UNIVERSITÉ DE MONTRÉAL

DÉPLOIEMENT ET MISE À JOUR DE COUPES DE CONCAVITÉ

MAIKEL GEAGEA
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
AVRIL 2014

© Maikel Geagea, 2014.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DÉPLOIEMENT ET MISE À JOUR DE COUPES DE CONCAVITÉ

présenté par : GEAGEA Maikel
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. ANJOS Miguel, Ph.D., président
M. AUDET Charles, Ph.D., membre et directeur de recherche
M. ALARIE Stéphane, M.Sc.A., membre

*À mes parents, à ma famille, à **mes** patries,*
*à **tous** mes amis,*
*et **en qui** je crois*
je vous offre les fruits de ce travail.

REMERCIEMENTS

J'aimerais remercier mon directeur de recherche, Charles Audet, pour le soutien et la confiance qu'il m'a procurés tout au long de mes recherches. Grâce à lui, je comprends mieux l'importance des études théoriques et de leurs applications pour l'évolution et l'aiguisement de nos valeurs de pensées. Son sens de la pédagogie et ses directives pertinentes n'ont cessé d'aiguiser mon sens pour la recherche.
Charles ! Avec une telle personne respectueuse, une maîtrise se sauve et une recherche s'accomplit. Ton sens de l'écoute et tes mots de persévérence m'ont enlevé tous les doutes concernant mes compétences et mes capacités.

Je tiens aussi à remercier tous mes collègues (en particulier mon collègue de bureau Bastien Talgorn), professeurs et personnels du *GERAD*. Dans une ambiance conviviale et avec un esprit de collaboration, j'ai eu la chance découvrir un monde scientifique merveilleux, multiculturel et riche en valeur humain.

Je suis reconnaissant à Sylvain Perron, Professeur à HEC, et François Soumis, Professeur à l'École Polytechnique, de m'avoir accueilli dans leur groupe sportif. Grâce à la pratique de leur activité sportive, j'ai dévéloppé un esprit de combativité et d'endurance.

Sans oublier de remercier les professeurs qui m'ont enseigné à l'école Polytechnique : Alain Hertz, Dominique Orban, Guy Desaulniers et Richard Labib. Leur sens objectif et leur passion de travail ont marqué pareillement ma personne.

Finalement, je souhaite remercier le Conseil de Recherche en Sciences Naturelles et en Génie (CRSNG) pour son aide financière durant mes études.

RÉSUMÉ

Dans le cadre de minimisation concave, un type de problème d'optimisation quadratique, nommé bilinéaire disjoint (BILD), peut se reformuler en deux problèmes linéaires symétriques MinMax (LMM). Comme les solutions optimales de BILD et de ses reformulations LMM sont liées par une simple bijection, la question de notre travail de recherche s'agit profiter de cette reformulation pour résoudre $BILD$.

Dans la littérature, une technique de calcul, appelée coupe de concavité, proposée par Tuy [39] en 1964, s'avère importante afin de résoudre les $BILD$. L'algorithme basé uniquement sur cette technique n'est pas sûrement de convergence finie. Cela est dû à plusieurs problèmes, parmi lesquels nous citons : le problème de dégénéréscene et le problème de cumul des coupes.

Depuis, des chercheurs tentent d'améliorer la convergence de l'algorithme des plans coupants. Pour ce faire, ils ont intégré cet algorithme avec d'autres techniques de calcul. En effet, Konno [23] a introduit la technique Mountain Climbing en 1971 pour évaluer à chaque itération une solution locale. Avec l'usage des pseudo-sommets, Marcus [34] a développé en 1999 des coupes similaires à celles de Tuy en décomposant le cône polyèdral. En 2001, Alarie et *al.* ont traité le problème de dégénérescence et ont exploité la technique de "Branch and Bound" pour les instances $BILD$ de grande dimension.

La recherche proposée dans ce mémoire se situe dans la continuation de ces travaux. Nous proposons deux nouvelles stratégies pour la génération de coupes de concavité. Dans la première, nous avons élaboré une mise à jour dynamique des coupes après qu'une amélioration de la valeur objectif soit faite. Dans la seconde, au lieu que les coupes soient associées à des sommets, nous les avons associées aux pseudo-sommets.

Ces deux nouvelles stratégies sont testées numériquement sur un ensemble de problèmes tests tirés de la littérature ainsi que sur une collection de problèmes générés aléatoirement.

ABSTRACT

In the context of concave minimisation, a type of quadratic optimization problem, called $BILD$ problem, can be reformulated into two symmetrical linear MinMax problems LMM. As there is a simple bijection between the optimal solution of $BILD$ and their reformulations LMM, our research question is to take advantage of this reformulations for solving $BILD$.

In the litterature, a computation technique , called concavity cut, proposed by Tuy [39] in 1964, has been shown to be useful solving the $BILD$ problem. However, it is still unknown whether the finite convergence of a cutting planes algorithm can be enforced by the concavity cut itself or not. This is due to several problems, among which we mention : the degeneracy problem and the accumulation of cuts.

Since then, researchers have attempted to improve the convergence of the cutting planes algorithm. To achieve this, they have integrated the algorithm with other computation techniques. Indeed, Konno [23] introduced in 1971 the mountain climbing (MC) technique to evaluate in each iteration a local optimal solution. In 1999, Marcus [34] used the pseudo-vertices to developed similar Tuy cuts by decomposing the polyhedral cone. Alarie and *al.* treated in 2001 the degeneracy problem ant they have exploited the "Branch and Bound" technique for solvign $BILD$ instances with large dimensions.

The proposed research in this project is the continuation of this works. Thus, we proposed two ways for deploying cuts. In the first one, we dynamically update previously generated cuts after an improvement of the objective function value. In the second, instead rooting the cuts at vertices, we root them at pseudo-vertices.

These two new strategies are tested numerically on a set of test problems issued from the literature as well as on a collection of randomly generated instances.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

BILD	Problème BILinéaire Disjoint
LMM	Problème Linéaire MinMax
PO	Problèmes d'Optimisation
PL	Problèmes Linéaire
MC	Mountain Climbing
MN1	Modèle Numérique 1
MN2	Modèle Numérique 2
MN3	Modèle Numérique 3
CC	Algorithme de Coupe de Concavité
CCU	Algorithme CC de Mise à Jour
CCP	Algorithme CC de Pseudo-sommet
CCUP	Algorithmes CCU et CCP combinés

LISTE DES ANNEXES

CHAPITRE 1

INTRODUCTION

Selon leur type, les problèmes d'optimisation (PO) sont répartis dans des classes pour distinguer leur structure. Afin de parler de ceux qui nous intéressent ($BILD$ et LMM), nous commençons ce chapitre par une série de définitions mathématiques. Suite aux descriptions des problèmes $BILD$ et LMM, nous présentons la structure de notre mémoire.

1.1 Minimisation concave et maximisation convexe

Nous supposons que le lecteur soit familier avec les concepts de base d'analyse convexe tels que l'espace $\mathbb{R}^n$, les hyperplans, les facettes, les arêtes, etc ... Sinon, ces concepts se trouvent dans la littérature ([25][30]).

Définition 1. *Un sous-ensemble P de $\mathbb{R}^n$ délimité par un nombre fini d'hyperplans est un polyèdre. Son expression mathématique est comme suit :*

$$P \;=\; \{x \in \mathbb{R}^n : Ax \leq a\} \text{ où } A \in \mathbb{R}^{m\times n} \text{ et } a \in \mathbb{R}^m.$$

Un polytope est un polyèdre borné.

Définition 2. *Un ensemble P est convexe si le segment joignant deux points quelconques de P est contenu entièrement dans P. Formellement, cela s'écrit :*

$$\text{pour tout } x, y \text{ de } P \text{ et tout } t \in]0,1[, \text{ on a : } tx + (1-t)y \in P.$$

Définition 3. *Soient P un ensemble convexe et φ une fonction réelle tel que $\varphi : P \to \mathbb{R}$.*

a. *φ est dite concave sur P si :*
pout tout x, y de P et tout $t \in]0,1[$, on a : $\varphi(tx + (1-t)y) \geq t\varphi(x) + (1-t)\varphi(y)$.

b. *φ est dite convexe sur P si $-\varphi$ est concave sur P.*

Définition 4. *Soient P un polytope et $\varphi : P \to \mathbb{R}$ une fonction concave (resp. convexe). Le PO suivant :*

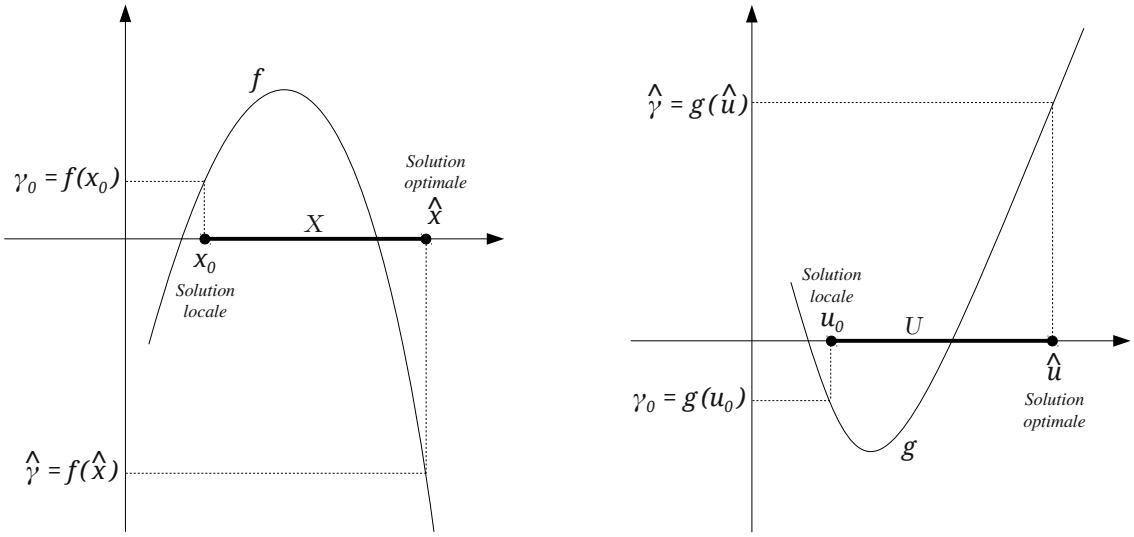
$$\min_{x\in P} \varphi(x) \quad \left(\textit{resp.} \quad \max_{x\in P} \varphi(x) \right)$$

est alors un problème de minimisation concave (resp. problème de maximisation convexe).

Minimiser une fonction concave est équivalent à maximiser une fonction convexe. Afin d'illustrer la difficulté liée à ces problèmes, considérons deux fonctions f et g, respectivement concave et convexe, et analysons leur PO :

$$\min_{x \in X} f(x)\,; \quad \max_{u \in U} g(u).$$

Cette notation signifie que nous cherchons un terme x dans X (resp. u dans U) qui donne la plus petite (resp. grande) valeur de f (resp. de g). Ce terme est appelé solution optimale dont la valeur est dite optimale. La figure 1.1 illustre cette situation avec la dimension 1, où le domaine est un intervalle fermé et borné.

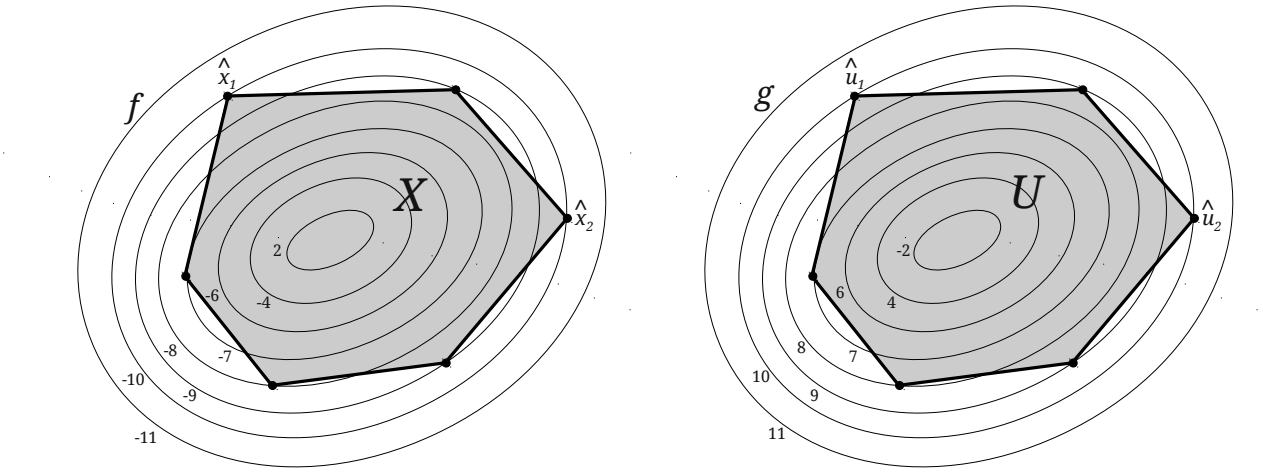


(a) Fonction concave avec $X \subseteq \mathbb{R}$ (b) Fonction convexe avec $U \subseteq \mathbb{R}$

Figure 1.1 Polytopes de dimension 1

Comme le montre ces figures, la fonction f concave (resp. g convexe), atteint la valeur optimale avec la solution optimale $\hat{x}$ (resp. $\hat{u}$). Aussi, le PO en f (resp. en g) respectivement comporte x_0 (resp. u_0) comme solution localement optimale.

Il est facile de présenter graphiquement le processus d'optimisation d'une fonction dans un espace de dimension 1. Alors que dans un espace de dimension $n = 2$, nous traçons à la figure 1.2 les courbes de niveau $F_\gamma = \{x \in \mathbb{R}^n : \ f(x) = \gamma\}$ et $G_\gamma = \{u \in \mathbb{R}^n : \ g(u) = \gamma\}$ pour différentes valeurs de γ.



(a) Fonction concave avec $X \subseteq \mathbb{R}^2$ (b) Fonction convexe avec $U \subseteq \mathbb{R}^2$

Figure 1.2 Polytopes de dimension 2

Ces figures montrent que la fonction f concave (resp. g convexe), atteint la valeur optimale aux solutions optimales $\hat{x}_1$ et $\hat{x}_2$ (resp. $\hat{u}_1$ et $\hat{u}_2$). Tous les autres sommets des polytopes représentent les solutions locales de f et g.

Dans ce qui suit, nous considérons un polytope $P \subseteq \mathbb{R}^n$, une fonction réelle φ sur P et une valeur donnée $\gamma \in \mathbb{R}$.

Théorème 1. *Si la fonction φ est concave, et si elle est bornée inférieurement sur P, alors il existe un sommet de P qui minimise φ sur P.*

Théorème 2. *Si la fonction φ est concave (resp. convexe), alors l'ensemble*

$$C_\gamma = \{x \in \mathbb{R}^n : \ \varphi(x) \geq \gamma\} \ \textit{(resp. } C_\gamma = \{x \in \mathbb{R}^n : \ \varphi(x) \leq \gamma\}\textit{)}$$

est convexe.

Le PO que nous considérons est

$$\min_{x \in P} \varphi(x)$$

où $\varphi : \mathbb{R}^n \to \mathbb{R}$ est la fonction objectif concave que l'on souhaite minimiser, $P \subset \mathbb{R}^n$ est le domaine réalisable. On adoptera la notation du tableau 1.1.

Tableau 1.1 Notation et terminologie

Notations	Définitions
φ	Fonction objectif ou coût à optimiser
P	Domaine réalisable du PO
$x \in P$	x : Solution réalisable
$x \notin P$	x : Solution non-réalisable
$\varphi(x^*) = \min_{x \in P} \varphi(x)$	x^* : Solution optimale de φ de valeur $\varphi(x^*)$
$\varphi(x^*) \leq \varphi(x)$ pour tout $x \in P \cap B_\varepsilon(x^*)$	x^* : Solution locale de φ de valeur $\varphi(x^*)$
$\min_{x \in P} \varphi(x) = -\infty$	PO non-borné
$P = \emptyset$ ou $\min_{x \in P} \varphi(x) = +\infty$	PO non-réalisable

Dans notre cadre de recherche, nous cherchons à minimiser un problème bilinéaire disjoint qu'on note $BILD$, tout en profitant de sa reformulation en deux PO concaves [32][20], notés par LMM.

1.2 Problème bilinéaire disjoint ($BILD$)

Le problème $BILD$ fut introduit par Konno [23] en 1971 pour généraliser l'approche de Mills [29] (1960) afin de trouver un point d'équilibre de Nash [31] (1951) d'un jeu bimatriciel. Ce genre de 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 $BILD$ se formule comme suit :

$$(BILD) \quad \begin{array}{ll} \min\limits_{x,u} & c^t x - u^t Q x + u^t d \\ \text{s.c.} & Ax \geq a, \quad u^t B \geq b^t \\ & x \geq 0, \qquad u \geq 0 \end{array} \quad = \quad \min_{x \in X, u \in U} h(x,u) \tag{1.1}$$

où

$$h(x,u) = c^t x - u^t Q x + u^t d$$

et

$$x,\ c \in \mathbb{R}^{n_x} \ ; \quad a \in \mathbb{R}^{n_v} \ ; \quad A \in \mathbb{R}^{n_v \times n_x} \ ;$$
$$u,\ d \in \mathbb{R}^{n_u} \ ; \quad b \in \mathbb{R}^{n_y} \ ; \quad B \in \mathbb{R}^{n_u \times n_y} \ ; \quad Q \in \mathbb{R}^{n_u \times n_x} \ .$$
$$X = \{x \in \mathbb{R}^{n_x} : Ax \geq a, x \geq 0\} \ ; \quad U = \{u \in \mathbb{R}^{n_u} : u^t B \geq b^t, u \geq 0\}.$$

1.3 Problème linéaire MinMax (LMM)

Les problèmes LMM sont la combinaison de deux problèmes linéaire (PL) qui incarnent respectivement deux classes de décisions diamétralement opposées. La première classe comprend la manipulation des variables de décision avec des intérêts minimales que la seconde va en profiter pour générer son intérêt maximal. L'une de leurs formulations est proposée pour la première fois par Falk [16], plus tard par Lutzenko et Martynov [26] et se formule comme suit :

$$(LMM) \quad \min_{x\in X}\left(c^t x + \begin{pmatrix} \max\limits_{y} & b^t y \\ \text{s.c.} & By \leq d - Qx \\ & y \geq 0 \end{pmatrix}\right) \quad = \quad \min_{x\in X}\left(c^t x + \begin{pmatrix} \max\limits_{y} & b^t y \\ \text{s.c.} & y \in Y(x) \end{pmatrix}\right) \tag{1.2}$$

où

$$X = \{x \in \mathbb{R}^{n_x} : Ax \geq a,\ x \geq 0\} \subseteq \mathbb{R}^{n_x}; \quad Y(x) = \{y \in \mathbb{R}^{n_y} : By \leq d - Qx,\ y \geq 0\} \subseteq \mathbb{R}^{n_y};$$
$$x, c \in \mathbb{R}^{n_x}; \quad a \in \mathbb{R}^{n_v}; \quad A \in \mathbb{R}^{n_v \times n_x}; \quad y, b \in \mathbb{R}^{n_y}; \quad d \in \mathbb{R}^{n_u}; \quad B \in \mathbb{R}^{n_u \times n_y}; \quad Q \in \mathbb{R}^{n_u \times n_x}.$$

Ainsi le premier niveau doit choisir la variable x de X qui lorsque combinée avec la réaction de la variable y de $Y(x)$ du second niveau, minimise la fonction objectif.

1.4 Structure du mémoire

Ce mémoire est structuré de la façon suivante.

Le chapitre deux comprend les outils mathématiques en lien avec la reformulation des $BILD$ et l'introduction des coupes de concavité. Ces outils sont suivis, dans le même chapitre, par une revue de la littérature et une discussion portant sur les problèmes de calcul qui peuvent subsister.

Au chapitre trois, la rencontre pratique des travaux vus en littérature se fût décrit avec des choix de modèles numériques et des conceptions de leurs algorithmes de résolution.

En proposant des améliorations dans le déploiement des coupes, un retour concret à nos objectifs sera fait au chapitre quatre. Ce dernier comprend un algorithme général englobant l'ensemble de nos travaux.

Le chapitre suivant présente les résultats numériques sur un ensemble de problèmes tests tirés de la littérature ainsi que sur une collection de problèmes générés aléatoirement.

Et finalement, le dernier chapitre décrit la synthèse et les perspectives de nos travaux.

CHAPITRE 2

OUTILS EN OPTIMISATION CONCAVE ET REVUE DE LITTÉRATURE

Pour focaliser notre recherche sur les exigences de nos objectifs scientifiques, nous procédons à la première section à une revue des outils mathématiques de base, puis à la deuxième, des liens existant entre BILD et ses reformulations. Par conséquent, à la troisième section, une revue de littérature portant sur trois axes problèmatiques va être élaborée.

2.1 Définitions et concepts de base

Cette section présente une série de définitions mathématiques objectifs et des concepts de base pour générer une coupe de concavité. L'ensemble de ces outils constituent une base essentielle pour lancer notre recherche.

2.1.1 Sommets, pseudo-sommets voisins et dégénérescence

La résolution d'un *PO* concave s'appuie sur la recherche, parmi les sommets d'un polytope, de celui dont la valeur objectif est optimale. Les notions de sommets, dégénérés ou non, d'hyperplans, de voisinage et de pseudo-sommets sont essentielles à cette étude.

Définition 5. *Un pseudo-sommet x d'un polytope P dans un espace de dimension n est un point x à l'intersection n hyperplans distincts contenant chacun une facette de P. Si $x \in P$, alors il s'agit d'un sommet.*

Définition 6. *Deux sommets ou pseudo-sommets distincts partageant $n-1$ hyperplans contenant chacun une facette de P sont dits voisins.*

Définition 7. *Dans un espace de dimension n, si **exactement** n facettes d'un polytope se rencontrent en un seul point, alors elles déterminent un point non dégénéré. S'il y a plus de n facettes qui s'y rencontrent, alors ce point est un point dégénéré.*

Nous commençons par illustrer, dans un espace de dimension 2, les sommets et les pseudo-sommets voisins. Ensuite, dans un espace de dimension 3, nous présenterons un exemple de dégénérescence.

En examinant la figure 2.1, nous pouvons tirer les observations suivantes :

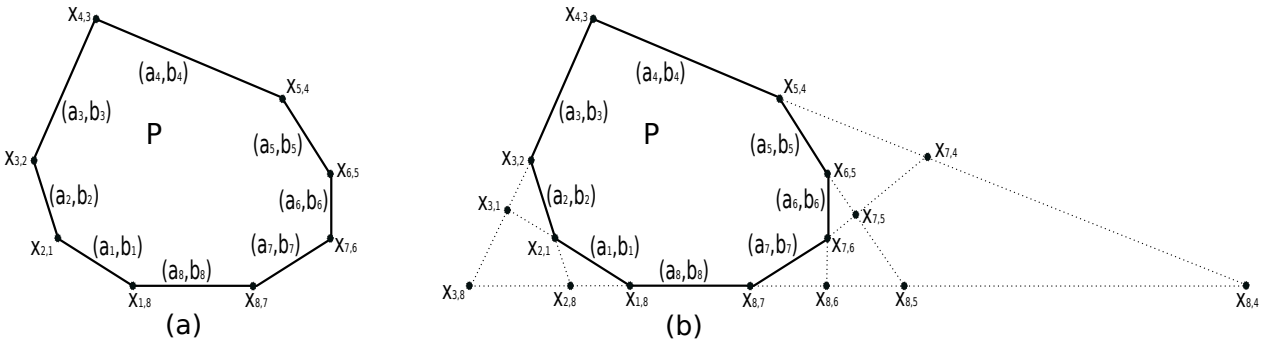


Figure 2.1 Sommets et pseudo-sommets

1. Le polytope P peut s'écrire sous la forme

$$P = \{x \in \mathbb{R}^2 : a_1^t x \leq b_1, a_2^t x \leq b_2, ..., a_8^t x \leq b_8\}$$

   dans laquelle $a_i^t x = b_i$ est l'équation de la i^{ème} facette du polytope P.

2. **Tous** les sommets et **certains** pseudo-sommets du polytope P sont illustrés par des points dans la Figure 2.1. Ils représentent les solution $x_{i,j}$ des équations matricielles $\begin{pmatrix} a_j^t \\ a_i^t \end{pmatrix} x = \begin{pmatrix} b_j \\ b_i \end{pmatrix}$.
3. Les sommets ou les pseudo-sommets liés par une arête, pointillée ou non, sont voisins.

En plus, nous distinguons dans la Figure 2.1 la notion d'un sommet voisin de celle d'un pseudo-sommet voisin comme suit.

- D'abord, à titre d'exemple, nous souhaitons citer les voisins de $x_{1,8}$ (sommets et pseudo-sommets) qui se trouvent sur la 8^{e} facette du polytope P.
- L'ensemble de sommets voisins de $x_{1,8}$ est le singleton $\{x_{8,7}\}$ (Figure 2.1(a)) alors que l'ensemble des pseudo-sommets voisins du même sommet est $\{x_{2,8}, x_{3,8}, x_{8,7}, x_{8,6}, x_{8,5}, x_{8,4}\}$ (Figure 2.1(b)). Remarquons bien que les indices du $x_{1,8}$ et ceux des ses voisins (sommets et pseudo-sommets) diffèrent d'un seul chiffre.
- Enfin, comme les indices d'un point font rapports avec les facettes qui le déterminent, nous constatons que les systèmes d'équations des sommets ou des pseudo-sommets voisins diffèrent d'une seul ligne. Par exemple, il est facile de remarquer que $x_{1,8}$ et $x_{8,4}$ sont les solutions des systèmes d'équation $\begin{pmatrix} a_1^t \\ a_8^t \end{pmatrix} x = \begin{pmatrix} b_1 \\ b_8 \end{pmatrix}$ et $\begin{pmatrix} a_4^t \\ a_8^t \end{pmatrix} x = \begin{pmatrix} b_4 \\ b_8 \end{pmatrix}$ respectivement, qui diffèrent d'une seule ligne.

Pour illustrer graphiquement la notion de la dégénérescence dans un espace de dimension 3, nous nous servons de la figure 2.2.

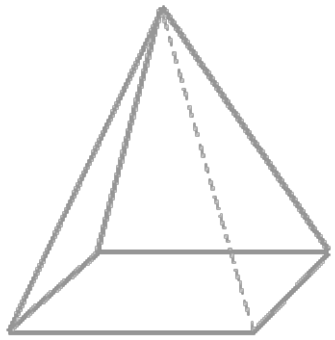


Figure 2.2 Polytope de dimension 3

Les quatre sommets à la base de la pyramide sont à l'intersection de trois facettes, et sont donc non dégénérés. Or, le sommet au haut de la pyramide est à l'intersection de quatre facettes et est dit dégénéré.

Considérons maintenant et formellement un polytope $P = \{x \in \mathbb{R}^n : Ax \leq a\}$ avec $A \in \mathbb{R}^{m \times n}$ et $a \in \mathbb{R}^m$ dans un espace de dimension $n \geq 2$.

Définition 8. *Un sommet x_0 du polytope P est une facette de dimension 0. C'est à dire, il existe une $(n, n+1)$-sous-matrice $(A_0,\, b_0)$ de la $(m, n+1)$-matrice $(A,\, b)$ qui vérifie les deux conditions suivantes :*

Condition 1. $Ax_0 \leq a$,

Condition 2. $A_0 x_0 = a_0$, i.e. $x_0 = A_0^{-1} b_0$.

Si la première condition est négligée, i.e. *$Ax_0 \neq a$, alors le sommet x_0 est dit pseudo-sommet. Ainsi, un pseudo-sommet devient sommet s'il appartient au polytope P.*
De plus, s'il existe une seule matrice inversible A_0 qui détermine x_0, alors x_0 est non dégénéré. Autrement, il est dit dégénéré.

Définition 9. *Deux sommets ou pseudo-sommets x_1 et x_2 du polytope P sont voisins s'il existe des $(n, n+1)$-sous-matrices $(A_1,\, a_1)$ et $(A_2,\, a_2)$ de la $(m, n+1)$-matrice $(A,\, a)$ qui vérifient les deux conditions suivantes :*

Condition 1. *x_1 et x_2 sont les solutions des systèmes d'équations $A_1 x = a_1$ et $A_2 x = a_2$,*

Condition 2. *les sous-matrices $(A_1,\, a_1)$ et (A_2, a_2) diffèrent d'une seule ligne.*

2.1.2 Coupes de concavité

Les coupes de concavité furent introduites pour la première fois par Tuy [39] en 1964, puis développées par Glover [17] en 1973, pour résoudre les problèmes de minimisation concaves. Pour détailler cette technique, nous présentons les éléments de base, les étapes de sa génération, et enfin son domaine d'application.

Éléments d'introduction d'une coupe

Nous définissons dans un espace $\mathbb{R}^n$ les éléments de base servant à la construction et à la validation d'une coupe de concavité.

Définition 10. *Un cône polyhédral convexe de sommet $\hat{x}$ dans un espace $\mathbb{R}^n$ est un ensemble de la forme*

$$\hat{x} + Cone(\delta_1, ..., \delta_k) = \hat{x} + \left\{ \sum_{i=1}^{k} \alpha_i \delta^i : \alpha_i \geq 0 \right\} = \left\{ \hat{x} + \sum_{i=1}^{k} \alpha_i \delta^i : \alpha_i \geq 0 \right\},$$

où les δ^i sont des vecteurs incidents du sommet $\hat{x}$. On dit que le cône est engendré par $\delta_1, ..., \delta_k$.

Définition 11. *L'enveloppe convexe des vecteurs $\delta_1, ..., \delta_k$ est l'ensemble de la forme*

$$Conv(\delta_1, ..., \delta_k) = \left\{ \sum_{i=1}^{k} \alpha_i \delta_i : \alpha_i \geq 0, \sum_{i}^{k} \alpha_i = 1 \right\}.$$

L'ensemble $Conv(\delta_1, ..., \delta_k)$ est un polytope.

Définition 12. *Soit $\varphi : \mathbb{R}^n \to \mathbb{R}$ une fonction concave. La $\gamma-$extension (parfois non bornée) est le point le plus éloigné de $\hat{x}$ dans la direction δ^i pour lequel la valeur de la fonction φ dépasse γ.*
En d'autres termes, pour le sommet $\hat{x}$ du polytope P, la $\gamma-$extension dans la direction δ^i, $(i = 1, 2, ..., k)$ est le point $\hat{x} + \hat{\alpha}_i \delta^i \in \mathbb{R}^n$ où $\hat{\alpha}_i$ est la valeur optimale de

$$\begin{aligned} \max_{\alpha_i} \quad & \alpha_i \\ s.c. \quad & \varphi(\hat{x} + \alpha_i \delta^i) \geq \gamma \end{aligned}$$

Ainsi, la concavité de la fonction φ assure que tout point de l'enveloppe convexe de l'ensemble $\{\hat{x}, \hat{x} + \hat{\alpha}_1 \delta^1, \hat{x} + \hat{\alpha}_2 \delta^2 ..., \hat{x} + \hat{\alpha}_k \delta^k\}$ possède une valeur de la fonction φ supérieure ou égale à γ (Horst et Tuy [21]).

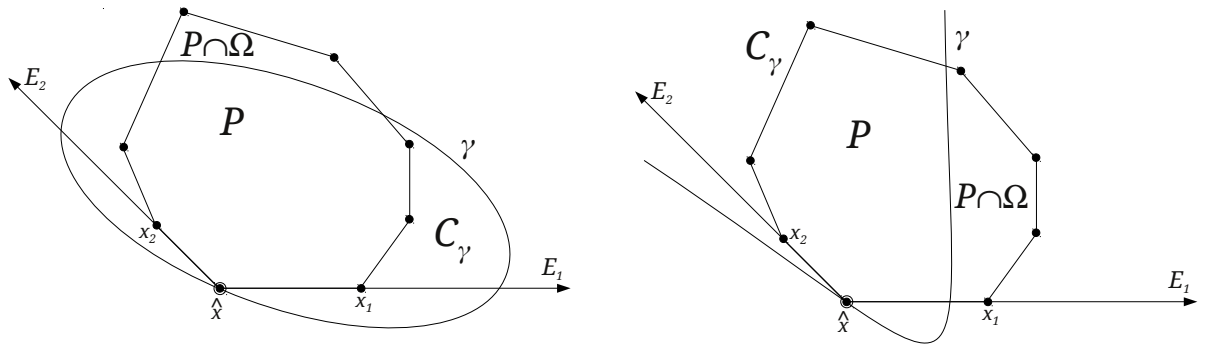
Définition 13. *Soit $\Omega \subseteq \mathbb{R}^n$ un ensemble et P un polytope. Une (P, Ω)-coupe est valide s'elle réduit le polytope P sans éliminer aucun point de $P \cap \Omega$.*

Génération d'une coupe de concavité

En minimisation concave, l'introduction d'une coupe de concavité consiste à retrancher une partie d'un polytope où la fonction objectif est plus grande ou égale à une valeur donnée. Pour ce faire, nous procédons d'abord à construire un cône polyhédral. Ensuite, pour introduire ce genre de coupe, nous évaluons les γ-extensions dans les directions de ce cône. Enfin, après la détermination de cette coupe, nous vérifions sa validité.

Étant donné une valeur $\gamma \in \mathbb{R}$, un ensemble convexe $C_\gamma = \{x \in \mathbb{R}^n : \varphi(x) \geq \gamma\}$ tel que $\varphi : \mathbb{R}^n \to \mathbb{R}$ est une fonction concave, un ensemble $\Omega = \{x \in \mathbb{R}^n : \varphi < \gamma\}$ complémentaire de C_γ, un polytope $P = \{x \in \mathbb{R}^n : Ax \leq a\}$ tels que $A \in \mathbb{R}^{m \times n}$, $a \in \mathbb{R}^m$ avec $m > n$, et un sommet non dégénéré $\hat{x}$ de P, nous construisons le cône polyédral associé à $\hat{x}$ avec les directions δ_i, $i = 1, 2, ..., n$ qui sont celles des sommets voisins de $\hat{x}$.

Comme exemple dans un espace de dimension 2, les figures 2.3a et 2.3b présentent un même cône polyédral issu du sommet $\hat{x}$ pour deux fonctions concaves différentes. La figure illustre aussi la courbe de niveau associée à la valeur γ.



(a) C_γ : Ensemble convexe fini

(b) C_γ : Ensemble convexe infini

Figure 2.3 Cône polyédral

Si $\hat{x}$ est un sommet polyhédral non dégénéré, alors il existe une seule $(n, n+1)$-sous-matrice $(A_0,\, a_0)$ de la matrice (m, n)-$(A,\, a)$ telle que $A_0\hat{x} = a_0$, *i.e.* $\hat{x} = A_0^{-1}a_0$. Graphiquement, cela s'interprête qu'il existe exactement n sommets voisins x_i de $\hat{x}$ dans le polytope P. Ainsi, ces sommets voisins constituent avec $\hat{x}$ des directions $\delta_i = x_i - \hat{x}$ (linéairement indépendantes) qui déterminent le cône polyédral relatif à $\hat{x}$ et à P. Dans la suite, ce cône s'écrit sous la forme $C\,(\hat{x}) = \hat{x} + Cone\,(\delta_1, ..., \delta_n)$.

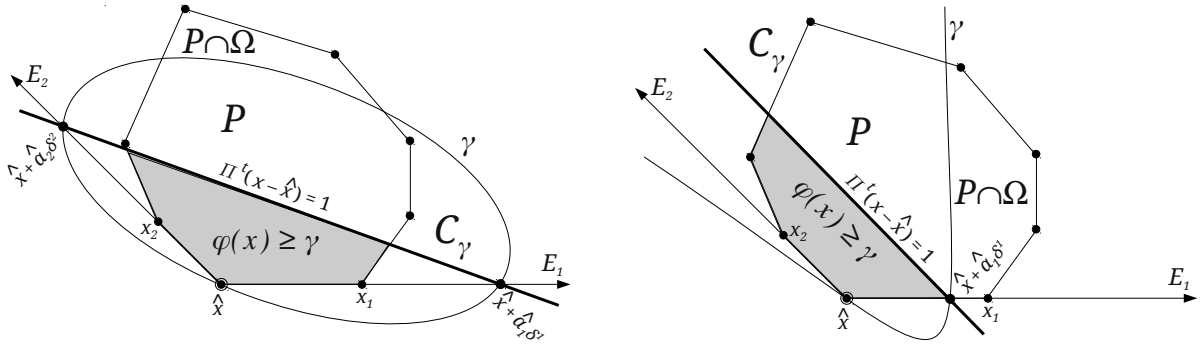
Après la construction du cône, nous évaluons les γ-extensions $\hat{x} + \hat{\alpha}_i\delta_i$ qui devraient être

les points d'intersection de l'arête $E_i = \{y_i(\alpha_i) = \hat{x} + \alpha_i\delta_i : \alpha_i \in \mathbb{R}_+\}$ et de la frontière $fr\,(C_\gamma)$. Si un tel point existe, alors $\hat{\alpha}_i < \infty$. Sinon, *i.e.* $E_i \subseteq C_\gamma$, on pose $\hat{\alpha}_i = +\infty$. Ainsi, une coupe passant par les γ-extensions finies peut être introduite. Elle croise les arêtes E_i lorsque $\hat{\alpha}_i < \infty$ et est parallèle à E_i sinon. Son équation est de la forme $\pi^t(x - \hat{x}) = 1$ où

$$\pi^t = (\frac{1}{\hat{\alpha_1}}, \frac{1}{\hat{\alpha_2}}, ..., \frac{1}{\hat{\alpha_n}})Q^{-1}$$

avec $Q = (\delta_1, \delta_2, ..., \delta_n)$ et avec la convention que $\frac{1}{\infty} = 0$.

Dans le même exemple que celui de la figure 2.3, des coupes sont introduites après évaluation des γ-extensions. L'une d'elles (figure 2.4a) rencontre les arêtes E_1 et E_2, l'autre (figure 2.4b) rencontre l'arête E_1 et est parallèle à l'arête E_2.



(a) γ-extensions finies (b) Une des γ-extensions est infinie

Figure 2.4 Coupes de concavités

Les parties hachurées des figures en dessus sont retranchée du polytope P. Elles comprennent des termes x pour lesquels la fonction objectif est plus grande ou égale à la valeur γ donnée. Le polytope P est ainsi réduit.

Comme φ est une fonction concave, la coupe $\pi^t(x - \hat{x}) \geq 1$ élimine seulement des points du cône $C\,(\hat{x})$ qui appartiennent à C_γ. En effet, en considérant l'ensemble $P \cap \Omega = \{x \in P : \varphi(x) < \gamma\}$, nous constatons que $(P \cap \Omega) \subseteq C(\hat{x})$ et que $C_\gamma \cap (P \cap \Omega) = \emptyset$. Ainsi, $\pi^t(x - \hat{x}) \geq 1$ élimine $\hat{x}$ sans éliminer aucun point de $P \cap \Omega$, *i.e.* $\pi^t(x - \hat{x}) \geq 1$ est une (P, Ω)-coupe, dite coupe de concavité ou coupe de Tuy [39].

Domaine d'application

L'usage des coupes de concavité peut être utilisé en pré-traitement ou bien jumelé à plusieurs algorithmes : algorithme de Branch and Bound [4], algorithme conique de plan coupant [28], etc... Aussi, elles étaient rencontrées dans d'autres problèmes d'optimisation : minimisation concave [12], programmation convexe-inverse [22], programmation en nombres entiers [27], programmation et optimisation lipschitzienne bilinéaire [9].

2.2 Résolution de $BILD$

Dans [7][37][6], les auteurs exploitent la structure particulière du problème $BILD$ pour trouver une solution optimale. En effet, ils ont d'abord reformulé $BILD$ en LMM concaves. Ensuite, ils ont exhibé les liens entre leurs solutions locales et optimales. Suite à ces résultats, ils ont enfin proposé une démarche de résolution. Cette chronologie est détaillée ci-dessous.

2.2.1 Passage de $BILD$ à LMM

Comme les ensembles X et U du problème $BILD$ (équ. 1.1) sont des polyèdres dans des espaces disjoints, alors $BILD$ est fondamentalement décomposable en deux problèmes LMM (équ. 1.2) symétriques et concaves.

En effet, par le procédé de fixer l'une des variables (x ou u) et de varier l'autre, deux expressions f et g peuvent être dérivées de $BILD$ comme suit :

$$f(x) = c^t x + \min_{u \in U} u^t(d - Qx) \quad \text{et} \quad g(u) = u^t d + \min_{x \in X}(c^t - u^t Q)x,$$

où $f : \mathbb{R}^{n_x} \to \mathbb{R}$ et $g : \mathbb{R}^{n_u} \to \mathbb{R}$ sont des fonctions que l'on doit minimiser.

En passant à leur dual, les fonctions f et g peuvent s'écrire de la façon suivante :

$$f(x) = c^t x + \max_{y \in Y(x)} b^t y\,; \quad g(u) = u^t d + \max_{v \in V(u)} v^t a,$$

où les domaines réalisables $Y(x)$ et $V(u)$ sont des polyèdres paramétrés exprimés sous forme

$$\begin{array}{rcll} Y(x) & = & \{y \in \mathbb{R}^{n_y} : By \leq d - Qx, y \geq 0\} & \subseteq \quad \mathbb{R}^{n_y}, \\ V(u) & = & \{v \in \mathbb{R}^{n_v} : v^t A \leq c^t - u^t Q, v \geq 0\} & \subseteq \quad \mathbb{R}^{n_v}. \end{array}$$

Ainsi, $BILD$ se reformule en l'un des problèmes LMM suivants :

$$(LMM_{xy}) \quad \min_{x \in X} f(x) \quad \text{ou bien} \quad (LMM_{uv}) \quad \min_{u \in U} g(u)$$

Dans la littérature, des discussions concernant l'optimisation concave (Pardalos *et al.* [32], Horst [20]), démontrent que les fonctions f et g sont linéaires par morceaux et concaves.

2.2.2 Résultats mathématiques

Les reformulations LMM de $BILD$ peuvent être utiles pour le résoudre. En effet, comme le montrent les propositions suivantes, les solutions de $BILD$ peuvent être déduites de ses reformulations LMM [7].

Proposition 1. *La solution (x^*, u^*) est une solution optimale de $BILD$ si et seulement si x^* est une solution optimale de LMM_{xy}, et u^* est une solution optimale de LMM_{uv}.*

Proposition 2. *Pour toute solution optimale locale x^* de LMM_{xy}, il existe un u^* dans U, tel que (x^*, u^*) est une solution optimale locale pour $BILD$.*

Proposition 3. *Si (x^*, u^*) est une solution optimale locale de $BILD$ telle que l'optimalité est stricte pour la variable x, i.e., il existe un $\epsilon > 0$ tel que la fonction objectif $h(x, u) = x^t - u^tQx + u^td$ de $BILD$ satisfait :*

$$h(x^*, u^*) \leq h(x, u) \quad \text{pour tout } x \in X \cap B_\epsilon(x^*), \quad \text{pour tout } u \in U \cap B_\epsilon(u^*),$$
$$h(x^*, u^*) < h(x^*, u) \quad \text{pour tout } u \in U \cap B_\epsilon(u^*), \quad u \neq u^*,$$

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

Ces propositions suggèrent qu'une façon de résoudre BILD se fasse par la résolution de ses reformulations. Cette démarche de résolution est due, d'une part, à l'existence d'une bijection entre les solutions optimales de $BILD$ et celles de ses reformulations LMM. D'autre part, elle est dû à l'injection qui donne une solution locale du $BILD$ en combinant les solutions locales de ses reformulations LMM (la réciproque n'est pas vraie [5]).

Dans le cas où l'un des ensembles du problème $BILD$ (X ou U) est non borné, il est nécessaire, avant de lancer la résolution et selon la littérature [7], de vérifier si $BILD$ est non borné avec la proposition 4.

Proposition 4. *Supposons que X est borné et U non borné. La valeur optimale z^* de $BILD$ est bornée si et seulement si pour tout x dans X, le polyèdre $Y(x)$ est non vide.*

Ainsi, un problème fortement $\mathcal{NP}$-complet FORBIDDANCE [7] découlant de la proposition 4 est résolu en premier. Si le résultat de cette résolution est strictement négatif, alors nous conclurons que $BILD$ est non borné. Sinon, nous nous procédons à résoudre $BILD$ ordinairement.

La possibilité d'avoir une meilleure solution pour $BILD$ revient à combiner des meilleures solutions de ses reformulations. Ce concept mène à chercher, en premier lieu, parmi les sommets voisins afin de choisir ceux des meilleurs. Ainsi, il est intéréssant d'élargir les notions théoriques du sommet voisin avec les problèmes $BILD$.

Afin de pouvoir en parler, nous présentons une paire de définitions en considérant comme notations les ensembles $N_X(\bar{x})$ et $N_U(\bar{u})$ de sommets voisins de $\bar{u}$ dans U et de $\bar{x}$ dans X respectivement.

Définition 14. *Soient $\bar{x}$ un sommet de X et $\bar{u}$ un sommet de U. Un voisin de $(\bar{x}, \bar{u})$ dans le domaine réalisable de $BILD$ est tout doublet qui s'écrit sous forme :*

$$(x^i, \bar{u}) \text{ avec } x^i \in N_X(\bar{x}) \ ; \quad (\bar{x}, u^i) \text{ avec } u^i \in N_U(\bar{u}).$$

Définition 15. *Soient $\bar{x}$ un sommet de X et $\bar{u}$ un sommet de U. La solution $(\bar{x}, \bar{u})$ est minimum N-locale de $BILD$ si et seulement si elle vérifie les deux conditions suivantes :*

$$h(\bar{x}, \bar{u}) \leq h(x^i, \bar{u}) \text{ pour tout } x^i \in N_X(\bar{x})$$
$$\text{et}$$
$$h(\bar{x}, \bar{u}) \leq h(\bar{x}, u^i) \text{ pour tout } u^i \in N_U(\bar{u}).$$

2.2.3 Procedure de résolution

La résolution de $BILD$ consiste à minimiser ses reformulations LMM concaves dans leur polyèdre. Cela implique la recherche, parmi les solutions locales x^* de LMM_{xy} et u^* de LMM_{uv}, de la solution optimale (x^*, u^*) de $BILD$ [Theo. 1]. L'obtention de cette dernière n'est pas évidente, mais il est facile d'obtenir une solution N-locales de $BILD$ avec l'algorithme "Augmented Mountain Climbing" (MC), introduit par Konno [23] en 1971. Nous décrivons cette technique à la section 4.1.1.

2.3 Revue de littérature

Notre recherche était inspirée de la littérature sur plusieurs axes de problèmatiques. La compléxite, la convergence et la dégénérescence furent parmi ces axes décrits dans cette section.

2.3.1 Compléxité

En général, la résolution de BILD est $\mathcal{NP}$-complet. En effet, selon Pardalos et Schnitger [33], la résolution de ce type de problème est $\mathcal{NP}$-complet, même pour un cas particulier de la minimisation d'une fonction concave quadratique sur un hypercube. Plus tard, en deux ans, Kalantari et Bagchi [8] ont prouvé cette complexité.

Marcus [34] a trouvé en 1999 une façon de réduire la complexité numérique des problèmes concaves. En effet, il a modifié l'algorithme de Tuy en y ajoutant des coupes assignées, non seulement aux sommets du polyèdre, mais aussi à ses pseudo-sommets. Ce type de coupes est

plus pronfond que celui de Tuy et l'algorithme résultant accélère la vitesse de convergence. En comparant ses résultats numériques avec ceux de Zwart [40] sur de mêmes types d'instances, jusqu'à 50% de coupes sont épargnées du calcul.

2.3.2 Convergence

Pour les problèmes de minimisation concave, il est difficile de prouver ou de réfuter la convergence finie d'un algorithme comprenant uniquement la technique des plans coupants. En effet, des expériences ont montré que l'algorithme des plans coupants, tel que proposé par Tuy, peut boucler sur certains problèmes [9][40]. En essayant d'éviter cette situation, Zwart [41] a proposé en 1973 des modifcations, sans pouvoir prouver la convergence de l'algorithme. Ils ont tous constaté que l'algorithme de coupe de concavité converge lentement vers une solution locale vu que les coupes deviennent presque parallèles éliminant ainsi une très petite partie de la région réalisable à chaque itération [Sec. 4.1].

L'étude de la convergence fut aussi traitée par Konno [24] en 1971 avec les problèmes $BILD$. Vaish et Shetty [36] ont également présenté en 1977 un algorithme similaire à celui de Konno [24], mais sans pouvoir garantir la convergence finie. De même, Horst [19] en 1976 n'a pas pu le prouver. Plus tard, Sherali et Shetty [35] l'ont démontré en 1980 par l'ajout judicieux des coupes disjonctives. Comme cette dernière approche est très coûteuse, elle est fréquemment mise de côté.

Seule, la technique des coupes de concavité ne suffit pas à garantir la convergence finie. D'où la necessité d'utiliser d'autres techniques pour l'assurer.

En exploitant la technique de "Branch and Bound" et en développant la stratégie de l'énumération implicite de Beale et Small [10], Audet [5] a pu démontrer en 1997, dans sa thèse doctorale, la convergence finie pour des problèmes convexes. Son approche, comme celle de Thieu [37] en 1988, ne requiert pas que les domaines X et U de BILD soient bornés. Elle comprend la procédure "augmented mountain climbing" de Konno [24] pour évaluer à chaque itération la borne inférieure de la solution optimale.

2.3.3 Dégénérescence

Dans un espace de dimension n, plus que $n+1$ directions peuvent être issues d'un sommet dégénéré et l'introduction d'une coupe n'est plus traditionnelle.

Devant une telle situation, Horst et Tuy [21] ont proposé la technique d'introduire des coupes de concavité, associées au sommet dégénéré, à travers les γ-extensions dans tous les directions des sommets voisins. Ce type de problème est considéré $\mathcal{NP}$-complet. Pour éviter l'énumération de tous les sommets voisins, Alarie et *al.* [4] ont proposé, en cas où la valeur du sommet dégénéré courant est différente de la meilleur valeur $\hat{\gamma}$ à date, la recherche aléatoire d'un sommet voisin non dégénéré afin de lui introduire une coupe.

2.4 Discussion

Ce chapitre constitue une base théorique essentielle pour nous rejoindre avec les chercheurs dans leurs avancées et leurs défis rencontrés. Pour explorer concrètement ces derniers, nous décrivons au chapitre suivant une approche des modèles numériques qui vont être résolus par des algorithmes spécifiques inspirés de la littérature. À la suite de cette exploration, nous proposons au chapitre quatre des améliorations à apporter dans ce domaine de recherche.

CHAPITRE 3

AVANCÉES ET DÉFIS RENCONTRÉS

Dans ce chapitre, des mécanismes de conception et d'implémentation vont être explorés pour joindre pratiquement les avancées et les défis rencontrés dans la littérature.

Les problèmes de compléxité, de convergence et de dégénérescence sont décrits en premier. Concernant les avancées rencontrées, la technique de réflexion et le choix du meilleur sommet voisin viennent après.

Cette partie du document comprend aussi l'esprit de recherche avec lequel nous abordons les problématiques, les interprétons et décrivons les algorithmes conçus à leur résolution en s'inspirant de la littérature.

3.1 Complexité et recouvrement total du polyèdre

La quantité de ressources (en temps et en espace) nécessaire pour résoudre les $BILD$ était prise en considération durant la conception des algorithmes.

L'énumération de **tous** les sommets voisins d'un sommet non dégénéré x_0 du polyèdre s'obtiennent par le calcul ordinaire des ratios

$$\min\{ratios > 0\}.$$

Tout comme dans la méthode du Simplexe pour la programmation linéaire.

Selon les observations dans la figure 2.1, l'énumération des pseudo-sommets voisins d'un sommet x_0 du polyèdre est très coûteuse. D'ici, cet énumération est rendue $\mathcal{NP}$-complet. Pour réduire cette compléxité, nous nous restreignons, dans chaque direction issue de x_0, aux pseudo-sommets voisins les plus proches dont la défintion est comme suit.

Définition 16. *Soit x_0 un sommet non dégénéré du polyèdre. Les pseudo-sommets voisins les plus proches de x_0 s'obtiennent par des pivots simples de Simplex dans le sens inverse de calcul des ratios*

$$\max\{ratios < 0\}$$

.

Ainsi, la capacité d'implémenter la recherche d'un pseudo-sommet voisin le plus proche est rendu pareil à celle d'un sommet voisin. En effet, il est facile de trouver **tous** les sommets voisins d'un sommet non dégénéré x_0 en pivotant autour de lui par le procédé de Simplex ordinaire des ratios ($min\{ratios > 0\}$). Il est rendu de même facile trouver **tous** les pseudo-sommets voisins les plus proches de x_0 par un procédé inverse de Simplex ($max\{ratios < 0\}$).

Le cône polyhédral de sommet x_0 et de directions les sommets voisins x_i ($\delta_i = x_i - x_0$), couvre totalement le polyèdre. Par le fait même, le cône polyédral construit à partir d'un pseudo-sommet p_0 renferme le polyèdre avec les directions ($\delta_i = p_i - p_0$) des points voisins p_i. Ces derniers peuvent être soit un sommet, soit un pseudo-sommet du polyèdre [34].

Dans notre algorithme de résolution de $BILD$, nous développons dans une même procédure le calcul des sommets voisins et celui des pseudo-sommets voisins les plus proches de x_0. De ces derniers, nous choisissons aléatoirement un pour implémenter le cônes polyhèdral couvrant tout le polyèdre.

3.2 Convergence finie

En théorie, la génération d'une suite de solutions réalisables, qui se rapprochent infiniment, est le critère de convergence d'un algorithme. En pratique, ce critère s'applique lorsque le polyèdre réduit devient vide durant le processus de résolution et une solution optimale sera identifiée.

3.2.1 Modèles numériques et algorithme des plans coupants

Un choix de modèles numériques et une conception d'un algorithme pour les résoudre sont rédigés dans cette sous-section afin de présenter numériquement l'étude de la convergence durant le processus de résolution. Ces modèles sont le modèle numérique 1 (MN1) et le modèle numérique 2 (MN2), et leur algorithme de résolution est **Algo.1**.

$$(MN1)\quad \begin{array}{ll} \min\limits_{x} & \varphi(x) \\ \text{s.c.} & A_1 x \geq a_1 \\ & x \geq 0 \end{array} \quad ; \quad (MN2)\quad \begin{array}{ll} \min\limits_{x} & \varphi(x) \\ \text{s.c.} & A_2 x \geq a_2 \\ & x \geq 0 \end{array}$$

Où,

$\varphi(x) = -(x_1 - 4)^2 + (x_2 - 3)^2$: Fonction objectif concave à minimiser avec $x = (x_1, x_2)$.

$$A_1 = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 7 \\ 1 & 0 \\ -1 & 5 \end{pmatrix} \quad ; \quad a_1 = \begin{pmatrix} 2 \\ 3 \\ 21 \\ 7 \\ -2 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 2 & -1 \\ -1 & -2 \\ -3 & -1 \\ 2 - 3 \times \varepsilon & \varepsilon \\ -3 & 1 \end{pmatrix} \quad ; \quad a_2 = \begin{pmatrix} 2 \\ 3 \\ -3 \\ -16 \\ -23 \\ 14 - 19 \times \varepsilon \\ -21 \end{pmatrix}$$

avec $\varepsilon > 0$ est la précision machine.

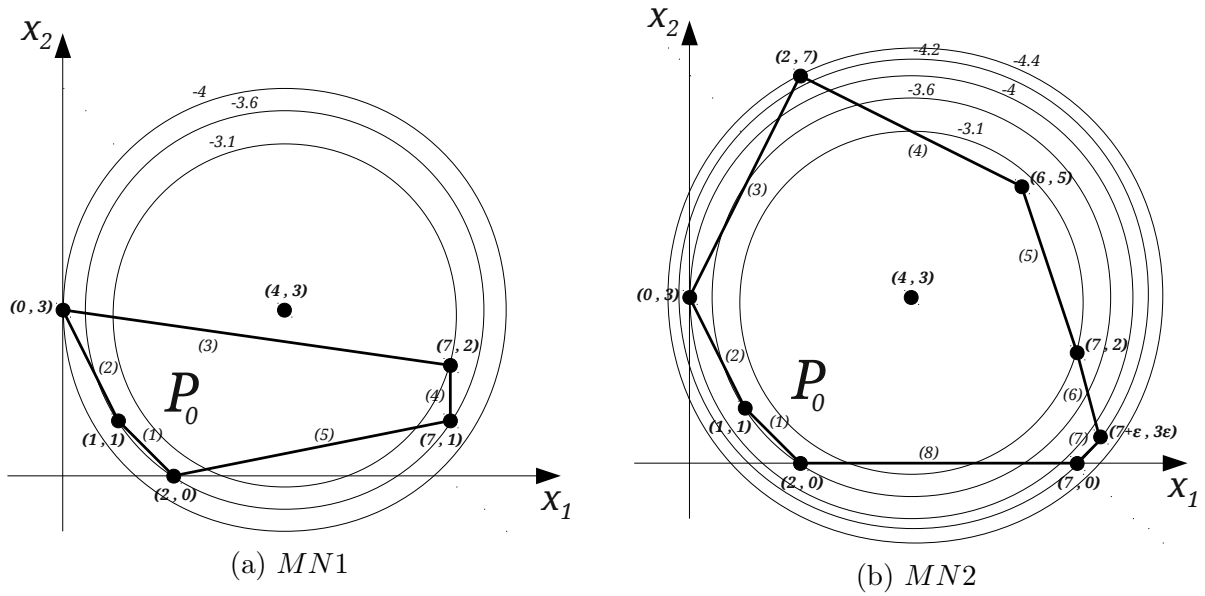
Et l'algorithme des plans coupants est comme suit :

Algo.1 à l'itération i

a.	Recherche d'un sommet x_i par le simplex. Poser $\gamma_i = \varphi(x_i)$
b.	Si $\gamma_i < \hat{\gamma}$ alors $\hat{\gamma} := \gamma_i$ et $\hat{x} := x_i$
c.	Si x_i non dégénéré alors calculer les $\hat{\gamma} - ext_k$ pour $0 \leq k < n$. Sinon, arrêter
d.	Détermination de coupe C_i à travers les $\hat{\gamma} - ext.$ $(C_i) : \pi^t(x - x_i) = 1$. Introduction de (C_i). $P_i := P_{i-1} \cap \{x : \pi^t(x - x_i) \geq 1\}$
e.	Si $P_i = \emptyset$ alors retourner $\hat{x}$ et $\hat{\gamma}$

Les manières d'apprécier l'éfficacité d'un algorithme sont la vitesse de convergence et le nombre d'itérations pour sortir de ses boucles.

Nous présentons graphiquement les modèles numériques avant d'interpréter leur processus de résolution avec **Algo.1**.



(a) $MN1$ (b) $MN2$

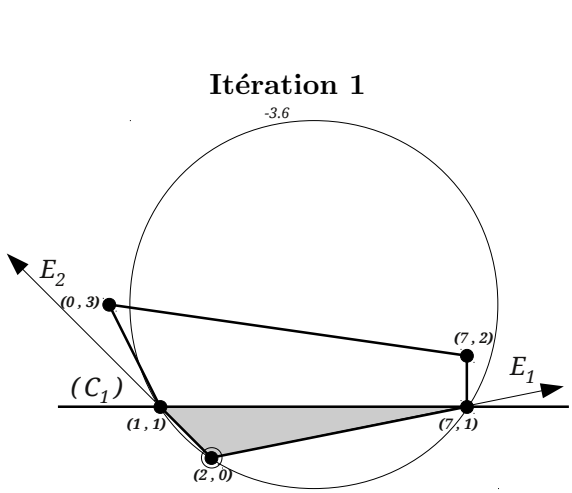
Figure 3.1 Présentation graphique de $MN1$ et $MN2$

La fonction objectif concave à minimiser, dans les modèles ci-dessus, est celle dont les coupes de niveau sont des cercles de centre $(4, 3)$ et de valeurs l'opposé des rayons de ces cercles (figure 3.1).

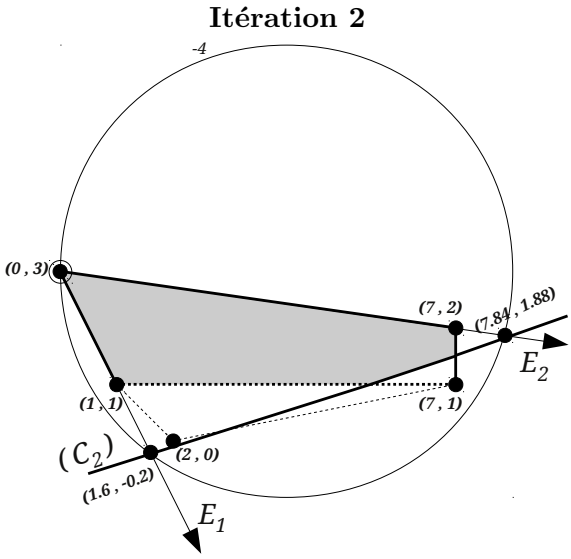
Le processus de résolution d'**Algo.1** se termine avec l'instance $MN1$ sans pouvoir le faire avec celle de $MN2$. La raison derrière cet échec est dû à l'erreur ε-machine qui se présente au voisinage du sommet $(7, 0)$ de l'instance $MN2$.

Convergence finie avec $MN1$

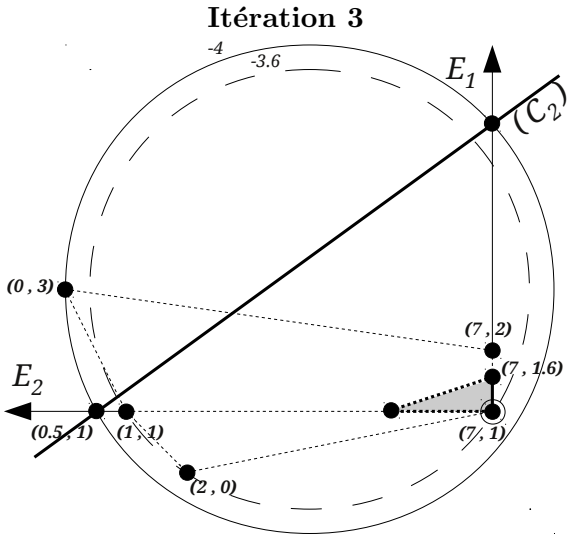
Itérativement, après introduction des coupes de concavité, le processus d'**Algo.1** réduit à vide (critère d'arrêt) le polytope P.



(a) $(C_1) : x_2 = 1$



(b) $(C_2) : 0.33x_1 - x_2 = 0.73$



(c) $(C_3) : 0.72x_1 - x_2 = 0.61$

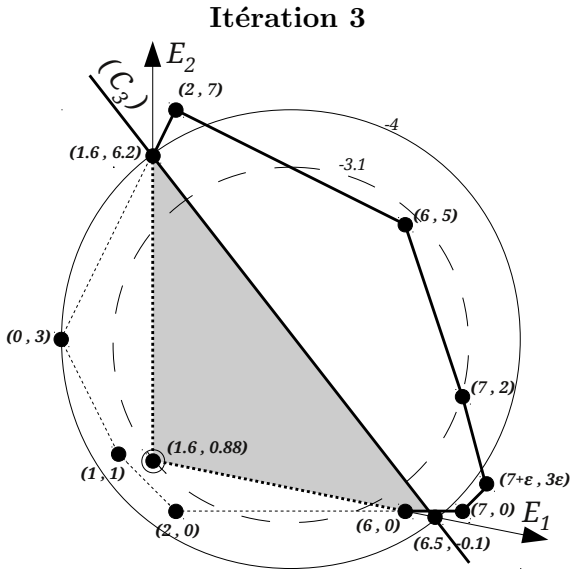
Figure 3.2 Processus d'exécution d'**Algo.1** avec $MN1$

L'**Algo.1** est de convergence finie avec l'instance $MN1$ et une solution optimale $\hat{x} = (0, 3)$ de valeur $\hat{\gamma} = -4$ est produite.

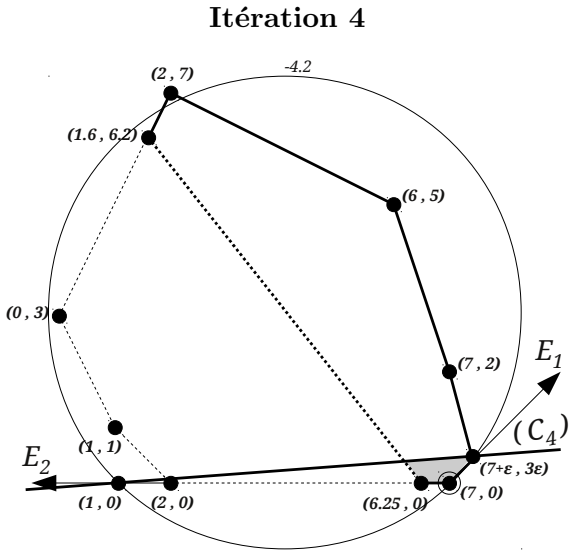
Convergence non finie avec $MN2$

Le processus d'**Algo.1** n'est pas de convergence finie avec cet instance. En effet, au voisinage du sommet $(7.0, 0.0)$, à partir de l'itération **4**, s'introduisent sans arrêt des coupes,

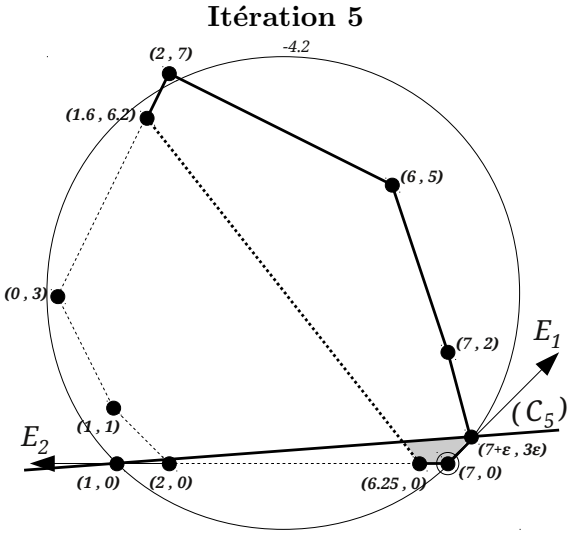
d'une façon parallèles et confondues, sans modifier le domaine réalisable et sans atteindre la solution optimale (fig. 3.3).



(a) $(C_3) : 4x_1 + 3x_2 = 25$



(b) $(C_4) : 3\varepsilon x_1 - (6 + \varepsilon)x_2 = 3\varepsilon$



(c) $(C_5) : 3\varepsilon x_1 - (6 + \varepsilon)x_2 = 3\varepsilon$

Figure 3.3 Processus d'exécution d'**Algo.1** avec $MN2$

Le critère d'arrêt échoue ($P_i \neq \emptyset$) avec des résultats non-optimaux. En effet, la meilleur solution à date est $x_4 = (7.0, 0.0)$ de valeur objectif $\gamma_4 = -4.24$ alors que la solution optimale du $MN2$ est celle du sommet $(2, 7)$ de valeur -4.47 (fig. 3.1b).

3.2.2 Problème de cumul des coupes

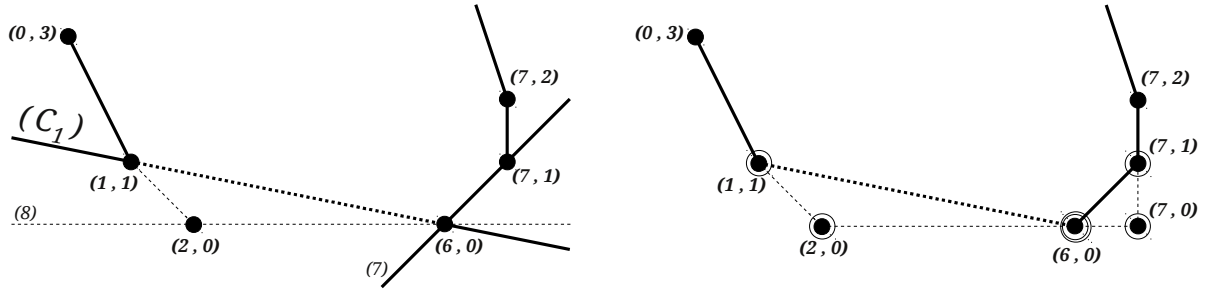
Le problème de cumul des coupes est souvent rencontré lorsque les coupes s'ajoutent sans arrêt au voisinage d'un sommet, d'une façon presque parallèle et confondue, dans le domaine réalisable restant. Ceci est dû en partie à l'erreur ε-machine recontrée avec les sommets ε-proches, dont certains sont produits par les coupes. Ce phénomène empêche la terminaison du processus de résolution et le retour de la solution optimale. En effet, comme le polyèdre restant se modifie légèrement, le critère d'arrêt échoue.

Pour contrer cette situation, nous nous inspirons de la littérature [4] avec la technique de réflexion d'une coupe [Sect. 3.4].

3.3 Dégénérescence

Dans un espace de dimension $n > 2$, une propriété générique d'un sommet du polyèdre est de visiter exactement ses n sommets voisins en pivotant autour de lui-même (cas non dégénéré). Si, par un pivot simple, la visite comprend des sommets voisins ou des pseudo-sommets voisins de plus, le sommet du polyèdre est dit dégénéré.

Comme mentionné dans la littérature, le comportement d'un pivot simple autour d'un sommet dégénéré est imprévisible. Ceci se voit clairement au sommet $(6, 0)$ dans la figure 3.4.



(a) Dégénérescence au sommet $(6, 0)$

(b) Sommets et pseudo-sommets voisins de $(6, 0)$

Figure 3.4 Effet de la dégénérescence dans $\mathbb{R}^2$

L'ajout de la contrainte (C_1) à l'ensemble des contraintes $(\{(c_7), (c_8)\})$ qui détermine le sommet $(6, 0)$ favorise la dégénérescence en ce dernier (fig. 3.4a). Par ce fait, les pivots simple autour de $(6, 0)$ génère les sommets voisins $(1, 1)$ et $(7, 1)$, et les pseudo-sommets voisins $(2, 0)$ et $(7, 0)$ (fig. 3.4b)

Un traitement de la dégénérescence avec les sommets voisins est proposé dans la littérature. En évitant l'énumération de tous les sommets voisins du sommet x_i dégénéré, Alarie et *al.* [4] ont proposé la recherche d'un sommet voisin non dégénéré afin de lui introduire une coupe. Pour que cette stratégie entre en action, il faut que la valeur courante $\gamma = \varphi(x_i)$ du sommet dégénéré x_i soit strictement supérieure à la meilleur valeur $\hat{\gamma}$ à date. Sinon, la dégénérescence va s'amplifier en x_i. La figure 3.5 illustre cette situation avec $x_i = (6, 0)$ et avec une coupe enracinée en $(7, 1)$.

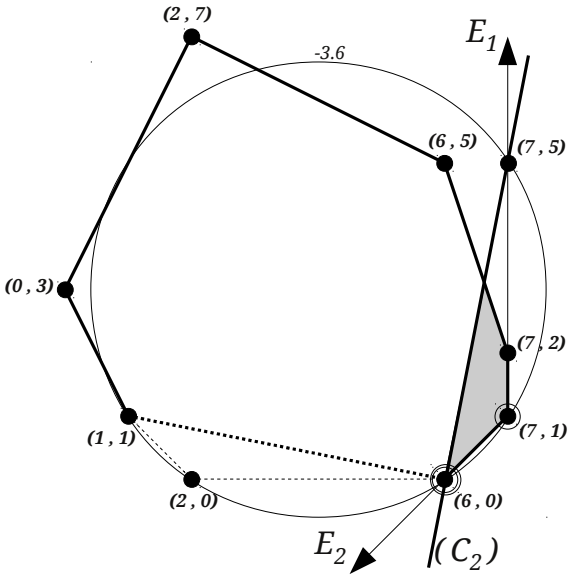


Figure 3.5 Coupe (C_2) associée au sommet non dégénéré

Comme $\gamma = \hat{\gamma}$, alors la coupe (C_2) assignée au sommet non dégénéré $(7, 1)$ voisin de $(6, 0)$ va amplifier la dégénérescence en ce dernier.

Le traitement de la dégénérescence intervient à l'étape **c.** de l'algorithme **Algo.1**. D'où, l'algorithme **Algo.2** aura lieu.

Étape c. de l'Algo.2 à l'itération i

c.	Si x_i est non dégénéré alors calculer les $\hat{\gamma} - ext_k$ pour $0 \leq k < n$. Sinon, appliquer le cas échéant : Si $\gamma = \hat{\gamma}$, arrêter. Sinon, chercher un sommet non dégénéré voisin de x_i et retourner à l'étape **b.**

Au chapitre suivant, le traitement de la dégénérescence avec les pseudo-sommets va suivre

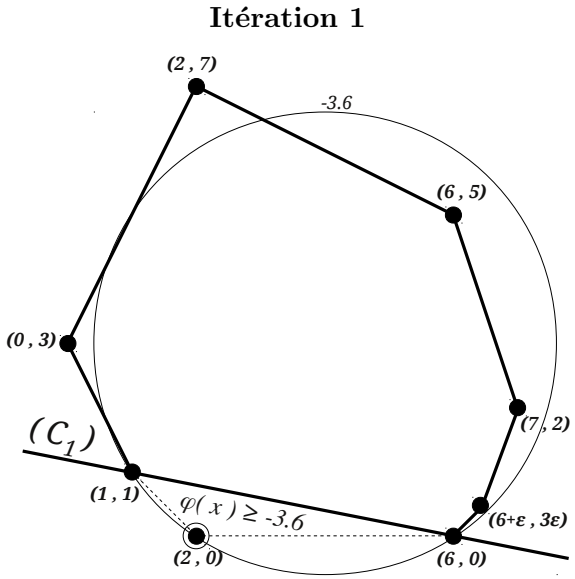
la même démarche que celui des sommets, *i.e.* recherche d'un pseudo-sommet non dégénéré voisin de x_i, en cas où $\gamma \neq \hat{\gamma}$, afin d'introduire une coupe.

3.4 Réflexion d'une coupe

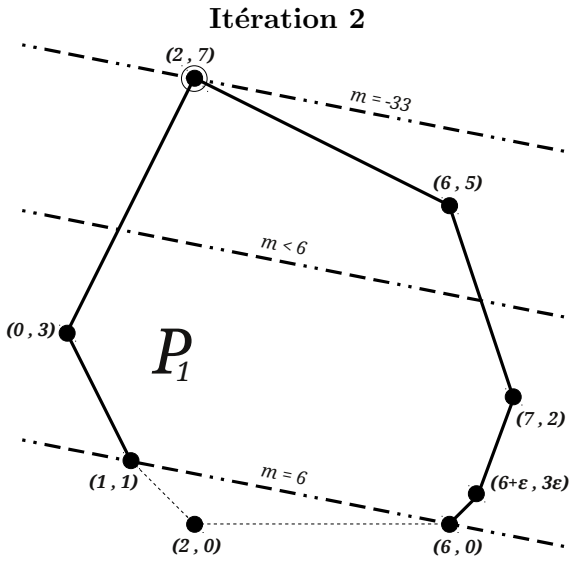
L'idée derrière la technique de réflexion d'une coupe est d'éviter tant que possible l'ajout des coupes assignées aux sommets qui sont ε-proches l'un de l'autre afin d'éviter le problème de cumul des coupes et d'améliorer la convergence de l'**Algo.1**.

Cette technique revient à trouver le sommet réfléchi de la dernière coupe introduite (C_i) associé à x_i. Pratiquement, l'implémentation de cette idée est réalisable par la résolution d'un problème de minimisation linéaire (PL) dont la fonction objectif est l'équation de la coupe (C_i). La solution ainsi générée est celle du sommet réfléchi.

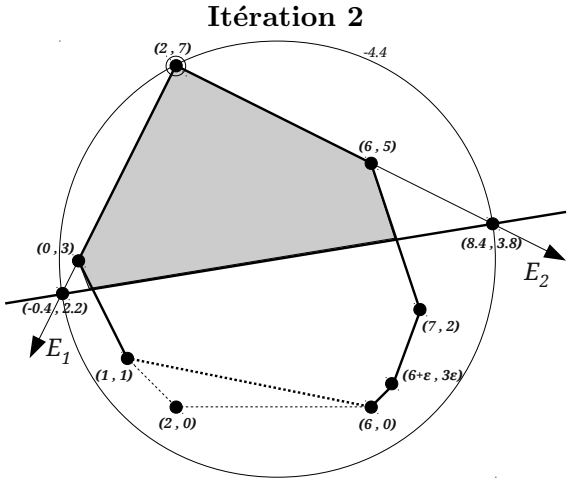
Par cette technique, le déploiement des coupes est plus large au cours du processus de résolution. À la figure 3.6, nous nous servons d'un modèle numérique afin d'interprêter ce déploiement.



(a) $(C_1) : x_1 - 5x_2 = 6$



(b) $(2,7) \in \ arg \min_{(x_1,x_2)\in P_1} x_1 - 5x_2$



(c) $(C_2) : -0.18x_1 + x_2 = 2.27$

Figure 3.6 Déploiement large des coupes

L'ajout de la deuxième coupe, à l'itération 2 (fig. 3.6b et fig. 3.6c), se fait au sommet optimal $(2, 7)$ au lieu de $(0, 3)$. Ceci résulte de la technique de réflexion qui minimise l'équation sans second membre de la coupe (C_1) sur le polytope réduit P_1.

$$(2,7) \in \ arg \min_{(x_1,x_2)\in P_1} x_1 - 5x_2$$

En intégrant cette technique au debut de l'**Algo.2**, l'algorithme **Algo.3** déduit est de

convergence finie sur un ensemble plus grand d'instances, y compris celle de MN2.

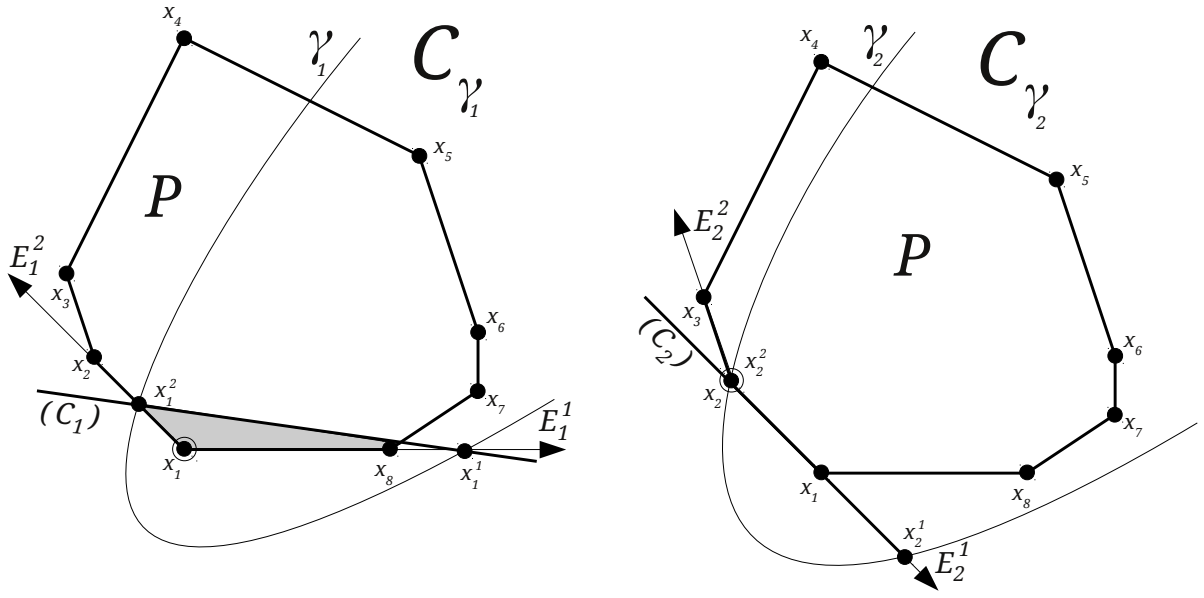
Étape a. de l'Algo.3 à l'itération i

a.	S'il y a une coupe introduite, chercher la réflexion x_i de la coupe (C_{i-1}). Sinon, chercher le sommet x_i par le simplex. Poser $\gamma_i = \varphi(x_i)$.

3.5 Meilleur sommet voisin et élimination de minima locaux

Il est nécessaire qu'un sommet non dégénéré x_0 associé à une coupe (C_0) soit de meilleure valeur que ses sommets voisins. Sinon, le calcul des γ_0-extensions est insuffisant et la coupe (C_0) ne sera pas la plus profonde.

Comme le montre la figure 3.7a, les extensions positives (E_1^1) et (E_1^2) issues du sommet x_1 rencontrent l'ensemble convexe C_{γ_1} aux γ_1-extensions x_1^1 et x_1^2 distincts de x_1 et distinctes entre elles. Avec le sommet x_2 (fig. 3.7b), l'une des extensions positives, soit (E_2^2), est complètement hors de l'ensemble convexe C_{γ_2}. Par suite, dans le même ordre, le calcul de l'une des γ_2-extensions va donner le sommet x_2^1 distinct de x_2 et l'autre γ_2-extension x_2^2 confondu avec x_2.



(a) γ_1-extensions

(b) γ_2-extensions

Figure 3.7 Recherche des γ-extensions

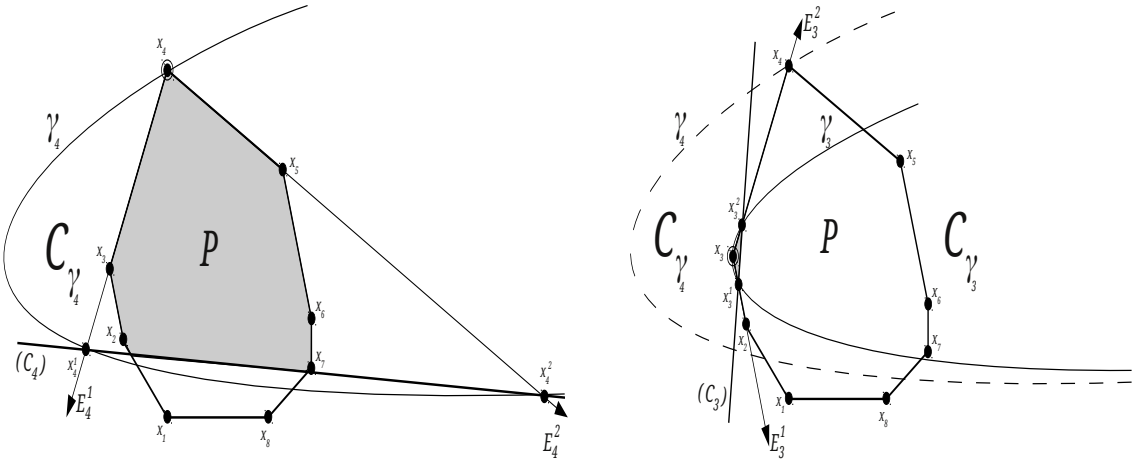
Comme la coupe (C_2) associée à x_2 passe par ce dernier, une dégénérescence se présente

en x_2 (fig. 3.7b).

Dans un espace de dimension 2, une telle coupe fait partie des contraintes qui détermine le polyèdre original. Dans un espace de dimension plus grand, une coupe pareille cause la dégénérescence sans qu'elle soit une contrainte originale du polyèdre.

Pour éviter cette situation, nous nous procédons de la recherche d'un meilleur sommet voisin afin d'élargir l'ensemble convexe C_γ et rendre plus avantageux le calcul des γ-extenions. Ce critère de recherche est valide puisque la fonction objectif est concave et l'ensemble C_γ est convexe [5].

Dans la figure 3.8a, le sommet x_4 est le meilleur sommet parmi ses sommets voisins. Parsuite, la coupe de concavité (C_4) associée à x_4 est profonde relativement à celles associées aux sommets voisins. Soit le sommet x_3 par exemple à la figure 3.8b, dans laquelle la coupe (C_3) tranche une toute petite partie de P.



(a) Choix du meilleur sommet (b) Choix d'un sommet ordinaire

Figure 3.8 Recherche d'un sommet voisin

Le critère du meilleur voisin se révèle aussi bénéfique dans l'élimination de minima locaux. En effet, dans la figure 3.8a, les minima locaux x_2, x_6 et x_7 sont éliminés par la coupe (C_4) et ensuite seront épargnés du reste de calcul. Ces observations nous permettent de raffiner l'**Algo.3** en proposant l'**Algo.4** : .

Étape a. de l'Algo.4 à l'itération i

a.	Recherche d'un meilleur sommet voisin x_i du sommet généré soit par le simplex, soit par réflexion de la coupe (C_{i-1})

3.6 Discussion

Nous avons présenté dans ce chapitre plusieurs modèles numériques ainsi que leurs algorithmes de résolution afin de donner une bonne approche de notre travail de recherche. Il fut ainsi facile à assimiler les problèmatiques vus en littérature. D'ici, nous allons proposer de nouvelles stratégies de déploiement des coupes avec les problèmes BILD dans le chapitre suivant.

CHAPITRE 4

DÉPLOIEMENT AMÉLIORÉ ET MISE À JOUR DES COUPES DE CONCAVITÉ

Tel que mentionné dans la littérature et tel que rencontré dans nos travaux de recherche, l'algorithme de résolution de $BILD$ peut s'accomplir en deux phases : phase de recherche et phase de coupe.

Dans la première phase, la mise à jour des solutions $(\hat{x}, \hat{u}, \hat{\gamma})$ vient à la suite d'une recherche des sommets x' et u', à partir du simplex et de la technique de réflexion, dans les deux ensembles X et U de $BILD$ respectivement. Nous appelons cette étape la recherche avancée. Dans la deuxième, l'introduction des coupes de concavité à travers les deux séries de $\hat{\gamma}$-extensions dont l'origine de chacune est x' et u' respectivement, s'accomplissent et une réduction des polyèdre X et U va se suivre. Cette démarche est décrite complètement par l'**algorithme CC**, tiré de [4]. Son résumé, à l'itération i, se présente comme suit :

Algorithme CC à l'itération i

A.	**Phase de recherche**	
	A.1	Recherche avancée d'un triplet (x', u', γ')
	A.2	Mise à jour de la meilleure solution connue $(\hat{x}, \hat{u}, \hat{\gamma})$
B.	**Phase de coupe**	
	B.1	Calcul des séries $\hat{\gamma}$-extensions dont l'origine de chacune est x' et u' respectivement
	B.2	Détermination des coupes de concavité $C_{\hat{\gamma}}(x')$ et $C_{\hat{\gamma}}(u')$. Voir si l'un des ensembes X_i ou U_i est vide.

Dans le but d'apporter des améliorations à l'**algorithme CC**, nous proposons trois nouvelles stratégies de déploiement de coupes qui seront incarnées dans trois nouveaux algorithmes : algorithme de coupe de concavité (CC), algorithme CC de mise à jour (CCU) et algorithme CC de pseudo-sommets (CCP). Combinés ensemble, ces trois algorithmes vont constituer un quatrième, nommée (CCUP).

4.1 Effets du choix des meilleurs sommets voisins et algorithme CC

Afin de présenter les effets du choix des meilleurs sommets voisins, nous décrivons d'abord la stratégie de MC, introduite par Konno [23] en 1971.

4.1.1 Algorithme MC

Dû à la symétrie entre les variables x et u dans le problème $BILD$, nous présentons l'algorithme MC, dans une seule direction et sans aucune perte de généralité, en le démarrant avec une solution donnée x^0 dans X.

Algorithme $MC(x^0)$

a.	Résoudre $\min_{u \epsilon U}(c^T x^0 - u^T Q x^0 + u^T d)$ pour obtenir un optimal u'.
	Soit z_u la valeur optimale.
b.	Résoudre $\min_{x \epsilon X}(c^T x - u'^T Q x + u'^T d)$ pour obtenir un optimale x'.
	Soit z_x la valeur optimale.
	Tant que $z_u \neq z_x$, affecter $x^0 \leftarrow x'$ et retourne en **a**.

Le processus $MC\,(x^0)$ se termine nécessairement en un nombre fini d'itérations [24] et va générer un triplet (x', u', γ') où (x', u') est la solution réalisable N-locale de valeur objectif γ'.

Un processus similaire se démarre avec une solution donnée u_0 dans U afin de générer le triplet (x'', u'', γ''). Le meilleur triplet conclut, en comparant les valeurs de γ, dans le sens le plus petit strict, sera selectionné et noté par (x', u', γ'). Suite à cette sélection, une mise à jours de la solution courante $(\hat{x}, \hat{u}, \hat{\gamma})$ et des coupes sera faite. Tant que les critères d'arrêts ne sont pas satisfaits, des coupes seront itérativement introduites dans les domaines X et U.

4.1.2 Effets et algorithme CC

Les effets du choix d'un meilleur sommet voisin sont décrits à la section 3.5 : obtention de coupes plus profondes et élimination plus important de minima locaux. Ce choix était fait dans un **seul** polyèdre. Ce concept de choix sera étalé avec l'**algorithme CC** [4], au moyen de MC, sur les polyèdres X et U de $BILD$ [4]. L'algorithme CC ainsi obtenu engendre la recherche d'un meilleur voisin. Sa phase de recherche, à l'itération i, est présentée comme suit :

Phase de recherche de l'algorithme CC à l'itération i

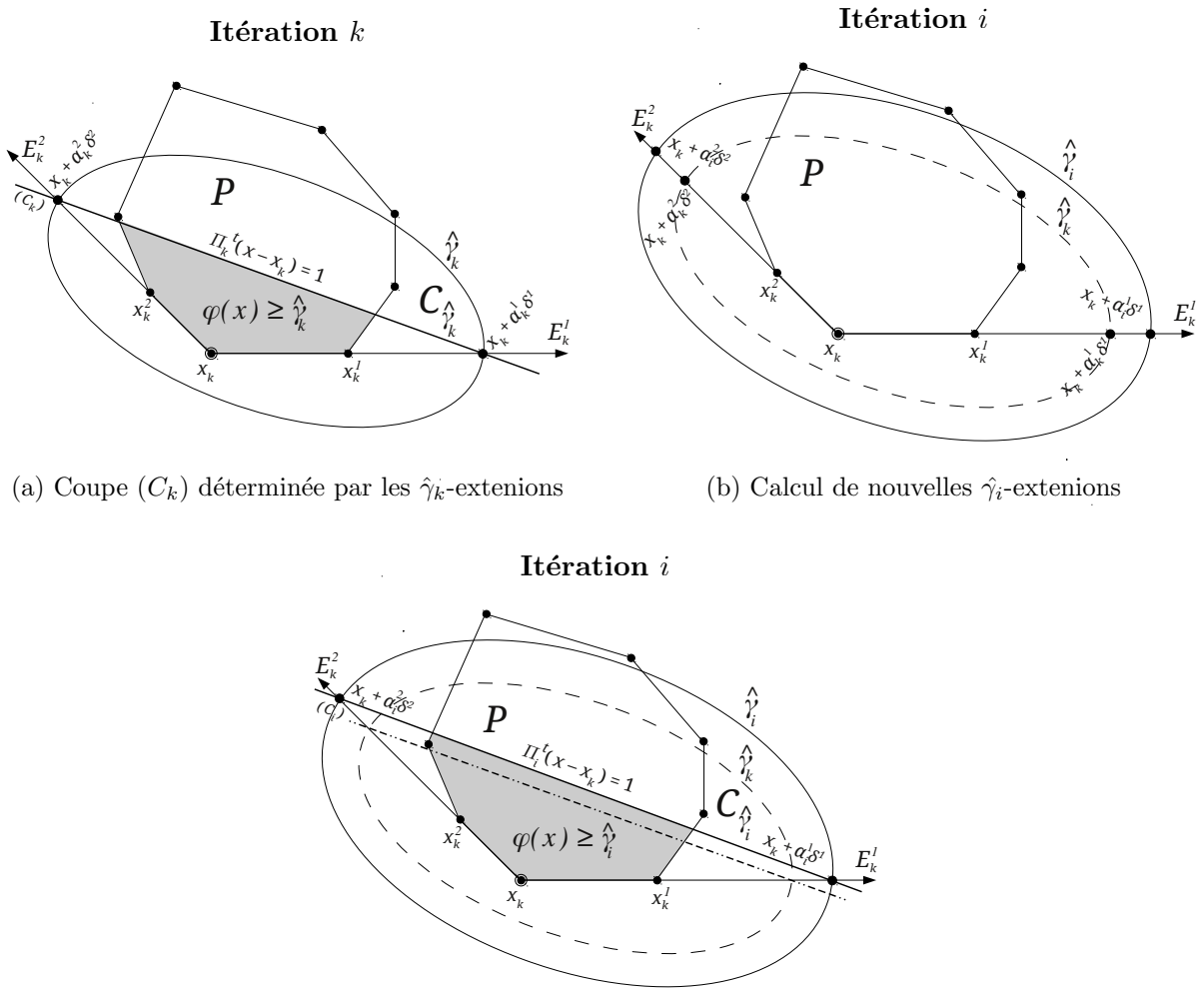
A.1	Recherche avancée d'un triplet (x', u', γ')
A.2	Choix des meilleurs sommets voisins de x' et u' avec MC
A.3	Mise à jour de $(\hat{x}, \hat{u}, \hat{\gamma})$

4.2 Effets de la mise à jour dynamique des coupes et algorithme CCU

Un nouveau procédé qui consiste à réduire dynamiquement le domaine réalisable, à chaque amélioration de la valeur $\hat{\gamma}$, pousse davantage les coupes de concavité et traite **parfois** les problèmes de dégénérescence et de cumul des coupes.

L'idée consiste à garder en mémoire toutes les coupes préalablement générées lors de l'application de l'algorithme. Ensuite, lorsqu'une nouvelle solution meilleure que la précédente est générée, alors toutes les coupes précédentes sont recalculées avec la nouvelle valeur de $\hat{\gamma}$. Il s'en suit que toutes ces coupes seront plus profondes.

Dans la figure 4.1, l'amélioration de la valeur $\hat{\gamma}$ mène à la poussée en profondeur des coupes de concavités introduites à une itération $k < i$ (fig. 4.1a). En effet, suite à cette amélioration, toutes les $\hat{\gamma}$-extensions antécédentes s'approfondissent dans leur extension positive jusqu'à ce qu'elles rencontrent la coupe de niveau de valeur $\hat{\gamma}$ améliorée (fig. 4.1b). Les nouvelles positions de ces $\hat{\gamma}$-extensions déterminent des coupes valides plus profondes que les antécédentes (fig. 4.1c).



(a) Coupe (C_k) déterminée par les $\hat{\gamma}_k$-extenions

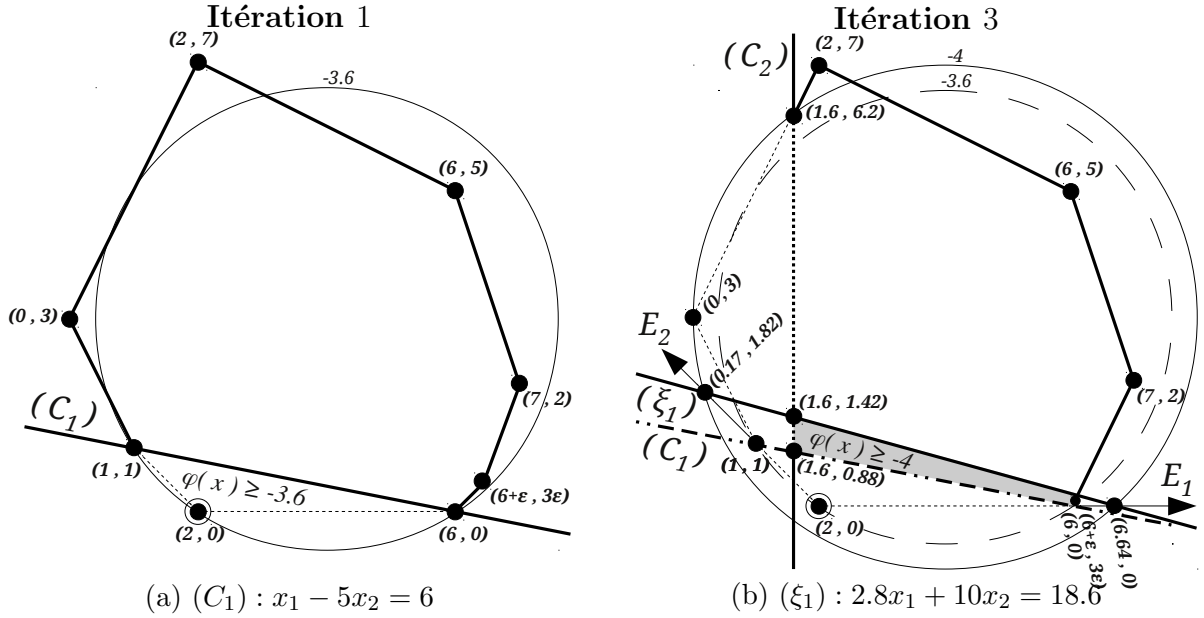
(b) Calcul de nouvelles $\hat{\gamma}_i$-extenions

(c) Coupe (ξ_i) déterminée par les $\hat{\gamma}_i$-extenions

Figure 4.1 Mise à jour de la coupe (C_k) à l'itération i

Se faisant, le domaine réalisable se réduit dynamiquement suivant les améliorations de la valeur $\hat{\gamma}$. Il n'est pas nécessaire de recalculer les directions voisines du sommet où la coupe est enracinée, il suffit simplement de recalculer les $\hat{\gamma}$-extensions.

La mise à jour règle parfois le problème du cumul des coupes et le problème de la dégénérescence. Le modèle numérique suivant l'illustre.



(a) $(C_1) : x_1 - 5x_2 = 6$ (b) $(\xi_1) : 2.8x_1 + 10x_2 = 18.6$

Figure 4.2 Mise à jour d'une coupe

Dans la figure 4.2b, la mise à jour de la coupe (C_1) mène à la coupe (ξ_1) qui enlève la dégénérescence au sommet $(6, 0)$ et évite le problème de cumul des coupes en éliminant les sommets ε-proches $(6, 0)$ et $(6 + \varepsilon, 3\varepsilon)$.

Dû à ses avantages, cette technique est intégrée à l'étape $A.3$ de l'**algorithme CC**. D'où, notre second algorithme CCU.

Phase de recherche de l'algorithme CCU à l'itération i

A.1	Recherche avancée d'un triplet (x', u', γ')
A.2	Choix des meilleurs sommets voisins de x' et u'
A.3	Mise à jour de $(\hat{x}, \hat{u}, \hat{\gamma})$ et des coupes antécédantes dans X et U

4.3 Nouveaux usages des pseudo-sommets et algorithme CCP

L'usages des pseudo-sommets pour la construction des coupes et le traitement de la dégénérescence sont décrits dans cette section.

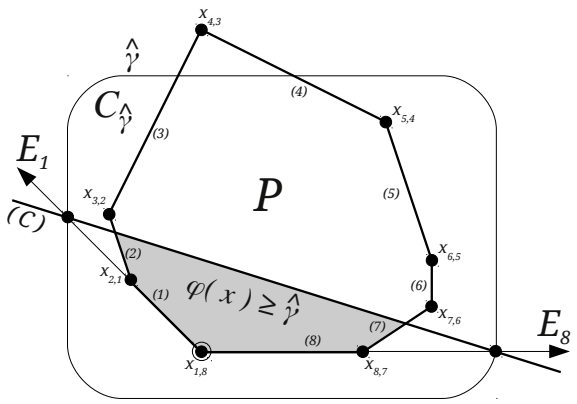
4.3.1 Cônes polyédraux et déploiement des coupes

La notion des pseudo-sommets a été proposée par Porembski [34] en 1999 pour introduire une coupe similaire à celle de Tuy, nommée coupe de décomposition. L'introduction d'une

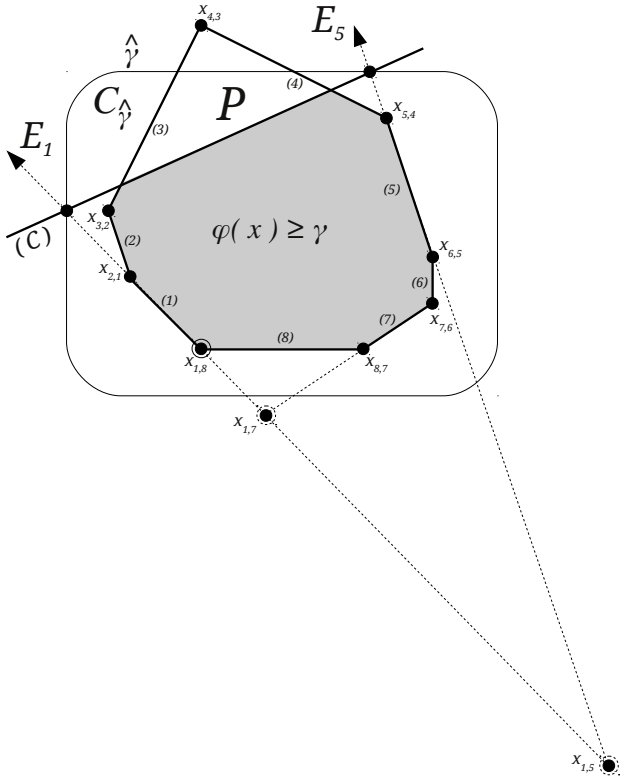
telle coupe se fait après la décomposition du cône en une multitude de cônes dont l'ensemble convexe couvre le polyèdre. Comme ce recouvrement est total, la validité des coupes introduite est assurée.

Sans décomposer les cônes polyédraux, nous avons adopté l'usage des pseudo-sommets pour les construire [Sect. 3.1]. Avec un seul pseudo-sommet, les cônes sont mono-pôles et leurs directions sont celles des points voisins ($\delta_i = p_i - p_0$). Ces derniers peuvent être soit des sommets, soit des pseudo-sommets.

Ce nouveau type de cônes va permettre avec ses directions, la détermination de différentes coupes. Ces dernières, nommées aussi coupes de concavité, s'introduisent dans une procédure de minimisation concave. L'interprétation de leur introduction est présenté dans la figure 4.3a avec sommet appartenant au polytope P, et dans l'autre figure 4.3b avec un pseudo-sommet n'appartenant pas à P. Sur cette exemple, la coupe issue du pseudo-sommet est nettement plus profonde que celle issue du sommet.



(a) Pseudo-sommet $x_{1,8}$ appartenant au polytope P



(b) Pseudo-sommet $x_{1,5}$ n'appartenant pas au polytope P

Figure 4.3 Coupes associées aux pseudo-sommets

4.3.2 Choix du pseudo-sommet

Dans un espace de dimension n, les sommets voisins du sommet non-dégénéré x_0 s'obtiennent par n pivots simples dans les n extensions positives issues de x_0. Ceci s'implémente facilement avec le procédé de calcul des ratios

$$\min\{ratios > 0\}.$$

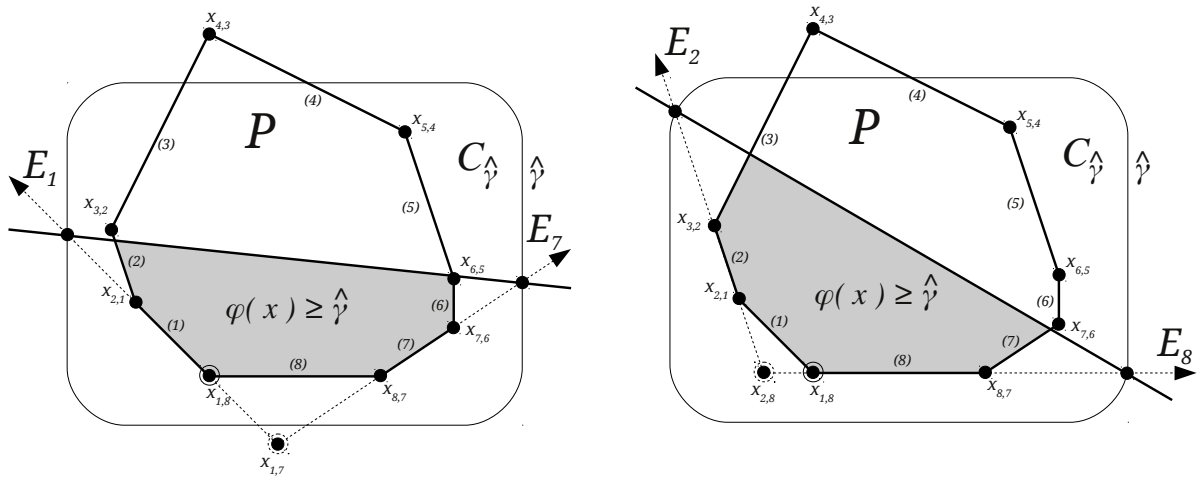
Pareillement, l'obtention des pseudo-sommets voisins **les plus proches** se fait par le procédé de calcul

$$\max\{ratios < 0\}$$

dans les n extensions négatives issues de x_0. Cette façon d'identifier le pseudo-sommet mène à la figure 4.3b à une coupe enracinée en $x_{1,5}$ plutôt qu'en $x_{1,7}$ comme l'aurait fait un ratio minimal.

La procédure de recherche des **autres** pseudo-sommets voisins de x_0 est coûteuse et compliquée à implémenter. Pour cela, elle est évitée dans notre conception des algorithmes.

Ainsi, un choix arbitraire d'un pseudo-sommet voisin parmi les plus proches de x_0 était conçu avec nos algorithmes de résolution. Ceci s'illustre dans un espace de dimension 2 dans la figure 4.4 en notant x_0 par $x_{1,8}$.



(a) Pire choix du pseudo-sommet $x_{1,7}$ voisin de $x_{1,8}$

(b) Non-pire choix du pseudo-sommet $x_{2,8}$ voisin de $x_{1,8}$

Figure 4.4 Coupes associées aux pseudo-sommets voisins les plus proches

Dans la figure ci-dessus, la coupe enracinée au pseudo-sommet $x_{1,7}$ tranche un espace

moindre du polytope P. Dans ce cas, nous disons que ce type de coupe est associé au pire choix des pseudo-sommets voisins.

4.3.3 Traitement de la dégénérescence et algorithme CCP

Dans un nouvel usage des pseudo-sommets, nous allons traiter la dégénérescence selon le même principe qu'Alarie et *al.* [4] ont suivi avec l'usage des sommets [Sect. 3.3]. Ceci consiste premièrement de vérifier si la valeur objectif courante γ' d'un sommet dégénéré (x' ou u') est différente de $\hat{\gamma}$. Deuxièment, d'introduire une coupe à l'un de ses points voisins non-dégénérés. Ces derniers, notés par p'_x et p'_u relativement aux polyèdres X et U respectivement, sont soient un sommet, soit un pseudo-sommet des polyèdres.

4.3.4 Algorithme CCP

Intuitivement, la technique d'introduction des coupes associées aux pseudo-sommets fait partie de la phase de coupe. Ainsi, elle sera intégrée dans l'algorithme CC pour donner notre troisième algorithme CCP.

Phase de coupe de l'algorithme CCP à l'itération i

B.1	Recherche des pseudo-sommets voisin p'_x et p'_u, les plus proche de x' et de u' respectivement
B.2	Calcul des séries $\hat{\gamma}$-extensions dont l'origine de chacune est p'_x et p'_u respectivement
B.3	Détermination des coupes de concavité $C_{\hat{\gamma}}(p'_x)$ et $C_{\hat{\gamma}}(p'_u)$. Voir si l'un des ensembles X_i ou U_i est vide.

4.4 Algorithme de résolution CCUP de $BILD$

L'algorithme CCUP combine les deux propositions originales de cette recherche, soit la mise à jour dynamique des coupes ainsi que la génération de coupes issues de pseudo-sommets. Une description détaillée de l'algorithme CCUP est présentée à l'Annexe A.

En résumé, nous profitons d'abord de la stratégie MC, afin de trouver une solution N-locale. Ensuite, si la solution courante résultante (x', u', γ') ou (x'', u'', γ'') porte d'amélioration, une mise à jour de la meilleur solution à date $(\hat{x}, \hat{u}, \hat{\gamma})$ et des coupes vont s'effectuer. Enfin, comme les reformulations de $BILD$ sont concaves [Sect. 2.2], des coupes de concavité s'associent itérativement aux composants x et u, qui peuvent être soient des sommets, soient

des pseudo-sommets. Les introductions de ces coupes sont valides et réduisent les ensembles X et U de $BILD$ respectivement.

4.5 Généralisation des algorithmes

Comme nous avons procédé de la division de nos algorithmes en deux phases (phase de recherche et phase de coupe), nous arrivons à les décrire en un seul algorithme. Ceci nécessite les notations suivantes :

P	:	polyèdre tel que $P = X$ ou $P = U$
ξ_P	:	points extrêmes de P
$\mathcal{D}_P$	:	sommets et pseudo-sommets dégénérés de P
$\mathcal{N}_P(s)$	:	voisins réalisables ou non d'un sommet s dans P incluant les points à l'infini

Les deux sous-sections suivantes comprennent des fonctions qui décrivent chaque phase. La dernière comprend l'algorithme général.

4.5.1 Phase de recherche

Dans cette phase, la fonction **Racine** retourne un point s'' non dégénéré. Ce dernier peut être soit un sommet, soit un pseudo-sommet du polyèdre P.

Fonction $s'' \leftarrow$ **Racine**(P, s')

Entrée :
P : polyèdre
s' : un point extrême de P
Sortie :
$$s'' = \begin{cases} \emptyset & : \text{ si on ne fait pas de coupe} \\ s'' & : \text{ un sommet ou pseudo-sommet non dégénéré de } P \end{cases}$$
Si $\gamma' = \hat{\gamma}$ et $s' \notin \mathcal{D}_P$:
retourner$(s'' = s')$
Sinon Si $\gamma' < \hat{\gamma}$:
Si $(\exists s'' \in \mathcal{N}_P(s') - \mathcal{D}_P)$
(*i.e.* s'il existe un sommet ou un pseudo-sommet voisin non dégénéré de s')
retourner(s'')
Sinon retourner$(s'' = \emptyset)$

4.5.2 Phase de coupe

S'il n'y a pas d'introduction de nouvelles coupes, la fonction **Coupe** retourne la réflexion de la dernière coupe introduite.

Fonction $s^0 \leftarrow$ **Coupe**(P, s'')

Entrée :

$\emptyset$: polyèdre

$$s'' = \begin{cases} \emptyset & : \text{ si on pas de coupe} \\ s'' & : \text{ un sommet ou pseudo-sommet non dégénéré de } P \end{cases}$$

Sortie :

s^0 la reflexion

Si $s'' = \emptyset$ retourner $(s^0 = 0)$

Si $(\exists s^0 \in \mathcal{N}_P(s'') \cap P$ avec $\gamma^0 < \hat{\gamma})$

retourner(s^0)

Sinon

Coupe de concavité en s''

Retourner(s^0 = reflexion de s'')

4.5.3 Algorithme général

L'algorithme général incarne les algorithmes de résolution de $BILD$ conçus dans notre recherche. Soient CC, CCU, CCP et CCUP.

Algorithme général

Initialisation :

$\hat{\gamma} \leftarrow +\infty$

$(x^0, u^0) \leftarrow (0, 0)$

Tant que $\hat{\gamma} > -\infty$ **ou** $X \neq \emptyset$ **ou** $U \neq \emptyset$:

$(x', u', \gamma') \leftarrow$ **min** $[MC(x^0), MC(u^0)]$

Si $\gamma' < \hat{\gamma}$ affecter (x', u', γ') à $(\hat{x}, \hat{u}, \hat{\gamma})$

et mettre à jour les coupes

Si $\hat{\gamma} > -\infty$ **et** $X \neq \emptyset$ **et** $U \neq \emptyset$ **:**

$x'' \leftarrow$ **Racine** (X, x')

$x^0 \leftarrow$ **Coupe** (X, x'')

$u'' \leftarrow$ **Racine** (U, u')

$u^0 \leftarrow$ **Coupe** (U, u'')

4.6 Discussion

Nous avons proposé deux nouvelles stratégies qui élargissent le déploiement des coupes. Elles étaient conçues dans des algorithmes afin que ces derniers soient comparés dans le chapitre suivant. Une telle comparaison mène à prouver ou à refuter l'efficacité de nos choix.

CHAPITRE 5

RÉSULTATS NUMÉRIQUES

Dans ce chapitre, nous présentons les résultats d'expériences numériques pour les quatres algorithmes. Nos tests ont été menés avec le language de programmation Python, et tous les programmes linéaires ont été résolus avec le logiciel Cplex.

Le premier algorithme (CC), inspiré de la littérature, comprend la technique des plans coupants. Le second (CCU), version améliorée de (CC), implique la technique de la mise à jour des coupes. Le troisième (CCP), pareil à (CC), sauf qu'il associe des coupes aux pseudo-sommets. Et le quatrième (CCUP), combine les algorithmes (CCU) et (CCP).

La validation des codes se vérifie sur un ensemble d'instances tirées de la littérature. Ceci va confirmer le bon fonctionnement des codes en comparant leurs résultats sortis avec ceux de la littérature.

Après cette validation, la phase suivante comprend l'exécution de ces quatres codes sur un ensemble d'instances $BILD$, de grande dimension, généré aléatoirement. Des commentaires suivront les résultats numériques de chaque catégorie d'instances.

5.1 Instances $BILD$ de la littérature

Les instances $BILD$ rencontrées dans la littérature ont constitué une référénce pour s'assurer du fonctionnement correct des algorithmes implémentés. Ceci se traduit par la comparaison des sorties (solutions et valeurs optimales) de leur code avec celles trouvées en littérature. La correspondance entre les divers résultats obtenus suggère la validité du fonctionnement de ces codes.

Ainsi de la littérature, sont tirées sept instances $BILD$. Les instances 6224 et 6225 sont trouvées dans les travaux d'Alarie et *al.* [4]. Et les autres instances (3223, 4444, 6556, 7226, 7667), des travaux d'Al-Khayyal [3].

Les noms de ces instances désignent les dimensions des matrices composantes du problème $BILD$ [Éqt. 1.1] :n_v, n_u, n_x et n_y. D'où, le format des noms est comme suit : $n_v n_u n_x n_y$.

Le tableau 5.1 décrit le comportement des 4 algorithmes sur 7 problèmes de la littérature. Les colonnes donnent le temps de résolution en secondes, la valeur optimale de la fonction objectif ainsi que le nombre de coupes requises en x et en u pour la résolution.

Tableau 5.1 Instances $BILD$ de la littérature

Inst	Algorithme CC				Algorithme CCU			
ances	temps	objectif	coupes x	coupes u	temps	objectif	coupes x	coupes u
3223	0.1472	-25	1	0	0.1243	-25	1	0
4444	0.9174	-25	3	3	0.98	-25	3	3
6224	0.4344	0	2	2	0.4488	0	2	2
6225	0.393	6	1	3	0.4425	6	1	3
6556	11.78	-17	39	38	11.81	-17	39	38
7226	0.107	7.5	0	1	0.1038	7.5	0	1
7667	0.2946	-213	1	0	0.2551	-213	1	0
	Algorithme CCP				Algorithme CCUP			
3223	0.1538	-25	1	0	0.1553	-25	1	0
4444	0.7075	-25	**1**	**2**	0.7311	-25	**2**	**1**
6224	0.3362	0	**1**	2	0.3465	0	**1**	2
6225	0.5108	6	1	**2**	0.5283	6	1	**2**
6556	19.1	-17	**45**	**45**	2.447	-17	**6**	**5**
7226	0.123	7.5	0	1	0.1232	7.5	0	1
7667	0.2561	213	1	0	0.2591	213	1	0

Pour chacune des instances, les quatre algorithmes identifient la même solution globalement optimale. De plus, et sans surprise, le temps d'exécution s'accroît avec le nombre de coupes introduites.

Dans quelques instances (4444, 6224, 6225 et 6556), la technique des pseudo-sommets intégrée dans les algorithmes CCP et CCUP présente un avantage sur les autres sauf pour l'instance 6556. Pour cette dernière, nous affirmons que les pires choix des pseudo-sommets voisins les plus proches étaient faits et ceci a eu comme conséquence d'augmenter le nombre de coupes requis pour la résolution.

Autres observations qui se fûrent rencontrées durant les expériences sont présentées là-dessous :

1. La technique de mise à jour se fut rencontrée avec l'instance 6225 sans modifier le nombre de coupes.

2. Le problème de dégénérescence, rencontré avec les instances 4444, 6224, 6225 et 7667, était résolu avec les anciennes [4] ainsi qu'avec les nouvelles stratégies de traitement.
3. Avec les instances 4444 et 7667, nous avons remarqué la possibilité de n'avoir aucun pseudo-sommet dans toutes les extensions négatives.
4. Le problème de cumul des coupes ne s'est jamais produit.

5.2 Instances $BILD$ aléatoires

Plusieurs catégories d'instances $BILD$ ont été générées aléatoirement tel que décrit dans [7]. Chaque catégorie est caractérisée par une dimension n variant de 10 à 40. Les instances de ces catégories peuvent être de densités variées d. Le paramètre de densité d indique la proportion d'éléments non nuls dans les matrices définissant le problème BILD. Ainsi, lorsque le générateur produit une instance $BILD$ de dimension n et de densité d, cet instance aura pour paramètres $n_v = n_u = n_x = n_y = n$ et $D = d$.

Dans les sous-sections suivantes, nous présentons les tableaux numériques de chaque catégories d'instances. Avec une densité d donnée, dix instances $BILD$ sont résolues dans chaque ligne des tableaux qui interprêtent les moyennes et les écart-types de chaque mesure de calcul : temps d'exécution et nombre de coupes introduites. La dernière colonne, intitulée échec, désigne le nombre d'instances dont le temps de résolution excède dix minutes [1].

À la lumière de la littérature [4], nous discutons au début de chaque sous-section les tableaux des résultats. Suite à ces derniers, deux séries de commentaires vont s'en déduire : celle de la technique de mise à jour des coupes et celle de la technique des pseudo-sommets.

5.2.1 Instances $BILD$ de dimension $n = 10$

Avec l'algorithme CC, les résultats du tableau 5.2 s'approchent étroitement de ceux de la littérature. Nous remarquons en plus qu'il n'y a aucun échec pour toutes les densités de cette catégorie, même avec nos quatre variantes algorithmiques.

Les chiffres en gras, présentés dans les tableaux, vont désigner trois types d'observations : les comparaisons, les exceptions et les précisions.

1. Dans la très grande majorité des cas, l'algorithme se comporte de l'une des deux façons suivantes. Ou bien il résout le problème en moins de 10 minutes, ou bien les problèmes de cumuls et les problèmes numériques rendent la résolution impossible. Des résultats très similaires ont été observés avec des seuils supérieurs à 10 minutes.

Tableau 5.2 Instances $BILD$ de dimension $n = 10$

Densité	Algorithme CC					Algorithme CCU				
	temps	σ_t	coupes	σ_c	échec	temps	σ_t	coupes	σ_c	échec
10	0.4193	0.0637	1.0	0	0	0.4303	0.0667	1.0	0	0
15	0.4652	0.1304	1.3	0.6403	0	0.4534	0.1418	1.3	0.6403	0
20	0.5494	0.2293	1.5	0.8062	0	0.5477	0.2204	1.5	0.8062	0
30	0.5073	0.1521	1.5	0.6708	0	0.5180	0.1617	1.5	0.6708	0
40	0.6650	0.3923	1.6	1.2810	0	0.5885	0.3117	1.6	1.2810	0
50	0.6698	0.3340	1.4	0.6633	0	0.6040	0.1831	1.4	0.6633	0
	Algorithme CCP					Algorithme CCUP				
10	0.6274	0.0869	1.1	0.3	0	0.6099	0.07033	1	0	0
15	0.7372	0.2764	1.4	0.8	0	0.7297	0.2975	1.5	1.025	0
20	0.7983	0.3164	1.7	1.005	0	0.8522	0.4064	1.7	1.269	0
30	0.8053	0.2676	1.6	0.8	0	0.8897	0.3101	1.7	0.9	0
40	0.9573	0.4482	1.7	1.269	0	0.9913	0.5765	1.7	1.552	0
50	0.9198	0.3277	1.4	0.9165	0	0.8862	0.2982	1.5	0.922	0

Les algorithmes CC et CCU se comportent de façon très similaires car aucune mise à jour n'a été effectuée.

Les colonnes des algorithmes CCP et CCUP présentent des valeurs supérieures aux deux autres. Ceci n'est pas uniquement dû au nombre de coupes plus élevé, mais effectivement à un supplément de calcul dans le code. D'une part, la recherche des pseudo-sommets voisins les plus proches d'un sommet du polyèdre est un ajout sur l'algorithme CC. D'autre part, le calcul des points voisins du pseudo-sommet ainsi trouvé prend du temps.

En résumé, avec cette catégorie, la mise à jour n'était pas appliquée et la technique des pseudo-sommets n'a pas montré davantage.

5.2.2 Instances $BILD$ de dimension $n = 20$

La colonne de l'algorithme CC (tabl. 5.3) présente un avantage sur la littérature. En effet, aucun échec est remarqué et les instances de densité $D = 50$ sont toutes résolues. En plus, le nombre de coupes est moins grand. Cet avantage est dû au remplacement de la technique du choix aléatoire d'un sommet voisin non dégénéré, considérée dans la littérature, par la technique du choix de meilleur sommet voisin non dégénéré.

Tableau 5.3 Instances $BILD$ de dimension $n = 20$

Densité	Algorithme CC					Algorithme CCU				
	temps	σ_t	coupes	σ_c	échec	temps	σ_t	coupes	σ_c	échec
10	0.991	0.3825	1.6	0.8	0	1.016	0.4363	1.6	0.8	0
15	1.007	0.3345	1.6	0.6633	0	0.9924	0.2723	1.6	0.6633	0
20	1.285	0.6998	2.2	1.327	0	1.33	0.7534	2.2	1.327	0
30	1.969	1.783	3.7	3.743	0	1.917	1.72	3.7	3.743	0
40	2.025	1.631	3.8	3.311	0	2.025	1.619	3.8	3.311	0
50	2.306	1.88	4.6	3.72	0	**2.114**	1.577	**4.1**	2.948	0
	Algorithme CCP					Algorithme CCUP				
10	1.607	0.6217	1.8	1.077	0	1.661	0.7368	1.8	1.077	0
15	1.66	0.5846	1.9	1.136	0	1.751	0.7784	1.8	0.9798	0
20	3.027	2.757	3.5	3.748	0	2.75	2.146	3.4	3.137	0
30	2.414	1.281	**3.1**	2.119	0	2.654	2.032	3.5	3.202	0
40	6.099	5.921	8.5	8.617	0	7.289	7.794	9.9	11.11	0
50	5.565	4.186	7.8	6.063	0	5.149	4.612	**6.9**	6.395	0

Des mises à jour étaient effectuées aux instances de densité $D = 50$ et le temps de calcul ainsi que le nombre de coupes ont diminué dans les algorithmes CCU et CCUP.

Dans cette catégorie d'instances, les coupes associées au pseudo-sommet sont généralement plus nombreuses que celles associées au sommet. Par ce fait, le temps de calcul de CCP et CCUP est plus élevé, hormis que CCP et CCUP sont plus coûteux en terme de compléxité. Une exception se voit avec la catégorie de densité $D = 30$. Ceci peut se reporter aux choix exceptionnels des pseudo-sommets voisin les plus proches.

Avec les améliorations portées à l'algorithme CC, la technique des pseudo-sommets n'a pas encore présenté un vrai avantage. Dans son application, la technique de la mise à jour a rempli son objectif.

5.2.3 Instances $BILD$ de dimension $n = 30$

Les chiffres de la colonne Algorithme CC reste avantageux relativement aux marges numériques vues en littérature (tabl. 5.4). Les échecs sont aussi rencontrés dans les mêmes catégories de densité.

Tableau 5.4 Instances $BILD$ de dimension $n = 30$

Densité	Algorithme CC					Algorithme CCU				
	temps	σ_t	coupes	σ_c	échec	temps	σ_t	coupes	σ_c	échec
10	2.236	0.9606	1.9	0.8307	0	2.237	0.9214	1.9	0.8307	0
15	2.86	1.476	3.3	1.847	0	2.93	1.581	3.3	1.847	0
20	57.09	154.5	**47.1**	**122.8**	0	64.9	180	**44.3**	116	**1**
30	123.6	238.8	74.7	146.6	2	123.7	239.3	75.3	141.1	2
40	199.5	263.7	142.3	190.4	3	193.5	267.7	123.9	162.3	3
50	148.3	229	**95.6**	135.1	**2**	148.4	229.2	96.8	130.8	**2**
	Algorithme CCP					Algorithme CCUP				
10	4.183	3.07	2.4	1.744	0	4.159	2.114	2.6	1.625	0
15	7.578	8.515	6.3	7.537	0	6.925	6.377	5.7	5.797	0
20	73.66	177.3	33.6	63.56	1	71.84	178.4	31.1	61.09	1
30	133.3	236.3	53.5	81.11	2	128.8	237.7	50.1	82.63	2
40	206.1	210.3	96	74.3	2	242.6	246.6	106.5	80.43	3
50	301.5	234.3	**140**	94.77	**3**	265.6	238.5	**128.7**	92.61	**3**

En général, les mises à jour effectuées en CCU et CCUP conduisent à un abaissement dans le nombre de coupe et ensuite dans le temps de résolution. Cependant, dans certains cas, le problème de cumul des coupes est apparu causant un échec de résolution. Nous pouvons le vérifier particulièrement avec **une** instance de densité $D = 20$ dont l'échec a conduit à un élévation dans le temps de calcul, en dépit des diminution de nombre de coupes faites avec les autres instances.

En présence d'échec avec la série d'instances de densité $D = 50$, la technique des pseudo-sommets ne révèle pas d'avantage. Ceci se reporte au problème de cumul des coupes qui se voit en comparant le nombre d'échec de CC et de CCU avec celui de CCP et de CCUP respectivement.

Le gravité du problème de cumul des coupe affecte les améliorations conçues dans notre recherche. Et celui du pire choix d'un pseudo-sommet voisin le plus proche, n'est pas vraiment diagnostiqué dans cette catégorie d'instances.

5.2.4 Instances $BILD$ de dimension $n = 40$

Relativement à la littérature, les résultats de l'algorithme CC sont bons. En plus, la résolution des instances de densité $D = 20$ n'a présenté aucun échec. (tabl. 5.5)

Tableau 5.5 Instances $BILD$ de dimension $n = 40$

Densité	Algorithme CC					Algorithme CCU				
	temps	σ_t	coupes	σ_c	échec	temps	σ_t	coupes	σ_c	échec
10	3.319	1.112	2.4	1.2	0	3.461	1.345	2.4	1.2	0
15	15.51	30.24	12.5	24.33	0	10.39	15.37	8.6	12.78	0
20	48.21	120.7	32.6	76.85	0	67.81	**179.6**	**35.4**	85.22	**1**
30	191	270	103.3	148	3	187.6	272.1	89.7	127.2	3
40	447.8	227.2	167.8	90.48	6	428.4	**234.4**	**171.2**	85.96	6
50	391.5	265.1	146.3	107.8	6	382.3	**274.2**	**155.1**	108	6
	Algorithme CCP					Algorithme CCUP				
10	8.955	7.147	4.5	3.956	0	8.68	4.33	4.4	2.871	0
15	70.87	178.4	32.7	78.56	1	40.92	89.83	20.5	41.94	0
20	85.12	175.7	31.7	47.71	1	84.11	177.6	29.5	48.07	1
30	224.5	259.7	67	67.78	3	196.5	268.8	54.1	64.65	3
40	565.7	114	148.7	35.18	9	578.2	84.57	169.1	23.52	9
50	470	224.5	122	55.52	7	461.4	232.4	129	58.51	7

L'algorithme CCU est préférable à CC dans les cas où $D = d$ avec $d \in \{15, 30, 40\}$, car le nombre de coupe ainsi que le temps de calcul a diminué approximativement du tiers.

En regardant diagonalement le tableau (CC vers CCUP), les chiffres n'indiquent pas d'améliorations dans les calculs numériques.

5.2.5 Instances $BILD$ de dimension $n >= 50$

Les tableaux 5.6 et 5.7, montrent que les calculs concernant les instances de densité $D = 10$, dans la colone de l'algorithme CC, restent dans les marges numériques vues en littérature.

La technique de mise à jour marque davantage en abscence d'échec avec les instances $BILD$ de dimension $n = 60$ et $d = 10$.

La technique des pseudo-sommets se voit désavantageuse en abscence d'échec.

Tableau 5.6 Instances $BILD$ de dimension $n = 60$

Densité	Algorithme CC					Algorithme CCU				
	temps	σ_t	coupes	σ_c	échec	temps	σ_t	coupes	σ_c	échec
10	13.78	**22.52**	**8.4**	13.77	0	14.09	22.43	8.3	13.81	0
15	184.3	210.4	75.9	81.62	1	187.6	226	70.8	77.45	2
20	309.8	294.4	98.2	98.02	5	308.7	295.4	100.4	94.36	5
30	317.5	286.9	109.7	104	5	313.8	291.6	106.2	96.43	5
40	399	260.5	120.9	83.73	5	383.5	255.3	123.7	82.15	4
50	471.2	235.3	137.3	78.02	7	460	240.1	143.1	75.09	7
	Algorithme CCP					Algorithme CCUP				
10	77.45	173.6	27.1	56.8	0	36.47	60.96	14.2	22.35	0
15	367.6	283.8	91.9	73.44	5	338.3	280	83.2	65.82	5
20	321.7	284.1	65.4	51.29	5	323.2	283.4	67.8	51.46	5
30	366.2	259.3	80.1	53.89	5	378.7	280.9	82.5	58.41	6
40	428.6	269.4	84.3	52.11	7	428.3	272.5	88.5	54.97	7
50	485.8	238	92.2	45.74	8	486.9	240.1	97.6	48.2	8

Tableau 5.7 Instances $BILD$ de dimension $n = 60$

Densité	Algorithme CC					Algorithme CCU				
	temps	σ_t	coupes	σ_c	échec	temps	σ_t	coupes	σ_c	échec
10	10.3	11.94	**4.4**	5.333	0	10.66	12.86	4.4	5.643	0
15	170.5	223.3	53.7	72.64	2	207.3	263.2	59.1	71.88	3
20	232.6	264.6	69.4	81.04	3	236.7	272.8	62.7	71.26	3
30	490.7	228.5	115.5	62.98	8	489	232.6	120.9	57.64	8
40	510	196.1	104.5	34.85	8	511.4	196.2	118.5	40.54	8
50	466	215.5	98.2	48.57	7	454.8	232.7	105.8	50.87	7
	Algorithme CCP					Algorithme CCUP				
10	59.59	124.1	16.9	34.24	0	53.43	90.8	15.5	24.25	0
15	315.2	260.2	59.2	43.65	4	337.7	272.7	64.3	46.33	5
20	326.1	281.3	55.2	45.68	5	269.5	278	45.4	40.61	4
30	502.9	209.5	75	29.15	8	509.6	200.6	80	27.56	8
40	553.2	159.6	81.4	25.54	9	550.3	170.9	86.7	26.08	9
50	561.6	131.6	84.7	20.04	9	558.9	144.7	92.3	22.1	9

5.3 Instances $BILD$ résolues

Dans cette section, nous discutons des instances qui ont été résolues par l'algorithme CC. En inspectant les résultats numériques, nous observons que chacune des instances a requis

exactement le même nombre de coupes pour les algorithmes CC et CCU sauf pour deux d'entre-elles, tel que décris dans le tableau 5.8.

Tableau 5.8 Instances $BILD$ résolues

Dimension	Densité	Algorithme CC			Algorithme CCU		
		temps	coupes x	coupes u	temps	coupes x	coupes u
20	50	5.085	6	6	3.512	4	3
50	30	29.55	10	10	24.31	8	7

Ces deux instances ont été résolues par l'algorithme CCU en requérant 5 coupes de moins qu'avec l'algorithme CC. Cependant, les techniques impliquant les pseudo-sommets ont parfois augmenté le nombre de coupes.

5.4 Discussion

Nous avons proposé deux variantes à l'algorithme de coupes de concavité et les avons testés sur deux ensembles de problèmes, l'un tiré de la littérature, et l'autre généré aléatoirement.

Pour les problèmes de la littérature, la meilleure stratégie s'est avérée être la combinaison des deux variantes proposées. Le problème de dégénérescence fut rencontré dans quelques unes de ces instances, et fut éliminé par les coupes enracinées aux pseudo-sommets.

La démarcation est moins évidente pour les problèmes générés aléatoirement. Lorsque les problèmes sont résolus dans le temps alloué, le nombre de coupes requise augmente occasionnellement lorsqu'elles sont calculées à partir d'un pseudo-sommet. Mais, avec la technique de mise à jour, ce nombre est demeuré inchangé dans la tous les cas, à l'exception de deux instances où il a diminué de cinq.

De plus, la complexité inhérente à ces variantes algorithmiques fait en sorte que le nombre de coupes généré dans le temps prescrit diminue.

En conclusion, il semblerait que la technique des pseudo-sommets n'apportent pas de notables améliorations dans les cas générés aléatoirement, contrairement à la technique de la mise à jour, mais réduit le nombre de coupes et le temps de calcul dans les instances tirées de la littérature.

CHAPITRE 6

CONCLUSION

6.1 Synthèse des travaux

Ce projet présente de nouveaux modes de déploiement des coupes de concavité afin de réduire itérativement les ensembles X et U pour chercher la solution optimale des problèmes $BILD$. Ces modes sont comme suit :

- mise à jour des coupes
- enracinement des coupes aux pseudo-sommets
- réflexion de la coupe de concavité

La réduction dynamique des ensembles X et U s'est révélée importante durant la résolution de nos modèles numériques. Cette réduction vient à la suite d'une mise à jour des coupes, produite après qu'une amélioration dans la valeur objectif de $BILD$ soit faite.

Dans la littérature, les coupes qui étaient enracinées aux sommets des polyèdres X et U de $BILD$, sont enracinées, dans notre recherche, aux pseudo-sommets de ces polyèdres.

La technique de réflexion de la coupe de concavité était exécutée par la résolution d'un problème linéaire dont la fonction objectif est l'équation sans second membre des dernières coupes de concavité introduites respectivement aux ensembles X et U de $BILD$.

La technique de traitement de la dégénérescence avec les pseudo-sommets, inspirée des travaux d'Alarie et *al.* [4], était aussi développée dans notre étude et appliquée avec succès sur les instances $BILD$ tirées de la littérature.

La technique de recherche d'un pseudo-sommet voisin le plus proche est conçue avec le procédé inverse du Simplex

$$\max\{ratios < 0\}$$

qui fonctionne pareillement à l'outil mathématique

$$\min\{ratios > 0\}.$$

Les tests numériques sur des instances $BILD$ de la littérature ont porté fruits. Elles ont prouvé que les nouveaux modes de déploiement de coupes, combinés ensembles, diminue le nombre de coupe et réduit le temps de calcul. En plus, elles ont montré l'efficacité de nos stratégies dans le traitement de la dégénérescence aux sommets des polyèdre.

Les tests sur des instances $BILD$ générées aléatoirement ont répondu partiellement à nos attentes. La technique de la mise à jour a toujours requis un nombre de coupe inférieur ou égal aux méthodes précédentes de la litérature. Les techniques impliquant les pseudo-sommets n'ont pas approté d'amélioration.

6.2 Limitations des idées proposées

Les idées de déploiement conçus dans ce projet devraient traiter toutes les instances de $BILD$. Le mauvais choix des pseudo-sommets et le problème de cumul des coupes limitent la taille des problèmes que nous pouvons résoudre.

Les sorties de ces algorithmes affichent en détails les résultats numériques. En effet, des commentaires et des décisions logiques avec explications suivent chaque étape de calcul numérique. Ceci permet la lecture, la compréhension et le suivie du processus de résolution par l'usager. Ainsi, il leur permet la perception de concevoir des améliorations futures.

6.3 Améliorations futures

De nombreux aspects de ce projet laissent place à l'amélioration.

Avec une étude approfondie de choix des pseudo-sommets, nos nouveaux modes de déploiements devraient améliorer les résultats avec n'importe quelles instances $BILDs$ générées aléatoirement. Après la révue des résultats des instances $BILD$ de la littérature, nous suggérons que cette étude débute avec des instantes $BILD$ dont leurs dimensions (n_v, n_u, n_x, n_y) doivent être plus grande ou égale à quatre. Cette suggestion est prévu du fait que le déploiement des coupes avec les pseudo-sommets ont montré davantage avec des dimensions plus petites

Il pourrait également être intéressant d'enraciner les cônes polyèdraux à plusieurs pseudo-sommets et introduire ensuite des coupes de concavité valides. Il pourrait éventuellement avoir

moins de coupes mais avec un calcul important pour construire ces cônes.

Finalement, il aurait été intéressant de valider ces nouveaux modes de déploiement sur d'autres PO concaves dont les domaines polyédriques.

RÉFÉRENCES

[1] Espace vectoriel normé @ONLINE http ://fr.wikipedia.org/wiki/espace_normé.

[2] Théorie de la complexité (informatique théorique) @ONLINE http ://fr.wikipedia.org/wiki/théorie_de_la_complexité_(informatique_théorique).

[3] F. Al-Khayyal et X. Ding. Accelerating convergence of cutting plane algorithms for disjoint bilinear programming. *Journal of Global Optimization*, vol. 38(no. 3) :pp. 421–436, 2007.

[4] S. Alarie, C. Audet, B. Jaumard, et G. Savard. Concavity cuts for disjoint bilinear programming. *Mathematical programming*, vol. 90(no. 2) :pp. 373–398, 2001.

[5] C. Audet. *Optimisation globale structurée : propriétés, équivalences et résolution*. Thèse de doctorat, École Polytechnique de Montréal, 1997.

[6] C. Audet, P. Hansen, B. Jaumard, et G. Savard. On the Linear Maxmin and Related Programming Problems. Multilevel Optimization : Algorithms and Applications. *Kluwer Acad. Publ., Dordrecht*, vol. 20 of Nonconvex Optimization and Applications(no. 2) :pp. 181–208, 1998.

[7] C. Audet, P. Hansen, B. Jaumard, et G. Savard. A Symmetrical Linear Maxmin Approach to Disjoint Bilinear Programming. *Mathematical programming*, vol. 85 :pp. 573–592, 1999.

[8] A. Bagchi et B. Jaumard. An alogorithm for quadratic zero-one program. *Naval Research Logistics*, vol. 37 :pp. 527–538, 1990.

[9] S.G. Bali. *Minimization of a concave function on a bounded convex polyhedron*. Thèse de doctorat, University of California at Los Angeles, 1973.

[10] E.M.L. Beale et R.E. Small. Mixed Integer Programming by a Branch and Bound Technique. In *Proceedings of the 3rd IFIP Congress 1965*, volume vol. 2, pages pp. 450–451, 1965.

[11] H.P. Benson. *Handbook of Global Optimization (éditeurs R. Horst et P.M. Pardalos)*, chapter Concave Minimization : Theory, Application and Algorithms, pages pp. 43–148. Kluwer Academic Publishers, Boston, 1995.

[12] H.P. Benson. Deterministic algorithms for constrained concave minimization : A unified critical survey. *Naval Research Logistics (NRL)*, vol. 43(no. 6) :pp. 765–795, 1996.

[13] A.V. Cabot. Variations on a Cutting Plane Method for Solving Concave Minimization Problems with Linear Constraints. *Naval Research Logistics Quarterly*, vol. 21 :pp. 265–274, 1974.

[14] P.H. Calamai, J.J. Júdice, et L.N. Vicente. Generation of disjointly constrained bilinear programming test problems. *Comput. Optim. Appl.*, vol. 13 :pp. 299–306, 1992.

[15] G. B. Dantzig. *Linear programming and extensions.* Princeton university press, 1965.

[16] J.E. Falk. A Linear Max-Min Problem. *Mathematical Programming*, vol. 5 :pp. 169–188, 1973.

[17] F. Glover. Convexity cuts et cut search. *Operations Research*, vol. 21 :pp. 123–134, 1973.

[18] P. Hansen, B. Jaumard, et G. Savard. New branch and bound rules for linear bilevel programming. *Mathematical programming*, vol. 13 :pp. 1194–1217, 1992.

[19] R. Horst. An algorithm for nonconvex programming problems. *Mathematical Programming*, vol. 10(no. 1) :pp. 312–321, 1976.

[20] R. Horst et P.M. Pardalos. *Handbook of global optimization*, chapter Concave minimization : Theory, applications and algorithms, pages pp. 43–148. Kluwer Academic Publishers, 1996.

[21] R. Horst et H. Tuy. *Global Optimization (Deterministic Approaches)*, chapter Some important classes of global optimization problems, pages pp. 3–51. Springer-Verlag, 1996.

[22] G.P. Karatzas et G.F. Finder. The solution of groundwater quality management problems with a nonconvex feasible region using a cutting plane optimization technique. *Water Resources Research*, vol. 32(no. 4) :pp. 1091–1100, 1996.

[23] H. Konno. *Bilinear Programming : Part II. Applications of Bilinear Programming.* Department of Operations Research, Stanford University, 1971.

[24] H. Konno. A cutting plane algorithm for solving bilinear programs. *Mathematical programming*, vol. 11(no. 1) :pp. 14–27, 1976.

[25] D. G. Luenberger. *Bilinear Programming : Part II. Applications of Bilinear Programming.* Addison-Wesley Publishing Company, 1989.

[26] A.D. Lutzenko et A.V. Martynov. Minimax Solutions of Problems in Linear and Quadratic Programming. *Izv. Akad. Nauk SSSR, Tekhn. Kibernitika 2*, vol. 8(no. 2) :pp. 22–27, 1968.

[27] A. Majthay et A. Whinston. Quasi-concave minimization subject to linear constraints. *Discrete Mathematics*, vol. 9(no. 1) :pp. 35–59, 1974.

[28] C. Meyer. *Algorithmes coniques pour la minimisation quasiconcave.* Thèse de doctorat, École Polytechnique de Montréal, 1998.

[29] H. Mills. Equilibrium Points in Finite Games. *Journal of the Society for Industrial and Applied Mathematics*, vol. 8(no. 2) :pp. 397–402, 1960.

[30] M. Minoux. *Programmation mathématique, théorie et algorithmes, tome 1.* Dunod, 1983.

[31] J.F. Nash. Noncooperative Games. *Annals of Mathematics*, vol. 54(no. 2) :pp. 286–295, 1951.

[32] P.M. Pardalos et J.B. Rosen. Methods for Global Concave Minimization : A Bibliographical Survey. *SIAM Review*, vol. 28(no. 3) :pp. 367–379, 1986.

[33] P.M. Pardalos et G. Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, vol. 7(no. 1) :pp. 33–35, 1988.

[34] M. Porembski. How to extend the concept of convexity cuts to derive deeper cutting planes. *Journal of Global Optimization*, vol. 15(no. 4) :pp. 371–404, 1999.

[35] H.D. Sherali et C.M. Shetty. A finitely convergent algorithm for bilinear programming-problems using polar cuts and disjunctive face cuts. *Mathematical programming*, vol. 19(no. 1) :pp. 14–31, 1980.

[36] C.M. Shetty et H. Vaish. A cutting plane algorithm for the bilinear programming problem. *Naval Research Logistics Quarterly*, vol. 24(no. 1) :pp. 83–94, 1977.

[37] T.V. Thieu. A note on the solution of bilinear-programming problems by reduction to concave minimization. *Mathematical programming*, vol. 41(no. 2) :pp. 249–260, 1988.

[38] T.V. Thieu. Convex analysis and global optimization. *Soviet Mathematics Doklady Academii Nauk SSSR 5*, 1998.

[39] H. Tuy. Concave programming under linear constraints. *Soviet Mathematics Doklady Academii Nauk SSSR 5*, pages pp. 1437–1440, 1964.

[40] P.B. Zwart. Computational aspects on the use of cutting planes in global optimization. In *Proceedings of the 1971 26th annual conference*, pages pp. 457–465, 1971.

[41] P.B. Zwart. Global maximization of a convex function with linear inequality constraints. *Operations Research*, vol. 21(no. 1) :pp. 602–609, 1973.

ANNEXE A

ALGORITHME CCUP

Nous présentons ici le pseudo-code de notre algorithme CCUP, qui contient comme cas particulier l'algorithme de base CC [4], ainsi que les deux contributions de ce travail, soient la mise à jour dynamique CCU et l'enracinement aux pseudo-sommets CCP.

A.1 Critères d'arrêt

Itérer jusqu'à ce que l'une des cinq conditions suivantes soit satisfaite :

C1 Une solution de valeur objectif négative et non-bornée est trouvée

C2 L'une des ensembles X ou U est réduit à l'ensemble vide : la meilleur solution à date est la solution optimale.

C3 Plus que m_x essaies d'ajout de coupe à X, et plus que m_u essais dans U

C4 La profondeur des dernières $k^{ième}$ coupes, introduites consécutivement, est plus petite que ε_x et celles des $k^{ième}$ introduites consécutivement dans U est plus petite que ε_u.

C5 Aucune coupes introduite à X et à U à la dernière itération.

A.2 Algorithme $CCUP$

Étape a. Recherche d'un point extrême réalisable (x^0, u^0).

S'il n'y a aucune coupe dans X et U, initialiser $(x^0, u^0) \leftarrow 0$. Autrement soit (x^0, u^0) la réfléxion du (x', u') à travers les dernière coupes introduites.

Étape b. Recherche d'une solution réalisable (x', u').

Effectuer deux séries d'itérations de "mountain climbing" : soit $(x', u', \gamma') \leftarrow MC(x^0)$ et $(x'', u'', \gamma'') \leftarrow MC(u^0)$. Selectionner celle de plus petite valeur : si $\gamma'' < \gamma'$, affecter (x'', u'', γ'') à (x', u', γ').

Étape c. Incumbent Update.

Si $\gamma' < \hat{\gamma}$ affecter (x', u', γ') à $(\hat{x}, \hat{u}, \hat{\gamma})$. Mise-à-jour des coupes. Arrêter si $\hat{\gamma} = -\infty$: la valeur optimal du $(BILD)$ est non bornée. Aller à l'étape **d**.

Étape d. Amélioration de la solution réalisable courante dans X.

Appliquer le cas échéant.

Cas **d1 :** x' est non dégénéré :

S'il y a un meilleur voisin x'' de x' tel que $f(x'') < f(x')$ remettre $(x', u', \gamma') \leftarrow MC(x'')$ et retourne à l'étape **c**. Sinon, procéder à trouver un pseudo-sommet x'' voisin de x'.

d1′ : En cas d'existence, soit x'' un plus proche pseudo-sommet voisin de x'. Adopter les points voisins de x'' comme choix de direction, affecter x'' à x' et aller à l'étape **e**.

d1″ : En cas d'inexistence, aller à l'étape **e** avec le sommet non dégénéré x' et ses sommets voisins comme choix de direction.

Cas **d2 :** x' est dégénéré :

Si $\gamma' = \hat{\gamma}$, n'ajouter aucune coupe à X et aller à l'étape **e**. Sinon, appeler simplex pour une variable hors-base. S'il est possible, par un pivot simple de simplex, d'obtenir un point non-dégénéré x'' voisin de x', appliquer l'une des cas échéants suivants :

d2′ : x'' est un pseudo-sommet :

Chercher, s'il est possible, un sommet non dégénéré et fini x''' voisin de x'' tel que $x''' \neq x'$. Appliquer le cas échéant :

i- Si $f(x''') < \hat{\gamma}$, remettre $(x', u', \gamma') \leftarrow MC(x''')$ et retourne à l'étape **c**.

ii- Si $f(x''') \geq \hat{\gamma}$, considérer les points voisins de x'' et les adopter comme choix de direction. Affecter x'' à x' et aller à l'étape **e**.

d2″ : x'' est un sommet :

Appliquer le cas échéant :

i- Si $f(x'') < \hat{\gamma}$, remettre $(x', u', \gamma') \leftarrow MC(x'')$ et retourne à l'étape **c**.

ii- Si $f(x'') \geq \hat{\gamma}$, considérer les sommets voisins de x''. S'il y a un sommet x''' voisin de x'' tel que $f(x''') < \hat{\gamma}$, remettre $(x', u', \gamma') \leftarrow MC(x''')$ et retourne à l'étape **c**. Autrement, adopter les sommets voisins de x'' comme choix de direction, affecter x'' à x' et aller à l'étape **e**.

Étape e. Amélioration de la solution réalisable courante dans U. Appliquer le cas échéant.

Cas **e1 :** u' est non dégénéré :

S'il y a un meilleur voisin u'' de u' tel que $g(u'') < g(u')$ remettre $(x', u', \gamma') \leftarrow MC(u'')$ et retourne à l'étape **c**. Sinon, procéder à trouver un pseudo-sommet u'' voisin de u'.

e1′ : En cas d'existence, soit u'' un plus proche pseudo-sommet voisin de u'. Adopter les points voisins de u'' comme choix de direction, affecter u'' à u' et aller à l'étape **f**.

e1″ : En cas d'inexistence, aller à l'étape **f** avec le sommet non dégénéré u' et ses sommets voisins comme choix de direction.

Cas **e2 :** u' est dégénéré :

Si $\gamma' = \hat{\gamma}$, n'ajouter aucune coupe à U et aller à l'étape **f**. Sinon, appeler simplex pour une variable hors-base. S'il est possible, par un pivot simple de simplex, d'obtenir un point non dégénéré u'' voisin de x', appliquer l'une des cas échéants suivants :

e2′ : u'' est un pseudo-sommet :

Chercher, s'il est possible, un sommet non-dégénéré et fini u''' voisin de x'' tel que $u''' \neq u'$. Appliquer le cas échéant :

i- Si $g(u''') < \hat{\gamma}$, remettre $(x', u', \gamma') \leftarrow MC(u''')$ et retourne à l'étape **c**.

ii- Si $g(u''') \geq \hat{\gamma}$, considérer les points voisins de u'' et les adopter comme choix de direction. Affecter u'' à u' et aller à l'étape **f**.

e2″ : u'' est un sommet :

Appliquer le cas échéant :

i- Si $g(u'') < \hat{\gamma}$, remettre $(x', u', \gamma') \leftarrow MC(u'')$ et retourne à l'étape **c**.

ii- Si $g(u'') \geq \hat{\gamma}$, considérer les sommets voisins de u''. S'il y a un sommet u''' voisin de u'' tel que $g(u''') < \hat{\gamma}$, remettre $(x', u', \gamma') \leftarrow MC(u''')$ et retourne à l'étape **c**. Autrement, adopter les sommets voisins de u'' comme choix de direction, affecter u'' à u' et aller à l'étape **f**.

Étape f. Ajout des coupes de concavité à X assignée à x'.

Si aucune coupe est requise à être ajoutée, alors aller à l'étape **g** ($\gamma' = \hat{\gamma}$). Autrement, calculer les $\hat{\gamma}$-extensions dans les directions des sommets choisies pour x'. Ajouter à X la coupe de concavité $\pi_0^T x = \pi$ qui passe par toutes les $\hat{\gamma}$-extensions.

Arrêter si $X = \emptyset$

Étape g. Ajout des coupes de concavité à U assignée à u'.

Si aucune coupe est requise à être ajoutée, alors arrêter ($\gamma' = \hat{\gamma}$). Autrement, calculer les $\hat{\gamma}$-extensions dans les directions des sommets choisies pour u'. Ajouter à U la coupe de concavité $\tau_0^T x = \tau$ qui passe par toutes les $\hat{\gamma}$-extensions.

Arrêter si $U = \emptyset$

ANNEXE B

INSTANCES $BILD$ DE LA LITTÉRATURE

Nous citons dans cet annexe les instances $BILD$ tirées de la littérature. Les noms de ces instances désignent les dimensions des matrices composantes du problème $BILD$ [éqt. 1.1] :n_v, n_u, n_x et n_y. D'où, le format des noms est comme suit : $n_v n_u n_x n_y$.

B.1 Instance $BILD$ 3223 (Al-Khayyal [3] 2007)

$$c = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad A = \begin{pmatrix} 1 & 4 \\ 4 & 1 \\ 3 & 4 \end{pmatrix} \in \mathbb{R}^{3\times 2} \quad ; \quad a = \begin{pmatrix} 8 \\ 12 \\ 12 \end{pmatrix} \in \mathbb{R}^3$$

$$d = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad B = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 3} \quad ; \quad b = \begin{pmatrix} 8 \\ 8 \\ 5 \end{pmatrix} \in \mathbb{R}^3$$

$$Q = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

B.2 Instance $BILD$ 4444 (Al-Khayyal [3] 2007)

$$c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^4 \quad ; \quad A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{pmatrix} \in \mathbb{R}^{4\times 4} \quad ; \quad a = \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \end{pmatrix} \in \mathbb{R}^4$$

$$d = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{pmatrix} \in \mathbb{R}^{4\times 4} \quad ; \quad b = \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \end{pmatrix} \in \mathbb{R}^4$$

$$Q = \begin{pmatrix} 3 & -1 & 0 & -1 \\ -1 & 4 & -2 & 0 \\ 0 & -2 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{pmatrix} \in \mathbb{R}^{4\times 4}$$

B.3 Instance $BILD$ 6224 (Alarie et *al.* [4] 2001)

$$c = \begin{pmatrix} 5 \\ 6 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad A = \begin{pmatrix} 2 & -1 \\ 1 & -1 \\ 0 & -1 \\ -1 & -1 \\ -3 & 2 \\ -1 & 4 \end{pmatrix} \in \mathbb{R}^{6\times 2} \quad ; \quad a = \begin{pmatrix} 0 \\ -1 \\ -4 \\ -12 \\ -16 \\ -2 \end{pmatrix} \in \mathbb{R}^6$$

$$d = \begin{pmatrix} 10 \\ 15 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad B = \begin{pmatrix} 47 & 19 & 1 & -1 \\ -16 & -9 & -3 & 4 \end{pmatrix} \in \mathbb{R}^{2\times 4} \quad ; \quad b = \begin{pmatrix} -16 \\ -16 \\ -16 \\ -2 \end{pmatrix} \in \mathbb{R}^4$$

$$Q = \begin{pmatrix} 0 & 3 \\ -2 & -4 \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

B.4 Instance $BILD$ 6225 modifiée (Alarie et *al.* [4] 2001)

$$c = \begin{pmatrix} 5 \\ 6 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad A = \begin{pmatrix} 2 & -1 \\ 1 & -1 \\ 0 & -1 \\ -1 & -1 \\ -3 & 2 \\ -1 & 4 \end{pmatrix} \in \mathbb{R}^{6\times 2} \quad ; \quad a = \begin{pmatrix} 0 \\ -1 \\ -4 \\ -12 \\ -16 \\ -2 \end{pmatrix} \in \mathbb{R}^6$$

$$d = \begin{pmatrix} 10 \\ 15 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad Q = \begin{pmatrix} 0 & 3 \\ -2 & -4 \end{pmatrix} \in \mathbb{R}^{2\times 2} \quad ; \quad b = \begin{pmatrix} -16 \\ -16 \\ -16 \\ -2 \\ 1 \end{pmatrix} \in \mathbb{R}^5$$

$$B = \begin{pmatrix} 47 & 19 & 1 & -1 & 1 \\ -16 & -9 & -3 & 4 & 2 \end{pmatrix} \in \mathbb{R}^{2\times 5}$$

B.5 Instance $BILD$ 7226 modifiée (Alarie et *al.* [4] 2001)

$$c = \begin{pmatrix} 5 \\ 6 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad A = \begin{pmatrix} 2 & -1 \\ 1 & -1 \\ 0 & -1 \\ -1 & -1 \\ -3 & 2 \\ -1 & 4 \\ 0.42 & -2.12 \end{pmatrix} \in \mathbb{R}^{7\times 2} \quad ; \quad a = \begin{pmatrix} 0 \\ -1 \\ -4 \\ -12 \\ -16 \\ -2 \\ -0.15 \end{pmatrix} \in \mathbb{R}^6$$

$$d = \begin{pmatrix} 10 \\ 15 \end{pmatrix} \in \mathbb{R}^2 \quad ; \quad Q = \begin{pmatrix} 0 & 3 \\ -2 & -4 \end{pmatrix} \in \mathbb{R}^{2\times 2} \quad ; \quad b = \begin{pmatrix} -16 \\ -16 \\ -16 \\ -2 \\ 1 \\ -1.17 \end{pmatrix} \in \mathbb{R}^6$$

$$B = \begin{pmatrix} 47 & 19 & 1 & -1 & 1 & -0.17 \\ -16 & -9 & -3 & 4 & 2 & 0.34 \end{pmatrix} \in \mathbb{R}^{2\times 6}$$

B.6 Instance $BILD$ 6556 (Alarie et *al.* 2001)

$$c = \begin{pmatrix} 21 \\ 22 \\ 22.5 \\ 23.5 \\ 23.75 \end{pmatrix} \in \mathbb{R}^5 \, ; \, d = \begin{pmatrix} -40 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \in \mathbb{R}^6 \, ; \, a = \begin{pmatrix} -40 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \in \mathbb{R}^6 \, ; \, b = \begin{pmatrix} -40 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \in \mathbb{R}^6$$

$$A = \begin{pmatrix} -20 & -12 & -11 & -7 & -4 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{6\times 5} \, ; \, Q = \begin{pmatrix} -20 & -12 & -11 & -7 & -4 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{6\times 5}$$

$$B = \begin{pmatrix} -20 & -1 & 0 & 0 & 0 & 0 \\ -12 & 0 & -1 & 0 & 0 & 0 \\ -11 & 0 & 0 & 0 & 0 & 0 \\ -7 & 0 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{5\times 6}$$

B.7 Instance $BILD$ 7667 (Alarie et *al.* 2001)

$$c = \begin{pmatrix} -5.25 \\ -3.75 \\ -1.75 \\ -1.25 \\ -0.75 \\ -5 \end{pmatrix} \in \mathbb{R}^6 \quad ; \quad d = \begin{pmatrix} -6.5 \\ -20 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \in \mathbb{R}^7 \quad ; \quad a = \begin{pmatrix} -6 \\ -20 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \in \mathbb{R}^7 \quad ; \quad b = \begin{pmatrix} -6.5 \\ -20 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \in \mathbb{R}^7$$

$$A = \begin{pmatrix} -6 & -3 & -3 & -2 & -1 & 0 \\ -10 & 0 & -10 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} \in \mathbb{R}^{7\times 6} \; ; \; Q = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

$$B = \begin{pmatrix} -6 & -10 & -1 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & -1 & 0 & 0 & 0 \\ -3 & -10 & 0 & 0 & -1 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{6\times 7}$$