ ne vérifie la condition (i) de la définition 1.4.5 en étant dominé par les points u^i et les arêtes $]y^{a_i}y^{b_i}[$ des points (i) et (ii).

Démonstration.

- (i) Par définition, $\forall i \in \{1, 2, 3\}$, $u_i^{f_i} = y_i$ avec $y \in \arg \min\{y_i : y \in [y^a y^b] : y \leq u^f\}$. Alors, pour chaque point $y' \in [y^a y^b] \setminus \{y\}$, on a soit $y'_i > y_i$, soit $y'_i < y_i$. Si $y'_i > y_i = u_i^{e_i}$, alors y' ne domine pas u^{e_i} . Si $y'_i < y_i$, alors il existe $j \neq i$ tel que $y'_j > u_j^e = u_j^{e_j}$, donc y' ne domine pas non plus u^{e_i} . Par conséquent, aucun point de $[y^a y^b]$ ne domine strictement u^{e_i} , donc ce point vérifie la condition (i) de la définition 1.4.5.
- (ii) La preuve est similaire à celle de la condition (ii) du lemme 4.5.1.
- (iii) Nous faisons les mêmes hypothèses à propos de $[y^a y^b]$ que dans la preuve du lemme 4.5.1. Nous supposons de plus, sans perte de généralité, que $u_1^c = u_1^d$, $u_2^c < u_2^d$ et $u_3^c > u_3^d$. Il nous faut donc énumérer ce qui arrive pour chaque projection de y^a et y^b .
 - $y^{a_1} : y_1^{a_1} = u_1^{c_2} = u_1^c$, $y_2^{a_1} = u_2^{c_2} = y_2^c$ et $y_3^{a_1} = y_3^a < u_3^{c_2} = u_3^c$ donc $y^{a_1} \leq u^{c_2}$. Par conséquent, y^{a_1} ne vérifie pas la condition (ii) de la définition 1.4.5.
 - $y^{b_1} : \text{Similairement, } y^{b_1} \leq u^{d_3}$. Par conséquent, y^{b_1} ne vérifie pas la condition (ii) de la définition 1.4.5.
 - $y^{b_2} : \text{Similairement, } y^{b_2} \leq u^{d_3}$. Par conséquent, y^{b_2} ne vérifie pas la condition (ii) de la définition 1.4.5.
 - $y^{a_3} : \text{Similairement, } y^{a_3} \leq u^{c_1}$ et $y^{a_2} \leq u^{c_2}$. Par conséquent, y^{a_3} ne vérifie pas la condition (ii) de la définition 1.4.5.
 - $y^{b_3} : \text{Ce point ne vérifie pas la condition (i) de la définition 1.4.5, donc il ne peut pas non plus en vérifier la condition (ii).}$
 - $y^{a_2} : y^{a_2}$ peut vérifier la condition (ii) de la définition 1.4.5, en fonction de la relation entre y^a et u^d . Si $y^a < u^d$, alors nous pouvons utiliser le même raisonnement que dans le point précédent pour montrer que $y^{a_2} \leq u^{d_1}$ et conclure que y^{a_2} ne vérifie pas la condition (ii) de la définition 1.4.5. Si $y_3^a > u_3^d$, alors $y^{a_2} = u^{e_1}$ où $u^e \in [u^c u^d]$ et tel que $u_3^e = y_3^a$. Par conséquent, y^{a_2} est également la projection d'un point de $[u^c u^d]$ et vérifie potentiellement la condition (ii) de la définition 1.4.5.
- (iv) Nous faisons les mêmes hypothèses que dans le point précédent. Dans ce contexte, pour tout $u \in [u^{c_3} u^{d_3}]$, $u \leq u^{d_3}$ donc u ne peut vérifier la définition 1.4.5-(ii).
- (v) La preuve est similaire que pour la condition (iv) du lemme 4.5.1 pour les points isolés et les arêtes plates. Ainsi, nous nous concentrons ici sur l'arête non plate $(y^{a_2} u^{d_1})$. Pour tout point $y \in (y^{a_2} u^{d_1})$, il existe $y' \in (y^a y^b)$ tel que $y_1 = y'_1$. Nous considérons $v^1 = y + \alpha e_1$ avec $\alpha \in \mathbb{R}_{>}$ et voulons montrer que v^1 est strictement dominé. Considérons donc $y'' \in [y^a y^b]$ tel que $y''_1 = \min(y_1 + \alpha, y^b)$. Alors, $\frac{y'_1 + y''_1}{2} < v^1$. De façon similaire, nous obtenons que $v^2 = y + \alpha e_2$ et $v^3 = y + \alpha e_3$ sont strictement dominés. Par conséquent, l'affirmation est démontrée.

□

Il nous faut maintenant observer la mise à jour d'une arête délimitante non plate (nous dirons *arête oblique*) à partir d'une arête réalisable.

Exemple 4.5.3 – Soient deux arêtes $I_1 = [y^a y^b]$ et $I_2 = [u^c u^d]$ telles que représentées dans la figure 4.5. I_1 est une arête réalisable et I_2 est une arête délimitante oblique, telle que définie par le lemme 4.5.2. Dans cet exemple, aucun nouvel objet délimitant n'est introduit.

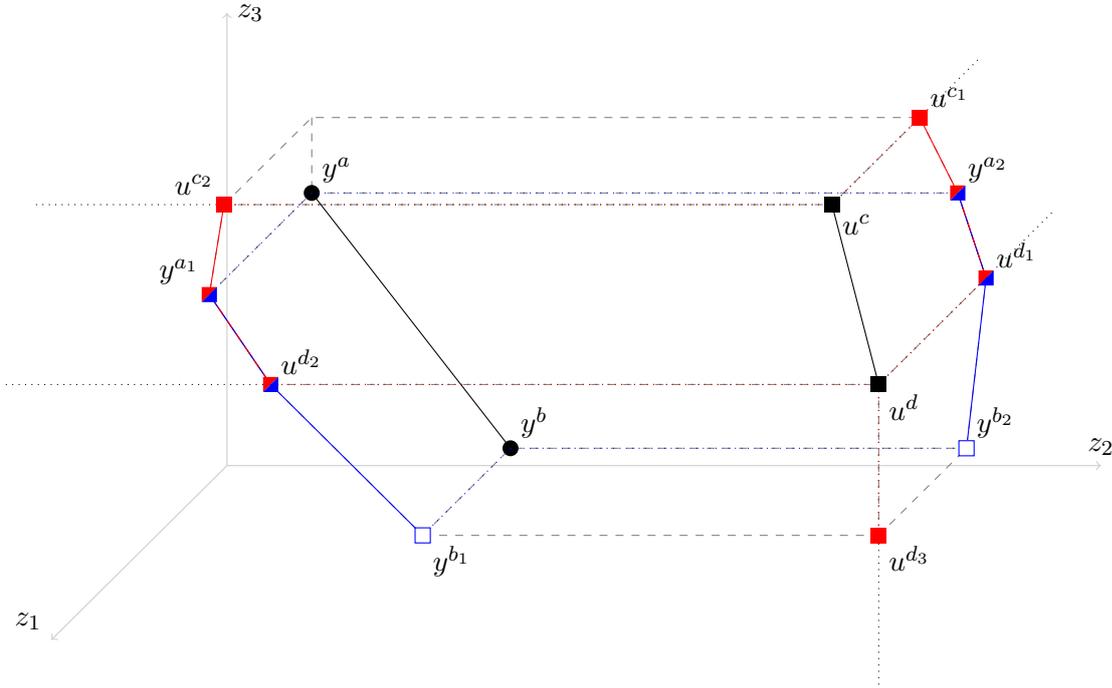


Figure 4.5 – Mise à jour de l’arête délimitante plate $[u^c u^d]$ grâce à l’arête réalisable $[y^a y^b]$.

Lemme 4.5.3. Soient $Q \subset U$, $I_1 = [y^a y^b] \subset U \setminus Q$ et $I_2 = [u^c u^d] \in D(Q \setminus I_1)$ tels que I_2 soit une arête oblique et que tout point de I_2 soit strictement dominé par au moins un point de I_1 .

- (i) Les points u^{f_i} vérifient la condition (i) de la définition 1.4.5.
- (ii) Nous considérons les résultats de la projection de $[y^a y^b]$ parallèlement à chaque z_i , $i \in \{1, \dots, 3\}$. Nécessairement, une de ces projections ne vérifie pas la condition (i) de la définition 1.4.5, à l’exception d’une de ses extrémités. Les deux autres projections vérifient la condition (i) de la définition 1.4.5.
- (iii) Parmi les points y^{a_i} et y^{b_i} , au plus deux vérifient la condition (ii) de la définition 1.4.5.
- (iv) Nous considérons le résultat de la projection de $[u^c u^d]$ parallèlement à chaque z_i , $i \in \{1, \dots, 3\}$. Nécessairement, une de ces arêtes ne vérifie pas la condition (ii) de la définition 1.4.5, à l’exception d’une de ses extrémités.
- (v) Il n’existe pas de point $v \in (u - \mathbb{R}_{\geq}^3)$ vérifiant la condition (i) de la définition 1.4.5 en étant dominé par les points u^i et les arêtes $]y^{a_i} y^{b_i}[$ des points (i) et (ii).

Démonstration.

- (i) La preuve est similaire à celle de la condition (i) du lemme 4.5.2.
- (ii) La preuve est similaire à celle de la condition (ii) du lemme 4.5.2.
- (iii) La preuve est similaire à celle de la condition (iii) du lemme 4.5.2, sauf que deux points vérifiant potentiellement la condition (ii) de la définition 1.4.5 sont obtenus.
- (iv) La preuve est similaire à celle de la condition (iv) du lemme 4.5.2.
- (v) La preuve est similaire à celle de la condition (v) du lemme 4.5.2.

□

Nous devons maintenant aborder le cas des facettes. Pour commencer, nous définissons les facettes délimitantes avant de montrer que toute facette non dominée est délimitante.

Définition 4.5.2 (Facette délimitante). Une facette A est une *facette délimitante* si tout point de son intérieur relatif est un point délimitant.

Lemme 4.5.4. *Toute facette non dominée de U est une facette délimitante de $D(U)$.*

Démonstration.

Soit $A \in U$ une facette non dominée par d'autres points de U . Soit $y \in A$ un point de son intérieur relatif. Montrons que y est un point délimitant (i.e. vérifie les deux conditions de la définition 1.4.5).

Tout d'abord, puisque A est par hypothèse non dominé, alors y est non dominé également. En particulier, il est donc strictement non dominé et vérifie donc la condition (i) de la définition 1.4.5.

Il nous reste à montrer que y vérifie la seconde condition. Soit donc v tel que $y \leq v$ et montrons que v ne vérifie pas la condition (i) de la définition 1.4.5. Si $y < v$, l'affirmation est évidente, donc nous supposons que y ne domine pas strictement v . Dans ce cas, y et v ont au moins une coordonnée commune. Sans perte de généralité, faisons l'hypothèse que $v = y + \epsilon e_1$. Soit alors $y' \in A$ tel que $y'_2 = v_2$ et $y'_3 = v_3$. On a $\frac{y+y'}{2} < v$, donc v ne vérifie pas la condition (i) de la définition 1.4.5.

Par conséquent, y est bien un point délimitant et la proposition est démontrée.

□

Corollaire 4.5.5. *Les facettes non dominées de l'ensemble U peuvent être ajoutées à l'ensemble $D(U)$ sans condition.*

Les cas concernant la mise à jour d'objets délimitants à partir d'une facette réalisable sont a priori plus complexes à décrire car il existe un très grand nombre de configurations possibles pour la forme de la facette. Néanmoins, beaucoup de ces configurations sont symétriques et nous allons donc nous restreindre à une seule configuration couvrant toutes ces possibilités. Dans l'exemple qui suit, nous établissons les trois cas observables pour tout type de facette.

Exemple 4.5.4 – Soit une facette triangulaire I définie par ses trois points extrêmes y^a , y^b et y^c et représentée par la figure 4.6. On note I_i le résultat de la projection de I sur la i^{me} coordonnée. On observe différents résultats pour les facettes I_i en fonction de la valeur de i :

- pour $i = 1$, seules les arêtes $[y^{a_1}y^{c_1}]$ et $[y^{c_1}y^{b_1}]$ vérifient la condition (i) de la définition 1.4.5 ;
- pour $i = 2$, seule l'arête $[y^{a_2}y^{b_2}]$ vérifie la condition (i) de la définition 1.4.5 ;
- pour $i = 3$, seul y^{a_3} ne vérifie la condition (i) de la définition 1.4.5.

Ces observations dépendent uniquement de la forme de la facette I , en particulier de la position relative de ses points extrêmes. Néanmoins, il est clair que la partie d'une projection I_i vérifiant la condition (i) de la définition 1.4.5 correspond uniquement aux points de I_i qui ne sont pas dominés par d'autres points de I_i . On peut donc en déduire que les trois seuls cas observables sont ceux de l'exemple 4.5.4.

Il reste donc à observer pour chaque I_i quels points sont délimitants.

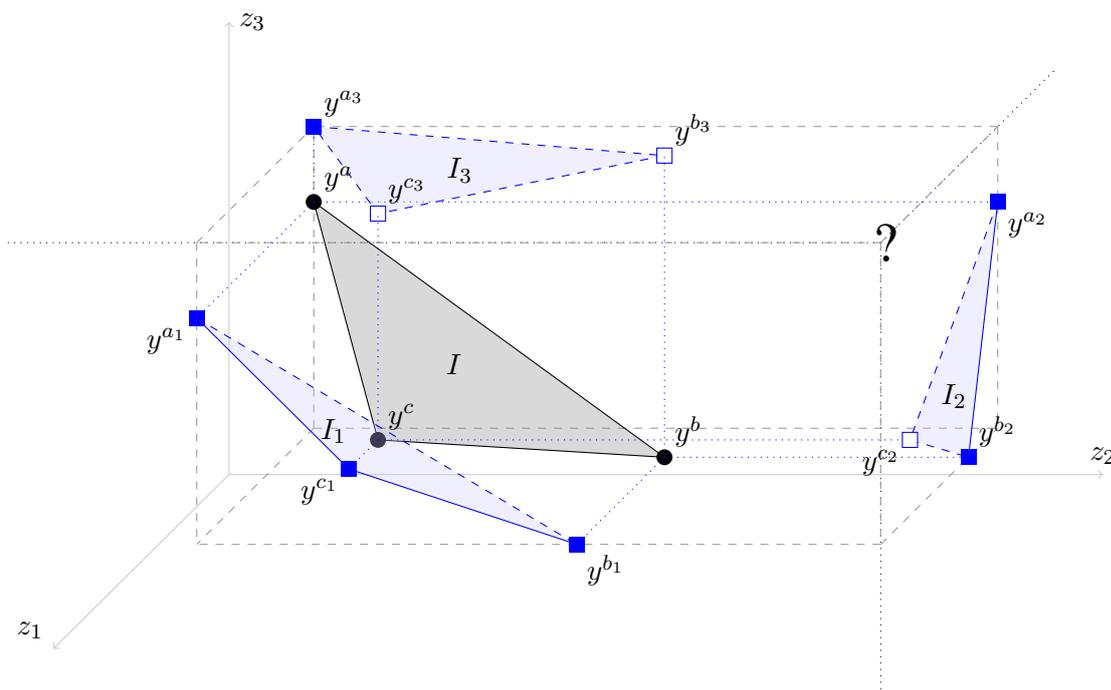


Figure 4.6 – Projections de la facette I pour le calcul de nouveaux objets délimitants. L'objet délimitant d'origine n'est pas représenté.

Lemme 4.5.6. Soit la facette I telle que représentée par la figure 4.6 et un objet délimitant u . Alors, dans la facette I_1 :

- (i) y^{a1} et y^{b1} ne vérifient pas la condition (ii) de la définition 1.4.5.
- (ii) les arêtes issues de $]y^a y^c[$ et $]y^c y^b[$ vérifient la condition (ii) de la définition 1.4.5.

Démonstration.

- (i) La preuve est similaire à celle concernant y^{a_i} et y^{b_i} dans le cas (iii) du lemme 4.5.1.
- (ii) Pour les arêtes issues de $]y^a y^c[$ et $]y^c y^b[$, la preuve est similaire à celle concernant $]y^{a1} y^{b1}[$ dans le cas (iii) du lemme 4.5.1. Pour y^{c1} , il suffit de considérer qu'il ne s'agit d'une extrémité et le même raisonnement s'applique.

□

Lemme 4.5.7. Soit la facette I telle que représentée par la figure 4.6. Alors, dans la facette I_2 :

- (i) y^{a2} et y^{b2} ne vérifient pas la condition (ii) de la définition 1.4.5.
- (ii) la ou les arêtes issues de $]y^a y^b[$ vérifient la condition (ii) de la définition 1.4.5.

Démonstration.

- (i) La preuve est similaire à celle concernant y^{a_i} et y^{b_i} dans le cas (iii) du lemme 4.5.1.
- (ii) La preuve est similaire à celle concernant $]y^{a_i} y^{b_i}[$ dans le cas (iii) du lemme 4.5.1.

□

Lemme 4.5.8. *Soit la facette I telle que représentée par la figure 4.6. Alors, dans la facette I_3 , y^{a_2} ne vérifie pas la condition (ii) de la définition 1.4.5.*

Démonstration.

À nouveau, la preuve est similaire à celle concernant y^{a_i} et y^{b_i} dans le cas (iii) du lemme 4.5.1.

□

Il est maintenant possible de conclure sur l'unique origine des facettes délimitantes.

Proposition 4.5.9. *Il y a équivalence entre les facettes non dominées de U et les facettes délimitantes de $D(U)$.*

Démonstration.

Le lemme 4.5.4 montre que toute facette non dominée est une facette délimitante. Puisque ces facettes sont non dominées, il n'y a pas besoin de les mettre à jour lors du calcul de l'ensemble des objets délimitants. Grâce aux lemmes 4.5.1, 4.5.2 et 4.5.3, nous savons de plus qu'il n'est possible d'obtenir que des points et arêtes délimitants lors de la mise à jour de points et arêtes délimitantes à partir d'arêtes réalisables. Enfin, les lemmes 4.5.6, 4.5.7 et 4.5.8 montrent la même chose à partir de facettes réalisables. Par conséquent, les seules facettes délimitantes obtenues sont les facettes non dominées, il y a donc équivalence.

□

Corollaire 4.5.10. *Seuls des points et arêtes délimitants peuvent être obtenus par la mise à jour de points et d'arêtes délimitants.*

Nous savons par conséquent que les facettes ne nécessitent pas de calcul supplémentaire pour déterminer qu'elles appartiennent à $D(U)$. Seuls les points et arêtes délimitants ont donc besoin d'être déterminés. Par conséquent, il est maintenant possible de décrire le calcul de la zone de recherche.

La procédure servant à déterminer la zone de recherche est développée dans les algorithmes 4.2, 4.3 et 4.4. L'algorithme 4.2 détermine un ensemble d'objets délimitants à partir de l'ensemble de facettes réalisables U . Pour cela, il considère séparément chaque facette $y \in U$, qu'il compare avec l'ensemble délimitant pour le mettre à jour.

Cette comparaison est décrite par l'algorithme 4.3 qui se déroule comme suit. La facette y est comparée à chaque objet délimitant u pour déterminer leurs relations de dominance. Si y domine strictement une partie de u (ligne 5), alors cette partie de u ne respecte plus la condition (i) de la définition 1.4.5 et doit donc être mise à jour. On supprime donc la partie dominée de u (ligne 6) tout en conservant sa partie non dominée (ligne 7). Ensuite, pour chaque direction possible, on calcule les u^i et les y^i vérifiant la condition (i) de la définition 1.4.5 que l'on ajoute à l'ensemble des éléments délimitants.

Quand l'algorithme 4.3 est terminé, il reste donc à filtrer les éléments ne vérifiant pas la condition (ii) de la définition 1.4.5. C'est le but de l'algorithme 4.4 qui compare simplement tous les éléments de $D(Q)$ entre eux. Quand un élément u domine une partie d'un autre élément u' , alors cette partie ne vérifie pas la condition (ii) de la définition 1.4.5 et est donc supprimée.

À la fin, il ne reste qu'à ajouter les facettes non dominées, qui sont les seules facettes délimitantes possibles.

Entrée Ensemble de facettes réalisables U , point nadir y^N et point idéal y^I
Sortie Ensemble $D(U)$

```

1:  $Q \leftarrow \emptyset$ 
2:  $D(U) \leftarrow y^N$ 
3: pour chaque  $y \in U$  faire
4:    $\text{updateUBSmixed}(y \downarrow, D(Q) \downarrow, D(Q \cup \{y\}) \uparrow)$ 
5:    $Q \leftarrow Q \cup \{y\}$ 
6:    $\text{filterUBSmixed}(D(Q))$ 
7: fin pour
8: pour chaque Triangle  $t \in U$  faire
9:    $D(Q) \leftarrow D(Q) \cup \{t\}$ 
10: fin pour

```

/ À chaque étape, Q est un sous-ensemble de U */*

/ Calcul de $D(U)$ en considérant chaque facette de U séparément */*

/ Ajout de toutes les facettes de dimensions 2 de U */*

Algorithme 4.2: Procédure computeUBSmixed

Entrée Facette réalisable $y \in U$, ensemble $D(Q)$
Sortie Ensemble $D(Q \cup \{y\})$

```

1:  $W \leftarrow \emptyset$ 
2:  $N \leftarrow \emptyset$ 
3: pour chaque  $u \in D(Q)$  faire
4:    $I_{dom} \leftarrow u \cap (y \cup \mathbb{R}_{>}^3)$ 
5:   si  $I_{dom} \neq \emptyset$  alors
6:      $W \leftarrow W \cup I_{dom}$ 
7:      $N \leftarrow N \cup (I \setminus I_{dom})$ 
8:     pour  $i \in \{1, \dots, p\}$  faire
9:       calcul des  $u^i$  et des  $y^i$ 
10:      pour  $u' \in \{u^i, y^i\}$  faire
11:         $N \leftarrow N \cup \{u'\}$ 
12:      fin pour
13:    fin pour
14:  fin si
15: fin pour
16:  $D(Q \cup \{y\}) \leftarrow (D(Q) \setminus W) \cup N$ 

```

/ W est l'ensemble de facettes de $D(Q) \setminus D(Q \cup \{y\})$ */*

/ N est l'ensemble de facettes de $D(Q \cup \{y\}) \setminus D(Q)$ */*

/ Comparaison de y avec chaque arête u de $D(Q)$ */*

/ I_{dom} est la partie de u strictement dominée par y */*

Algorithme 4.3: Procédure updateUBSmixed

Entrée Ensemble $D(Q)$

Sortie Ensemble $D(Q)$ filtré

```

1: pour chaque  $u \in D(Q)$  faire
2:   pour chaque  $u' \neq u \in D(Q)$  faire
3:      $I_{dom} \leftarrow u \setminus (u' \cup \mathbb{R}_{\leq}^3)$ 
4:     si  $I_{dom} \neq \emptyset$  alors
5:        $D(Q) \leftarrow D(Q) \setminus u' \cup (u' \setminus I_{dom})$ 
6:     fin si
7:   fin pour
8: fin pour

```

Algorithme 4.4: Procédure filterUBSmixed

4.6 Résultats expérimentaux

Pour tester et comparer la méthode en deux phases avec l'algorithme de Branch & Bound, les instances utilisées ont été générées suivant la même procédure que celle présentée dans la section 2.5

Les résultats expérimentaux sont présentés dans le tableau 4.1. Trois méthodes y sont comparées :

- l'algorithme de Branch & Bound, utilisé globalement ;
- l'algorithme de Branch & Bound auquel on fournit l'ensemble bornant supérieurement obtenu à partir du calcul de Y_{SN} ;
- la méthode en deux phases calculant Y_{SN} puis appliquant l'algorithme de Branch & Bound localement.

Les instances sont générées de manière similaire à celles utilisées dans les chapitres précédents. Les trois fonctions objectifs ont été générées aléatoirement et les tailles d'instances sont représentées par la même notation.

| Taille | <i>Branch & Bound</i> | Phase 1 + <i>B & B</i> | Deux phases |
|------------|---------------------------|----------------------------|-------------|
| 10 × 10(5) | 1,5 | 1,5 | 0,7 |
| 12 × 12(6) | 3,2 | 3,1 | 3 |
| 14 × 14(7) | 31,9 | 55,2 | 16 |
| 16 × 16(8) | 89 | 250 | 174 |
| 18 × 18(9) | 429,8 | 458 | 193 |

Table 4.1 – Comparaison des temps moyens de résolution de l'algorithme de Branch & Bound global, de l'algorithme de Branch & Bound global utilisant une initialisation et de la méthode en deux phases utilisant localement le Branch & Bound.

4.6.1 Observations générales

La première observation ressortant du tableau 4.1 de résultats est que les temps de résolutions sont significativement plus longs que dans le contexte biobjectif. Les problèmes sont évidemment plus difficiles à résoudre et les différents algorithmes paient la perte des optimisations liées à l'ordre naturel des points non dominés.

Une autre remarque que l'on peut faire concerne les instances de la taille $16 \times 16(8)$, pour laquelle les temps moyens de résolution pour les méthodes débutant par le calcul de Y_{SN} sont étonnamment lents. Ces différences sont dues à une instance particulière, pour laquelle la première phase peine à s'achever.

4.6.2 Comparaison des temps de résolutions

Parmi les trois méthodes comparées, les deux employant l'algorithme de Branch & Bound global avec et sans initialisation de l'ensemble bornant supérieurement, ont des temps comparables pour les deux plus petites tailles d'instances. Pour les instances plus grandes, le Branch & Bound sans initialisation prend l'avantage (429,8 s contre 458 s).

Cette différence peut s'expliquer par la durée de la phase 1. En effet, quasiment toutes les facettes non dominées sont obtenues lors de la première phase, mais les nombreuses comparaisons qu'elles imposent rendent ce calcul lent. Par la suite, l'algorithme de Branch & Bound est utilisé globalement et va donc devoir réexplorer toutes les branches menant aux solutions déjà obtenues, sans possibilité de les couper. On peut donc supposer que le fait que la phase 1 obtienne beaucoup des facettes non dominées, combiné au fait que les comparaisons sont bien plus difficiles dans le contexte triobjectif, empêchent l'algorithme de sonder de grandes parts des branches menant aux bonnes solutions déjà connues.

En ce qui concerne l'algorithme de Branch & Bound employé seul, l'exploration est également globale et ne peut s'appuyer sur le bon ensemble bornant obtenu lors de la phase 1. En revanche, le temps perdu faute d'ensemble bornant initialisé ne semble pas compenser le temps passé à calculer cet ensemble, et le Branch & Bound seul est donc plus rapide sans initialisation.

Enfin, la méthode en deux phases est la plus rapide, quelle que soit la taille d'instance. La seule taille pour laquelle les résultats sont comparables est $12 \times 12(6)$, mais la différence s'accroît ensuite largement au fur et à mesure que les instances grandissent (à l'exception du cas $16 \times 16(8)$ pour lequel une instance problématique nuit aux résultats). Pour la plus grande taille d'instance, on atteint donc un ratio de 2,22 (193 s contre 429,8 s). Nous avons déjà vu que la première phase n'était pas utile quand elle était suivie d'une exploration globale. Les résultats de la méthode en deux phases montrent en revanche que des explorations locales de la zone de recherche permettent de considérablement réduire le temps de résolution. En effet, chaque exploration locale peut sonder plus efficacement son arbre de recherche et c'est cette importante différence qui rend la méthode en deux phases efficace. Néanmoins, la part la plus importante du temps de résolution n'est pas dédiée à la première phase ni aux explorations locales de la zone de recherche, mais à la détermination de celle-ci.

4.6.3 Potentiels facteurs d'amélioration

La méthode en deux phases est donc la plus rapide des trois méthodes testées et nous avons vu que l'étape la plus consommatrice de temps est celle de la détermination de la zone de recherche.

En effet, les différentes comparaisons effectuées par les algorithmes 4.2, 4.3 et 4.4 sont nécessairement globales. Il s'agit d'un passage obligé pour permettre à l'exploration de se faire localement lors de la deuxième phase.

L'explosion du nombre de facettes impliquée par le recours à la triangulation est à mettre directement en cause dans cette contre-performance : la simplification du traitement des facettes se fait au détriment de l'efficacité générale de la résolution. Une piste possible serait donc de revenir sur la triangulation pour conserver des facettes toujours convexes (c'est-à-dire qui ne contiennent pas de partie dominée) mais potentiellement plus complexes. Le nombre de facettes serait donc restreint, ce qui limiterait l'impact sur le nombre de comparaisons.

Par ailleurs, un grand nombre d'arêtes inutiles sont générées lors de l'exécution des algorithmes 4.2, 4.3 et 4.4. En effet, beaucoup des arêtes produites à partir d'une facette donnée ne vérifieront plus la condition (i) de la définition 1.4.5 lorsqu'une facette non dominée voisine sera considérée par la suite. Il y a par conséquent un risque de devoir gérer un grand nombre d'arêtes intermédiaires inutiles. Puisque les faces sont structurées et localisées en fonction du MOLP auquel elles appartiennent, une autre possibilité d'amélioration serait donc de considérer les MOLP entiers et non les faces séparément lors du calcul de la zone de recherche. Ce regroupement permettrait de réduire le nombre d'étapes intermédiaires en produisant directement des arêtes vérifiant la condition (i) de la définition 1.4.5 à l'échelle du MOLP courant et de ceux déjà considérés.

4.7 Conclusion

Dans ce chapitre, la méthode de résolution des programmes linéaires biobjectifs en variables mixtes a été étendue au contexte triobjectif. Pour ce faire, il a été nécessaire d'introduire un nouveau type d'objet non dominé : les facettes. Malgré les importantes difficultés propres à ce contexte, nous avons pu définir une caractérisation des ensembles non dominés ainsi qu'une méthode pour les mettre à jour.

Par la suite, l'algorithme de Branch & Bound a également été adapté au cadre triobjectif par le biais de ses différents composants. De même, nous avons également adapté la méthode en deux phases en nous appuyant sur la littérature existante et en identifiant le principal point de blocage dans le contexte mixte : la description de la zone de recherche. Cette description a donc été détaillée et de premiers résultats expérimentaux ont pu être obtenus.

Ces premiers résultats témoignent de la nette différence de difficulté entre les contextes biobjectif et triobjectif. Les instances résolues sont en effet de taille bien plus modeste, mais différentes pistes d'amélioration apparaissent déjà.

CHAPITRE 5

Solutions informatiques et application

Ce dernier chapitre se divise en deux parties. La première partie est consacrée à l'important travail d'implémentation mis en œuvre dans cette thèse. Deux composants logiciels importants y sont mis en exergue : le solveur implémentant l'algorithme du simplexe paramétrique biobjectif et celui implémentant la méthode d'approximation externe de Benson. La seconde partie concerne l'application des algorithmes proposés à un problème d'économie réel, dans lequel il s'agit d'évaluer une politique de choix de véhicules et de source d'énergie. Les résultats ont fait l'objet de communication en conférences internationales [De Schepper et al., 2013a, De Schepper et al., 2013b].

| | | |
|------------|--|------------|
| 5.1 | Implémentation logicielle | 105 |
| 5.1.1 | Besoins logiciels | 105 |
| 5.1.2 | Solveur implémentant l'algorithme du simplexe paramétrique biobjectif | 105 |
| 5.1.3 | Solveur implémentant la méthode de Benson pour la résolution de programmes linéaires à trois objectifs | 106 |
| 5.2 | Application à un problème réel | 106 |
| 5.2.1 | Description et formulation du problème | 107 |
| 5.2.2 | Résultats | 110 |
| 5.3 | Conclusion | 114 |

5.1 Implémentation logicielle

L'implémentation logicielle a représenté une part importante du travail de cette thèse. Il était en effet nécessaire de mettre en œuvre nos propositions afin de les évaluer. Nous commençons par un rapide inventaire de nos besoins. Nous décrivons ensuite les choix effectués et les difficultés rencontrées.

5.1.1 Besoins logiciels

Afin de réaliser les expérimentations, les algorithmes décrits dans les précédents chapitres ont été implémentés. Certains composants de ces algorithmes ont été mis en œuvre en utilisant des bibliothèques logicielles ou en détournant leur usage. Les composants ayant nécessité l'usage d'une bibliothèque externe sont :

- la résolution de MILP dans le cadre du calcul des solutions supportées et de l'évaluation des partitions par la méthode de ranking. Le solveur libre Coin-OR CBC* a été utilisé à cette fin ;
- des opérations géométriques en espace tridimensionnel, en particulier la triangulation et le découpage de facettes. Ces opérations ont été réalisées au moyen de la bibliothèque de géométrie computationnelle CGAL†, récemment libérée
- la résolution de MOLP biobjectif et triobjectif pour évaluer les sous-problèmes aux feuilles et lors des relaxations linéaire. Ces réalisations sont développées par la suite.

L'implémentation des composants de résolution des MOLP biobjectif et triobjectif est maintenant détaillée. Parallèlement à nos travaux, une implémentation de solveur pour les MOLP a été introduite [Schenker, 2013].

5.1.2 Solveur implémentant l'algorithme du simplexe paramétrique biobjectif

Un solveur multiobjectif existe déjà dans la littérature : ADBASE ([Steuer, 1975]). Nous avons cependant décidé de ne pas l'utiliser pour des raisons d'accessibilité, de licence et de choix dans le langage de programmation.

Un premier solveur a donc été développé, il s'agissait d'une implémentation simple de toutes les étapes de l'algorithme du simplexe paramétrique. Cependant, les performances n'étaient pas satisfaisantes et il nous a donc fallu envisager une option intermédiaire basée sur un solveur mono-objectif. Dans un premier temps, ce solveur a pour rôle de résoudre le problème en ne considérant que le premier objectif. Ensuite, l'interface de programmation du solveur est utilisée pour exécuter les pivots permettant d'obtenir les solutions extrêmes suivantes. La bibliothèque logicielle que nous avons retenue pour cela est Coin-OR CLP‡ qui présente plusieurs avantages : ce logiciel était déjà bien connu dans l'équipe, son interface de programmation permettait de faire des pivots sur commande et sa licence est libre.

Parmi les avantages supplémentaires, le projet Coin-OR publie également la bibliothèque d'abstraction OSI§ qui propose une interface de programmation unifiée, quel que soit le solveur sous-jacent utilisé. Cela permet donc d'utiliser notre implémentation du simplexe biobjectif

*. <http://projects.coin-or.org/Cbc>

†. <http://www.cgal.org>

‡. <http://projects.coin-or.org/Clp>

§. <http://projects.coin-or.org/Osi>

à partir de n'importe quel solveur mono-objectif pris en charge par OSI, y compris des logiciels commerciaux comme CPLEX ou Gurobi.

Cependant, cette implémentation n'est hélas pas exempte de défauts. La principale difficulté vient indirectement du fait que la bibliothèque de résolution de CLP n'est pas conçue pour le contexte biobjectif. En effet, toutes les optimisations classiques (utilisation de la dualité, bon choix de pivots pour éviter les erreurs de précisions, etc.) ne sont plus applicables dès que la première solution est déterminée. Il en résulte que des imprécisions numériques peuvent s'accumuler rapidement lorsque des pivots *manuels* sont effectués, ce qui mène à la corruption du tableau simplexe qui devient inutilisable. En l'état actuel, nous expérimentons donc des problèmes de stabilité quand la taille des problèmes augmente. C'est ce qui a hélas limité la plupart de nos expérimentations.

Une réécriture de ce solveur est actuellement en projet pour éviter certaines erreurs de conception et utiliser l'interface de programmation la plus récente d'OSI. Cette réécriture devrait permettre d'alléger les manipulations sur le tableau simplexe et donc renforcer la stabilité et la rapidité globale de résolution.

5.1.3 Solveur implémentant la méthode de Benson pour la résolution de programmes linéaires à trois objectifs

Les problèmes de stabilité rencontrés dans le contexte biobjectif nous ont rapidement incités à utiliser une autre méthode de résolution pour les programmes linéaires triobjectifs continus. En effet, alors que la méthode biobjectif peut énumérer les points extrêmes de *gauche à droite*, il était attendu dans le cas à trois objectifs que les nombreux retours en arrière nécessaires à l'exploration de toutes les solutions efficaces causent beaucoup plus d'imprécisions numériques et diminuent la stabilité du solveur.

Pour cette raison, nous avons opté pour la méthode d'approximation externe de Benson [Benson, 1998c], dont l'un des avantages sur le simplexe multiobjectif est qu'elle consiste en de multiples résolutions de problèmes mono-objectifs. Ainsi, puisqu'il ne s'agit plus de faire évoluer manuellement un même problème pour calculer toutes les solutions efficaces, les problèmes liés aux dégénérescences sont évités et les risques de corruption du tableau simplexe sont nuls. Il est donc nécessaire de résoudre un grand nombre de problèmes mono-objectifs, mais le fait de résoudre ces programmes indépendamment est garante de stabilité.

5.2 Application à un problème réel

Les programmes linéaires multiobjectifs en variables mixtes sont tout indiqués pour traiter de nombreux problèmes, puisque les variables binaires (ou discrètes) servent à modéliser des choix tandis que les variables continues représentent des quantités ou des proportions.

En particulier, les applications prenant en compte un deuxième objectif écologique sont courantes (voir par exemple [Higgins *et al.*, 2008, Mavrotas *et al.*, 2013]).

Nous nous sommes en particulier intéressés à un problème d'économie. Celui-ci nous a été proposé par Ellen de Schepper [¶], doctorante en économie à l'université de Hasselt (Belgique) qui mène ses travaux de recherche sur le présent problème et a été présente dans l'équipe OPTI de septembre à novembre 2012. Le travail réalisé en coopération couvre :

- la formalisation du problème qui a conduit à une modélisation de MOMILP ;

¶. <http://www.uhasselt.be/fiche?voornaam=ellen&naam=deschepper>

| Technologie | | Option | |
|-------------|-------------|---------------|---------------------------------|
| $i = 1$ | électricité | $j = 1$ | réseau |
| | | $j = n_1 = 2$ | panneaux solaires |
| $i = 2$ | transports | $j = 1$ | classiques |
| | | $j = 2$ | électriques (réseau) |
| | | $j = n_2 = 3$ | électriques (panneaux solaires) |

Table 5.1 – Organisation et indices des technologies considérées.

- la collecte rigoureuse des données réelles nécessaires pour alimenter le modèle ([De Schepper *et al.*, 2013b]);
- la résolution des instances numériques ainsi obtenues;
- l’analyse des résultats observés en terme économique.

5.2.1 Description et formulation du problème

Le problème concerne une entreprise souhaitant optimiser à la fois ses performances économiques et environnementales. Pour cela, il lui est possible de faire des choix concernant sa politique d’utilisation de deux technologies :

- l’électricité, provenant du réseau électrique classique ou de panneaux solaires;
- le parc automobile, pouvant comprendre des véhicules classiques ou électriques. Les véhicules électriques peuvent de plus être distingués en fonction de la provenance de leur électricité (réseau ou panneaux solaires).

Les variables de décision sont donc les x_{ij} (réelles et comprises entre 0 et 1), représentant la proportion de la demande de la technologie i satisfaite par l’option j . Les constantes n_i sont également introduites pour représenter le nombre d’options de la technologie i . On a donc $n_1 = 2$ et $n_2 = 3$. L’organisation des technologies étudiées est représentée dans le tableau 5.1.

Les coûts économique et écologique de chaque option sont représentés par les coefficients c_{ij}^1 et c_{ij}^2 , où i et j sont à nouveau la technologie et l’option considérées.

La seule contrainte est a priori de satisfaire la demande en énergie et en transports. Une version simplifiée du problème a donc la formulation suivante :

$$\begin{aligned}
 \min z_1 &= \sum_{i=1}^2 \sum_{j=1}^{n_i} c_{ij}^1 x_{ij} \\
 \min z_2 &= \sum_{i=1}^2 \sum_{j=1}^{n_i} c_{ij}^2 x_{ij} \\
 \text{s.c.} \quad & \sum_{j=1}^{n_i} x_{ij} = 1 \quad i = 1, 2 \\
 & x_{ij} \in \mathbb{R}^+ \quad i = 1, 2 \quad j = 1, \dots, n_i
 \end{aligned} \tag{P_{simple}}$$

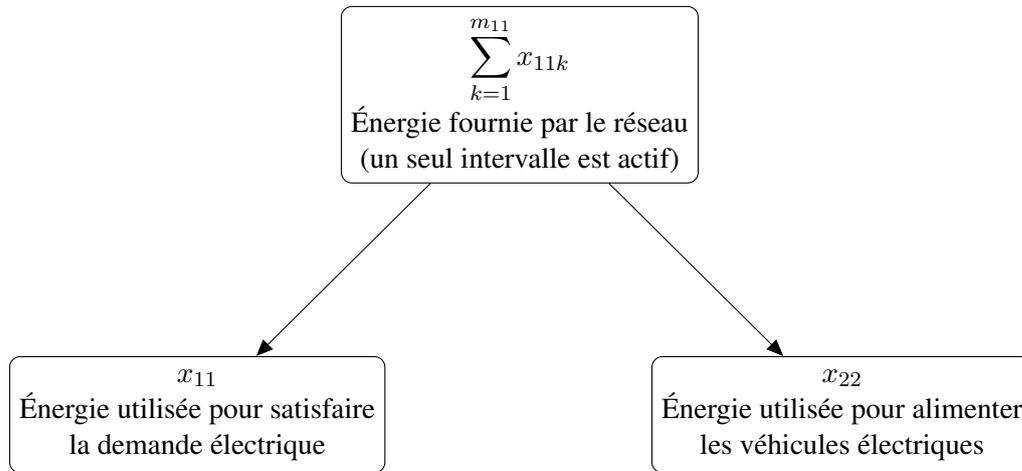


Figure 5.1 – Utilisation de l'énergie fournie par le réseau

Cette formulation ne tient pas compte de potentielles économies d'échelle. Il est en effet possible de faire baisser le coût unitaire d'une technologie quand une grande quantité est achetée. Cependant, l'apparition de ces coûts dépendant de la quantité choisie entre en conflit avec la linéarité des fonctions objectif. Pour contourner ce problème, on peut recourir à des variables binaires pour représenter les intervalles de prix. Les variables x_{ij} sont donc décomposées en m_{ijk} sous-variables x_{ijk} représentant la valeur de x_{ij} quand elle se trouve dans le k^{eme} intervalle, borné par l_{ijk} et u_{ijk} . Pour garantir qu'une seule variable x_{ijk} soit active pour un x_{ij} donné, on introduit également les variables binaires x_{ijk}^b valant 1 si x_{ijk} est active et 0 sinon. Enfin, les coefficients des fonctions objectifs sont également raffinés en c_{ijk}^1 et c_{ijk}^2 (le premier objectif ne changeant pas en fonction des économies d'échelles, tous les c_{ijk}^1 restent égaux).

Pour compléter le modèle, il est également nécessaire de lier certaines variables d'énergie x_{1j} à certaines variables de transport x_{2j} . En effet, une partie de l'électricité provenant du réseau est utilisée pour alimenter des véhicules électriques. Cependant, la demande électrique ne tient pas compte des besoins de ces véhicules. L'énergie supplémentaire requise ne participe donc pas à la satisfaction de la demande et ne doit pas être comptabilisée comme telle. La distinction entre les utilisations de l'énergie provenant du réseau est illustrée par la figure 5.1. Le même raisonnement s'applique pour l'énergie produite par des panneaux solaires et on note e_j la quantité totale d'électricité utilisée quand tous les véhicules sont de type x_{2j} .

Le modèle complet peut être formulé comme suit :

$$\begin{aligned}
(1) \quad \min z_1 &= \sum_{i=1}^2 \sum_{j=1}^{n_i} \sum_{k=1}^{m_{ij}} c_{ijk}^1 x_{ijk} \\
(2) \quad \min z_2 &= \sum_{i=1}^2 \sum_{j=1}^{n_i} \sum_{k=1}^{m_{ij}} c_{ijk}^2 x_{ijk} \\
(3) \quad s.c. \quad &\sum_{j=1}^{n_i} x_{ij} = 1 \quad i = 1, 2 \\
(4) \quad &l_{ijk} x_{ijk}^b \leq x_{ijk} \leq u_{ijk} x_{ijk}^b \quad i = 1, 2 \quad j = 1, \dots, n_i \quad k = 1, \dots, m_{ij} \\
(5) \quad &\sum_{k=1}^{m_{ij}} x_{ijk}^b = 1 \quad i = 1, 2 \quad j = 1, \dots, n_i \\
(6) \quad &\sum_{k=1}^{m_{11}} x_{11k} = x_{11} + e_2 x_{22} \\
(7) \quad &\sum_{k=1}^{m_{12}} x_{12k} = x_{12} + e_3 x_{23} \\
(8) \quad &\sum_{k=1}^{m_{2j}} x_{2jk} = x_{2j} \quad j = 1, \dots, n_2 \\
&x_{ij} \in \mathbb{R}^+ \\
&x_{ijk} \in \mathbb{R}^+ \\
&x_{ijk}^b \in \{0, 1\}
\end{aligned}$$

(P_{complet})

Les fonctions objectifs (1) et (2) représentent respectivement l'impact écologique et le coût, que l'on cherche à minimiser. La contrainte (3) assure que la demande soit satisfaite par les alternatives possibles j pour chaque technologie i . La contrainte (4) contraint, quand la variable x_{ijk}^b est active, chaque variable x_{ijk} à correspondre au k^{eme} intervalle, délimité par la borne inférieure l_{ijk} et la borne supérieure u_{ijk} . Quand x_{ijk}^b est nulle, l'intervalle correspondant est restreint à 0. La contrainte (5) garantit qu'un seul intervalle soit actif pour chaque option j de chaque technologie i . La contrainte (6) assure la correspondance entre la variable x_{11} et les sous-variables x_{11k} , en déduisant la part d'électricité utilisée par les voitures électriques assure la correspondance entre la variable x_{11} et les sous-variables x_{11k} , en déduisant la part d'électricité utilisée par les voitures électriques. La contrainte (7) est similaire, de même que la contrainte (8) qui est plus simple.

Données du problème

Quatre jeux de données sont considérés et comparés. Ceux-ci diffèrent uniquement par les coefficients de la seconde fonction objectif concernant le coût des solutions.

La première différence est liée à la considération de subventions visant à favoriser les énergies propres et renouvelables. Pour étudier l'intérêt incitatif de telles mesures gouvernementales, nous envisageons deux situations. La première tient compte des subventions pour réduire les prix d'achat et d'entretien des panneaux solaires et des véhicules électriques. La seconde situation considère simplement les prix du marché.

| Technologie | Option | Intervalles de prix |
|-------------|---------|---------------------|
| $i = 1$ | $j = 1$ | $m_{ij} = 1$ |
| | $j = 2$ | $m_{ij} = 7$ |
| $i = 2$ | $j = 1$ | $m_{ij} = 3$ |
| | $j = 2$ | $m_{ij} = 3$ |
| | $j = 3$ | $m_{ij} = 3$ |

Table 5.2 – Intervalles de prix pour chaque technologie.

Par ailleurs, deux types de coût économique peuvent être évalués. Le premier est le coût d'investissement des solutions, c'est-à-dire la somme d'argent initiale investie. Le second est le coût du cycle de vie qui comprend les différents coûts d'entretien sur le long terme en plus du coût d'investissement.

Au final, les quatre jeux de données évalués ont pour second objectif :

- le coût d'investissement ignorant les subventions ;
- le coût d'investissement tenant compte des subventions ;
- le coût du cycle de vie ignorant les subventions ;
- le coût du cycle de vie tenant compte des subventions.

Les intervalles de coûts du problème étudié sont communs aux quatre jeux de données et leur nombre est représenté dans le tableau 5.2. Le problème est de taille très modeste puisque seulement 189 combinaisons différentes de variables binaires sont possibles. Néanmoins, l'intérêt de notre méthode ne réside pas ici dans l'efficacité de la résolution (qui est instantanée), mais dans l'exactitude de la représentation des solutions, la procédure de mise à jour dédiée et l'interprétation économique des solutions au regard des performances observées sur les deux objectifs.

5.2.2 Résultats

Le problème a été résolu dans toutes ses variantes au moyen de notre algorithme de Branch & Bound présenté dans le chapitre 2 et une IHM spécifique a été développée pour interpréter les résultats. Pour les deux variantes minimisant le coût d'investissement, la figure 5.2 représente l'ensemble non dominé avec et sans subventions (respectivement en bas et en haut). Pour les deux variantes minimisant le coût du cycle de vie, la figure 5.3 représente l'ensemble non dominé avec et sans subventions (respectivement en bas et en haut). Il est important de remarquer que l'échelle des ordonnées n'est pas la même entre les deux figures.

Sur chaque figure, plusieurs informations sont représentées :

- à l'emplacement de chaque solution extrême, un camembert représente la proportion de chaque type de véhicule :
 - les véhicules classiques en gris ;
 - les véhicules électriques alimentés par le réseau électrique en vert ;
 - les véhicules électriques alimentés par des panneaux solaires en orange.
- sur les arêtes, l'évolution du rapport entre électricité du réseau (en noir) et électricité solaire (en orange). Le cadre coloré à côté de chaque figure montre la correspondance entre proportion d'énergie solaire utilisée et couleur de l'arête.

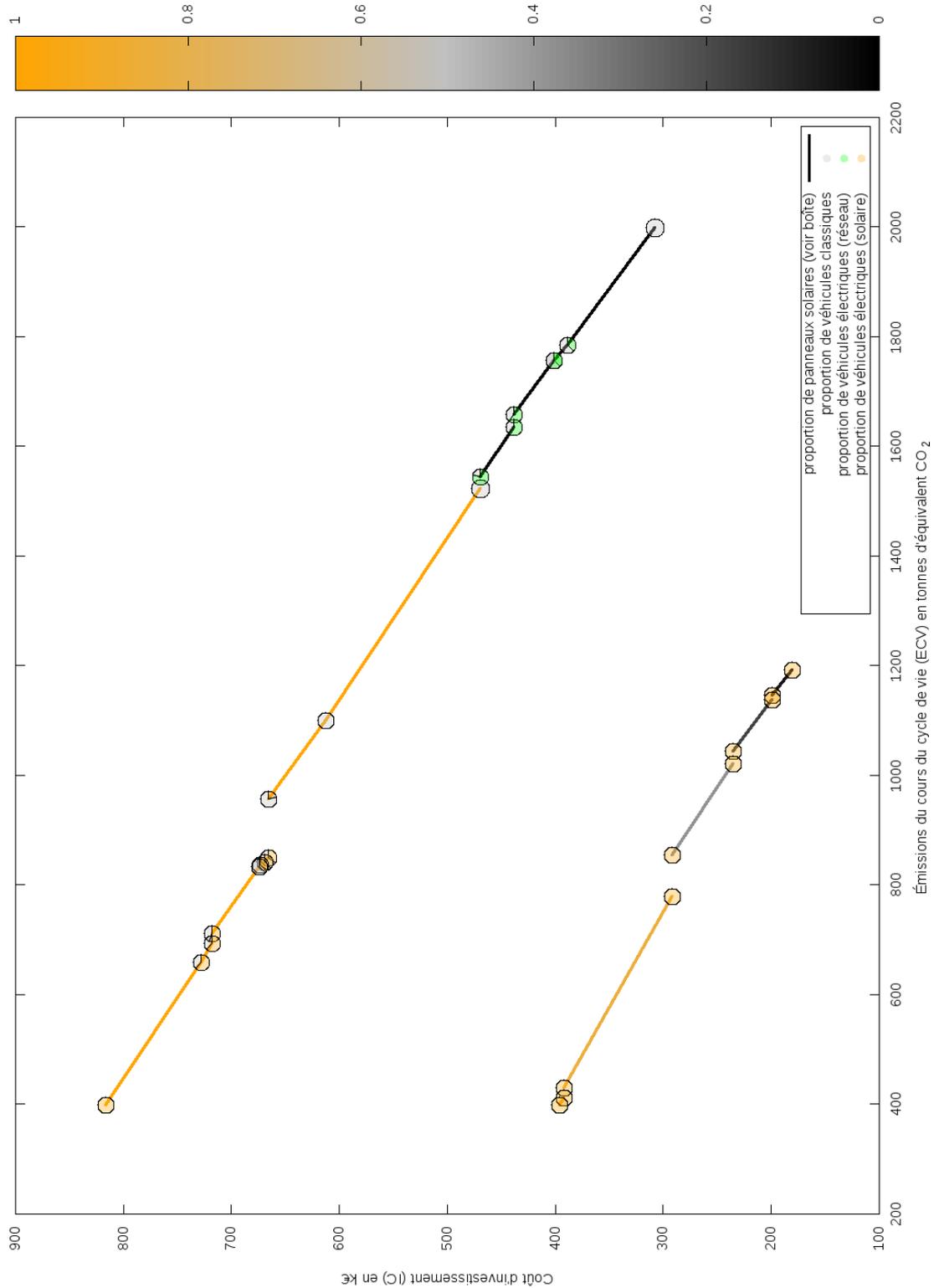


Figure 5.2 – Ensemble non dominé du problème minimisant les émissions de CO_2 et le coût d'investissement (avec et sans subventions).

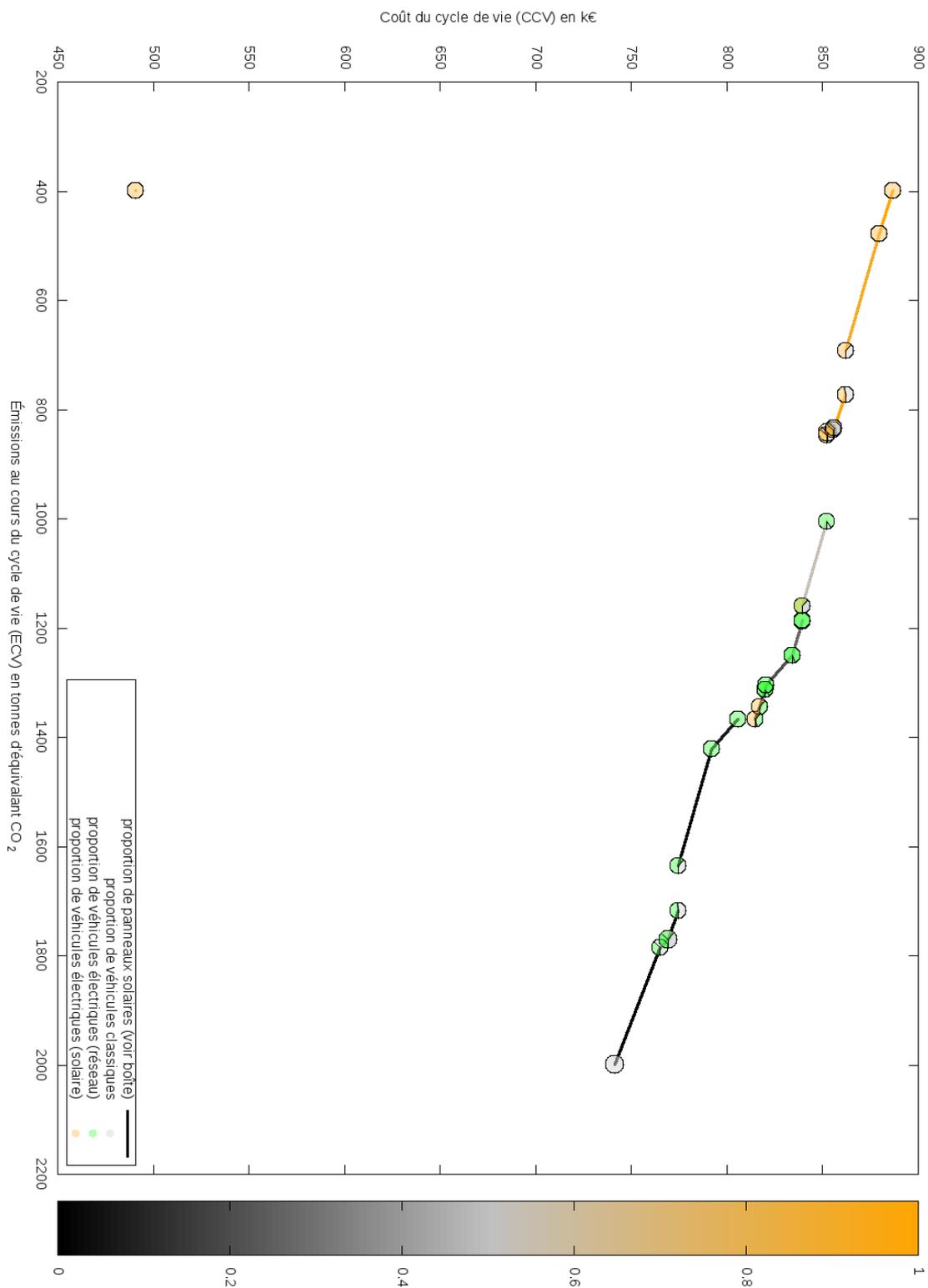


Figure 5.3 – Ensemble non dominé du problème minimisant les émissions de CO₂ et le coût du cycle de vie (avec et sans subventions).

Nous détaillons maintenant les résultats en fonction des objectifs considérés.

Considération du coût d'investissement

Nous commençons par commenter les solutions représentant le problème sans prise en compte de subventions gouvernementales et allons décrire la transition de la solution minimisant le coût d'investissement à celle minimisant l'objectif écologique. La solution optimale au regard du seul objectif de coût d'investissement est évidemment celle n'utilisant que de l'électricité en provenance du réseau et des véhicules classiques. Il s'agit en effet des options technologiques les moins onéreuses : les véhicules à moteur à combustion sont les moins chers et tirer son électricité du réseau n'implique aucun frais d'installation. Le premier levier d'amélioration de l'objectif écologique est le recours aux véhicules électriques alimentés par le réseau qui représentent l'alternative la moins chère pour améliorer l'objectif écologique. Puis, à partir d'un certain seuil (environ 80 % de véhicules électriques), une rupture se fait et il devient plus intéressant de n'utiliser que des véhicules classiques en optant en revanche pour l'électricité d'origine exclusivement solaire. Dès que les panneaux solaires sont installés, l'amélioration écologiques se fait via la transition progressive du parc automobile vers du tout électrique.

Quand les subventions gouvernementales sont prises en compte, il est intéressant de remarquer que toutes les solutions efficaces ont recours à un parc automobile totalement électrique et alimenté par de l'énergie solaire. En effet, les mesures incitatives rendent cette technologie bien plus intéressante, à tel point que les panneaux solaires acquis ne servent qu'à alimenter les véhicules. L'évolution vers la solution la plus écologique se fait à nouveau en remplaçant l'énergie du réseau par de l'énergie solaire.

Considération du coût de cycle de vie

La considération du coût sur le cycle de vie entier et non du seul coût d'investissement a une influence importante sur les solutions proposées. En effet, dans la version ignorant les subventions gouvernementales, l'évolution des solutions est similaire, mais le recours à l'énergie solaire se fait nettement plus tardivement et progressivement. Cela est imputable aux coûts d'entretien élevés des panneaux solaires.

Mais la différence la plus étonnante reste dans le cas où les subventions sont prises en compte, puisqu'une solution idéale existe, minimisant à la fois le coût du cycle de vie et les émissions polluantes. La présence de cette solution se justifie simplement. Tout d'abord, l'optimalité sur l'objectif écologique impose naturellement de choisir les options les plus propres. C'est ce que nous observons puisque la solution propose de n'utiliser que des véhicules électriques alimentés par des panneaux solaires satisfaisant par ailleurs la demande en électricité. Il serait donc à attendre que cette solution soit peu performante sur le plan économique. Cependant, les subventions favorisent fortement les solutions propres non seulement lors de leur acquisition (nous l'avons vu avec les résultats précédents) mais aussi au long de leur cycle de vie. Ce double effet avantage donc les options écologiques au point de les rendre plus intéressantes que leur alternatives classiques. Ce résultat met clairement en évidence l'impact des subventions à visée écologique.

5.3 Conclusion

Dans ce chapitre, nous avons dans un premier temps abordé le travail d'implémentation logicielle accompli au cours de cette thèse. En particulier, les choix d'implémentation des deux méthodes de résolution de programmes linéaires multiobjectifs en variables continues ont été précisés.

La seconde partie de ce chapitre était dédiée à l'étude d'un problème menée à l'université de Hasselt (Belgique). Ce problème, modélisé à partir de données économiques réelles, s'intéresse à l'optimisation simultanée d'un objectif économique et d'un objectif écologique dans l'évaluation d'une politique de choix d'énergie et de véhicules. La disponibilité de différents jeux de données nous a permis de faire varier l'objectif économique quant aux subventions disponibles et l'appréhension à plus ou moins long terme des coûts. Les résultats montrent l'impact non négligeable des politiques de subventions gouvernementales sur le choix des alternatives vertes. Enfin, cette étude valide l'intérêt descriptif de notre représentation des solutions efficaces puisqu'elle permet de mettre en évidence et de donner un sens aux composantes connexes des solutions.

Conclusions et perspectives

Constatant l'absence d'algorithmes exacts et complets pour la résolution de problèmes multiobjectifs en variables mixtes, la motivation centrale de nos travaux fut de proposer des réponses efficaces pour cette classe de problèmes d'optimisation.

Nos propositions remplissent cet objectif dans le cas précis où :

- la structure du problème qui pourrait être éventuellement disponible n'est pas exploitée ;
- les variables du problème sont continues ou binaires ;
- le problème à traiter comporte deux ou trois fonctions objectif.

La représentation des ensembles de solutions pour les contextes à deux et trois objectifs a fait l'objet d'une attention particulière dans nos travaux. Partant d'une représentation appropriée, différentes variantes algorithmiques sont proposées et ont fait l'objet d'expérimentations numériques sur des instances de test et sur un cas d'étude réel.

Par ailleurs, aucun solveur informatique (libre ou commercial) n'est à notre connaissance disponible pour cette classe de problème. Toutes les productions algorithmiques ont fait l'objet d'une réalisation informatique nécessitant la mise au point de certaines briques logicielles indirectes comme un simplexe paramétrique. Les composants les plus pertinents seront mis à disposition de la communauté sous la forme de logiciel libre, apportant ainsi une production opérationnelle.

Partant de la représentation exacte des ensembles de points non dominés, nous avons introduit la notion d'arête non dominée qui permet de prendre en compte le caractère continu de certaines des variables. Puisque l'ensemble non dominé n'est plus uniquement composé de points, cette proposition s'est accompagnée d'une procédure de mise à jour adéquate pour le contexte biobjectif.

Dans un deuxième temps, nous avons cherché à améliorer l'efficacité de notre algorithme nouvellement corrigé. Tout d'abord, la représentation correcte de l'ensemble non dominé constitue de fait une amélioration de l'ensemble bornant supérieurement, mais d'autres propositions ont également été faites sur les composants du Branch & Bound. La procédure de cheminement a été affinée grâce au calcul de coefficients d'efficacité pour les variables. L'apport de cette procédure est un argument fort en faveur d'un cheminement dynamique, qui sera certainement plus efficace encore pour mener vers les bonnes feuilles de l'arbre. Une comparaison de plusieurs ensembles bornant inférieurement a également été faite et a permis de conclure que la meilleure façon d'étudier un nœud intermédiaire dans notre contexte est d'utiliser la relaxation linéaire du sous-problème correspondant. Il s'agit en effet du meilleur compromis entre efficacité de sondage et coût de calcul. De façon surprenante, l'ensemble bornant inférieurement le plus efficace en terme de nombre de nœuds sondés n'est en effet pas celui qui permet d'obtenir les résolutions les plus rapides. Une explication proposée est que l'utilisation d'un solveur MILP mono-objectif pour calculer cette borne est handicapée par l'absence de structure des instances testées. Cet ensemble bornant n'est donc pas à écarter puisqu'il sera certainement bien plus compétitif dans le cadre de problèmes à structure. Enfin, une initialisation de l'ensemble bornant supérieurement par le calcul des solutions supportées extrêmes permet également d'orienter plus rapidement la méthode de résolution vers les branches intéressantes de l'arbre de recherche.

La combinaison de ces propositions nous a permis de réduire significativement les temps de résolution (d'un facteur 400 sur les plus grandes instances testées).

Afin d'évaluer nos travaux sur le Branch & Bound, nous avons également travaillé à l'adaptation d'une autre méthodologie utilisée avec succès dans le contexte de l'optimisation combinatoire : la méthode en deux phases. La première phase, déjà mise à profit pour initialiser l'ensemble bornant supérieurement du Branch & Bound global, est étendue pour tirer parti de la nature continue de certaines variables. Deux méthodes ont été comparées dans le cadre de la deuxième phase. Tout d'abord, nous avons réutilisé notre algorithme de Branch & Bound en l'appliquant successivement dans chaque triangle de recherche. La seconde méthode employée est celle de ranking que nous avons adaptée au contexte mixte. En particulier, la borne représentant la valeur au-delà de laquelle toute somme pondérée des objectifs correspond à un point dominé peut-être resserrée grâce à l'information supplémentaire fournie par les arêtes. Cette modification nous a incité à recourir au concept de point délimitant, d'habitude propre au contexte triobjectif, pour remplacer les points nadir locaux.

La comparaison des résultats expérimentaux du Branch & Bound global et de la méthode en deux phases penche fortement en faveur de cette dernière, quelle que soit la seconde phase employée. Il est en revanche plus difficile de classer les deux secondes phases, mais cet état de fait nous a par la suite inspiré une nouvelle méthode combinant leurs points forts : Rank & Bound. Cette nouvelle méthode s'est avérée obtenir les meilleurs résultats. Cette partie de notre travail a également ouvert de nombreuses perspectives. Premièrement, le ranking pourrait être amélioré en réduisant le nombre de MILP résolus à chaque exploration. Pour cela, il faudrait parvenir à partitionner l'espace des décisions de façon à placer les meilleures solutions dans les partitions ayant le moins de variables libres possibles. Ainsi, le nombre de partitions issues du redécoupage nécessaire serait plus faible, ce qui implique moins de MILP à résoudre. À ce titre, une stratégie inspirée de celle basée sur les coefficients d'efficacité $e(k)$ semble prometteuse.

Concernant Rank & Bound, le ratio apparemment constant entre cette méthode et celle du ranking mérite davantage d'investigations. Une explication proposée concerne l'efficacité de l'ensemble bornant utilisé pour le sondage, qui serait sans effet dans les partitions dont le nombre de variables libres est trop important. Pour contourner ce problème, il pourrait être bénéfique d'adapter le choix de l'ensemble bornant inférieurement en fonction de ce nombre. Enfin, il serait intéressant de confirmer les bons résultats de cette nouvelle méthode sur d'autres classes de problèmes.

Comme attendu, le passage du traitement de deux à trois fonctions objectifs a mis en évidence de nouvelles problématiques. En effet, en plus des difficultés classiques inhérentes aux problèmes triobjectifs, le contexte des variables mixtes voit apparaître des difficultés qui lui sont propres. La première difficulté est bien entendu la représentation des ensembles non dominés, qui doit désormais comporter des facettes de dimension 2 en plus d'arêtes et de points. La représentation proposée étend celle du cas biobjectif et met notamment en lumière certaines de ses idées simples et naturelles pour les réutiliser dans le contexte à trois objectifs, comme le découpage des facettes en triangles. La procédure de mise à jour est également inspirée du cas biobjectif mais moins de concepts étaient réutilisables et nous nous sommes donc orientés vers une procédure plus générale. Les deux cadres de résolution que sont l'algorithme de Branch & Bound et la méthode en deux phases ont par la suite été étendus pour traiter les problèmes triobjectifs. Si la première méthode n'a pas posé de difficulté particulière, la seconde a en revanche mis en évidence une problématique inattendue : la description de la zone de recherche telle qu'elle est proposée en optimisation combinatoire n'est pas suffisante. En effet, de la même façon que les variables mixtes impliquent de nouveaux objets non dominés, elles imposent également de nouveaux objets délimitants plus

complexes (des arêtes et des facettes délimitantes) dont il convient de formaliser la description et le calcul.

Finalement, les résultats expérimentaux permettent de comparer l'algorithme de Branch & Bound global à la méthode en deux phases et confirment la supériorité de la méthode en deux phases dans le contexte triobjectif. Néanmoins, d'importants leviers d'amélioration restent à notre disposition. En premier lieu, le nombre très important de comparaisons globales entre triangles constitue un frein important à la vitesse de résolution. Si ce problème semble partiellement contourné par la méthode en deux phases (dont le but est justement de restreindre l'espace de recherche), il reste possible d'encore améliorer la situation. Tout d'abord, bien que le recours à la triangulation des facettes ait permis de grandement simplifier le traitement des facettes, ce choix engendre la création d'un nombre très important d'objets. Il semble donc intéressant de remplacer la triangulation par le calcul de composantes convexes, potentiellement plus grandes mais moins nombreuses. Une autre solution consiste à filtrer plus efficacement les cas où deux facettes ne sont pas comparables. Cette idée nécessiterait néanmoins des calculs plus coûteux. Enfin, une importante amélioration pourrait avoir lieu lors du calcul de la zone de recherche précédant la seconde phase, qui constitue une part significative du temps de résolution. Dans la méthode proposée, les objets délimitants y sont en effet calculés à partir des différentes faces composant l'ensemble non dominé. Pourtant, de nombreuses faces sont structurées en MOLP et les objets délimitants générés par une face seront certainement mis à jour par les faces voisines, ce qui multiplie les calculs inutiles. Considérer chaque MOLP dans son ensemble devrait donc permettre de calculer immédiatement les objets délimitants pertinents.

Nos travaux ont pu être appliqués à un problème étudié dans le secteur de l'économie de l'environnement. Malgré sa taille modeste, cette application nous a permis de valider notre représentation de l'ensemble non dominé. En effet, en plus de rendre nos algorithmes de résolution corrects et complets, cette représentation au moyen d'arêtes est également porteuse de sens puisqu'elle indique la transition d'une option technologique vers une autre. Ces résultats nous encouragent donc à poursuivre cet effort de mise en application et à considérer d'autres problèmes en variables mixtes importants dans la littérature, comme le problème de localisation de services.

Bibliographie

- [Aneja et Nair, 1979] ANEJA, Y. et NAIR, K. (1979). Bicriteria transportation problem. *Management Science*, 25(1):73–78.
- [Armand et Malivert, 1991] ARMAND, P. et MALIVERT, C. (1991). Determination of the efficient set in multiobjective linear programming. *Journal of Optimization Theory and Applications*, 70(3):467–489.
- [Benson, 1998a] BENSON, H. (1998a). Further analysis of an outcome set-based algorithm for multiple-objective linear programming. *Journal of Optimization Theory and Applications*, 97(1):1–10.
- [Benson, 1998b] BENSON, H. (1998b). Hybrid approach for solving multiple-objective linear programs in outcome space. *Journal of Optimization Theory and Applications*, 98(1):17–35.
- [Benson, 1998c] BENSON, H. P. (1998c). An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13:1–34.
- [Chvatal, 1983] CHVATAL, V. (1983). *Linear programming*. WH Freeman.
- [Dauer et Saleh, 1990] DAUER, J. et SALEH, O. (1990). Constructing the set of efficient objective values in multiple objective linear programs. *European Journal of Operational Research*, 46(3): 358–365.
- [Dauer, 1993] DAUER, J. P. (1993). On degeneracy and collapsing in the construction of the set of objective values in a multiple objective linear program. *Annals of Operations Research*, 46(2):279–292.
- [Dauer et Gallagher, 1996] DAUER, J. P. et GALLAGHER, R. J. (1996). A combined constraint-space, objective-space approach for determining high-dimensional maximal efficient faces of multiple objective linear programs. *European journal of operational research*, 88(2):368–381.
- [Dauer et Liu, 1990] DAUER, J. P. et LIU, Y.-H. (1990). Solving multiple objective linear programs in objective space. *European Journal of Operational Research*, 46(3):350–357.
- [De Loera et al., 2010] DE LOERA, J. A., RAMBAU, J. et SANTOS, F. (2010). *Triangulations : Structures for algorithms and applications*, volume 25. Springer.
- [De Schepper et al., 2013a] DE SCHEPPER, E., VAN PASSEL, S., LIZIN, S., MARTIN, B., VINCENT, T. et GANDIBLEUX, X. (2013a). Optimizing economic and environmental performances of solar power and electric vehicles : A momilp application. *In 22nd International Conference on Multiple Criteria Decision Making*, Málaga, Spain.
- [De Schepper et al., 2013b] DE SCHEPPER, E., VAN PASSEL, S., LIZIN, S., VINCENT, T., MARTIN, B. et GANDIBLEUX, X. (2013b). Multi-objective optimization of clean energy and transportation technologies considering economic and environmental objectives. *In 8th Conference on Sustainable Development of Energy, Water and Environment Systems*, Dubrovnik, Croatia.
- [Delort, 2011] DELORT, C. (2011). *Algorithmes d'énumération implicite pour l'optimisation multi-objectifs : exploitation d'ensembles bornant et applications aux problèmes de sac à dos et d'affectation*. Thèse de doctorat, Université Pierre et Marie Curie (Paris VI).

- [Ecker et Kouada, 1978] ECKER, J. et KOUADA, I. (1978). Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, 14(1):249–261.
- [Ehrgott, 2005] EHRGOTT, M. (2005). *Multicriteria optimization*. Springer, Berlin - Heidelberg, 2 édition.
- [Ehrgott et Gandibleux, 2001] EHRGOTT, M. et GANDIBLEUX, X. (2001). Bounds and bound sets for biobjective combinatorial optimization problems. In *Multiple Criteria Decision Making in the New Millennium*, pages 241–253. Springer.
- [Ehrgott et Gandibleux, 2007] EHRGOTT, M. et GANDIBLEUX, X. (2007). Bound sets for biobjective combinatorial optimization problems. *Computers and Operations Research*, 34(9): 2674–2694.
- [Florios *et al.*, 2010] FLORIOS, K., MAVROTAS, G. et DIAKOULAKI, D. (2010). Solving multiobjective, multiconstraint knapsack problems using mathematical programming and evolutionary algorithm. *European Journal of Operational Research*, 203(1):14–21.
- [Gal, 1977] GAL, T. (1977). A general method for determining the set of all efficient solutions to a linear vectormaximum problem. *European Journal of Operational Research*, 1(5):307–322.
- [Geoffrion, 1968] GEOFFRION, A. (1968). Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22:618–630.
- [Gomory, 1960] GOMORY, R. (1960). An algorithm for the mixed integer problem. Rapport technique, DTIC Document.
- [Hansen, 1980] HANSEN, P. (1980). Bicriterion path problems. In *Multiple criteria decision making theory and application*, pages 109–127. Springer.
- [Higgins *et al.*, 2008] HIGGINS, A. J., HAJKOWICZ, S. et BUI, E. (2008). A multi-objective model for environmental investment decision making. *Computers & operations research*, 35(1):253–266.
- [Isermann, 1974] ISERMANN, H. (1974). Proper efficiency and the linear vector maximum problem. *Operations Research*, 22(1):189–191.
- [Isermann, 1977] ISERMANN, H. (1977). The enumeration of the set of all efficient solutions for a linear multiple objective program. *Operational Research Quarterly*, pages 711–725.
- [Jorge, 2010] JORGE, J. (2010). *Nouvelles propositions pour la résolution exacte du sac à dos multi-objectif unidimensionnel en variables binaires*. Thèse de doctorat, Université de Nantes.
- [Kiziltan et Yucaoglu, 1983] KIZILTAN, G. et YUCAOGLU, E. (1983). An algorithm for multiobjective zero-one linear programming. *Management Science*, 29:1444–1453.
- [Lawler, 1972] LAWLER, E. (1972). A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science*, 18(7):401 – 405.
- [Mavrotas et Diakoulaki, 1998] MAVROTAS, G. et DIAKOULAKI, D. (1998). A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research*, 107(3):530–541.
- [Mavrotas et Diakoulaki, 2005] MAVROTAS, G. et DIAKOULAKI, D. (2005). Multi-criteria branch and bound : A vector maximization algorithm for Mixed 0-1 Multiple Objective Linear Programming. *Applied Mathematics and Computation*, 171(1):53–71.

- [Mavrotas *et al.*, 2013] MAVROTAS, G., SKOULAXINOY, S., GAKIS, N., KATSOUROS, V. et GEORGOPOULOU, E. (2013). A multi-objective programming model for assessment the ghg emissions in msw management. *Waste Management*.
- [Özpeynirci et Köksalan, 2010] ÖZPEYNIRCI, Ö. et KÖKSALAN, M. (2010). An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science*, 56(12):2302–2315.
- [Philip, 1972] PHILIP, J. (1972). Algorithms for the vector maximization problem. *Mathematical Programming*, 2(1):207–229.
- [Philip, 1977] PHILIP, J. (1977). Vector maximization at a degenerate vertex. *Mathematical Programming*, 13(1):357–359.
- [Przybylski, 2006] PRZYBYLSKI, A. (2006). *Méthode en deux phases pour la résolution exacte de problèmes d'optimisation combinatoire comportant plusieurs objectifs : nouveaux développements et application au problème d'affectation linéaire*. Thèse de doctorat, Université de Nantes.
- [Przybylski *et al.*, 2010] PRZYBYLSKI, A., GANDIBLEUX, X. et EHRGOTT, M. (2010). A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7(3):149–165.
- [Przybylski *et al.*, 2012] PRZYBYLSKI, A., GANDIBLEUX, X. et KLAMROTH, K. (2012). An algorithm to compute all nondominated extreme point of a multi-objective integer linear programme. In *CORS/MOPGP'12*, Niagara Falls, Canada.
- [Ramos *et al.*, 1998] RAMOS, R., ALONSO, S., SICILIA, J. et GONZÁLEZ, C. (1998). The problem of the optimal biobjective spanning tree. *European Journal of Operational Research*, 111:617–628.
- [Schandl *et al.*, 2002] SCHANDL, B., KLAMROTH, K. et WIECEK, M. M. (2002). Norm-based approximation in multicriteria programming. *Computers & Mathematics with Applications*, 44(7):925–942.
- [Schenker, 2013] SCHENKER, S. (2013). About an implementation of a 3-objective linear programming solver. In *22nd International Conference on Multiple Criteria Decision Making*, Málaga, Spain.
- [Shao et Ehrgott, 2008] SHAO, L. et EHRGOTT, M. (2008). Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Mathematical Methods of Operations Research*, 68(2):257–276.
- [Sourd et Spanjaard, 2008] SOURD, F. et SPANJAARD, O. (2008). A Multiobjective Branch-and-Bound Framework : Application to the Biobjective Spanning Tree Problem. *INFORMS Journal on Computing*, 20(3):472 – 484.
- [Steuer, 1975] STEUER, R. (1975). Adbase : An adjacent efficient basis algorithm for vector-maximum and interval weighted-sums linear programming problems. *Journal of Marketing Research*, 12:454–455.
- [Steuer, 1986] STEUER, R. (1986). *Multiple Criteria Optimization : Theory, Computation, and Application*.
- [Strijbosch *et al.*, 1991] STRIJBOSCH, L. W., VAN DOORNE, A. G. et SELEN, W. J. (1991). A simplified molp algorithm : the molp-s procedure. *Computers & operations research*, 18(8): 709–716.

- [Teghem, 2012] TEGHEM, J. (2012). *La recherche opérationnelle Tome 1 : Les méthodes d'optimisation*. Editions Ellipses et Editions de l'ULB.
- [Ulungu et Teghem, 1995] ULUNGU, E. et TEGHEM, J. (1995). The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2):149–165.
- [Ulungu et Teghem, 1997] ULUNGU, E. et TEGHEM, J. (1997). Solving multi-objective knapsack problem by a branch-and-bound procedure. In CLÍMACO, J., éditeur : *Multicriteria analysis*, pages 269–278. Springer Verlag, Berlin.
- [Villarreal et Karwan, 1981] VILLARREAL, B. et KARWAN, M. H. (1981). Multicriteria integer programming : A (hybrid) dynamic programming recursive approach. *Mathematical Programming*, 21(1):204–223.
- [Vincent *et al.*, 2011] VINCENT, T., PRZYBYLSKI, A. et GANDIBLEUX, X. (2011). Two phase method for biobjective mixed 0-1 linear programming. In *21st International Conference on Multiple Criteria Decision Making*, Jyväskylä, Finland.
- [Vincent *et al.*, 2013a] VINCENT, T., PRZYBYLSKI, A. et GANDIBLEUX, X. (2013a). Rank and bound for bi-objective mixed integer linear programmes. In *Guepard Workshop*, Paris, France.
- [Vincent *et al.*, 2013b] VINCENT, T., PRZYBYLSKI, A. et GANDIBLEUX, X. (2013b). Résolution exacte de programmes linéaires triobjectifs en variables mixtes. In *14e conférence ROADEF*, Troyes, France.
- [Vincent *et al.*, 2013c] VINCENT, T., PRZYBYLSKI, A. et GANDIBLEUX, X. (2013c). Two-phase method for multiobjective mixed 0-1 linear programming. In *22nd International Conference on Multiple Criteria Decision Making*, Málaga, Spain.
- [Vincent *et al.*, 2013d] VINCENT, T., SEIPP, F., RUZIKA, S., PRZYBYLSKI, A. et GANDIBLEUX, X. (2013d). Multiple objective branch and bound for mixed 0-1 linear programming : Corrections and improvements for the biobjective case. *Computers & Operations Research*, 40:498 – 509.
- [Visée *et al.*, 1998] VISÉE, M., TEGHEM, J., PIRLOT, M. et ULUNGU, E. (1998). Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12:139–155.
- [Wolsey, 1998] WOLSEY, L. (1998). *Integer Programming*. A Wiley-Interscience publication. Wiley.
- [Zeleny, 1974] ZELENY, M. (1974). Linear multiobjective programming. *Lecture Notes in Economics and Mathematical Systems*, 95.

Liste des tableaux

| | | |
|-----|---|-----|
| 1.1 | Récapitulatif des choix de composants pour les méthodes abordées. | 27 |
| 2.1 | Temps de résolution et nombre de nœuds visités en fonction de l'ensemble bornant inférieurement et de la stratégie de branchement utilisés. | 59 |
| 2.2 | Temps de résolution et nombre de nœuds visités en fonction de l'ensemble bornant inférieurement et de la stratégie de branchement utilisés lorsque l'ensemble bornant supérieurement est initialisé. | 60 |
| 2.3 | Temps de résolution et nombre de nœuds visités en fonction des améliorations proposées pour une taille d'instance plus grande (70 contraintes, 35 variables continues, 35 variables binaires). | 62 |
| 3.1 | Comparaison des temps moyens de résolution pour les instances non corrélées (en secondes) de l'algorithme de Branch & Bound global et de la méthode en deux phases utilisant localement le Branch & Bound, le ranking et la méthode Rank & Bound. | 76 |
| 3.2 | Comparaison des temps moyens de résolution pour les instances corrélées (en secondes) de l'algorithme de Branch & Bound global et de la méthode en deux phases utilisant localement le Branch & Bound, le ranking et la méthode Rank & Bound. | 77 |
| 4.1 | Comparaison des temps moyens de résolution de l'algorithme de Branch & Bound global, de l'algorithme de Branch & Bound global utilisant une initialisation et de la méthode en deux phases utilisant localement le Branch & Bound. | 99 |
| 5.1 | Organisation et indices des technologies considérées. | 107 |
| 5.2 | Intervalles de prix pour chaque technologie. | 110 |

Liste des figures

| | | |
|------|--|----|
| 1.1 | Ensemble réalisable du problème MOPL1 dans l'espace des décisions et l'espace des objectifs | 14 |
| 1.2 | Ensemble réalisable du problème MOIPL1 dans l'espace des décisions et l'espace des objectifs | 15 |
| 1.3 | Résolution du problème MOPL1 par l'algorithme du simplexe biobjectif paramétrique. | 17 |
| 1.4 | Illustration de la méthode d'approximation externe de Benson. | 19 |
| 1.5 | Exemple d'arbre binaire. | 23 |
| 1.6 | Comparaison de l'EBI d'un nœud (en trait plein bleu) avec l'EBS (en trait pointillé rouge) du problème global pour déterminer si un nœud peut être sondé ou non. | 25 |
| 1.7 | y^3 n'est dominé ni par y^1 ni par y^2 mais par certaines de leurs combinaisons convexes comme y' | 28 |
| 1.8 | Exemple d'exécution de la première phase appliqué au problème MOIPL1. | 30 |
| 1.9 | Exemple de la méthode de partitionnement de [Lawler, 1972]. x^1 et x^2 sont connues et x^3 doit être déterminé parmi les branches ouvertes. | 31 |
| 1.10 | Exemple d'exécution de la méthode du ranking dans un triangle $\Delta(y^1 y^2)$ dans le cadre de la seconde phase. | 32 |
| 1.11 | Illustration des points délimitant | 35 |
| 1.12 | Mise à jour d'un point délimitant dans le contexte discret. | 36 |
| 2.1 | y^2 et y^3 devraient être éliminés puisqu'ils sont dominés par des points de $[tu]$ | 41 |
| 2.2 | Comparaison entre les deux représentations de Z | 43 |
| 2.3 | y est dominé par sa projection y^\uparrow qui appartient à $[z^1 z^2]$ | 47 |
| 2.4 | y domine sa projection y^\uparrow et n'est donc dominé par aucun point de $[z^1 z^2]$ | 47 |
| 2.5 | Tout point situé dans les régions A_1 ou A_2 n'a aucune relation de dominance avec y . y domine tout point dans B et est dominé par tout point dans C | 48 |
| 2.6 | Tout point situé dans les régions A_i n'a aucune relation de dominance avec l'arête ; tout point de l'arête domine tout point appartenant à B tandis que seule une sous-partie de l'arête domine les points situés dans les régions E_i . C représente la région dans laquelle tout point domine l'intégralité de l'arête et les régions D_i regroupent les points ne dominant qu'une partie de l'arête. | 48 |
| 2.7 | Illustration de la proposition 2.3.4 ii). Aucun point de $[y^1 p]$ ne peut dominer $[pz^2]$ et aucun point de $[pz^2]$ ne peut dominer $[y^1 p]$ | 49 |
| 2.8 | y^\uparrow constitue l'extrémité droite ouverte d'une arête potentiellement non dominée. | 50 |
| 2.9 | $y^{2\leftrightarrow}$ constitue l'extrémité gauche ouverte d'une arête potentiellement non dominée. | 50 |
| 2.10 | Comparaison entre les ensembles bornant induits par les deux représentations de Z comme proposé dans notre amélioration. | 53 |
| 2.11 | Comparaison de la zone bornée par les points de Y_{SN} (en gris foncé) et de celle bornée par les points Y_{SN} et leurs MOLP (la partie gris clair s'ajoute). | 54 |
| 2.12 | Hiérarchie des ensembles bornant supérieurement, du plus faible au plus fort. | 55 |

| | | |
|------|--|-----|
| 2.13 | Hiérarchie des ensembles bornant inférieurement, du plus faible au plus fort. On constate qu'il n'existe pas d'ordre total car on ne peut pas comparer $(\text{conv } E_L + \mathbb{R}_{\geq}^P)_N$ et y^I . | 55 |
| 2.14 | La valeur des deux coefficients objectifs c_1 et c_2 de chaque variable binaire permet de déterminer la valeur à lui attribuer en premier ainsi que le moment où elle doit être fixée. | 56 |
| 2.15 | Sans stratégie de branchement particulière, un grand arbre binaire doit être exploré pour trouver les BOLD qui fourniront des solutions efficaces (en rouge). | 57 |
| 2.16 | Avec un branchement approprié, la taille de la partie à explorer peut être fortement réduite, ce qui permet de sonder rapidement le reste de l'arbre binaire. | 57 |
| 3.1 | La stratégie de cheminement globale suit une direction générale considérant équitablement les deux fonctions objectif. | 68 |
| 3.2 | Les stratégies de cheminement locales tirent partie des informations provenant de chaque triangle pour favoriser la recherche en direction de l'hypoténuse des triangles. | 68 |
| 3.3 | Exemple du processus de la méthode de ranking pour les problèmes en variables mixtes | 70 |
| 3.4 | y^a et y^b ne sont pas connectés et ne se dominent pas. | 71 |
| 3.5 | y^a est faiblement dominé par y^b et ils ne sont pas connectés. | 71 |
| 3.6 | y^a et y^b sont connectés et ne se dominent pas. | 71 |
| 3.7 | Tous les MOLP énumérés par la méthode de ranking ne sont pas nécessairement conservés, même partiellement. | 73 |
| 3.8 | La méthode de ranking ne dispose pas de mécanisme permettant de ne pas considérer les partitions qui ne mènent qu'à des points dominés. | 74 |
| 3.9 | Lors de l'exploration du triangle C , toute la région se trouvant sous la bande délimitée par β est visitée. Si cette bande reste élevée, de nombreux autres points risquent d'être énumérés. | 74 |
| 3.10 | Exemple d'arbre de partitionnement correspondant à l'exemple de la figure 3.8. | 75 |
| 4.1 | Dominance problématique, seule une partie intérieure de la facette est dominée. | 84 |
| 4.2 | La triangulation dans le contexte triobjectif est comparable au découpage des arêtes dans le contexte biobjectif. | 85 |
| 4.3 | Mise à jour du point délimitant u grâce à l'arête réalisable $[y^a y^b]$. | 90 |
| 4.4 | Mise à jour de l'arête délimitante plate $[u^c u^d]$ grâce à l'arête réalisable $[y^a y^b]$. | 92 |
| 4.5 | Mise à jour de l'arête délimitante plate $[u^c u^d]$ grâce à l'arête réalisable $[y^a y^b]$. | 94 |
| 4.6 | Projections de la facette I pour le calcul de nouveaux objets délimitants. L'objet délimitant d'origine n'est pas représenté. | 96 |
| 5.1 | Utilisation de l'énergie fournie par le réseau | 108 |
| 5.2 | Ensemble non dominé du problème minimisant les émissions de CO_2 et le coût d'investissement (avec et sans subventions). | 111 |
| 5.3 | Ensemble non dominé du problème minimisant les émissions de CO_2 et le coût du cycle de vie (avec et sans subventions). | 112 |

Table des matières

| | |
|---|-----------|
| Introduction | 1 |
| Notations | 5 |
| 1 Optimisation multiobjectif | 7 |
| 1.1 Optimisation mono-objectif | 9 |
| 1.1.1 Programmation mathématique | 9 |
| 1.1.2 Programmation linéaire | 9 |
| 1.1.3 Programmation linéaire en variables entières et optimisation combinatoire | 9 |
| 1.1.4 Programmation linéaire en variables mixtes | 10 |
| 1.2 Optimisation multiobjectif | 11 |
| 1.2.1 Définitions | 11 |
| 1.2.2 Propriétés | 11 |
| 1.3 Méthodes de résolution des programmes linéaires multiobjectifs | 15 |
| 1.3.1 Simplexe multiobjectif | 15 |
| 1.3.2 Méthode d'approximation externe de Benson | 18 |
| 1.4 Méthodes de résolution programmes linéaires multiobjectifs en variables entières et combinatoires | 18 |
| 1.4.1 Ensembles bornant | 20 |
| 1.4.2 Branch and bound | 22 |
| 1.4.3 Méthode en deux phases (2 et 3 objectifs) | 29 |
| 1.5 Discussion | 37 |
| 2 Algorithme de Branch & Bound pour les problèmes biobjectifs | 39 |
| 2.1 Défauts de l'algorithme de Mavrotas et Diakoulaki | 41 |
| 2.2 Caractérisation correcte des solutions efficaces | 42 |
| 2.3 Méthode de mise à jour des ensembles non dominés | 45 |
| 2.3.1 Comparaison d'un point avec une arête dont l'intérieur relatif est non vide | 46 |
| 2.3.2 Comparaison de deux arêtes dont l'intérieur relatif est non vide | 48 |
| 2.3.3 Formulation de la procédure de mise à jour | 51 |
| 2.4 Améliorations de l'algorithme de Branch & Bound | 53 |
| 2.4.1 Ensemble bornant supérieurement | 53 |
| 2.4.2 Ensemble bornant inférieurement | 54 |
| 2.4.3 Procédure de branchement | 55 |
| 2.5 Résultats expérimentaux | 57 |
| 2.5.1 Comparaison des ensembles bornant inférieurement | 58 |
| 2.5.2 Apports de la procédure de branchement proposée | 61 |
| 2.5.3 Apports de l'initialisation de l'ensemble bornant supérieurement | 61 |
| 2.5.4 Tests supplémentaires sur de plus grandes instances | 62 |
| 2.6 Conclusion | 63 |

| | | |
|----------|--|------------|
| 3 | Méthodes en deux phases pour les problèmes biobjectifs | 65 |
| 3.1 | Méthode en deux phases avec Branch & Bound | 67 |
| 3.1.1 | Restriction de l'espace de recherche | 67 |
| 3.1.2 | Adaptation des composants du Branch & Bound | 67 |
| 3.2 | Méthode en deux phases avec ranking | 68 |
| 3.2.1 | Adaptation aux programmes en variables mixtes | 69 |
| 3.2.2 | Calcul de la borne β | 69 |
| 3.2.3 | Particularités du contexte mixte | 72 |
| 3.3 | Proposition d'une nouvelle méthode pour la phase 2 | 73 |
| 3.3.1 | Problèmes potentiels de la méthode de ranking | 73 |
| 3.3.2 | Principe de fonctionnement de rank&bound | 75 |
| 3.4 | Résultats expérimentaux | 76 |
| 3.4.1 | Remarques générales | 76 |
| 3.4.2 | Résultats de la méthode de Branch&Bound globale | 77 |
| 3.4.3 | Résultats de la méthode en deux phases avec Branch&Bound | 78 |
| 3.4.4 | Résultats de la méthode en deux phases avec Ranking | 78 |
| 3.4.5 | Résultats de la méthode en deux phases avec Rank&Bound | 79 |
| 3.5 | Conclusion | 79 |
| 4 | Résolution de problèmes à trois objectifs | 81 |
| 4.1 | Difficultés attendues | 83 |
| 4.1.1 | Difficultés inhérentes au contexte mixte | 83 |
| 4.1.2 | Réduction à un seul type de facette | 84 |
| 4.2 | Caractérisation des points dominés et non dominés | 86 |
| 4.3 | Méthode de mise à jour des ensembles non dominés | 87 |
| 4.4 | Adaptation des composants du Branch & Bound | 88 |
| 4.4.1 | Ensembles bornant | 88 |
| 4.4.2 | Procédure de cheminement | 88 |
| 4.5 | Méthode en deux phases | 89 |
| 4.5.1 | Adaptation de la première phase au contexte mixte | 89 |
| 4.5.2 | Deuxième phase | 89 |
| 4.6 | Résultats expérimentaux | 99 |
| 4.6.1 | Observations générales | 100 |
| 4.6.2 | Comparaison des temps de résolutions | 100 |
| 4.6.3 | Potentiels facteurs d'amélioration | 100 |
| 4.7 | Conclusion | 101 |
| 5 | Solutions informatiques et application | 103 |
| 5.1 | Implémentation logicielle | 105 |
| 5.1.1 | Besoins logiciels | 105 |
| 5.1.2 | Solveur implémentant l'algorithme du simplexe paramétrique biobjectif | 105 |
| 5.1.3 | Solveur implémentant la méthode de Benson pour la résolution de programmes linéaires à trois objectifs | 106 |
| 5.2 | Application à un problème réel | 106 |
| 5.2.1 | Description et formulation du problème | 107 |
| 5.2.2 | Résultats | 110 |

| | |
|------------------------------------|------------|
| TABLE DES MATIÈRES | 129 |
| 5.3 Conclusion | 114 |
| Conclusions et perspectives | 115 |
| Bibliographie | 119 |
| Liste des tableaux | 123 |
| Liste des figures | 125 |
| Liste des algorithmes | 126 |

Caractérisation des solutions efficaces et algorithmes d'énumération exacts pour l'optimisation multiobjectif en variables mixtes binaires

Thomas VINCENT

Résumé

Dans ce travail, nous nous intéressons à la résolution exacte de problèmes d'optimisation multiobjectif en variables mixtes binaires. La nature mixte des variables introduit d'importantes différences par rapport aux contextes purement discrets ou continus. Nous proposons donc de prendre en compte ces différences grâce à une représentation appropriée des ensembles de solutions ainsi qu'une procédure de mise à jour dédiée.

Ces propositions nous permettent, dans le contexte biobjectif, d'adapter deux méthodes de résolution usuellement appliquées aux problèmes combinatoires : la procédure de Branch & Bound et la méthode en deux phases. Nous proposons de nombreux affinements pour ces méthodes, comme de nouveaux ensembles bornant ou des stratégies de cheminement. À partir de nos observations sur leurs performances, nous proposons une nouvelle routine pour la seconde phase de la méthode en deux phases, reprenant les points forts des méthodes étudiées.

Dans le contexte triobjectif, nous étendons notre représentation des ensembles de solutions en procédant par analogie avec le cas biobjectif. Les méthodes de résolution sont également adaptées à ce contexte et étudiées. En particulier, la décomposition de la zone de recherche lors de la seconde phase est décrite en détail.

La solution logicielle proposée a été appliquée sur un problème réel : l'évaluation d'une politique de choix de véhicules. Les choix concernés vont de véhicules conventionnels aux véhicules électriques, eux-mêmes alimentés par une source d'électricité classique ou par panneaux solaires.

Mots-clés : programmation linéaire multiobjectif en variables mixtes, branch & bound, méthode en deux phases, résolution exacte.

Abstract

The purpose of this work is the exact solution of multiple objective binary mixed integer linear programmes. The mixed nature of the variables implies significant differences with purely continuous or purely discrete programmes. Thus, we propose to take these differences into account using a proper representation of the solution sets and a dedicated update procedure.

These propositions allow us to adapt for the biobjective case two solution methods commonly used for combinatorial problems: the Branch & Bound algorithm and the two phase method. Several improvements are proposed, such as bound sets or visiting strategies. We introduce a new routine for the second phase of the two phase method that takes advantage of all the relevant features of the previously studied methods.

In the 3-objective context, the solution sets representation is extended by analogy with the biobjective case. Solutions methods are extended and studied as well. In particular, the decomposition of the search area during the second phase is thoroughly described.

The proposed software solution has been applied on a real world problem: evaluation of a vehicle choice policy. The possible choices range from classical to electric vehicles that are powered by grid or solar power.

Keywords: multiple objective mixed integer linear programmes, branch & bound, two-phases method, exact solution.