Titre: Title:	Optimisation globale structurée : propriétés, équivalences et résolution
Auteur: Author:	Charles Audet
Date:	1997
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Audet, C. (1997). Optimisation globale structurée : propriétés, équivalences et résolution [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/6787/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/6787/
Directeurs de recherche: Advisors:	Brigitte Jaumard, & Gilles Savard
Programme: Program:	Unspecified

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION GLOBALE STRUCTURÉE : PROPRIÉTÉS, ÉQUIVALENCES ET RÉSOLUTION

CHARLES AUDET
DÉPARTEMENT DE MATHÉMATIQUES
ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME PHILOSOPHIAE DOCTOR (Ph.D.)
(MATHÉMATIQUES DE L'INGÉNIEUR)
NOVEMBRE 1997

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

OPTIMISATION GLOBALE STRUCTURÉE : PROPRIÉTÉS, ÉQUIVALENCES ET RÉSOLUTION

présentée par : AUDET Charles
en vue de l'obtention du diplôme de : Philosophiae Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. GAUVIN Jacques, Ph.D., président
Mme JAUMARD Brigitte, T.Doc., T.Hab., membre et directrice de recherche
M. SAVARD Gilles, Ph.D., membre et codirecteur de recherche
M. MARCOTTE Patrice, Ph.D., membre
M. VIAL Jean-Philippe, Ph.D., membre externe

REMERCIEMENTS

Je désire d'abord et avant tout remercier Annie qui m'a définitivement aidé plus qu'elle ne le pense. En m'écoutant patiemment, et en discutant maintes fois de la nature et l'avancement de mes travaux, elle m'a permis de synthétiser, comprendre et expliquer un grand nombre de questions.

Merci à mes parents, Christiane et Pierre pour m'avoir inculqué le goût d'apprendre, de m'avoir enseigné à penser (même latéralement), et de m'avoir encouragé sans cesse pour toujours aller plus loin. Merci à mon frère François qui a toujours su alimenter ma curiosité scientifique; comme tu vois ta contribution est bel et bien présente.

Je n'aurais certainement pas pu produire une thèse dont je suis si fier sans l'influence de mes co-auteurs. Je me permets ici de renverser l'ordre alphabétique. Un gros merci à Gilles, Brigitte et Pierre. J'ai eu énormément de plaisir à travailler avec vous, et j'aimerais poursuivre ma carrière dans le même genre d'atmosphère.

Merci à Jean-Philippe Vial, Patrice Marcotte et Jacques Gauvin pour vos commentaires, corrections et suggestions qui ont amélioré la qualité de ce document, et ce, pas seulement au niveau mathématique.

Il m'est impossible de passer sous silence l'immense influence qu'ont eue mes amis sur moi. Je tiens à vous remercier tous, mais malheureusement, je crains d'en oublier. Pour cette raison, je mentionnerai des groupes de gens, plutôt que des individus. Merci à tous ceux avec qui j'ai parlé de mathématiques. Merci à ceux avec qui j'ai jasé autour d'une bière, merci à mes amis d'escalade, merci à mes amis suisses, merci à mes amis moitié grec moitié espagnol moitié québecois, merci à mes amis qui se sont reproduits, merci à ceux qui m'ont aidé avec LaTeX, merci à mes amis mathématiciens, merci à mes amis du GERAD, merci à mes amis ackieurs. Finalement, je remercie tous ceux que j'ai omis de remercier...

J'aimerais aussi remercier le Conseil de recherches en sciences naturelles et en génie (CRSNG), le Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR), l'Institut des sciences mathématiques (ISM) et l'École Polytechnique de Montréal pour leur aide financière.

RÉSUMÉ

De façon générale, l'optimisation globale est une branche de la recherche opérationnelle dont la question fondamentale consiste à trouver un point qui optimise une fonction sur un domaine donné, ou de montrer qu'un tel point n'existe pas.

Ce domaine fut développé par différents chercheurs qui se sont spécialisés dans l'étude de diverses classes de problèmes. Celles-ci sont souvent nées de questions rencontrées par l'industrie. L'analyse de la structure de ces classes a permis la découverte de plusieurs principes fondamentaux et théoriques, sur lesquels reposent maintes méthodes de résolution pour des problèmes spécifiques.

La contribution de cette thèse est d'identifier, à partir de l'étude approfondie de plusieurs classes de problèmes d'optimisation globale, les liens qui unissent ces dernières, tant au niveau de la nature de celles-ci qu'au niveau des méthodes de résolution qui les traitent, afin de transférer, d'approfondir et de développer les connaissances aux plans théorique et algorithmique.

En particulier, parmi maintes reformulations entre diverses classes de problèmes d'optimisation globale, l'équivalence entre les problèmes linéaire maxmin et bilinéaire disjoint permet le développement de deux familles de coupes de concavité, ainsi qu'un algorithme d'énumération exact et fini. Celui-ci repose entre autres sur les conditions d'optimalité (écarts complémentaires, monotonie) des problèmes linéaires maxmin pour résoudre le problème bilinéaire disjoint. Nous montrons que dans le cas général, la vérification de l'existence d'une solution optimale de valeur bornée se réduit au problème NP-complet de PROHIBITION, défini dans cette thèse. Nous indiquons comment l'algorithme peut, lorsqu'appliqué à des problèmes auxiliaires, répondre à cette question de finitude. Des problèmes de taille relativement élevée sont résolus.

L'introduction du concept d'*algorithme plongé* permet la comparaison d'algorithmes définis pour des classes de problèmes différentes. À titre d'exemple, nous

prouvons qu'un algorithme classique pour le problème à variables mixtes binaires est *plongé* dans un autre pour le problème linéaire biniveau via une reformulation entre ces deux classes de problèmes. C'est-à-dire, si l'algorithme pour la classe mixte est appliqué à n'importe quel problème particulier, et que celui pour la classe biniveau est appliqué à la reformulation du problème en question, alors les sous-étapes engendrées par les deux algorithmes seront absolument équivalentes. Cette constatation conduit également à la généralisation, au problème biniveau, d'un test qui était spécifique au problème mixte. Le concept d'*algorithme plongé* permet d'établir une hiérarchie de difficulté à l'intérieur de la classe de problèmes NP-complets qui diffère de l'analyse produite par la théorie classique de la complexité.

Les reformulations entre classes de problèmes nous permettent de spécialiser l'algorithme pour la programmation bilinéaire introduit ci-dessus, à une question importante de la théorie des jeux. L'ensemble des points d'équilibre d'un jeu bimatriciel est formé de l'union de polytopes possiblement disjoints. La connaissance des points extrêmes de ces polytopes est suffisante pour décrire tous les points d'équilibre. L'adaptation de l'algorithme à la question d'énumérer tous les points d'équilibre extrêmes d'un jeu bimatriciel se simplifie à un point tel que la plupart des tests pour le problème bilinéaire deviennent redondants. L'algorithme résultant de cette approche issue de l'optimisation globale permet l'énumération de tous les points d'équilibres pour des jeux dont la taille est significativement plus élevée que ce qui était préalablement connu dans la littérature.

Nous développons une méthode de résolution pour un problème d'optimisation extrêmement général et difficile, soit le problème quadratique non convexe avec contraintes quadratiques. Celui-ci englobe tous les autres problèmes apparaissant dans cette thèse, ce qui lui permet de modéliser un vaste éventail de situations rencontrées par l'industrie. Nous présentons quatre classes d'inégalités valides décrivant l'approximation extérieure des termes quadratiques, et indiquons comment choisir la meilleure coupe selon un critère donné de chaque classe. Les coupes résultantes sont intégrées dans un algorithme d'énumération implicite, convergeant en temps fini vers une solution approchée. La performance de cet algorithme est illustrée sur plusieurs applications réelles, et résout certains problèmes avec une précision inégalée.

ABSTRACT

In a general way, global optimization is a branch of Operations Research whose central question consists in finding a point that optimizes a function on a given domain, or to show that no such point exists.

That area was developped by different research groups that specialized themselves in studying various classes of problems. These classes frequently arose from questions generated by the industry. The analysis of the structure of these classes allowed the discovery of many fondamental and theoretical principles, upon which many solution methods for specific problems are built.

The contribution of this thesis is to identify, from the in-depth study of several global optimization problem classes, the links that unify them. This unification, which arises not only from the nature of the classes, but also from the solution methods defined for them, is studied in order to transfer, enhance and develop theoretical and algorithmic knowledge for these classes.

In particular, among many reformulations of various classes of global optimization problems, the equivalence between the linear maxmin and disjoint bilinear programming problems allows the computation of two families of concavity cuts, as well as an exact and finite branch and bound algorithm. The latter relies on optimality conditions (complementarity slackness and monotonicity) for the linear maxmin reformulations to solve the bilinear problem. We show that in the general case, verifying the existence of a bounded optimal value reduces to the NP-complete FORBIDDANCE problem, defined in this thesis. We indicate how the proposed algorithm can be applied to auxiliary problems in order to answer that question about finiteness. Relatively large sized instances are solved.

The introduction of the *embedded algorithm* concept allows the comparison of algorithms designed for different classes of problems. To that effect, we show that a

classical algorithm for the mixed integer problem is *embedded* in another for the linear bilevel problem through a given reformulation between these two classes of problems. The signification of that statement is that, if the algorithm for the mixed problem is applied to any particular instance, and that the bilevel algorithm is applied to the reformulation of that instance, then the sub-steps generated by both algorithms are absolutely equivalent. That observation also leads to the generalization to the bilevel problem, of a test that was specific to the mixed problem. The *embedded algorithm* concept introduces a hierarchy of difficulty within the class of NP-complete problems that differs from the classical complexity theory analysis.

The reformulation between classes of problems allows the specialization of the above-mentionned algorithm for the bilinear problem to an important game theory question. The set of equilibria of a bimatrix game is the union of polytopes that may be disjoint. Knowledge of every extreme point of these polytopes is sufficient to describe all the equilibrium points. The adaptation of the algorithm to the question of enumerating all extreme equilibria of a bimatrix game simplifies to such an extent that most of the tests become redundant. The algorithm resulting from that global optimization approach allows enumeration for games whose size are significantly greater than that previously known in the literature.

We develop a solution method for an extremely difficult and general optimization problem, that is the non convex quadratically constrained quadratic programming problem. It encompasses every problems appearing in this thesis. The flexibility of this problem allows the modeling of a wide range of situations stemming from the industry. We present four classes of valid inequalities that describe an outer approximation of the quadratic terms, and show how to select the best cut of each class, through a given criterion. The resulting cuts are integrated into a branch and bound algorithm, that converges in finite time to an approximated solution. The performance of the algorithm is illustrated on many real applications, and solves problems to a precision that was never reached.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SYMBOLES

$\|x\|$	La norme euclidienne du vecteur x.
$\|x\|_\infty$	La norme infinie du vecteur x.
$\leqq$	Égal ou bien inférieur à.
$\mathbb{1}$	Vecteur composé entièrement de 1
$Al(P)$	Un algorithme qui résout le problème d'optimisation P.
$\mathcal{M}_{m\times n}$	Ensemble des matrices de m lignes et n colonnes.
$A_{i\cdot}$	La i^e ligne de la matrice $A \in \mathcal{M}_{m\times n}$.
$B_{\cdot j}$	La j^e colonne de la matrice $B \in \mathcal{M}_{m\times n}$.
$B_\epsilon(x)$	Boule ouverte de rayon ϵ centrée en $x \in \mathbb{R}^n$.
$conv(X)$	L'enveloppe convexe de l'ensemble X.
$ext(X)$	L'ensemble des points extrêmes de l'ensemble X.
polyèdre	Ensemble défini par des égalités et inégalités linéaires.
polytope	Polyèdre borné.
s.c.	Abréviation de *sous les contraintes.*

INTRODUCTION

Celui qui accroît sa connaissance accroît sa douleur.
Ecclésiaste I,18

L'optimisation globale est une branche des mathématiques en pleine expansion depuis quelques années. Ce domaine étudie dans un cadre général la question de trouver parmi un grand nombre de situations, la plus satisfaisante selon des critères donnés. La nomenclature habituelle stipule que l'on désire optimiser (maximiser ou minimiser) une fonction sous diverses contraintes. Dès le XVII^{e} siècle, Newton décrit dans *La méthode des fluxions et des suites infinies* des conditions nécessaires pour déterminer la plus grande ou la plus petite valeur d'une fonction continue, bornée et non contrainte.

> *Une quantité qui est devenuë la plus grande ou la moindre qu'il se peut, n'augmente ni ne diminuë, c'est-à-dire, ne flue ni en avant ni en arrière dans cet instant; car si elle augmente, c'est une marque qu'elle était plus petite & que tout à l'heure elle va être plus grande qu'elle n'était, ce qui est contre la supposition, & c'est le contraire si elle diminue. Ainsi trouvez sa Fluxion (...) & supposez la égale à zéro.*

Cette méthode d'annulation du gradient permet de trouver, dans un voisinage réalisable, un optimum local. Elle ne s'applique pas directement lors de la présence de contraintes. La généralisation de cet énoncé définit l'essence de l'optimisation globale : déterminer parmi tous les optima locaux satisfaisant les contraintes, lequel optimise la valeur de la fonction objectif. Pour ce faire, les méthodes proposées explorent explicitement ou implicitement la totalité ou une partie du domaine réalisable déterminé par les contraintes. Ces méthodes reposent habituellement sur la structure connue du problème d'optimisation étudié. En effet, toute information concernant la nature

du problème, telle la convexité de la fonction objectif ou du domaine réalisable, la présence de variables binaires ou de termes quadratiques, etc. inculque une structure qui peut être exploitée afin de faciliter la résolution du problème. Les méthodes actuelles sont définies pour des problèmes ayant une structure précise. Différentes voies de recherches se sont spécialisées dans l'étude de problèmes spécifiques, souvent à la demande de l'industrie qui désire résoudre les problèmes associés à la modélisation de leur environnement. Beaucoup de résultats concernant différentes classes furent obtenus en parallèle. La théorie de la complexité confirme qu'un grand nombre d'entre elles partagent le même degré de difficulté.

Il est possible que différentes classes de problèmes, qui à première vue semblent très éloignées, sont telles que le fondement de leur structure soit similaire. L'essence de la difficulté reliée à diverses classes peut être de la même nature. Il serait avantageux d'étudier les ressemblances entre celles-ci afin de pouvoir transférer les connaissances d'une classe à l'autre.

Pour résoudre un problème précis, on peut souvent le réécrire de façon équivalente comme cas particulier d'une autre classe de problèmes pour laquelle on connaît des méthodes de résolution. Il suffit ensuite de résoudre la reformulation et de transférer le résultat obtenu au problème initial. Cependant, cette approche n'exploite pas la structure particulière du problème d'origine. Une seconde approche consisterait à transférer non pas le problème et la solution, mais plutôt d'adapter les outils de résolution de la seconde classe à la première afin de tirer profit de sa structure particulière. La spécialisation de ces méthodes peut engendrer des outils plus puissants. Pour ce faire, on doit bien comprendre la nature des liens qui unissent les deux classes. Inversement, il est parfois possible de construire des outils pour la seconde classe en s'inspirant ou en généralisant ceux de la première.

Les travaux exécutés au cours de cette thèse s'inscrivent dans ce cadre général. Nous présentons une description détaillée de plusieurs problèmes d'optimisation globale, puis nous exhibons des liens entre eux afin d'illustrer leur structure commune ou bien leur différences fondamentales. Pour ce faire, nous étudions entre autres des conditions d'optimalité locales et globales, des conditions nécessaires ou suffisantes d'existence d'une solution, des principes d'énumération, des approximations

extérieures de domaines, des relations primales-duales, etc. Les résultats provenant de ces analyses permettent l'élaboration de méthodes de résolution pour ces classes de problèmes. Cette thèse comporte cinq chapitres, structurés de la façon suivante.

Le premier chapitre, servant de référence aux autres, comporte deux parties. D'abord la gamme complète des classes de problèmes d'optimisation globale apparaissant dans cette thèse y sont formellement décrits afin d'établir clairement les bases sur lesquelles repose cette étude. Ensuite, les équivalences et les inclusions liant ces classes entre elles sont détaillées. La reformulation d'un problème d'une classe à une autre illustre un degré de similitude existant entre ces deux classes. Ces liens nous permettent d'analyser en profondeur certaines de ces classes de problèmes dans les chapitres qui suivent.

Le second chapitre est consacré à l'étude du problème linéaire maxmin. Celui-ci modélise un système hiérarchisé où deux agents doivent agir successivement selon des intérêts conflictuels. Par l'intermédiaire des reformulations du premier chapitre, deux interprétations fondamentalement différentes sont explicitées. L'une d'elle l'associe au problème linéaire biniveau, et la seconde au problème bilinéaire disjoint. Le chapitre se poursuit avec une discussion sur l'existence d'une solution optimale de valeur bornée et une étude des conditions d'optimalité locale et globale des problèmes maxmin et bilinéaire. Nous montrons que la détermination de la finitude du problème bilinéaire est équivalente au problème NP-complet de PROHIBITION, défini au même chapitre. Nous prouvons que l'approche du problème bilinéaire par ses deux reformulations symétriques équivalentes maxmin n'introduit pas d'optima locaux. Nous étudions ensuite l'évaluation de coupes de concavité associées aux deux reformulations, et discutons des avantages de l'une sur l'autre. Cette étude détaillée du problème bilinéaire permet l'élaboration d'un algorithme d'énumération implicite exact et fini, qui exploite l'équivalence des reformulations maxmin, et qui détermine en temps fini la meilleure solution de valeur bornée. Nous montrons aussi comment utiliser ce même algorithme pour résoudre le problème de PROHIBITION. Son efficacité est illustrée sur des problèmes issus de la littérature et un grand nombre d'autres générés aléatoirement.

Nous introduisons au troisième chapitre le concept d'*algorithme plongé*. Ceci

permet la comparaison de deux algorithmes définis pour des classes de problèmes d'optimisation différentes. Il est possible que l'étude indépendante de ces deux classes ait généré des algorithmes qui, lorsqu'appliqués à deux problèmes équivalents, engendrent des séries de tests, d'étapes et de comparaisons essentiellement identiques. Nous illustrons ce concept à deux algorithmes trouvés indépendamment pour le problème à variables mixtes binaires et le problème biniveau. Ce premier est *plongé* dans le second via la reformulation du premier chapitre en ce sens qu'ils génèrent des suites de sous-problèmes identiques via la même reformulation. Nous discutons des conséquences d'une telle équivalence, de généralisations et spécialisations de tests d'un algorithme à l'autre. À titre d'exemple, nous généralisons au problème biniveau un test qui était spécifique au problème mixte et à sa structure. Il en résulte une certaine hiérarchisation de certaines classes de problèmes appartenant à la famille NP; elles ne sont pas simplement reliées par des reformulations, mais aussi par des algorithmes qui les résolvent.

De nombreux agents économiques sont modélisés et analysés mathématiquement à l'aide de la théorie des jeux. Ce domaine étudie l'interaction stratégique entre divers joueurs. Chacun d'eux doit prendre une décision en anticipant celle des autres. Les stratégies des joueurs sont dites d'*équilibre* si elles sont optimales pour chacun d'eux, c'est-à-dire, aucun joueur n'a intérêt à modifier unilatéralement la sienne. Au quatrième chapitre, les jeux bimatriciels sont analysés sous un point de vue d'optimisation globale. Nous y présentons un algorithme d'énumération implicite qui détermine en temps fini toutes les stratégies d'équilibre d'un tel jeu. Il est connu que la détermination d'une telle stratégie équivaut à la résolution d'un problème bilinéaire. L'algorithme présenté au deuxième chapitre pour la résolution du problème bilinéaire se simplifie énormément et permet l'énumération des points d'équilibre pour des problèmes dont la taille est nettement plus grande que ceux précédemment obtenus dans la littérature.

Le cinquième et dernier chapitre de cette thèse est voué à l'étude de linéarisations pour le problème quadratique non convexe avec contraintes quadratiques. Ce problème est sans aucun doute le plus difficile puisqu'il englobe tous les autres problèmes étudiés dans cette thèse, ainsi que plusieurs autres. Sa flexibilité permet la modélisation d'un grand nombre de situations rencontrées par l'industrie. La

présence de termes quadratiques à la fois dans la fonction objectif et dans les contraintes engendre une grande difficulté de résolution et du même coup la généralité du problème. Nous présentons quatre classes d'inégalités valides servant d'approximation extérieure aux termes quadratiques. Chacune de ces classes permet l'élimination de la solution courante obtenue par la résolution d'une relaxation linéaire. Pour chacune d'elles, nous définissons une façon de déterminer l'inégalité la plus forte, c'est-à-dire, celle qui nous garantit que l'erreur résultante soit minimale (selon un critère précis). Ces coupes, valides sur tout le domaine, sont imbriquées dans un algorithme d'énumération implicite qui détermine en temps fini une solution approchée du problème quadratique à contrainte quadratique. La versatilité de cette classe de problèmes ainsi que la performance de l'algorithme sont illustrées sur de multiples applications et exemples tirés de la littérature.

CHAPITRE 1

Classes de problèmes d'optimisation globale

> Grâce à la difficulté, à la *particularité* qui se présente ici, je bondis d'un seul coup vers le secret, secret que je n'aurais jamais pu atteindre sans la particularité et les inductions qu'elle me fournit...
>
> Edgar Allan Poe

De manière générale, et selon la définition proposée dans l'un des premiers ouvrages voué exclusivement à l'optimisation globale, Horst et Tuy [108], le paradigme de l'optimisation globale se formule formellement ainsi :

> *Étant donné l'ensemble fermé et non vide* $D \subseteq \mathbb{R}^n$ *et la fonction continue* $f : A \to \mathbb{R}$ *où* $D \subseteq A \subseteq \mathbb{R}^n$, *trouver un vecteur* $x^* \in D$ *tel que* $f(x^*) \geq f(x)$ *pour tout* $x \in D$, *ou bien montrer qu'un tel vecteur n'existe pas.*

Nous abrégeons cet énoncé par

$$\max_{x \in D} f(x),$$

signifiant que nous cherchons la solution x^* qui maximise la fonction f sur le domaine réalisable D. La valeur représentée par cette formulation est le nombre réel $f(x^*)$. Dans le cas où le domaine D est vide, nous adoptons la convention classique stipulant que $\max_{x \in D} f(x) = -\infty$. L'énoncé

$$\arg\max_{x \in D} f(x),$$

représente l'ensemble des solutions optimales $\{x \in D : f(x) = f(x^*)\}$.

Il n'existe pas de méthodes résolvant efficacement ce problème dans le cas général. En pratique, la majorité des articles se concentrent sur des classes de problèmes dont une certaine structure est connue, comme par exemple la convexité de la fonction f ou bien celle du domaine D.

L'objectif visé par ce chapitre est d'établir les bases sur lesquelles repose cette thèse. D'abord, nous définissons toutes les classes de problèmes d'optimisation globale rencontrées dans ce document. Ensuite nous exhibons des liens entre plusieurs de ces classes, qui à première vue peuvent sembler fort différentes. Ces similitudes facilitent l'unification et l'étude de ce domaine. Les résultats présentés dans ce chapitre sont parus pour la plupart dans Audet *et al.* [18] [15] (cette deuxième référence est une version détaillée de la première).

1.1 Formulations de problèmes structurés

Tous les problèmes d'optimisation globale rencontrés à l'intérieur de ce manuscrit sont présentés ci-dessous afin de faciliter leur repérage lors de la lecture de ce document. Une revue de la littérature plus approfondie concernant les problèmes que nous jugeons plus importants dans le cadre de cette étude apparaît au début des chapitres ultérieurs.

1.1.1 Problème de programmation linéaire mixte 0-1 MIP_{0-1}

Considérons le problème de programmation linéaire où certaines variables sont restreintes à prendre des valeurs binaires. Le problème de programmation linéaire mixte 0-1 s'écrit :

$$MIP_{0-1} \qquad \begin{array}{ll} \max\limits_{x,u} & c^t x + e^t u \\ \text{s.c.} & Ax + Eu \leq b, \\ & x \geq 0,\ u \in \{0,1\}^{n_u}, \end{array}$$

où $x, c \in \mathbb{R}^{n_x}$; $e, u \in \mathbb{R}^{n_u}$; $b \in \mathbb{R}^m$; $A \in \mathcal{M}_{m \times n_x}$; $E \in \mathcal{M}_{m \times n_u}$.

Nous présentons en détail l'algorithme classique d'énumération implicite de Beale et Small [40] ainsi que le renforcement des pénalités de Tomlin [193] au chapitre 3. Cet algorithme illustre à titre d'exemple le concept d'*algorithme plongé* décrit au même chapitre.

1.1.2 Problème de programmation bilinéaire disjoint $BILD$

Le problème de programmation bilinéaire disjoint fut introduit par Konno [126] comme généralisation de l'approche de Mills [151] pour trouver un point d'équilibre de Nash [159] d'un jeu bimatriciel (ce thème est abordé au chapitre 4 sous un point de vue d'optimisation globale).

Ce problème modélise une situation où le domaine réalisable est défini par deux ensembles disjoints, et où la fonction objectif est bilinéaire (c'est-à-dire, elle devient linéaire lorsque les valeurs d'un des deux groupes de variables sont fixées). Le problème de programmation bilinéaire disjoint se formule :

$$BILD \qquad \begin{array}{ll} \max\limits_{x,u} & c^t x - u^t Q x + u^t d \\ \text{s.c.} & Ax \leq a, \quad u^t B \leq b^t, \\ & x \geq 0, \qquad u \geq 0, \end{array}$$

où $x, c \in \mathbb{R}^{n_x}$; $a \in \mathbb{R}^{n_v}$; $A \in \mathcal{M}_{n_v \times n_x}$; $Q \in \mathcal{M}_{n_u \times n_x}$;
$d, u \in \mathbb{R}^{n_u}$; $b \in \mathbb{R}^{n_y}$; $B \in \mathcal{M}_{n_u \times n_y}$.

Le choix de la notation pour la dimension des vecteurs a et b est motivé par les liens entre ce problème et le problème linéaire maxmin décrits à la section 1.2.

Nous analysons au chapitre 2 divers aspects de ce problème. En particulier nous présentons des conditions nécessaires et suffisantes pour vérifier la finitude de la valeur optimale lorsque le domaine réalisable n'est pas borné. Nous y développons aussi un algorithme exact et fini.

1.1.3 Problème de programmation bilinéaire à contraintes bilinéaires BIL

Le problème $BILD$ se généralise au cas où, tout comme la fonction objectif, les contraintes font intervenir des produits de variables. Il en résulte le problème de programmation bilinéaire à contraintes bilinéaires :

$$BIL \qquad \begin{array}{ll} \max\limits_{x,u} & c^{0t}x - u^tQ^0x + u^td^0 \\ \text{s.c.} & c^{jt}x - u^tQ^jx + u^td^j \leq b_j \qquad j = 1, 2, \ldots, m, \\ & x \geq 0, \qquad u \geq 0, \end{array}$$

où $x, c^j \in \mathbb{R}^{n_x}; d^j, u \in \mathbb{R}^{n_u}; b \in \mathbb{R}^m; Q^j \in \mathcal{M}_{n_u \times n_x}$ pour $j = 0, 1, \ldots, m$.

Ce problème est indirectement étudié au chapitre 5 par l'intermédiaire du problème quadratique avec contraintes quadratiques.

1.1.4 Problème de programmation quadratique concave QP_+

Tout comme le problème bilinéaire disjoint, ce prochain problème comporte des termes quadratiques dans la fonction objectif, sans toutefois séparer les variables en deux groupes. Le problème de programmation quadratique concave se formule :

$$QP_+ \qquad \begin{array}{ll} \max\limits_{x} & c^tx + x^tQx \\ \text{s.c.} & Ax \leq b, \\ & x \geq 0, \end{array}$$

où $x, c \in \mathbb{R}^n; b \in \mathbb{R}^m; A \in \mathcal{M}_{m\times n}$ et où $Q \in \mathcal{M}_{n\times n}$ est une matrice semi-définie positive.

1.1.5 Problème de programmation quadratique avec contraintes quadratiques QQP

En ajoutant des contraintes quadratiques et en éliminant la condition sur le signe de la matrice (c'est-à-dire qu'on ne suppose pas que la matrice soit semi-définie positive), le problème QP_+ se généralise naturellement en le problème de programmation quadratique non convexe avec contraintes quadratiques :

$$QQP \qquad \begin{array}{ll} \max\limits_{x} & c^{0t}x + x^tQ^0x \\ \text{s.c.} & c^{jt}x + x^tQ^jx \leq b_j \qquad j = 1, 2, \ldots, m, \\ & x \geq 0, \end{array}$$

où $x, c^j \in \mathbb{R}^n; b \in \mathbb{R}^m; Q^j \in \mathcal{M}_{n\times n}; j = 0, 1, \ldots, m.$

Le chapitre 5 décrit une méthode d'approximations linéaires successives des termes quadratiques. Il en résulte un algorithme d'énumération implicite jumelé avec des coupes valides sur tout le domaine réalisable.

1.1.6 Problème de complémentarité linéaire généralisée $GLCP$

Un cas particulier du problème QQP est le problème de complémentarité linéaire généralisée présenté par Ibaraki [109] :

$$GLCP \qquad \begin{array}{ll} \min\limits_{x} & c^t x \\ \text{s.c.} & Ax \leq b, \\ & x \geq 0, \\ & Mx + q \geq 0, \\ & x^t(Mx+q) = 0, \end{array}$$

où $x, c, q \in \mathbb{R}^n$; $M \in \mathcal{M}_{n\times n}$; $A \in \mathcal{M}_{m\times n}$; $b \in \mathbb{R}^m$.

Le nom de ce problème provient du problème de réalisabilité de complémentarité linéaire qui consiste à trouver un $x \geq 0$ satisfaisant $Mx+q \geq 0$ et $x^t(Mx+q) = 0$.

1.1.7 Problème de programmation linéaire maxmin LMM

Le problème de programmation linéaire maxmin modélise une situation où deux individus doivent successivement prendre une décision selon leur propre intérêt. Tout comme dans les jeux de Stackelberg [209], le premier joueur doit prendre sa décision en anticipant celle du second. Dans ce modèle maxmin, les intérêts (les fonctions objectifs) des deux joueurs sont diamétralement opposés, et l'ensemble des décisions possibles pour le second joueur dépend de la décision du premier. Le problème de programmation linéaire maxmin s'écrit :

$$LMM \qquad \begin{array}{ll} \max\limits_{x} & c^t x + \left(\begin{array}{ll} \min\limits_{y} & b^t y \\ \text{s.c.} & By \geq d - Qx, \\ & y \geq 0 \end{array} \right) \\ \text{s.c.} & Ax \leq a, \\ & x \geq 0, \end{array}$$

où $x, c \in \mathbb{R}^{n_x}$; $a \in \mathbb{R}^{n_v}$; $A \in \mathcal{M}_{n_v\times n_x}$; $Q \in \mathcal{M}_{n_u\times n_x}$;
$d \in \mathbb{R}^{n_u}$; $y, b \in \mathbb{R}^{n_y}$; $B \in \mathcal{M}_{n_y\times n_u}$.

Cette notation sous-entend que le premier joueur veut maximiser la somme de $c^t x$ et de la valeur optimale du problème de minimisation du second niveau. Le

choix du dimensionnement des vecteurs a et d est motivé par les liens décrits à la section 1.2 qui unissent ce problème au problème bilinéaire disjoint.

Le chapitre 2 est voué à l'étude du problème linéaire maxmin. On y trouve en particulier deux interprétations fondamentalement différentes de ce problème.

1.1.8 Problème de programmation linéaire biniveau BLP

Le problème LMM se généralise dans les fonctions objectifs, et dans les contraintes au problème de programmation linéaire biniveau :

$$
BLP \qquad
\begin{aligned}
\max_{x,y} \quad & c^{1t}x + d^{1t}y \\
\text{s.c.} \quad & A^1x + B^1y \leq b^1, \\
& x \geq 0, \\
& y \in \arg\max_{y} \quad d^{2t}y \\
& \qquad\qquad \text{s.c.} \quad B^2y \leq b^2 - A^2x, \\
& \qquad\qquad\quad\; y \geq 0,
\end{aligned}
$$

où $c^1, x \in \mathbb{R}^{n_x}$; $A^1 \in \mathcal{M}_{m_1 \times n_x}$; $A^2 \in \mathcal{M}_{m_2 \times n_x}$; $b^1 \in \mathbb{R}^{m_1}$,
$d^1, d^2, y \in \mathbb{R}^{n_y}$; $B^1 \in \mathcal{M}_{m_1 \times n_y}$; $B^2 \in \mathcal{M}_{m_2 \times n_y}$; $b^2 \in \mathbb{R}^{m_2}$.

Le polyèdre définit par les contraintes des premier et second niveau se nomme le *domaine réalisable relaxé*. Un point $(\hat{x}, \hat{y})$ du domaine réalisable relaxé tel que $\hat{y}$ est une solution optimale du problème de second niveau est dit *rationnel*. La *relaxation du premier niveau* de BLP est le programme linéaire où l'on maximise $c^{1t}x + d^{1t}y$ sur le domaine réalisable relaxé :

$$
\begin{aligned}
\max_{x,y} \quad & c^{1t}x + d^{1t}y \\
\text{s.c.} \quad & A^1x + B^1y \leq b^1, \\
& A^2x + B^2y \leq b^2, \\
& x \geq 0, \\
& y \geq 0.
\end{aligned}
$$

Ces définitions s'appliquent également au problème LMM.

Plusieurs auteurs considèrent BLP sans contrainte de premier niveau. Bialas et Karwan [47], Bard [31] ainsi que Benson [44] montrent que l'ensemble des solutions rationnelles est alors connexe et est l'union de faces du polytope défini par les contraintes de second niveau.

Savard [175] remarque que ce résultat n'est plus valide lors de la présence de variables de second niveau dans les contraintes de premier niveau. L'exemple suivant illustre que l'ensemble des solutions rationnelles peut même être discret.

Exemple 1.1 *Soit* $(\hat{x}, \hat{y}) \in I\!R^2$ *une solution rationnelle du problème*

$$\begin{array}{lll} \max\limits_{x,y} & x & \\ s.c. & 0 \leq x \leq 1, & \\ & y = 0, & \\ & y \in \arg\max\limits_{y} & y \\ & & s.c. \quad y \leq x, \\ & & \qquad y \leq 1 - x. \end{array}$$

Il en résulte que $\hat{y}$ *doit être à la fois une solution optimale du problème de second niveau et satisfaire la contrainte de premier niveau* $y = 0$. *Ces conditions sont remplies si et seulement si* $\hat{x} \in \{0, 1\}$. *En effet, si* $\hat{x} \in]0, 1[$, *alors la variable* y *est contrainte par le second niveau à être strictement positive (car elle serait égale au minimum entre* $\hat{x}$ *et* $1 - \hat{x}$*), et par le premier à être nulle, ce qui est impossible. L'ensemble des solutions rationnelles contient seulement deux éléments:* $\{(0, 0), (1, 0)\}$. ∎

Cet exemple renferme l'essence du lien décrit à la prochaine section qui associe le problème MIP_{0-1} à BLP. Nous présentons en détail l'algorithme d'énumération implicite de Hansen *et al.* [96] pour BLP au chapitre 3, où nous montrons des liens qui l'unissent à un algorithme classique pour le problème MIP_{0-1}.

1.1.9 Problème de programmation linéaire biniveau, mixte au premier niveau $MIBLP_{0-1}$

Les problèmes MIP_{0-1} et BLP se généralisent tous les deux en problème de programmation linéaire biniveau, mixte au premier niveau :

$$
MIBLP_{0-1} \qquad
\begin{aligned}
\max_{x,y,u} \quad & c^{1t}x + d^{1t}y + e^{1t}u \\
\text{s.c.} \quad & A^1x + B^1y + E^1u \leq b^1, \\
& x \geq 0, \quad u \in \{0,1\}^{n_u}, \\
& y \in \arg\max_{y} \quad d^{2t}y \\
& \qquad\qquad \text{s.c.} \quad B^2y \leq b^2 - A^2x - E^2u, \\
& \qquad\qquad\qquad y \geq 0,
\end{aligned}
$$

où $x, c^1 \in \mathbb{R}^{n_x}$, $A^1 \in \mathcal{M}_{m_1 \times n_x}$, $A^2 \in \mathcal{M}_{m_2 \times n_x}$, $b^1 \in \mathbb{R}^{m_1}$,
$y, d^1, d^2 \in \mathbb{R}^{n_y}$, $B^1 \in \mathcal{M}_{m_1 \times n_y}$, $B^2 \in \mathcal{M}_{m_2 \times n_y}$, $b^2 \in \mathbb{R}^{m_2}$,
$u, e^1 \in \mathbb{R}^{n_u}$, $E^1 \in \mathcal{M}_{m_1 \times n_u}$, $E^2 \in \mathcal{M}_{m_2 \times n_u}$.

Wen et Yang [212] étudient le cas particulier de $MIBLP_{0-1}$ où il n'y a pas de variables continues au premier niveau et où les seules contraintes de premier niveau sont $u \in \{0,1\}^{n_u}$. Leur méthode de résolution est brièvement discutée à la section 3.4.2, pour des fins de comparaison d'algorithmes.

1.2 Liens entre les problèmes

Il est bien connu que tout problème NP-complet se réduit polynomialement en tout autre problème NP-complet. La classe de problèmes NP-dur est plus vaste. Tout problème NP-dur est au moins aussi difficile que n'importe quel problème NP-complet puisque par définition, tout problème NP-complet se réduit polynomialement en un problème NP-dur. La réciproque n'est pas nécessairement vraie, un problème NP-dur peut ne pas être réductible en un problème NP-complet.

1.2.1 Définition de la reformulation

Une *réduction de Turing en temps polynomial* (voir Garey et Johnson [86]) de la classe de problèmes P_A à la classe P_B est un algorithme $Al(P_A)$ résolvant P_A en temps polynomial, en consultant un oracle de temps polynomial pour résoudre P_B. Un tel algorithme est divisé en trois phases. Premièrement, étant donné un problème A de P_A, l'algorithme reformule A en $B(A)$, un problème de P_B. Deuxièmement, l'oracle est consulté afin de résoudre $B(A)$ en temps polynomial. Troisièmement, cette solution est transformée en une solution du problème A.

Mis à part l'oracle, un élément clef de la réduction de Turing en temps polynomial est la reformulation $B(\cdot)$ qui permet le transfert en temps polynomial d'une solution optimale de $B(A)$ en une solution optimale de A. La définition suivante insiste sur cette troisième phase.

Définition 1.2 *Soient P_A et P_B deux classes de problèmes d'optimisation. Une* reformulation $B(\cdot)$ *de P_A en P_B est une transformation de P_A à P_B telle que pour tout problème A de P_A ainsi qu'une solution optimale de $B(A)$, on peut obtenir une solution optimale de A en temps polynomial.*

La reformulation impliquée dans une réduction de Turing en temps polynomial est une transformation en temps polynomial. De façon générale, la définition ci-dessus ne force pas la reformulation à être une transformation polynomiale, et ce même si la solution optimale de A peut être obtenue en temps polynomial à partir de celle de $B(A)$. Par exemple, Balas, Ceria et Cornuéjols [26], Lovász et Schrijver [138] et Sherali et Adams [179] présentent des méthodes reformulant des problèmes en nombres entiers en problèmes linéaires équivalents en générant toutes les facettes du domaine réalisable. Les problèmes linéaires contiennent un nombre exponentiel de contraintes, mais le transfert de la solution optimale est trivial.

Une reformulation est entièrement caractérisée par P_A et $B(P_A)$. La figure 1.1 schématise chacune des reformulations présentées ci-dessous entre les problèmes décrits à la section précédente. Certaines reformulations nécessitent l'ajout de constantes de grande taille (mais de taille bornée). Celles-ci sont représentées à l'aide de flèches doubles.

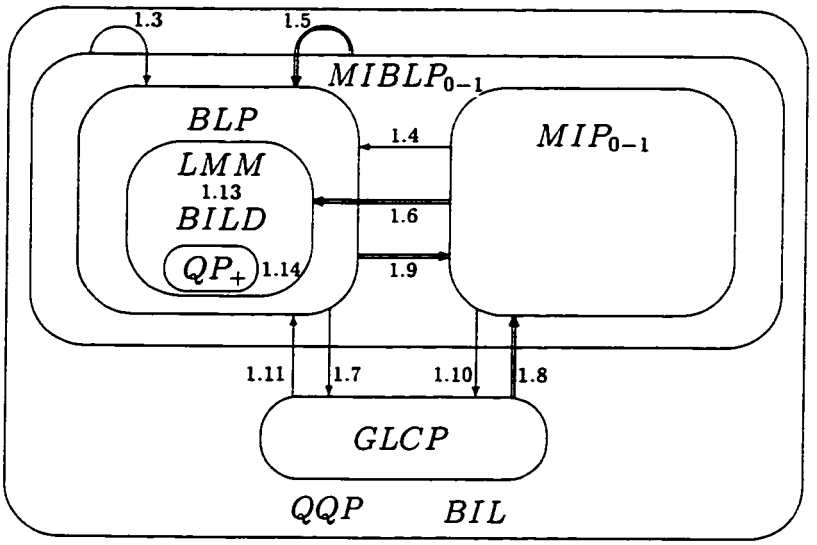


Figure 1.1 – *Reformulations des problèmes*

L'ordre dans lequel les reformulations sont présentées apparaît dans la figure. Les reformulations triviales, telle l'inclusion de $GLCP$ dans QQP ne sont pas explicitées. La seule inclusion ne découlant pas simplement des définitions est celle de QP_+ dans $BILD$ décrite à la proposition 1.14.

1.2.2 Reformulations des problèmes

Chacun des deux problèmes MIP_{0-1} et $MIBLP_{0-1}$ est reformulé de deux façons distinctes en BLP. La seconde paire de reformulations introduit une grande constante de taille finie L qui dépend des paramètres définissant le problème. La

première paire n'implique pas de telle constante.

Proposition 1.3 (Reformulation $MIBLP_{0-1} \rightarrow BLP$)
Si (x^*, y^*, u^*) est une solution optimale de $MIBLP_{0-1}$, alors $(x^*, y^*, u^*, 0)$ est une solution optimale de

$$\begin{aligned}
\max_{x,y,u,v} \quad & c^{1t}x + d^{1t}y + e^{1t}u \\
s.c. \quad & A^1x + B^1y + E^1u \leq b^1, \\
& 0 \leq u \leq \mathbb{1}, \\
& x \geq 0, \\
& v = 0, \\
(y, v) \in \quad & \arg\max_{y,v} \quad d^{2t}y + \mathbb{1}^t v \\
& \qquad s.c. \quad B^2y \leq b^2 - A^2x - E^2u, \\
& \qquad\quad y \geq 0, \\
& \qquad\quad v \leq u, \\
& \qquad\quad v \leq \mathbb{1} - u.
\end{aligned}$$

De plus, si (x^*, y^*, u^*, v^*) est une solution optimale de ce dernier problème, alors (x^*, y^*, u^*) est une solution optimale de $MIBLP_{0-1}$, et $v^* = 0$.

Démonstration Les contraintes de premier niveau de cette reformulation assure que toute solution rationnelle $(\hat{x}, \hat{y}, \hat{u}, \hat{v})$ satisfait $\hat{v} = 0$. De plus, la rationalité de la solution implique que $\hat{v} = \min\{\hat{u}, \mathbb{1} - \hat{u}\}$, et donc $\hat{u} \in \{0,1\}^{n_u}$. Il s'ensuit que la solution $(\hat{x}, \hat{y}, \hat{u})$ est rationnelle pour $MIBLP_{0-1}$.

Inversement, à toute solution rationnelle $(\hat{x}, \hat{y}, \hat{u})$ de $MIBLP_{0-1}$, correspond la solution rationnelle $(\hat{x}, \hat{y}, \hat{u}, \hat{v})$ de la reformulation $MIBLP_{0-1} \rightarrow BLP$ où $\hat{v} = 0$. Le résultat suit du fait qu'une solution optimale est rationnelle. ∎

Un corollaire direct suit de la spécialisation de ce résultat au problème MIP_{0-1}.

Corollaire 1.4 (Reformulation $MIP_{0-1} \rightarrow BLP$)
Si (x^*, u^*) est une solution optimale de MIP_{0-1}, alors $(x^*, u^*, 0)$ est une solution

optimale de

$$\begin{aligned}
\max_{x,u,v} \quad & c^t x + e^t u \\
s.c. \quad & Ax + Eu \leq b, \\
& 0 \leq u \leq \mathbb{1}, \\
& x \geq 0, \\
& v = 0, \\
& v \in \arg\max_{v} \quad \mathbb{1}^t v \\
& \qquad s.c. \quad v \leq u, \\
& \qquad\qquad v \leq \mathbb{1} - u.
\end{aligned}$$

De plus, si (x^*, u^*, v^*) *est une solution optimale de ce dernier problème, alors* (x^*, u^*) *est une solution optimale de MIP, et* $v^* = 0$. ∎

L'autre paire de reformulations (de problèmes ayant des variables binaires) en problème à deux niveaux requiert l'introduction de constantes supplémentaires. Au lieu de forcer au premier niveau la variable v à être nulle, elle est fortement pénalisée dans la fonction objectif de premier niveau.

La proposition suivante généralise légèrement le résultat de Vicente *et al.* [205] défini pour le cas où toutes les variables de premier niveau de $MIBLP_{0-1}$ sont contraintes à être binaires. La preuve étant similaire est omise.

Proposition 1.5 (Reformulation $MIBLP_{0-1} \Rightarrow BLP$)
Il existe une constante $L > 0$ *de taille finie telle que si* (x^*, y^*, u^*) *est une solution*

optimale de $MIBLP_{0-1}$, alors $(x^*, y^*, u^*, 0)$ est une solution optimale de

$$\begin{aligned}
\max_{x,y,u,v} \quad & c^{1t}x + d^{1t}y + e^{1t}u - L\mathbb{1}^t v \\
s.c. \quad & A^1x + B^1y + E^1u \leq b^1, \\
& 0 \leq u \leq 1, \\
& x \geq 0, \\
& (y, v) \in \arg\max_{y,v} \quad d^{2t}y + \mathbb{1}^t v \\
& \qquad\qquad s.c. \quad B^2y \leq b^2 - A^2x - E^2u, \\
& \qquad\qquad\quad y \geq 0, \\
& \qquad\qquad\quad v \leq u, \\
& \qquad\qquad\quad v \leq \mathbb{1} - u.
\end{aligned}$$

De plus, si (x^*, y^*, u^*, v^*) est une solution optimale de ce dernier problème pour un tel L, alors (x^*, y^*, u^*) est une solution optimale de $MIBLP_{0-1}$ et $v^* = 0$. ∎

La constante L doit être suffisamment grande afin d'assurer que les valeurs de la fonction objectif évaluées aux points extrêmes du domaine réalisable relaxé, où $v \neq 0$ soient inférieurs à ceux où $v = 0$. Ces dernières valeurs dépendent de L, contrairement aux premières. Il en résulte que la détermination de la valeur de L peut se faire en principe par une méthode d'énumération de points extrêmes. En pratique, des valeurs croissantes de L sont générées jusqu'à ce qu'une solution rationnelle $(\hat{x}, \hat{y}, \hat{u}, \hat{v})$ avec $\hat{v} = 0$ soit obtenue. La grande taille potentielle de L peut engendrer des difficultés numériques.

Tout comme précédemment, cette reformulation se spécialise directement au problème MIP_{0-1}.

Corollaire 1.6 (Reformulation $MIP_{0-1} \Rightarrow LMM$)
Il existe une constante $L > 0$ de taille finie telle que si (x^*, u^*) est une solution

optimale de MIP_{0-1}, *alors* $(x^*, u^*, 0)$ *est une solution optimale de*

$$\begin{aligned}
\max_{x,u,v} \quad & c^t x + e^t u - L\mathbb{1}^t v \\
s.c. \quad & Ax + Eu \leq b, \\
& 0 \leq u \leq \mathbb{1}, \\
& x \geq 0, \\
& v \in \arg\max_{v} \quad \mathbb{1}^t v \\
& \qquad s.c. \quad v \leq u, \\
& \qquad\qquad v \leq \mathbb{1} - u.
\end{aligned}$$

De plus, si (x^*, u^*, v^*) *est une solution optimale de ce dernier problème pour un tel* L, *alors* (x^*, u^*) *est une solution optimale de* MIP_{0-1} *et* $v^* = 0$. ∎

Le problème résultant de cette reformulation appartient à la classe LMM puisque la variable v n'apparaît pas dans les contraintes de premier niveau, et les fonctions objectifs des deux niveaux sont diamétralement opposées (on peut remplacer, sans modifier la nature du problème, la fonction objectif du second niveau par $L\mathbb{1}^t v$).

Vicente *et al.* [205] montrent que le problème de programmation linéaire biniveau avec des variables binaires au second niveau ne se reformule pas en un problème triniveau en utilisant une astuce semblable à celle de la reformulation $MIBLP_{0-1} \Rightarrow BLP$, puisque l'existence d'une constante finie L ne peut être assurée.

La reformulation $MIBLP_{0-1} \rightarrow BLP$ n'implique aucune constante L, et donc des arguments similaires établissent que le problème de programmation linéaire à n-niveaux à variables mixtes $0-1$ à tous les niveaux se reformule en un problème de programmation linéaire à $(n+1)$-niveaux. L'intégralité de toutes les variables binaires u est assurée en imposant via le prochain niveau que le minimum de u et $1-u$ soit égal à 0, tout comme dans la reformulation ci-dessus. Le problème de programmation linéaire triniveau est étudié par, e.g., Bard [32], Wen et Bialas [211] et White [215], et le problème de programmation linéaire multiniveau par, e.g., Blair [48], Bard et Falk [35] ainsi que par Benson [44].

Fortuny-Amat et McCarl [81] proposent une reformulation du problème biniveau (comprenant des termes bilinéaires dans les fonctions objectifs) en MIP_{0-1}. Leur reformulation se fait en deux étapes, soit par l'intermédiaire du problème $GLCP$. L'adaptation de leur résultat au problème BLP donne comme première étape la reformulation du problème BLP en $GLCP$, où le second niveau est remplacé par des conditions d'optimalité équivalentes.

Proposition 1.7 (Reformulation $BLP \rightarrow GLCP$)
Si (x^*, y^*) est une solution optimale de BLP, alors il existe $\lambda^* \in \mathbb{R}^{m_2}$ tel que (x^*, y^*, λ^*) est une solution optimale de

$$\begin{array}{lll} \max\limits_{x,y,\lambda} & c^{1t}x + d^{1t}y & \\ s.c. & A^1x + B^1y \leq b^1, & \lambda^t B^2 \geq d^{2t}, \\ & x \geq 0, & \lambda \geq 0, \\ & A^2x + B^2y \leq b^2, & \lambda^t(b^2 - A^2x - B^2y) = 0. \\ & y \geq 0, & y^t(B^{2t}\lambda - d^2) = 0. \end{array}$$

De plus, si (x^*, y^*, λ^*) est une solution optimale de ce dernier problème, alors (x^*, y^*) est une solution optimale de BLP.

Démonstration Pour x donné, le problème de second niveau de BLP s'écrit respectivement sous forme primale et duale :

$$\begin{array}{lll} \max\limits_{y} & d^{2t}y & \\ \text{s.c.} & B^2y \leq b^2 - A^2x, & \\ & y \geq 0 & \end{array} \qquad \begin{array}{lll} \min\limits_{\lambda} & \lambda^t(b^2 - A^2x) & \\ \text{s.c.} & \lambda^t B^2 \geq d^{2t}, & \\ & \lambda \geq 0. & \end{array}$$

Le théorème des écarts complémentaires assure qu'à toute solution optimale, les conditions

$$\lambda^t(b^2 - A^2x - B^2y) = 0 \quad \text{et} \quad (\lambda^t B^2 - d^{2t})y = 0.$$

sont respectées. Le résultat suit en remplaçant le problème de second niveau par ces conditions et par les contraintes de réalisabilité primale et duale. La structure

de complémentarité linéaire généralisée est plus apparente suite aux substitutions suivantes

$$x \leftarrow \begin{bmatrix} x \\ y \\ \lambda \end{bmatrix}, \; c \leftarrow \begin{bmatrix} c^1 \\ d^1 \\ 0 \end{bmatrix}, \; M \leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B^{2t} \\ -A^2 & -B^2 & 0 \end{bmatrix}, q \leftarrow \begin{bmatrix} 0 \\ -d^2 \\ b^2 \end{bmatrix},$$

$$A \leftarrow [A^1 \; B^1 \; 0], \quad \text{et} \quad b \leftarrow b^1. \qquad \blacksquare$$

La deuxième étape proposée par Fortuny-Amat et McCarl [81], consiste à reformuler le problème $GLCP$ en MIP_{0-1} en exploitant le caractère binaire des conditions d'optimalité.

Proposition 1.8 (Reformulation $GLCP \Rightarrow MIP_{0-1}$)
Supposons que la valeur optimale de $GLCP$ soit bornée, et soit x^* une solution optimale finie. Il existe une constante $L > 0$ finie et $u^* \in \mathbb{R}^n$ tel que (x^*, u^*) est une solution optimale de

$$\begin{array}{lll} \max\limits_{x,u} & c^t x & \\ s.c. & Ax \leq b, & \\ & x \geq 0, & Mx + q \geq 0, \\ & x \leq Lu, & Mx + q \leq L(\mathbb{1} - u), \\ & u \in \{0,1\}^n. & \end{array}$$

De plus, pour un tel L, si (x^*, u^*) est une solution optimale de ce dernier problème, alors x^* est une solution optimale de $GLCP$.

Démonstration Soient $\tilde{x}$ une solution optimale finie de $GLCP$ et (x^*, u^*) une solution optimale du problème reformulé pour $L = \max\{\|\tilde{x}\|_\infty, \|M\tilde{x} + q\|_\infty\}$.

Pour chaque i de $\{1, 2, \ldots, n_u\}$, posons $\tilde{u}_i = 1$ si $x_i^* > 0$, et $\tilde{u}_i = 0$ autrement. Il s'ensuit que $\tilde{x} \leq L\tilde{u}$, et $M\tilde{x} + q \leq L(\mathbb{1} - \tilde{u})$ et donc $c^t x^* \geq c^t \tilde{x}$ car $(\tilde{x}, \tilde{u})$ est réalisable pour la reformulation. Inversement, (x^*, u^*) satisfait $x^*(Mx^* + q) = 0$, et donc x^* satisfait les contraintes de $GLCP$, d'où l'égalité $c^t x^* = c^t \tilde{x}$ $\blacksquare$

La combinaison des deux reformulations $BLP \rightarrow GLCP$ et $GLCP \Rightarrow MIP_{0-1}$ donne la suivante.

Corollaire 1.9 (Reformulation $BLP \Rightarrow MIP_{0-1}$)
Supposons que la valeur optimale de BLP soit bornée, et soit (x^*, y^*) une solution optimale finie. Il existe une constante $L > 0$ finie et des vecteurs $\lambda^* \in \mathbb{R}^{m_2}$, $u^* \in \mathbb{R}^{n_u}$ et $v^* \in \mathbb{R}^{n_u}$ tel que $(x^*, y^*, \lambda^*, x^*, u^*)$ est une solution optimale de

$$\begin{aligned}
\max_{x,y,\lambda,u,v} \quad & c^{1t}x + d^{1t}y \\
s.c. \quad & A^1x + B^1y \leq b^1, \\
& x \geq 0, \\
& A^2x + B^2y \leq b^2, && -B^{2t}\lambda \leq -d^2, \\
& y \geq 0, && \lambda \geq 0, \\
& -A^2x - B^2y + Lu \leq L\mathbb{1} - b^2, && \lambda - Lu \leq 0, \\
& y + Lv \leq L\mathbb{1}, && B^{2t}\lambda - Lv \leq d^2, \\
& u \in \{0,1\}^{m_2}, && v \in \{0,1\}^{n_v}.
\end{aligned}$$

De plus, pour un tel L, si $(x^*, y^*, \lambda^*, x^*, u^*)$ est une solution optimale de ce dernier problème, alors (x^*, y^*) est une solution optimale de BLP. ∎

Il y a des réciproques à ces dernières reformulations. Júdice et Mitra [120] montrent que le problème MIP_{0-1} se reformule en $GLCP$.

Proposition 1.10 (Reformulation $MIP_{0-1} \rightarrow GLCP$)
Une solution (x^*, u^*) est optimale pour MIP_{0-1} si et seulement si elle est une solution optimale de

$$\begin{aligned}
\max_{x,u} \quad & c^t x + e^t u \\
s.c. \quad & Ax + Eu \leq b, \\
& x \geq 0, \\
& 0 \leq u \leq \mathbb{1}, \\
& u^t(\mathbb{1} - u) = 0.
\end{aligned}$$

Démonstration Une variable u satisfait la contrainte $u \in \{0,1\}^{n_u}$ si et seulement si elle satisfait $0 \leq u \leq \mathbb{1}$ et $u^t(\mathbb{1} - u) = 0$. ∎

À son tour, le problème $GLCP$ peut être reformulé en BLP.

Proposition 1.11 (Reformulation $GLCP \rightarrow BLP$)
Si x^* est une solution optimale de GLCP, alors $(x^*, 0)$ est une solution optimale de

$$\begin{aligned}
\max_{x,v} \quad & c^t x \\
s.c. \quad & Ax \leq b, \\
& x \geq 0, \\
& Mx + q \geq 0, \\
& v = 0, \\
& v \in \arg\max_{v} \quad \mathbb{1}^t v \\
& \qquad s.c. \quad v \leq x, \\
& \qquad\qquad v \leq Mx + q.
\end{aligned}$$

De plus, si (x^*, v^*) est une solution optimale de ce dernier problème, alors x^* est une solution optimale de GLCP, et $v^* = 0$.

Démonstration Une solution $(\hat{x}, \hat{v})$ est rationnelle si et seulement si elle satisfait $\hat{v} = \min\{\hat{x}, M\hat{x} + q\} = 0$. ∎

Remarquons que si les deux reformulations $MIP_{0-1} \rightarrow GLCP$ et $GLCP \rightarrow BLP$ étaient successivement appliquées, alors le résultat serait identique à la reformulation $MIP_{0-1} \rightarrow BLP$.

On pourrait être tenté de croire, en se basant sur la reformulation $MIP_{0-1} \Rightarrow LMM$, que le problème $GLCP$ puisse être reformulé en BLP en pénalisant la variable v dans la fonction objectif du premier niveau, au lieu de la forcer à être nulle.

Toutefois, la reformulation

$$\begin{aligned}
\max_{x,v} \quad & c^t x - L\mathbb{1}^t v \\
\text{s.c.} \quad & Ax \leq b, \\
& x \geq 0, \\
& Mx + q \geq 0, \\
& v \in \arg\max_{v} \quad L\mathbb{1}^t v \\
& \qquad \text{s.c.} \quad v \leq x, \\
& \qquad\qquad v \leq Mx + q,
\end{aligned}$$

où $L > 0$ est une constante finie, n'est pas toujours valide comme l'illustre l'exemple suivant.

Exemple 1.12 *Soit le problème $GLCP$ dans R^1 et sa reformulation biniveau suggérée*

$$\begin{aligned}
\max_{x \in \mathbb{R}^1} \quad & x \\
s.c. \quad & 0x \leq 0, \\
& x \geq 0, \\
& 0x + 1 \geq 0, \\
& x^t(0x + 1) = 0,
\end{aligned}
\qquad\qquad
\begin{aligned}
\max_{x,v \in \mathbb{R}^1} \quad & x - Lv \\
s.c. \quad & 0x \leq 0, \\
& x \geq 0, \\
& 0x + 1 \geq 0, \\
& v \in \arg\max_{v \in \mathbb{R}^1} \quad Lv \\
& \qquad s.c. \quad v \leq x, \\
& \qquad\qquad v \leq 0x + 1.
\end{aligned}$$

On peut voir aisément que $x^* = 0$ est la seule solution optimale de $GLCP$, et que la valeur de la fonction objectif du problème biniveau devient non bornée lorsque la solution rationnelle (x, v) tend vers $(\infty, 1)$. Il n'existe pas de constante finie L pour laquelle la reformulation suggérée est valide. ∎

Ce contre-exemple découle du fait que la valeur optimale de la relaxation du premier niveau de la reformulation biniveau suggérée ci-dessus n'est pas nécessairement bornée.

Les inclusions de $GLCP$ (et donc de $MIBLP_{0-1}$ via BLP) et de BIL dans QQP sont immédiates. Celles de QQP en BIL s'obtient en ajoutant la variable

y dans $\mathbb{R}^n$, que l'on impose d'être égale à x, et en remplaçant x^tQ^jx par x^tQ^jy pour chaque indice j. Hansen et Jaumard [94] présentent d'autres reformulations qui minimisent l'ajout de variables ou de contraintes supplémentaires.

L'équivalence entre les problèmes LMM et $BILD$ est brièvement esquissée ci-dessous. Elle est reprise en détail au chapitre 2 où d'autres liens sont exprimés entre ces problèmes. Afin d'alléger et d'uniformiser la présentation des résultats, nous introduisons la fonction $h : \mathbb{R}^{n_x+n_u} \rightarrow \mathbb{R}$, où $h(x,u) = c^tx - u^tQx + u^td$, les ensembles suivants

$$\begin{aligned} X &= \{x \geq 0 : Ax \leq a\}, \qquad Y(x) = \{y \geq 0 : By \geq d - Qx\}, \\ U &= \{u \geq 0 : u^tB \leq b\}, \qquad V(u) = \{v \geq 0 : v^tA \geq c^t - u^tQ\}, \end{aligned}$$

et les programmes linéaires obtenus en fixant l'une ou l'autre des variables x ou u de $BILD$

$$\begin{aligned} LP_u(x) &= c^tx + \max_{u \in U} u^t(d - Qx), \\ LP_x(u) &= u^td + \max_{x \in X}(c^t - u^tQ)x. \end{aligned}$$

Les problèmes duaux s'écrivent

$$\begin{aligned} D_y(x) &= c^tx + \min_{y \in Y(x)} b^ty, \\ D_v(u) &= u^td + \min_{v \in V(u)} v^ta. \end{aligned}$$

On peut aisément vérifier que ces quatre fonctions $LP_u(x), LP_u(u)$, $D_y(x)$ et $D_v(u)$ sont linéaires par morceaux et convexes en les variables x et u (elles sont des fonctions implicites des variables u, x, y ou v). Le lecteur peut se référer aux travaux de Pardalos et Rosen [162] et de Benson [45] pour des discussions concernant l'optimisation concave (la maximisation de fonctions convexes dans notre cas). Le problème $BILD$ s'écrit

$$\max_{x \in X} LP_u(x) = \max_{u \in U} LP_x(u),$$

ou bien

$$\max_{x \in X} D_y(x) = \max_{u \in U} D_v(u).$$

Proposition 1.13 (Reformulation $BILD \leftrightarrow LMM$)
La solution $(x^*,\ u^*)$ est une solution optimale bornée de $BILD$, si et seulement si x^* est une solution optimale bornée de $\max_{x\in X} D_y(x)$, et u^* est une solution optimale bornée de $\max_{u\in U} D_v(u)$.

Démonstration Soit (x^*, u^*) une solution optimale de $BILD$. Par hypothèse, x^* et u^* sont bornés. Il s'ensuit que $x^* \in \arg LP_x(u^*)$ et $u^* \in \arg LP_u(x^*)$ La théorie de la dualité nous assure qu'il existe une solution optimale bornée y^* de $D_y(x^*)$.

Supposons que la solution x^* ne soit pas optimale pour $\max_{x\in X} D_y(x)$. Soit $\hat{x}$ une solution de $\max_{x\in X} D_y(x)$ satisfaisant $D_y(\hat{x}) > D_y(x^*)$. Choisissons $\hat{u}$ dans $argLP_u(\hat{x})$ et $\hat{y}$ dans $argD_y(\hat{x})$. La solution réalisable $(\hat{x},\ \hat{u})$ de $BILD$ satisfait

$$c^t\hat{x} - \hat{u}^tQ\hat{x} + \hat{u}^td \ = \ c^t\hat{x} + b^t\hat{y} \ > \ c^tx^* + b^ty^* \ = \ c^tx^* - u^{*t}Qx^* + u^{*t}d,$$

ce qui contredit l'optimalité de (x^*, u^*).

Le résultat inverse se déduit par des arguments similaires. Soient x^* une solution optimale de $\max_{x\in X} D_y(x)$, et u^* une solution optimale de $\max_{u\in U} D_v(u)$. Supposons que $(\hat{x}, \hat{u})$ soit une solution réalisable de $BILD$ telle que $h(\hat{x}, \hat{u}) > h(x^*,\ u^*)$. Choisissons y^* parmi $argD_y(x^*)$ et $\hat{y}$ parmi $argD_y(\hat{x})$. Il s'ensuit que

$$c^t\hat{x} + b^t\hat{y} \ = \ c^t\hat{x} - \hat{u}^tQ\hat{x} + \hat{u}^td \ > \ c^tx^* - u^{*t}Qx^* + u^{*t}d \ = \ c^tx^* + b^ty^*.$$

L'optimalité de la solution $(x^*,\ y^*)$ est contredite. ∎

Les deux reformulations LMM de $BILD$ symétriques obtenues à partir des problèmes duaux $D_y(x)$ et $D_v(u)$ s'écrivent explicitement

$$LMM_{xy} \qquad \begin{array}{ll} \max\limits_{x} & c^tx + \left(\begin{array}{ll} \min\limits_{y} & b^ty \\ \text{s.c.} & By \geq d - Qx, \\ & y \geq 0 \end{array} \right) \\ \text{s.c.} & Ax \leq a, \\ & x \geq 0, \end{array}$$

$$LMM_{uv} \qquad \begin{array}{lll} \max\limits_{u} & u^t d + \left(\begin{array}{ll} \min\limits_{v} & v^t a \\ \text{s.c.} & v^t A \geq c^t - u^t Q, \\ & v \geq 0 \end{array} \right) & \\ \text{s.c.} & u^t B \leq b^t, & \\ & u \geq 0, & \end{array}$$

Nous reprenons ici la preuve de Konno [128] qui montre que le problème QP_+ se reformule en $BILD$.

Proposition 1.14 (Reformulation $QP_+ \rightarrow BILD$)
Si x^* est une solution optimale de QP_+, alors (x^*, x^*) est une solution optimale de

$$\max_{\substack{x \in X \\ u \in X}} \; (h(x,u) \;=\; \frac{1}{2} c^t x + u^t Q x + \frac{1}{2} u^t c).$$

De plus, si (x^*, u^*) est une solution optimale de ce dernier problème, alors x^* et u^* sont deux solutions optimales de QP_+.

Démonstration Soient $\tilde{x}$ une solution optimale de QP_+ de valeur $\tilde{z}$, et (x^*, u^*) une solution optimale de cette reformulation de valeur z^*. On peut aisément vérifier que $z^* \geq \tilde{z}$ puisque $(\tilde{x}, \tilde{x})$ satisfait les contraintes de cette reformulation.

L'addition des deux inégalités valides

$$\begin{aligned} 0 \;\leq\; h(x^*,u^*) - h(x^*,x^*) \;&=\; \frac{1}{2} c^t (u^* - x^*) + x^{*t} Q (u^* - x^*) \\ 0 \;\leq\; h(x^*,u^*) - h(u^*,u^*) \;&=\; \frac{1}{2} c^t (x^* - u^*) + u^{*t} Q (x^* - u^*) \end{aligned}$$

donne $(u^* - x^*)^t Q (u^* - x^*) \leq 0$. Puisque la matrice carré Q est semi-définie positive et symétrique, il s'ensuit que $Q(u^* - x^*) = 0$, d'où $x^{*t} Q = u^{*t} Q$ et $c^t x^* = c^t u^*$. L'inégalité

$$\tilde{z} \;=\; c^t \tilde{x} + \tilde{x}^t Q \tilde{x} \;\geq\; c^t x^* + x^{*t} Q x^* \;=\; \frac{1}{2}(c^t x^* + u^{*t} c) + u^{*t} Q x^* \;=\; z^*$$

entraîne le résultat. ∎

1.3 Discussion

Nous avons présenté dans ce chapitre plusieurs classes de problèmes structurés d'optimisation globale. Ils furent généralement introduits indépendamment afin de modéliser de façon naturelle différentes réalités. Les reformulations entre ces classes, permettent d'entrevoir que même si leur nature est distincte, plusieurs d'entre elles partagent une structure intrinsèque commune. Dans les chapitres qui suivent, nous adaptons, transférons, généralisons ou spécialisons des résultats inhérents d'une classe à une autre.

CHAPITRE 2

Le problème maxmin

D'ailleurs, les caves sont aussi cons qu'les cons sont caves.
Plume Latraverse

La première formulation du problème de programmation linéaire maxmin est due à Falk [71]. Nous débutons ce chapitre par la présentation de cette formulation et discutons de deux interprétations fondamentalement différentes. Le problème

$$\max_x \min_y \{c^t x + b^t y : (x, y) \in P\},$$

où P est un polyèdre de $\mathbb{R}^{n_x+n_y}$ et $c, x \in \mathbb{R}^{n_x}, b, y \in \mathbb{R}^{n_y}$, se réécrit

$$\max_{x \in P_X} \{c^t x + \min_{y \in Y(x)} b^t y\},$$

où $P_X = \{x \in \mathbb{R}^{n_x} : \exists y$ pour lequel $(x, y) \in P\}$ et $Y(x) = \{y \in \mathbb{R}^{n_y} : (x, y) \in P\}$ sont des projections de P. Le premier niveau doit choisir le x de P_X qui lorsque combiné avec la réaction du second niveau y de $Y(x)$, maximise la fonction objectif $c^t x + b^t y$. La figure 2.1 illustre un problème particulier qui est repris ci-dessous afin de mettre en évidence les différentes interprétations. La solution optimale associée à la formulation précédente est atteinte au point (x^c, y^c).

Nous étudions une formulation plus générale du problème qui s'approche davantage d'un problème considéré par Lutzenko et Martynov [140] :

$$LMM \qquad \max_{x \in X} \{c^t x + \min_{y \in Y(x)} b^t y\},$$

où X est un polyèdre de $\mathbb{R}^{n_x}$ qui n'est pas nécessairement la projection P_X de P. Le polyèdre $Y(x)$ est à nouveau défini implicitement par l'entremise de P. Cette formulation admet plus de flexibilité pour les contraintes de premier niveau.

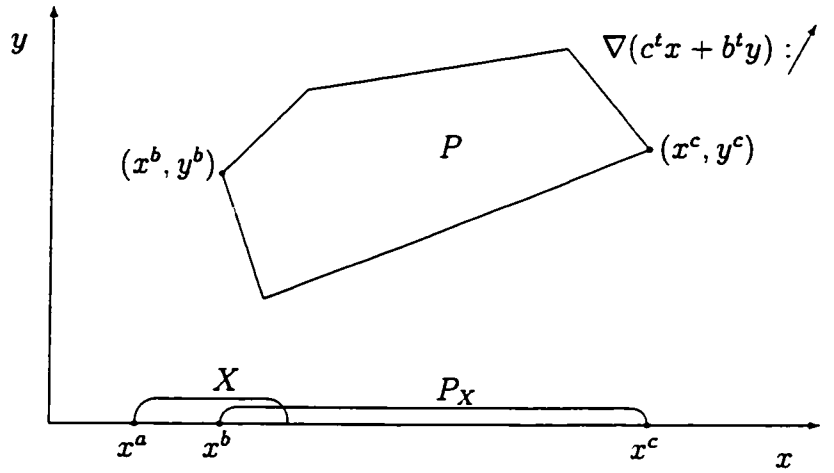


Figure 2.1 – *Un problème LMM*

Lorsque le polyèdre X n'est pas un sous-ensemble de la projection P_X, il y a deux interprétations possible de LMM. Dans un tel cas, il existe un point x de X faisant en sorte que $Y(x)$ est vide. Les cas possibles sont ceux où l'on interdit, et où l'on permet au premier niveau de choisir un tel point. Ce premier problème s'apparente au problème biniveau, tandis que ce dernier s'approche du problème bilinéaire disjoint.

Ce chapitre est structuré de la façon suivante. La prochaine section décrit les deux interprétations du problème LMM et présente une revue de la littérature. La suite du chapitre est consacrée à l'étude du problème $BILD$ par l'intermédiaire des deux reformulations symétriques maxmin. La section 2.2 porte sur l'existence d'une solution optimale de valeur bornée, et une méthode de vérification de la présence d'une telle solution est proposée. On démontre aussi qu'une telle vérification définit un problème fortement NP-complet. À la section 2.3, on étudie les liens entre les optima locaux des problèmes $BILD$ et LMM. La section 2.4 présente et compare deux types de coupes de concavité [194] pour le problème LMM. Ces coupes démontrent l'intérêt des reformulations équivalentes entre les problèmes LMM et $BILD$. La section 2.5 décrit un algorithme d'énumération implicite exact et fini

pour le problème $BILD$, dont la performance est illustrée sur plusieurs problèmes. Tout au long de ce chapitre nous utilisons la notation introduite à la fin de la section 1.2.2. Le développement des coupes de concavité est paru dans Audet *et al.* [14], et la quasi totalité des autres résultats de ce chapitre dans Audet *et al.* [16].

2.1 Le problème linéaire maxmin : revue de la littérature

Les deux interprétations du problème LMM ont des liens étroits avec différents problèmes d'optimisation. Le cas où l'on interdit au premier niveau de choisir une variable x dans X qui rend le polyèdre $Y(x)$ vide s'inscrit dans le cadre de la programmation biniveau. Le cas ou un tel choix est permis s'identifie à la programmation bilinéaire disjointe. Dans cette section nous décrivons ces deux approches, et présentons une revue de la littérature de ces deux problèmes.

2.1.1 Interprétation biniveau de LMM

Considérons le cas où le premier niveau doit choisir une variable qui n'annihile pas le domaine réalisable du second niveau. Une formulation plus précise de LMM serait alors une des suivantes

$$\max_{x \in X \cap P_X} (c^t x + \min_{y \in Y(x)} b^t y), \quad \text{ou} \quad \begin{array}{ll} \max\limits_{x,y} & c^t x + b^t y \\ \text{s.c.} & x \in X, \\ & y \in \arg\min\limits_{y} \quad b^t y \\ & \qquad \text{s.c.} \quad (x,y) \in P. \end{array}$$

Revenons à la figure 2.1, où l'ensemble X n'est pas inclus dans P_X (l'introduction de l'ensemble X vise à définir un problème plus général que celui considéré par Falk ci-dessus). Même si x^a appartient à X, le premier niveau ne peut pas le choisir puisque l'ensemble $Y(x^a)$ est vide. La solution optimale, associée à cette formulation biniveau, de ce problème est atteinte au point (x^b, y^b).

Le problème BLP (voir la section 1.1.8) généralise LMM en permettant à la variable de second niveau y de paraître dans les contraintes de premier niveau, et en ne contraignant pas les fonctions objectif à être diamétralement opposées. Ces deux généralisations entraînent d'importantes différences de modélisation.

La présence de la variable y au premier niveau peut avoir un effet rétroactif: certaines des décisions du premier niveau, lorsque combinées avec la réaction du second peuvent violer les contraintes du premier niveau. Contrairement au LMM, par le fait que les fonctions objectif puissent ne pas être opposées, il est possible que pour le choix x du premier niveau, plusieurs solutions de $Y(x)$ soient toutes aussi intéressantes pour le second niveau (l'ensemble des solutions optimales du second niveau n'est pas un singleton). La formulation du problème BLP fait en sorte qu'il y ait une coopération implicite entre les deux niveaux. Parmi toutes les solutions optimales du second niveau, le premier niveau peut choisir celle qu'il préfère, c'est-à-dire, celle qui maximise $b^{1t}y$. Loridan et Morgan [137] discutent de l'existence et de la stabilité d'une solution d'un problème BLP lorsque l'ensemble des solutions optimales du second niveau en comporte plusieurs.

Jeroslow [117], Ben-Ayed et Blair [42] et Bard [34] montrent que BLP est NP-dur. Ce résultat est renforcé par Hansen *et al.* [96] qui prouvent que le problème LMM, et conséquemment BLP, est fortement NP-dur. Il n'existe donc pas de méthodes d'approximation complètement polynomiales, sauf si P = NP. Vicente et Calamai [202] présentent une revue bibliographique, et Ben-Ayed [41] une revue de la programmation biniveau. Des thèmes voisins sont abordés par Migdalas et Pardalos [148], Du et Pardalos [66] et Pardalos [161].

Falk [71] propose un algorithme d'énumération implicite exact et fini pour résoudre cette formulation biniveau du problème LMM. L'idée générale consiste à obtenir une solution rationnelle, et ensuite à diviser le problème en autant de sous-problèmes qu'il y a de variables en base. La solution rationnelle donne une borne inférieure, et la relaxation de premier niveau donne une borne supérieure. Dans chaque sous-problème, la variable correspondante en base est forcée à la quitter, c'est-à-dire, elle est fixée à zéro. Tous ces sous-problèmes LMM ont alors une

variable de moins. L'algorithme est récursivement appliqué à ces sous-problèmes jusqu'à ce qu'ils soient tous résolus à l'optimalité ou bien jusqu'à ce qu'ils deviennent non réalisables.

Plusieurs algorithmes résolvent BLP. Parmi les plus efficaces sont ceux de Hansen *et al.* [96], Júdice et Faustino [119] et Bard et Moore [36]. Ces deux derniers résolvent la reformulation $BLP \rightarrow GLCP$. Júdice et Faustino construisent une suite de problèmes de complémentarité linéaire qui converge vers une solution $\epsilon-$optimale de BLP. Bard et Moore utilisent un algorithme d'énumération implicite basé sur le caractère binaire de $GLCP$. Hansen *et al.* traitent directement le problème BLP à l'aide d'un algorithme d'énumération implicite. Ce dernier utilise des pénalités et exprime des conditions d'optimalité en relations logiques. Cet algorithme est détaillé à la section 3.2.1 où nous montrons les liens étroits qui l'unissent à celui de Beale et Small [40] pour le problème MIP_{0-1}. Gendreau *et al.* [87] développent une méthode de descente de type Tabou au problème où il n'y a pas de contraintes au premier niveau.

2.1.2 Interprétation bilinéaire de LMM

Considérons maintenant le cas où l'on permet au premier niveau de prendre une décision qui annihile le domaine réalisable du second niveau. Revenons une dernière fois à la figure 2.1. N'importe quel x choisi dans l'intervalle $[x^a, x^b[$ entraîne une solution optimale de valeur non bornée. En effet, le problème de minimisation est non réalisable, et par convention sa valeur est $+\infty$. Le premier niveau a alors tout intérêt à choisir x dans cet intervalle.

En remplaçant le problème de second niveau par son dual on obtient la reformulation $LMM \leftrightarrow BILD$. La paramétrisation en x n'apparaît plus dans les contraintes mais dans la fonction objectif. La signification bilinéaire de l'existence d'un x annulant $Y(x)$ est qu'il existe un rayon extrémal de U dans la direction duquel la valeur de la fonction objectif $u^t(d - Qx)$ croît vers l'infini.

Ivanilov et Mukhamediev [111] proposent un algorithme de résolution du problème LMM via sa reformulation bilinéaire. L'algorithme consiste à générer une suite de points extrêmes qui sont des solutions rationnelles et dont la valeur de la fonction objectif croît strictement. Des itérations de type Gauss-Seidel où l'on résout alternativement les problèmes linéaires $LP_u(x)$ et $LP_x(u)$ (voir la proposition 2.19), donnent la solution réalisable $(\hat{x}, \hat{u})$ de la formulation $BILD$, à partir de laquelle la solution rationnelle $(\hat{x}, \hat{y})$ de LMM est obtenue. La coupe $c^t x + d^t y \geq \gamma$, où γ est la valeur de la fonction objectif de la solution $(\hat{x}, \hat{y})$, est ajoutée au polytope de second niveau $Y(x)$. Le procédé est réitéré avec ce nouveau problème LMM. L'hypothèse selon laquelle la nouvelle solution aura une valeur objectif strictement plus grande que γ n'est pas discutée par les auteurs.

Plusieurs méthodes résolvent le problème $BILD$. Toutes sauf celle décrite par Thieu [190], requièrent que les ensembles X et U soient bornés. Des revues de la littérature concernant $BILD$ peuvent être trouvées dans les travaux d'Al-Khayyal [7] [8] et Floudas et Visweswaran [80].

Une première classe de méthodes comprend les algorithmes de plans de coupure. Konno [127] propose un tel algorithme obtenant une solution ϵ-optimale. La convergence globale vers la solution optimale n'est pas assurée dans tous les cas. L'algorithme est constitué de deux étapes principales qui sont répétées jusqu'à l'obtention d'une solution de valeur ϵ-optimale. La première étape consiste à trouver une solution réalisable par les itérations de type Gauss-Seidel décrites précédemment. La deuxième étape sert à calculer une coupe de concavité à partir de la solution obtenue (nous discutons de ces coupes à la section 2.4). La coupe retranche des solutions dont la valeur dans la fonction objectif peut dépasser de ϵ la meilleure valeur obtenue à date. Rares sont les articles détaillant des résultat numériques. Konno présente des résultats obtenus avec une dizaine de problèmes de taille relativement petite. Le plus gros problème résolu est tel que les ensembles disjoints ont respectivement 10 et 13 variables apparaissant dans 22 et 24 contraintes.

Vaish et Shetty [198] proposent un algorithme essentiellement identique à celui de Konno discuté précédemment. Cependant, ils affirment fallacieusement que l'algorithme converge vers le point optimal. Leur raisonnement repose sur l'argumentation

suivante :

> *L'algorithme engendre une suite de points $\{x_i\}_{i\geq 1}$ appartenant tous à un ensemble compact; il existe alors un point d'accumulation x_k appartenant à cette suite.*

En effet, par le théorème de Weierstrass, on peut affirmer qu'il existe un point d'accumulation, mais on ne peut certainement pas conclure que ce point appartient à la suite $\{x_i\}_{i\geq 1}$.

La convergence de ces deux derniers algorithmes n'est pas nécessairement finie. Sherali et Shetty [184] montrent qu'une convergence finie peut être obtenue en ajoutant des coupes disjonctives. Cependant, de telles coupes sont coûteuses à générer.

L'approximation polyédrale définit une deuxième classe de méthodes. Vaish et Shetty [197] proposent un algorithme d'approximation intérieure qui converge en temps fini. Ils mentionnent que des difficultés numériques apparaissent rapidement lorsque la taille du problème augmente. Gallo et Ülkücü [85] présentent un algorithme combinant les plans de coupures et l'approximation intérieure, mais pour lequel la convergence n'est pas finie. Thieu [190] développe une approche d'approximation extérieure utilisant la convexité de la fonction paramétrée $LP_y(\cdot)$ pour résoudre en temps fini le problème $BILD$.

Plus récemment, White [214] résout la reformulation $BILD \leftrightarrow LMM$ de $BILD$ via une suite finie de problèmes linéaires dont les vecteurs de contraintes sont successivement déterminés par une énumération d'optima d'un programme linéaire maître.

2.2 Existence d'une solution optimale de valeur bornée

À la section 2.5.3, nous présentons un algorithme résolvant le problème de

programmation

$$\max_{\substack{x \in ext(X) \\ u \in ext(U)}} h(x,u).$$

L'optimisation de la fonction h se fait non pas sur la totalité des polyèdres X et U, mais sur les points extrêmes de ceux-ci. La solution optimale de ce problème résout le problème initial $BILD$ si et seulement si la valeur optimale de $BILD$ est bornée. En effet, si tel est le cas, il existera nécessairement une solution optimale composée de points extrêmes. Il est alors important de vérifier la finitude de cette valeur avant d'appliquer l'algorithme.

La réalisabilité de $BILD$ se vérifie en temps polynomial en s'assurant que les polyèdres X et U ne sont pas vides. L'existence d'une solution optimale de valeur bornée est cependant beaucoup plus difficile à vérifier.

On peut vérifier en temps polynomial que les polyèdres X et U sont bornés. S'ils le sont tous les deux, alors la valeur optimale l'est nécessairement aussi. Nous étudions dans un premier temps le cas où un seul d'entre eux est borné. Ensuite, nous développons le cas où ni l'un ni l'autre n'est borné.

2.2.1 Domaine partiellement borné

La symétrie entre les variables x et u nous permet de restreindre l'analyse au cas où X est borné et U est non borné. Ce chapitre contient plusieurs résultats symétriques entre les variables x et u. Sans aucune perte de généralité, une seule direction est développée dans les preuves.

Proposition 2.1 *Supposons que X soit borné et U non borné. La valeur optimale z^* de $BILD$ est bornée si et seulement si pour tout x de X le polyèdre $Y(x)$ est non vide.*

Démonstration Le résultat suit des équivalences suivantes :

$$\begin{aligned} z^* < \infty &\iff \text{pour tout } x \text{ de } X, \text{ la valeur optimale de } LP_u(x) \text{ est bornée} \\ &\iff \text{pour tout } x \text{ dans } X, \text{ la valeur optimale de } D_y(x) \text{ est bornée} \\ &\iff \text{pour tout } x \text{ dans } X,\ Y(x) \neq \emptyset. \end{aligned}$$

La dernière équivalence provient du fait que la valeur $c^t x$ est bornée pour tout x de X. ∎

Si l'ensemble X est borné mais U ne l'est pas, alors vérifier que la valeur optimale de $BILD$ est bornée se réduit à vérifier s'il existe un x de X tel que le polyèdre $Y(x)$ est vide. Cette question définit le problème *PROHIBITION* (pour n'importe quel polyèdre X et n'importe quel polyèdre projeté $Y(x)$). Hansen *et al.* [96] montrent que le problème LMM est fortement NP-dur. Le preuve du résultat suivant s'inspire de la leur.

Théorème 2.2 *Le problème PROHIBITION est fortement NP-complet.*

Démonstration Pour tout x de X donné, on peut déterminer en temps polynomial si le polyèdre $Y(x)$ est vide en appliquant la phase I d'un algorithme de programmation linéaire (sous l'hypothèse que x soit représentable sur l'ordinateur). Il s'ensuit que le problème PROHIBITION est dans NP.

Pour montrer que le problème PROHIBITION est NP-complet, nous utilisons une réduction de NOYAU – aussi connu sous le nom anglais de *KERNEL* –. Chvátal [58] prouve que ce problème est NP-complet. Le problème du NOYAU se formule ainsi :

> *Étant donné le graphe orienté $G = (V,\ E)$, existe-t-il un sous-ensemble $K \subseteq V$ de sommets à la fois stables (c'est-à-dire qu'aucune paire de sommets de K ne soient adjacents) et absorbants (c'est-à-dire que tout sommet $v_\ell \in V \setminus K$ soit à l'origine d'un arc $(v_\ell,\ v_k)$ de E où $v_k \in K$)?*

Considérons les inégalités

$$x_k + x_\ell \leq 1 \qquad (v_\ell,\ v_k) \in E, \tag{2.1}$$

$$x \geq 0, \tag{2.2}$$

$$x_k + y_k \leq 1 \qquad k = 1, 2, \ldots, |V|, \tag{2.3}$$

$$x_k + y_\ell \leq 1 \qquad (v_\ell,\ v_k) \in E, \tag{2.4}$$

$$\sum_{k=1}^{|V|} y_k \geq \frac{1}{2}, \tag{2.5}$$

$$y \geq 0, \tag{2.6}$$

où X est défini par (2.1), (2.2) et $Y(x)$ par (2.3)–(2.6). Supposons que G possède un noyau K. Fixons

$$x_k = \begin{cases} 1 & \text{si } v_k \in K, \\ 0 & \text{sinon.} \end{cases}$$

Les contraintes (2.1) et (2.2) sont nécessairement satisfaites. De plus, il s'ensuit que $y_k = 0$ pour tout les k satisfaisant $v_k \in K$ (par la contrainte (2.3)) et que $y_\ell = 0$ pour tout les ℓ satisfaisant $v_\ell \in V \setminus K$ (par la contrainte (2.4)). Et donc $y = 0$, ce qui viole (2.5). Alors pour un tel x, l'ensemble $Y(x)$ est vide.

Supposons maintenant que G ne possède pas de noyau. Considérons un vecteur x satisfaisant (2.1), (2.2) et soit L l'ensemble des sommets $v_k \in V$ tels que k satisfait $x_k > \frac{1}{2}$. La contrainte (2.1) assure que L est un ensemble stable de G. Comme G ne possède pas de noyau, il existe un sommet $v_\ell \in V \setminus L$ qui n'est pas le sommet origine d'un arc se terminant à un sommet de L. Conséquemment, la construction de L assure que $x_\ell \leq \frac{1}{2}$ et $x_k \leq \frac{1}{2}$ pour chaque x_k satisfaisant $(v_\ell,\ v_k) \in E$. Il s'ensuit que la solution où $y_\ell = \frac{1}{2}$ et $y_k = 0$, pour $k \neq \ell$, satisfait les contraintes (2.3)–(2.6), et donc $Y(x)$ n'est pas vide.

Nous avons montré que la réponse au problème PROHIBITION est *oui* si et seulement si la réponse au problème NOYAU est *oui*. De plus, le NOYAU n'est pas un problème de nombres (voir Garey et Johnson [86]), alors le problème PROHIBITION

est fortement NP-complet. ∎

Corollaire 2.3 *Vérifier la finitude de la valeur optimale de $BILD$ définit un problème fortement NP-complet.*

Démonstration Le résultat découle directement de la proposition 2.1 et du théorème 2.2. ∎

La proposition suivante conduit à une méthode de résolution pour le problème PROHIBITION, et conséquemment de la vérification de la finitude de la valeur optimale de $BILD$.

Proposition 2.4 *Soit X un polytope (polyèdre borné). La valeur optimale de $BILD$ est non bornée si et seulement si celle du problème auxiliaire*

$$\begin{array}{rl} \max\limits_{x\in X,\ u} & u^t(d-Qx) \\ s.c. & u^tB \leq 0, \\ & u^t\mathbb{1} \leq 1, \\ & u \geq 0, \end{array} \tag{2.7}$$

est strictement positive.

Démonstration Le problème PROHIBITION est vérifié si et seulement si la valeur optimale du problème linéaire maxmin

$$\max_{x\in X}\min_{y\in Y(x)} 0 \;=\; \max_{x\in X}\left(\begin{array}{ll} \min\limits_{y} & 0 \\ \text{s.c.} & By \geq d-Qx, \\ & y \geq 0, \end{array}\right)$$

est non bornée. En remplaçant le problème de second niveau par son dual on obtient

$$\begin{array}{rl} \max\limits_{x\in X,\ u} & u^t(d-Qx) \\ \text{s.c.} & u^tB \leq 0, \\ & u \geq 0. \end{array}$$

Le domaine réalisable de la variable u est un cône. Le résultat suit en observant que pour toute valeur de x fixée, la fonction $u^t(d - Qx)$ est linéaire par rapport à la variable u. La contrainte $u^t\mathbb{1} \leq 1$ est introduite uniquement pour borner le domaine réalisable (un problème dont le domaine réalisable est borné se résout plus facilement que s'il ne l'était pas). ∎

Des arguments similaires montrent que si U est borné, la valeur optimale de $BILD$ est non bornée si et seulement si celle du problème auxiliaire

$$\begin{aligned} \max_{x,\ u \in U} \quad & (c^t - u^t Q)x \\ \text{s.c.} \quad & Ax \leq 0, \\ & \mathbb{1}^t x \leq 1, \\ & x \geq 0, \end{aligned} \tag{2.8}$$

est strictement positive. L'algorithme présenté à la section 2.5.3 peut être utilisé pour vérifier si la valeur optimale de $BILD$ est bornée puisque le domaine réalisable des problèmes (2.7) et (2.8) est borné par rapport aux variables x et u.

2.2.2 Domaine non borné

L'existence d'une solution optimale de valeur bornée de $BILD$ est encore plus difficile à vérifier lorsqu'aucun des deux ensembles X et U n'est borné. Dans ce cas, des difficultés supplémentaires interviennent lorsque le problème PROHIBITION est vérifié seulement pour des solutions non bornées.

La proposition suivante décrit des conditions suffisantes pour que la valeur optimale du problème $BILD$ soit non bornée. La preuve est omise puisqu'elle est

très semblable à celle de la proposition 2.4.

Proposition 2.5 *Si la valeur optimale d'au moins un des problèmes bilinéaires*

$$\begin{array}{lll} \max\limits_{x\in ext(X),\ u} & u^t(d-Qx) & \max\limits_{x,\ u\in ext(U)} \quad (c^t-u^tQ)x \\ s.c. & u^tB \leq 0, & s.c. \quad Ax \leq 0, \\ & u^t\mathbb{1} \leq 1, & \quad\ \mathbb{1}^tx \leq 1, \\ & u \geq 0, & \quad\ x \geq 0, \end{array} \tag{2.9}$$

est strictement positive, alors la valeur optimale de $BILD$ est non bornée.

Le problème PROHIBITION est alors résolu pour des ensembles restreints, c'est-à-dire que la proposition vérifie s'il existe un x de $ext(X)$ ou un u de $ext(U)$ tel que $Y(x) = \emptyset$ ou $V(u) = \emptyset$. L'exemple suivant montre que ces tests ne sont pas suffisants pour vérifier la finitude de la valeur optimale de $BILD$ lorsque les deux ensembles X et U ne sont pas bornés.

Exemple 2.6 *Considérons le problème bilinéaire disjoint où x et u appartiennent à $\mathbb{R}^1$:*

$$\max_{\substack{x\in X=\{x\geq 0\}\\ u\in U=\{u\geq 0\}}} ux.$$

Les deux problèmes apparaissant dans (2.9) ont une valeur optimale 0. ∎

Le critère suivant révèle que la valeur optimale de cet exemple est bel et bien bornée.

Proposition 2.7 *Si la valeur optimale du problème bilinéaire disjoint*

$$\begin{array}{lll} \max\limits_{x,\ u} & -u^tQx & \\ s.c. & Ax \leq 0, & u^tB \leq 0, \\ & \mathbb{1}^tx \leq 1, & u^t\mathbb{1} \leq 1, \\ & x \geq 0, & u \geq 0, \end{array} \tag{2.10}$$

est strictement positive, alors la valeur optimale de $BILD$ est non bornée.

Démonstration Soient x^0 et u^0 des points extrêmes de X et U, et $(\hat{x}, \hat{u})$ une solution réalisable de (2.10) dont la valeur de la fonction objectif est strictement positive. Pour tout scalaire $\theta > 0$ la solution $(x^0 + \theta\hat{x}, u^0 + \theta\hat{u})$ est réalisable pour $BILD$ et la valeur correspondante de la fonction objectif est

$$h(x^0,\ u^0) + \theta((c^t - u^{0t}Q)\hat{x} + \hat{u}^t(d - Qx^0)) - \theta^2\hat{u}^tQ\hat{x}.$$

Lorsque θ tend vers infini, la valeur de la fonction objectif devient non bornée. ∎

À nouveau, ces tests peuvent être appliqués par l'intermédiaire de l'algorithme de la section 2.5.3 puisque le domaine réalisable des problèmes (2.9) et (2.10) est borné. Montrons maintenant que ces tests définissent des conditions nécessaires et suffisantes pour la vérification de la finitude de la valeur optimale de $BILD$.

Théorème 2.8 *Soient X et U des polyèdres non bornés. La valeur optimale de $BILD$ est non bornée si et seulement si au moins une des valeur optimale des problèmes apparaissant dans (2.9) et (2.10) est strictement positive.*

Démonstration Seule l'implication directe reste à être démontrée. Soit une suite $\{(\hat{x}^k,\ \hat{u}^k)\}_{k=0}^{\infty}$ de solutions réalisables convergeant vers une solution optimale de $BILD$, c'est-à-dire, $\lim_{k\to\infty} h(\hat{x}^k,\ \hat{u}^k) = \infty$. Sans aucune perte de généralité, supposons que $\lim_{k\to\infty} \|\hat{x}^k\| = \infty$. Soit x^0 une combinaison linéaire de sommets et r_x un rayon extrémal unitaire de X tels que les points $x^k = x^0 + kr_x$ pour $k \geq 0$ convergent aussi à $\lim_{k\to\infty} \hat{x}^k$.

En un premier temps, examinons le cas où $\lim_{k\to\infty} \|\hat{u}^k\| = \infty$. Soit u^0 une combinaison linéaire de sommets et r_u un rayon extrémal unitaire de U tels que les points $u^k = u^0 + kr_u$ pour $k \geq 0$ convergent aussi à $\lim_{k\to\infty} \hat{u}^k$. La valeur de la fonction objectif s'écrit

$$h(x^k,\ u^k) = h(x^0,\ u^0) + k((c - u^{0t}Q)r_x + r_u^t(d - Qx^0)) - k^2r_u^tQr_x.$$

Il s'ensuit que $-r_u^tQr_x \geq 0$. Si $-r_u^tQr_x > 0$, alors la valeur optimale de (2.10) est strictement positive. Autrement, si $-r_u^tQr_x = 0$, alors $(c - u^{0t}Q)r_x + r_u^t(d - Qx^0) >$

0. La valeur optimale d'un des problèmes apparaissant dans (2.9) est strictement positive.

En un deuxième temps, examinons le cas où $\lim_{k\to\infty} \|\hat{u}^k\| < \infty$. La solution $u^* = \lim_{k\to\infty} u^k$ appartient à U. Écrivons $u^* = u^0 + r_u$, où u^0 appartient à l'enveloppe convexe de $ext(U)$ et où r_u est une combinaison convexe des rayons extrémaux de U. Considérons la suite $\{(x^k,\ u^*)\}_{k=0}^{\infty}$. La valeur de la fonction objectif est

$$h(x^k,\ u^*) = h(x^0,\ u^*) + k(c - u^{*t}Q)r_x.$$

Il s'ensuit que $0 < (c - u^{*t}Q)r_x = (c - u^{0t}Q)r_x - r_uQr_x$. Et donc, la valeur optimale du second problème de (2.9) ou celle du problème (2.10) est strictement positive. ∎

Nous avons prouvé qu'il est fortement NP-complet de vérifier si le problème initial $BILD$ possède une solution optimale de valeur bornée. Tout algorithme qui résout $BILD$ sous l'hypothèse que le domaine réalisable est borné peut être utilisé pour résoudre (2.7) et (2.8) mais pas pour les problèmes (2.9) et (2.10). Dans ces derniers cas, l'algorithme décrit à la section 2.5.3 peut être utilisé puisqu'il retourne le point extrême dont la valeur objectif est la plus élevée pour le problème bilinéaire considéré.

2.3 Optimalité locale et globale

Comme mentionné précédemment, nous pouvons nous restreindre au cas où il existe une solution optimale de valeur bornée de $BILD$.

L'introduction des reformulations linéaires maxmin de $BILD$ est motivée par le fait qu'elles n'ajoutent pas à la complexité du problème. La proposition 1.13 assure qu'il existe une bijection entre les optima globaux de $BILD$ et des reformulations linéaires maxmin. De plus, pour chaque optimum local d'une des reformulations, il existe un optimum local de $BILD$. La réciproque n'est pas nécessairement vraie.

Proposition 2.9 *Pour chaque solution localement optimale x^* du problème LMM_{xy}. il existe un point u^* de U, tel que $(x^*,\ u^*)$ est une solution localement optimale de $BILD$.*

Démonstration Soit x^* une solution localement optimale de LMM_{xy}, il existe alors un $\epsilon > 0$ tel que $D_y(x^*) \geq D_y(\hat{x})$ pour tout $\hat{x}$ de $X \cap B_\epsilon(x^*)$. Soit u^* une solution optimale de $LP_u(x^*)$. Pour tout $(\hat{x},\ \hat{u}) \in (X \times U) \cap B_\epsilon(x^*,\ u^*)$, l'inégalité suivante est valide

$$\begin{aligned} c^t\hat{x} - \hat{u}^tQ\hat{x} + \hat{u}^td \ \leq\ LP_u(\hat{x}) \ &=\ D_y(\hat{x}) \\ \text{(puisque } \hat{x} \in X \cap B_\epsilon(x^*)) \qquad &\leq\ D_y(x^*) \ =\ LP_u(x^*) \ =\ c^tx^* - u^{*t}Qx^* + u^{*t}d, \end{aligned}$$

et donc $(x^*,\ u^*)$ est une solution localement optimale de $BILD$. ∎

L'exemple suivant montre qu'il peut y avoir une solution localement optimale $(x^*,\ u^*)$ de $BILD$ telle que x^* n'est pas une solution localement optimale de LMM_{xy} et u^* n'est pas une solution localement optimale de LMM_{uv}.

Exemple 2.10 *Considérons le problème bilinéaire où les variables x et u appartiennent à $\mathbb{R}^2$:*

$$\begin{aligned} \max_{x,\ u} \quad & h(x,\ u) = -x_1 + 2x_1u_1 + 2x_2u_2 - u_2 \\ s.c. \quad & 0 \leq x \leq \mathbb{1}, \quad 0 \leq u \leq \mathbb{1}. \end{aligned}$$

La solution $(x^*,\ u^*) = (0, 0)$ est une solution localement optimale puisque pour tout $\epsilon = (\epsilon_1,\ \epsilon_2) \geq 0$ et $\delta = (\delta_1,\ \delta_2) \geq 0$ assez petit, $h(\epsilon,\ \delta) = \epsilon_1(2\delta_1 - 1) + \delta_2(2\epsilon_2 - 1) \leq 0$. Cependant, x^* n'est pas une solution localement optimale de LMM_{xy} puisque pour tout $\epsilon_1 > 0$

$$D_y(\epsilon_1,\ 0) \ =\ -\epsilon_1 + \left(\begin{array}{ll} \min\limits_{y \geq 0} & y_1 + y_2 \\ s.c. & y_1 \geq 2\epsilon_1, \\ & y_2 \geq -1 \end{array} \right) = \epsilon_1 > 0 = D_y(0, 0).$$

Et de plus, u^* n'est pas une solution localement optimale de LMM_{uv} puisque pour tout $\delta_2 > 0$

$$D_v(0,\ \delta_2) \ = \ -\delta_2 + \left(\begin{array}{ll} \min\limits_{v \geq 0} & v_1 + v_2 \\ s.c. & v_1 \geq -1, \\ & v_2 \geq 2\delta_2 \end{array} \right) \ = \ \delta_2 > 0 = D_v(0,0).$$

■

Afin d'assurer l'existence d'une solution localement optimale pour une reformulation linéaire maxmin de $BILD$, de plus fortes hypothèses doivent être posées.

Proposition 2.11 *Si* $(x^*,\ u^*)$ *est une solution localement optimale de* $BILD$ *telle que l'optimalité locale est stricte par rapport à la variable* x*, c'est-à-dire, la fonction objectif* h *de* $BILD$ *satisfait*

$$h(x,\ u) \leq h(x^*,\ u^*) \quad \forall x \in X \cap B_\epsilon(x^*), \quad \forall u \in U \cap B_\epsilon(u^*), \tag{2.11}$$

$$h(x^*,\ u) < h(x^*,\ u^*) \quad \forall u \in U \cap B_\epsilon(u^*), \quad u \neq u^*, \tag{2.12}$$

alors x^* *est une solution localement optimale de* LMM_{xy}.

Démonstration Soit une suite de points $\{x^k\}_{k=0}^{\infty}$ de X convergeant vers x^*, et choisissons $\hat{u}$ un point de U.

Si $\hat{u} \in B_\epsilon(u^*)$, alors choisissons l'indice k assez grand pour que $x^k \in B_\epsilon(x^*)$. L'inégalité (2.11) implique que $h(x^k,\ \hat{u}) \leq h(x^*,\ u^*)$.

Autrement $\hat{u} \in U \setminus B_\epsilon(u^*)$. Lorsque la variable x^* est fixée, la fonction $h(x^*, \cdot)$ est linéaire et alors l'inégalité (2.12) assure que la différence δ est strictement positive :

$$\delta \ \equiv \ h(x^*, u^*) - h(x^*, \hat{u}) \ > \ 0. \tag{2.13}$$

La figure 2.2 illustre cette construction. Choisissons l'indice k assez grand pour que $|h(x^k,\ \hat{u}) - h(x^*,\ \hat{u})| < \delta$. En combinant ceci avec (2.13) on obtient $h(x^k,\ \hat{u}) < h(x^*,\ \hat{u}) + \delta = h(x^*,\ u^*)$.

Il en résulte que x^* est une solution localement optimale de LMM_{xy}. ■

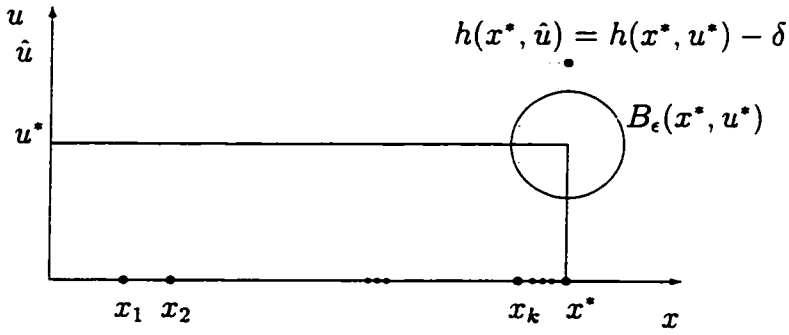


Figure 2.2 – *Construction pour l'optimalité locale de BILD*

Les propositions précédentes montrent que l'utilisation des reformulations linéaires maxmin de $BILD$ n'accroissent pas le nombre d'optima locaux. Leur nombre peut même décroître. De ce point de vue, l'utilisation des reformulations ne complexifie pas davantage le problème.

2.4 Coupes de concavité

Les coupes de concavité furent introduites par Tuy [194] pour des problèmes de minimisation concave (d'où leur nom). Elles servent à éliminer du domaine réalisable une région ne contenant pas de meilleure solution que celle qui est connue. Considérons le problème général

$$\max_{x \in X} f(x)$$

où f est une fonction convexe définie sur le polyèdre X. Soient $\hat{x}$ un sommet de X et γ un réel tels que

$$f(x) \leq \gamma \text{ pour tout } x \in B_\epsilon(\hat{x}) \cap X,$$

pour un $\epsilon > 0$. Une coupe de concavité est une inégalité $\pi^t(x - \hat{x}) \geq 1$ telle que si $x \in X$ et $\pi^t(x - \hat{x}) < 1$, alors $f(x) \leq \gamma$. Alors, pour vérifier s'il existe un $x \in X$ dont la valeur de la fonction objectif dépasse γ, on peut se limiter au polyèdre réduit $\{x \in X : \pi^t(x - \hat{x}) \geq 1\}$.

Après une revue de la génération de coupes de concavité inspirée du livre de Horst et Tuy [108], nous présentons dans cette section des coupes de concavité pour les deux reformulations LMM_{xy} et LMM_{uv}.

2.4.1 Génération de coupes de concavité

La première étape dans l'évaluation d'une coupe de concavité consiste à déterminer une partie du domaine où la valeur de la fonction convexe f ne dépasse pas γ. Pour ce faire, nous devons évaluer les γ-*extensions* dans les directions $\delta_i, i = 1, 2, \ldots, k$ des k sommets voisins de $\hat{x}$ dans X.

Définition 2.12 *(Horst et Tuy [108]) Pour le sommet $\hat{x}$ de X, la γ-extension dans la direction δ^i, $(i = 1, 2, \ldots, k)$ est le point $\hat{x} + \theta_i \delta^i \in \mathbb{R}^{n_x}$ où θ_i est la valeur optimale de*

$$\begin{aligned} \max_{\theta} \quad & \theta \\ s.c. \quad & f(\hat{x} + \theta\delta^i) \leq \gamma. \end{aligned}$$

■

La γ-extension (qui peut ne pas être bornée) est le point le plus éloigné de $\hat{x}$ dans la direction δ^i pour lequel la valeur de la fonction objectif ne dépasse pas γ. La convexité de la fonction f assure que tout point de l'enveloppe convexe de l'ensemble $\{\hat{x}, \hat{x} + \theta_1\delta^1, \hat{x} + \theta_2\delta^2, \ldots, \hat{x} + \theta_k\delta^k\}$ possède une valeur de la fonction f inférieure ou égale à γ.

Pour évaluer une γ-extension, on doit considérer la fonction

$$f_i(\theta) \equiv f(\hat{x} + \theta\delta^i).$$

Dans les deux sections suivantes, cette fonction d'une seule variable possède une structure particulière qui facilite l'évaluation des γ-extensions. Lorsque la fonction f_i est définie par l'intermédiaire un problème de minimisation, le lemme suivant réduit

l'évaluation de θ_i à la résolution d'un problème d'optimisation simplifié.

Lemme 2.13 *Soit* $f_i(\theta) = \min_{\omega \in \Omega(\theta)} g_i(\omega)$, *où* g_i *est une fonction de* $\mathbb{R}^n$ *dans* $\mathbb{R}$ *et* $\Omega(\theta) \subseteq \mathbb{R}^n$. *La* γ*-extension dans la direction* δ^i *est telle que* θ_i *est égal à la valeur optimale de*

$$\begin{array}{ll} \max\limits_{\theta,\omega} & \theta \\ s.c. & g_i(\omega) \leq \gamma, \\ & \omega \in \Omega(\theta). \end{array}$$

Démonstration Soit (θ^*, ω^*) une solution optimale de ce dernier problème, et θ_i un scalaire tel que $\hat{x} + \theta_i \delta^i$ est la γ-extension de $\hat{x}$ dans la direction δ^i. On peut alors écrire par la définition 2.12

$$\begin{aligned} \theta_i = \left(\begin{array}{ll} \max\limits_{\theta} & \theta \\ \text{s.c.} & f_i(\theta) \leq \gamma \end{array} \right) &= \left(\begin{array}{ll} \max\limits_{\theta} & \theta \\ \text{s.c.} & \left(\min\limits_{\omega \in \Omega(\theta)} g_i(\omega) \right) \leq \gamma \end{array} \right) \\ &= \left(\begin{array}{ll} \max\limits_{\theta,\delta} & \theta \\ \text{s.c.} & g_i(\omega) \leq \gamma, \\ & \omega \in \Omega(\theta). \end{array} \right) \end{aligned}$$

La dernière égalité provient du fait que

$$\left(\min_{\omega \in \Omega(\theta)} g_i(\omega) \right) \leq \gamma \iff \text{il existe } \omega \in \Omega(\theta) \text{ tel que } g_i(\omega) \leq \gamma.$$

Il en résulte que $\theta_i = \theta^*$. ∎

Horst et Tuy [108] montrent que $\pi^t(x - \hat{x}) \geq 1$ où

$$\begin{array}{lll} \pi \in \arg\min\limits_{\pi} & \sum\limits_{i=1}^{k} \theta_i \pi^t \delta^i & \\ \text{s.c.} & \pi^t \delta^j \geq \frac{1}{\theta_j} & j = 1, 2, \ldots, k \end{array}$$

est une coupe de concavité valide. Plus les valeurs θ_i sont élevées, plus la coupe de concavité est profonde.

2.4.2 Une approche directe

Considérons le problème LMM_{xy} présenté à la section 1.2.2

$$\max_{x \in X} D_y(x) = c^t x + \min_{y \in Y(x)} b^t y.$$

sous les deux hypothèses suivantes

i- X est un polytope.

ii- Il n'existe pas de $x \in X$ tel que $Y(x) = \emptyset$
(voir le problème PROHIBITION à la section 2.2.1).

Soient $\hat{x} \in ext(X)$, et $\gamma \in \mathbb{R}$ tels que

$$D_y(x) \leq \gamma \text{ pour tout } x \in B_\epsilon(\hat{x}) \cap X,$$

pour un $\epsilon > 0$. L'évaluation des γ-extensions découle du lemme 2.13, le corollaire est cité sans preuve.

Corollaire 2.14 *La γ-extension $\hat{x} + \theta_i \delta^i$ est telle que θ_i est la valeur optimale du programme linéaire*

$$\theta_i = \left(\begin{array}{ll} \max\limits_{\theta, y} & \theta \\ s.c. & c^t \delta_i \theta + b^t y \leq \gamma - c^t \hat{x} \\ & Q \delta_i \theta + B y \geq d - Q \hat{x} \\ & y \geq 0. \end{array} \right)$$

■

Pour chacune des directions voisines δ_i, on doit simplement résoudre un programme linéaire ayant $n_y + 1$ variables et $n_u + 1$ contraintes. Il est possible que la valeur $D_y(\hat{x} + \theta_i \delta^i)$ soit strictement inférieure à γ. Si tel est le cas, alors pour tout $\epsilon > 0$, l'ensemble $Y(\hat{x} + (\theta_i + \epsilon)\delta^i)$ est vide, et donc $D_y(\hat{x} + (\theta_i + \epsilon)\delta^i)$ est non borné. Il est possible de construire des coupes de concavité plus profondes en considérant une

nouvelle fonction convexe $\tilde{D}_y(\cdot)$, bornée sur tout l'espace et qui coïncide avec $D_y(\cdot)$ sur le domaine X. Il suffit de prendre

$$\tilde{D}_y(x) \equiv c^t x + \max_{u \in ext(U)} u^t(d - Qx).$$

Une telle approche n'est cependant pas directement praticable puisqu'elle requiert l'énumération des points extrêmes de U.

2.4.3 Une approche indirecte via la reformulation symétrique

Considérons la reformulation équivalente LMM_{uv} présenté à la section 1.2.2

$$\max_{u \in U} D_v(u) = u^t d + \min_{v \in V(u)} v^t a,$$

sous les hypothèses indirectes présentées à la section précédente.

Soit $\hat{u} \in ext(U)$, et $\gamma \in \mathbb{R}$ tels que

$$D_v(u) \leq \gamma \text{ pour tout } u \in B_\epsilon(\hat{u}) \cap U,$$

pour un $\epsilon > 0$. L'évaluation des γ-extensions découle encore une fois du lemme 2.13, le corollaire est cité sans preuve.

Corollaire 2.15 *La γ-extension $\hat{u} + \theta_i \delta^i$ est telle que θ_i est la valeur optimale du programme linéaire*

$$\theta_i = \left(\begin{array}{ll} \max\limits_{\theta, v} & \theta \\ s.c. & \theta \delta_i^t d + v^t a \leq \gamma - \hat{u}^t d \\ & \theta \delta_i^t Q + v^t A \geq c^t - \hat{u}^t Q \\ & v \geq 0, \end{array} \right)$$

où δ_i est la direction d'un sommet de U voisin de $\hat{u}$. ∎

Pour chacune des directions voisines δ_i, on doit simplement résoudre un programme linéaire ayant $n_v + 1$ variables et $n_x + 1$ contraintes. Cette fois-ci, lorsque θ_i est borné, les valeurs $D_v(\hat{u} + \theta_i \delta^i)$ sont nécessairement égales à γ puisque la fonction $D_v(\cdot)$ est bornée sur $\mathbb{R}^{n_u}$. En effet, pour tout $u \in \mathbb{R}^{n_u}$,

$$D_v(u) = LP_x(u) = u^t d + \max_{x \in X}(c^t - u^t Q)x$$

est borné car nous avons supposé que X soit un polytope, et donc il est borné.

Il en résulte que dans le cas où l'ensemble X est borné, les coupes de concavité associées à la formulation LMM_{uv} sont plus profondes que celles associées à LMM_{xy}.

2.5 Méthode de résolution

L'ensemble des résultats obtenu aux sections précédentes permet l'élaboration d'une méthode de résolution du problème $BILD$. La force de l'algorithme présenté ci-dessous réside dans l'utilisation répétée des conditions d'optimalité des formulations linéaires maxmin LMM_{xy} et LMM_{uv}.

Dans un algorithme d'énumération implicite de séparation et évaluations successives, il arrive fréquemment en pratique que le processus de branchement génère un sous-problème qui est beaucoup plus facile à résoudre que l'autre. Ceci entraîne un arbre d'exploration déséquilibré. Le processus de branchement de l'algorithme proposé engendre deux sous-problèmes dont l'information est comparable, et donc la probabilité de générer un arbre d'énumération équilibré est accrue. L'un des sous-problèmes possède une variable (ou une variable d'écart) supplémentaire du premier niveau fixée à zéro, tandis que l'autre en a une du second niveau fixée à zéro.

2.5.1 Processus de branchement

Nous proposons une méthode de résolution dont le branchement consiste à fixer à zéro une variable d'un des problèmes LMM_{xy} ou LMM_{uv}. La validité du processus

de branchement est assurée par les conditions d'écarts complémentaires.

Proposition 2.16 *Il existe une solution optimale de LMM_{xy} et une solution optimale de LMM_{uv} satisfaisant les conditions d'écarts complémentaires*

$$u^t(Qx + By - d) = 0, \quad (b^t - u^tB)y = 0,$$
$$(v^tA + u^tQ - c^t)x = 0, \quad v^t(a - Ax) = 0.$$

Démonstration Le résultat découle des conditions d'écarts complémentaires des paires de problèmes primaux-duaux $LP_u(x)$, $D_y(x)$ et $LP_x(u)$, $D_v(u)$. ∎

Nous introduisons les variables binaires $\alpha^t = [\alpha^u, \alpha^y, \alpha^x, \alpha^v]^t$, où $\alpha^y \in \{0,1\}^{n_y}$, $\alpha^u \in \{0,1\}^{n_u}$, $\alpha^v \in \{0,1\}^{n_v}$, $\alpha^x \in \{0,1\}^{n_x}$ afin d'indiquer si les contraintes des problèmes linéaires maxmin sont satisfaites avec égalité :

$$\alpha_i^u = \begin{cases} 1 & \text{si } B_{i\cdot}y = d_i - Q_{i\cdot}x, \\ 0 & \text{sinon,} \end{cases} \quad i \in N_u, \qquad \alpha_i^y = \begin{cases} 0 & \text{si } u^tB_{\cdot i} = b_i, \\ 1 & \text{sinon,} \end{cases} \quad i \in N_y,$$

$$\alpha_i^x = \begin{cases} 1 & \text{si } v^tA_{\cdot i} = c_i - u^tQ_{\cdot i}, \\ 0 & \text{sinon,} \end{cases} \quad i \in N_x, \qquad \alpha_i^v = \begin{cases} 0 & \text{si } A_{i\cdot}x = a_i, \\ 1 & \text{sinon,} \end{cases} \quad i \in N_v,$$

où les ensembles d'indices sont $N_x = \{1, 2, \ldots, n_x\}$, $N_u = \{1, 2, \ldots, n_u\}$, $N_y = \{1, 2, \ldots, n_y\}$ et $N_v = \{1, 2, \ldots, n_v\}$. Le branchement dichotomique ($\alpha_i = 1$ versus $\alpha_i = 0$) est motivé par la proposition 2.16. La complémentarité assure l'existence d'une solution optimale telle que si $\alpha_i^u = 0$ alors $u_i = 0$, si $\alpha_i^x = 0$ alors $x_i = 0$, si $\alpha_i^y = 1$ alors $y_i = 0$ et si $\alpha_i^v = 1$ alors $v_i = 0$.

Des informations supplémentaires peuvent être obtenues en considérant le signe des paramètres. La preuve de la proposition suivante est omise puisqu'elle est une

conséquence directe d'un résultat de [96].

Proposition 2.17 *Les relations de monotonie*

$$1 \leq r_j^y = \begin{cases} \alpha_j^y + \sum_{i \in N_u : B_{ij} > 0} \alpha_i^u & si\ b_j > 0 \\ \sum_{i \in N_u : B_{ij} < 0} \alpha_i^u & si\ b_j < 0 \\ 1 & si\ b_j = 0 \end{cases} \qquad j \in N_y,$$

$$1 \leq r_j^v = \begin{cases} \alpha_j^v + \sum_{i \in N_x : A_{ji} > 0} \alpha_i^x & si\ a_j > 0 \\ \sum_{i \in N_x : A_{ji} < 0} \alpha_i^x & si\ a_j < 0 \\ 1 & si\ a_j = 0 \end{cases} \qquad j \in N_v,$$

sont valides aux solutions rationnelles de LMM_{xy} et LMM_{uv}.

Ces relations logiques définissent des conditions nécessaires d'optimalité des problèmes de minimisation $D_y(x)$ et $D_v(u)$. Chaque relation indique que parmi un ensemble de contraintes au moins l'une d'entre elles doit être satisfaite à égalité.

Le processus de branchement engendre deux sous-problèmes. Dans l'un deux, une variable de premier niveau d'un des problèmes LMM_{xy} ou LMM_{uv} ($x \in X$ ou $u \in U$) est fixée à égalité, et dans l'autre, une contrainte ($y \in Y(x)$ ou $v \in V(u)$) est fixée à égalité.

2.5.2 Évaluation de bornes

À chaque noeud de l'arbre d'exploration engendré par l'algorithme, des bornes inférieures et supérieures doivent être évaluées. La borne supérieure sert à élaguer l'arbre (à éliminer le noeud courant). Soit $\overline{z}$ le minimum des valeurs optimales des deux relaxations de premier niveau (voir la section 1.1.8)

$$LR_{xy} \quad \begin{array}{ll} \max\limits_{x\in X,\ y} & c^t x + b^t y \\ \text{s.c.} & Qx + By \geq d, \\ & y \geq 0, \end{array} \qquad \text{et} \qquad LR_{uv} \quad \begin{array}{ll} \max\limits_{u\in U,\ v} & u^t d + v^t a \\ \text{s.c.} & u^t Q + v^t A \geq c^t, \\ & v \geq 0. \end{array}$$

des problèmes linéaires maxmin (rappelons que $\overline{z}$ est fixé à $-\infty$ dans le cas où l'un des deux problème est non réalisable). Il s'ensuit que la valeur $\overline{z}$ est une borne supérieure valide pour le noeud courant de l'arbre d'exploration.

À un noeud autre que la racine, ces relaxations contiennent implicitement certaines des variables α_i fixées à 0 ou 1, c'est-à-dire, certaines contraintes sont fixées à égalité ($A_{i\cdot}x = a_i, u^t B_{\cdot i} = b_i, Q_{i\cdot}x + B_{i\cdot}y = d_i, u^t Q_{\cdot i} + v^t A_{\cdot i} = c_i$) et certaines variables sont fixées à zéro ($x_i = 0, u_i = 0, v_i = 0, y_i = 0$).

Cette borne supérieure est souvent inutile ($\overline{z} = \infty$) aux noeuds de faible profondeur de l'arbre d'exploration puisque les ensembles $Y(x)$ et $V(u)$ sont habituellement non bornés. Comme l'exécution de l'algorithme descend en profondeur dans l'arbre d'exploration, plusieurs contraintes se fixent à égalité et donc, $\overline{z}$ devient rapidement borné. De plus, le sous-problème courant est éliminé dès qu'une des relaxations LR_{xy} ou LR_{uv} est non réalisable.

Cette borne peut être améliorée au prix d'un plus grand effort de calcul.

Proposition 2.18 *La valeur z' obtenue par la résolution de deux programmes linéaires disjoints*

$$z' \equiv \max_{x\in X} c^t x + \left(\begin{array}{ll} \min\limits_{y} & b^t y \\ s.c. & By \geq d - \Phi \\ & y \geq 0, \end{array} \right)$$

où $\Phi_i \equiv \min\limits_{x\in X} Q_{i\cdot}x$ pour $i = 1, 2, \ldots, n_u$, est une borne supérieure valide pour $BILD$ au moins aussi bonne que la valeur optimale de LR_{xy}.

Démonstration La valeur optimale z^* de $BILD$ satisfait nécessairement

$$z^* \leq \max_{\substack{x\in X \\ u\in U}} c^t x - u^t \Phi + u^t d = \max_{x\in X} c^t x + \max_{u\in U} u^t (d - \Phi) = z',$$

et donc z' est une borne valide.

Soit (x', y') une solution optimale de la relaxation présentée dans l'énoncé. Il s'ensuit que d'une part $x' \in X$ et $y \geq 0$, et d'autre part $By' \geq d - \Phi$ et $\Phi \leq Qx'$ assurent que $Qx' + By' \geq d$. Cette solution est donc réalisable pour la relaxation LR_{xy}, d'où le résultat. ∎

La borne $\overline{z}$ pourrait donc être améliorée en évaluant le minimum entre z' et la valeur optimale de la relaxation symétrique en les variables u et v.

La relaxation peut aussi être obtenue par une relaxation lagrangienne de $BILD$. En effet, en pénalisant la contrainte $u^t B \leq d^t$ dans l'objectif, la valeur optimale z^* de $BILD$ satisfait

$$\begin{aligned}
z^* &= \min_{y \geq 0} \max_{\substack{x \in X \\ u \in U}} (c^t x - u^t Q x + u^t d + (b^t - u^t B) y) \\
&\leq \min_{y \geq 0} \max_{\substack{x \in X \\ u \geq 0}} (c^t x - u^t Q x + u^t d + b^t y - u^t B y) \\
&= \min_{y \geq 0} \left(b^t y + \max_{\substack{x \in X \\ u \geq 0}} c^t x + u^t (d - Qx - By) \right) \\
&= \min_{y \geq 0} \left(b^t y + \begin{cases} \max\limits_{x \in X} c^t x & \text{si } \Phi \geq d - By \\ \infty & \text{sinon} \end{cases} \right) = z'.
\end{aligned}$$

Cette dernière borne n'est pas utilisée dans l'algorithme en raison de sa complexité d'évaluation.

La borne inférieure de $BILD$ est obtenue à partir d'une solution réalisable par une approche de type Gauss-Seidel.

Proposition 2.19 *La valeur optimale obtenue en résolvant alternativement les problèmes linéaires paramétrés $LP_u(x)$ et $LP_x(u)$ jusqu'à ce que deux itérations donnent la même valeur optimale, donne en temps fini une borne inférieure valide pour le problème initial $BILD$.*

Démonstration Une telle solution est nécessairement réalisable pour $BILD$. Le

nombre d'itérations requis par cette procédure ne peut excéder le nombre de sommets de X et U puisque la solution se déplace à chaque itération d'un sommet à l'autre et la suite de valeurs optimales ne décroît jamais. ∎

La solution produite par une telle procédure ne correspond pas à une borne inférieure valide pour le noeud courant puisqu'elle ne tient pas compte des contraintes de second niveau $y \in Y(x)$ ou $v \in V(u)$ fixées à égalité (seuls les problèmes primaux $LP_y(x)$ et $LP_v(u)$ sont considérés). Cette solution permet la mise à jour de la meilleure solution connue. Contrairement à la borne supérieure, la borne inférieure est souvent utile aux noeuds de faible profondeur puisqu'elle engendre rapidement une bonne solution réalisable.

2.5.3 Algorithme pour le problème $BILD$

L'algorithme de résolution du problème $BILD$ est divisé en deux phases. La première vérifie si la valeur optimale de $BILD$ est bornée ou non. Si elle l'est, la second phase trouve la meilleure solution (qui est un point extrême) du problème bilinéaire disjoint. L'algorithme se formule dans toute sa généralité comme suit : (lorsque nous indiquons que $BILD$ est borné ou non borné, il est entendu que cela signifie que la valeur optimale du problème $BILD$ est bornée ou non bornée).

ALGORITHME $Al(BILD)$.

PHASE I.
Si la valeur optimale de $BILD$ est non bornée, arrêter. Sinon, poursuivre à la **PHASE II** pour résoudre $BILD$.

PHASE II.
Déterminer le point extrême dont la valeur de la fonction objectif est la plus élevée du problème bilinéaire disjoint donné (soit $BILD$ ou un problème auxiliaire). ∎

Réécrivons de façon plus précise ces deux phases. La première utilise la seconde pour résoudre les problèmes bilinéaires disjoints auxiliaires (2.7), (2.8), (2.9) et (2.10).

PHASE I. Vérification de l'existence d'une solution optimale bornée

a. Test de réalisabilité direct.

Si X ou U est vide, arrêter : le problème est non réalisable.

b. Test d'optimalité direct, première partie.

Si X et U sont bornés, alors $BILD$ est borné.

Sinon si X est borné, appliquer **PHASE II** pour obtenir la valeur optimale de (2.7).

Si la valeur optimale est strictement positive, alors $BILD$ est non borné.

Autrement $BILD$ est borné.

Sinon si U est borné, appliquer la **PHASE II** pour obtenir la valeur optimale de (2.8).

Si la valeur optimale est strictement positive, alors $BILD$ est non borné.

Autrement $BILD$ est borné.

Sinon, appliquer la **PHASE II** pour obtenir les valeurs optimales de (2.9) et (2.10).

Si une des valeurs optimales est strictement positive, alors $BILD$ est non borné.

Autrement $BILD$ est borné.

c. Test d'optimalité direct, deuxième partie.

Si la valeur optimale de $BILD$ est non bornée, fin.

Sinon, aller à la **PHASE II** pour résoudre $BILD$. ∎

La seconde phase de l'algorithme résout un problème bilinéaire disjoint. Elle est utilisée soit pour résoudre l'un des problèmes auxiliaires (2.7), (2.8), (2.9) ou bien (2.10), ou pour résoudre $BILD$ lorsque la phase I conclut que la valeur optimale est bornée.

PHASE II. Détermination du meilleur point extrême d'un problème bilinéaire disjoint.

a. Initialisation.

Obtenir la solution réalisable $(\hat{x},\ \hat{u})$ de valeur $\hat{z}$ en résolvant alternativement $LP_u(x)$ et $LP_x(u)$ avec $x = 0$ comme point de départ.

b. Test de réalisabilité direct.

Si l'un où l'autre de LR_{xy} ou LR_{uv} est non réalisable, aller à l'étape **f**.

c. Premier test d'optimalité (borne supérieure).
Soit $(\overline{x},\ \overline{y})$ une solution optimale de LR_{xy}, et $\overline{z}_{xy}$ la valeur optimale correspondante.
Si $\overline{z}_{xy} < \hat{z}$, aller à l'étape **f**.
Soit $(\overline{u},\ \overline{v})$ une solution optimale de LR_{uv}, et $\overline{z}_{uv}$ la valeur optimale correspondante.
Si $\overline{z}_{uv} < \hat{z}$, aller à l'étape **f**.
Poser $\overline{z} = \min\{\overline{z}_{xy},\ \overline{z}_{uv}\}$.

d. Second test d'optimalité (borne inférieure).
Obtenir la solution réalisable $(\underline{x},\ \underline{u})$ de valeur $\underline{z}$ en résolvant alternativement $LP_u(x)$ et $LP_x(u)$ avec $x = \overline{x}$ comme point de départ. Si $\underline{z} > \hat{z}$, poser $\hat{z} = \underline{z}$, $\hat{x} = \underline{x}$, et $\hat{u} = \underline{u}$.
Si $\underline{z} \geq \overline{z}$, aller à l'étape **f**.
Obtenir la solution réalisable $(\underline{x},\ \underline{u})$ de valeur $\underline{z}$ en résolvant alternativement $LP_u(x)$ et $LP_x(u)$ avec $u = \overline{u}$ comme point de départ. Si $\underline{z} > \hat{z}$, poser $\hat{z} = \underline{z}$, $\hat{x} = \underline{x}$, et $\hat{u} = \underline{u}$.
Si $\underline{z} \geq \overline{z}$, aller à l'étape **f**.

e. Branchement.
Choisir une variable de branchement α_i où $i \in \{1, 2, \ldots, N_y + N_x + N_u + N_v\}$ selon une règle de branchement prédéterminée. Fixer $\alpha_i = 1$, ce qui crée un nouveau sous-problème et aller à l'étape **b** (ceci crée un sous-arbre enraciné à $\alpha_i = 1$).
Lorsque le sous-arbre enraciné à $\alpha_i = 1$ est complètement exploré, libérer toutes les variables fixées à l'intérieur de ce sous-arbre. Fixer $\alpha_i = 0$, ce qui crée un nouveau sous-problème et aller à l'étape **b** (ceci crée un sous-arbre enraciné à $\alpha_i = 0$).
Lorsque le sous-arbre enraciné à $\alpha_i = 1$ est complètement exploré, libérer toutes les variables fixées à l'intérieur de ce sous-arbre.

f. Retour arrière.
Il n'est plus nécessaire de développer davantage le noeud courant de l'arbre d'exploration. S'il s'agit de la racine, arrêter; $(\hat{x},\ \hat{u})$ est une solution optimale de valeur optimale $\hat{z}$. Sinon, compléter le branchement à l'étape **e** au noeud père de l'arbre d'exploration.

■

Théorème 2.20 *L'algorithme Al(BILD) résout le problème BILD en temps fini.*

Démonstration Il existe une solution optimale à un sommet des polyèdres définis par les domaines réalisables relaxés de LR_{xy} et LR_{uv}.

La proposition 2.17 assure que le branchement consistant à imposer que les contraintes soient serrées énumère implicitement toutes les bases possibles. Il s'ensuit qu'une solution optimale est éventuellement trouvée.

Chaque noeud de l'arbre d'exploration traite un nombre fini de programmes linéaires dont la taille est finie. Chaque noeud possède au moins une variable fixée à zéro de plus que son père. Le nombre de variables est fini, et donc l'algorithme se termine en temps fini. ∎

Des tests supplémentaires peuvent être effectués afin d'accroître la vitesse de convergence de l'algorithme en réduisant le nombre de noeuds de l'arbre d'exploration. Par exemple, lorsque toutes les variables α_i apparaissant dans une relation de monotonie r_j sont fixées à zéro sauf une, la proposition 2.17 assure que la contrainte associée peut être fixée à égalité.

Un autre test qui réduit considérablement le nombre de noeuds de l'arbre d'exploration consiste à vérifier périodiquement si la fixation d'une des contraintes de LR_{xy} ou de LR_{uv} à égalité rend le problème non réalisable ou si la borne supérieure descend sous la meilleure valeur connue. Dans ce cas, la proposition 2.16 assure que la contrainte complémentaire peut être fixée à égalité. Cette analyse a un effet rétroactif : le problème complémentaire possède un nouvelle contrainte fixée à égalité, et donc il est plus probable qu'il devienne non réalisable ou bien que sa borne supérieure glisse sous la meilleure valeur connue.

Nous illustrons à la section suivante, la performance de l'algorithme en l'appliquant à plusieurs problèmes. Certains sont tirés de la littérature, et d'autres sont générés aléatoirement.

2.5.4 Résultats numériques

L'algorithme est écrit en C et utilise les librairies CPLEX2.1 pour la résolution des programmes linéaires. Une station SPARC SS20/514MP sous Solaris 2.4-27 est utilisée pour les calculs numériques.

L'implantation de l'algorithme est faite de la façon suivante. On peut en théorie évaluer des relations de monotonie à chaque noeud de l'arbre d'énumération. Cependant, nous n'utilisons que les relations associées au problème original. Ce choix augmente légèrement le nombre de noeuds générés, mais réduit d'approximativement 30% le temps de calcul pour les problèmes à faible densité. Il semble que le temps utilisé pour évaluer de nouvelles relations de monotonie ne soit pas compensé par le temps sauvé.

Un élément clef de l'implantation réside dans le choix de la variable de branchement α_i. Celle dont le produit des variables complémentaires est le plus grand (proposition 2.17) est choisie. Tel que mentionné à la fin de la section précédente, l'effet de fixer une variable α_i à 0 ou à 1 est étudié régulièrement dans l'arbre d'exploration.

Vicente *et al.* [203] présentent une méthode en deux étapes qui génère aléatoirement des problèmes bilinéaires disjoints dont on peut fixer *a priori* certaines propriétés. En un premier temps, ils montrent comment combiner plusieurs petits problèmes simples (dont on connaît explicitement les propriétés) en un problème plus grand (tout en conservant ces propriétés). Ensuite, la structure de ce dernier problème est brouillée en appliquant des transformations linéaires aux variables x et u. Parmi les propriétés susmentionnées il y a le nombre d'optima locaux et globaux.

Le tableau 2.1 illustre la performance de l'algorithme $Al(BILD)$ pour des problèmes ayant un grand nombre de solutions optimales. Les colonnes *temps* et *noeuds* indiquent les temps en secondes et le nombre de noeuds requis par l'arbre d'exploration. Pour ces problèmes (de taille fixée), il semble que le temps requis par l'algorithme croît exponentiellement lorsque le nombre de solutions globalement optimales croît exponentiellement. Le temps semble insensible au nombre d'optima locaux.

Tableau 2.1 – *Problèmes dont le nombre d'optima est connu*

$n_x = 20, n_u = 18, n_v = 30, n_y = 26$			
Solutions optimales		temps	noeuds
globale	locale	(sec)	
2^1	2^9	10,7	97
2^2	2^{10}	7,5	85
2^3	2^{11}	69,3	639
2^4	2^{12}	360,4	3035
2^5	2^{13}	1918,4	18609
2^6	2^{14}	12538,5	130105

$n_x = 40, n_u = 35, n_v = 60, n_y = 50$			
Solutions optimales		temps	noeuds
globale	locale	(sec)	
1	2^{15}	7,1	29
1	2^{18}	9,7	49
1	2^{21}	33,4	143
1	2^{24}	16,9	63
1	2^{27}	16,3	63
1	2^{30}	16,0	63

Nous n'avons pas trouvé d'autre séries de problèmes bilinéaires disjoints dans la littérature. Cependant, comme mentionné à la section 1.2.2, le problème QP_+ où la matrice de la fonction objectif est semi-définie positive se formule comme en $BILD$. Nous avons appliqué l'algorithme $Al(BILD)$ aux problèmes quadratiques présentés par Floudas et Pardalos [77].

Sherali et Tuncbilek [186] proposent une technique de reformulation-convexification (notée ST ci-dessous) pour obtenir une solution $\epsilon-$optimale pour des problèmes quadratiques non convexes. Leur approche entrent dans le cadre général des linéarisations décrites au chapitre 5. Phillips et Rosen [164] développent une méthode déterministe d'énumération implicite jumelée avec des sous-estimations linéaires afin d'obtenir une solution $\epsilon-$optimale pour des problèmes concaves contraints linéairement. L'algorithme $Al(BILD)$ a confirmé que les solutions aux problèmes de [77] obtenues par ces approches sont bel et bien optimales, et ce sans aucune approximation.

Le tableau 2.2 présente les résultats (Phillips et Rosen [164] ne spécifient ni la valeur de ϵ ni les temps de calcul) aux problèmes de Floudas et Pardalos [77]. La colonne *noeud optimal* donne le numéro du noeud où la solution optimale est trouvée. Pour ces problèmes, ce nombre est en moyenne moindre qu'un dixième du nombre total de noeuds. La différence de temps entre les deux algorithmes semble être due à la tolérance de 5% permise par l'algorithme ST. Le temps de calcul requis augmenterait si ϵ était réduit près de zéro.

Tableau 2.2 – *Problèmes tests de Floudas et Pardalos*

$n_x = n_u = 20$ $n_y = n_v = 10$	$Al(BILD)$ $\epsilon = 0\%$ SPARC SS20/514MP			ST $\epsilon = 5\%$ IBM 3090
numéro du problème	temps (sec)	noeuds	noeud optimal	temps (sec)
2.7-1	238,2	2461	241	3,29
2.7-2	266,2	2967	43	2,61
2.7-3	270,8	2189	162	2,55
2.7-4	272,2	2755	179	2,61
2.7-5	181,5	1685	397	15,94

Tous les autres tableaux présentés dans cette section contiennent les moyennes (μ) et les écarts types (σ) de dix problèmes générés au hasard de densité paramétrée D. La méthode de génération est la suivante. Les composantes des vecteurs a, b, c et d sont choisies aléatoirement et uniformément dans l'intervalle $[-10, 10]$. Pour chaque élément des matrices A, B et Q, un nombre aléatoire entre 0 et 1 est généré. S'il est inférieur à D, alors l'élément est choisi entre -20 et 20, autrement il est fixé à zéro. Afin d'assurer qu'aucune ligne ou colonne des matrices A ou B ne soit vide, des coefficients non nuls sont ajoutés aux lignes ou colonnes nécessitantes. De plus, les contraintes supplémentaires $\mathbb{1}^t x \leq n_x$ et $u^t \mathbb{1} \leq n_u$ assurent que les ensembles X et U soient bornés (seulement pour les tableaux 2.3, 2.4 et une partie de 2.5). Il s'ensuit que la densité réelle est plus élevée que le paramètre D, spécialement pour les problèmes de petite taille et où la valeur de D est faible. Le nombre entre parenthèse suivant D est la densité réelle des matrices.

L'algorithme pour le problème BLP proposé par Hansen *et al.* [96] (noté par $Al(BLP)$) décrit à la section 3.2.1, peut être utilisé pour résoudre $BILD$ en l'appliquant à l'une des reformulations linéaires maxmin LMM_{xy} ou LMM_{uv}. Cependant, cet algorithme requiert que les solutions optimales y et u soient bornées. Des contraintes additionnelles bornant les variables y et v sont introduites afin de l'utiliser. Pour améliorer la performance de l'algorithme, l'implantation de $Al(BLP)$ n'inclut pas les pénalités et évalue les relations de monotonie seulement au problème initial. Le tableau 2.3 contient les résultats des deux algorithmes appliqués à divers problèmes générés aléatoirement, utilisant le même ordinateur, langage de programmation et

librairies informatiques.

Tableau 2.3 – *Comparaison des algorithmes $Al(BILD)$ et $Al(BLP)$*

$n_x = n_u =$ $n_v = n_y$		temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds
25	D	1% (9%)		5% (11%)		10% (14%)		20% (23%)	
	$Al(BILD)\mu$	0,61	9,4	3,86	64,6	72,5	562,9	1687	6061
	σ	0,19	8,1	2,16	32,8	109,4	795,6	1500	5180
	$Al(BLP)\mu$	5,06	385,6	26,1	1601,2	180,4	7922,4	3477	90927
	σ	6,57	460,3	24,1	1396,8	124,0	5153,8	3705	92034
30	D	1% (8%)		5% (10%)		10% (13%)		20% (23%)	
	$Al(BILD)\mu$	0,87	13,2	9,8	110,7	269,0	1236,4	18955	39687
	σ	0,18	5,5	5,1	51,8	193,8	765,0	9305	19621
	$Al(BLP)\mu$	7,4	488,8	180,6	8883,6	768,8	24119,0		
	σ	2,2	143,1	203,8	9858,9	679,3	20911,6		
35	D	1% (7%)		5% (9%)		10% (13%)			
	$Al(BILD)\mu$	1,29	20,2	37,4	290,8	3210,9	8911,0		
	σ	0,27	7,8	24,8	165,6	1926,0	5414,5		
	$Al(BLP)\mu$	15,1	845,6	723,9	26864,0	7448,3	171781,9		
	σ	5,4	274,3	1033,1	37519,6	4653,6	108251,9		
40	D	1% (6%)		5% (8%)		10% (12%)			
	$Al(BILD)\mu$	1,83	27,7	241,9	1228,8	12759,8	22439,6		
	σ	1,25	32,8	248,0	1329,9	13452,4	24609,0		
	$Al(BLP)\mu$	26,2	1267,8	2894,4	83239,9				
	σ	14,7	668,4	4263,7	116355,1				

L'avantage principal que possède l'algorithme $Al(BILD)$ versus $Al(BLP)$ est qu'il génère beaucoup moins de noeuds dans l'arbre d'exploration. Ceci est dû à l'exploitation des deux reformulations symétriques qui engendre un arbre d'énumération plus équilibré. Le nombre de noeuds croît beaucoup plus rapidement avec l'algorithme $Al(BLP)$.

L'algorithme $Al(BILD)$ résout des problèmes de plus grande taille. Le tableau 2.4 montre l'effet de l'augmentation du nombre de variables sur des problèmes de faible densité. Même si le temps de résolution et le nombre de noeuds croît exponentiellement avec la taille du problème, il semble que des problèmes de taille

modérément importante peuvent être résolus exactement.

Tableau 2.4 – *Problèmes de grande taille*

$n_v = n_y = 100$	$D = 1\%(2.5\%)$									
$n_x = n_u$	100		125		150		175		200	
	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds
$Al(BILD)\mu$	83,7	223,0	436,2	825,7	1283,5	1581.3	3001,2	2567,9	18041.4	11342
σ	46,7	119,9	469,4	904,0	1194,8	1395,2	2090.3	1614,7	15691.4	9924

Le tableau 2.5 illustre la performance de l'algorithme $Al(BILD)$ lorsqu'un seul ensemble est borné. Rappelons que lorsque la valeur optimale est non bornée, la phase I de l'algorithme s'arrête dès qu'une valeur réalisable strictement positive est obtenue, ce qui explique les temps de calcul rapides.

Tableau 2.5 – *L'ensemble X est borné mais U est non borné*

$Al(BILD)$	$n_x = n_u = n_v = n_y = 25$											
	Valeur optimale bornée						Valeur optimale non bornée					
D	1% (7%)		5% (9%)		10% (12%)		1% (7%)		5% (9%)		10% (12%)	
Phase	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds	temps (sec)	noeuds
I μ	165,8	2595,6	384,7	5614,0	385,5	3513,9	0,86	7,8	5,8	69,2	2,9	30,6
σ	112,2	1725,9	553,8	8086,2	482,6	4123,5	0,29	5,6	6,5	73,2	3,2	54,6
II μ	0,42	4,5	0,95	14,4	36,8	319,6						
σ	0,16	3,1	0,55	12,8	52,5	411,1						

Lorsque la valeur optimale est bornée, la majeure partie du temps de calcul est passée à la phase I. Ce n'est pas surprenant puisque la solution optimale du problème bilinéaire auxiliaire (2.7) est hautement dégénérée pour la solution optimale de la variable u. Le temps requis à la phase II est comparable à celui du tableau 2.3.

Le tableau 2.6 présente les résultats de l'algorithme $Al(BILD)$ lorsque les deux ensembles sont non bornés. La phase I comporte trois étapes identifiées (a), (b) et (c) correspondant à la résolution des problèmes apparaissant dans (2.9) et (2.10).

Tableau 2.6 – *Les ensembles X et U sont non bornés*

$Al(BILD)$	$n_x = n_u = n_v = n_y = 25$											
	Valeur optimale bornée						Valeur optimale non bornée					
D	1% (5%)		5% (7%)		10% (11%)		1% (5%)		5% (7%)		10% (11%)	
	temps	noeuds	temps	noeuds	temps	noeuds	temps	noeuds	temps	noeuds	temps	noeuds
Phase	(sec)		(sec)		(sec)		(sec)		(sec)		(sec)	
I(a) μ	3,54	71,2	79,0	1197,8	17,5	157,1	1,67	30,8	1,65	22,8	1,81	12,8
σ	2,45	69,7	196,1	3081,4	24,1	181,3	2,22	55,1	1,20	25,2	1,56	15,9
(b) μ	15,5	318,6	12,2	174,0	25,0	256,4	0,14	1,00	0,31	2,80	0,09	0,50
σ	30,6	608,0	15,1	187,8	21,3	214,7	0,30	2,11	0,43	4,78	0,28	1,58
(c) μ	3380	52238	15714	247800	10152	105504	0,0	0,0	0,09	0,50	0,0	0,0
σ	3196	47802	37095	594642	7350	77159	0,0	0,0	0,29	1,58	0,0	0,0
II μ	0,43	3,40	0,80	12,5	7,0	74,9						
σ	0,13	2,07	0,27	7,9	5,7	53,1						

Tout comme précédemment, lorsque la valeur optimale est bornée, la plus grande partie du temps de calcul est passée à la phase I. Le problème (2.10) résolu à la phase I(c) est dégénéré par rapport aux deux variables x et u. Les phases I(a) et I(b) le sont par rapport à seulement une de ces variables. Ceci explique pourquoi la phase I(c) requiert significativement plus de temps.

On pourrait être tenté de croire que la phase I(c) puisse être remplacée par la résolution de $BILD$ avec les contraintes supplémentaires $\mathbb{1}^t x \leq L$ et $u^t \mathbb{1} \leq L$ où L est un scalaire de grande taille, et ensuite vérifier si ces contraintes sont serrées à l'optimalité. Cependant, une telle approche n'est pas valide puisqu'il est possible que l'optimum global de ce problème restreint soit un optimum local de $BILD$ éliminé par ces contraintes additionnelles.

Le seul autre algorithme de la littérature qui ne requiert pas la compacité du domaine réalisable est celui de Thieu [190]. Dans l'article en question, il est appliqué à des problèmes dont la taille n'excède pas $n_x = n_u = 8$ et $n_v = n_y = 10$. Nous ne croyons pas qu'une comparaison plus détaillée soit constructive.

2.6 Discussion

Nous avons présenté un algorithme fini qui résout exactement le problème $BILD$ et ce sans aucune hypothèse concernant la compacité du domaine réalisable ou de la valeur optimale.

Dans le cas où le domaine réalisable est non borné, nous avons développé des conditions nécessaires et suffisantes afin de vérifier la finitude de la valeur optimale. Ces conditions utilisent des problèmes bilinéaires disjoints auxiliaires. La question de déterminer si la valeur optimale est bornée est montrée être équivalente au problème PROHIBITION (défini ci-dessus). Nous prouvons par une réduction du NOYAU que ce problème est fortement NP-complet.

L'approche de la résolution du problème $BILD$ par les reformulations équivalentes LMM_{xy} et LMM_{uv} est motivée par le fait qu'elles n'accroissent pas le nombre d'optima locaux. Leur nombre peut même décroître. Nous montrons aussi que sous certaines hypothèses concernant une des formulation, l'utilisation de coupe de concavité est plus efficace en considérant la reformulation symétrique.

L'algorithme repose essentiellement sur les deux reformulations symétriques linéaires maxmin. L'information inhérente à celles-ci, telles les conditions d'optimalité associées aux écarts complémentaires et aux relations de monotonie, la détermination de bornes valides, se transfère et s'ajoute à celle du problème bilinéaire disjoint.

L'algorithme est appliqué à des problèmes tirés de Floudas et Pardalos [77], et à des problèmes générés aléatoirement dont la taille va jusqu'à 200 variables pour x et pour u, et 100 contraintes d'inégalités pour X pour U. L'exécution de l'algorithme aux problèmes générés selon la méthode de Vicente *et al.* [203] indique que le temps de résolution croît directement avec le nombre d'optima globaux.

CHAPITRE 3

Algorithmes plongés

≪ À quoi te fait penser ce poisson?
– À d'autres poissons.
– À quoi te font penser ces autres poissons?
– À d'autres poissons. ≫

Joseph Heller

Il est bien connu que toute paire de classes de problèmes NP-complets P_A et P_B sont équivalentes en ce sens que tout problème A de la classe P_A se reformule en un problème $B(A)$ de P_B, avec une croissance polynomiale en taille. De plus, la résolution de P_A ou de P_B requiert un temps exponentiel en la taille du problème, sauf si $P = NP$.

Cependant, nous observons en pratique, que parmi la classe de problèmes NP-complets et NP-durs, certains d'entre eux sont beaucoup plus difficiles à résoudre que d'autres. Aussi, la reformulation d'un problème A en $B(A)$ peut entraîner un temps de résolution de l'algorithme $Al(P_B)$ pour P_B considérablement supérieur à celui de l'algorithme $Al(P_A)$ pour P_A. Ceci suggère que nous devrions nous concentrer davantage sur les difficultés pratiques de la résolution de problèmes NP-complets et NP-durs en examinant les reformulations ainsi que les liens entre les algorithmes qui les résolvent.

Une question fondamentale se pose : est-il possible qu'un algorithme $Al(P_B)$ contienne un algorithme $Al(P_A)$ en ce sens que, lorsqu'appliqué à $B(A)$, il exécute exactement les même étapes (ou des étapes équivalentes) que l'algorithme $Al(P_A)$

pour A? Cette question se précise avec le concept d'*algorithme plongé*, défini ci-dessous. Si $Al(P_A)$ est plongé dans $Al(P_B)$ à travers la transformation $B : P_A \rightarrow P_B$, alors P_B semble être au moins aussi difficile que P_A.

Les résultats principaux de ce chapitre sont la définition, l'illustration et la discussion de l'utilisation potentielle du concept d'algorithme plongé. En particulier, nous montrons que l'algorithme classique de Beale et Small [40] pour MIP_{0-1} est plongé dans celui de Hansen *et al.* [96] pour BLP à travers la reformulation $MIP_{0-1} \rightarrow BLP$ présentée à la section 1.2.2. L'étude des algorithmes plongés permet la généralisation ou la spécialisation de tests d'un algorithme à l'autre. Par exemple, nous étendons les pénalités de $Al(BLP)$ aux pénalités plus fortes de Tomlin [193] pour MIP_{0-1}. Le contenu de ce chapitre se retrouve aussi à la fois dans Audet *et al.* [18] et dans la version détaillée [15] du même article. Les mêmes auteurs présentent dans [19] une synthèse des trois premiers chapitres de cette thèse.

3.1 Définition du plongement

L'algorithme $Al(P_A)$ est une méthode qui, lorqu'appliquée à un problème A_0 de la classe P_A, génère et traite une suite de sous-problèmes $\{A_\ell\}_{\ell \geq 0}$ jusqu'à ce qu'une solution optimale de A_0 soit trouvée, où jusqu'à ce qu'il soit démontré que A_0 soit non réalisable (si $Al(P_A)$ est un algorithme de séparation et d'évaluations successives, les sous-problèmes A_ℓ correspondent habituellement à des noeuds de l'arbre d'exploration; ce concept s'applique aux algorithmes de plans de coupure, de génération de colonnes, etc). La nature exacte de l'algorithme est définie par la façon dont il génère et traite les sous-problèmes.

Afin de comparer des algorithmes définis pour des problèmes de classes distinctes mais reliés (par l'intermédiaire d'une transformation), nous introduisons le concept *d'algorithme plongé* qui compare les deux suites de sous-problèmes engendrées par ceux-ci.

Définition 3.1 *Soient P_A et P_B deux classes de problèmes d'optimisation. L'algorithme $Al(P_A)$ est* plongé *dans $Al(P_B)$ via la transformation $B : P_A \rightarrow P_B$, $A \mapsto B(A)$ si pour tout problème A_0 de P_A, la suite $\{A_\ell\}_{\ell \geq 0}$ engendrée par l'application de $Al(P_A)$ à A_0, et la suite $\{B_\ell\}_{\ell \geq 0}$ engendrée par l'application de $Al(P_B)$ à $B(A_0)$ sont telles que pour tout $\ell \geq 0$, $B_\ell = B(A_\ell)$.*

Cette définition se veut aussi générale que possible. La transformation B ne doit pas être nécessairement une reformulation, puisque le lien entre les deux algorithmes est caractérisé par la façon dont les suites de sous-problèmes sont engendrées, et non par la nature de la transformation. Cette flexibilité n'exclut pas la comparaison d'algorithmes traitant des problèmes qui ne sont pas liés par une réduction de Turing en temps polynomial. De plus, il est possible que différents types d'algorithmes soient comparés. Par exemple, certains algorithmes de plans de coupure peuvent être vus via une transformation au dual, comme des algorithmes de génération de colonnes (en particulier, la méthode de Benders et la décomposition de Dantzig-Wolfe sont souvent interprétées comme des méthodes duales l'une de l'autre).

3.2 Description de deux algorithmes

Nous illustrons le concept d'algorithme plongé aux algorithmes de Hansen *et al.* [96] pour BLP et de Beale et Small [40] pour MIP_{0-1} à travers la reformulation $MIP_{0-1} \rightarrow BLP$. Comme des rapprochements étroits devront être exhibés, il est nécessaire de les présenter en détail. Ce premier algorithme est noté $Al(BLP)$, et ce second $Al(MIP_{0-1})$.

3.2.1 Algorithme pour le problème BLP

En plus de la relaxation de premier niveau LR_{BLP} (décrite à la section 1.1.8), l'algorithme utilise la *relaxation du second niveau* $FR_{BLP}(\hat{x})$, qui est le problème de second niveau courant dans lequel x est fixé à $\hat{x}$. L'algorithme $Al(BLP)$ considère

aussi le *sous-problème du second niveau* $FS_{BLP}(\hat{x})$, qui est le problème de second niveau du problème initial dans lequel x est fixé à $\hat{x}$.

Tout comme pour l'algorithme $Al(BILD)$ présenté au chapitre précédent, nous associons à la i^e contrainte de second niveau (incluant celles de non négativité), la variable binaire α_i. Si la variable est à 1, la contrainte correspondante est fixée à égalité. Si elle est à 0, la contrainte duale associée est fixée à égalité. À nouveau on obtient des relations de monotonie, c'est-à-dire, des conditions nécessaires à l'obtention d'une solution rationnelle. Toute solution rationnelle de BLP est telle que les contraintes de second niveau satisfont

$$\sum_{i:B^2_{ij}>0} \alpha_i \geq 1 \qquad \text{si} \quad d^2_j > 0,$$

$$\text{et} \qquad \sum_{i:B^2_{ij}<0} \alpha_i + \alpha_{m_2+j} \geq 1 \qquad \text{si} \quad d^2_j < 0$$

pour $j = 1, 2, \ldots, n_y$. L'ensemble des relations de monotonie est noté R. Chaque relation de monotonie r_k s'écrit sous la forme $\sum_{i \in I_k} \alpha_i \geq 1$, où I_k est un ensemble d'indices.

L'effet de fixer une variable α_i à 1, c'est-à-dire, fixer une contrainte à égalité, peut être anticipé jusqu'à un certain point par l'évaluation d'une pénalité. Considérons les équations correspondant au tableau optimal de LR_{BLP} pour le sous-problème courant :

$$\begin{aligned} z &= z^*_L - \sum_{j \in N} c^*_j s_j \\ s_i &= b^*_i - \sum_{j \in N} A^*_{ij} s_j \qquad i \in B \end{aligned}$$

où s_i est soit une variable x_i, y_i soit une variable d'écart, et B est l'ensemble d'indices de variables en base et N l'ensemble d'indices de variables hors base. Les valeurs indicées d'un astérisque correspondent aux coefficients du tableau optimal. Si s_i est une variable d'écart de la i^e contrainte, la pénalité pour fixer s_i à 0 est :

$$p_i = b_i^* \min_{j \in N: A_{ij}^* > 0} \frac{c_j^*}{A_{ij}^*}.$$

Elle correspond à la variation de la valeur de la fonction objectif z_L lors de la première itération duale de la méthode du simplexe après l'ajout de la contrainte $s_i = 0$.

Décrivons maintenant en détail l'algorithme d'énumération implicite.

ALGORITHME $Al(BLP)$.

a. Initialisation.
Fixer z_{opt} à $-\infty$. Considérer toutes les variables α_i $(i = 1, 2, \ldots, m_2 + n_y)$ comme étant libres. Poser $R = \emptyset$.

b. Premier test direct d'optimalité.
Résoudre LR_{BLP}; soit (x_L^*, y_L^*) une solution optimale et z_L^* la valeur optimale.
Si $z_L^* \leq z_{opt}$, aller à l'étape **m** (retour arrière).

c. Premier test direct de réalisabilité.
Résoudre le dual de $FR_{BLP}(x_L^*)$. S'il est non réalisable, aller à l'étape **m**.

d. Test de résolution directe, première partie.
Considérer à nouveau la solution optimale (x_L^*, y_L^*) de LR_{BLP}.
Vérifier si (x_L^*, y_L^*) est rationnel pour le sous-problème courant : résoudre $FR_{BLP}(x_L^*)$, et poser y_F^* une solution optimale. Si $d^{2t} y_L^* = d^{2t} y_F^*$ alors (x_L^*, y_L^*) est rationnel; autrement aller à l'étape **f**.

e. Test de résolution directe, deuxième partie.
Considérer à nouveau la solution optimale (x_L^*, y_L^*) de LR_{BLP}.
Vérifier si (x_L^*, y_L^*) est rationnel pour le problème initial: résoudre $FS_{BLP}(x_L^*)$, et poser y_{FS}^* une solution optimale.
Si $d^{2t} y_L^* = d^{2t} y_{FS}^*$ alors (x_L^*, y_L^*) est rationnel: mettre à jour z_{opt} et (x_{opt}, y_{opt}) puis aller à l'étape **m**. Sinon aller à l'étape **f**.

f. Second test d'optimalité direct.
Évaluer toutes les pénalités p_i associées aux variables d'écarts strictement positives du tableau optimal de LR_{BLP}. Fixer les autres p_i à 0. Ensuite, pour chaque k satisfaisant $r_k \in R$, évaluer $\pi_k = \min_{i \in I_k} p_i$, et poser $\Pi = \max_k \pi_k$. Si $z_{opt} \geq z_L^* - \Pi$, aller à l'étape **m**.

h. Premier test d'optimalité conditionnel
Considérer à nouveau LR_{BLP} avec les pénalités p_i. Pour chaque i satisfaisant $z_{opt} \geq z_L^* - p_i$, fixer α_i à 0 et mettre à jour l'ensemble de relations de monotonies R.

i. Troisième test d'optimalité direct.
Si R contient une relation r_k telle que $\alpha_i = 0$ pour chaque $i \in I_k$, et donc la relation de monotonie r_k ne peut être satisfaite, poursuivre à l'étape **m**.

j. Test d'optimalité relationnel.
Pour les autres y_j apparaissant dans $d^{2t}y$, ajouter à R les relations logiques impliquant α_i, si elles ne sont pas redondantes. Éliminer de R les relations devenues redondantes.

k. Second test d'optimalité conditionnel.
Si R contient une relation r_k telle que $\alpha_j = 0$ pour chaque $j \in I_k$ sauf pour un seul indice i, fixer la variable α_i correspondante à 1. Éliminer du sous-problème courant une variable y_j apparaissant à la i^e contrainte et retourner à l'étape **b**.

l. Branchement.
Appliquer une règle de branchement prédéterminée (pour la comparaison avec l'algorithme $Al(MIP_{0-1})$, nous choisissons la variable binaire α_i associée à la plus grande pénalité p_i; des choix alternatifs sont proposés à la section 3.3.3) pour sélectionner ou bien une variable libre α_i ou une relation $r_k \in R$ pour laquelle toutes les variables α_i sont telles que les $i \in I_k$ sont libres. Dans ce premier cas, brancher en fixant α_i à 1. Dans ce dernier cas, brancher en fixant la première variable α_i de r_k à 1. Éliminer la variable y_j apparaissant à la i^e contrainte. Retourner à l'étape **b**.

m. Retour arrière.
Si le branchement correspondait à une variable, trouver le dernier α_i sur lequel un branchement de type $\alpha_i = 1$ eut lieu, fixer $\alpha_i = 0$ et libérer tous les α_j fixés à 0 après la fixation de α_i à 1. Sinon considérer la dernière relation logique r_k pour laquelle moins de $|I_k|$ branches furent explorées; considérer la branche suivante. S'il n'y a pas de telles variables ou de telles relations: fin, la solution optimale est (x_{opt}, y_{opt}) de valeur z_{opt}. Sinon mettre à jour le sous-problème courant et retourner à l'étape **b**. ∎

La version originale de $Al(BLP)$ contient le test **g. Second test de réalisabilité direct**, qui stipule que *si LR_{BLP} est non réalisable, aller à l'étape* **m**. Si on considère qu'une solution non réalisable a une valeur de $-\infty$, alors l'étape **b** couvre ce cas.

3.2.2 Algorithme pour le problème MIP_{0-1}

L'algorithme $Al(MIP_{0-1})$ utilise la *relaxation continue* $CR_{MIP_{0-1}}$ de MIP_{0-1}, où les contraintes d'intégralité $u \in \{0,1\}^{n_u}$ sont remplacées par $u \in [0,1]^{n_u}$.

L'effet de fixer une variable u_i à 0 ou à 1, peut être anticipé jusqu'à un certain point par l'évaluation de pénalités. Comme ci-dessus, posons p_i la pénalité associée à la fixation de s_i à 0, où s_i est soit une variable x_i, u_i ou une variable d'écart.

L'algorithme $Al(MIP_{0-1})$ considère les pénalités p_i^0 associée à la fixation de u_i à 0, et p_i^1 associée à la fixation de la variable d'écart correspondant à la contrainte $u_i \leq 1$ à 0 (c'est-à-dire, pour fixer u_i à 1).

L'algorithme d'énumération implicite de Beale et Small [40] avec pénalités pour le problème MIP_{0-1} s'écrit comme suit (les lettres associées aux tests ne sont pas consécutives afin de faciliter la comparaison avec l'algorithme $Al(BLP)$).

ALGORITHME $Al(MIP_{0-1})$.

a. Initialisation.
Fixer $z_{opt} = -\infty$.

b. Premier test direct d'optimalité.
Résoudre $CR_{MIP_{0-1}}$; soit (x_C^*, u_C^*) une solution optimale, et z_C^* la valeur optimale. Si $z_C^* \leq z_{opt}$, aller à l'étape **m** (retour arrière).

d. Test de résolution direct.
Considérer à nouveau la solution optimale (x_C^*, u_C^*) de $CR_{MIP_{0-1}}$.
Si la solution est réalisable (c'est-à-dire, si $u_C^* \in \{0,1\}^{n_u}$), alors mettre à jour z_{opt} et (x_{opt}, u_{opt}) et puis aller à l'étape **m**. Sinon aller à l'étape **f**.

f. Second test d'optimalité direct.
Évaluer les pénalités p_i^0 associées aux variables strictement positives u_i^* et p_i^1 associées aux variables d'écart strictement positives correspondant aux contraintes $u_i \leq 1$, du tableau optimal de $CR_{MIP_{0-1}}$. Fixer les autres p_i^h à 0, $h = 0, 1$. Ensuite, pour chaque i de $\{1, 2, \ldots, n_u\}$, évaluer $\pi_i = \min\{p_i^0, p_i^1\}$, et poser $\Pi = \max_i \pi_i$. Si $z_{opt} \geq z_C^* - \Pi$, aller à l'étape **m**.

k. Test d'optimalité conditionnel
S'il y a des indices i et h satisfaisant $z_C^* - p_i^h \leq z_{opt}$, alors fixer $u_i = 1 - h$ et aller à l'étape **b**.

l. Branchement.
Soit h et i les indices tels que p_i^h est maximal. Brancher en fixant $u_i = h$. Retourner à l'étape **b**.

m. Retour arrière.
Trouver la dernière variable u_i sur lequel un branchement de type $u_i = h$ eut lieu, fixer $u_i = 1 - h$, libérer les variables fixées depuis la fixation de u_i à h et retourner à l'étape **b**. S'il n'y en a plus : fin, la solution optimale est (x_{opt}, u_{opt}) de valeur z_{opt}. ∎

3.3 Plongement de l'algorithme $Al(MIP_{0-1})$ dans $Al(BLP)$

Nous présentons dans cette section le résultat principal de ce chapitre, c'est-à-dire, l'algorithme $Al(MIP_{0-1})$ est plongé dans $Al(BLP)$ via la reformulation $MIP_{0-1} \rightarrow BLP$. Tout au long de cette section, A se réfère à un problème de la classe MIP_{0-1} et $B(A)$ est la reformulation biniveau de A, définie par la reformulation $MIP_{0-1} \rightarrow BLP$.

3.3.1 Preuve du plongement

Quelques résultats préliminaires sont nécessaires à la preuve du théorème principal.

Lemme 3.2 *Les relaxations CR_A et $LR_{B(A)}$ sont identiques.*

Démonstration La relaxation $LR_{B(A)}$ possède la contrainte $v = 0$. Le résultat suit en éliminant la variable v. ∎

Nous présentons maintenant des propriétés concernant les algorithmes eux-mêmes. La nature de la reformulation $B(A)$ entraîne des simplifications importantes aux relations de monotonie utilisées par $Al(BLP)$. Pour $k \in \{1, 2, \ldots, n_u\}$, la relation de monotonie r_k s'écrit $\alpha_i^0 + \alpha_i^1 \geq 1$ où $i = k$ et les variables binaires sont liées aux contraintes de second niveau de la façon suivante :

$$(\alpha_i^0) \quad v_i \leq u_i, \qquad\qquad (\alpha_i^1) \quad v_i \leq 1 - u_i.$$

Ces relations se réécrivent plus simplement sans avoir recours à l'ensemble I_k. La comparaison avec $Al(MIP_{0-1})$ est facilitée en utilisant l'indice i au lieu de k.

Proposition 3.3 *Soit le problème A' obtenu à partir de A en fixant u_i à $1-h$ (pour $h \in \{0,1\}$ et $i \in \{1, 2, \ldots, n_u\}$), et B' obtenu à partir de $B(A)$ en fixant α_i^h à 0 et*

α_i^{1-h} *à* 1, *alors* $B' = B(A')$.

Démonstration Si $h = 0$, alors B' sera tel que $\alpha_i^0 = 0$ et $\alpha_i^1 = 1$, et donc $v_i = 1 - u_i$. De plus la contrainte de premier niveau $v_i = 0$ assure que $u_i = 1$. Autrement, si $h = 1$, alors B' sera tel que $\alpha_i^1 = 0$ et $\alpha_i^0 = 1$, et donc $v_i = u_i$. De plus la contrainte de premier niveau $v_i = 0$ assure que $u_i = 0$.

Dans les deux cas, B' définit précisément le même problème de la classe BLP que $B(A')$. ∎

Lemme 3.4 *Pour tout* $\hat{u} \in [0,1]^{n_u}$, *le problème dual de* $FS_{B(A)}(\cdot, \hat{u})$ *est toujours réalisable et de valeur optimale bornée.*

Démonstration La solution optimale de $FS_{B(A)}(\cdot, \hat{u})$ est $\hat{v} = \min\{\hat{u}, 1 - \hat{u}\}$ qui est nécessairement réalisable et de norme bornée. Le résultat suit de la théorie de la dualité. ∎

Les algorithmes $Al(MIP_{0-1})$ et $Al(BLP)$ évaluent tous deux des pénalités. Le lemme suivant montre les équivalences entre elles. Lorsque nous nous référons aux variables d'écart de second niveau, nous nous référons en réalité à la variable u ou bien à la variable d'écart associée à la contrainte $u \leq 1$.

Proposition 3.5 *Les pénalités utilisées par* $Al(MIP_{0-1})$ *appliqué à* A *sont les mêmes que celles utilisées par* $Al(BLP)$ *appliqué à* $B(A)$.

Démonstration Les pénalités p_i^0 et p_i^1 utilisé par $Al(MIP_{0-1})$ mesurent respectivement la variation dans la valeur de la fonction objectif lors de la première itération duale de la méthode du simplexe de CR_A lorsque u_i est fixé à 0 et à 1, pour $i \in \{1, 2, \ldots, n_u\}$.

Les pénalités utilisé par $Al(BLP)$ sont évaluées pour les variables de second niveau strictement positives (incluant les variables d'écarts). Rappelons que la solution optimale de $LR_{B(A)}$ satisfait $v_L^* = 0$, et donc les pénalités sont évaluées pour les

variable d'écarts des contraintes

$$\begin{array}{lll} (p_i^0) & v_i \leq u_i & \text{si } u_i > 0 \\ (p_i^1) & v_i \leq 1 - u_i & \text{si } u_i < 1. \end{array}$$

La contrainte de premier niveau $v = 0$ assure que les deux pénalités sont respectivement

- p_i^0, la variation dans la valeur de la fonction objectif lors de la première itération duale de la méthode du simplexe lorsque la contrainte $u_i = 0$ est ajoutée et
- p_i^1, la variation dans la valeur de la fonction objectif lors de la première itération duale de la méthode du simplexe lorsque la contrainte $1 - u_i = 0$ est ajoutée.

Le lemme 3.2 implique que ces pénalités sont exactement les mêmes que celles générées par $Al(MIP_{0-1})$. ∎

Nous sommes maintenant en mesure d'énoncer et de prouver le résultat principal de ce chapitre.

Théorème 3.6 *L'algorithme $Al(MIP_{0-1})$ est plongé dans $Al(BLP)$ via la reformulation $MIP_{0-1} \rightarrow BLP$.*

Démonstration Soient $A_0 = MIP_{0-1}$ et $B_0 = B(A_0)$ (qui est en fait la reformulation $MIP_{0-1} \rightarrow BLP$). Soient $\{A_\ell\}_{\ell \geq 0}$ et $\{B_\ell\}_{\ell \geq 0}$ les suites de sous-problèmes générées par les algorithmes appliqués à A_0 et B_0.

La preuve est faite par induction sur l'indice ℓ. Le résultat pour $\ell = 0$, c'est-à-dire que $B_\ell = B(A_\ell)$, suit de la définition.

Supposons que pour un $\ell \geq 0$ donné, $B_\ell = B(A_\ell)$ et que les meilleures valeurs connues des deux algorithmes z_{opt} coïncident. Nous montrons que $B_{\ell+1} = B(A_{\ell+1})$ et que les deux meilleures valeurs connues résultantes coïncident en comparant les algorithmes étape par étape.

L'étape **a** (celle où $\ell = 0$) des deux algorithmes est identique, la meilleure valeur connue est fixée à $-\infty$.

Le lemme 3.2 assure que l'étape **b** est la même dans les deux algorithmes. Le lemme 3.4 montre que l'étape **c** de $Al(BLP)$ est inutile. De plus, les étapes **d** et **e** de $Al(BLP)$ peuvent être combinées en la vérification que $0 = \min\{u_L^*, 1 - u_L^*\}$. Il s'ensuit que ces deux étapes, sont équivalentes à l'étape **d** de $Al(MIP_{0-1})$ (d'où la meilleure valeur connue z_{opt} est mise à jour de la même façon).

La relation de monotonie de la reformulation biniveau r_i, pour $i \in \{1, 2, \ldots, n_u\}$ est $\alpha_i^0 + \alpha_i^1 \geq 1$. Il s'ensuit que pour $Al(BLP)$, $\pi_i = \min\{p_i^0, p_i^1\}$, tout comme pour $Al(MIP_{0-1})$. La proposition 3.5 implique que l'étape **f** est la même dans les deux algorithmes. L'étape **i** de $Al(BLP)$ est redondante : si l'algorithme s'y rend, alors $z_{opt} < z_L^* - \pi_i$ pour chaque indice i, et donc l'étape **h** ne fixera jamais les deux variables α_i^0 et α_i^1 à 0.

Les étapes **h**, **j** et **k** de algorithme $Al(BLP)$ sont équivalentes à l'étape **k** de $Al(MIP_{0-1})$. Après avoir complété les l'étapes **h**, **j** et **k**, l'algorithme $Al(BLP)$ se rend à l'étape **b** si et seulement si il y a un indice i tel que $\alpha_i^h = 0$. Considérons le cas où il y a un tel indice i. La variable correspondante α_i^{1-h} est fixée à 1. La proposition 3.3 implique que ces trois étapes coïncident avec l'étape **k** de $Al(MIP_{0-1})$ qui fixe u_i à $1 - h$. D'où $B_{\ell+1} = B(A_{\ell+1})$. Dans le cas où il n'y a pas de tels indices, les deux algorithmes poursuivent à l'étape **l**.

L'étape **l** de $Al(BLP)$ crée deux sous-problèmes, l'un où $\alpha_i^h = 0$ (B') et l'autre où $\alpha_i^h = 1$ (B''). L'étape **l** de $Al(MIP_{0-1})$ crée aussi deux sous-problèmes, l'un où $u_j = 0$ (A') et l'autre où $u_j = 1$ (A''). La proposition 3.5 assure que les règles de branchements sont identiques dans les deux algorithmes, c'est-à-dire, $i = j$. La proposition 3.3 garantit que $B' = B(A')$ et $B'' = B(A'')$.

À l'étape **m** des deux algorithmes, le noeud courant de l'arbre d'exploration est éliminé récursivement jusqu'à ce qu'une branche inexplorée soit atteinte (s'il en reste), et l'hypothèse d'induction assure que les deux algorithmes retournent à l'étape **b** avec $B_{\ell+1} = B(A_{\ell+1})$. ∎

L'algorithme $Al(BLP)$ appliqué à la reformulation $MIP_{0-1} \Rightarrow BLP$ (voir la proposition 1.4) ne retourne pas exactement les mêmes étapes. Il est possible que des

branches où une solution rationnelle qui ne satisfait pas $\hat{u} \in \{0,1\}^{n_u}$ soient générées (puisqu'il est possible qu'il existe une solution rationnelle pour laquelle $\hat{v} \neq 0$). Au lieu d'élaguer rapidement une branche en utilisant la règle $u_i = 0$ versus $u_i = 1$ comme avec la reformulation $MIP_{0-1} \rightarrow BLP$, l'algorithme développera davantage la branche en séparant selon le critère $v_i = u_i$ versus $v_i = 1 - u_i$.

3.3.2 Exemple du plongement

Considérons le problème particulier de la classe MIP_{0-1} et sa reformulation BLP:

$$\begin{aligned} \max_{x,u} \quad & 12x + 8u_1 + 3u_2 \\ \text{s.c.} \quad & 4x - u_1 - 4u_2 \leq 1, \\ & 5x + 4u_1 + 2u_2 \leq 3, \\ & x \geq 0, \\ & u \in \{0,1\}^2, \end{aligned}$$

$$\begin{aligned} \max_{x,u,v} \quad & 12x + 8u_1 + 3u_2 \\ \text{s.c.} \quad & 4x - u_1 - 4u_2 \leq 1, \\ & 5x + 4u_1 + 2u_2 \leq 3, \\ & x \geq 0, \qquad 0 \leq u \leq \mathbb{1}, \\ & v = 0, \\ & v \in \arg\max_{v} \quad v_1 + v_2, \\ & \quad (\alpha_1^0) \ \text{s.c.} \quad v_1 \leq u_1, \\ & \quad (\alpha_1^1) \qquad v_1 \leq 1 - u_1, \\ & \quad (\alpha_2^0) \qquad v_2 \leq u_2, \\ & \quad (\alpha_2^1) \qquad v_2 \leq 1 - u_2. \end{aligned}$$

Chacune des colonnes montre respectivement étape par étape le déroulement des algorithmes $Al(MIP_{0-1})$ et $Al(BLP)$.

La résolution de $CR_{MIP_{0-1}}$ donne $(x_C^*, u_C^*) = (1/2, 0, 1/4)$ et $z_C^* = 27/4 > z_{opt} = -\infty$ (étape **b**). Cette solution n'est pas réalisable puisque $u_{C2}^* \notin \{0,1\}$ (étape **d**).

Les relations de monotonie sont $\alpha_1^0 + \alpha_1^1 \geq 1$ et $\alpha_2^0 + \alpha_2^1 \geq 1$.
La résolution de LR_{BLP} donne $(x_L^*, u_L^*, v_L^*) = (1/2, 0, 1/4, 0, 0)$ et $z_L^* = 27/4 > z_{opt} = -\infty$ (étape **b**). Cette solution n'est pas rationnelle puisque la valeur optimale de $FR_{BLP}(x_L^*, u_L^*)$ est $1/4 \neq 0$ (étapes **d** et **e**).

Les deux algorithmes génèrent les pénalités (étape **f**) :

$$p_1^0 = 0, \quad p_1^1 = 1\frac{1/4}{1} = \frac{1}{4}, \quad \pi_1 = 0,$$
$$p_2^0 = \frac{1}{4}\min\{\frac{1/4}{3/4}, \frac{15/7}{1/7}\} = \frac{1}{12}, \quad p_2^1 = \frac{3}{4}\frac{9/28}{5/28} = \frac{27}{20}, \quad \pi_2 = \frac{1}{12}, \quad \Pi = \frac{1}{12}.$$

(Les pénalités améliorées développées par Tomlin [193] ainsi que leur contrepartie biniveau, présentées à la section 3.4.1 ci-dessous, appliquées à p_2^0 donnent toutes deux $p_2^{*0} = \min\left\{\frac{1}{4}\max\{1, \frac{1/4}{3/4}\}, \frac{1}{4}\frac{15/7}{1/7}\right\} = \frac{1}{4}$.)

Cette branche n'est pas élaguée puisque $27/4 - \Pi > z_{opt} = -\infty$.

La pénalité maximale est p_2^1, et donc la branche $u_2 = 1$ est explorée (étape **l**). La résolution de $CR_{MIP_{0-1}}$ donne $(x_C^*, u_C^*) = (1/5, 0, 1)$ et $z_C^* = 27/5 > z_{opt} = -\infty$ (étape **b**). Cette solution est réalisable et donc, la meilleur solution connue est fixée à $(x_{opt}, u_{opt}) = (1/5, 0, 1)$ et $z_{opt} = 27/5$ (étape **d**).

Le retour arrière (étape **m**) nous amène à explorer la branche dans laquelle $u_2 = 0$. La résolution de $CR_{MIP_{0-1}}$ donne $(x_C^*, u_C^*) = (1/3, 1/3, 0)$ et $z_C^* = 20/3 > z_{opt} = 27/5$ (étape **b**). Cette solution n'est pas réalisable $u_{C1}^* \notin \{0, 1\}$ (étape **d**).

La pénalité maximale est p_2^1, et donc la branche $\alpha_2^1 = 1$, $u_2 = 1 - v_2$, est explorée (étape **l**). La résolution de LR_{BLP} donne $(x_L^*, u_L^*, v_L^*) = (1/5, 0, 1, 0, 0)$ et $z_L^* = 27/5 > z_{opt} = -\infty$ (étape **b**). Cette solution est rationnelle et donc, la meilleur solution connue est fixée à $(x_{opt}, u_{opt}, v_{opt}) = (1/5, 0, 1, 0, 0)$ et $z_{opt} = 27/5$ (étape **d**).

Le retour arrière (étape **m**) nous amène à explorer la branche dans laquelle $\alpha_2^1 = 0$. Les relations de monotonie impliquent $\alpha_2^0 = 1$, d'où $u_2 = v_2$. La résolution de LR_{BLP} donne $(x_L^*, u_L^*, v_L^*) = (1/3, 1/3, 0, 0, 0)$ et $z_L^* = 20/3 > z_{opt} = 27/5$ (étape **b**). Cette solution n'est pas rationnelle puisque la valeur optimale de $FR_{BLP}(x_L^*, u_L^*)$ est $1/3 \neq 0$ (étapes **d** et **e**).

Les deux algorithmes génèrent les pénalités (étape **f**) :

$$p_1^0 = \frac{1}{3}\frac{44/21}{4/21} = \frac{11}{3}, \quad p_1^1 = \frac{2}{3}\frac{8/21}{5/21} = \frac{16}{15}, \quad \pi_1 = \frac{16}{15}, \quad \Pi = \frac{16}{15}.$$

Cette branche n'est pas élaguée puisque $20/3 - \Pi = 28/5 > z_{opt} = 27/5$.

La variable u_1 est fixée à 1 puisque $z_C^* - p_1^0 = 10/3 \leq z_{opt} = 27/5$ (étape **k**). La relaxation $CR_{MIP_{0-1}}$ est non réalisable (étape **b**).

La variable α_1^0 est fixée à 0 puisque $z_L^* - p_1^0 = 10/3 \leq z_{opt} = 27/5$ (étape **h**). Les relations de monotonie impliquent que $\alpha_1^1 = 1$, et donc $u_1 = 1 - v_1$ (étape **k**). La relaxation LR_{BLP} est non réalisable (étape **b**).

Le retour arrière amène les deux algorithmes à leur fin avec la solution optimale $(x_{opt}, u_{opt}) = (1/5, 0, 1)$. La figure 3.1 illustre chacun des arbres d'exploration engendrés par les algorithmes, et représente à l'aide de pointillés les liens entre les sous-problèmes.

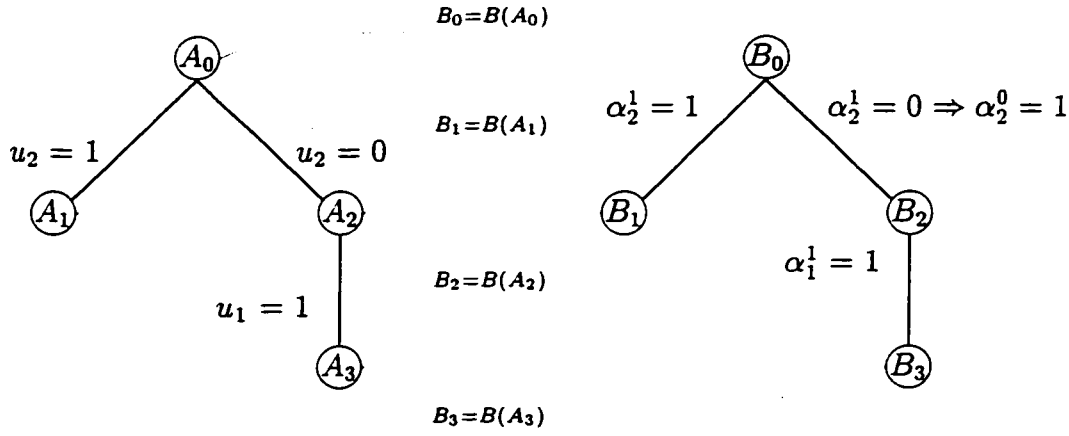


Figure 3.1 – *Arbre d'exploration*

3.3.3 Règles de branchement

Les règles de branchements suivantes sont proposées dans [96] pour $Al(BLP)$. Représentons la variable d'écart associée à la i^e contrainte de second niveau par s_i et la variable duale correspondante par λ_i pour $i = 1, 2, \ldots, m_2 + n_y$.

BR1 : *Branchement multiple.* (i) Choisir la relation logique $r_k \in R$ de cardinalité minimale. (ii) Briser les ex æquo en choisissant la relation ayant le plus grand

nombre de variables d'écart s_i en base. (iii) Ordonner les variables α_i de r_k selon l'ordre décroissant des valeurs des pénalités p_i.

BR2 : Identique à BR1, sauf pour (iii). Ordonner les variables α_i par ordre décroissant des produits $s_i\lambda_i$.

BR3 : *Branchement binaire.* Choisir la variable binaire α_i associée à la plus grande pénalité p_i.

BR4 : Choisir la variable α_i associée au plus grand produit $s_i\lambda_i$ (cette règle est identique à celle de l'algorithme de Bard et Moore [36]).

BR5 : *Branchement multiple ou binaire.* (i) Choisir la relation $r_k \in R$ contenant deux variables α_i et où les deux variables d'écarts correspondantes s_i sont en base, tel que $\sum_{i \in I_k} \lambda_i s_i$ est maximal. (ii) S'il n'y a pas de telles relations, utiliser BR4.

BR6 : Identique à BR5 sauf qu'à l'étape (ii), on branche sur la variable α_i ayant la plus grande pénalité p_i parmi celles où $s_i\lambda_i > 0$.

BR7 : Identique à BR5 sauf qu'à l'étape (ii), on branche sur la variable dont le produit $s_i\lambda_i > 0$ est minimal.

Ces diverses règles de branchement ont leur contreparties pour $Al(MIP_{0-1})$.

BR1, BR2 : Ces règles ne sont pas utiles puisque dans la reformulation $MIP_{0-1} \rightarrow BLP$, toutes les relations logiques ont une cardinalité égale à deux. Ces relations logiques sont trop simples pour supporter des règles de branchement.

BR3 : Cette règle est utilisée par le théorème 3.6.

BR4 : La proposition suivante montre que cette règle de branchement correspond à une règle bien connue en programmation $0-1$.

Proposition 3.7 *La règle de branchement BR4 équivaut à choisir la variable* u_i, *pour* $i \in \{1, 2, \ldots, n_u\}$, *dont la valeur est la plus près de* $1/2$.

Démonstration Soient s^0, s^1 les variables d'écarts associées aux contraintes $v \leq u$, $v \leq 1-u$. Le problème dual s'écrit en introduisant les variables duales λ^0 et λ^1

$$\begin{aligned} \min_{\lambda^0,\lambda^1} \quad & u^t\lambda^0 + (1-u)^t\lambda^1 \\ \text{s.c.} \quad & \lambda^0 + \lambda^1 \geq 1, \\ & \lambda^0 \geq 0,\ \lambda^1 \geq 0. \end{aligned}$$

À l'optimalité, elles prennent les valeurs

$$\lambda_i^0 = \begin{cases} 1 & \text{si } u_i \leq 1/2 \\ 0 & \text{si } u_i > 1/2 \end{cases}, \quad \lambda_i^1 = \begin{cases} 0 & \text{si } u_i \leq 1/2 \\ 1 & \text{si } u_i > 1/2 \end{cases},$$

et donc, la règle de branchement BR4 choisit la variable associée au plus grand produit (rappelons qu'une solution optimale de LR_{BLP} satisfait $v = 0$)

$$s_i^0\lambda_i^0 = \begin{cases} u_i & \text{si } u_i \leq 1/2 \\ 0 & \text{si } u_i > 1/2 \end{cases}, \quad s_i^1\lambda_i^1 = \begin{cases} 0 & \text{si } u_i \leq 1/2 \\ 1-u_i & \text{si } u_i > 1/2 \end{cases}.$$

Pour un i donné,

$$\max\{s_i^0\lambda_i^0,\ s_i^1\lambda_i^1\} = \begin{cases} u_i & \text{si } u_i \leq 1/2 \\ 1-u_i & \text{si } u_i > 1/2 \end{cases},$$

d'où le résultat. ∎

BR5, BR6, BR7 : La première étape de ces règles est satisfaite s'il existe au moins une variable u_i non entière. La somme en question est $s_i^0\lambda_i^0 + s_i^1\lambda_i^1$ qui est en fait égale à $\max\{s_i^0\lambda_i^0,\ s_i^1\lambda_i^1\}$, et donc ces règles sont toutes identiques à BR4.

3.4 Extensions des résultats

Dans cette section, nous appliquons les notions obtenues précédemment à trois questions distinctes. En un premier temps nous généralisons les pénalités développées par Tomlin [193] pour le problème MIP_{0-1} au problème BLP. Ensuite nous comparons l'algorithme de Wen et Yang [212] pour le problème $MIBLP_{0-1}$ avec l'algorithme $Al(BLP)$ appliqué à la reformulation $MIBLP_{0-1} \rightarrow BLP$. Puis nous montrons que la réciproque du théorème 3.6 n'est pas vraie lorsqu'on utilise la reformulation inverse présentée à la section 1.2.2.

3.4.1 Pénalités améliorées

Tomlin [193] propose une façon de déterminer des pénalités plus fortes pour un problème en nombres entier. L'idée consiste à considérer la variable qui entre en base (à la première itération duale de la méthode du simplexe après avoir fixé une variable en base à une valeur entière). Si la variable qui entre en base est contrainte à être entière, alors sa valeur doit croître au moins jusqu'à 1. La pénalité résultante est donc le maximum entre la pénalité classique et le coût réduit. Pour le problème MIP_{0-1}, cette pénalité est le coût réduit puisque cette variable ne peut excéder 1.

Pour le problème BLP, la pénalité classique lorsque la i^e variable quitte la base est $p_i = b_i^* \min_{j \in N: A_{ij}^* > 0} \{c_j^*/A_{ij}^*\}$ (où N est l'ensemble des indices des variables hors base, et les valeurs dotées d'un astérisque correspondent au coefficients du tableau optimal). Nous montrons comment améliorer les pénalités à $\overline{p}_i = \min_{j \in N: A_{ij}^* > 0} p_{ij}$, où p_{ij} est la pénalité améliorée lorsque la j^e variable s_j entre en base. Soit $j \in N$ un indice tel que $A_{ij}^* > 0$. Dans le cas où s_j ne correspond pas à une variable d'écart de second niveau, alors la pénalité p_{ij} est simplement $b_i^* c_j^*/A_{ij}^*$. Autrement, la proposition suivante montre qu'elle peut être améliorée.

Proposition 3.8 *Pour le problème BLP, les pénalités améliorées*

$$\overline{p}_i = \min_{j \in N: A_{ij}^* > 0} p_{ij},$$

où

$$p_{ij} = \begin{cases} c_j^* \max\left\{ \frac{b_i^*}{A_{ij}^*}, \max_{r_k \in R} \left\{ \min_{h \in I_k \setminus \{j\}: A_{hj}^* > 0} \frac{b_h^*}{A_{hj}^*} \right\} \right\} & \text{si } s_j \text{ est une variable d'écart de second niveau,} \\ \frac{b_i^* c_j^*}{A_{ij}^*} & \text{sinon,} \end{cases}$$

sont valides.

Démonstration Comme mentionné précédemment, nous n'avons qu'à considérer le cas où s_j correspond à une variable d'écart de second niveau. À cette nouvelle

solution, $\alpha_j = 0$. Rappelons que R est l'ensemble de toutes les relations de monotonie, c'est-à-dire, la relation $r_k \in R$ s'écrit $\sum_{h \in I_k} \alpha_h \geq 1$.

Pour chaque relation r_k, la variable qui entre en base, s_j, doit augmenter suffisamment pour qu'une contrainte de second niveau d'indice $h \in I_k \setminus \{j\}$ soit satisfaite à égalité (c'est-à-dire, pour que la relation de monotonie soit à nouveau satisfaite). Pour un indice $h \in I_k \setminus \{j\}$ donné tel que $A^*_{hj} > 0$, la contrainte correspondante sera serrée (c'est-à-dire, $\alpha_h = 1$, la h^e variable d'écart décroît jusqu'à 0) lorsque s_j croît jusqu'à b^*_h / A^*_{hj}. Donc, pour une relation $r_k \in R$ donnée, s_j doit croître au moins jusqu'à $\delta_k = \min_{h \in I_k \setminus \{j\} : A^*_{hj} > 0} \{b^*_h / A^*_{hj}\}$.

Cette propriété doit être satisfaite pour chacune des relations de R, et donc s_j doit croître au moins jusqu'à $\delta = \max_{r_k \in R} \delta_k$. La pénalité résultante est alors $p_{ij} = c^*_j \max\{b^*_i / A^*_{ij},\ \delta\}$. Le résultat suit en explicitant δ et δ_k. ∎

Ces pénalités améliorées sont illustrées sur l'exemple de la section 3.3.2. Si elles sont substituées aux pénalités de l'étape **f** de $Al(BLP)$, et si les pénalités de Tomlin [193] remplacent celles de l'étape **f** de $Al(MIP_{0-1})$, alors cet algorithme demeure plongé dans le premier algorithme via la reformulation $MIP_{0-1} \rightarrow BLP$.

3.4.2 Le problème $MIBLP_{0-1}$

Wen et Yang [212] proposent un algorithme d'énumération implicite pour résoudre une sous-classe de $MIBLP_{0-1}$. Il s'agit du cas où les variables de premier niveau sont toutes binaires (les variables x ne sont donc pas présentes) et les seules contraintes de premier niveau sont $u \in \{0,1\}^{n_u}$.

Le coeur de l'algorithme réside dans la façon de borner la valeur optimale de la relaxation de premier niveau de de $MIBLP_{0-1}$. Celle-ci s'écrit comme un problème de la classe MIP_{0-1} :

$$LR_{0-1} \qquad \begin{array}{ll} \max\limits_{x,u} & d^{1t}y + e^{1t}u \\ \text{s.c.} & B^2y + E^2u \leq b^2, \\ & y \geq 0,\ u \in \{0,1\}^{n_u}, \end{array}$$

La borne supérieure proposée par ces auteurs est

$$z^U \equiv z^* + \sum_{i=1}^{n_u} \max\{0,\ (e^{1t} - \lambda^{*t}E^2)_i\},$$

où z^* et λ^* sont les valeur optimale et la solution duale de la relaxation linéaire de LR_{0-1} lorsque la variable u est fixée à 0 (les auteurs supposent que $b^2 \geq 0$. et donc LR_{0-1} est toujours réalisable).

Proposition 3.9 *La borne z^U n'est jamais meilleure que la valeur optimale z_C^* de la relaxation continue de LR_{0-1}.*

Démonstration Soit (y_C^*, u_C^*) une solution optimale de la relaxation continue de LR_{0-1}. La dualité forte et le fait que $u_C^* \in [0,1]^{n_u}$, impliquent que

$$\begin{aligned} z_C^* = d^{1t}y_C^* + e^{1t}u_C^* &= \left(\begin{array}{ll} \min\limits_{\lambda} & \lambda^t(b^2 - E^2u_C^*) \\ \text{s.c.} & \lambda^t B^2 \geq d^{1t}, \\ & \lambda \geq 0 \end{array} \right) + e^{1t}u_C^*, \\ &\leq \lambda^{*t}(b^2 - E^2u_C^*) + e^{1t}u_C^*, \\ &= z^* + (e^{1t} - \lambda^{*t}E^2)u_C^* \leq z^U, \end{aligned}$$

ce qui conclut la preuve. ∎

La borne proposée par $Al(BLP)$ appliquée à la reformulation $MIBLP_{0-1} \rightarrow BLP$ est en fait la valeur optimale de la relaxation continue. Les branchements proposés dans l'algorithme de Wen et Yang se font de façon lexicographique. Aucun autre test n'est utilisé. Il semble que $Al(BLP)$ appliqué à la reformulation $MIBLP_{0-1} \rightarrow BLP$, qui utilise plus d'outils et de plus puissants, résolve plus efficacement le problème $MIBLP_{0-1}$.

3.4.3 Plongement de l'algorithme $Al(BLP)$ dans $Al(MIP_{0-1})$

Le théorème 3.6 montre que $Al(MIP_{0-1})$ est plongé dans $Al(BLP)$ via la reformulation $MIP_{0-1} \rightarrow BLP$. Afin de démontrer le résultat inverse, c'est-à-dire, que $Al(BLP)$ est plongé dans $Al(MIP_{0-1})$, nous devons identifier une transformation de BLP à MIP_{0-1} et comparer les suites de sous-problèmes engendrées par les algorithmes.

Le contre-exemple suivant montre que la reformulation $BLP \Rightarrow MIP_{0-1}$ ne permet pas d'arriver à une telle conclusion.

Exemple 3.10 *La colonne de gauche contient la formulation BLP, et la droite la formulation MIP_{0-1} (pour $L \geq 2$).*

$$\begin{array}{lll} \max\limits_{y} & -x-2y & \\ s.c. & x \geq 0, & \\ & y \in \max\limits_{y} & y \\ & & s.c. \quad x+y \leq 1, \\ & & \qquad y \geq 0, \end{array}$$

$$\begin{array}{ll} \max\limits_{x,y,\lambda,u,v} & -x-2y \\ s.c. & x \geq 0, \\ & 1+L(u-1) \leq x+y \leq 1, \\ & 0 \leq y \leq L(1-v), \\ & 1 \leq \lambda \leq Lu, \\ & \lambda - 1 \leq Lv, \\ & u,\ v \in \{0,1\}. \end{array}$$

La résolution de LR_{BLP} donne $(x_L^*, y_L^*) = (0,0)$ et $z_L^* = 0 > z_{opt} = -\infty$ (étape **b***). Cette solution n'est pas rationnelle puisque la valeur optimale de $FR_{BLP}(x_L^*)$ est $1 \neq 0$ (étapes d et e). La pénalité associée à la contrainte de second niveau $x+y \leq 1$ est $p_1 = 1 \min\{1/1, 2/1\} = 1$ (étape* **f***).*

La résolution de $CR_{MIP_{0-1}}$ donne $(x_C^*, y_C^*, \lambda_C^*, u_C^*, v_C^*) = (0,0,1,1/L,0)$ et $z_C^* = 0 > z_{opt} = -\infty$ (étape **b***). Cette solution n'est pas réalisable puisque $u_C^* \notin \{0,1\}$ (étape* **d***). Les pénalités associées à la variable u sont $p^0 = \infty$ et $p^1 = 0$ (étape* **f***). Ces mauvaises pénalités sont dues à la dégénérescence (la solution $(0, 0, L-1, (L-1)/L, (L-2)/L)$ est aussi une solution optimale de la relaxation $CR_{MIP_{0-1}}$).*

Une conséquence de la dégénérescence inhérente de la reformulation biniveau est que les pénalités seront fréquemment nulles. Le branchement engendré par les deux algorithmes peut alors différer. ∎

Cet exemple suggère, mais ne suffit pas à conclure, que $Al(BLP)$ n'est pas plongé dans $Al(MIP_{0-1})$, puisqu'il peut y avoir une autre transformation de BLP à MIP_{0-1} qui satisfasse les conditions de la définition 3.1.

3.5 Discussion

Ce chapitre décrit une nouvelle façon de comparer des algorithmes qui traitent différentes classes de problèmes d'optimisation. Une telle comparaison diffère des méthodes traditionnelles utilisées dans la théorie de la complexité puisqu'une reformulation polynomiale n'est pas requise. Et donc, elle permet, au moins en théorie, d'étudier le plongement d'algorithmes appliqués à toute paire de problèmes d'optimisation.

Ces concepts sont appliquées aux problèmes BLP et MIP_{0-1}. Le théorème 3.6 montre que $Al(MIP_{0-1})$ est plongé dans $Al(BLP)$ via la reformulation MIP_{0-1} $\rightarrow$ BLP. La transformation de MIP_{0-1} à BLP est simple, c'est-à-dire, elle n'introduit aucune grande constante supplémentaire inconnue, et de plus, elle est une reformulation (étant donné une solution optimale de la reformulation de l'instance de MIP_{0-1}, on peut obtenir en temps polynomial une solution optimale de l'instance en un temps polynomial, voir la définition 1.2). Cette étude suggère que le problème BLP est au moins aussi difficile que MIP_{0-1}. Une hiérarchisation de la difficulté de ces problèmes NP-complets s'installe. Nous avons aussi montré comme extension à la proposition 3.8 qu'un test spécifique à la programmation en nombres entiers se généralise à la programmation biniveau.

À première vue, la découverte qu'un algorithme est plongé dans un autre peut sembler être un résultat négatif, puisque ce premier problème peut alors paraître

redondant. Si tel était le cas, de telles simplifications et unifications pourraient aussi être perçues positivement. Mais cela n'est pas nécessairement le cas. En effet, que $Al(P_A)$ soit plongé dans $Al(P_B)$ signifie que ce dernier algorithme, lorsqu'appliqué à la transformation $B(A)$ du problème A de P_A, génère une suite de sous-problèmes *équivalents via la même transformation* à ceux générés par $Al(P_A)$. Cela ne signifie pas que le temps requis par les deux algorithmes soit comparable. Même si la transformation augmente la taille de P_A par seulement un facteur constant, l'augmentation du temps de calcul peut ne pas être négligeable. Il pourrait être élevé ou même très élevé si la transformation impliquait un accroissement de taille polynomial ou non polynomial. De plus, même si l'augmentation du temps de calcul était modérée, le premier algorithme pourrait mériter d'être conservé, pour des raisons de simplicités ou pédagogiques (nous ne croyons pas que l'algorithme classique de Beale et Small [40] pour le MIP_{0-1} soit remplacé par celui de [96] pour BLP même en considérant les résultats de ce chapitre).

L'intérêt du concept d'algorithme plongé, et ses implications au développement d'algorithmes devient apparent de façon plus indirecte. Premièrement, d'un point de vue algorithmique, la connaissance que $Al(P_A)$ est plongé dans $Al(P_B)$ montre que les structures de P_A et P_B partagent beaucoup de similitude. Et donc, des tests de plusieurs algorithmes pour P_B peuvent être spécialisés pour P_A. Lorsqu'ils ne permettent pas de tirer de nouvelles conclusions, cela illustre que leur force est basée sur l'exploitation de difficultés intrinsèques à P_B qui ne sont pas présentes dans P_A, ce qui met en évidence certaines différences fondamentales entre la structure des problèmes. Inversement, des tests conçus pour P_A peuvent à l'occasion être généralisés à P_B, ce qui met en évidence certaines ressemblances fondamentales entre la structure des problèmes.

Deuxièmement, du point de vue de la complexité, trouver que $Al(P_A)$ est plongé dans $Al(P_B)$ mais que l'inverse ne semble pas vrai, suggère que le problème P_B est plus difficile que P_A, même si les deux problèmes P_A et P_B partage la même classe de complexité (NP-complet où NP-dur par exemple). Ce point pourrait être corroboré par l'étude de d'autres types d'algorithmes pour ces problèmes. En particulier, il pourrait être intéressant d'étudier si des algorithmes de plans de coupure pour le problème MIP_{0-1} sont plongés ou non dans d'autres algorithmes de plans de coupure

pour le BLP.

Troisièmement, l'étude systématique d'algorithmes pourrait révéler une structure de difficulté relative parmi les problèmes NP-complets et NP-durs. La symétrie entre deux algorithmes, c'est-à-dire, le cas où chacun d'eux est plongé dans l'autre via des transformations données, implique qu'il y a une très forte similitude entre les problèmes qu'ils traitent. De plus, le plongement d'algorithme est transitif, et donc la structure induite entre différents problèmes définit un pré-ordre (c'est-à-dire, transitif, réflexif mais pas nécessairement symétrique ou antisymétrique). En utilisant cette structure, on pourrait étudier de quelle façon généraliser ou spécialiser des tests spécifiques à certains problèmes. Éventuellement, cette démarche pourrait engendrer de nouveaux tests et algorithmes, et permettre une meilleure compréhension et organisation de ceux-ci.

CHAPITRE 4

Les jeux bimatriciels

> Dans toute stratégie, on finit éventuellement par élaborer un modèle de l'adversaire qui conditionne ensuite ce que vous allez faire ...
>
> Arturo Perez-Reverte

Un jeu bimatriciel se formule comme suit : étant donné une paire de matrices A et B de $\mathcal{M}_{n\times m}$, le joueur I désire maximiser son gain x^tAy en choisissant le vecteur de probabilité x dans $\mathbb{R}^n$, et simultanément, le joueur II souhaite maximiser son gain x^tBy en choisissant le vecteur de probabilité y dans $\mathbb{R}^m$.

Nash [158] [159] montre qu'il existe toujours un point d'*équilibre*, c'est-à-dire qu'il existe une paire de stratégies $(\hat{x}, \hat{y})$ qui sont des solutions optimales des problèmes linéaires paramétrés suivants :

$$\begin{array}{rl} \hat{x} \in X(\hat{y}) = \arg\max\limits_{x} & x^tA\hat{y} \\ \text{s.c.} & x^t\mathbb{1} = 1, \\ & x \geq 0, \end{array} \quad \text{et} \quad \begin{array}{rl} \hat{y} \in Y(\hat{x}) = \arg\max\limits_{y} & \hat{x}^tBy \\ \text{s.c.} & \mathbb{1}^ty = 1, \\ & y \geq 0. \end{array} \tag{4.1}$$

À un point d'équilibre $(\hat{x}, \hat{y})$ aucun des deux joueurs n'a intérêt à modifier unilatéralement sa stratégie $\hat{x}$ ou $\hat{y}$.

Mills [151] et ensuite Mangasarian et Stone [143] appliquent les conditions d'optimalité de (4.1) afin d'exprimer des conditions nécessaires et suffisantes d'équilibre.

En introduisant les variables duales α et β, les problèmes duaux de (4.1) s'écrivent

$$\begin{array}{ll} \min\limits_{\alpha} & \alpha \\ \text{s.c.} & \mathbb{1}\alpha \geq A\hat{y}, \end{array} \quad \text{et} \quad \begin{array}{ll} \min\limits_{\beta} & \beta \\ \text{s.c.} & \beta\mathbb{1}^t \geq \hat{x}^t B. \end{array} \tag{4.2}$$

La paramétrisation par les variables $\hat{y}$ et $\hat{x}$ n'apparaît plus dans la fonction objectif, mais dans les contraintes. Il s'ensuit que $(\hat{x}, \hat{y})$ est une stratégie d'équilibre si et seulement si il existe des scalaires $\hat{\alpha}$ et $\hat{\beta}$ tels que les contraintes de réalisabilité primale et duale combinées

$$\begin{aligned} (\hat{x}, \hat{\beta}) \in X &\equiv \{(x, \beta) \in \mathbb{R}^{n+1} : \ x^t B \leq \beta\mathbb{1}^t, \ x^t\mathbb{1} = 1, \ x \geq 0\}, \\ (\hat{y}, \hat{\alpha}) \in Y &\equiv \{(y, \alpha) \in \mathbb{R}^{m+1} : \ Ay \leq \mathbb{1}\alpha, \ \mathbb{1}^t y = 1, \ y \geq 0\}, \end{aligned}$$

sont satisfaites, et que les valeurs des fonctions objectifs des problèmes primaux et duaux sont égales pour chacune des deux paires : c'est-à-dire $\hat{x}^t A\hat{y} = \hat{\alpha}$ et $\hat{x}^t B\hat{y} = \hat{\beta}$. Mills [151] et Mangasarian et Stone [143] montrent que ces dernières conditions se combinent en une seule contrainte bilinéaire $\hat{x}^t(A + B)\hat{y} = \hat{\alpha} + \hat{\beta}$. On peut donc poser la question de déterminer un point d'équilibre sous la forme d'un problème d'optimisation globale.

Le lecteur peut se référer à la récente synthèse de McKelvey et McLennan [146] pour un survol des développements récents du calcul d'équilibres en théorie des jeux. Le *Scandinavian Journal of Economics* voue un numéro complet [201] aux travaux de Nash, Harsanyi et Selten qui leur ont valu le prix Nobel d'économie de 1994. La théorie des jeux est un puissant outil de modélisation d'interaction stratégique, lorsque différents agents doivent intéragir en anticipant les réactions respectives de chacun. Osborne et Rubinstein [160] mentionnent qu'un vaste éventail de problématiques peuvent être décrites par la théorie des jeux. Par exemple, les oligopoles peuvent être étudiés par la théorie des équilibres de Nash, et des situations de promesses ou de menaces peuvent être modélisées par la théorie des jeux répétés. De façon un peu moins classique, la théorie des stratégies d'équilibres mixtes peut expliquer la distribution de la longueur de langues d'abeilles et de tubes de fleurs.

Un jeu bimatriciel comporte habituellement plus d'un point d'équilibre, et chacun d'eux possède des caractéristiques différentes ce qui les rend inégalement attrayants. Certains sont dominés (voir par exemple, Luce et Raiffa [139]) par d'autres

selon un critère donné. En particulier, il y a les équilibres de Pareto [50], les équilibres parfaits [177] satisfaisant des critères de stabilité supplémentaires, et des équilibres propres [157] qui renforcent ces conditions. Kohlberg [124] et Van Damme [199] [200] discutent de ces raffinements ainsi que de plusieurs autres.

L'ensemble E de tous les point d'équilibres d'un jeu bimatriciel est l'union d'un nombre fini de polytopes [150] qui peuvent être disjoints. Chacun de ces polytope est appelé un *sous-ensemble de Nash maximal.* Dans ce chapitre, nous étudions l'énumération des points extrêmes de ces polytopes. À partir de ces points extrêmes il est possible d'identifier complètement l'ensemble E.

Un point d'*équilibre extrême* $(\hat{x}, \hat{y})$ est un point d'équilibre satisfaisant les conditions (4.1)

$$\hat{x} \in ext(X(\hat{y})) \quad \text{et} \quad \hat{y} \in ext(Y(\hat{x}))$$

($\hat{x}$ et $\hat{y}$ sont alors appelées les stratégies extrêmes). Il s'ensuit que le point $\hat{x}$ est un sommet de l'ensemble des meilleures répliques à la stratégie $\hat{y}$ et vice versa. Les gains respectifs des joueurs s'écrivent $\hat{\alpha} = \hat{x}^t A\hat{y}$ et $\hat{\beta} = \hat{x}^t B\hat{y}$. Le point $(\hat{x}, \hat{\beta})$ est un sommet de X et $(\hat{y}, \hat{\alpha})$ un sommet de Y. Le nombre de points d'équilibre extrêmes est nécessairement fini puisque le nombre de sommets de X et Y le sont. Mangasarian [141] montre que tout point d'équilibre s'écrit comme une combinaison convexe de points d'équilibre extrêmes.

La connaissance de l'ensemble $ext(E)$ est suffisante pour décrire l'ensemble non convexe E. Vorob'ev [210] montre que si S_x est un ensemble contenant un certain nombre de stratégies extrêmes du joueur I, alors tout point $(\hat{x}, \hat{y})$ pour lequel $\hat{x}$ appartient à l'enveloppe convexe de S_x et où $\hat{y}$ appartient à $Y(x)$ pour chaque stratégie extrême x de S_x, est un point d'équilibre. L'ensemble E est obtenu en considérant l'union de tels ensembles sur tous les sous-ensembles de stratégies extrêmes du joueur I. La formulation mathématique de cet énoncé est :

$$E = \bigcup_{S_x \subseteq \{x:(x,y)\in ext(E)\}} \{conv(S_x) \times \bigcap_{x \in S_x} Y(x)\}.$$

Afin d'obtenir tous les points d'équilibre, il suffit alors de connaître tous les points d'équilibre extrêmes. L'algorithme $Al(BILD)$ présenté à la section 2.5.3 peut

être légèrement modifié afin d'énumérer tous les équilibres extrêmes d'un jeux bimatriciel en considérant le problème bilinéaire disjoint présenté ci-dessous. Cependant, la spécialisation de $Al(BILD)$ à cette question entraîne des simplifications importantes à l'algorithme. La structure bilinéaire, les relations de monotonie, ainsi que les reformulations linéaires maxmin se réduisent à des trivialités. Nous présentons dans ce chapitre une méthode qui énumère tous ces équilibres extrêmes en exploitant les conditions d'optimalité de KKT des deux paires de programmes linéaires paramétrés (4.1) et (4.2). Cet algorithme provient de l'adaptation et de la simplification de $Al(BILD)$.

Ce chapitre est structuré de la façon suivante. La prochaine section présente une revue de la littérature concernant l'obtention et l'énumération de points d'équilibre. De plus, nous montrons la redondance des reformulations LMM du problème bilinéaire disjoint considéré ici. La section 4.2 présente la description détaillée de l'algorithme proposé qui énumére en temps fini tous les équilibres extrêmes. L'avantage de cette approche est que seulement un sous-ensemble des sommets de X et Y est énuméré. Les conditions d'optimalité sont exploitées afin d'élaguer rapidement des branches de l'arbre d'énumération. Aucune hypothèse de non-dégénérescence n'est requise pour prouver la finitude de l'algorithme. Ce dernier est illustré à section 4.2.2 sur un petit exemple, et à la dernière section, il est appliqué à des problèmes générés aléatoirement de taille allant jusqu'à 29×29 lorsque les deux dimensions sont égales, et jusqu'à 700×5 lorsque la seconde dimension est fixée. Le nombre d'équilibres extrêmes croît exponentiellement avec la taille du problème mais demeure modéré pour les problèmes considérés. On retrouve les résultats contenus dans ce chapitre dans Audet *et al.* [17].

4.1 Énumération de points d'équilibre : revue de la littérature

Lemke et Howson [133] proposent une méthode de construction de chemin pour trouver un point d'équilibre dans le cas d'un jeu non dégénéré. Ils présentent aussi

une façon de perturber légèrement un tel jeu afin d'éliminer la dégénérescence. Aggarwal [4] montre que l'algorithme de Lemke et Howson [133] ne peux pas être directement modifié afin d'énumérer tous les équilibres extrêmes puisque certains d'entre eux sont inaccessibles au chemin.

Il existe très peu d'algorithmes dans la littérature qui énumèrent tous les équilibres extrêmes. Le premier est dû à Vorob'ev [210], et fut simplifié ultérieurement par Kuhn [129]. L'idée principale consiste à écrire toutes les sous-matices de B dont les lignes sont indépendantes, et à leur ajouter une colonne de 1. Une vérification de chacune de ces matrices indique si la stratégie correspondante engendre un point d'équilibre.

Mangasarian [141] suggère d'énumérer tous les sommets de X et Y (en utilisant l'algorithme de Balinski [29]), et ensuite de vérifier pour chaque paire de sommets x de X et y de Y si la condition d'optimalité $x^t(A+B)y = \alpha+\beta$ est vérifiée. Winkels [218] propose un algorithme qui diffère de celui de Mangasarian [141] seulement pour la condition d'optimalité : la condition vérifiée étant les écarts complémentaires associés aux problèmes (4.1) et (4.2).

Mukhamediev [154] approche la question d'un point de vue d'optimisation globale. Cette approche qui à première vue semble complètement différente de celles susmentionnées, exploite la formulation bilinéaire disjointe proposée par Mills [151] et Mangasarian et Stone [143] :

$$\max_{\substack{(x,\beta)\in X,\\(y,\alpha)\in Y}} x^t(A+B)y-\alpha-\beta = \max_{(x,\beta)\in X} -\beta + \left(\begin{array}{ll} \max\limits_{y,\alpha} & x^t(A+B)y-\alpha \\ \text{s.c.} & Ay-\mathbb{1}\alpha \leq 0 \\ & \mathbb{1}^t y = 1 \\ & y \geq 0. \end{array} \right)$$

Une solution réalisable est un point d'équilibre si et seulement si la valeur de la fonction objectif est nulle. De plus, zéro borne supérieurement la valeur de la fonction objectif sur le domaine réalisable.

Considérons la reformulation $BILD \leftrightarrow LMM$ (voir la proposition 1.13) qui

introduit les variables duales $\mu \in \mathbb{R}^n$ et $\theta \in \mathbb{R}$:

$$\max_{(x,\beta)\in X} -\beta + \left(\begin{array}{ll} \min\limits_{\mu,\theta} & \theta \\ \text{s.c.} & \mu^t A + \theta \mathbb{1}^t \geq x^t(A+B) \\ & \mu^t \mathbb{1} = 1 \\ & \mu \geq 0. \end{array} \right)$$

Étant donné que la fonction objectif du problème bilinéaire est bornée supérieurement par zéro, il s'ensuit que pour tout $(\hat{x}, \hat{\beta})$ appartenant à X, la valeur optimale $\hat{\theta}$ du problème paramétré de second niveau est bornée supérieurement par $\hat{\beta}$. De plus, si $(\hat{x}, \hat{y})$ est un point d'équilibre où les gains sont $\hat{\alpha}$ et $\hat{\beta}$, alors la valeur optimale du problème de minimisation est nécessairement égale à $\hat{\beta}$ (puisque $\hat{x}^t(A+B)\hat{y} - \hat{\alpha} = \hat{\beta}$).

L'algorithme de Mukhamediev énumère les sommets du polyèdre définissant le domaine réalisable relaxé de la reformulation linéaire maxmin. Cependant, l'information supplémentaire apportée par ce problème maxmin est redondante. En effet, pour toute stratégie d'équilibre $\hat{x}$ de gain $\hat{\beta}$ (pour le joueur II), la solution $\mu = \hat{x}$ et $\theta = \hat{\beta}$ est rationnelle et réduit les contraintes à celles définissant le polyèdre X :

la contrainte $\mu^t A + \theta \mathbb{1}^t \geq \hat{x}^t(A+B)$ devient $\hat{\beta}\mathbb{1}^t \geq \hat{x}^t B$;

les contraintes $\mu^t \mathbb{1} = 1$ et $\mu \geq 0$ deviennent $\hat{x}^t \mathbb{1} = 1$, et $\hat{x} \geq 0$.

Pour cette même raison, l'application de l'algorithme $Al(BILD)$ (légèrement modifié pour l'énumération) n'est pas efficace puisque la force de l'algorithme repose sur l'exploitation des deux reformulations $BILD \leftrightarrow LMM$ symétriques qui sont dans ce cas redondantes. De plus, l'ensemble des relations de monotonie de la reformulation linéaire maxmin n'en comporte qu'une seule. Elle indique que parmi toutes les contraintes $\mu^t A_{\cdot j} + \theta \geq \hat{x}^t(A+B)_{\cdot j}$, $j \in \{1, 2, \ldots, m\}$, au moins une d'entre elles sera à égalité à l'optimalité. Cette condition nécessaire, mais non suffisante, de rationalité apporte très peu d'information.

Le dernier algorithme consiste à énumérer chacun des $(2^n - 1)(2^m - 1)$ supports possibles (l'ensemble de stratégies pures auxquelles sont assignées une probabilité strictement positive), et ensuite vérifier les conditions d'équilibre. Pour un support donné, l'ensemble des points d'équilibre peut être vide, un singleton ou un polytope. Dans ce dernier cas, tous les points extrêmes doivent être énumérés. Dickhaut et

Kaplan [65] ainsi que McKelvey et McLennan [146] implantent cette méthode. Ces derniers résolvent des problèmes dont la taille va jusqu'à 8×8 et mentionnent que des problèmes de taille 12×12 pourraient être résolus.

Toutes les méthodes présentées ci-dessus impliquent plus ou moins directement l'énumération de tous les points extrêmes de polytopes. Le nombre de sommets peut être très élevé, et ce, même lorsque la dimension de l'espace est petite. De plus, d'importantes difficultés combinatoires apparaissent lorsqu'il y a dégénérescence, c'est-à-dire, lorsqu'il y a plusieurs représentations en base d'un même sommet. Des problèmes de taille modérément importante semblent donc difficilement abordables par ces méthodes.

4.2 Méthode de résolution

Rappelons qu'un point d'équilibre extrême $(\hat{x}, \hat{y})$ est une paire de stratégies dont les gains sont $\hat{\alpha} = \hat{x}^t A \hat{y}$ pour le joueur I, et de $\hat{\beta} = \hat{x}^t B \hat{y}$ pour le joueur II, et qui de plus est telle que $(\hat{x}, \hat{\beta})$ est un sommet de X, et $(\hat{y}, \hat{\alpha})$ est un sommet de Y. Nous proposons un algorithme d'énumération implicite qui détermine tous les équilibres extrêmes en énumérant les sommets (x, β) de X et (y, α) de Y qui satisfont les conditions d'écarts complémentaires $x^t(\mathbf{1}\alpha - Ay) = 0$ et $(\beta \mathbf{1}^t - x^t B)y = 0$ des problèmes (4.1) et (4.2). Plusieurs sommets qui ne satisfont pas ces conditions ne sont pas énumérés par l'algorithme.

4.2.1 Algorithme d'énumération des équilibres extrêmes

Considérons les deux programmes linéaires paramétrés dans la fonction objectif :

$$\begin{aligned} P(y) &\equiv \max_{(x,\beta)\in X} x^t A y - \beta, \\ Q(x) &\equiv \max_{(y,\alpha)\in Y} x^t B y - \alpha. \end{aligned}$$

Ces problèmes sont des agrégations des problèmes (4.1) et (4.2). En effet, les contraintes apparaissant dans la définition de X et Y sont celles des problèmes primaux et duaux. De plus, les fonctions objectifs $x^tAy - \beta$ et $x^tBy - \alpha$ sont la somme des fonctions objectifs primales et duales.

L'algorithme génère une suite de sous-problèmes formant un arbre d'exploration. Une paire de *sous-problèmes courants* $\tilde{P}(y)$ et $\tilde{Q}(x)$ est associée à chaque noeud de l'arbre d'exploration. Ces sous-problèmes sont identiques à $P(y)$ et à $Q(x)$, sauf qu'un certain nombre d'inégalités ou de contraintes de non négativité sont fixées à égalité. À tous les noeuds de l'arbre d'exploration, l'une des trois éventualités se produit :

i- ou bien $\tilde{P}(\cdot)$ ou $\tilde{Q}(\cdot)$ est non réalisable;

ii- $\tilde{P}(\cdot)$ et $\tilde{Q}(\cdot)$ sont tous les deux réalisables et il y a suffisamment d'information pour en déduire un point d'équilibre;

iii- $\tilde{P}(\cdot)$ et $\tilde{Q}(\cdot)$ sont tous les deux réalisables mais il n'y a pas assez d'information pour en déduire un point d'équilibre.

Dans ce premier cas, le noeud courant est éliminé. Dans les deux autres cas, le noeud courant (le *père*) est remplacé par plusieurs autres (les *fils*). Chaque fils est identique à son père sauf pour une contrainte de $\tilde{P}(\cdot)$ ou de $\tilde{Q}(\cdot)$, choisie en fonction d'une règle de branchement prédéterminée.

Un noeud est caractérisé par la paire de sous-problèmes courants $\tilde{P}(\cdot)$ et $\tilde{Q}(\cdot)$ ainsi que par une variable x ou y. Sauf possiblement à la racine de l'arbre, la variable x ou y décrit une stratégie réalisable pour un des sous-problèmes courants. Étant donné les problèmes $\tilde{P}(\cdot)$ et $\tilde{Q}(\cdot)$, nous définissons :

$P^i(\cdot)$ comme étant $\tilde{P}(\cdot)$ mais où la contrainte $x_i \geq 0$ est remplacée par $x_i = 0$,
$P_j(\cdot)$ comme étant $\tilde{P}(\cdot)$ mais où la contrainte $x^tB_{\cdot j} \leq \beta$ est remplacée par $x^tB_{\cdot j} = \beta$,
$Q^j(\cdot)$ comme étant $\tilde{Q}(\cdot)$ mais où la contrainte $y_j \geq 0$ est remplacée par $y_j = 0$,
$Q_i(\cdot)$ comme étant $\tilde{Q}(\cdot)$ mais où la contrainte $A_{i\cdot}y \leq \alpha$ est remplacée par $A_{i\cdot}y = \alpha$,

où $i \in \{1, 2, \ldots, n\}$ et $j \in \{1, 2, \ldots, m\}$.

L'algorithme est maintenant brièvement présenté et suivi d'une discussion des étapes les plus importantes.

ALGORITHME EEE (Énumération des Équilibres Extrêmes).

a. Initialisation.
Initialiser l'ensemble des noeuds à explorer $T = \{(x, P(\cdot), Q(\cdot))\}$ où x est fixé à une valeur arbitraire (typiquement $x = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$). Poursuivre à l'étape **b**.

b. Sélection d'un noeud.
Si l'ensemble T des noeuds à explorer est vide, alors terminer l'algorithme. Sinon, choisir et enlever un noeud N de T. Si $N = (x, \tilde{P}(\cdot), \tilde{Q}(\cdot))$ alors aller à l'étape **c**. sinon $N = (y, \tilde{P}(\cdot), \tilde{Q}(\cdot))$ et aller à l'étape **d**.

c. Test de réalisibilité direct $(Q(\cdot))$.
Si le problème $\tilde{Q}(x)$ est non réalisable alors poursuivre à l'étape **b**, sinon choisir $(y, \alpha) \in argmax\tilde{Q}(x)$, mettre à jour $(x, \beta) \in argmax\tilde{P}(y)$ et poursuivre à l'étape **e**.

d. Test de réalisibilité direct $(P(\cdot))$.
Si le problème $\tilde{P}(y)$ est non réalisable alors poursuivre à l'étape **b**, sinon choisir $(x, \beta) \in argmax\tilde{P}(y)$, mettre à jour $(y, \alpha) \in argmax\tilde{Q}(x)$ et poursuivre à l'étape **e**.

e. Branchement.
Soient

$$\tau_i = \begin{cases} x_i(\alpha - A_{i\cdot}y) & \text{la variable } x_i \text{ de } \tilde{P}(\cdot) \text{ n'est pas forcée à être nulle,} \\ & \text{et la contrainte } A_{i\cdot}y \leq \alpha \text{ de } \tilde{Q}(\cdot) \text{ n'est pas forcée} \\ & \text{d'être satisfaite avec égalité,} \\ -1 & \text{sinon,} \end{cases}$$

$$\pi_j = \begin{cases} (\beta - x^t B_{\cdot j})y_j & \text{la variable } y_j \text{ de } \tilde{Q}(\cdot) \text{ n'est pas forcée à être nulle,} \\ & \text{et la contrainte } x^t B_{\cdot j} \leq \beta \text{ de } \tilde{P}(\cdot) \text{ n'est pas forcée} \\ & \text{d'être satisfaite avec égalité,} \\ -1 & \text{sinon,} \end{cases}$$

pour $i = 1, 2, \ldots, n$ et $j = 1, 2, \ldots, m$. Puis choisissons les indices i et j qui maximisent les valeurs respectives de τ_i et π_j.

Si $\tau_i = -1$ et $\pi_j = -1$, alors (x, y) est une stratégie d'équilibre puisque les conditions d'écarts complémentaires sont toutes satisfaites. Mettre à jour la liste des équilibres, puis poser

$$\begin{aligned} T = T \;\; \cup \;\; & \{(y, P^i(\cdot), \tilde{Q}(\cdot)) : x_i > 0\} \cup \{(x, \tilde{P}(\cdot), Q_i(\cdot)) : A_{i\cdot} y < \alpha\} \\ \cup \;\; & \{(x, \tilde{P}(\cdot), Q^j(\cdot)) : y_j > 0\} \cup \{(y, P_j(\cdot), \tilde{Q}(\cdot)) : x^t B_{\cdot j} < \beta\}. \end{aligned}$$

Sinon, deux nouveaux sous-problèmes sont créés. Dans le cas où $-1 \neq \tau_i \geq \pi_j$, poser

$$T = T \cup \{(y, P^i(\cdot), \tilde{Q}(\cdot)),\ (x, \tilde{P}(\cdot), Q_i(\cdot))\}.$$

Autrement, $\tau_i < \pi_j$, poser

$$T = T \cup \{(x, \tilde{P}(\cdot), Q^j(\cdot)),\ (y, P_j(\cdot), \tilde{Q}(\cdot))\}.$$

Poursuivre à l'étape **b**. ∎

À l'étape d'initialisation, l'ensemble des noeuds à explorer est défini de façon à ce qu'il n'en comporte qu'un seul : celui correspondant au jeu bimatriciel original. Les fonctions objectif des problèmes $P(\cdot)$ et $Q(\cdot)$ sont choisies de sorte que les valeurs $x^t A y$ et $x^t B y$ soient élevées, et qu'au moins une contrainte apparaissant dans $x^t B \leq \beta \mathbb{1}^t$ et dans $Ay \leq \mathbb{1}\alpha$ soit satisfaite à égalité. L'unique relation de monotonie de chacune des reformulations symétriques linéaires maxmin est alors satisfaite. Les relations de monotonie ne peuvent donc pas enrichir l'algorithme. Les solutions résultantes sont fréquemment des solutions optimales des paires de problèmes (4.1) et (4.2).

L'ordre dans lequel les noeuds sont choisis dans l'ensemble T (à l'étape **b**) importe peu puisqu'ils doivent éventuellement tous être choisis afin d'assurer l'énumération exhaustive des équilibres extrêmes.

Les étapes **c** et **d** sont celles de retour arrière. Pour chacune d'elles, la vérification de la réalisibilité d'un seul problème $\tilde{P}(\cdot)$ ou $\tilde{Q}(\cdot)$ est requise puisque qu'un seul diffère de leur père (l'identification du problème modifié s'effectue par la variable x ou y). Lorsque ce problème est réalisable, ces étapes se terminent avec une solution réalisable (x, y). La résolution des programmes linéaires paramétrés se fait par un algorithme du simplexe, et donc les solutions sont des sommets de X et Y (contrairement à l'utilisation d'un algorithme de points intérieurs).

L'étape e crée de nouveaux noeuds (les fils du noeud courant) dans l'arbre d'exploration. Lorsque la profondeur du noeud courant dans l'arbre d'exploration est inférieure ou égale à $n + m$ (et donc, au moins une contrainte de complémentarité linéaire n'est pas forcée à être respectée), alors deux nouveaux fils sont créés : un dans lequel une variable x_i est fixée à 0 ou bien une contrainte $x^t B_{\cdot j} \leq \beta$ est remplacée par $x^t B_{\cdot j} = \beta$, et l'autre où une variable y_j est fixée à 0 ou bien une contrainte $A_{i\cdot} y \leq \alpha$ est remplacée par $A_{i\cdot} y = \alpha$. La figure 4.1 illustre ces règles de branchement.

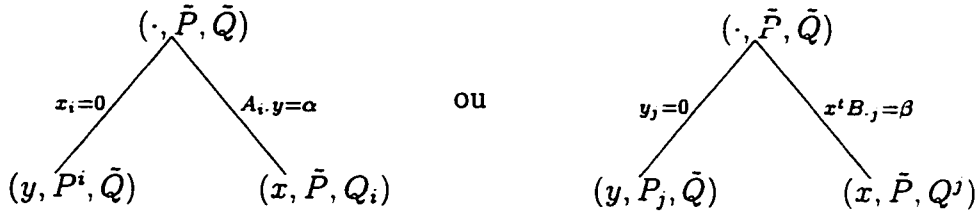


Figure 4.1 – *Règles de branchement de l'algorithme EEE*

Lorsque la profondeur du noeud courant dans l'arbre d'exploration est supérieure à $n + m$, la solution correspondante (x, y) est nécessairement un équilibre extrême puisque chacune des conditions d'écarts complémentaires est satisfaite, et la réalisibilité de la solution est assurée par les étapes **c** et **d**. Un nouveau fils est créé pour chacune des variables (incluant les variables d'écarts) strictement positives. Ce branchement est nécessaire puisque certains points d'équilibre extrêmes peuvent être tels que les conditions d'écarts complémentaires soient fortement satisfaites, c'est-à-dire que $x_i = 0$ et $A_{i\cdot} y = \alpha$ ou bien que $y_j = 0$ et $x^t B_{\cdot j} = \beta$. Dans ce cas, il y a dégénérescence et il est possible que le même équilibre soit obtenu à plusieurs noeuds de l'arbre d'exploration, mais il n'est enregistré qu'une seule fois.

Théorème 4.1 *L'algorithme EEE énumère en temps fini tous les points d'équilibre extrêmes d'un jeu bimatriciel.*

Démonstration Soit $(\hat{x}, \hat{y})$ un point d'équilibre extrême, dont les gains $\hat{\alpha}$ et $\hat{\beta}$ sont respectivement $\hat{x}^t A \hat{y}$ et $\hat{x}^t B \hat{y}$. Le point $\hat{s} = (\hat{x}, \hat{y}, \hat{\alpha}, \hat{\beta})$ est alors un point extrême de $X \times Y$. Définissons les ensembles d'indices indiquant quelles contraintes sont serrées

à ce point :

$$I^x = \{i : \hat{x}_i = 0\};\quad J^y = \{j : \hat{y}_j = 0\};\quad I^y = \{i : A_{i\cdot}\hat{y} = \alpha\};\quad J^x = \{j : \hat{x}^t B_{\cdot j} = \beta\}.$$

Par définition, le point $\hat{s}$ appartient à l'ensemble

$$S = \left\{ (x,\beta,y,\alpha) \in X \times Y : \begin{array}{rr} x_i = 0\ \forall i \in I^x; & y_j = 0\ \forall j \in J^y; \\ A_{i\cdot}y = \alpha\ \forall i \in I^y; & x^t B_{\cdot j} = \beta\ \forall j \in J^x; \end{array} \right\}.$$

Supposons que l'ensemble S ne soit pas un singleton, c'est-à-dire, qu'il existe un point s' dans S distinct de $\hat{s}$. Pour un $\epsilon > 0$ donné, considérons le point $s_\epsilon = s' + \epsilon(\hat{s} - s')$. Si $\epsilon \in]0, 1]$, alors la convexité de S garantit l'appartenance de s_ϵ à S. Rappelons que $\hat{s}$ est un point extrême de $X \times Y$; il s'ensuit que pour $\epsilon > 1$, s_ϵ n'appartient ni à $X \times Y$ ni à S. Il en résulte l'existence d'une contrainte serrée au point $\hat{s}$ qui ne l'est pas au point s'. Puisque S est défini par l'entremise de toutes les contraintes serrées en $\hat{s}$, alors s' ne peut appartenir à S. Cette contradiction assure que l'ensemble S est le singleton $\{(\hat{x}, \hat{\beta}, \hat{y}, \hat{\alpha})\}$.

Les conditions d'écarts complémentaires de (4.1) et (4.2) assurent que ces ensembles d'indices satisfont $I^x \cup I^y = \{1, 2, \ldots, n\}$ et $J^x \cup J^y = \{1, 2, \ldots, m\}$. Dans le cas où le point d'équilibre n'est pas dégénéré, alors ces paires d'ensembles sont disjoints, c'est-à-dire, $I^x \cap I^y = \emptyset$ et $J^x \cap J^y = \emptyset$. S'il y a dégénérescence, les multiples représentations en base impliquent des intersections non vides.

Considérons maintenant le cas où le point d'équilibre extrême $(\hat{x}, \hat{y})$ n'est pas encore détecté par l'algorithme. À un noeud de l'arbre d'exploration de profondeur moindre que $|I^x|+|I^y|+|J^x|+|J^y|$ (cette somme vaut $n+m$ dans le cas non dégénéré), la règle de branchement choisit une variable i ou j de façon à ce qu'un des fils de ce noeud aura nécessairement une contrainte supplémentaire parmi :

$$\begin{array}{llll} x_i = 0 \text{ où } i \in I^x, & \text{ou} & y_j = 0 \text{ où } j \in J^y, & \text{ou} \\ A_{i\cdot}y = \alpha \text{ où } i \in I^y, & \text{ou} & x^t B_{\cdot j} = \beta \text{ où } i \in J^x. & \end{array}$$

Si $(\hat{x}, \hat{y})$ est un équilibre extrême du problème associé au noeud père, alors il le sera aussi pour le fils en question puisque cette nouvelle contrainte ne rend pas le problème du fils non réalisable.

Il y aura donc un noeud de profondeur $|I^x| + |I^y| + |J^x| + |J^y|$ contenant les contraintes de l'ensemble S. À ce noeud, l'étape **e** de algorithme découvre le point d'équilibre extrême $(\hat{x}, \hat{y})$.

Le nombre de contraintes et de variables des ensembles X et Y est fini. Dans l'arbre d'exploration, chaque noeud possède un problème ayant une variable (ou variable d'écart) de plus que son père, qui est fixée à 0, et donc en vertu des contraintes définissant (4.1), chaque branche de l'arbre est voué à se terminer avec un problème non réalisable. Il en résulte que le nombre de noeuds exploré est fini.

De plus, le traitement de chaque noeud se fait en temps fini puisqu'à chaque fois, seulement deux programmes linéaires (dont la taille est bornée par $P(\cdot)$ et $Q(\cdot)$) sont résolus. Le temps requis par l'algorithme est alors fini. ∎

L'algorithme EEE peut être interprété comme une spécialisation de la seconde phase de l'algorithme $Al(BILD)$ (présenté à la section 2.5.3). En effet, la question de déterminer un point d'équilibre se reformule comme un problème de programmation bilinéaire disjoint. Le choix de la variable de branchement et les tests d'optimalité sont simplifiés. L'étape de retour arrière est éliminée puisque l'on désire obtenir tous les points d'équilibre extrêmes.

4.2.2 Exemple d'un jeu bimatriciel

La méthode est illustrée sur un exemple de dimension 2×2 tiré d'un article fondamental de Nash [158], comprenant un seul point d'équilibre :

$$A = \begin{bmatrix} 1 & -10 \\ 10 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 10 \\ -10 & -1 \end{bmatrix}.$$

À l'étape d'initialisation, les problèmes

$$\begin{aligned} P(y) \;=\; & \max_{x\geq 0,\beta} (y_1 - 10y_2)x_1 + (10y_1 - y_2)x_2 - \beta \\ & \text{s.c.} \qquad x_1 + x_2 = 1, \\ & \qquad\quad x_1 - 10x_2 \leq \beta, \\ & \qquad\quad 10x_1 - x_2 \leq \beta, \\ Q(x) \;=\; & \max_{y\geq 0,\alpha} (x_1 - 10x_2)y_1 + (10x_1 - x_2)y_2 - \alpha \\ & \text{s.c.} \qquad y_1 + y_2 = 1, \\ & \qquad\quad y_1 - 10y_2 \leq \alpha, \\ & \qquad\quad 10y_1 - y_2 \leq \alpha, \end{aligned}$$

ainsi que $x^t = (\frac{1}{2}, \frac{1}{2})$ constituent le premier noeud placé dans l'ensemble T. Ce noeud est choisi à l'étape **b** et retiré de T. L'étape **c** résout $Q(\frac{1}{2}, \frac{1}{2})$ engendrant $y_1 = 0, y_2 = 1$, puis résout $P(0,1)$ donnant $x_1 = 0, x_2 = 1$. À la fin de cette étape, $x^t = y^t = (0,1)$, $\alpha = \beta = -1$ et les vecteurs des variables d'écarts associées aux contraintes $Ay \leq \mathbf{1}\alpha$ et $x^t B \leq \beta \mathbf{1}^t$ sont tous deux égaux à $(9,0)$.

L'étape de branchement choisit l'indice $i = 1$. L'ensemble T comporte maintenant deux noeuds : un premier dans lequel la contrainte $x_1 = 0$ est ajoutée à $\tilde{P}(\cdot)$ (avec $y^t = (0,1)$ comme solution initiale de $\tilde{Q}(\cdot)$), et un second dans lequel la contrainte $A_{1\cdot} y = \alpha$ est ajoutée à $\tilde{Q}(\cdot)$ (avec $x^t = (0,1)$ comme solution initiale de $\tilde{P}(\cdot)$).

La figure 4.2 illustre l'arbre d'exploration complet créé par l'algorithme. La sélection des noeuds à l'étape **b** se fait selon une stratégie de *profondeur d'abord*. Les numéros encerclés indiquent l'ordre dans lequel les noeuds sont traités.

Nous décrivons quelques noeuds importants de l'arbre d'exploration. Au noeud 3, le sous-problème courant $\tilde{P}(\cdot)$ est non réalisable puisque la contrainte $x_1 + x_2 = 1$ est en contradiction avec les deux nouvelles contraintes $x_1 = x_2 = 0$. L'étape **d** détecte ce fait, et effectue un retour arrière à l'étape **b**. L'ensemble T contient alors les noeuds 4 et 13.

Au noeud 7, chacune des conditions d'écarts complémentaires est satisfaite, les

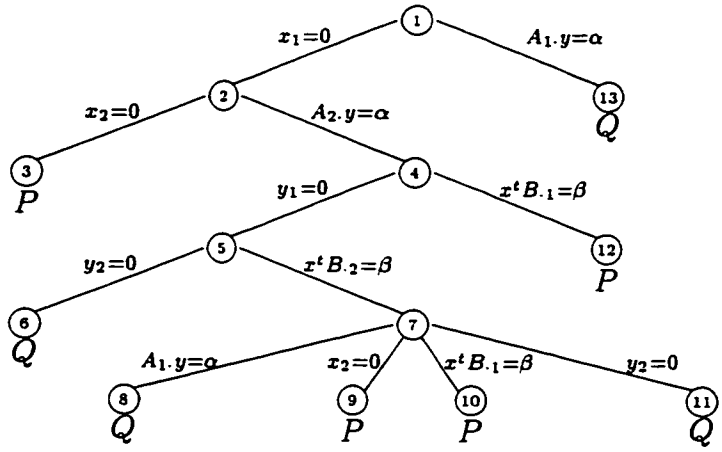


Figure 4.2 – *Arbre d'exploration de l'algorithme EEE*

sous-problèmes courants $\tilde{P}(\cdot)$ et $\tilde{Q}(\cdot)$ sont réalisables, et conséquemment, l'unique point d'équilibre est trouvé. Même si ce point $x^t = y^t = (0, 1)$ fut obtenu à la racine de l'arbre, il n'est enregistré qu'à ce noeud plus profond. Ceci réduit le temps passé à vérifier si un point est un équilibre ou non. Le temps requis pour trouver un premier équilibre pourrait alors être réduit, mais ce n'est pas notre objectif. À l'étape **e**, quatre nouveaux noeuds sont ajoutés à l'ensemble T. Le premier correspond à la variable d'écart strictement positive associée à la contrainte $A_1.y = y_1 - 10y_2 \leq \alpha$, le second à $x_2 \geq 0$, le troisième à la variable d'écart associée à la contrainte $x^tB_{.1} = x_1 - 10x_2 \leq \beta$, et le dernier à $y_2 \geq 0$. Chacun des ces nouveaux noeuds engendre un sous-problème non réalisable détecté à l'étape **c** ou **d**. L'ensemble T contient ensuite les noeuds 12 et 13.

Au noeud 13, le sous-problème $\tilde{Q}(\cdot)$ est non réalisable. En effet, la combinaison linéaire de la contrainte $y_1 - 10y_2 = \alpha$ à neuf fois $y_1 + y_2 = 1$, viole la contrainte $10y_1 - y_2 \leq \alpha$. L'étape **c** détecte que ce problème n'est pas réalisable et effectue un retour arrière à l'étape **b**. L'ensemble T est maintenant vide, ce qui complète l'exécution de l'algorithme.

L'utilisation des conditions d'écarts complémentaires est la clef de l'efficacité de l'algorithme. Pour illustrer ce point, considérons le noeud 13. Comme mentionné ci-dessus, la condition d'écart complémentaire $A_1.y = \alpha$ obtenue par la règle de

branchement permet l'élagage immédiat de la branche. Si cette règle était remplacée par $x_1 = 0$ versus $x_1 > 0$, l'élagage ne serait pas immédiatement possible puisqu'il existe un point extrême $(x, \beta) = (1, 0, 10)$ de l'ensemble X satisfaisant la contrainte $x_1 > 0$. Ce point serait alors un candidat potentiel à une stratégie d'équilibre extrême. Sous cette règle de branchement affaiblie, l'étude du noeud 13 devrait être poursuivie. Il en résulterait un plus vaste arbre d'exploration.

4.2.3 Résultats numériques

L'algorithme est écrit en C et utilise les librairies CPLEX2.1 pour résoudre les programmes linéaires paramétrés $\tilde{P}(\cdot)$ et $\tilde{Q}(\cdot)$. Une station SPARC SS20/514MP sous Solaris 2.4-27 est utilisée pour les expériences de calcul.

L'application de algorithme à l'exemple dégénéré de Winkel [218] de taille 6×2 révèle un point d'équilibre extrême omis par l'auteur : $x^t = (\frac{7}{8}, 0, 0, 0, 0, \frac{1}{8}), y^t = (\frac{1}{4}, \frac{3}{4})$. L'énumération des douze équilibres extrêmes requiert approximativement trois quarts de secondes et 417 noeuds dans l'arbre d'exploration.

Le tableau 4.1 présente les moyennes (μ) et les écarts types (σ) de dix problèmes générés aléatoirement où les coefficients des matrices A et B sont issus d'une distribution uniforme sur l'intervalle $[0, 10]$. Les colonnes *temps* et *noeuds* indiquent les temps en secondes et le nombre de noeuds requis par l'arbre d'exploration. Les colonnes *équilibre* indiquent le nombre *total* d'équilibres extrêmes, ainsi que le numéro du noeud où le *premier* équilibre extrême est trouvé.

Le nombre total de points d'équilibre extrêmes ne croît pas très rapidement. Étonnamment, il demeure presque stable pour des problèmes de grande taille lorsque m est fixé à 5. Le numéro moyen du noeud où le premier équilibre est trouvé est passablement élevé puisque la méthode vérifie les conditions d'équilibre qu'aux noeuds dont la profondeur dépasse $m + n$.

Tableau 4.1 – *Résultats pour des problèmes générés aléatoirement*

$m = 5$				
	temps	noeuds	équilibre	
n	(sec)		total	premier
100 μ	81,3	11189	19,6	484,8
σ	33,0	4808	10,6	307,8
200 μ	400,1	34864	27,2	1436,0
σ	186,0	16202	12,9	1187,9
300 μ	1190,0	71865	34,2	2224,0
σ	429,8	26301	14,1	1909,5
400 μ	2303,6	105576	38,6	1659,2
σ	861,5	40074	15,0	528,9
500 μ	3992,6	147541	41,6	2948,6
σ	1421,3	53457	17,7	2466,4
600 μ	6523,2	199437	47,6	4989,8
σ	2631,5	79950	15,2	3733,7
700 μ	10060,4	263500	47,8	4246,6
σ	2656,8	71379	9,3	2955,1

$m = 10$				
	temps	noeuds	équilibre	
n	(sec)		total	premier
30 μ	59,0	11138	45,2	205,4
σ	26,7	4781	18,7	131,6
50 μ	143,2	23599	63,2	219,0
σ	109,0	15561	32,1	288,4
70 μ	696,0	86059	105,4	549,2
σ	275,2	34126	43,5	515,6
90 μ	1641,5	166143	155,8	944,0
σ	659,7	66673	66,3	687,9
110 μ	3108,8	278523	182,0	739,2
σ	1290,0	115584	83,2	869,6
130 μ	5822,0	452405	222,6	3483,2
σ	1717,3	130085	75,3	4606,8
150 μ	9583,3	677202	266,0	1122,6
σ	2817,4	192742	69,5	1439,8

$m = 15$				
	temps	noeuds	équilibre	
n	(sec)		total	premier
10 μ	10,7	2768	20,2	89,0
σ	8,1	2015	12,9	41,9
20 μ	73,7	12970	58,6	173,0
σ	41,9	6889	28,5	125,2
30 μ	330,1	50118	126,0	198,8
σ	195,1	28996	77,9	141,5
40 μ	1150,3	143245	222,8	508,4
σ	683,7	83440	122,6	689,1
50 μ	3252,4	345908	355,8	666,4
σ	1745,6	174208	127,8	591,7
60 μ	6898,2	674661	491,6	1194,2
σ	2961,0	274986	171,1	1254,7
70 μ	13671,3	1196031	666,5	1028,0
σ	5803,0	504102	227,6	1165,2

$m = n$				
	temps	noeuds	équilibre	
n	(sec)		total	premier
5 μ	0,2	96	3,2	16,6
σ	0,0	20	1,1	5,1
9 μ	1,6	541	8,4	38,2
σ	1,2	403	6,3	20,4
13 μ	9,5	2323	18,0	80,4
σ	6,2	1471	11,0	55,2
17 μ	83,4	15421	61,4	238,8
σ	78,6	12648	33,9	190,6
21 μ	705,7	97725	204,4	475,8
σ	671,9	85925	144,1	470,4
25 μ	2427,6	280202	441,3	452,4
σ	1190,5	133640	226,8	586,6
29 μ	16216,0	1444924	1254,5	1597,6
σ	11214,7	921033	679,6	3731,4

La figure 4.3 présente la relation entre le logarithme des temps d'exécution en fonction de la taille des problèmes considérés.

Lorsque $n = m$, il semble que le temps requis par l'algorithme croît exponentiellement avec la valeur de n. Pour de grandes valeurs de n, certains problèmes sont plus difficiles à résoudre, ce qui entraîne un écart type élevé. En particulier, un des dix problèmes généré pour $n = 21$ nécessita 2140 secondes et 272034 noeuds. Lorsque m est fixé à 5, 10 ou 15, le temps d'exécution semble tendre asymptotiquement vers

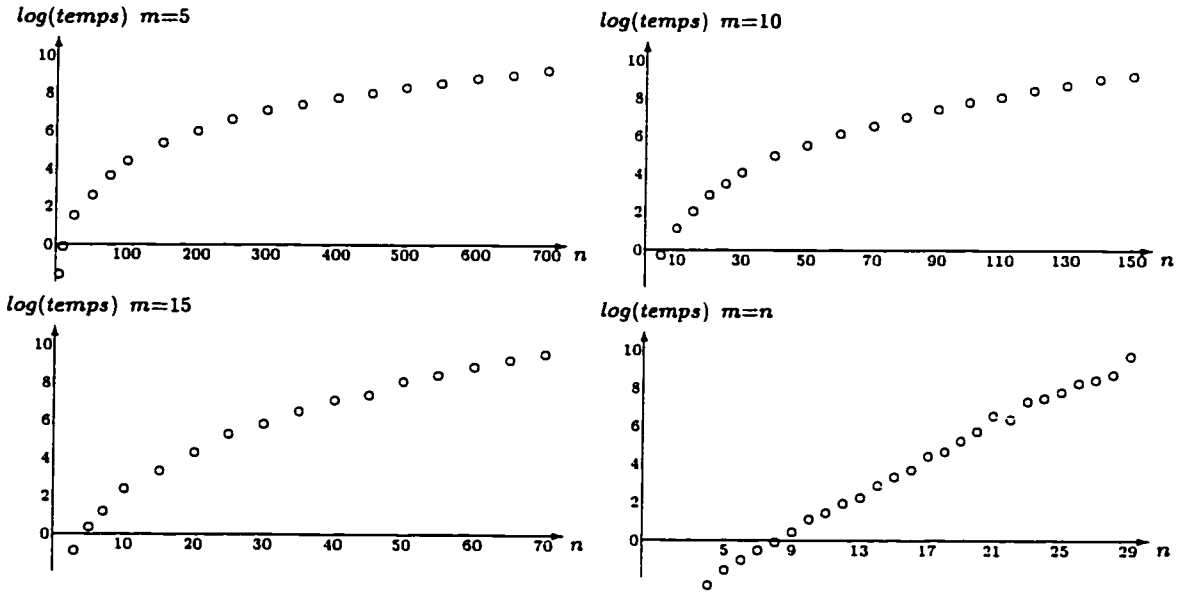


Figure 4.3 – *Logarithme des temps d'exécution*

une croissance exponentielle puisque le logarithme du temps d'exécution approche un comportement linéaire pour de grandes valeurs de n. On observe que le temps requis par l'algorithme à chaque noeud est approximativement proportionnel au produit des dimensions n et m.

4.3 Discussion

L'algorithme présenté à la section 4.2 permet l'énumération de tous les points d'équilibre extrêmes pour des jeux bimatriciel dont la taille est substantiellement plus élevée que celle des jeux considérés par les algorithmes précédents. De plus, le nombre de tels équilibres, même s'il peut croître exponentiellement avec la taille du problème, tend à être modéré. Rappelons cependant que ces observations sont issues de problèmes générés aléatoirement. Différents résultats pourraient être obtenus pour d'autres problèmes, en particulier pour des problèmes issus d'applications réelles ou pour ceux générés par des méthodes différentes.

Les applications économiques des jeux bimatriciels font intervenir des raffinements des équilibres de Nash. Pour certains d'entre eux, il est difficile de les identifier lorsqu'on ne connaît pas l'ensemble des points d'équilibre extrêmes. Une voie de recherche ultérieure serait d'ajouter à l'algorithme une série de tests afin de déterminer l'appartenance des équilibres à différentes classes. Pour certaines d'entre elles (par exemple, les équilibres complètement mixtes, Pareto, ou réguliers), la reconnaissance est triviale, tandis que pour plusieurs autres, la difficulté est importante. Il s'ensuit qu'une exploration assistée par ordinateur des raffinements des équilibres de Nash semble possible même pour des jeux bimatriciels de grande taille.

CHAPITRE 5

Le problème quadratique

> Il scintillait, limpide et buté, cachant derrière sa transparence son dur secret. En viendrai-je à bout?
>
> Simone de Beauvoir

Le problème quadratique à contraintes quadratiques est d'une telle généralité qu'il englobe tous les problèmes d'optimisation présentés à la section 1.1. Plusieurs autres, comme l'optimisation polynomiale et fractionnaire se reformulent également en problème QQP. Cette flexibilité entraîne des difficultés importantes lors de la résolution. Remarquons à cet effet que même dans le cas où le domaine réalisable est polyédrique, le problème demeure fortement NP-complet car ce dernier contient le problème $BILD$. Le haut degré de difficulté ne réside pas uniquement dans la fonction objectif. Même si on l'ignore complètement, la simple question de trouver une solution réalisable de QQP définit encore une fois un problème fortement NP-complet puisque le domaine réalisable, non convexe et non connexe, généralise le problème de complémentarité linéaire décrit à la section 1.1.6. Il en résulte qu'une approche itérative de type Gauss-Seidel (voir la section 2.1.2) appliquée à la reformulation BIL de QQP ne produit pas nécessairement une solution réalisable. Une autre difficulté importante associée à ce problème est que même si tous les coefficients apparaissant dans une formulation sont entiers, il est possible que la solution optimale soit irrationnelle. On ne doit donc pas s'attendre à trouver d'ici peu un algorithme exact qui résolve en temps raisonnable des problèmes de grande taille.

Nous utilisons une formulation différente mais équivalente à celle introduite à la section 1.1.5. Cette nouvelle forme alourdit légèrement l'énoncé du problème mais

simplifie son traitement et sa manipulation. Nous considérons dans ce chapitre le problème quadratique à contraintes quadratiques suivant

$$QQP \qquad \begin{array}{ll} \min\limits_{x \in X} & Q^0(x) \\ \text{s.c.} & Q^k(x) \mathrel{\overline{\leq}} b_k \qquad k = 1, 2, \ldots, \overline{k}, \end{array}$$

où $X = \{x \in \mathbb{R}^n : \ Ax \leq a\}$ et

$$\begin{array}{rcl} Q^k : \mathbb{R}^n & \to & \mathbb{R} \\ x & \mapsto & Q^k(x) = \sum\limits_{(i,j) \in M} C_{ij}^k x_i x_j + \sum\limits_{i \in N} c_i^k x_i^2 + \sum\limits_{i \in N} d_i^k x_i, \end{array}$$

définissent des fonctions quadratiques pour chaque indice k appartenant à $K = \{0, 1, \ldots, \overline{k}\}$, et où $N = \{1, 2, \ldots, n\}$ et $M = \{(i,j) \in N \times N : i > j\}$ sont des ensembles d'indices. Les dimensions des matrices et des vecteurs sont les suivantes :

$$x \in \mathbb{R}^n; A \in \mathcal{M}_{m \times n}; a \in \mathbb{R}^m; b \in \mathbb{R}^{\overline{k}};$$
$$C_{ij}^k \in \mathbb{R} \text{ et } c^k, d^k \in \mathbb{R}^n \text{ pour chaque } (i,j) \in M \text{ et } k \in K.$$

Comme dans la plupart des articles concernant le problème QQP, nous adoptons la forme de minimisation et non de maximisation. La définition du polyèdre X est simplifiée de façon à ce qu'elle ne comprenne pas de contraintes d'égalité. Nous supposons que la contrainte $x \geq 0$ est ou bien présente dans $Ax \leq a$ ou bien implicite dans l'ensemble des contraintes. Nous supposons aussi qu'il soit possible d'évaluer des bornes positives finies sur toutes les variables. Cette hypothèse est décrite à la section 5.2.1.

La difficulté de QQP réside dans les termes quadratiques présents à la fois dans la fonction objectif et dans les contraintes. Considérons les deux fonctions

$$\begin{array}{rcl} f : \mathbb{R} & \to & \mathbb{R} \\ x_i & \mapsto & f(x_i) = x_i^2 \end{array} \qquad \text{et} \qquad \begin{array}{rcl} g : \mathbb{R}^2 & \to & \mathbb{R} \\ (x_i, x_j) & \mapsto & g(x_i, x_j) = x_i x_j. \end{array}$$

L'approximation de la fonction f est plus facile que celle de g puisqu'elle est convexe sur son domaine. Toute droite tangente à f en un point α définit une sous-estimation valide sur tout le domaine $\mathbb{R}$. Une étude plus détaillée est requise pour la fonction g. Considérons le plan tangent à la fonction g au point (α_i, α_j) de $\mathbb{R}^2$. Le hessien de

la fonction s'écrit $\nabla^2 g(x_i, x_j) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, et alors $[\delta_i \ \delta_j] \nabla^2 g(x_i, x_j) \begin{bmatrix} \delta_i \\ \delta_j \end{bmatrix} = 2\delta_i\delta_j$.
Il en résulte qu'au point $(\alpha_i + \delta_i, \alpha_j + \delta_j)$, le plan tangent

- coïncide avec g lorsque δ_i ou δ_j est nul,
- est une sous-estimation de g lorsque $\delta_i\delta_j > 0$
  (la fonction g est alors convexe dans la direction (δ_i, δ_j)),
- est une surestimation de g lorsque $\delta_i\delta_j < 0$
  (la fonction g est alors concave dans la direction (δ_i, δ_j)).

Nous présentons dans ce chapitre des méthodes d'approximation de tous les termes quadratiques par des linéarisations successives. Quatre classes de linéarisation sont développées, et pour chacune d'elles la meilleure (selon un critère défini ci-dessous) est choisie. Ces techniques permettent l'élaboration d'un algorithme générant une solution approchée dans l'objectif et les contraintes de QQP. L'étude de ce problème est motivée par sa flexibilité qui permet une modélisation précise de plusieurs situations réelles, et dans sa généralité qui lui permet d'englober un grand nombre de questions d'optimisation. L'algorithme est appliqué à plusieurs problèmes tirés de situations réelles. Ces résultats apparaissent aussi dans Audet *et al.* [20].

5.1 Le problème quadratique à contraintes quadratiques : revue de la littérature

Au cours des dernières années, plusieurs auteurs se sont penchés sur le problème de la linéarisation de fonctions non linéaires. Al-Khayyal et Falk [9] ont développé une méthode d'énumération implicite pour un problème plus général que BIL (rappelons que les problèmes BIL et QQP sont équivalents). Ce problème contient deux ensembles de variables et repose seulement sur les trois hypothèses suivantes :

i- la fonction objectif est biconvexe
(c'est-à-dire, convexe lorsque les variables d'un des deux ensembles sont fixées);

ii- le domaine réalisable est convexe et fermé;

iii- on peut aisément calculer des bornes sur les variables : $x_i \in [\ell_i, u_i]$.

La méthode repose sur une approximation extérieure de la fonction de deux variables $g(x_i, x_j) = x_i x_j$ en utilisant l'enveloppe convexe sur un hyperrectangle $[\ell_i, u_i] \times [\ell_j, u_j]$. Une telle linéarisation est exacte seulement sur la frontière de cet ensemble. Si une solution réalisable (α_i, α_j) de cette relaxation se trouve dans l'intérieur strict de l'hyperrectangle, alors la linéarisation doit être raffinée. De nouvelles linéarisations sont alors évaluées sur les quatre sous-ensembles $[\ell_i, \alpha_i] \times [\ell_j, \alpha_j]$, $[\alpha_i, u_i] \times [\ell_j, \alpha_j]$, $[\ell_i, \alpha_i] \times [\alpha_j, u_j]$ et $[\alpha_i, u_i] \times [\alpha_j, u_j]$. Ces quatre nouveaux sous-problèmes sont récursivement examinés. Il apparaît que la difficulté d'un problème est directement reliée au nombre de variables impliquées dans des termes quadratiques qui à l'optimalité ne sont pas à l'une de leurs bornes.

Al-Khayyal [7] améliore cette méthode d'énumération implicite en évaluant aussi l'enveloppe concave de la fonction. Puis, Al-Khayyal [8], spécialise cette idée à BIL en linéarisant non seulement la fonction objectif, mais aussi le domaine réalisable. Al-Khayyal *et al.* [11] proposent une version légèrement différente de cette méthode pour le problème QQP. Au lieu de générer quatre nouveaux sous-problèmes, cette dernière méthode en crée seulement deux, en subdivisant le plus long intervalle en son milieu. Des résultats numériques sont présentés sur des problèmes générés au hasard ayant jusqu'à seize variables et huit contraintes.

Sherali et Alamedine [181] [182] améliorent la linéarisation de BIL (où il n'y a pas de contraintes quadratiques) en considérant le produit de toutes les paires de contraintes définissant le polyèdre X au lieu d'uniquement celles définissant les bornes sur les variables. Sherali et Tuncbilek [185] généralisent cette méthode d'énumération implicite au cas où les fonctions Q^k sont polynomiales. Leur approche ne consiste pas à reformuler le problème polynomial en problème quadratique en ajoutant plusieurs nouvelles variables et contraintes. Ils introduisent plutôt des linéarisations de degré supérieur à deux. Sherali et Tuncbilek [186] spécialisent cette approche à QQP où il n'y a pas de contraintes quadratiques. Ils discutent de plusieurs améliorations telles que la linéarisation de termes cubiques, la diagonalisation de la matrice Q^0 (lorsque c'est possible), l'approximation convexe de la fonction $f(x_i) = x_i^2$ ainsi

que la résolution de la relaxation par une méthode lagrangienne. Sherali et Tuncbilek [187] comparent différentes méthodes d'évaluation de bornes pour l'optimisation polynomiale.

La décomposition généralisée de Benders [43] définit une autre voie de recherche pour la résolution de BIL. De telles approches sont décrites dans Geoffrion [88], Wolsey [219], Simoes [188], Flippo [75], Floudas et Visweswaran [78], [79], Visweswaran et Floudas [206], ainsi que Flippo et Rinnooy Kan [76].

5.2 Linéarisation des termes quadratiques

Les linéarisations des fonctions quadratiques f et g présentées à la section 5.2.2 proviennent d'approximations extérieures. Pour chaque i appartenant à N, la variable $v_i \geq 0$ est introduite afin d'estimer le carré x_i^2, et pour chaque (i,j) appartenant à M, la variable additionnelle w_{ij} sert à estimer le produit $x_i x_j$. Des contraintes impliquant les variables x_i, v_i et w_{ij} sont successivement ajoutées de façon à raffiner l'approximation tout en garantissant que la solution où $v_i = x_i^2$ et $w_{ij} = x_i x_j$ demeure réalisable.

Nous utilisons la notation introduite par Sherali et Tuncbilek [185]; la linéarisation décrite ci-dessus sera noté $[\cdot]_\ell$. Typiquement, $[(a_p - A_{p\cdot}x)(a_q - A_{q\cdot}x)]_\ell$ (où $A_{p\cdot}$ représente la p^e ligne de la matrice A) désigne la linéarisation du produit

$$(a_p - A_{p\cdot}x)(a_q - A_{q\cdot}x) \;=\; \sum_{i\in N}\sum_{j\in N} A_{pi}A_{qj}x_i x_j - \sum_{i\in N}(a_p A_{qi} + a_q A_{pi})x_i + a_p a_q,$$

et s'écrit explicitement en introduisant les variables de linéarisation v et w

$$\sum_{(i,j)\in M}(A_{pi}A_{qj} + A_{pj}A_{qi})w_{ij} + \sum_{i\in N} A_{pi}A_{qi}v_i - \sum_{i\in N}(a_p A_{qi} + a_q A_{pi})x_i + a_p a_q.$$

Étant donné que la définition du polyèdre X comprend les contraintes $A_{p\cdot}x \leq a_p$ et $A_{q\cdot}x \leq a_q$, on peut obtenir une inégalité valide dans un espace de dimension supérieure en imposant à la linéarisation du produit d'être positive. Nous sommes

donc en mesure de formuler une première relaxation linéaire de QQP.

Proposition 5.1 *Soient P un sous-ensemble d'indices de $\{1, 2, \ldots, m\}$, et Q_p un sous-ensemble de $\{p, p+1, \ldots, m\}$ pour chaque p de P. Le problème*

$$[QQP]_\ell \quad \begin{array}{lll} \min\limits_{x \in X, v, w} & [Q^0(x)]_\ell & \\ s.c. & [Q^k(x)]_\ell \overline{\overline{\leq}} b_k & k = 1, 2, \ldots, \overline{k}, \\ & [(a_p - A_{p.}x)(a_q - A_{q.}x)]_\ell \geq 0 & p \in P,\ q \in Q_p \end{array}$$

est une relaxation linéaire de QQP.

Démonstration Soit x^* une solution optimale de QQP. Posons $v_i^* = x_i^{*2}$ pour chaque i appartenant à N, et $w_{ij}^* = x_i^* x_j^*$ pour chaque (i, j) appartenant à M.

Pour chaque indice k de K, la valeur $[Q^k(x^*)]_\ell = \sum_{(i,j) \in M} C_{ij}^k w_{ij}^* + \sum_{i \in N} c_i^k v_i^* + \sum_{i \in N} d_i^k x_i^*$, est identique à $Q^k(x^*)$. De plus, en remplaçant les variables v_i^* et w_{ij}^* par les produits respectifs x_i^{*2} et $x_i^* x_j^*$, les contraintes de linéarisation deviennent $(a_p - A_{p.}x^*)(a_q - A_{q.}x^*) \geq 0$ pour $p \in P$ et $q \in Q_p$. Le point (x^*, v^*, w^*) est donc réalisable pour le problème $[QQP]_\ell$ et sa valeur objectif est égale à $Q^0(x^*)$, d'où le résultat. ∎

Dans le cas où toutes les linéarisations possibles sont présentes dans la relaxation $[QQP]_\ell$, lorsque $P = \{1, 2, \ldots, m\}$, et $Q_p = \{p, p+1, \ldots, m\}$ pour chaque p de P, Sherali et Tuncbilek [186] montrent que si au moins l'une des variables x_i est bornée inférieurement et supérieurement dans X, alors la contrainte $x \in X$ est redondante dans cette linéarisation. Elle peut donc être retirée de $[QQP]_\ell$.

Cette technique de linéarisation offre un cadre général pour la définition d'une méthode d'approximation extérieure consistant essentiellement en une suite de raffinements d'approximations de fonctions quadratiques. Nous présentons une méthode divisée en deux étapes principales, et en discutons dans les sous-sections qui suivent. La première étape sert à l'obtention de bornes de qualité sur les variables. Ces bornes ainsi que les contraintes définissant X permettent d'obtenir une relaxation initiale

de QQP. La seconde étape résout et raffine une série de relaxations. Cette étape est réitérée et prend fin lorsqu'une précision requise est atteinte. Nous présentons quatre classes d'inégalités valides permettant l'élimination du point obtenu lors de la résolution de la relaxation courante de QQP. Pour chacune de ces classes, nous choisissons l'inégalité qui minimise l'erreur d'approximation potentielle.

5.2.1 Évaluation de bornes

La nature non convexe des contraintes de QQP rend non triviale l'obtention de bornes de qualité sur les variables. Soient x^-, x^+ des bornes sur x_i telles que $0 \leq x^- \leq x_i \leq x^+$, et v^-, v^+ des bornes sur v_i telles que $0 \leq v^- \leq v_i \leq v^+$ obtenues en remplaçant successivement la fonction objectif de $[QQP]_\ell$ par les quatres fonctions $\pm x_i$ et $\pm v_i$. Il s'ensuit que les valeurs résultant de l'intersection des intervalles $[x^-, x^+]$ et $[\sqrt{v^-}, \sqrt{v^+}]$ définissent des bornes valides sur x_i. Définissons $\ell_i \equiv \max\{x^-, \sqrt{v^-}\}$ et $u_i \equiv \min\{x^+, \sqrt{v^+}\}$. Si $\ell_i \leq u_i$, alors ℓ_i et u_i sont des bornes valides pour la variable x_i du domaine réalisable de QQP. Si $\ell_i > u_i$, alors le domaine réalisable de QQP est vide.

Toutefois, il est possible que ces bornes puissent être améliorées dans le cas où $\ell_i > \min\{x_i : x \in X\}$ ou bien si $u_i < \max\{x_i : x \in X\}$. En ajoutant ces dernières bornes au polyèdre X, on peut réitérer ce processus afin d'en obtenir d'autres plus serrées. Considérons l'exemple suivant.

Exemple 5.2 *Soit le problème quadratique dont le domaine réalisable est défini par les deux contraintes*

$$x_1 + x_1^2 \leq 6 \quad \textit{et} \quad x_1 \geq 1.$$

On peut aisément vérifier que l'intervalle réalisable est $[1, 2]$. *L'approche décrite précédemment donne les contraintes de la relaxation* $[QQP]_\ell$ *:*

$$x_1 + v_1 \leq 6,\ x_1 \geq 1 \ \textit{ et } \ [(x_1 - 1)^2]_\ell = v_1 - 2x_1 + 1 \geq 0.$$

Les premières bornes obtenues sont $x_1 \in [1, \frac{7}{3}]$ *et* $v_1 \in [1, 5]$, *d'où les secondes* $x_1 \in [1, \sqrt{5}]$. *Puisque* $X = \{x : x_1 \geq 1\}$ *n'est pas borné supérieurement, il est possible*

que la nouvelle borne supérieure $\sqrt{5}$ puisse être améliorée. En ajoutant la contrainte $x_1 \leq \sqrt{5}$ à X, on obtient les nouvelles linéarisations

$$\begin{aligned} [(\sqrt{5}-x_1)^2]_\ell &= v_1 - 2\sqrt{5}x_1 + 5 \geq 0 \quad et \\ [(\sqrt{5}-x_1)(x_1-1)]_\ell &= -v_1 + (1+\sqrt{5})x_1 - \sqrt{5} \geq 0. \end{aligned}$$

À leur tour, ces contraintes engendrent de nouvelles bornes : $x_1 \in [1, \frac{11}{2\sqrt{5}+1}]$ et $v_1 \in [1, 13-4\sqrt{5}]$. Puisque $\frac{11}{2\sqrt{5}+1} \approx 2{,}010 < \sqrt{13-4\sqrt{5}} \approx 2{,}014 < \sqrt{5} \approx 2{,}236$, la contrainte $x_1 \leq \sqrt{5}$ peut alors être remplacée par la meilleure borne $x_1 \leq \frac{11}{2\sqrt{5}+1}$. La répétition de ce procédé donne une intervalle qui converge vers l'intervalle réalisable de x_1, soit vers $[1, 2]$. ∎

Toute inégalité valide appartenant aux deux premières classes de linéarisations présentées à la section suivante, peut être ajoutée lors de l'évaluation de bornes. Il est alors possible que de meilleures bornes soient générées. À la section 5.3.1, nous présentons un algorithme dont la phase de pré-traitement consiste à itérer cette méthode d'évaluation de bornes jusqu'à ce que l'amélioration de celles-ci soit négligeable.

5.2.2 Classes d'inégalités valides

Considérons une relaxation linéaire de QQP ainsi qu'une solution optimale $(\hat{x}, \hat{v}, \hat{w})$ de celle-ci pour laquelle la linéarisation est inexacte. C'est-à-dire qu'il existe un indice i de N pour lequel $\hat{x}_i^2 \neq \hat{v}_i$, ou une paire d'indices (i, j) de M pour laquelle $\hat{x}_i\hat{x}_j \neq \hat{w}_{ij}$. Nous présentons quatre classes d'inégalités valides permettant d'éliminer la solution optimale de la relaxation sans toutefois éliminer de solutions réalisables de QQP. L'ajout de membres de ces classes raffine l'approximation extérieure des termes quadratiques. La solution $(\hat{x}, \hat{v}, \hat{w})$ est appelée le *point courant.*

Classe I : Sous-estimation de la fonction carrée

La première classe de linéarisations est la plus intuitive et la plus simple à décrire. Considérons la fonction d'une seule variable $f(x_i) = x_i^2$ sur l'intervalle $[\ell_i, u_i]$. Supposons que le point courant soit tel qu'il existe un $i \in N$ tel que $\hat{v}_i < \hat{x}_i^2$. Nous montrons que toute droite tangente à f évaluée à un point α de l'intervalle $]\hat{x}_i - \sqrt{\hat{x}_i^2 - \hat{v}_i}, \hat{x}_i + \sqrt{\hat{x}_i^2 - \hat{v}_i}[$ définit une inégalité valide éliminant le point courant. La figure 5.1 illustre cette construction.

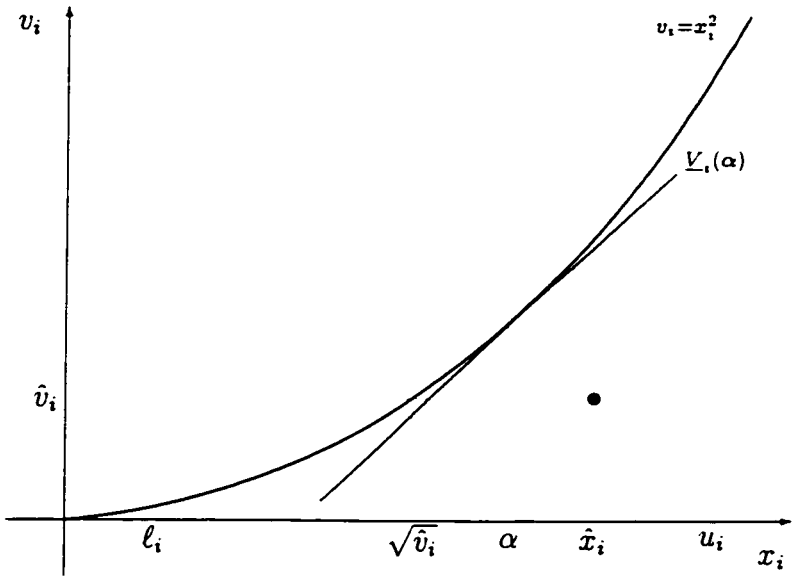


Figure 5.1 – *Sous-estimation de la fonction* $f(x_i) = x_i^2$

Proposition 5.3 *Soient* $\ell_i \leq \sqrt{\hat{v}_i} < \hat{x}_i \leq u_i$. *L'inégalité*

$$\underline{V}_i(\alpha) : \qquad 2\alpha x_i - v_i \leq \alpha^2$$

où $\alpha \in]\hat{x}_i - \sqrt{\hat{x}_i^2 - \hat{v}_i}, \hat{x}_i + \sqrt{\hat{x}_i^2 - \hat{v}_i}[$, *élimine le point* $(\hat{x}_i, \hat{v}_i)$, *mais n'élimine aucun point* (x_i, v_i) *satisfaisant* $v_i = f(x_i)$.

Démonstration Le point $(\hat{x}_i, \hat{v}_i)$ est éliminé puisque les inégalités

$$\hat{x}_i - \sqrt{\hat{x}_i^2 - \hat{v}_i} < \alpha < \hat{x}_i + \sqrt{\hat{x}_i^2 - \hat{v}_i}$$

impliquent que $(\hat{x}_i - \alpha)^2 < \hat{x}_i^2 - \hat{v}_i$ et donc $2\alpha\hat{x}_i - \hat{v}_i > \alpha^2$.

Le développement de Taylor de la fonction f autour du point α montre que pour tout point (x_i, v_i) satisfaisant $v_i = f(x_i)$

$$\begin{aligned} f(x_i) &= f(\alpha) + f'(\alpha)(x_i - \alpha) + \frac{1}{2}f''(\alpha)(x_i - \alpha)^2 \\ &= \alpha^2 + 2\alpha(x_i - \alpha) + (x_i - \alpha)^2 \ \geq \ 2\alpha x_i - \alpha^2, \end{aligned}$$

d'où le résultat. ∎

Définissons la première classe d'inégalités valides pour toutes les valeurs de α.

Définition 5.4 *Soient $\hat{x}_i \in [\ell_i, u_i]$ et $\hat{v}_i \in [\ell_i^2, \hat{x}_i^2[$. La classe de sous-estimations de v_i est*

$$C_I = \{\underline{V}_i(\alpha) : \alpha \in]\hat{x}_i - \sqrt{\hat{x}_i^2 - \hat{v}_i}, \hat{x}_i + \sqrt{\hat{x}_i^2 - \hat{v}_i}[\}.$$

$\underline{V}_i(\alpha)$ est appelée la linéarisation de f au point α. ∎

Les membres de de cette classe peuvent aussi être obtenus par les linéarisations d'Al-Khayyal et Falk [9]. En effet, l'inégalité $[(x_i - \alpha)^2]_\ell \geq 0$ s'écrit $v_i - 2\alpha x_i + \alpha^2 \geq 0$.

Classe II : Sous-estimation d'une paraboloïde

La seconde classe de linéarisations généralise la première. Les linéarisations sont obtenues par une sous-approximation linéaire d'une paraboloïde. Considérons la fonction de paramètre γ

$$\begin{aligned} h_\gamma : \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (x_i, x_j) &\mapsto h_\gamma(x_i, x_j) = (x_i + \gamma x_j)^2, \end{aligned}$$

où γ est un nombre réel, défini ci-dessous. Tout plan tangent à la fonction convexe h_γ peut servir de sous-estimation valide.

Proposition 5.5 *Soient $\alpha_i \in [\ell_i, u_i], \alpha_j \in [\ell_j, u_j]$ et $\gamma \in \mathbb{R}$. L'inégalité*

$$\underline{P}_{ij}^{\gamma}(\alpha_i, \alpha_j) : \quad 2(\alpha_i + \gamma\alpha_j)x_i + 2(\alpha_i + \gamma\alpha_j)\gamma x_j - v_i - \gamma^2 v_j - 2\gamma w_{ij} \leq (\alpha_i + \gamma\alpha_j)^2$$

n'élimine aucun point $(x_i, x_j, v_i, v_j, w_{ij})$ *satisfaisant* $v_i + 2\gamma w_{ij} + \gamma^2 v_j = h_\gamma(x_i, x_j)$.

Démonstration Le développement de Taylor de h_γ autour du point (α_i, α_j) donne l'approximation

$$h_\gamma(x_i, x_j) \geq h_\gamma(\alpha_i, \alpha_j) + 2(\alpha_i + \gamma\alpha_j)(1, \gamma) \cdot (x_i - \alpha_i, x_j - \alpha_j)^t.$$

L'inégalité provient du fait que le hessien de la fonction h_γ est une matrice semi-définie positive. En remplaçant le membre de gauche par sa linéarisation $[h_\gamma(x_i, x_j)]_\ell$, et en développant le membre de droit on obtient la coupe

$$v_i + 2\gamma w_{ij} + \gamma^2 v_j \geq -(\alpha_i + \gamma\alpha_j)^2 + 2(\alpha_i + \gamma\alpha_j)(x_i + \gamma x_j),$$

qui donne le résultat désiré. ∎

L'inégalité $\underline{P}^\gamma_{ij}(\alpha_i, \alpha_j)$ dépend des trois paramètres α_i, α_j et γ. Ces deux premiers définissent le point où le plan tangent est évalué (ils sont déterminés à la section 5.2.3). Nous choisissons la valeur de γ de façon à maximiser la profondeur de cette coupe. Réécrivons l'inégalité valide de façon à faire ressortir la paramétrisation en la variable γ :

$$(v_j - 2\alpha_j x_j + \alpha_j^2)\gamma^2 + 2(w_{ij} - \alpha_i x_j - \alpha_j x_i + \alpha_i\alpha_j)\gamma + (v_i - 2\alpha_i x_i - \alpha_i^2) \geq 0. \quad (5.1)$$

Dans le cas où cette fonction quadratique est convexe, nous choisissons le γ correspondant au minimum. Le cas où la fonction est concave n'est pas développé car il se réduit à l'utilisation d'un γ très grand, ce qui équivaut aux linéarisations de la classe C_I, présentées ci-dessus.

Proposition 5.6 *Soient* $\ell_i \leq \hat{x}_i < \alpha_i < \sqrt{\hat{v}_i} \leq u_i$ *et* $\ell_j \leq \hat{x}_j < \alpha_j < \sqrt{\hat{v}_j} \leq u_j$. *Posons*

$$\tau_i \equiv \hat{v}_i - 2\alpha_i\hat{x}_i + \alpha_i^2, \quad \tau_j \equiv \hat{v}_j - 2\alpha_j\hat{x}_j + \alpha_j^2, \quad \pi_{ij} \equiv \hat{w}_{ij} - \alpha_i\hat{x}_j - \alpha_j\hat{x}_i + \alpha_i\alpha_j.$$

Si $\tau_i\tau_j < \pi_{ij}^2$, *alors la coupe* $\underline{P}^\gamma_{ij}(\alpha_i, \alpha_j)$ *où* $\gamma = -\frac{\pi_{ij}}{\tau_j}$ *élimine le point* $(\hat{x}_i, \hat{x}_i, \hat{v}_i, \hat{v}_j, \hat{w}_{ij})$.

Démonstration Au point courant, la valeur de la fonction quadratique (5.1) s'écrit $\tau_j\gamma^2 + 2\pi_{ij}\gamma + \tau_i \geq 0$. Elle est convexe puisque $\tau_j > \hat{x}_j^2 - 2\alpha_j\hat{x}_j + \alpha_j^2 \geq 0$, et son minimum est atteint lorsque $\gamma = -\frac{\pi_{ij}}{\tau_j}$. Elle s'écrit alors

$$\frac{\pi_{ij}^2}{\tau_j} - \frac{2\pi_{ij}^2}{\tau_j} + \tau_i \geq 0 \qquad \text{ou bien} \qquad \pi_{ij}^2 \leq \tau_i\tau_j.$$

Puisque $\tau_i\tau_j < \pi_{ij}^2$, cette coupe élimine le point courant. ∎

La deuxième classe d'inégalités valides consiste à considérer toutes les valeurs de α_i et α_j.

Définition 5.7 *Soient $\ell_i \leq \hat{x}_i < \sqrt{\hat{v}_i} \leq u_i$ et $\ell_j \leq \hat{x}_j < \sqrt{\hat{v}_j} \leq u_j$. La classe de sous-estimations de la paraboloïde est*

$$\mathcal{C}_{II} = \{\underline{P}_{ij}^{\gamma}(\alpha_i, \alpha_j) : \alpha_i \in]\hat{x}_i, \sqrt{\hat{v}_i}[, \alpha_j \in]\hat{x}_j, \sqrt{\hat{v}_j}[, \gamma = -\frac{\hat{w}_{ij} - \alpha_i\hat{x}_j - \alpha_j\hat{x}_i + \alpha_i\alpha_j}{\hat{v}_j - 2\alpha_j\hat{x}_j + \alpha_j^2}\}.$$

$\underline{P}_{ij}^{\gamma}(\alpha_i, \alpha_j)$ est appelée la linéarisation de la paraboloïde h_γ au point (α_i, α_j). ∎

Les membres de cette classe peuvent être obtenus en bornant inférieurement la linéarisation $[h_\gamma(x_i, x_j)]_\ell$ par le plan tangent à h_γ au point (α_i, α_j).

Classe III : Sur-estimation de la fonction carrée

La convexité stricte de la fonction $f(x_i) = x_i^2$ ne permet pas une surapproximation linéaire serrée. Nous développons dans cette section une surapproximation linéaire par morceaux. Définissons une première inégalité valide, notée $\overline{V}_i(\ell_i, u_i)$, par l'intermédiaire de la corde reliant les points (ℓ_i, ℓ_i^2) et (u_i, u_i^2) :

$$(\ell_i + u_i)x_i - v_i \geq \ell_i u_i.$$

Cette linéarisation est obtenue par les techniques d'Al-Khayyal et Falk [9] :

$$0 \leq [(x_i - \ell_i)(u_i - x_i)]_\ell = -v_i + (\ell_i + u_i)x_i - \ell_i u_i.$$

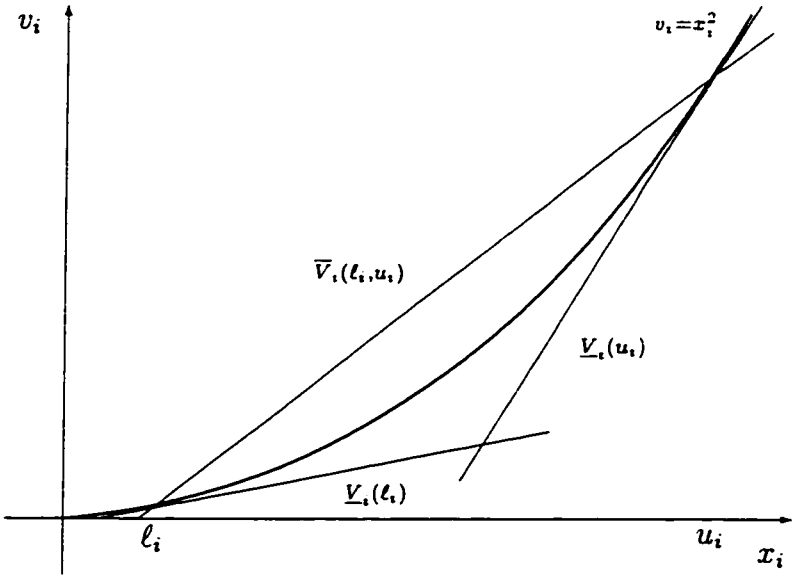


Figure 5.2 – *Linéarisation de la fonction* $f(x_i) = x_i^2$

Cette surapproximation ainsi que les tangentes $\underline{V}_i(\ell_i)$ et $\underline{V}_i(u_i)$ de la classe C_l sont illustrées à la figure 5.2. Les points réalisables appartiennent à l'intérieur du triangle apparaissant à la même figure.

La dichotomie est nécessaire au raffinement d'une surapproximation de f. Supposons que le point courant soit tel qu'il existe un indice $i \in N$ tel que $\hat{v}_i > \hat{x}_i^2$. Pour $\alpha \in [\ell_i, u_i]$, définissons la variable $\delta_i(\alpha) \in [0,1]$. Cette variable sert à la construction d'une inégalité valide. Elle est définie de façon à ce que si $\delta_i(\alpha) = 0$ alors la variable x_i est forcée à être inférieure à α, et si $\delta_i(\alpha) = 1$ alors x_i est forcée à être supérieure à α. L'introduction de ces variables contribuera à l'élaboration d'un algorithme d'énumération implicite.

Proposition 5.8 *Soient* $\ell_i \leq \alpha < \beta \leq u_i$. *Considérons les linéarisations* $\underline{V}_i(\alpha)$, $\underline{V}_i(\beta)$, *et l'inégalité*

$$\overline{V}_i(\alpha,\beta): \qquad (\alpha+\beta)x_i - v_i \geq \alpha\beta + (u_i-\ell_i)^2(\delta_i(\alpha) - \delta_i(\beta) - 1),$$

où $\delta_i(\alpha) \geq \delta_i(\beta)$. *Si* $\hat{v}_i = \hat{x}_i^2$, *alors il existe* $\hat{\delta}_i(\alpha)$ *et* $\hat{\delta}_i(\beta)$ *dans* $\{0,1\}$ *satisfaisant ces inégalités.*

Démonstration Considérons le cas où $\hat{\delta}_i(\alpha) = 1$ et $\hat{\delta}_i(\beta) = 0$. La linéarisation $\overline{V}_i(\alpha, \beta)$ est alors la corde reliant les points (α, α^2) et (β, β^2) et s'écrit $(\alpha+\beta)x_i - v_i \geq \alpha\beta$. Elle est valide sur l'intervalle $[\alpha, \beta]$ et découle de

$$0 \leq [(x_i - \alpha)(\beta - x_i)]_\ell = -v_i + (\alpha + \beta)x_i - \alpha\beta.$$

L'inégalité $\underline{V}_i(\alpha) : 2\alpha x_i - v_i \leq \alpha^2$ assure que

$$\alpha^2 - \alpha x_i \geq \alpha x_i - v_i \geq \alpha\beta - \beta x_i,$$

d'où $x_i(\beta - \alpha) \geq \alpha(\beta - \alpha)$. De manière similaire, l'inégalité $\underline{V}_i(\beta) : 2\beta x_i - v_i \leq \beta^2$ assure que

$$\alpha\beta - \alpha x_i \leq \beta x_i - v_i \leq \beta^2 - \beta x_i,$$

d'où $x_i(\beta - \alpha) \leq \beta(\beta - \alpha)$. Il s'ensuit que $x_i \in [\alpha, \beta]$.

Dans le cas où $\hat{\delta}_i(\alpha) = \hat{\delta}_i(\beta) = 0$ ou bien $\hat{\delta}_i(\alpha) = \hat{\delta}_i(\beta) = 1$, l'inégalité $\overline{V}_i(\alpha, \beta)$ est valide sur tout l'intervalle $[\ell_i, u_i]$ puisqu'elle est dominée par $\overline{V}_i(\ell_i, u_i)$. ∎

Supposons maintenant que nous voulons éliminer le point courant satisfaisant $\hat{v}_i > \hat{x}_i^2$ et que ce point n'est pas éliminé par la linéarisation $\overline{V}_i(\underline{\beta}, \overline{\beta})$ où $\ell_i \leq \underline{\beta} \leq \hat{x}_i \leq \sqrt{\hat{v}_i} \leq \overline{\beta} \leq u_i$.

Proposition 5.9 *Les linéarisations $\overline{V}_i(\underline{\beta}, \alpha)$ et $\overline{V}_i(\alpha, \overline{\beta})$ où $\alpha \in [\hat{x}_i, \sqrt{\hat{v}_i}]$ et où $\delta_i(\alpha)$ appartient à $\{0, 1\}$, éliminent le point courant.*

Démonstration Si $\delta_i(\alpha) = 0$, alors $\overline{V}_i(\underline{\beta}, \alpha)$ s'écrit

$$(\underline{\beta} + \alpha)x_i - v_i \;\geq\; \underline{\beta}\alpha + (u_i - \ell_i)^2(\delta_i(\underline{\beta}) - 1) \;=\; \underline{\beta}\alpha.$$

Cette inégalité élimine le point courant car $(\underline{\beta} + \alpha)\hat{x}_i - \hat{v}_i < (\underline{\beta} + \alpha)\alpha - \alpha^2$.

Si $\delta_i(\alpha) = 1$, alors $\overline{V}_i(\alpha, \overline{\beta})$ s'écrit

$$(\alpha + \overline{\beta})x_i - v_i \;\geq\; \alpha\overline{\beta} - (u_i - \ell_i)^2\delta_i(\overline{\beta}) \;=\; \alpha\overline{\beta}.$$

Cette inégalité élimine le point courant car $(\alpha + \overline{\beta})\hat{x}_i - \hat{v}_i < (\alpha + \overline{\beta})\alpha - \alpha^2$. ∎

Suite à ces propositions, nous sommes en mesure de définir la troisième classe d'inégalités valides pour l'ensemble des valeurs de α.

Définition 5.10 *Soient* $\ell_i \leq \underline{\beta} \leq \hat{x}_i \leq \sqrt{\hat{v}_i} \leq \overline{\beta} \leq u_i$. *La classe de surestimations de* v_i *est*

$$C_{III} = \{\overline{V}_i(\underline{\beta}, \alpha), \overline{V}_i(\alpha, \overline{\beta}), \underline{V}_i(\alpha) : \alpha \in]\hat{x}_i, \sqrt{\hat{v}_i}[\}.$$

$\overline{V}_i(\alpha, \beta)$ *est appelée la linéarisation de* f *au point* (α, β). ∎

Le rôle des variables $\delta_i(\cdot)$ conjointement avec les linéarisations $\underline{V}_i(\cdot)$ et $\overline{V}_i(\cdot)$, est de fixer le sous-intervalle de recherche. La linéarisation $\overline{V}_i(\ell_i, u_i)$ présentée au début de cette section s'inscrit dans le cadre de la définition 5.10 lorsque l'on fixe $\delta_i(\ell_i) = 1$ et $\delta_i(u_i) = 0$. Ceci assure que x_i appartient à l'intervalle $[\ell_i, u_i]$.

Classe IV : Estimation d'un produit de deux variables

Considérons maintenant l'approximation tangente de la fonction $g(x_i, x_j) = x_i x_j$ autour du point (α_i, α_j). Le plan tangent Π satisfait les trois propriétés suivantes :

i- (x_i, x_j, w_{ij}) appartient à Π si et seulement si $x_i = \alpha_i$ ou $x_j = \alpha_j$.

ii- Π sous-estime strictement (x_i, x_j, w_{ij}) si et seulement si $x_i < \alpha_i$ et $x_j < \alpha_j$, ou bien si $x_i > \alpha_i$ et $x_j > \alpha_j$.

iii- Π surestime strictement (x_i, x_j, w_{ij}) si et seulement si $x_i < \alpha_i$ et $x_j > \alpha_j$, ou bien si $x_i > \alpha_i$ et $x_j < \alpha_j$.

Définissons les quatre quadrants associés au point (α_i, α_j) :

$$\begin{aligned} \Omega^I &= \{(x_i, x_j) : x_i \geq \alpha_i, x_j \geq \alpha_j\}, & \Omega^{II} &= \{(x_i, x_j) : x_i \leq \alpha_i, x_j \geq \alpha_j\}, \\ \Omega^{III} &= \{(x_i, x_j) : x_i \leq \alpha_i, x_j \leq \alpha_j\}, & \Omega^{IV} &= \{(x_i, x_j) : x_i \geq \alpha_i, x_j \leq \alpha_j\}. \end{aligned}$$

Nous pouvons aisément obtenir des inégalités valides. La classe de linéarisations résultante permet l'obtention de l'enveloppe convexe et concave de la fonction g. Rappelons que l'enveloppe convexe (concave) d'une fonction est le suprémum (infémum)

point par point de toutes les sous-estimations (sur) linéaires de cette fonction.

Proposition 5.11 *Soient* $\alpha_i \in [\ell_i, u_i]$ *et* $\alpha_j \in [\ell_j, u_j]$. *Posons* $L_i = \alpha_i - \ell_i, L_j = \alpha_j - \ell_j$, *et* $U_i = u_i - \alpha_i, U_j = u_j - \alpha_j$. *Les inégalités*

$$\begin{aligned}
\underline{W}^{I}_{ij}(\alpha_i, \alpha_j): &\quad \alpha_j x_i + \alpha_i x_j - w_{ij} \leq \alpha_i\alpha_j + L_iU_j(1-\delta_i(\alpha_i)) + U_iL_j(1-\delta_j(\alpha_j)) \\
\underline{W}^{III}_{ij}(\alpha_i, \alpha_j): &\quad \alpha_j x_i + \alpha_i x_j - w_{ij} \leq \alpha_i\alpha_j + U_iL_j\delta_i(\alpha_i) + L_iU_j\delta_j(\alpha_j)
\end{aligned}$$

définissent respectivement l'enveloppe convexe de g *sur les quadrants* Ω^I *et* Ω^{III}. *Les inégalités*

$$\begin{aligned}
\overline{W}^{II}_{ij}(\alpha_i, \alpha_j): &\quad \alpha_j x_i + \alpha_i x_j - w_{ij} \geq \alpha_i\alpha_j - U_iU_j\delta_i(\alpha_i) - L_iL_j(1-\delta_j(\alpha_j)) \\
\overline{W}^{IV}_{ij}(\alpha_i, \alpha_j): &\quad \alpha_j x_i + \alpha_i x_j - w_{ij} \geq \alpha_i\alpha_j - L_iL_j(1-\delta_i(\alpha_i)) - U_iU_j\delta_j(\alpha_j)
\end{aligned}$$

définissent respectivement l'enveloppe concave de g *sur les quadrants* Ω^{II} *et* Ω^{IV}. *Et donc, ces inégalités n'éliminent aucun point* (x_i, x_j, w_{ij}) *satisfaisant* $w_{ij} = g(x_i, x_j)$.

Démonstration Soit $(x_i, x_j) \in [\ell_i, u_i] \times [\ell_j, u_j]$. Si $\delta_i(\alpha_i) = \delta_j(\alpha_j) = 1$, alors ce point appartient à Ω^I et les linéarisations deviennent :

$$\begin{aligned}
\underline{W}^{I}_{ij}(\alpha_i, \alpha_j): &\quad [(x_i - \alpha_i)(x_j - \alpha_j)]_\ell \geq 0, \\
\overline{W}^{II}_{ij}(\alpha_i, \alpha_j): &\quad [(\alpha_i - x_i)(x_j - \alpha_j)]_\ell + (u_i - \alpha_i)(u_j - \alpha_j) \geq 0, \\
\underline{W}^{III}_{ij}(\alpha_i, \alpha_j): &\quad [(\alpha_i - x_i)(\alpha_j - x_j)]_\ell + (u_i - \alpha_i)(\alpha_j - \ell_j) + (\alpha_i - \ell_i)(u_j - \alpha_j) \geq 0, \\
\overline{W}^{IV}_{ij}(\alpha_i, \alpha_j): &\quad [(x_i - \alpha_i)(\alpha_j - x_j)]_\ell + (u_i - \alpha_i)(u_j - \alpha_j) \geq 0.
\end{aligned}$$

La contrainte $\underline{W}^{I}_{ij}(\alpha_i, \alpha_j)$ est l'enveloppe convexe de la fonction g sur Ω^I et domine $\underline{W}^{III}_{ij}(\alpha_i, \alpha_j)$ (cette dernière inégalité est donc valide). Les contraintes $\overline{W}^{II}_{ij}(\alpha_i, \alpha_j)$ et $\overline{W}^{IV}_{ij}(\alpha_i, \alpha_j)$ sont identiques et découlent des inégalités valides $u_i - \alpha_i \geq x_i - \alpha_i \geq 0$ et $u_j - \alpha_j \geq x_j - \alpha_j \geq 0$.

Si $\delta_i(\alpha_i) = 0$ et $\delta_j(\alpha_j) = 1$, alors (x_i, x_j) appartient à Ω^{II} et les linéarisations s'écrivent :

$$\begin{aligned}
\underline{W}^{I}_{ij}(\alpha_i, \alpha_j): &\quad [(x_i - \alpha_i)(x_j - \alpha_j)]_\ell + (\alpha_i - \ell_i)(u_j - \alpha_j) \geq 0, \\
\overline{W}^{II}_{ij}(\alpha_i, \alpha_j): &\quad [(\alpha_i - x_i)(x_j - \alpha_j)]_\ell \geq 0, \\
\underline{W}^{III}_{ij}(\alpha_i, \alpha_j): &\quad [(\alpha_i - x_i)(\alpha_j - x_j)]_\ell + (\alpha_i - \ell_i)(u_j - \alpha_j) \geq 0, \\
\overline{W}^{IV}_{ij}(\alpha_i, \alpha_j): &\quad [(x_i - \alpha_i)(\alpha_j - x_j)]_\ell + (\alpha_i - \ell_i)(\alpha_j - \ell_j) + (u_i - \alpha_i)(u_j - \alpha_j) \geq 0.
\end{aligned}$$

La contrainte $\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$ est l'enveloppe concave de la fonction g sur Ω^{II} et domine $\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$. Les contraintes $\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$ et $\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$ sont identiques et découlent des inégalités valides $\alpha_i - \ell_i \geq \alpha_i - x_i \geq 0$ et $u_j - \alpha_j \geq x_j - \alpha_j \geq 0$.

Si $\delta_i(\alpha_i) = \delta_j(\alpha_j) = 0$, alors (x_i, x_j) appartient à Ω^{III} et les linéarisations deviennent :

$\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$: $[(x_i - \alpha_i)(x_j - \alpha_j)]_\ell + (\alpha_i - \ell_i)(u_j - \alpha_j) + (u_i - \alpha_i)(\alpha_j - \ell_j) \geq 0$,
$\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$: $[(\alpha_i - x_i)(x_j - \alpha_j)]_\ell + (\alpha_i - \ell_i)(\alpha_j - \ell_j) \geq 0$,
$\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$: $[(\alpha_i - x_i)(\alpha_j - x_j)]_\ell \geq 0$,
$\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$: $[(x_i - \alpha_i)(\alpha_j - x_j)]_\ell + (\alpha_i - \ell_i)(\alpha_j - \ell_j) \geq 0$.

La contrainte $\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$ est l'enveloppe convexe de la fonction g sur Ω^{III} et domine $\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$. Les contraintes $\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$ et $\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$ sont identiques et découlent des inégalités valides $\alpha_i - \ell_i \geq \alpha_i - x_i \geq 0$ et $\alpha_j - \ell_j \geq \alpha_j - x_j \geq 0$.

Si $\delta_i(\alpha_i) = 1$ et $\delta_j(\alpha_j) = 0$, alors (x_i, x_j) appartient à Ω^{IV} et les linéarisations s'écrivent :

$\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$: $[(x_i - \alpha_i)(x_j - \alpha_j)]_\ell + (u_i - \alpha_i)(\alpha_j - \ell_j) \geq 0$,
$\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$: $[(\alpha_i - x_i)(x_j - \alpha_j)]_\ell + (u_i - \alpha_i)(u_j - \alpha_j) + (\alpha_i - \ell_i)(\alpha_j - \ell_j) \geq 0$,
$\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$: $[(\alpha_i - x_i)(\alpha_j - x_j)]_\ell + (u_i - \alpha_i)(\alpha_j - \ell_j) \geq 0$,
$\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$: $[(x_i - \alpha_i)(\alpha_j - x_j)]_\ell \geq 0$.

La contrainte $\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$ est l'enveloppe concave de la fonction g sur Ω^{IV} et domine $\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$. Les contraintes $\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$ et $\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$ sont identiques et découlent des inégalités valides $u_i - \alpha_i \geq x_i - \alpha_i \geq 0$ et $\alpha_j - \ell_j \geq \alpha_j - x_j \geq 0$.

Tout point satisfaisant $w_{ij} = g(x_i, x_j)$ appartient aux enveloppes convexe et concave de g, et donc n'est pas éliminé par ces inégalités. ∎

La proposition suivante montre que ces inégalités éliminent le point courant lorsque $\hat{w}_{ij} > g(\hat{x}_i, \hat{x}_j)$.

Proposition 5.12 *Soient* $\ell_i \leq \hat{x}_i < \alpha_i < \frac{\hat{w}_{ij}}{\hat{x}_j} \leq u_i$ *alors l'inégalité* $\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$ *élimine le point* $(\hat{x}_i, \hat{x}_j, \hat{w}_{ij})$.

Démonstration Le point $(\hat{x}_i, \hat{x}_j)$ se trouve à l'intérieur du quadrant Ω^{II} par rapport au point (α_i, α_j) et donc $\delta_i(\alpha_i) = 0$ et $\delta_j(\alpha_j) = 1$. La linéarisation $\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$ s'écrit alors $-w_{ij} + \alpha_j x_i + \alpha_i x_j - \alpha_i \alpha_j \geq 0$. Au point $(\hat{x}_i, \hat{x}_j, \hat{w}_{ij})$, les inégalités $\hat{x}_i < \alpha_i$ et $\alpha_i < \frac{\hat{w}_{ij}}{\hat{x}_j}$ impliquent que

$$-\hat{w}_{ij} + \alpha_j \hat{x}_i + \alpha_i \hat{x}_j - \alpha_i \alpha_j \;<\; -\hat{w}_{ij} + \alpha_j \alpha_i + \alpha_i \hat{x}_j - \alpha_i \alpha_j \;<\; -\hat{w}_{ij} + \frac{\hat{w}_{ij}}{\hat{x}_j} \hat{x}_j \;=\; 0.$$

Il s'ensuit que la linéarisation élimine le point courant. ∎

On peut montrer de façon symétrique que l'inégalité $\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$ élimine le point $(\hat{x}_i, \hat{x}_j, \hat{w}_{ij})$ lorsque $w_{ij} > \hat{x}_i \hat{x}_j$. De même, chacune des inégalités $\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$ et $\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$ élimine ce point lorsque $w_{ij} < \hat{x}_i \hat{x}_j$. Définissons les deux classes d'estimations de la fonction g.

Définition 5.13 *Soient $(\hat{x}_i, \hat{x}_j) \in [\ell_i, u_i] \times [\ell_j, u_j]$, et $\hat{w}_{ij} \neq \hat{x}_i \hat{x}_j$. Pour $\hat{w}_{ij} < \hat{x}_i \hat{x}_j$, la classe de sous-estimations de w_{ij} est*

$$\underline{\mathcal{C}}_{IV} \;=\; \{\, \underline{W}_{ij}^{I}(\alpha_i, \alpha_j),\ \underline{W}_{ij}^{III}(\alpha_i, \alpha_j) :\ (\alpha_i, \alpha_j) \in [\frac{\hat{w}_{ij}}{\hat{x}_j}, \hat{x}_i] \times [\ell_j, u_j] \,\}.$$

$\underline{W}_{ij}^{I}(\alpha_i, \alpha_j)$ et $\underline{W}_{ij}^{III}(\alpha_i, \alpha_j)$ sont respectivement les sous-estimations de g au point (α_i, α_j) dans les quadrants Ω^I, et Ω^{III}
Pour $w_{ij} > \hat{x}_i \hat{x}_j$, la classe de surestimations de w_{ij} est

$$\overline{\mathcal{C}}_{IV} \;=\; \{\, \overline{W}_{ij}^{II}(\alpha_i, \alpha_j),\ \overline{W}_{ij}^{IV}(\alpha_i, \alpha_j) :\ (\alpha_i, \alpha_j) \in [\hat{x}_i, \frac{\hat{w}_{ij}}{\hat{x}_j}] \times [\ell_j, u_j] \,\}.$$

$\overline{W}_{ij}^{II}(\alpha_i, \alpha_j)$ et $\overline{W}_{ij}^{IV}(\alpha_i, \alpha_j)$ sont respectivement les surestimations de g au point (α_i, α_j) dans les quadrants Ω^{II}, et Ω^{IV}. ∎

5.2.3 Choix des meilleures inégalités valides

Dans cette section, nous étudions la question de déterminer parmi les quatre classes d'inégalités valides présentées à la section 5.2.2, laquelle ou lesquelles devraient être ajoutées à la relaxation afin de raffiner le mieux possible l'approximation des termes quadratiques.

Classe I : Raffinement d'une sous-estimation de la fonction carré

Supposons que nous désirons éliminer le point courant satisfaisant $\hat{v}_i < \hat{x}_i^2$. La question consiste à déterminer la meilleure valeur de α dont la linéarisation $\underline{V}_i(\alpha)$ élimine ce point. Le point $(\hat{x}_i, \hat{v}_i)$ est issu de la résolution d'une relaxation de QQP. L'inexactitude de l'approximation de $\hat{x}_i^2$ par $\hat{v}_i$ peut être interprétée de deux façons : ou bien la valeur $\hat{v}_i$ est trop petite, ou bien celle de $\hat{x}_i$ est trop grande. Il n'est pas déraisonnable de croire que la valeur recherchée (celle où $v_i = x_i^2$) se trouve à l'intérieur de l'intervalle $[\sqrt{\hat{v}_i}, \hat{x}_i]$.

Nous proposons de choisir le point α qui minimise l'erreur pondérée maximale sur l'intervalle en question. L'erreur à minimiser est illustrée à la figure 5.3 : le maximum de e_1 et de αe_2. La pondération du deuxième terme d'erreur par α est motivée par le fait que e_1 représente une différence de carrés, tandis que ce n'est pas le cas pour e_2.

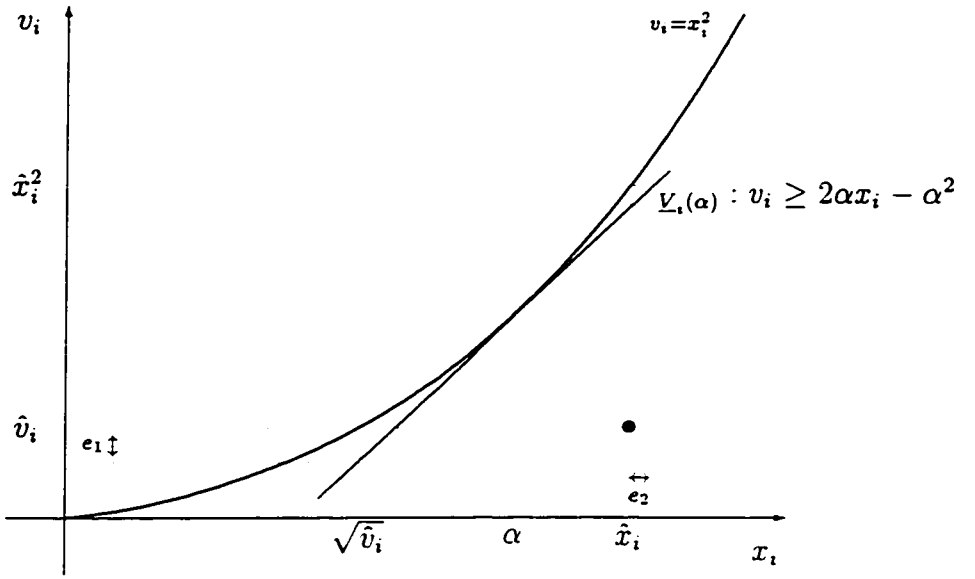


Figure 5.3 – *Minimisation de l'erreur d'une sous-estimation d'un carré*

Proposition 5.14 *La valeur* $\alpha = (\sqrt{2} - 1)\hat{x}_i + (2 - \sqrt{2})\sqrt{\hat{v}_i} \in [\sqrt{v_i}, x_i]$ *minimise le maximum entre* $e_1 = \hat{v}_i - (2\alpha\sqrt{\hat{v}_i} - \alpha^2)$ *et* αe_2 *où* $e_2 = (\frac{\hat{x}_i^2}{2\alpha} + \frac{\alpha}{2}) - \hat{x}_i$.

Démonstration La valeur minimale du maximum de e_1 et αe_2 est atteinte lorsque ces deux termes sont égaux. On peut aisément vérifier que $e_1 = \alpha e_2$ si et seulement si

$$\alpha^2 + 2(\hat{x}_i - 2\sqrt{\hat{v}_i})\alpha + 2\hat{v}_i - \hat{x}_i^2 = 0.$$

La seule racine de cette fonction quadratique appartenant à l'intervalle $[\sqrt{v_i}, x_i]$ est $\alpha = (2 - \sqrt{2})\sqrt{\hat{v}_i} + (\sqrt{2} - 1)\hat{x}_i$. ∎

Classe II : Raffinement d'une sous-estimation d'une paraboloïde

À la section 5.2.2, nous avons déterminé la valeur de γ générant la coupe la plus profonde pour la fonction $h_\gamma(x_i, x_j) = (x_i + \gamma x_j)^2$ au point (x_i, x_j). Il reste à déterminer ce point où le plan tangent sera évalué. Nous cherchons les valeurs de α_i et de α_j qui minimisent l'erreur potentielle maximale. La figure 5.4 illustre les erreurs minimisées à la proposition suivante.

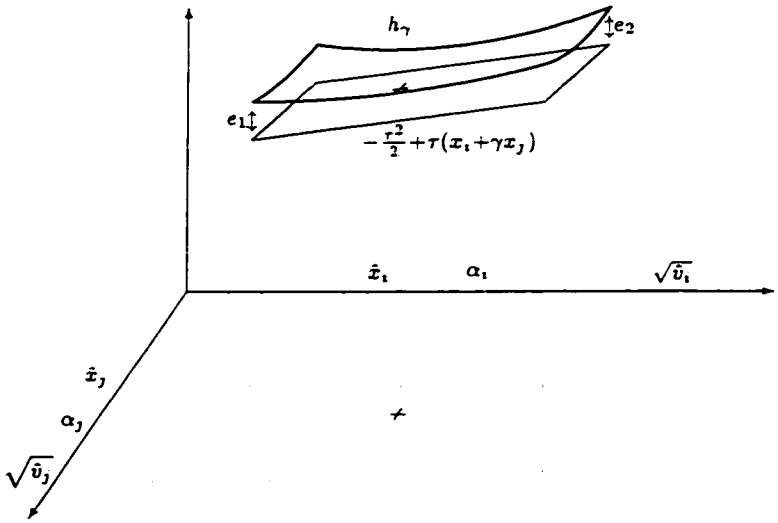


Figure 5.4 – *Minimisation de l'erreur associée à une paraboloïde*

Proposition 5.15 *La valeur* $(\alpha_i, \alpha_j) = (\hat{x}_i + \sqrt{\hat{v}_i}, \hat{x}_j + \sqrt{\hat{v}_j})/2 \in [x_i, \sqrt{v_i}] \times [x_j, \sqrt{v_j}]$ *minimise le maximum entre les distances entre la paraboloïde et le plan tangent au*

point $(\hat{x}_i, \sqrt{\hat{v}_j})$ *:* $e_1 = h_\gamma(\hat{x}_i, \sqrt{\hat{v}_j}) - \tau(\hat{x}_i + \gamma\sqrt{\hat{v}_j}) + \tau^2/2$ *et au point* $(\sqrt{\hat{v}_i}, \hat{x}_j)$ *:* $e_2 = h_\gamma(\sqrt{\hat{v}_i}, \hat{x}_j) - \tau(\sqrt{\hat{v}_i} + \gamma\hat{x}_j) + \tau^2/2$, *où* $\tau = 2(\alpha_i + \gamma\alpha_j)$.

Démonstration La valeur minimale du maximum entre e_1 et e_2 est atteinte lorsque ces deux termes sont égaux. On peut aisément vérifier que $e_1 = e_2$ si et seulement si

$$\tau = (\hat{x}_i + \sqrt{\hat{v}_i}) + \gamma(\hat{x}_j + \sqrt{\hat{v}_j}).$$

Puisque $\tau = 2(\alpha_i + \gamma\alpha_j)$, les valeurs $\alpha_i = (\hat{x}_i + \sqrt{\hat{v}_i})/2$ et $\alpha_j = (\hat{x}_j + \sqrt{\hat{v}_j})/2$ satisfont le critère. ∎

On peut montrer de façon similaire que ce même point minimise la distance maximale entre la paraboloïde et le plan tangent aux points $(\hat{x}_i, \hat{x}_j)$ et $(\sqrt{\hat{v}_i}, \sqrt{\hat{v}_j})$.

Classe III : Raffinement d'une surestimation de la fonction carré

Supposons maintenant que nous désirons éliminer une surapproximation de la fonction f, lorsque le point courant satisfait $\hat{v}_i > \hat{x}_i^2$. Nous discutons du choix de α qui minimise l'erreur pondérée maximale. Ce choix est plus délicat que celui de la sous-estimation car rappelons que la linéarisation en un point fait intervenir les valeurs des points d'évaluations inférieur et supérieur. Soient $\underline{\beta}$ et $\overline{\beta}$ des points d'évaluation tels que $[\hat{x}_i, \sqrt{\hat{v}_i}] \subset]\underline{\beta}, \overline{\beta}[$. On désire alors raffiner l'approximation sur ce premier intervalle. La figure 5.5 illustre les erreurs e_1 et e_2 associées au point de linéarisation α.

Proposition 5.16 *La valeur* $\alpha = \frac{-q+\sqrt{q^2+4p\overline{\beta}(\overline{\beta}-\hat{x}_i)\hat{x}_i}}{2p} \in [x_i, \sqrt{v_i}]$ *où* $p = (\hat{x}_i - \underline{\beta}) + (\overline{\beta} - \sqrt{\hat{v}_i})$ *et* $q = (\hat{x}_i - \underline{\beta})(\overline{\beta} - \hat{x}_i) - (\overline{\beta} - \sqrt{\hat{v}_i})\sqrt{\hat{v}_i}$, *minimise le maximum entre* $e_1 = (\underline{\beta} + \alpha)\hat{x}_i - \underline{\beta}\alpha - \hat{x}_i^2$ *et* αe_2 *où* $e_2 = \sqrt{\hat{v}_i} - \frac{\hat{v}_i + \alpha\overline{\beta}}{\alpha + \overline{\beta}}$.

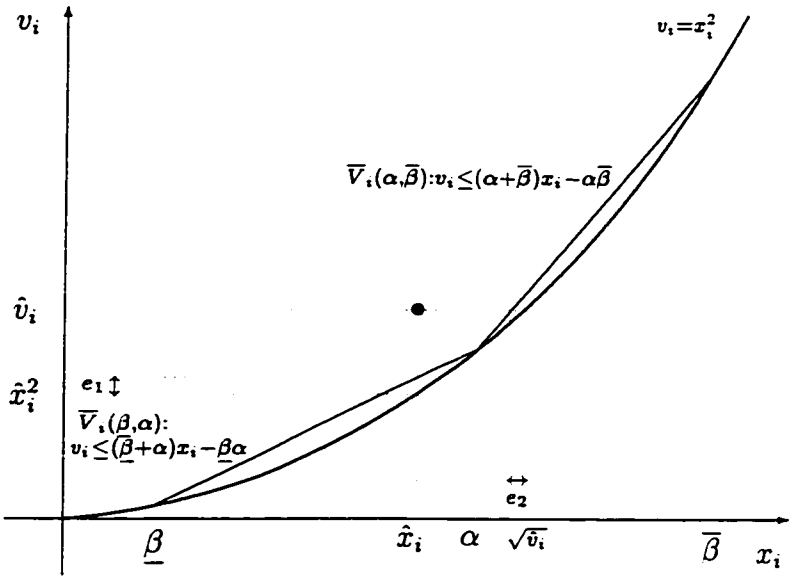


Figure 5.5 – *Minimisation de l'erreur d'une surestimation d'un carré*

Démonstration La valeur minimale du maximum de e_1 et αe_2 est atteinte lorsque ces deux termes sont égaux. On peut vérifier que $e_1 = \alpha e_2$ si et seulement si

$$(\underline{\beta}+\alpha)(\alpha+\overline{\beta})\hat{x}_i - \underline{\beta}\alpha(\alpha+\overline{\beta}) - \hat{x}_i^2(\alpha+\overline{\beta}) = \sqrt{\hat{v}_i}(\alpha+\overline{\beta})\alpha - (\hat{v}_i+\alpha\overline{\beta})\alpha$$

$$\Longleftrightarrow$$

$$(\hat{x}_i - \underline{\beta} + \overline{\beta} - \sqrt{\hat{v}_i})\alpha^2 + ((\hat{x}_i-\underline{\beta})(\overline{\beta}-\hat{x}_i) - (\overline{\beta}-\sqrt{\hat{v}_i})\sqrt{\hat{v}_i})\alpha - (\hat{x}_i-\underline{\beta})\overline{\beta}\hat{x}_i = 0$$

$$\Longleftrightarrow$$

$$p\alpha^2 + q\alpha - (\hat{x}_i-\underline{\beta})\overline{\beta}\hat{x}_i = 0.$$

Cette équation quadratique convexe possède nécessairement une racine dans l'intervalle $[\hat{x}_i, \sqrt{\hat{v}_i}]$. En effet, si $\alpha = \hat{x}_i$ alors $e_1 = 0$ et $\alpha e_2 > 0$. En augmentant α de façon continue jusqu'à $\sqrt{\hat{v}_i}$, e_1 devient positif et αe_2 décroît jusqu'à 0. Le théorème des valeurs intermédiaires assure qu'il existe une valeur α dans l'intervalle telle que $e_1 = \alpha e_2$.

La seconde racine est de valeur négative. En effet, si $\alpha = 0$ alors $e_1 < 0$ et $\alpha e_2 = 0$. Si α décroît, alors αe_2 décroît plus rapidement que e_1, et donc il existe un $\alpha < 0$ tel que $e_1 = \alpha e_2$.

La racine recherchée est celle qui est positive : celle de l'énoncé. ∎

Classe IV : Raffinement d'un produit de deux variables

Considérons maintenant le cas où l'on désire raffiner un produit de deux variables x_i et x_j. Nous ne traitons explicitement que le cas de la surapproximation puisque la sous-estimation se traite de façon similaire.

Soit $\hat{w}_{ij}$ une approximation de $\hat{x}_i\hat{x}_j$ telle que $\hat{w}_{ij} > \hat{x}_i\hat{x}_j$, et $\underline{\beta}$ et $\overline{\beta}$ les points d'évaluation tels que $[\hat{x}_j, \frac{\hat{w}_{ij}}{\hat{x}_i}] \subset]\underline{\beta}, \overline{\beta}[$. On cherche à déterminer un point d'évaluation $\alpha \in [\hat{x}_i, \frac{\hat{w}_{ij}}{\hat{x}_j}]$ pour la variable x_i, afin de raffiner l'approximation du produit x_ix_j lorsque $(x_i, x_j) \in [\hat{x}_i, \frac{\hat{w}_{ij}}{\hat{x}_j}] \times [\underline{\beta}, \overline{\beta}]$ pour éliminer la surestimation. Puisque dans l'algorithme présenté à la section 5.3.1 nous branchons sur une seule variable à la fois, nous supposons ici que la variable x_j est fixée à $\alpha_j = (\hat{x}_j + \sqrt{\hat{v}_j})/2$. Il est alors probable que la valeur recherchée de x_i se trouve dans l'intervalle $[\hat{x}_i, \frac{w_{ij}}{\hat{x}_j}[$. Nous cherchons la valeur α qui minimise l'erreur maximale pondérée. La figure 5.6 illustre ce cas.

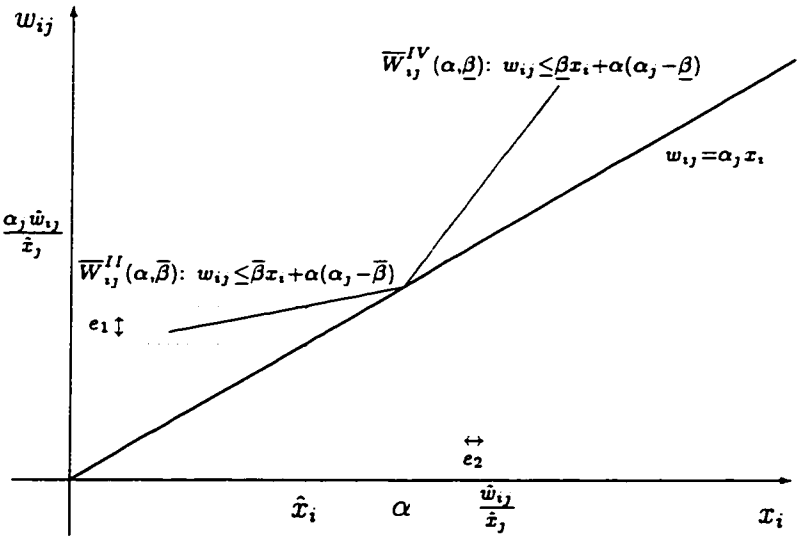


Figure 5.6 – *Élimination d'une surestimation d'un produit de variables*

Proposition 5.17 *La valeur*

$$\alpha = \frac{(1 - \frac{\overline{\beta}}{\alpha_j})\hat{x}_i + (1 - \frac{\alpha_j}{\underline{\beta}})\frac{\hat{w}_{ij}}{\hat{x}_j}}{(1 - \frac{\overline{\beta}}{\alpha_j}) + (1 - \frac{\alpha_j}{\underline{\beta}})}$$

minimise le maximum entre $e_1 = \overline{\beta}\hat{x}_i + \alpha(\alpha_j - \overline{\beta}) - \alpha_j\hat{x}_i$ *et* $\alpha_j e_2$, *où* $e_2 = \frac{\hat{w}_{ij}}{\hat{x}_j} - (\frac{\alpha_j\hat{w}_{ij}}{\hat{x}_j\underline{\beta}} - \frac{\alpha(\alpha_j - \underline{\beta})}{\underline{\beta}})$.

Démonstration La valeur minimale du maximum de e_1 et $\alpha_j e_2$ est atteinte lorsque ces deux termes sont égaux. On peut vérifier que $e_1 = \alpha_j e_2$ si et seulement si

$$\alpha(\alpha_j - \overline{\beta} - \frac{\alpha_j(\alpha_j - \underline{\beta})}{\underline{\beta}}) = -\overline{\beta}\hat{x}_i + \alpha_j(\hat{x}_i + \frac{\hat{w}_{ij}}{\hat{x}_j} - \frac{\alpha_j\hat{w}_{ij}}{\hat{x}_j\underline{\beta}})$$

Le résultat suit en isolant la variable α. ∎

Des résultats similaires peuvent être obtenus pour le cas de la sous-approximation d'un produit de deux variables.

5.3 Méthode de résolution

Nous sommes maintenant en mesure d'utiliser les outils développés aux sections précédentes afin de construire une méthode de résolution du problème QQP. Rappelons que la difficulté intrinsèque du problème fait en sorte qu'un algorithme exact et fini semble hors d'atteinte pour le moment. Nous définissons alors la notion d'une solution approchée à la fois dans sa réalisabilité et dans sa valeur.

Définition 5.18 *Une solution* $(\hat{x}, \hat{v}, \hat{w})$ *est dite* ϵ_r-ϵ_z-optimale, *(pour* $\epsilon_r > 0$*: la tolérance de réalisabilité, et* $\epsilon_z > 0$*: la précision de la valeur optimale), si elle est* ϵ_r-approchée, *c'est-à-dire si elle satisfait les trois conditions*

i- $\hat{x} \in X$,

ii- $|\hat{x}_i^2 - \hat{v}_i| < \epsilon_r$ *pour chaque* i *de* N,

iii- $|\hat{x}_i\hat{x}_j - \hat{w}_{ij}| < \epsilon_r$ *pour chaque* (i,j) *de* M,

et si $z^* - [Q^0(\hat{x},\hat{v},\hat{w})]_\ell < \epsilon_z$ *où* z^* *est la valeur optimale minimale de la linéarisation* $[Q^0]_\ell$ *sur toutes les solutions* ϵ_r*-approchées.* ∎

5.3.1 Algorithme pour le problème QQP

Au début de l'exécution de l'algorithme, le problème linéaire $[QQP]_\ell$ est créé et contient alors un certain nombre de linéarisations. Puis, des bornes sont évaluées pour chacune des variables. Ensuite, l'approximation des termes quadratiques est raffinée à l'aide des linéarisations présentées à la section 5.2.2, sans toutefois ajouter de variables δ_i.

La première phase de l'algorithme, le *pré-traitement*, consiste à itérer ces étapes afin d'obtenir de meilleures bornes. Ces itérations prennent fin lorsqu'aucune borne n'est améliorée par plus de ϵ_r.

La stratégie utilisée lors de la seconde phase de l'algorithme, l'*exploration de l'arbre d'énumération*, est de type meilleur d'abord. Les branchements se font sur la dichotomie $\delta_i(\cdot) = 0$ versus $\delta_i(\cdot) = 1$. Chaque noeud de l'arbre d'énumération (sauf la racine) comporte deux listes Δ^0 et Δ^1 qui contiennent les variables $\delta_i(\cdot)$ fixées respectivement à 0 et à 1. Les noeuds comportent aussi une borne supérieure de la valeur de la fonction objectif, ainsi que l'information sur la variable de branchement.

L'exploration de l'arbre d'énumération se résume succinctement à itérer les étapes suivantes jusqu'à ce que la précision exigée soit atteinte. On choisit et enlève de la liste, le noeud ayant le meilleur potentiel (celui ayant la plus petite borne inférieure), puis traiter (explicité ci-dessous) ses deux fils : celui où la variable de branchement $\delta_i(\cdot)$ est ajouté à Δ^0, ainsi que celui où $\delta_i(\cdot)$ est ajouté à Δ^1. Voici une description détaillée du traitement d'un noeud.

ALGORITHME *Al(QQP)* **–traitement d'un noeud.**

a. Initialisation.
Fixer les variables $\delta_i(\cdot)$ de Δ^0 à 0, et celles de Δ^1 à 1; les autres sont libres dans l'intervalle $[0, 1]$.

b. Test d'admissibilité.
Si le programme linéaire est non réalisable, alors le traitement du noeud est terminé. S'il est réalisable, évaluer une solution optimale $(\hat{x}, \hat{v}, \hat{w})$; poursuivre à l'étape **c**.

c. Raffinement d'une sous-estimation d'un carré (classe C_I).
Pour chaque $i \in N$ tel que $\hat{v}_i < \hat{x}_i\hat{x}_i - \epsilon_r$, ajouter les linéarisations $\underline{V}_i(\alpha)$ où α est défini à la proposition 5.14, puis poursuivre à l'étape **d**.

d. Raffinement d'une paraboloïde (classe C_{II}).
Pour chaque $(i, j) \in M$ et γ tels que les conditions de la proposition 5.6 sont satisfaites, ajouter les linéarisations $\underline{P}^{\gamma}_{ij}(\alpha_i, \alpha_j)$, où (α_i, α_j) est défini à la proposition 5.15, puis poursuivre à l'étape **e**.

e. Raffinement d'un produit (classes $\underline{C}_{IV}$ et $\overline{C}_{IV}$).
Pour chaque $(i, j) \in M$ tel que $\hat{w}_{ij} < \hat{x}_i\hat{x}_j - \epsilon_r$, ajouter les linéarisations $\underline{W}^{I}_{ij}(\underline{\beta}, \overline{\beta})$ et $\underline{W}^{III}_{ij}(\underline{\beta}, \overline{\beta})$ si elles ne sont pas déjà présentes (en utilisant les variables $\delta_i(\cdot)$ et $\delta_j(\cdot)$ courantes), où $\underline{\beta}$ et $\overline{\beta}$ sont des points d'évaluation tels que l'intervalle $[\underline{\beta}, \overline{\beta}] \supseteq [\frac{\hat{w}_{ij}}{\hat{x}_j}, \hat{x}_i]$ est le plus petit possible.
Pour chaque $(i, j) \in M$ tel que $\hat{w}_{ij} > \hat{x}_i\hat{x}_j + \epsilon_r$, ajouter les linéarisations $\overline{W}^{II}_{ij}(\underline{\beta}, \overline{\beta})$ et $\overline{W}^{IV}_{ij}(\underline{\beta}, \overline{\beta})$ si elles ne sont pas déjà présentes (en utilisant les variables $\delta_i(\cdot)$ et $\delta_j(\cdot)$ courantes), où $\underline{\beta}$ et $\overline{\beta}$ sont des points d'évaluation tels que l'intervalle $[\underline{\beta}, \overline{\beta}] \supseteq [\hat{x}_i, \frac{\hat{w}_{ij}}{\hat{x}_j}]$ Dans le cas où aucune linéarisation n'est ajoutée à l'une des étapes **c**, **d** ou **e**, poursuivre à l'étape **f**. Sinon, retourner à l'étape **b**.

f. Test d'optimalité.
Si la valeur optimale de la relaxation est plus grande que la meilleure valeur connue (moins ϵ_z), on cesse le traitement de ce noeud. Autrement, on poursuit à l'étape **g**.

g. Raffinement d'une surestimation d'un carré (classe C_{III}).
On crée un nouveau noeud qu'on ajoute à la liste. Les ensembles Δ^0 et Δ^1 sont identiques à ceux du noeud courant (puisque la variable de branchement fut ajoutée avant le traitement du noeud). La borne supérieure est la valeur optimale de la relaxation linéaire.
Déterminer une variable et une valeur de branchement à l'aide des propositions 5.16 ou 5.17. Si possible (et utile) on tentera de réutiliser les variables de branchement introduites à un autre noeud de l'arbre.
Fin du traitement du noeud. ∎

Un des éléments des plus importants de l'algorithme est le branchement réalisé à l'étape **g**. La variable de branchement retenue est celle dont l'erreur d'approximation est la plus élevée : l'indice i ou j qui maximise $\hat{v}_i - \hat{x}_i^2$ ou $|\hat{w}_{ij} - \hat{x}_i\hat{x}_j|$. Suite à l'obtention de cette variable de branchement x_i, on doit déterminer une valeur de branchement. Si l'algorithme a préalablement créé une variable $\delta_i(\alpha)$ et que cette variable est libre dans $[0, 1]$ et qu'elle pourrait éliminer la solution courante par l'intermédiaire des linéarisations correspondantes, alors α sera retenu comme valeur de branchement. Ce choix n'augmente pas le nombre de variables, et possiblement n'ajoute pas de contraintes. Autrement, l'algorithme détermine la valeur de branchement qui minimise l'erreur pondérée maximale. Pour ce faire, une nouvelle variable binaire est introduite via l'introduction d'une coupe de la classe C_{III}. Le prochain théorème démontre la validité de l'algorithme.

Théorème 5.19 *L'algorithme trouve en temps fini une solution ϵ_r-ϵ_z-optimale du problème QQP.*

Démonstration La phase de pré-traitement se termine dès que deux itérations successives n'améliorent pas de bornes d'une valeur supérieure à ϵ_r. Il en résulte nécessairement un nombre fini d'itérations. À chacune d'elles, on résout un nombre fini de problèmes linéaires. Le pré-traitement prend fin en temps fini.

Considérons un noeud généré lors de la phase d'exploration de l'arbre d'énumération où l'on raffine une surapproximation d'une fonction carré. Supposons que les

linéarisations de la variable x_i aient lieu au point $\alpha \in]\ell_i, u_i[$.

Nous montrons maintenant que les linéarisations associées à chacun des fils du noeud éliminent une région de norme non négligeable du domaine. Pour le noeud où $x_i \leq \alpha$, la linéarisation de la variable x_i sera à l'intérieur de la tolérance requise si $x_i \geq \alpha - \epsilon_r/u_i$. En effet, si cette condition est remplie, l'erreur maximale sera de

$$(\ell_i + \alpha)(\alpha - \frac{\epsilon_r}{u_i}) - \ell_i\alpha - (\alpha - \frac{\epsilon_r}{u_i})^2 = \frac{\alpha\epsilon_r}{u_i} - \frac{\ell_i\epsilon_r}{u_i} - \frac{\epsilon_r^2}{u_i^2} \leq \frac{\alpha\epsilon_r}{u_i} \leq \epsilon_r.$$

Pour le fils où $x_i \geq \alpha$, la linéarisation de la variable x_i sera à l'intérieur de la tolérance requise si $x_i \leq \alpha + \epsilon_r/u_i$. En effet, si cette condition est remplie, l'erreur maximale sera de

$$(\alpha + u_i)(\alpha + \frac{\epsilon_r}{u_i}) - \alpha u_i - (\alpha + \frac{\epsilon_r}{u_i})^2 = \epsilon_r - \frac{\epsilon_r^2}{u_i^2} - \frac{\alpha\epsilon_r}{u_i} \leq \epsilon_r.$$

Pour chacun des fils une un intervalle de longueur $\epsilon_r/u_i > 0$ est linéarisée de façon acceptable, et donc, si éventuellement on obtient une solution appartenant à ce domaine, la linéarisation ne sera pas raffinée davantage. Il peut donc n'y avoir qu'un nombre fini de linéarisations pour chacune des variables. Le même type de raisonnement s'applique pour l'approximation d'un produit de deux variables. La solution obtenue sera donc ϵ_r-approchée.

Les propositions de la section 5.2.2 impliquent que les inégalités sont valides et donc, la solution optimale (à l'intérieur de la tolérance ϵ_z) n'est jamais retranchée. Il existera donc un noeud de l'arbre d'énumération où une solution ϵ_r-ϵ_z-optimale sera identifiée. Étant donné que les phases de pré-traitement et d'exploration de l'arbre d'énumération se terminent en temps fini, il en résulte que l'algorithme trouve en temps fini une solution ϵ_r-ϵ_z-optimale de QQP. ∎

Cette preuve de convergence théorique ne donne aucune indication sur le temps réel requis par l'algorithme. Nous présentons maintenant les résultats de l'exécution de l'algorithme sur des problèmes tirés de la littérature issus d'applications pratiques.

5.3.2 Applications du problème QQP

L'algorithme décrit à la section précédente est appliqué à une série de problèmes provenant de la littérature. Afin d'améliorer sa performance, nous avons reformulé certains d'entre eux. En particulier, nous avons exploité les relations de monotonie décrites par Hansen *et al.* [95], nous avons mis à l'échelle plusieurs variables afin qu'elles partagent le même ordre de grandeur, et nous avons réindexé plusieurs variables. Les problèmes présentés ici sont des formulations équivalentes; leur structure originale peut être difficile à retracer. Pour chacun d'eux, une solution ϵ_r-ϵ_z-optimale obtenue par l'algorithme est aussi présentée.

Exemple 5.20 *Cet exemple classique de mélange pétrochimique présenté dans Haverly [98] et repris dans Floudas et Pardalos [77] (No 6.2) et Foulds* et al. *[82] (No 1). Le problème modélisé est le suivant. Nous avons quatre réservoirs : $R1, R2, R3$ et $R4$. Les deux premiers reçoivent trois produits de sources distinctes, puis leur contenu est combiné dans les deux autres afin de créer les mélanges désirés. La question est de déterminer la quantité de chaque produit à acheter afin de maximiser les profits.*

Dans cet exemple, le réservoir $R1$ reçoit deux produits de qualité (e.g., teneur en soufre) 3 et 1 en quantité q_0 et q_1. La qualité du mélange contenu dans $R1$ est $s = \frac{3q_0+q_1}{q_0+q_1} \in [1,3]$. Le réservoir $R2$ contient un produit de la troisième source, de qualité 2 en quantité q_2. On désire obtenir dans les réservoirs $R3$ et $R4$ de capacité 10 et 20, des produits de qualité d'au plus 2,5 et 1,5.

La figure 5.7, où les variables x_1 à x_4 représentent des quantités, illustre cette situation. Les prix unitaires des produits achetés sont respectivement 60, 160 et 100; ceux des produits finis sont 90 et 150. La différence entre les coûts d'achat et de vente est donc $60q_0 + 160q_1 + 100q_2 - 90(x_1 + x_3) - 150(x_2 - x_4)$. Les qualités des mélanges finaux sont $\frac{sx_1+2x_3}{x_1+x_3}$ et $\frac{sx_2+2x_4}{x_2+x_4}$.

L'ajout des contraintes de conservation de volume $q_0 + q_1 = x_1 + x_2$ et $q_2 = x_3 + x_4$ permet l'élimination des variables q_0, q_1 et q_2 :

$$q_0 = (s-1)(x_1 + x2)/2, \quad q_1 = (3-s)(x_1 + x2)/2, \quad q_2 = x_3 + x_4.$$

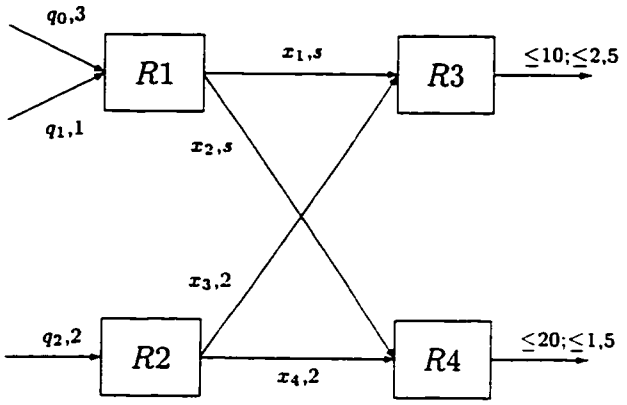


Figure 5.7 – *Un problème de mélange*

Le modèle mathématique transformé (la variable s est notée x_5) est le suivant :

$$\begin{aligned}
\min_{x} \quad & 120x_1 + 60x_2 + 10x_3 - 50x_4 - 50x_1x_5 - 50x_2x_5 \\
s.c. \quad & 2{,}5x_1 + 0{,}5x_3 - x_1x_5 \geq 0, \\
& 1{,}5x_2 - 0{,}5x_4 - x_2x_5 \geq 0, \\
& x_1 + x_3 \leq 10, \\
& x_2 + x_4 \leq 20, \\
& x_i \geq 0, \quad i = 1,2,3,4, \qquad 1 \leq x_5 \leq 3.
\end{aligned}$$

Solution : $\epsilon_r = \epsilon_z = 10^{-6}$, $z = -400$, $x = (0; 10; 0; 10; 1)$. ∎

Exemple 5.21 *Ce problème, rencontré par* Proctor and Gamble Co., *est tiré de Colville [60] (No 3). Il est repris dans Dembo [64] (No 2), Himmelblau [103] (No A11), Hock et Schittkowski [104] (No 83), Floudas et Pardalos [77] (No 3.2) et analysé dans Hansen* et al. *[95].*

$$\begin{aligned}
\min_{x} \quad & 372{,}93293x_1 + 83{,}56891x_1x_4 + 535{,}78547x_2^2 \\
s.c. \quad & -6{,}665593 \leq 0{,}22053x_2x_4 - 1{,}876314x_4 - 0{,}06262x_1x_3, \\
& 10{,}699039 \leq 0{,}47026x_2x_4 + 0{,}19085x_2x_3 + 0{,}12547x_1x_2 \leq 15{,}699039, \\
& 7{,}8 \leq x_1 \leq 10{,}2 \qquad\qquad 2{,}7 \leq x_3 \leq 4{,}5, \\
& 2{,}7 \leq x_2 \leq 4{,}5 \qquad\qquad 2{,}7 \leq x_4 \leq 4{,}5.
\end{aligned}$$

Solution : $\epsilon_r = \epsilon_z = 10^{-6}$, $z = 10126{,}6$, $x = (7{,}8; 2{,}99953; 4{,}5; 3{,}67758)$. ∎

Exemple 5.22 *Cet exemple provient de Bartholomew-Biggs [38]. Il est repris dans Hock et Schittkowski [104] (No 71) et dans Hansen et Jaumard [94] (No 5). Il s'agit d'un problème polynomial de degré quatre reformulé de façon quadratique.*

$$\begin{aligned}
\min_x \quad & x_3 + x_1x_5 + x_2x_5 + x_3x_5 \\
s.c. \quad & x_5 - x_1x_4 = 0, \\
& x_6 - x_2x_3 = 0, \\
& x_1^2 + x_2^2 + x_3^2 + x_4^2 = 40, \\
& x_5x_6 \geq 25, \\
& 1 \leq x_1 \leq 5 \qquad 1 \leq x_3 \leq 5, \\
& 1 \leq x_2 \leq 5 \qquad 1 \leq x_4 \leq 5.
\end{aligned}$$

Solution : $\epsilon_r = \epsilon_z = 10^{-6}$, $z = 17{,}014$, $x = (1; 4{,}74319; 3{,}8209; 1{,}37944; 1{,}37944; 18{,}1233)$. ∎

Exemple 5.23 *Cet exemple d'échangeurs de chaleur est décrit dans Avriel et Williams [21]. Il est repris dans Dembo [64] (No 5), Hock et Schittkowski [104] (No 106), Floudas et Pardalos [77] (No 3.1) et analysé dans Hansen* et al. *[95].*

Soit le système composé de trois échangeurs de chaleur disposés en série tel qu'illustré à la figure 5.8. On désire faire passer la température d'un fluide de t_0 à t_3 en l'envoyant à travers trois échangeurs.

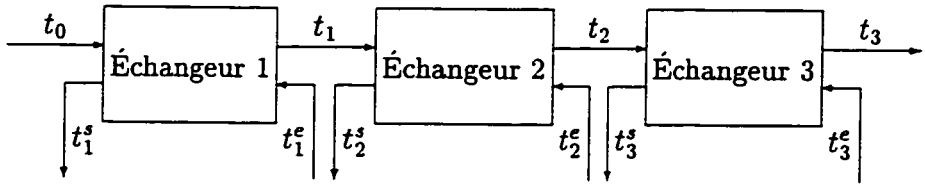


Figure 5.8 – *Échangeurs de chaleur*

Dans chacun des échangeurs $i = 1, 2, 3$, un fluide entre avec une température de t_i^e et en ressort avec une température t_i^s. Le débit de ces trois sources de chaleur est le même que celui du fluide que l'on désire chauffer. Les coefficients de transfert de chaleur u_1, u_2 et u_3 sont des constantes données.

On cherche à minimiser la somme des trois aires de transfert de chaleur : $x_1 + x_2 + x_3$. *Les inégalités suivantes assurent que la quantité de chaleur absorbée par le fluide froid est au plus, celle perdue par les sources de chaleur :*

$$t_i - t_{i-1} \ \leq \ t_i^e - t_i^s \qquad i = 1, 2, 3.$$

Les inégalités de transfert de chaleur s'écrivent

$$fc(t_i - t_{i-1}) \ \leq \ u_i x_i (t_i^e - t_{i-1}) \qquad i = 1, 2, 3,$$

où f *et* c *sont des constantes représentant le débit des fluides et la chaleur spécifique du fluide.*

Nous étudions l'exemple reformulé suivant :

$$\begin{array}{lll} \min\limits_{x} & 100x_1 + 100x_2 - 250x_4 & \\ s.c. & -30x_1 + 8{,}3333252x_3 + x_1x_3 & \leq 83{,}333333, \\ & -40x_2 - 12{,}5x_3 + 12{,}5x_4 + x_2x_4 & \leq 0, \\ & -x_3 + x_4 \leq 39, & \\ & 1 \leq x_1 \leq 100 & 1 \leq x_3 \ \leq 39, \\ & 10 \leq x_2 \leq 100 & 10 \leq x_4 \leq 46. \end{array}$$

Solution : $\epsilon_r = \epsilon_z = 10^{-5}$, $z = -5450{,}75$, $x = (5{,}79419; 13{,}5981; 18{,}2027; 29{,}5599)$. ∎

Exemple 5.24 *Bracken et McCormick [51] décrivent un problème d'alkylation d'oléfines. Cet exemple est reformulé et légèrement modifié dans Dembo [64] (No 3). Nous présentons ici la formulation de Liebman* et al. *[134], reprise dans Quesada et Grossmann [169] (No 11) (une faute de frappe apparaît dans cette dernière formulation du problème : la valeur* 9800 *de la cinquième contrainte du problème non transformé devrait être remplacée par* 98000*). Nous reprenons brièvement ici la description de la modélisation d'une unité d'alkylation présentée dans Bracken et McCormick [51].*

L'alkylation est un procédé utilisé en pétrochimie pour créer des alkanes à chaîne latérale ayant un taux d'octane élevé à partir d'oléfines. Ces dernières sont des hydrocarbones possédant des doubles liens carbones-carbones. Ces alkanes, saturés

en hydrogènes, sont habituellement mélangés (voir l'exemple 5.20) à la gazoline afin d'améliorer sa performance. Une réaction typique [168] est

$$\underset{\textit{Oléfine}}{(CH_3)_2C=CH_2} \quad + \quad \underset{\textit{Isobutane}}{(CH_3)_3CH} \quad \xrightarrow{\textit{Catalyseur}} \quad \underset{\textit{Alkane.}}{(CH_3)_2CHCH_2C(CH_3)_3}$$

La figure 5.9 illustre de façon simplifiée le procédé complet d'alkylation. La transformation mentionnée ci-dessus a lieu dans le réacteur. *Les oléfines et les isobutanes réagissent ensembles par l'intermédiaire du catalyseur. Les hydrocarbones produits sont ensuite séparés dans le* fractionneur. *Les molécules assez longues sont acceptées comme alcanes, et les autres sont redirigées vers le réacteur comme isobutanes.*

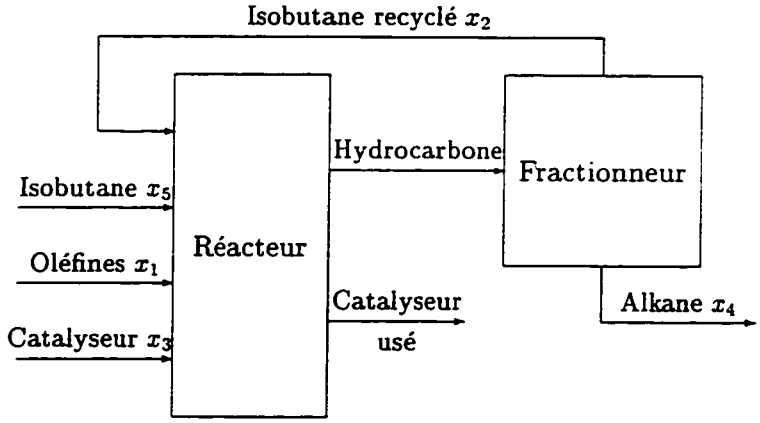


Figure 5.9 – *Unité d'alkylation*

Les variables x_1 à x_5 présentes sur la figure sont des variables de flot. Les autres variables nécessaires à la modélisation sont les suivantes :

x_6 : *force du catalyseur*
x_7 : *taux d'octane*
x_8 : *rapport externe isobutane-oléfine*
x_9 : *facteur de dilution du catalyseur*
x_{10} : *indice de performance F-4.*

Le rapport isobutane-oléfine x_8 est la proportion entre l'apport d'isobutane total x_2 et x_5 et la quantité d'oléfine x_1 :

$$x_8 = \frac{x_2 + x_5}{x_1}.$$

Le taux d'octane x_7 dépend du rapport isobutane-oléfine x_8 et de la force du catalyseur x_6 :

$$x_7 \leq 86{,}35 + 1{,}098x_8 - 0{,}038x_8^2 + 0{,}325(x_6 - 89).$$

La force du catalyseur x_6 est reliée au taux d'injection de catalyseur x_3, au taux de production d'alkane x_4 et au facteur de dilution du catalyseur x_9 comme suit :

$$1000x_3 = \frac{x_4 x_6 x_9}{98 - x_6}.$$

Cette équation s'écrit à l'aide de contraintes quadratiques impliquant une variable artificielle x_{11}

$$x_3 = x_6 x_{11} \quad \text{et} \quad 98000x_{11} - 1000x_3 = x_4 x_9.$$

Le taux de production d'alkane x_4 varie en fonction du taux d'oléfine x_1 et du rapport isobutane-oléfine x_8 de la façon suivante :

$$x_4 \leq x_1(1{,}12 + 0{,}13167x_8 - 0{,}00667x_8^2).$$

En ajoutant une variable artificielle x_{12}, on obtient les contraintes quadratiques équivalentes :

$$x_4 = x_1 x_{12} \quad \text{et} \quad x_{12} \leq 1{,}12 + 0{,}13167x_8 - 0{,}00667x_8^2.$$

La fonction objectif est linéaire en le coût des produits sources x_1, x_3 et x_5, et en le coût de l'isobutane recyclé x_2, et dépend linéairement du produit de la quantité d'alkane par le taux d'octane $x_4 x_7$.

Les autres contraintes des bornes physiques sur les variables et des relations linéaires. Suite à l'élimination de variables, une mise à l'échelle et une renumérotation des indices, on obtient le problème suivant :

$$
\begin{array}{lll}
\min\limits_{x} & 614{,}88x_1 - 171{,}5x_2 - 6{,}3x_1x_4 + 4{,}27x_1x_5 - 3{,}5x_2x_5 + 10x_3x_6 & \\
s.c. & -0{,}325x_3 + x_4 - 1{,}098x_5 + 0{,}038x_5^2 & \leq 57{,}425, \\
 & -0{,}13167x_5 + x_7 + 0{,}00667x_5^2 & \leq 1{,}12, \\
 & 0{,}1 \leq 12{,}2x_1 - 10x_2 & \leq 200, \\
 & 0{,}1 \leq -10x_2 + 12{,}2x_1x_5 - 10x_2x_5 & \leq 1600, \\
 & 65{,}346x_1 - 980x_6 - 0{,}666x_1x_4 + 10x_2x_5 & = 0, \\
 & x_1 - 1{,}22x_1x_7 + x_2x_7 & = 0, \\
 & x_3x_6 \leq 120, & \\
 & 0{,}01 \leq x_1 \leq 32{,}786885 & 3 \leq x_5 \leq 12, \\
 & 0 \leq x_2 \leq 20 & x_6 \leq 1{,}411765, \\
 & 85 \leq x_3 \leq 95 & 0{,}8196722 \leq x_7, \\
 & 92{,}66666 \leq x_4 \leq 95. &
\end{array}
$$

Solution : $\epsilon_r = \epsilon_z = 10^{-5}$, $z = -1161{,}34$, $x = (30{,}5649; 20; 90{,}6168; 94{,}188; 10.4111; 1{,}0833; 1{,}76786)$. ∎

Exemple 5.25 *Cet exemple de séparation membranaire en trois phases est tiré de Dembo [64] (No 6). Il est repris dans Hock et Schittkowski [104] (No 116) et dans Hansen et Jaumard [94] (No 2).*

Pour les membranes perméables au liquide ou au gaz, les techniques de séparation sont particulièrement utiles pour les trois applications suivantes (tirées textuellement de Perry et al. *[163]). i- La séparation de mélanges de composés ayant des propriétés chimiques et physiques similaires. ii- La séparation de mélanges d'isomeres structurés ou positionnés. iii- La séparation de mélanges contenant des composés thermalement instables.*

$$
\begin{aligned}
\min_{x} \quad & 12{,}62626(x_8 + x_9 + x_{10}) - 1{,}231059(x_1x_8 + x_2x_9 + x_3x_{10}) \\
s.c. \quad & 50 \leq 12{,}62626(x_8 + x_9 + x_{10}) - 1{,}231059(x_1x_8 + x_2x_9 + x_3x_{10}) \leq 250, \\
& 12{,}62626x_8 - 1{,}231059x_1x_8 && \leq 150, \\
& 12{,}62626x_9 - 1{,}231059x_2x_9 && \leq 150, \\
& 12{,}62626x_{10} - 1{,}231059x_3x_{10} && \leq 150, \\
& -3{,}475x_1 + 100x_4 + 0{,}0975x_1^2 - 9{,}75x_1x_4 && = 0, \\
& -3{,}475x_2 + 100x_5 + 0{,}0975x_2^2 - 9{,}75x_2x_5 && \leq 0, \\
& -3{,}475x_3 + 100x_6 + 0{,}0975x_3^2 - 9{,}75x_3x_6 && \leq 0, \\
& -x_4x_7 + x_5x_7 - x_1x_8 + x_4x_8 && \geq 0, \\
& 50x_5 + 50x_6 - x_1x_8 + x_5x_8 + x_2x_9 - x_6x_9 && = 500, \\
& -50x_2 + 50x_6 + x_2x_9 - x_6x_9 + x_2x_{10} - x_3x_{10} && \geq 0, \\
& 0 \leq x_7 - x_8 \leq 50, \\
& x_5 \leq x_1 \leq x_2 && x_6 \leq x_2 \leq x_3, \\
& 1 \leq x_1 \leq 8{,}03773157 && 1 \leq x_6 \leq 9, \\
& 4{,}5 \leq x_2 \leq 9{,}71853961 && 0{,}01 \leq x_7 \leq 100. \\
& 9 \leq x_3 \leq 10 && 0{,}01 \leq x_8 \leq 54{,}91813487, \\
& 0{,}001 \leq x_4 \leq 10 && 50 \leq x_9 \leq 100, \\
& 1 \leq x_5 \leq 4{,}68381588 && 0{,}01 \leq x_{10} \leq 50.
\end{aligned}
$$

Solution: $\epsilon_r = \epsilon_z = 10^{-5}$, $z = 97{,}5875$, $x = (8{,}03773; 8{,}99986; 9{,}71827; 1; 1{,}90813; 4{,}68136; 57{,}4077; 7{,}40772; 50{,}0017; 0{,}01)$. ∎

Exemple 5.26 *Cet exemple de séparation membranaire en cinq phases est tiré de Dembo [64] (No 7). Il s'agit d'une généralisation de l'exemple précédent.*

$$
\begin{aligned}
\min_x \quad & z = 12{,}62626(x_{12}+x_{13}+x_{14}+x_{15}+x_{16}) \\
& \quad -1{,}231059(x_1x_{12}+x_2x_{13}+x_3x_{14}+x_4x_{15}+x_5x_{16}) \\
s.c. \quad & 50 \le z \le 250, \\
& -3{,}475x_1+100x_6+0{,}0975x_1^2-9{,}75x_1x_6 \le 0, \\
& -3{,}475x_2+100x_7+0{,}0975x_2^2-9{,}75x_2x_7 \le 0, \\
& -3{,}475x_3+100x_8+0{,}0975x_3^2-9{,}75x_3x_8 \le 0, \\
& -3{,}475x_4+100x_9+0{,}0975x_4^2-9{,}75x_4x_9 \le 0, \\
& -3{,}475x_5+100x_{10}+0{,}0975x_5^2-9{,}75x_5x_{10} \le 0, \\
& -x_6x_{11}+x_7x_{11}-x_1x_{12}+x_6x_{12} \ge 0, \\
& 50x_7-50x_8-x_1x_{12}+x_2x_{13}+x_7x_{12}-x_8x_{13} = 0, \\
& 50x_8+50x_9-x_2x_{13}+x_3x_{14}+x_8x_{13}-x_9x_{14} \le 500, \\
& -50x_9+50x_{10}-x_3x_{14}+x_4x_{15}-x_8x_{15}+x_9x_{14} \le 0, \\
& 50x_4-50x_{10}-x_4x_{15}-x_4x_{16}+x_5x_{16}+x_{10}x_{15} \le 0, \\
& 50x_4-x_4x_{16}+x_5x_{16} \ge 450, \\
& -x_1+2x_7 \le 1, \\
& x_1 \le x_2 \le x_3 \le x_4 \le x_5 \qquad x_6 \le x_7, \\
& x_8 \le x_9 \le x_{10} \le x_4 \qquad 0 \le x_{11}-x_{12} \le 50, \\
& 1 \le x_1 \le 8{,}03773157 \qquad 1 \le x_9 \le 9, \\
& 1 \le x_2 \le 9 \qquad 1 \le x_{10} \le 9, \\
& 4{,}5 \le x_3 \le 9 \qquad 0{,}1 \le x_{11} \le 100, \\
& 4{,}5 \le x_4 \le 9 \qquad 10^{-7} \le x_{12} \le 100, \\
& 9 \le x_5 \le 10 \qquad 1 \le x_{13} \le 50, \\
& 0{,}001 \le x_6 \le 1 \qquad 50 \le x_{14} \le 100, \\
& 1 \le x_7 \le 4{,}51886579 \qquad 50 \le x_{15} \le 100, \\
& 1 \le x_8 \le 9 \qquad 10^{-7} \le x_{16} \le 50.
\end{aligned}
$$

Solution : $\epsilon_r = \epsilon_z = 10^{-5}$, $z = 174{,}788$, $x = (8{,}03773; 8{,}161; 9; 9; 9; 1; 1{,}07026; 1{,}90837; 1{,}90837; 1{,}90837; 50{,}5042; 0{,}504236; 7{,}26387; 50; 50; 0)$. ∎

Exemple 5.27 *Graham [92] montre que parmi tous les hexagones de diamètre unitaire, celui dont l'aire est la plus grande surface n'est pas l'hexagone régulier. Rappelons que le diamètre d'un polyèdre est la longueur du plus long segment de droite compris à l'intérieur de celui-ci. Il présente l'unique hexagone optimal, dont l'aire est approximativement 4% supérieure à celle de l'hexagone régulier. Reinhardt [172] prouve que pour les polygones ayant un nombre impair de sommets, les polygones réguliers maximisent la surface.*

Graham [92] conjecture que l'octogone de plus grande surface aura la disposition présentée à la figure 5.10. Les sommets $A_i, i = 0, 1, \ldots, 7$ sont les sommets de l'octogone, et les lignes représentent des diamètres unitaires. L'octogone est alors défini par l'enveloppe convexe des sommets.

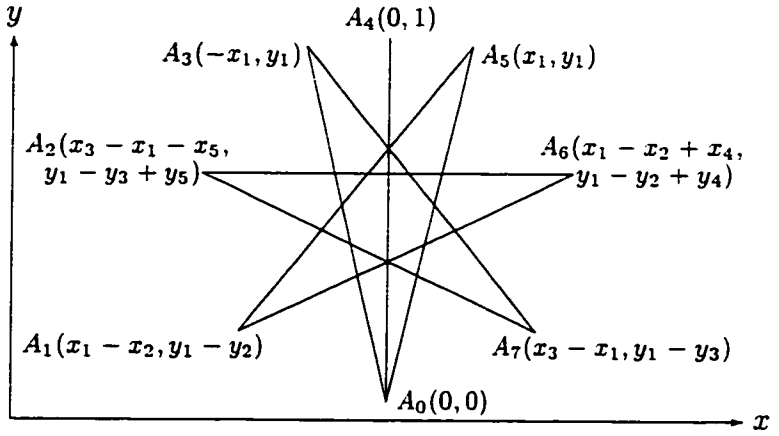


Figure 5.10 – *Octogone de diamètre unitaire d'aire maximale*

Hansen et Xiong [97] formulent cette question en un problème quadratique à contraintes quadratiques. Nous avons ajouté la contrainte $x_2 \geq x_3$, basée sur l'aspect symétrique du problème.

$$
\begin{aligned}
\min_{x} \quad & \tfrac{1}{2}\{(x_2+x_3-4x_1)y_1+(3x_1-2x_3+x_5)y_2+(3x_1-2x_2+x_4)y_3 \\
& \qquad +(x_3-2x_1)y_4+(x_2-2x_1)y_5\}+x_1 \\
s.c. \quad & \|A_0-A_1\|\le 1: \ (x_1-x_2)^2+(y_1-y_2)^2\le 1, \\
& \|A_0-A_2\|\le 1: \ (-x_1+x_3-x_5)^2+(y_1-y_3+y_5)^2\le 1, \\
& \|A_0-A_6\|\le 1: \ (x_1-x_2+x_4)^2+(y_1-y_2+y_4)^2\le 1, \\
& \|A_0-A_7\|\le 1: \ (-x_1+x_3)^2+(y_1-y_3)^2\le 1. \\
& \|A_1-A_2\|\le 1: \ (2x_1-x_2-x_3+x_5)^2+(-y_2-y_3+y_5)^2\le 1, \\
& \|A_1-A_3\|\le 1: \ (2x_1-x_2)^2+y_2^2\le 1, \\
& \|A_1-A_4\|\le 1: \ (x_1-x_2)^2+(y_1-y_2-1)^2\le 1, \\
& \|A_1-A_7\|\le 1: \ (2x_1-x_2-x_3)^2+(-y_2+y_3)^2\le 1, \\
& \|A_2-A_3\|\le 1: \ (x_3-x_5)^2+(-y_3+y_5)^2\le 1, \\
& \|A_2-A_4\|\le 1: \ (-x_1+x_3-x_5)^2+(y_1-y_3+y_5-1)^2\le 1, \\
& \|A_2-A_5\|\le 1: \ (2x_1+x_3-x_5)^2+(-y_3+y_5)^2\le 1, \\
& \|A_2-A_6\|= 1: \ (2x_1-x_2-x_3+x_4+x_5)^2+(-y_2+y_3+y_4-y_5)^2= 1, \\
& \|A_3-A_6\|\le 1: \ (-2x_1+x_2-x_4)^2+(y_2-y_4)^2\le 1, \\
& \|A_4-A_6\|\le 1: \ (x_1-x_2+x_4)^2+(y_1-y_2+y_4-1)^2\le 1, \\
& \|A_4-A_7\|\le 1: \ (x_1-x_3)^2+(1-y_1+y_3)^2\le 1, \\
& \|A_5-A_6\|\le 1: \ (x_2-x_4)^2+(y_2-y_4)^2\le 1, \\
& \|A_5-A_7\|\le 1: \ (2x_1-x_3)^2+y_3^2\le 1, \\
& \|A_6-A_7\|\le 1: \ (2x_1-x_2-x_3+x_4)^2+(-y_2+y_3+y_4)^2\le 1, \\
& x_2-x_3\ge 0 \\
& x_i^2+y_i^2=1 \qquad i=1,2,3,4,5, \\
& 0\le x_1\le 0{,}5 \qquad 0\le x_i\le 1, \quad i=2,3,4,5.
\end{aligned}
$$

Solution : $\epsilon_r=\epsilon_z=10^{-5}$, $z=0{,}72681$, $x=(0{,}25914;0{,}673516;0{,}670078;0{,}911984;0{,}913326)$, $y=(0{,}96584;0{,}739168;0{,}742285;0{,}410225;0{,}407218)$. ∎

5.3.3 Résultats numériques

L'algorithme de résolution du problème QQP est écrit en C++ et utilise les librairies CPLEX4.0 pour résoudre les problèmes de programmation linéaire. Les

expériences de calcul sont faites sur une station SPARC SS20/514MP sous Solaris 2.4-27. Les problèmes résolus ainsi que la précision utilisée sont énoncés à la section précédente.

Le tableau 5.1 résume les caractéristiques de ces exemples. Les trois premières colonnes contiennent respectivement le nombre de variables qui ne sont pas impliquées dans des termes quadratiques, le nombre de variables qui le sont, et le nombre total. La colonne centrale indique le nombre de termes quadratiques. Les trois dernières contiennent respectivement le nombre de contraintes d'inégalités linéaires (incluant les bornes sur les variables), et le nombre d'inégalités et d'égalités comprenant des termes quadratiques.

Tableau 5.1 – *Caractéristiques des problèmes*

Ex	Variables			Termes	Contraintes		
	Lin	Quad	Tot	Quad	Lin	Quad	
					$\leq$	$\leq$	$=$
5.20	2	3	5	2	7	2	0
5.21	0	4	4	6	8	3	0
5.22	0	6	6	9	8	1	3
5.23	0	4	4	2	9	2	0
5.24	0	7	7	6	14	4	2
5.25	0	10	10	15	26	9	2
5.26	0	16	16	24	42	10	1
5.27	0	10	10	43	10	17	6

Pour chacun des exemples 5.20 à 5.26, l'algorithme a été exécuté à deux reprises. En un premier temps, aucune information n'est donnée sur la nature d'un optimum. Ceci démontre la performance de l'algorithme lorsqu'aucune bonne solution n'est connue. En un deuxième temps, une solution ϵ_r-ϵ_z-optimale est fournie et l'algorithme doit prouver qu'elle est bel et bien ϵ_r-ϵ_z-optimale. La motivation derrière cette deuxième exécution repose sur l'utilisation d'heuristiques rapides générant souvent une excellente solution, mais sans aucune garantie sur sa précision.

Le tableau 5.2 présente les résultats de l'application de l'algorithme à ces exemples. Un tiret signifie que la preuve d'optimalité a été faite dans l'étape de

pré-traitement, et l'arbre d'énumération n'est pas généré. Les colonnes *borne* indiquent le nombre de variables qui sont à leur bornes après l'étape de pré-traitement. Les colonnes δ et *Ctr sup* contiennent respectivement le nombre total de nouvelles variables et contraintes (coupes) introduites au cours de l'exécution. Les colonnes *Noeuds* indiquent le nombre total de noeuds de l'arbre d'énumération. Les colonnes *Temps*, en secondes, sont divisées en trois : le temps du pré-traitement, de l'arbre d'énumération et le total.

Tableau 5.2 – *Résultats numériques de l'algorithme Al(QQP)*

Ex	Sans information							Preuve d'optimalité						
	Variables		Ctr	Noeuds	Temps (sec)			Variables		Ctr	Noeuds	Temps (sec)		
	borne	δ	sup		PT	Ar	Tot	borne	δ	sup		PT	Ar	Tot
5.20	3	5	34	9	0,76	0,08	0,84	5	0	7	1	0,34	0,01	0,35
5.21	2	6	69	7	2,89	0,09	2,98	-	-	305	-	0,93	-	0,93
5.22	1	21	460	51	6,7	7,7	14,4	-	-	80	-	95	-	95
5.23	0	71	674	571	2,6	57,4	60	0	39	362	329	312	13	325
5.24	1	70	1414	405	13,3	204	217	-	-	272	-	173	-	173
5.25	3	70	1529	405	493	362	855	-	-	663	-	346	-	346
5.26	9	161	5413	2277	737	15656	16393	11	2	108	5	277	0,1	277

Le temps requis pour la preuve d'optimalité est généralement plus court que celui où l'on ne dispose d'aucune information, en particulier lorsque l'étape de pré-traitement améliore les bornes sur les variables.

La formulation quadratique du problème de maximisation de la surface de l'octogone 5.27 renferme un grand nombre de termes quadratiques. Une analyse géométrique permet d'améliorer la performance de l'algorithme. On peut aisément vérifier que l'aire de l'heptagone régulier de diamètre unitaire est supérieure à celle de l'octogone régulier de diamètre unitaire. De façon générale, on peut vérifier que pour $n \geq 3$, l'aire du polygone régulier à $2n-1$ côtés : $\frac{2n-1}{2}(\frac{1-\cos\frac{\pi}{2n-1}}{\tan\frac{\pi}{2n-1}})$ est supérieure à celle du polygone à $2n$ côtés : $\frac{n}{4}\sin\frac{\pi}{n}$. Une solution initiale non optimale fut utilisée afin de diriger les recherches de l'algorithme dans une direction prometteuse. Il s'agit de l'octogone construit à partir de l'heptagone auquel nous avons remplacé une arête par deux nouvelles se rejoignant à un nouveau sommet. Le tableau 5.3 détaille les étapes requises pour la résolution.

Tableau 5.3 – *Évaluations successives d'aire d'octogones*

Étape	ϵ	Variables borne	δ	Ctr sup	Noeuds	Temps (hrs) PT	Ar	Tot	Bornes inf	sup
1	0,0005	0	90	9208	10959	0,1	30,6	30,6	0,7261	0,7290
2	0,0001	0	81	7714	5197	0,4	14,5	14,9		0,7262
3	0,0001	0	96	9562	8863	0,2	26,4	26,6		0,7262
4	0,0001	-	-	149	-	0,3	-	0,3		0,7262
5	0,0001	0	24	1356	219	5,1	0,1	5,2		0,7262
6	10^{-5}	0	96	9996	9901	1,4	31,0	32,4	0,7268	0,7268

La première étape consiste à résoudre le problème avec une précision $\epsilon_r = \epsilon_z = 0{,}0005$. En modifiant légèrement la solution produite afin d'obtenir une solution réalisable, on obtient un octogone de diamètre unitaire d'aire 0,7261. Cette étape est nécessaire car rappelons que l'algorithme produit une solution ϵ-réalisable. L'algorithme nous donne aussi une borne supérieure de l'aire maximale. À ce moment, on peut établir que l'aire optimale sera entre 0,7261 et 0,7290. Cette meilleure solution connue est utilisée comme solution de départ aux étapes 2 à 6.

Dans la solution réalisable que nous avons obtenue, la valeur de la variable x_1 était approximativement 0,27. Afin d'améliorer la précision de l'algorithme, il est nécessaire de restreindre le domaine réalisable, autrement il requiert trop de temps ou d'espace mémoire. Pour ce faire, nous avons à l'étape 2 ajouté la contrainte $x_1 \geq 0{,}3$ et avons redémarré l'algorithme avec la solution précédemment obtenue. Une borne supérieure fut générée garantissant qu'il n'y a pas de solution dont la valeur objectif dépasse 0,7262 lorsque $x_1 \geq 0{,}3$. Le même résultat est obtenu aux étapes 3, 4 et 5 où nous avons respectivement fixé $x_1 \leq 0{,}25$, $x_4 \geq 0{,}94$ et $x_4 \leq 0{,}87$.

Donc, s'il existe un octogone dont l'aire est supérieure à 0,7262, il sera tel que $x_1 \in [0{,}25; 0{,}3]$ et $x_3 \in [0{,}87; 0{,}94]$. En effet, en incorporant ces contraintes à l'étape 6 et en diminuant la précision à 10^{-5}, une meilleure solution est générée.

Selon la conjecture de Graham [92], on peut affirmer (après 110 heures de calculs par l'ordinateur) que l'octogone de diamètre unitaire de surface maximale renferme

une aire dont les quatre premières décimales sont 0,7268, et les coordonnées des sommets sont : $(0,0)$, $(-0{,}41266, 0{,}2251)$, $(-0{,}5, 0{,}63384)$, $(-0{,}25914, 0{,}96584)$, $(0,1)$, $(0{,}25914, 0{,}96584)$, $(0{,}5, 0{,}63384)$ et $(0{,}41266, 0{,}2251)$. La figure 5.11 contient l'octogone régulier, l'octogone de départ (généré à partir de l'heptagone régulier) et l'octogone d'aire maximale (toujours selon la même conjecture).

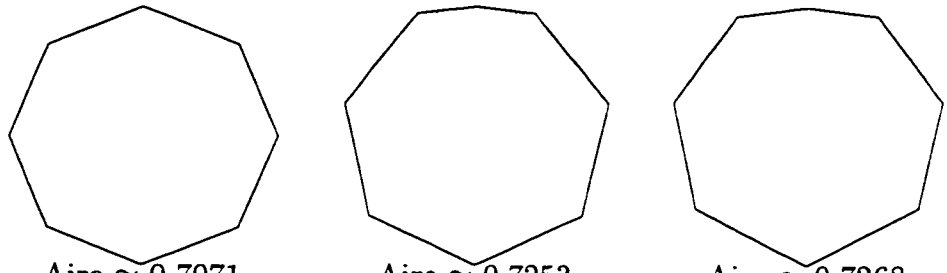


Figure 5.11 – *Comparaison d'octogones de diamètre unitaire*

5.4 Discussion

Le problème quadratique à contraintes quadratiques est un problème doublement difficile, car la seule question de trouver une solution réalisable définit un problème fortement NP-complet. De plus, même si le le domaine réalisable était un polyèdre, le problème demeurerait fortement NP-complet.

L'intérêt de l'étude de ce problème est qu'il englobe un vaste éventail d'applications réelles. Un grand nombre de problèmes, dont plusieurs proviennent du domaine de raffinage du pétrole, se formulent naturellement en problèmes quadratiques à contraintes quadratiques.

Nous avons présenté une approche d'approximation extérieure en générant des coupes aux noeuds d'un arbre d'énumération implicite. Celles-ci sont choisies parmi quatre classes de coupes, et sont valides sur tout le domaine. La sélection d'une coupe particulière d'une classe se fait selon un critère de minimisation d'une erreur potentielle. L'introduction de telles coupes nécessite occasionnellement de nouvelles

variables et contraintes. L'arbre d'énumération réutilise le plus possible les variables déjà introduites afin de garder la relaxation du problème de taille raisonnable. Cette réutilisation de l'information déjà présente dans la relaxation réduit considérablement le temps de calcul.

L'algorithme proposé converge en temps fini vers une solution approchée. Étant donné la difficulté du problème, une telle solution semble à l'heure actuelle, tout à fait acceptable. Nous avons obtenu des résultats encourageants en appliquant l'algorithme à plusieurs problèmes de la littérature issus de classes de problèmes différentes. Chacun d'eux est maintenant résolu avec une précision très exigeante. Les solutions présentées dans la littérature étaient souvent heuristiques, et aucune garantie n'était fournie tant qu'à leur précision. L'algorithme, même s'il n'exploite pas la structure intrinsèque additionnelle présente dans ces sous-classes, semble compétitif avec certains d'entre eux spécialisés à la résolution de celles-ci.

CONCLUSION

> La vie entière n'est *rien* autre que des questions devenues formes, qui portent en elles le germe de leur réponse — et des réponses grosses de questions.
>
> Gustav Meyrink

Une division naturelle en deux étapes principales se produit lorsque l'on désire répondre à une question d'optimisation globale issue d'une application réelle. La première étape consiste à décrire mathématiquement la situation rencontrée. Habituellement, la complexité du modèle croît très rapidement avec le réalisme et la précision exigée pour celui-ci. La seconde étape comprend la résolution numérique de ce modèle. Les outils disponibles comportent des méthodes algorithmiques d'optimisation définies pour différentes classes de problèmes, dont certaines englobent le modèle retenu.

Un équilibre doit être atteint afin de répondre à la question initiale. D'un côté, la modélisation doit être suffisamment raffinée afin de refléter la réalité de façon acceptable, et de l'autre, le modèle (ou une reformulation équivalente de celui-ci) doit faire partie d'une classe de problèmes dont nous disposons d'outils mathématiques assez puissants pour le résoudre.

Les modélisations actuelles sont souvent trop complexes pour une résolution précise par les méthodes existantes. Fréquemment, les méthodes proposées donnent des solutions approchées ou heuristiques, dont la précision est inconnue. La qualité de la modélisation est alors dégradée par une résolution mathématique inadéquate. Il y a un net besoin pour des méthodes de résolution plus performantes.

Nous nous sommes concentrés à l'intérieur de cette thèse sur la phase de résolution. Pour ce faire, nous avons en un premier temps étudié la structure et

les liens entre différentes classes de problèmes d'optimisation globale. Ensuite, les résultats théoriques engendrés par cette analyse ont conduit à l'élaboration de diverses méthodes de résolution plus efficaces que celles préalablement décrites dans la littérature.

Les reformulations d'une classe de problèmes à l'autre ne doivent pas être simplement interprétées comme des outils pour transformer un problème en un autre. Elles procurent un point de vue qui permet une meilleure compréhension des caractéristiques inhérentes des classes. Il est souvent bénéfique de d'étudier quelles propriétés se conservent suite à une reformulation. Réciproquement, certaines propriétés du problème reformulé ont leur contrepartie dans le problème original.

Basé sur de telles équivalences, nous avons développé un algorithme de résolution pour le problème bilinéaire disjoint qui exploite ses reformulations symétriques maxmin. Une voie de recherche ultérieure consisterait à étendre l'algorithme présenté au chapitre 2 par l'ajout de coupes de concavité (par exemple, à toutes les fois que la meilleure solution connue est améliorée). Probablement que ces coupes diminueraient significativement le nombre de noeuds à explorer.

L'étude des algorithmes plongés à d'autres classes de problèmes d'optimisation révéleraient possiblement des liens entre celles-ci dont on ignore actuellement l'existence. Il semble à première vue, que l'algorithme d'Ibaraki [109] pour le problème de complémentarité linéaire généralisé ait des similitudes avec celui de Hansen *et al.* [96] pour le problème linéaire biniveau. Afin de poursuivre l'analyse amorcée dans cette thèse, il serait intéressant de généraliser des coupes pour le problème mixte binaire (comme celles de Gomory par exemple) au problème biniveau. De manière générale, l'étude systématique de méthodes de résolution pour plusieurs problèmes d'optimisation mériteraient d'être élaborée afin d'identifier leur contrepartie par l'intermédiaire du plongement d'algorithmes. De plus, si deux algorithmes sont réciproquement plongés l'un dans l'autre, les deux classes de problèmes qu'ils résolvent sont alors très étroitement liées.

L'analyse des jeux bimatriciels sous un point de vue d'optimisation globale a permis l'énumération de toutes les stratégies d'équilibre pour des jeux de taille

importante. La spécialisation de l'algorithme pour le problème bilinéaire le simplifie à un point tel qu'un grand nombre de tests basés sur l'aspect bilinéaire se réduit à des trivialités. Une prochaine étape dans l'étude des jeux bimatriciel pourrait comprendre l'utilisation de cet outil afin d'explorer plus profondément nombreux raffinements du concept d'équilibre de Nash. Même pour les jeux de grande taille, le nombre modéré d'équilibres extrêmes (du moins pour les problèmes que nous avons considérés) fait en sorte qu'une étude énumérative soit possible.

L'étude du problème quadratique à contraintes quadratiques ouvre la porte à un très grand nombre d'applications. Ce problème très général englobe la modélisation de plusieurs situations réelles. La complexité de ce problème semble toutefois être nettement supérieure à celle associée aux autres problèmes étudiés dans le cadre de cette thèse. La méthode de linéarisation présentée ici, combinant l'énumération implicite et l'ajout de coupes valides sur tout le domaine, permet la résolution de problèmes difficiles. Des voies de recherches complémentaires pourraient naître de l'ajout de coupes non linéaires, ou de l'introduction de nouvelles classes de coupes, ou de la détermination de la meilleure coupe d'une classe selon différents critères. Toujours en vertu des idées directrices des reformulations et des plongements d'algorithmes, la spécialisation de notre méthode aux problèmes fractionnaires ou polynomiaux, adaptée selon les conclusions de l'étude approfondie de leur structure spécifiques pourraient entraîner d'importants avancements théoriques et algorithmiques.

L'approche unificatrice de l'étude parallèle de diverses classes de problèmes d'optimisation globale abordée dans cette thèse, combinée avec la venue d'ordinateurs de plus en plus puissants, fait qu'il est maintenant possible de répondre à des questions posées par l'industrie dont les solutions étaient, il n'y a pas si longtemps, inaccessibles.

BIBLIOGRAPHIE

[1] ADAMS W.P. et SHERALI H.D.(1986), "A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems," *Management Science* Vol.32 No.10, 1274–1290.

[2] ADAMS W.P. et SHERALI H.D.(1990), "Linearization Strategies for a Class of Zero-One Mixed Integer Programming Problems," *Operations Research* Vol.38 No.2, 217–226.

[3] ADAMS W.P. et SHERALI H.D.(1993), "Mixed-Integer Bilinear Programming Problems," *Mathematical Programming* 59, 279–305.

[4] AGGARWAL V.(1973), "On the Generation of All Equilibrium Points for Bimatrix Games through the Lemke-Howson Algorithm," *Mathematical Programming* 4, 233-234.

[5] AL-KHAYYAL F.A.(1986), "Linear Quadratic, and Bilinear Programming Approaches to the Linear Complementary Problem," *European Journal of Operational Research* 24, 216–227.

[6] AL-KHAYYAL F.A.(1987), "An Implicit Enumeration Procedure for the General Linear Complementary Problem," *Mathematical Programming Study* 31, 1–20.

[7] AL-KHAYYAL F.A.(1990), "Jointly Constrained Bilinear Programs and Related Problems: An Overview," *Computers & Mathematics with Applications* Vol.19 No.11, 53–62.

[8] AL-KHAYYAL F.A.(1992), "Generalized Bilinear Programming, Part I: Models, Applications and Linear Programming Relaxation," *European Journal of Operational Research* 60, 306–314.

[9] AL-KHAYYAL F.A. et FALK J.E.(1983), "Jointly Constrained Biconvex Programming," *Mathematics of Operations Research* Vol.8 No.2, 273–286.

[10] AL-KHAYYAL F.A. et LARSEN C.(1990), "Global Optimization of a Quadratic Function Subject to a Bounded Mixed Integer Constraint Set," *Annals of Operations Research* 25, 169–180.

[11] AL-KHAYYAL F.A., LARSEN C. et VAN VOORHIS T.(1995), "A Relaxation Method for Nonconvex Quadratically Constrained Quadratic Programs," *Journal of Global Optimization* 6, 215–230.

[12] ALTMAN M.(1968), "Bilinear Programming," *Bulletin de l'Académie Polonaise des Sciences, Série des Sciences Mathématiques, Astronomiques et Physiques* 16, 741–746.

[13] ANDROULAKIS I.P., MARANAS C.D. et FLOUDAS C.A.(1995), "αBB: A Global Optimization Method for General Constrained Nonconvex Problems." *Journal of Global Optimization* 6, 337–363.

[14] AUDET C., JAUMARD B. et SAVARD G.(1994), "Concavity Cuts for the Linear Maxmin Problem," *Les Cahiers du GERAD* G-94-52, Montréal.

[15] AUDET C., HANSEN P., JAUMARD B. et SAVARD G.(1995), "Links between the Linear Bilevel and Mixed $0-1$ Programming Problems," *Les Cahiers du GERAD* G-95-20, Montréal.

[16] AUDET C., HANSEN P., JAUMARD B. et SAVARD G.(1996), "A Symmetrical Linear Maxmin Approach to Disjoint Bilinear Programming," *Les Cahiers du GERAD* G-96-06, Montréal.

[17] AUDET C., HANSEN P., JAUMARD B. et SAVARD G.(1996), "Enumeration of All Extreme Equilibrium Strategies of Bimatrix Games," *Les Cahiers du GERAD* G-96-32, Montréal.

[18] AUDET C., HANSEN P., JAUMARD B. et SAVARD G.(1997), "Links between Linear Bilevel and Mixed $0-1$ Programming Problems," *Journal of Optimization Theory and Applications* Vol.93 No.2, 273–300.

[19] AUDET C., HANSEN P., JAUMARD B. et SAVARD G.(1997), "On the Linear Maxmin and Related Programming Problems," *Multilevel Optimization: Algorithm, Complexity and Applications*, (éditeurs MIGDALAS A., PARDALOS P.M. and VÄRBRAND P.), Kluwer Academic Publisher.

[20] AUDET C., HANSEN P., JAUMARD B. et SAVARD G.(1997), "A Branch and Cut Algorithm for Nonconvex Quadratically Constrained Quadratic Programming," à paraître dans *Les Cahiers du GERAD.*

[21] AVRIEL M. et WILLIAMS A.C.(1971), "An Extension of Geometric Programming with Applications in Engineering Optimization," *Journal of Engineering Mathematics* Vol.5 No.3, 187–194.

[22] BAKER T.E. et LASDON L.S.(1985), "Successive Linear Programming at Exxon," *Management Science* Vol.31 No.3, 264–274.

[23] BALAS E.(1971), "Intersection Cuts – A New Type of Cutting Planes for Integer Programming," *Operations Research* 19, 19–39.

[24] BALAS E.(1985), "Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems," *Siam Journal on Algebraic and Discrete Methods* Vol.6 No.3, 466–486.

[25] BALAS E., BOWMAN V.J., GLOVER F. et SOMMER D. (1971), "An Intersection Cut from the Dual of the Unit Hypercube," *Operations Research* 19, 40–44.

[26] BALAS E., CERIA S. et CORNUÉJOLS G.(1993), "A Lift-and-Project Cutting Plane Algorithm for Mixed $0-1$ Programs," *Mathematical Programming* 58, 295–324.

[27] BALAS E., CERIA S. et CORNUÉJOLS G.(1996), "Mixed $0-1$ Programming by Lift-and-Project in a Branch-and-Cut Framework," *Management Science* Vol.42 No.9, 1229–1246.

[28] BALAS E. et JEROSLOW R.G.(1980), "Strengthening Cuts for Mixed Integer Programs," *European Journal of Operational Research* 4, 224–234.

[29] BALINSKI M.L.(1961), "An Algorithm for Finding all Vertices of Convex Polyhedral Sets," *Journal of the Society for Industrial and Applied Mathematics* 9, 72–88.

[30] BARON D.P.(1972), "Quadratic Programming with Quadratic Constraints," *Naval Research Logistics Quarterly* 19, 253–260.

[31] BARD J.F.(1984), "Optimality Conditions for the Bilevel Programming Problem," *Naval Research Logistic Quarterly* 31, 13–26.

[32] BARD J.F.(1984), "An Investigation of the Linear Three Level Programming Problem," *IEEE Transactions on Systems, Man, and Cybernetics* Vol.14 No.5, 711–717.

[33] BARD J.F.(1985), "Geometric and Algorithmic Developments for a Hierarchical Planning Problem," *European Journal of Operational Research* 19, 372–383.

[34] BARD J.F.(1991), "Some Properties of the Bilevel Programming Problem," *Journal of Optimization Theory and Applications* 68, 371–378.

[35] BARD J.F. et FALK J.(1982), "An Explicit Solution to the Multi-Level Programming Problem," *Computers & Operations Research* Vol.9 No.1, 77–100.

[36] BARD J.F. et MOORE J.T.(1990), "A Branch and Bound Algorithm for the Bilevel Linear Programming Problem," *SIAM Journal on Scientific and Statistical Computing* 11, 281–292.

[37] BARD J.F. et MOORE J.T.(1992), "An Algorithm for the Discrete Bilevel Programming Problem," *Naval Research Logistics* 39, 419–435.

[38] BARTHOLOMEW-BIGGS M.C.(1976), "A Numerical Comparison Between Two Approaches to Nonlinear Programming Problems," *Technical Report #77* Numerical Optimization Center, Hatfield England.

[39] BASTIAN M.(1976), "Another Note on Bimatrix Games," *Mathematical Programming* 11, 299–300.

[40] BEALE E.M.L. et SMALL R.E.(1965), "Mixed Integer Programming by a Branch and Bound Technique," *Proceedings of the 3rd IFIP Congress 1965* 2, 450–451.

[41] BEN-AYED O.(1993), "Bilevel Linear Programming," *Computers & Operations Research* Vol.20 No.5, 485–501.

[42] BEN-AYED O. et BLAIR C.E.(1990), "Computational Difficulties of Bilevel Linear Programming," *Operations Research* 38, 556–559.

[43] BENDERS J.F.(1962), "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik* 4, 238–252.

[44] BENSON H.P.(1989), "On the Structure and Properties of a Linear Multilevel Programming Problem," *Journal of Optimization Theory and Applications* 60, 353–373.

[45] BENSON H.P.(1995), "Concave Minimization: Theory, Applications and Algorithms," dans *Handbook of Global Optimization* (éditeurs HORST R. et PARDALOS P.M.), Kluwer Academic Publishers, Boston, 43–148.

[46] BEN-TAL A. et TEBOULLE M.(1996), "Hidden Convexity in some Nonconvex Quadratically Constrained Quadratic Programming," *Mathematical Programming* 72, 51–63.

[47] BIALAS W. et KARWAN M.(1982), "On Two-Level Optimization," *IEEE Transactions on Automatic Control* 27, 211–214.

[48] BLAIR C.E.(1992), "The Computational Complexity of Multi-Level Linear Programs," *Annals of Operations Research* 34, 13–19.

[49] BLAU G.E. et WILDE D.J.(1969), "Generalized Polynomial Programming," *The Canadian Journal of Chemical Engineering* 47, 317–326.

[50] BORM P.E., JANSEN M.J.M., POTTERS J.A.M. et TIJS S.H.(1993), "Pareto Equilibria for Bimatrix Games," *Computers & Mathematics with Applications* 25, 19–25.

[51] BRACKEN J. et McCORMICK G.P.(1968), *Selected Applications of Nonlinear Programming* John Wiley and Sons, New York.

[52] BRETTHAUER K.M., CABOT A.V. et VENKATARAMANAN M.A.(1994), "An Algorithm and New Penalties for Concave Integer Minimization over a Polyhedron," *Naval Reseach Logistics* 41, 435–454.

[53] BRETTHAUER K.M.(1994), "A Penalty for Concave Minimization Derived from the Tuy Cutting Plane," *Naval Reseach Logistics* 41, 455–463.

[54] CABOT A.V. et FRANCIS R.L.(1970), "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points," *Operations Research* 18, 82–86.

[55] CANDLER W. et TOWNSLEY R.(1982), "A Linear Two-Level Programming Problem," *Computers & Operations Research* Vol.9 No.1, 59–76.

[56] CAROTENUTO L. et RAICONI G.(1987), "On the Minimization of Quadratic Functions with Bilinear Constraints via Augmented Lagrangeans," *Journal of Optimization Theory and Applications* Vol.55 No.1, 23–36.

[57] CHUNG S.J.(1989), "NP-Completeness of the Linear Complementarity Problem," *Journal of Optimization Theory and Applications* Vol.60 No.3, 393–399.

[58] CHVÁTAL V.(1973), "On the Computational Complexity of Finding a Kernel," Report No. CRM–300, Centre de Recherches Mathématiques, Université de Montréal.

[59] CLARK P.A. et WESTERBERG A.W.(1988), "A Note on the Optimality Conditions for the Bilevel Programming Problem," *Naval Research Logistics* 35, 413–418.

[60] COLVILLE A.R.(1970), *A Comparative Study of Nonlinear Programming Codes* dans *Princeton Symposium on Mathematical Programming* (éditeur KUHN H.W.), Princeton University Press.

[61] CZOCHRALSKA I.(1982), "Bilinear Programming," *Zastosowania Matematyki* 17, 495–514.

[62] CZOCHRALSKA I.(1982), "The Method of Bilinear Programming for Nonconvex Quadratic Programming," *Zastosowania Matematyki* 17, 515–525.

[63] DANTZIG G.B.(1963), *Linear Programming and Extensions* Princeton University Press, Princeton, New Jersey.

[64] DEMBO R.S.(1976), "A Set of Geometric Programming Test Problems and their Solutions," *Mathematical Programming* 10, 192–213.

[65] DICKHAUT J. et KAPLAN T.(1991), "A Program for Finding Nash Equilibria," *The Mathematica Journal* 1, 87–93.

[66] DU D.Z. et PARDALOS P.M.(1995), *Minimax and Applications*, World Scientific.

[67] EAVES B.C.(1971), "The Linear Complementarity Problem," *Management Science* Vol.17 No.19, 612–634.

[68] EAVES B.C. et LEMKE C.E.(1981), "Equivalence of LCP and PLS." *Mathematics of Operations Research* Vol.6 No.4. 475–484.

[69] EDMUNDS T.A. et BARD J.F.(1991), "Algorithms for Nonlinear Bilevel Mathematical Programs," *IEEE Transactions on Systems, Man, and Cybernetics* Vol.21 No.1, 83–89.

[70] FALK J.E.(1969), "Lagrange Multiplier and nonconvex Programs," *SIAM Journal on Control* 7, 534–545.

[71] FALK J.E.(1973), "A Linear Max-Min Problem," *Mathematical Programming* 5, 169–188.

[72] FALK J.E. et HOFFMAN K.R.(1976), "A Successive Underestimation Method for Concave Minimization Problems," *Mathematics of Operations Research* Vol.1 No.3, 251–259.

[73] FALK J.E. et LIU J.(1995), "On Bilevel Programming, Part I: General Nonlinear Cases," *Mathematical Programming* 70, 47–72.

[74] FALK J.E. et SOLAND R.M.(1969), "An Algorithm for Separable Nonconvex Programming Problems," *Management Science* Vol.15 No.9, 550–569.

[75] FLIPPO O.E.(1989), "Stability, Duality and Decomposition in General Mathematical Programming," Ph.D Thesis, ERASMUS University.

[76] FLIPPO O.E. et RINNOOY KAN A.H.G.(1993), "Decomposition in General Mathematical Programming," *Mathematical Programming* 60, 361–382.

[77] FLOUDAS C.A. et PARDALOS P.M.(1990), *A Collection of Test Problems for Constrained Global Optimization Algorithms* Vol. 455 of *Lecture Notes in Computer Sciences* Springer-Verlag, Berlin New-York.

[78] FLOUDAS C.A. et VISWESWARAN V.(1990), "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs-I. Theory," *Computers and Chemical Engineering* Vol.14 No.12, 1397–1417.

[79] FLOUDAS C.A. et VISWESWARAN V.(1993), "Primal-Relaxed Dual Global Optimization Approach," *Journal of Optimization Theory and Applications* Vol.78 No.2, 237–260.

[80] FLOUDAS C.A. et VISWESWARAN V.(1995), "Quadratic Optimization," dans *Handbook of Global Optimization* (éditeurs HORST R. et PARDALOS P.M), Kluwer Academic Publishers, Boston, 217–269.

[81] FORTUNY-AMAT J. et McCARL B.(1981), "A Representation and Economic Interpretation of a Two-Level Programming Problem," *Journal of the Operational Research Society* 32, 783–792.

[82] FOULDS L.R, HAUGLAND D. et JÖRNSTEN K.(1992), "A Bilinear Approach to the Pooling Problem," *Optimization* 24, 165–180.

[83] FRANGIONI A.(1995), "On a New Class of Bilevel Programming Problems and its Use for Reformulating Mixed Integer Problems," *European Journal of Operational Research* 82, 615–646.

[84] FRANK M. et WOLFE P.(1956), "An Algorithm for Quadratic Programming," *Naval Research Logistics Quarterly* 3, 95–110.

[85] GALLO G. et ÜLKÜCÜ A.(1977), "Bilinear Programming: an Exact Algorithm," *Mathematical Programming* 12, 173–194.

[86] GAREY M.R. et JOHNSON D.S.(1979), "Computers and Intractability," W.H. Freeman and Company, New York.

[87] GENDREAU M., MARCOTTE P. et SAVARD G.(1996), "A Hybrid Tabu-Asent Algorithm for the Linear Bilevel Programming Problem," *Journal of Global Optimization* 8, 217–233.

[88] GEOFFRION A.M.(1972), "Generalized Benders Decomposition," *Journal of Optimization Theory and Applications* Vol.10 No.4, 237–260.

[89] GEOFFRION A.M. et MARSTEN R.E.(1972), "Integer Programming Algorithms: A Framework and State-of-the-Art Survey," *Management Science* Vol.18 No.9, 465–491.

[90] GILBOA I. et ZEMEL E.(1989), "Nash and Correlated Equilibria: Some Complexity Consideration," *Games and Economic Behavior* 1, 80–93.

[91] GOWDA M.S.(1996), "On the Extended linear Complementarity Problem," *Mathematical Programming* 72, 33–50.

[92] GRAHAM R.L.(1975), "The Largest Small Hexagon," *Journal of Combinatorial Theory* 18, 165–170.

[93] GÜLER O.(1995), "Generalized Linear Complementarity Problems," *Mathematics of Operations Research* Vol.20 No.2, 441–448.

[94] HANSEN P. et JAUMARD B.(1992), "Reduction of Indefinite Quadratic Programs to Bilinear Programs," *Journal of Global Optimization* 2, 41–60.

[95] HANSEN P., JAUMARD B. et LU S.(1991), "An analytical Approach to Global Optimization," *Mathematical Programming* 52, 227–254.

[96] HANSEN P., JAUMARD B. et SAVARD G.(1992), "New Branch-and-Bound Rules for Linear Bilevel Programming," *SIAM Journal on Scientific and Statistical Computing* 13, 1194–1217.

[97] HANSEN P. and XIONG J.(1996), "The Largest Small Octagon," *Les Cahiers du GERAD* G-96-27, Montréal.

[98] HAVERLY C.A.(1996), "Studies of the behaviour of Recursion for the Pooling Problem," *ACM SIGMAP Bulletin* 25, 19–28.

[99] HE B.(1992), "A Projection and Contraction Method for a Class of Linear Complementarity Problems and its Application in Convex Quadratic Programming," *Applied Mathematics and Optimization* 25, 247–262.

[100] HEUER G.A.(1975), "On Completely Mixed Strategies in Bimatrix Games," *The Journal of the London Mathematical Society* 11, 17–20.

[101] HEUER G.A.(1975), "Uniqueness of Equilibrium Points in Bimatrix Games," *International Journal of Game Theory* Vol.8 No.1, 13–25.

[102] HEUER G.A. et MILLHAM C.B.(1976), "On Nash Subsets ans Mobility Chains in Bimatrix Games," *Naval Research Logistics Quarterly* 23, 311–319.

[103] HIMMELBLAU D.M.(1972), *Applied Nonlinear Programming,* McGraw-Hill, New York London.

[104] HOCK W. et SCHITTKOWSKI K.(1981), *Test Examples for Nonlinear Programming Codes,* Vol. 187 of *Lecture Notes in Economics and Mathematical Systems* Springer-Verlag, Berlin New-York.

[105] HORST R. et PARDALOS P.M. (1995), *Handbook of Global Optimization* Kluwer Academic Publishers Boston.

[106] HORST R. et THOAI N.V.(1989), "Modification, Implementation and Comparison of Three Algorithms for Globally Solving Linearly Constrained Concave Minimization Problems," *Computing* 42, 271–289.

[107] HORST R. et THOAI N.V.(1992), "Conical Algorithm for the Global Minimization of Linearly Constrained Decomposable Concave Minimization Problems," *Journal of Optimization Theory and Applications* 74, 469–486.

[108] HORST R. et TUY H.(1996), *Global Optimization (Deterministic Approaches), third edition* Springer-Verlag Berlin New-York.

[109] IBARAKI T.(1971), "Complementary Programming," *Operations Research* 19, 1523–1528.

[110] ISAACSON K. et MILLHAM C.B.(1980), "On a Class of Nash-Solvable Bimatrix Games and some Related Nash Subsets," *Naval Research Logistics Quarterly* 27, 407–412.

[111] IVANILOV Y.P. et MUKHAMEDIEV B.M.(1976), "An Algorithm for Solving the Linear Max-Min Problem," *Izv. Akad. Nauk SSSR, Tekhn. Kibernitika* 6, 3–10 [version anglaise :*Engineering Cybernetics* 14, 1–7].

[112] JANSEN M.J.M.(1981), "Maximal Nash Subsets for Bimatrix Games," *Naval Research Logistics Quarterly* 18, 147–152.

[113] JANSEN M.J.M.(1981), "Regularity and Stability of Equilibrium Points in Bimatrix Games," *Mathematics of Operations Research* Vol.6 No.4, 530–550.

[114] JANSEN M.(1993), "On the Set of Proper Equilibria of a Bimatrix Game," *International Journal of Game Theory* 22, 97–106.

[115] JAUMARD B., SAVARD G., XIONG X.(1995), "An exact Algorithm for Convex Bilevel Programming," *Les Cahiers du GERAD* G-95-33, Montréal.

[116] JEROSLOW R.G.(1973), "There Cannot be any Algorithm for Integer Programming with Quadratic Constraints," *Operations Research* 21, 221–224.

[117] JEROSLOW R.G.(1985), "The Polynomial Hierarchy and a Simple Model for Competitive Analysis," *Mathematical Programming* 32, 146–164.

[118] JÚDICE J.J. et FAUSTINO A.M.(1988), "The Solution of the Linear Bilevel Programming Problem by using the Linear Complementary Problem," *Investigação Operacional* 8, 77–95.

[119] JÚDICE J.J. et FAUSTINO A.M.(1992), "A Sequential LCP Method for Bilevel Linear Programming," *Annals of Operations Research* 34, 89–106.

[120] JÚDICE J.J. et MITRA G.(1988), "Reformulation of Mathematical Programming Problems as Linear Complementary Problems and Investigation of their Solution Methods," *Journal of Optimization Theory and Applications* Vol.57 No.1, 123–149.

[121] JÚDICE J.J. et VICENTE L.N.(1994), "On the Solution and Complexity of a Generalized Linear Complementarity Problem," *Journal of Global Optimization* 4, 415–424.

[122] KALANTARI B. et ROSEN J.B.(1987), "An Algorithm for Global Minimization of Linearly Constrained Concave Quadratic Functions," *Mathematics of Operations Research* Vol.12 No.3, 544–561.

[123] KOCHENBERGER G.A. et RICHARD V.H.(1982), "A Simple, All Primal Branch and Bound Approach to Pure and Mixed Integer Binary Programs," *Operations Research Letters* Vol.1 No.5, 182–185.

[124] KOHLBERG E. (1990), "Refinements of the Nash Equilibrium: The Main Ideas," dans *Game Theory and Applications* (éditeurs ICHIISHI T., NEYMAN A. et TAUMAN Y.), San Diego Academic Press, 3–45.

[125] KOHLBERG E. et MERTENS J.F.(1986), "On the Strategic Stability of Equilibria," *Econometrica* Vol.54 No.5, 1003–1037.

[126] KONNO H.(1971), "Bilinear Programming: Part II. Applications of Bilinear Programming," Technical Report No.71-10, Operations Research House, Department of Operations Research, Stanford University, Stanford.

[127] KONNO H.(1976), "A Cutting Plane Algorithm for Solving Bilinear Programs," *Mathematical Programming* 11, 14–27.

[128] KONNO H.(1976), "Maximization of a Convex Quadratic Function Over Linear Constraints," *Mathematical Programming* 11, 117–127.

[129] KUHN H.W.(1961), "An Algorithm for Equilibrium Points in Bimatrix Games," *Proceedings of the National Academy of Sciences* 47, 1657–1662.

[130] LABBÉ M., MARCOTTE P. et SAVARD G.(1996), "A Bilevel Model of Taxation and its Application to Optimal Highway Pricing," *Les Cahiers du GERAD* G-96-22, Montréal.

[131] LASDON L., WAREN A. SARKAR S. et PALACIOS.(1979), "Solving the Pooling Problem Using Generalized Reduced Gradient and Successive Linear Programming Algorithms," *ACM Sigmap Bulletin* 27, 9-25.

[132] LEMKE C.E.(1965), "Bimatrix Equilibrium Points and Mathematical Programming," *Management Science* Vol.11 No.7, 681–689.

[133] LEMKE C.E. et HOWSON T.T.(1964), "Equilibrium Points of Bimatrix Games," *Journal of the Society for Industrial and Applied Mathematics* 12, 413–423.

[134] LIEBMAN J., LASDON L, SCHRAGE L. et WAREN A.(1986), *Modelling and Optimization with GINO* The Scientific Press, Palo Alto, CA.

[135] LIU Y.H. et HART S.M.(1994), "Characterizing an Optimal Solution to the Bilevel Programming Problem," *European Journal of Operational Research* 79, 164–166.

[136] LODWICK W.A.(1992), "Preprocecing Nonlinear Functional Constraints with Applications to the Pooling Problem ," *ORSA Journal on Computing* Vol.4 No.2, 119-131.

[137] LORIDAN P. et MORGAN J. (1996), "Weak via Strong Stackelberg Problem: New Results," *Journal of Global Optimization* 8, 263–287.

[138] LOVÁSZ L. et SCHRIJVER A.(1991), "Cones of Matrices and Set-Functions and $0-1$ Optimization," *SIAM Journal on Optimization* 1, 166–190.

[139] LUCE R.D. et RAIFFA H.(1957), *Games and Decisions* John Wiley and Sons, New-York.

[140] LUTZENKO A.D. et MARTYNOV A.V.(1968), "Minimax Solutions of Problems in Linear and Quadratic Programming," *Izv. Akad. Nauk SSSR, Tekhn. Kibernitika* 2 [version anglaise :*Engineering Cybernetics* Vol.8 No.2 22–27].

[141] MANGASARIAN O.L.(1964), "Equilibrium Points of Bimatrix Games," *Journal of the Society for Industrial and Applied Mathematics* 12, 778–780.

[142] MANGASARIAN O.L.(1995), "The Linear Complementarity Problem as a Separable Bilinear Program," *Journal of Global Optimization* 6, 153–161.

[143] MANGASARIAN O.L. et STONE H.(1964), "Two-Person Nonzero-Sum Games and Quadratic programming," *Journal of Mathematical Analysis and Applications* 9, 348–355.

[144] MARCOTTE P. et SAVARD G.(1991), "A Note on the Pareto Optimality of Solutions to the Linear Bilevel Programming Problem," *Computers & Operations Research* Vol.18 No.4, 355–359.

[145] MARCOTTE P., WU S. et CHEN Y.(1993), "A Cutting-Plane Algorithm for the Linear Bilevel Programming Problem," *Cahiers du Centre de Recherche sur les Transports* CRT 925, Université de Montréal.

[146] McKELVEY R.D. et McLENNAN A. (forthcomming), "Computation of Equilibria in Finite Games," *Handbook of Computational Economics, Vol. I*, (éditeurs AMMAN H.M., KENDRICK D.A. et RUST J.), Elsevier, Amsterdam, 87–142.

[147] MIGDALAS A.(1995), "Bilevel Programming in Traffic Planning: Models, Methods and Challenge," *Journal of Global Optimization* 7, 381–405.

[148] MIGDALAS A. et PARDALOS P.M. -Editors- (1996), "Special Issue on Hierarchical and Bilevel Programming," *Journal of Global Optimization* 8.

[149] MILLHAM C.B.(1968), "On the Structure of Equilibrium Points in Bimatrix Games," *SIAM Review* Vol.10 No.4, 447–448.

[150] MILLHAM C.B.(1974), "On Nash Subsets of Bimatrix Games," *Naval Research Logistics Quarterly* 74, 307–317.

[151] MILLS H.(1960), "Equilibrium Points in Finite Games," *Journal of the Society for Industrial and Applied Mathematics* 8, 397–402.

[152] MOORE J.T. et BARD J.F.(1990), "The Mixed Integer Linear Bilevel Programming Problem," *Operations Research* Vol.38 No.5, 911–921.

[153] MOSHIRVAZIRI K., AMOUZEGAR M.A., JACOBSEN S.E.(1996), "Test Problem Construction for Linear Bilevel Programming Problems," *Journal of Global Optimization* 8,235–243.

[154] MUKHAMEDIEV B.M(1978), "The Solution of Bilinear Programming Problems and Finding the Equilibrium Situations in Bimatrix Games," *Zh. vychisl. Mat. mat. Fiz.* Vol.18 No.2, 351–359 [version anglaise : *U.S.S.R. Computational Mathematics and Mathematical Physics* Vol.18 No.2, 60–66].

[155] MUKHAMEDIEV B.M.(1982), "Approximate Method of Solving Concave Programming Problems," *U.S.S.R. Computational Mathematics and Mathematical Physics* Vol.22 No.3, 238–245.

[156] MURTY K.G. et KABADI S.N.(1987), "Some NP-Complete Problems in Quadratic and Nonlinear Programming," *Mathematical Programming* 39, 117–130.

[157] MYERSON R.B.(1978), "Refinements of the Nash Equilibrium Concept," *International Journal of Game Theory* Vol.7 No.2, 73–80.

[158] NASH J.F.(1950), "Equilibrium Points in n-Person Games," *Proceedings of the National Academy of Sciences* 36, 48–49.

[159] NASH J.F.(1951), "Noncooperative Games," *Annals of Mathematics* 54, 286–295.

[160] OSBORNE M.J. et RUBINSTEIN A.(1996), *A Course in Game Theory* The MIT Press, Cambridge, London.

[161] PARDALOS P.M. (1996), "Continuous Approaches to Discrete Optimization Problems," dans *Nonlinear Optimization and Applications* (éditeurs DI PILLO G. et GIANNESSI F.), Plenum Publishing, 313–328.

[162] PARDALOS P.M. et ROSEN J.B.(1986), "Methods for Global Concave Minimization: A Bibliographical Survey," *SIAM Review* Vol.28 No.3, 367–379.

[163] PERRY R.H., GREEN D.W. et MALONEY J.O.(1984), *Perry's Chemical Engineer's Handbook* 6th ED, McGraw Hill, New York London 17.14–17.18.

[164] PHILLIPS A.T. et ROSEN J.B.(1992), "Sufficient Conditions for Fast Solution of Linearly Constrained global Minimization Problems," *Journal of Global Optimization* 3, 79–94.

[165] PHILLIPS A.T. et ROSEN J.B.(1994), "Computational Comparison of Two Methods for Constrained Global Optimization," *Journal of Global Optimization* 5, 325–332.

[166] PHILLIPS A.T., ROSEN J.B. et VAN VLIET M.(1994), "A Parallel Stochastic Method for Solving Linearly Constrained Concave Global Minimization Problems," *Journal of Global Optimization* 2, 243–258.

[167] PHONG T.Q., TAO P.D. et HOAI AN L.T.(1995), "A Method for Solving D.C. Programming Problems. Application to Fuel Mixture Nonconvex Optimization Problem," *Journal of Global Optimization* 6, 87–105.

[168] PINE S.H.(1987), *Organic Chemistry* McGraw-Hill, New York London.

[169] QUESADA I. et GROSSMANN I.E.(1995), "A Global Optimization Algorithm for Linear Fractional and Bilinear Programs," *Journal of Global Optimization* 6, 39–76.

[170] RAGHAVAN T.E.S.(1970), "Completely Mixed Strategies in Bimatrix Games," *The Journal of the London Mathematical Society* 2, 709–712.

[171] REEVES G.R.(1975), "Global Minimization in Nonconvex All-Quadratic Programming," *Management Science* Vol.22 No.1, 76–86.

[172] REINHARDT K.(1922), "Extremale Polygone gegebenen Durchmessers ," *Jber dtsch. Math.-Ver.* 31, 251-270.

[173] ROBINSON J.(1951), "An Iterative Method of Solving a Game," *Annals of Mathematics* 54, 296–301.

[174] STANFORD W.(1996), "The Limit Distribution of Pure Strategy Nash Equilibria in Symmetric Bimatrix Games," *Mathematics of Operations Research* Vol.21 No.3, 726–733.

[175] SAVARD G.(1989), "Contributions à la Programmation Mathématique à Deux Niveaux," Thèse de Doctorat, École Polytechnique de Montréal.

[176] SAVARD G. et GAUVIN J.(1994), "The Steepest Descent Direction for the Nonlinear Bilevel Programming Problem," *Operations Research Letters* 15, 265–272.

[177] SELTEN R.(1975), "Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games," *International Journal of Game Theory* Vol.4 No.1, 25–55.

[178] SHAPLEY L.S.(1974), "A Note on the Lemke-Howson Algorithm," *Mathematical Programming Study* 1, 74–189.

[179] SHERALI H.D. et ADAMS W.P.(1990), "A Hierarchy of Relaxations between the Continuous and Convex Hull representations for Zero-One Programming Problems," *SIAM Journal on Discrete Mathematics* Vol.3 No.3, 411–430.

[180] SHERALI H.D. et ADAMS W.P.(1994), "A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems," *Discrete Applied Mathematics* 52, 83–106.

[181] SHERALI H.D. et ALAMEDDINE A.(1990), "An Explicit Characterization of the Convex Envelope of a Bivariate Bilinear Function over Special Polytopes," *Annals of Operations Research* 25, 197–210.

[182] SHERALI H.D. et ALAMEDDINE A.(1992), "A New Reformulation-Linearization Technique for Bilinear Programming Problems," *Journal of Global Optimization* 2, 379–410.

[183] SHERALI H.D., KRISHNAMURTHY R.S. et AL-KHAYYAL F.A.(1994), "A Parametric Concave Programming Approach for Linear Complementary Problems," Research Report, Georgia Institute of Technology, School of Industrial and Systems Enfineering.

[184] SHERALI H.D. et SHETTY C.M.(1980), "A Finitely Convergent Algorithm for Bilinear Programming Problem using Polar Cuts and Disjunctive Face Cuts," *Mathematical Programming* 19, 14–31.

[185] SHERALI H.D. et TUNCBILEK C.H.(1992), "A Global Optimization Algorithm for Polynomial Programming Using a Reformulation-Linearization Technique," *Journal of Global Optimization* 2, 101–112.

[186] SHERALI H.D. et TUNCBILEK C.H.(1995), "A Reformulation-Convexification Approach for Solving Nonconvex Quadratic Programming Problems," *Journal of Global Optimization* 7, 1–31.

[187] SHERALI H.D. et TUNCBILEK C.H.(1997), "Comparison of Two Reformulation-Linearization Technique Based Linear Programming Relaxations for Polynomial Programming Problems," *Journal of Global Optimization* 10, 381–390.

[188] SIMOES L.M.C.(1987), "Search for the Global Optimum of Least Volume Trusses," *Engineering Optimization* 11, 49–63.

[189] SOLAND R.M.(1971), "An Algorithm for Separable Nonconvex Programming Problems II: Nonconvex Constraints," *Management Science* Vol.17 No.11, 759–773.

[190] THIEU T.V.(1988), "A Note on the Solution of Bilinear Problems by Reduction to Concave Minimization," *Mathematical Programming* 41, 249–260.

[191] TODD M.J.(1976), "Coments on a Note by Aggarwal," *Mathematical Programming* 10, 130–133.

[192] TODD M.J.(1978), "Bimatrix Games–An Appendum," *Mathematical Programming* 14, 112–115.

[193] TOMLIN J.A.(1971), "An Improved Branch-and-Bound Method for Integer Programming," *Operations Research* 19, 1070–1075.

[194] TUY H.(1964), "Concave Programming under Linear Constraints," *Doklady Academii Nauk SSSR* 159, 32–35 [version anglaise :*Soviet Mathematics* 5, 1437–1440].

[195] TUY H., MIGDALAS A. et VÄRBRAND P.(1993), "A Global Optimization Approach for the Linear Two-Level Program," *Journal of Global Optimization* 3, 1–23.

[196] TUY H., MIGDALAS A. et VÄRBRAND P.(1994), "A Quasiconcave Minimization Method for Solving Linear Two-Level Programs," *Journal of Global Optimization* 4, 243–263.

[197] VAISH H. et SHETTY C.M.(1976), "The Bilinear Programming Problem," *Naval Research Logistics Quarterly* 23, 303–309.

[198] VAISH H. et SHETTY C.M.(1977), "A Cutting Plane Algorithm for the Bilinear Programming Problem," *Naval Research Logistics Quarterly* 24, 83–94.

[199] VAN DAMME E. (1992), "Refinements of Nash Equilibrium," dans *Advances in Economic Theory* (éditeur LAFFONT J.J.), Cambridge University Press, 32–75.

[200] VAN DAMME E. (1996), *Stability and Perfection of Nash Equilibria, Second, Revised and Enlarged Edition* Springer-Verlag Berlin New-York.

[201] VAN DAMME E. et WEIBULL J.W.(1995), "Equilibrium in Strategic Interaction: The Contributions of John C.Harsanyi, John F.Nash and Reinhard Selten," *Scandinavian Journal of Economics* Vol.97, No.1, 15–40.

[202] VICENTE L.N. et CALAMAI P.H.(1994), "Bilevel and Multilevel Programming: a Bibliography Review," *Journal of Global Optimization* Vol.5 No.3, 291–306.

[203] VICENTE L.N., CALAMAI P.H. et JÚDICE J.J.(1992), "Generation of Disjointly Constrained Bilinear Programming Test Problems," *Computational Optimization and Applications* 13, 299–306.

[204] VICENTE L.N., SAVARD G. et JÚDICE J.J.(1992), "Descent Approaches for Quadratic Bilevel Programming," *Les Cahiers du GERAD* G-92-36, Montréal.

[205] VICENTE L.N., SAVARD G. et JÚDICE J.J.(1996), "Discrete Linear Bilevel Programming Problem," *Journal of Optimization Theory and Application* Vol.89 No.3.

[206] VISWESWARAN V. et FLOUDAS C.A.(1990), "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs-II. Applications of Theory and Test Problems," *Computers and Chemical Engineering* Vol.14 No.12, 1419–1434.

[207] VISWESWARAN V., FLOUDAS C.A., IERAPETRITOU M.G. et PISTIKOPOULOS E.N.(1996), *A Decomposition-Based Global Optimization Approach for Solving Bilevel Linear and Quadratic Programs* Kluwer Academic Publishers, Boston.

[208] VON NEUMANN J.(1928), "Zur Theorie des Gesellschaftsspiele," *Mathematische Annalen* 100, 295–320 [version anglaise: Bargmann S.(1959) "On the Theory of Games and Strategy," dans *Contributions to the Theory of Games* (éditeurs LUCE R.D. et TUCKER A.W.), 4, 13–42].

[209] VON STACKELBERG (1952), *The Theory of Market Economy,* Oxford University Press, Oxford, England.

[210] VOROB'EV N.N.(1958), "Equilibrium Points in Bimatrix Games," *Theoriya Veroyatnostej i ee Primwneniya* 3, 318–331 [version anglaise: *Theory of Probability and its Applications* 3, 297–309].

[211] WEN U. et BIALAS W.(1986), "The Hybrid Algorithm for Solving the Three-Level Linear Programming Problem," *Computers & Operations Research* Vol.13 No.4, 367–377.

[212] WEN U.P. et YANG Y.H.(1990), "Algorithms for Solving the Mixed Integer Two-Level Linear Programming Problem," *Computers & Operations Research* Vol.17 No.2, 133–142.

[213] WENDELL R.E. et HURTER A.P.(1976), "Minimization of a Non-Separable Objective Function Subject to Disjoint Constraints," *Operations Research* Vol.24 No.4, 643–657.

[214] WHITE D.J.(1992), "A Linear Programming Approach to Solving Bilinear Programmes," *Mathematical Programming* 56, 45–50.

[215] WHITE D.J.(1997), "Penalty Function Approach to Linear Trilevel Programming," *Journal of Optimization Theory and Applications* Vol.93 No.1, 183–197.

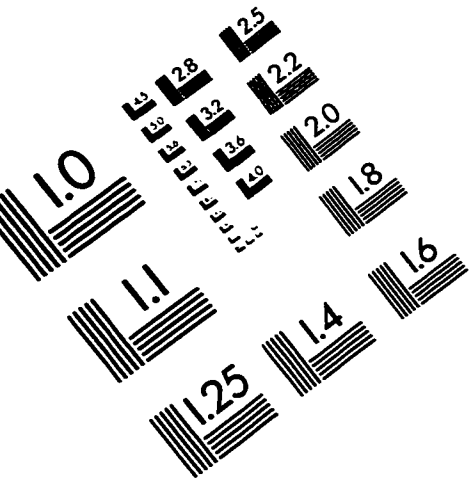
[216] WHITE D.J. et ANANDALINGAM G.(1993), "A Penalty Function Approach for Solving Bi-Level Linear Programs," *Journal of Global Optimization* 3, 397–419.

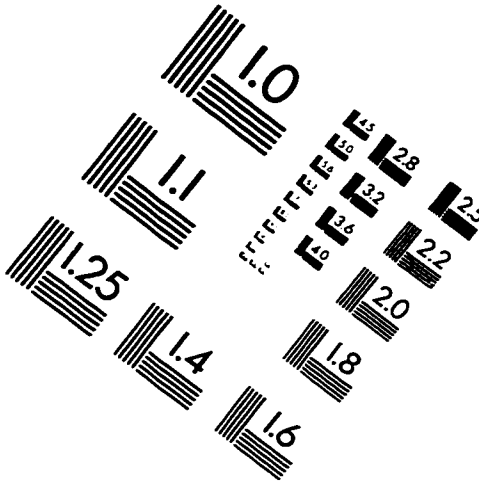
[217] WILSON R.(1972), "Computing Equilibria of Two-Person Games from the Extensive Form," *Management Science* Vol.18 No.7, 448–460.

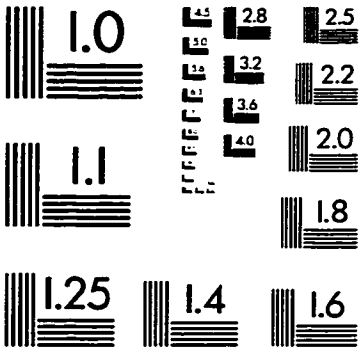
[218] WINKELS H.M.(1979), "An Algorithm to Determine all Equilibrium Points of a Bimatrix Game," in *Game Theory and Related Topics* (éditeurs MOESCHLIN O. et PALLASCHKE D.), North-Holland Publishing Company, Amsterdam, New-York, Oxford, 137–148.

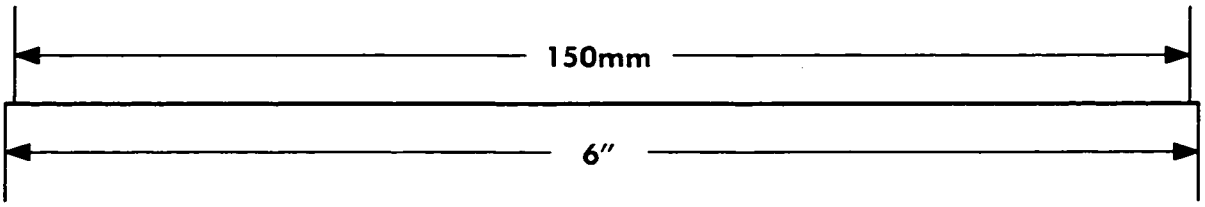
[219] WOLSEY L.A.(1981), "A Resource Decomposition Algorithm for General Mathematical Programs," *Mathematical Programming Study* 14, 244–257.

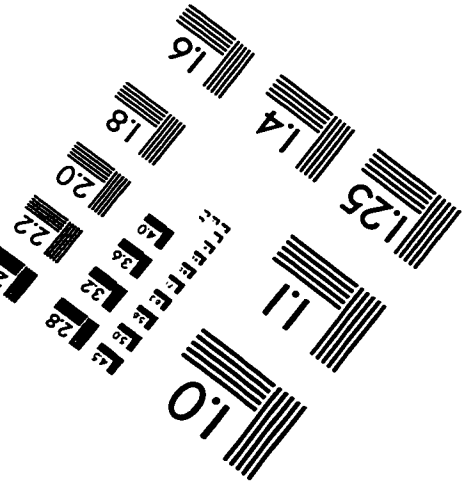
IMAGE EVALUATION
TEST TARGET (QA-3)

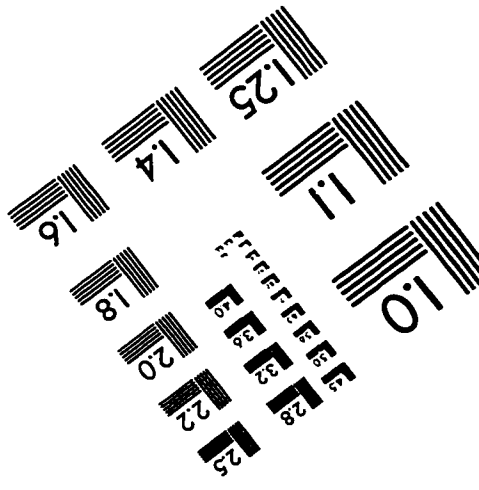












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989