Université de Montréal

Énumération des rayons extrêmes d'un cône et applications en minimisation concave

par
Éric Springuel
Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Docteur ès sciences (Ph.D.)
en informatique

Novembre 2005

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée :

Énumération des rayons extrêmes d'un cône et applications en minimisation concave

présentée par :

Éric Springuel

a été évaluée par un jury composé des personnes suivantes :

Professeur Patrice Marcotte, président-rapporteur

Professeur Bernard Gendron, directeur de recherche

Professeur Patrick Soriano, co-directeur de recherche

Professeur Jacques Ferland, membre du jury

Professeur David Avis, examinateur externe

Thèse acceptée le 12 janvier 2006

Sommaire

Nous nous intéressons dans le cadre de cette thèse à l'énumération des rayons extrêmes d'un cône polyédral pointé de pleine dimension dans $\Re^n$ à partir de sa représentation sous forme d'hyperplans. Lorsqu'un tel cône est défini par plus de n hyperplans utiles à sa description, il existe plusieurs combinaisons d'hyperplans dont seulement certaines d'entre elles décrivent un des rayons extrêmes et le point extrême correspondant à la pointe du cône est alors dit fortement dégénéré. L'ensemble des algorithmes pour résoudre ce problème sont classés relativement à deux axes principaux : 1) l'approche d'énumération, soit selon un algorithme de balayage d'un graphe, soit selon un algorithme constructif; 2) le traitement de la dégénérescence, soit en produisant toutes les faces de dimension inférieure à n, soit en perturbant la représentation sous forme d'hyperplans, soit en conservant l'information sur l'adjacence des rayons extrêmes.

Nous nous intéressons aussi dans le cadre de cette thèse à l'application de l'énumération des rayons extrêmes d'un cône polyédral pointé à certains algorithmes déterministes de minimisation concave sur un polyèdre. À l'exception de tous les algorithmes de séparation-évaluation progressive dit simpliciaux ou rectangulaires, tous les algorithmes déterministes de minimisation concave sur un polyèdre dépendent fortement de l'énumération des rayons extrêmes d'un cône polyédral pointé sur un point extrême du polyèdre.

Les principales contributions de cette thèse se résument comme suit. Dans un premier temps, nous avons utilisé des principes complémentaires développés dans des contextes différents dans le but de résoudre plus efficacement le problème de l'énumération des rayons extrêmes d'un cône polyédral pointé à partir de sa représentation sous forme d'hyperplans. En effet, nous avons présenté un nouvel algorithme de balayage d'un graphe perturbant la représentation sous forme d'hyperplans, en se basant sur des concepts complémentaires issus de la programmation mathématique et de la géométrie algorithmique. Par la suite, nous avons identifié dans le contexte de certains algorithmes déterministes de minimisation concave sur un polyèdre, l'algorithme le plus performant pour traiter la dégénérescence forte lors de l'énumération des rayons extrêmes d'un cône polyédral pointé. Plus précisément, nous avons comparé les performances de notre nouvel algorithme de balayage d'un graphe à celles du plus efficace des algorithmes constructifs de type double description pour traiter la dégénérescence forte. Finalement, nous avons développé un algorithme déterministe de minimisation concave

sur un polyèdre, qui exploite l'algorithme le plus efficace identifié précédemment pour énumérer les rayons extrêmes d'un cône polyédral pointé. Plus spécifiquement, nous avons présenté un nouvel algorithme de division conique simple et facile à implanter qui détermine un minimum global exact en un temps fini.

Mots clés : énumération, cône polyédral pointé, dégénérescence, minimisation concave, division conique.

Summary

In this Thesis, we study the problem of enumerating the extreme rays of a full dimensional polyhedral pointed cone lying in $\Re^n$ given its hyperplane representation. When such a cone is defined by more than n hyperplanes, all necessary to its description, there exist many combinations of hyperplanes of which only few describe one of the extreme rays, and the extreme point on which the cone is pointed on is said to be strongly degenerated. All solution algorithms for this problem are classified relatively to two general axes : 1) the enumeration approach, which is either by a graph traversal algorithm or by a constructive algorithm; 2) the treatment of degeneracy, which is either by producing the face lattice, by perturbing the hyperplane representation or by maintaining adjacency information of the extreme rays.

We also study in this Thesis the application of the enumeration of the extreme rays of a polyhedral pointed cone to deterministic algorithms for concave minimization problems over a polyhedron. With the exception of all simplicial and rectangular branch and bound algorithms, all deterministic algorithms for concave minimization are heavily dependent upon the enumeration of the extreme rays of the polyhedral cone pointed on an extreme point of the polyhedron.

The main contributions of this Thesis are summarized as followed. First, we use complementary principles developed in different contexts in order to enumerate more efficiently the extreme rays of a polyhedral pointed cone given its hyperplane representation. To this end, we introduce a new graph traversal algorithm based on a perturbation technique which uses concepts developed in mathematical programming and computational geometry. Second, we identify, in the context of concave minimization over a polyhedron, the most efficient algorithm treating strong degeneracy for the enumeration of the extreme rays of the polyhedral pointed cone. More specifically, we compare the performances of our new graph traversal algorithm to the most efficient type of double description constructive algorithm. Third, we develop a deterministic concave minimization algorithm which exploits the most efficient extreme ray enumeration algorithm for a polyhedral pointed cone, as previously identified. More precisely, we present a new simple conical splitting algorithm easy to implement and finitely converging to an exact global minimum.

Keywords : enumeration, polyhedral pointed cone, degeneracy, concave minimization, conical splitting.

Table des matières

Liste des tableaux

Remerciements

Je désire dans un premier temps remercier mes directeurs de recherche, les professeurs Bernard Gendron et Patrick Soriano, pour leur soutien financier et leur aide au niveau de la rédaction des articles. Je tiens également à remercier le FCAR et la Banque Nationale du Canada pour leur soutien financier.

Je profite de cette occasion pour remercier chaleureusement mes collègues et amis, Serge Bisaillon et François Guertin, sans qui aucun résultat numérique ne serait disponible. Voyez dans ce témoignage ma profonde gratitude pour toute l'aide que vous m'avez apportée. Je remercie également mes amis, Walter Rei et Nikolaj Van Omme, pour leur soutien de tous les instants.

Finalement, tout mon respect va à Catheline qui a su endurer mes états d'âme lors de la réalisation de cette thèse et sans qui je n'aurais jamais eu la chance d'avoir le temps de réaliser ce rêve. Je tiens également à saluer les deux soleils de ma vie, Laurie-Annick et Sandrine, qui me poussent, sans le savoir, à franchir ces étapes. Je réserve mon mot de la fin à une pensée toute particulière pour ma grande soeur défunte Annick.

À Catheline, Laurie-Annick et Sandrine

Chapitre 1

Introduction

Nous nous intéressons dans le cadre de cette thèse à l'énumération des rayons extrêmes d'un cône polyédral pointé à partir de sa représentation sous forme d'hyperplans. Lorsqu'un tel cône de dimension n dans $\Re^n$ est défini par seulement n hyperplans utiles à sa description, les n combinaisons de $n-1$ hyperplans définissent chacune un des n rayons extrêmes et le point extrême correspondant à la pointe du cône est dit non-dégénéré. Par contre, lorsqu'un tel cône est défini par plus de n hyperplans utiles à sa description, il existe plus de n combinaisons de $n-1$ hyperplans dont seulement certaines d'entre elles décrivent un des rayons extrêmes et le point extrême correspondant à la pointe du cône est alors dit fortement dégénéré. Plusieurs algorithmes sont disponibles pour résoudre ce problème, dont certains reposent sur le principe de la simple énumération de toutes les combinaisons d'hyperplans, tandis que d'autres plus performants reposent sur l'exploitation de l'information inhérente au problème. L'ensemble des algorithmes pour résoudre ce problème sont classés relativement à deux axes principaux selon Avis, Bremner et Seidel (1997) : 1) l'approche d'énumération soit selon un algorithme de balayage d'un graphe, soit selon un algorithme constructif; 2) le traitement de la dégénérescence soit en produisant toutes les faces de dimension inférieure à n, soit en perturbant la représentation sous forme d'hyperplans, soit en conservant l'information sur l'adjacence des rayons extrêmes (dans ce dernier cas, l'énumération s'effectue selon un algorithme constructif du type double description, dû à Motzkin, Raiffa, Thompson et Thrall (1953)).

Nous nous intéressons aussi dans le cadre de cette thèse à l'application de l'énumération des rayons extrêmes d'un cône polyédral pointé à certains algorithmes déterministes de minimisation concave sur un polyèdre. Ces algorithmes s'appuient sur les concepts d'énumération de points extrêmes, d'hyperplans coupants, d'approximation extérieure ou de séparation-évaluation progressive. Précisons qu'à l'exception de tous les algorithmes de séparation-évaluation progressive dit simpliciaux ou rectangulaires basés respectivement sur ceux de Horst (1976) ou de Falk et Soland (1969), tous les algorithmes déterministes de minimisation concave sur un polyèdre dépendent fortement de l'énumération des rayons extrêmes d'un cône polyédral pointé sur un point extrême du polyèdre, d'où l'importance de ce

problème dans le contexte de la minimisation concave.

Un premier objectif de cette thèse consiste à utiliser des principes complémentaires développés dans des contextes différents dans le but de résoudre plus efficacement le problème de l'énumération des rayons extrêmes d'un cône polyédral pointé. En effet, nous présentons un nouvel algorithme de balayage d'un graphe perturbant la représentation sous forme d'hyperplans, en se basant sur des concepts complémentaires issus de la programmation mathématique et de la géométrie algorithmique. Un second objectif de cette thèse consiste à identifier dans le contexte de certains algorithmes déterministes de minimisation concave sur un polyèdre, l'algorithme le plus performant pour traiter la dégénérescence forte lors de l'énumération des rayons extrêmes d'un cône polyédral pointé. Plus précisément, nous comparons les performances de notre nouvel algorithme de balayage d'un graphe à celles du plus efficace des algorithmes constructifs de type double description pour traiter la dégénérescence forte (notons que, dans le contexte de la minimisation concave, les algorithmes produisant toutes les faces de dimension inférieure à n génèrent beaucoup trop d'information pour les fins recherchées). Un troisième objectif de cette thèse consiste à développer un algorithme déterministe de minimisation concave sur un polyèdre, qui exploite l'algorithme le plus efficace identifié précédemment pour énumérer les rayons extrêmes d'un cône polyédral pointé. En particulier, nous présentons un nouvel algorithme de division conique simple et facile à implanter qui détermine un minimum global exact en un temps fini et qui utilise l'algorithme d'ordonnancement des points extrêmes de Murty (1968) d'une manière novatrice.

La thèse est organisée de la manière suivante. Le chapitre 2 présente une revue de la littérature sur les principaux algorithmes d'énumération des rayons extrêmes d'un cône polyédral pointé, de même que sur les principaux algorithmes déterministes de minimisation concave sur un domaine polyédral, qui requièrent ce type d'énumération. Le chapitre 3 présente un article soumis à Discrete Applied Mathematics portant sur le nouvel algorithme de balayage d'un graphe perturbant la représentation sous forme d'hyperplans. Le chapitre 4 contient un article soumis à Journal of Global Optimization traitant de la comparaison, dans le contexte de certains algorithmes déterministes de minimisation concave sur un polyèdre, des performances de l'algorithme présenté au chapitre 3 et de celles du plus efficace des algorithmes constructifs de type double description pour traiter la dégénérescence forte. Le chapitre 5 présente un article soumis à Mathematical Programming décrivant le nouvel algorithme de division conique simple pour déterminer un minimum global exact en un temps fini, en se basant sur les résultats décrits au chapitre 4. Finalement, le chapitre 6 contient un résumé de nos contributions et des futures avenues de recherches. Tous les articles sont écrits en collaboration avec mes directeurs de recherche, Bernard Gendron et Patrick Soriano.

Bibliographie

D. AVIS, D. BREMNER and R. SEIDEL (1997), "How Good Are Convex Hull Algorithms?", *Computational Geometry*, vol. 7, 265-301.

J.E. FALK and R.M. SOLAND (1969), "An Algorithm for Separable Nonconvex Programming Problems", *Management Science*, vol. 15, 550-569.

R. HORST (1976), "An Algorithm for Nonconvex Programming Problems", *Mathematical Programming*, vol. 10, 312-321.

T.S. MOTZKIN, H. RAIFFA, G.L. THOMPSON et R.M. THRALL (1953), "The Double Description Method", Contribution to the Theory of Games II, *Annals of Mathematics Studies*, 51-73.

K.G. MURTY (1968), "Solving the Fixed Charge Problem by Ranking the Extreme Points", *Operations Research*, vol. 16, 268-279.

Chapitre 2

Revue de la littérature

Étant donné que l'énumération des rayons extrêmes d'un cône polyédral pointé à partir de sa représentation sous forme d'hyperplans est un problème retrouvé dans de nombreux contextes des mathématiques appliquées, il est difficile de dresser une liste exhaustive des algorithmes pour résoudre ce problème. Néanmoins, nous tentons dans la première section de ce chapitre de déterminer les principaux algorithmes rencontrés en programmation mathématique et en géométrie algorithmique. Par ailleurs, nous décrivons dans la seconde section de ce chapitre les principaux algorithmes déterministes de minimisation concave sur un polyèdre qui requièrent l'énumération des rayons extrêmes d'un cône polyédral pointé.

2.1 *Énumération des rayons extrêmes d'un cône polyédral pointé*

En translatant à l'origine, pour des fins pratiques, la pointe du cône polyédral pointé K de pleine dimension dans $\Re^n$, considérons sa représentation sous forme d'hyperplans, $Hx \geq 0$, où $H \in \Re^p \times \Re^n$ et p est le nombre d'hyperplans utiles à sa description. L'énumération des rayons extrêmes de K consiste alors à déterminer explicitement $R \in \Re^n \times \Re^q$, où q est le nombre de rayons extrêmes, tel que $x = R\lambda$, $\lambda \geq 0$. Ce problème est équivalent au problème de l'énumération des hyperplans du cône polaire de K à partir de sa représentation sous forme de rayons extrêmes, étant donné qu'un hyperplan définissant K est un rayon extrême sur son ensemble polaire, noté $pol(K)$.

La plupart des algorithmes traitent l'énumération des points extrêmes d'un polyèdre borné D à partir de sa représentation sous forme d'hyperplans $\{x \in \Re^n \mid Ax \geq b, x \geq 0\}$, où $A \in \Re^m \times \Re^n$ et $b \in \Re^m$, plutôt que de traiter directement l'énumération des rayons extrêmes de K. Néanmoins, la relation qui existe entre D et son homogénéisation $D_{Hom} = \{x \in \Re^n, x_{n+1} \in \Re \mid Ax - bx_{n+1} \geq 0, x \geq 0, x_{n+1} \geq 0\}$ est utile dans le cas présent. En effet, il est concevable d'appliquer des algorithmes pour l'énumération des points extrêmes lorsque le polyèdre résultant de la déshomogénéisation de K (défini par $K_{DHom} = \{x \in \Re^{n-1} \mid H'x \geq H_{\bullet n}\}$, où H' est H diminué de sa n-ème colonne $H_{\bullet n}$) est borné, puisque le j-ème rayon extrême de K est $\left(x_1^j, x_2^j, \ldots, x_{n-1}^j, 1\right)$ où x_i^j correspond à la i-ème

coordonnée du j-ème point extrême de K_{DHom}. Lorsqu'un point extrême de K_{DHom} est défini par $n-1$ hyperplans, alors le rayon extrême de K est dit non-dégénéré tandis que lorsqu'un point extrême de K_{DHom} est défini par plus de $n-1$ hyperplans, alors le rayon extrême de K est dit fortement dégénéré.

Les deux premières parties de cette section décrivent succintement chacun des grands axes déterminés par Avis, Bremner et Seidel (1997) pour classifier les algorithmes d'énumération des rayons extrêmes de K. Plus précisément, la première partie décrit les approches d'énumération, tandis que la seconde partie présente les méthodes de traitement de la dégénérescence forte, lorsque la pointe de K est définie par plus de n hyperplans utiles à sa description. La troisième partie de cette section classifie les principaux algorithmes en programmation mathématique et en géométrie algorithmique pour l'énumération des rayons extrêmes de K selon la structure présentée précédemment.

2.1.1 *Approches d'énumération*

La première approche d'énumération des rayons extrêmes de K suit un algorithme de balayage d'un graphe G, où chaque noeud est un ensemble de variables de base formant la matrice B qui correspond à un sous-ensemble de p colonnes linéairement indépendantes du système d'équations

$$Hx - Is = 0, s \geq 0,$$

où $I \in \Re^p \times \Re^p$ est la matrice identité et s est le vecteur des variables d'excédent. Deux noeuds de G sont dits à distance k lorsque les deux ensembles de variables de base ont $p-k$ variables communes tandis que deux noeuds de G sont reliés par une arête de G si les bases correspondantes sont à distance 1. Dans le but d'alléger la notation, introduisons $y^T = (x^T, s^T)$, y_B l'ensemble des variables de base formant B, le sous-ensemble $y_B^{\min}$, qui correspond aux variables de base s, et y_N l'ensemble des variables hors base formant la matrice N avec les colonnes restantes du système d'équations précédent. Si les n premières colonnes de B sont celles de H tandis que les $p-n$ dernières colonnes sont celles de $-I$ tel que B est non-singulière, alors le système d'équations précédent devient

$$By_B + Ny_N = 0, y_{n+1} \geq 0, \ldots, y_{n+p} \geq 0,$$

qui est équivalent à

$$y_B = -(B)^{-1}(N)y_N, y_{n+1} \geq 0, \ldots, y_{n+p} \geq 0.$$

Selon cette construction, si $p = n$, alors il n'existe qu'une seule combinaison de colonnes pour former B ce qui implique que G ne possède qu'un seul noeud et que tous les rayons extrêmes de K sont identifiés avec les colonnes de $-(B)^{-1}N$. Par contre, si $p > n$, alors il y a $(p!)/(n!(p-n)!)$ combinaisons possibles de colonnes pour former B desquelles seulement certaines satisfont la condition de l'indépendance linéaire, et ainsi être des noeuds de G. Un rayon extrême de K est alors identifié par les n premiers éléments d'une colonne $-(B)^{-1}N$ si les

$p-n$ derniers éléments de cette colonne sont positifs, puisque, dans le cas contraire, un scalaire positif de cette colonne viole les contraintes de non-négativité de $y_B^{\min}$. Finalement, lorsque tous les noeuds de G sont visités par un balayage quelconque, alors tous les rayons extrêmes de K sont identifiés, mais certains le sont plus d'une fois, ce qui implique qu'il est requis pour la plupart des algorithmes de balayage de n'ajouter à la liste des rayons extrêmes de K que ceux qui n'ont pas encore été identifiés.

La seconde approche d'énumération des rayons extrêmes de K suit un algorithme constructif. Étant donné que K est de pleine dimension, il existe toujours un sous-ensemble de n demi-espaces engendrés par n hyperplans linéairement indépendants définissant un cône polyédral pointé $\overline{K}$ approchant par l'extérieur K. L'inverse de la matrice de la représentation sous forme d'hyperplans de $\overline{K}$, $\overline{H}x \geq 0$, où $\overline{H} \in \Re^n \times \Re^n$, contient dans chacune de ses n colonnes un des rayons extrêmes de $\overline{K}$ puisque

$$\overline{H}x = \lambda \geq 0 \Leftrightarrow x = (\overline{H})^{-1}\lambda \text{ et } \lambda \geq 0.$$

Le principe d'un algorithme constructif consiste à déterminer les rayons de $\overline{K}^{'}$ qui correspond à l'intersection de $\overline{K}$ et d'un demi-espace $(h')^T x \geq 0$ engendré par un des hyperplans de la représentation de K qui n'est pas dans le sous-ensemble définissant $\overline{K}$. Bien entendu, un rayon extrême de $\overline{K}$ est toujours réalisable si le produit scalaire avec h' est supérieur ou égal à 0 mais il ne l'est plus si le produit scalaire est inférieur à 0. Par contre, de nouveaux rayons extrêmes de $\overline{K}^{'}$ sont formés par l'intersection de $(h')^T x = 0$ avec chaque face de dimension 2 de $\overline{K}$ dont l'un des rayons extrêmes est toujours réalisable tandis que l'autre ne l'est plus. Une fois que tous les rayons extrêmes de $\overline{K}^{'}$ sont identifiés, il est nécessaire déterminer les rayons de $\overline{K}^{''}$ qui correspond à l'intersection de $\overline{K}^{'}$ et d'un demi-espace $(h'')^T x \geq 0$ engendré par un des hyperplans de la représentation de K qui n'est pas dans le sous-ensemble définissant $\overline{K}^{'}$ et ainsi de suite jusqu'à ce que tous soient considérés.

2.1.2 *Traitement de la dégénérescence forte*

Une première méthode de traitement de la dégénérescence forte du point extrême correspondant à la pointe de K est de générer toutes les faces de dimension inférieure à n, dont celles de dimension 1 correspondent aux rayons extrêmes de K. Cette méthode de traitement de la dégénérescence forte est habituellement appliquée lorsqu'il est requis d'avoir la structure de toutes les faces définissant K ce qui n'est pas le cas pour les fins recherchés lors de l'énumération des rayons extrêmes de K à partir de sa représentation sous forme d'hyperplans. En effet, il serait laborieux d'obtenir toutes les faces de dimension supérieure à 1 lorsque seulement celles de dimension 1 sont d'intérêt en plus d'être probablement peu efficace, compte tenu des calculs requis pour la caractérisation de chacune des faces de dimension supérieure à 1.

Une seconde méthode de traitement de la dégénérescence forte du point extrême correspondant à la pointe de K est de perturber sa représentation sous forme d'hyperplans dans le but de simuler la non-dégénérescence. En effet, lorsqu'un algorithme est efficace sans la présence de dégénérescence forte, il le demeure malgré la présence de dégénérescence forte, mais il devient toutefois sensible au nombre de rayons extrêmes engendrés par la perturbation, qui peut être très grand, voire exponentiel dans certains cas. Il est important de noter que l'affirmation précédente est valide lorsque la perturbation n'est pas numérique mais plutôt symbolique comme dans le cas de la perturbation lexicographique introduite par Dantzig, Orden et Wolfe (1955). Finalement, cette méthode de traitement de la dégénérescence forte est plus appropriée pour les fins recherchés car elle ne détermine que de l'information utile, contrairement à la méthode consistant à générer toutes les faces de dimension inférieure à n.

Une troisième méthode de traitement de la dégénérescence forte du point extrême correspondant à la pointe de K est de conserver de l'information sur l'adjacence des rayons extrêmes lorsque l'énumération s'effectue selon l'algorithme constructif de type double description, dû à Motzkin, Raiffa, Thompson et Thrall (1953). En effet, plutôt que d'utiliser un test algébrique fondé sur le rang de la matrice des vecteurs normaux des hyperplans actifs pour chaque couple de rayons de $\overline{K}$, il est possible d'utiliser un test combinatoire afin d'identifer une face de dimension 2 : deux rayons extrêmes sont dits adjacents et définissent une face de dimension 2 de $\overline{K}$ si et seulement si il n'existe aucun autre rayon extrême de $\overline{K}$ qui possède les mêmes hyperplans actifs que ceux communs aux deux rayons extrêmes en question. Cette opération constitue la pierre angulaire de tout algorithme constructif de type double description pour l'énumération des rayons extrêmes de K. De plus, il est important de noter que lorsque le test combinatoire est utilisé, l'ordre d'insertion des demi-espaces est crucial puisqu'il est souhaitable de déterminer une séquence de cônes polyédraux pointés approchant par l'extérieur K dont la croissance du nombre de rayons extrêmes est faible au début, étant donné que le nombre de combinaisons possibles ne croit alors pas trop rapidement.

2.1.3 *Classification des principaux algorithmes d'énumération*

Selon l'argumentation précédente, la classification est effectuée seulement pour les principaux algorithmes qui traitent la dégénérescence forte du point extrême correspondant à la pointe de K, soit en perturbant sa représentation sous forme d'hyperplans, soit en conservant l'information sur l'adjacence des rayons extrêmes lorsque l'énumération s'effectue selon un algorithme constructif de type double description.

2.1.3.1 *Algorithmes de balayage d'un graphe*

Dans la présentation qui suit, nous nous attardons sur la méthode de parcours de G, ainsi que sur le traitement de la dégénérescence forte étant donné que ces deux aspects définissent les

caractéristiques principales de tout algorithme de balayage d'un graphe.

Lorsque K_{DHom} est un polyèdre borné, l'algorithme de Balinski (1961) consiste à exploiter le fait qu'il est facile d'énumérer les bases d'une face de dimension 2 par l'algorithme du simplexe. Sommairement, cet algorithme détermine pour $p-n$ hyperplans de K_{DHom} l'ensemble de toutes les faces de dimension 2 en prenant le soin pour chacune d'elles d'énumérer toutes ses bases et de conserver celles qui n'ont pas encore été trouvées et qui appartiennent à K_{DHom}. En ce qui concerne le traitement de la dégénérescence forte par cet algorithme, aucune précision n'est apportée par Balinski ce qui laisse croire que toutes les combinaisons de base doivent être générées. L'algorithme de Manas et Nedoma (1968) requiert à chaque itération de vérifier s'il existe dans la liste des bases à traiter une base à distance k (en commencant avec $k = 1$) de celle qui vient d'être ajoutée à la liste des bases traitées. Dans l'affirmative, la base identifiée est ajoutée à la liste des bases traitées tandis que les bases à distance 1 de la base identifiée sont ajoutées à la liste des bases à traiter, en omettant les duplications. La construction de l'algorithme de Manas et Nedoma est telle que toutes les combinaisons de base doivent être énumérées pour un point extrême fortement dégénéré de K_{DHom}. L'algorithme de Murty (1968) détermine quant à lui la séquence des bases dans un ordre non-croissant par rapport à une fonction objectif donnée, où la première base est une base optimale, tandis que la seconde est trouvée parmi les bases à distance 1 de la première, alors que la troisième est trouvée parmi les bases à distance 1 de la première et de la seconde et ainsi de suite. Tout comme l'algorithme de Manas et Nedoma, l'algorithme de Murty est tel que toutes les combinaisons de base doivent être énumérées pour un point extrême fortement dégénéré de K_{DHom}. L'algorithme de Matheiss (1971) consiste à enchâsser K_{DHom} dans un espace supérieur d'une dimension par l'ajout d'une variable d'écart généralisée, de telle sorte que la projection des nouveaux points extrêmes sur l'espace original soient intérieurs à K_{DHom} et dont chacun a un rang qui lui est attribué. Un arbre de recouvrement de ces points intérieurs est par la suite construit en fonction du rang, où les bases terminales correspondent aux points extrêmes de K_{DHom}. Selon Matheiss, la perturbation de la représentation sous forme d'hyperplans de K_{DHom} est utilisée seulement pour simplifier la présentation de l'algorithme, mais aucun détail supplémentaire n'est donné lorsqu'aucune perturbation est appliquée. L'algorithme de McKeown (1975), qui est semblable à l'algorithme de Murty, utilise l'algorithme constructif de Chernikova (1965) afin d'énumérer les rayons extrêmes du cône polyédral pointé correspondant au sous-ensemble des hyperplans actifs de K_{DHom} pour un point extrême fortement dégénéré. L'algorithme de Dyer et Proll (1977) exige, lors de la k-ème itération, que soient déterminées toutes les bases qui sont à une distance k de la base initiale et qui n'ont pas encore été générées. Cet algorithme diffère de celui de Manas et Nedoma uniquement dans le fait qu'une approche en largeur est utilisée plutôt qu'une approche en profondeur, ce qui permet toutefois d'exploiter efficacement les concepts du simplexe révisé. Selon Dyer et Proll, il est recommendé d'utiliser la perturbation de la

représentation sous forme d'hyperplans de K_{DHom} dans le but de réduire le nombre de bases visitées correspondant à un point extrême fortement dégénéré de K_{DHom}. L'algorithme d'Avis et Fukuda (1992) est pour sa part sembable à celui de Manas et Nedoma. Cependant, afin d'omettre les duplications, il n'est plus nécéssaire de vérifier à chaque itération la présence de chaque base à distance 1 de la base traitée dans la liste des bases à traiter. En effet, l'algorithme utilise judicieusement la règle de Bland (1977) pour choisir de manière non-ambigüe la variable sortant de la base, étant donné un choix unique de la variable hors-base selon une fonction objectif particulière. L'algorithme d'Avis et Fukuda est tel qu'il requiert de générer toutes les combinaisons de base ce qui est particulièrement problématique lorsque la base optimale relativement à la fonction objectif particulière est dégénérée, car il doit alors être appliqué pour chaque base optimale. Finalement, l'algorithme de Bremner, Fukuda et Marzetta (1998) consiste à alterner, selon la complexité du problème à résoudre, entre la résolution du problème de l'énumération des points extrêmes de K_{DHom} à partir de sa représentation sous forme d'hyperplans et celle du problème de l'énumération des hyperplans de $pol(K_{DHom})$ à partir de sa représentation sous forme de points extrêmes. Cet algorithme se sert de celui développé par Avis (2000), qui est une spécialisation de l'algorithme d'Avis et Fukuda où la règle de Bland est remplacée par la perturbation lexicographique de Dantzig, Orden et Wolfe, et pour lequel il est suffisant d'appliquer une seule fois l'algorithme à partir de l'unique base optimale au sens lexicographique.

Basé plutôt sur l'énumération des rayons extrêmes de K, l'algorithme de Kruse (1986) est semblable à celui de Dyer et Proll. Cependant, toutes les bases sont engendrées par la perturbation lexicographique de Dantzig, Orden et Wolfe dans le but de réduire le nombre de bases visitées correspondant toutes à la pointe de K. Pour ce cas particulier, Gal (1985) définit G comme étant un graphe de dégénérescence qui possède des noeuds internes inutiles à la description d'au moins un rayon extrême de K et des noeuds de transition qui identifient toujours au moins un rayon extrême de K. L'algorithme de Geue (1993) est quant à lui semblable à celui de Manas et Nedoma. Toutefois, toutes les bases sont engendrées selon une perturbation lexicographique généralisée dans le but de réduire le nombre de bases visitées. Un cas particulier de la pertubation lexicographique généralisée est la règle de pivotage sur les noeuds de transition de G introduite par Gal et Geue (1992), où toutes les bases engendrées par l'algorithme sont telles qu'au moins un rayon extrême de K y est identifié.

2.1.3.2 *Algorithmes constructifs*

Dans la présentation qui suit, nous nous attardons sur la manière d'insérer les demi-espaces ainsi que sur la manière de déterminer l'adjacence de deux rayons extrêmes définissant une face de dimension 2 intersectée par l'hyperplan engendrant le demi-espace ajouté étant donné que ces deux aspects définissent les caractéristiques principales de chaque algorithme constructif. De plus, une attention particulière est apportée au traitement de la dégénérescence

forte lorsque l'algorithme constructif n'est pas du type double description.

Basé sur l'énumération des rayons extrêmes de K, l'algorithme de Motzkin, Raiffa, Thompson et Thrall (1953) insère à chaque itération le demi-espace dont l'hyperplan correspondant a le plus petit indice et utilise un test combinatoire afin de déterminer si deux rayons extrêmes sont adjacents. L'algorithme d'Uzawa (1958) utilise aussi un ordre d'insertion correspondant au demi-espace dont l'hyperplan a le plus petit indice, mais pour lequel il n'est plus requis de vérifier si deux rayons extrêmes sont adjacents, ce qui permet une exécution plus rapide au prix d'une consommation énorme de l'espace mémoire, étant donné la présence de nombreux rayons surperflus. L'algorithme de Chernikova (1964, 1965) est identique à l'algorithme de Motzkin, Raiffa, Thompson et Thrall. L'algorithme de Greenberg (1975) insère à chaque itération le demi-espace dont l'hyperplan a le plus petit indice en plus d'introduire une condition nécessaire pour l'adjacence de deux rayons extrêmes, qui n'est toutefois pas suffisante tel que démontré par Sherman (1977). Néanmoins, Dyer et Proll (1980) déterminent un autre test combinatoire à effectuer à la fin de chaque itération ou encore à la fin de l'algorithme, dans le but d'éliminer les rayons superflus générés par l'algorithme de de Greenberg. Une première version de l'algorithme de Fukuda et Prodon (1996) est une alternative à l'algorithme de Chen, Hansen et Jaumard (1991). Dans cet algorithme, seuls les couples de rayons adjacents qui produisent un nouveau rayon extrême au cours des itérations subséquentes sont conservés en mémoire, lorsque l'ordre d'insertion des demi-espaces est connu d'avance. Lorsque l'ordre d'insertion des demi-espaces n'est pas connu d'avance, une seconde version de l'algorithme de Fukuda et Prodon consiste à utiliser la notion d'adjacence héritée, ainsi qu'une autre condition nécessaire à l'adjacence de deux rayons extrêmes, généralement plus forte que la condition nécessaire de Greenberg, afin d'obtenir un algorithme aussi performant que le première version.

Lorsque K_{DHom} est un polyèdre borné, l'algorithme de Thieu, Tam et Ban (1983) utilise le test algébrique dans le but de déterminer un nouveau point extrême lorsqu'un couple de points extrêmes définissant une arête de l'approximation extérieure de K_{DHom} est intersectée par l'hyperplan engendrant le demi-espace ajouté. Dans ce cas particulier, l'ordre d'insertion des demi-espaces est déterminé par l'algorithme déterministe d'approximation extérieure servant à identifier un minimum global d'une fonction concave sur un polyèdre. L'algorithme de Horst, Thoai et de Vries (1988) consiste à appliquer un algorithme de balayage d'un graphe semblable à celui de Manas et Nedoma sur chaque cône polyédral pointé définissant un point extrême réalisable par rapport à l'hyperplan engendrant le demi-espace ajouté. Certaines arêtes de l'approximation extérieure de K_{DHom} sont ainsi identifiées et de nouveaux points extrêmes sont produits lorsque l'autre extrémité de chaque arête est non-réalisable. Étant développé dans le contexte des algorithmes déterministes de minimisation concave sur un polyèdre, un ordre d'insertion des demi-espaces est déterminé par chaque algorithme déterministe qui requiert cet

algorithme d'énumération. L'algorithme de Clarkson et Shor (1988) se base sur un ordre d'insertion aléatoire des demi-espaces et exige que soit perturbée la représentation sous forme d'hyperplans de K_{DHom}. L'algorithme conserve en mémoire le graphe d'incidence de K_{DHom}, ainsi qu'une liste de conflits pour chacune de ses faces, qui correspond à tous les hyperplans des demi-espaces non-traités coupant la face en question, dans le but de déterminer l'adjacence de deux points extrêmes. L'algorithme de Chen, Hansen et Jaumard (1991) maintient explicitement la liste des points extrêmes adjacents pour chaque point extrême, ce qui s'avère très coûteux en espace mémoire. Étant aussi développé dans le contexte des algorithmes déterministes de minimisation concave sur un polyèdre, un ordre d'insertion des demi-espaces est déterminé par chaque algorithme déterministe qui requiert cet algorithme d'énumération. L'algorithme de Chazelle (1993) diffère de celui de Clarkson et Shor uniquement par l'exploitation d'un ordre d'insertion déterministe très sophistiqué des demi-espaces.

2.2 *Algorithmes déterministes de minimisation concave*

Le problème de la minimisation d'une fonction continue et concave $f: \Re^n \to \Re$ sur un polyèdre de pleine dimension D dans $\Re^n$ s'écrit ainsi

$$\begin{aligned} &\min f(x) \\ &\text{s.c. } Ax \geq b \\ &x \geq 0 \end{aligned}$$

où $A \in \Re^m \times \Re^n$ and $b \in \Re^m$. Lorsque D est un ensemble compact, le théorème de Weierstrass garantit l'existence d'au moins un minimum global de f sur D, puisque f est continue tandis que le théorème de Caratheodory indique qu'un minimum global est en fait un point extrême de D. Lorsque D est non-borné, s'il existe au moins un rayon extrême r de D pour lequel $f(\alpha r) \to -\infty$ quand $\alpha \to \infty$, alors f n'est pas bornée inférieurement sur D. En d'autres termes, si f est bornée inférieurement sur D, alors un minimum global est un point extrême de D. Fréquemment appelé dans la littérature le problème de minimisation concave de base (*MCB*), ce problème est dit multi-extrémal, puisque l'application d'un algorithme standard d'optimisation convexe peut conduire à un point extrême qui est un minimum local sans toutefois être un minimum global (la localité est définie ici par l'ensemble des points extrêmes voisins d'un point extrême quelconque).

Selon Horst et Tuy (1996), il existe plusieurs algorithmes déterministes pour résoudre *MCB* dont chacun s'appuie sur les concepts d'énumération de points extrêmes, d'hyperplans coupants, d'approximation extérieure ou de séparation-évaluation progressive. À l'exception de tous les algorithmes de séparation-évaluation progressive dit simpliciaux ou rectangulaires basés respectivement sur ceux de Horst (1976) ou de Falk et Soland (1969), l'ensemble des algorithmes déterministes pour résoudre *MCB* dépendent fortement de l'énumération des rayons extrêmes d'un cône polyédral pointé sur un point extrême de D à partir de sa

représentation sous forme d'hyperplans. Parmi ces algorithmes, les principaux sont l'algorithme des hyperplans coupants, l'algorithme d'approximation extérieure, l'algorithme d'annexation polyédrale et l'algorithme de division conique.

Chaque partie de cette section décrit succintement les grands principes sous-jacents à chacun de ces algorithmes déterministes pour résoudre *MCB*, en prenant soin de mettre en relief l'importance de l'énumération des rayons extrêmes d'un cône polyédral pointé sur un point extrême de D. La dernière partie de cette section contient, en plus, une présentation des algorithmes de division conique rencontrés dans la littérature sur les algorithmes déterministes pour résoudre *MCB*. Dans le but d'alléger la présentation des grands principes, nous supposons que D est un polyèdre borné et que l'ensemble $\{x \in \Re^n \mid f(x) \geq \xi\}$ est borné pour tout nombre réel ξ. Pour un traitement plus général, nous référons le lecteur à la section 4.3 de cette thèse pour une présentation générale des trois premiers algorithmes et à la section 5.3 pour une présentation du dernier algorithme lorsque l'ensemble $\{x \in \Re^n \mid f(x) \geq \xi\}$ n'est plus nécessairement borné pour tout nombre réel ξ.

2.2.1 *Algorithme des hyperplans coupants*

En s'appuyant sur Tuy (1964), l'algorithme des hyperplans coupants consiste à démontrer $D \backslash G_\gamma = D \backslash \{x \in \Re^n \mid f(x) \geq \gamma\} = \emptyset$, où $f(v) = \gamma$, v étant un point extrême minimum local candidat à titre de minimum global. Si v est non-dégénéré ou faiblement dégénéré (c.-à-d. que v est défini par n hyperplans utiles à sa description), alors la γ-coupe de concavité relative à v est un demi-espace dénoté $h_z^T(x - v) \geq 1$, où l'hyperplan h_z contient les n points $z \in bd(G_\gamma) = \{x \in \Re^n \mid f(x) = \gamma\}$ trouvés sur chacun des n rayons extrêmes du plus petit cône polyédral pointé sur v contenant D. Si v est fortement dégénéré (c.-à-d. que v est défini par plus de n hyperplans utiles à sa description), alors le plus petit cône polyédral pointé sur v contenant D possède plus de n rayons extrêmes et il est parfois impossible de trouver un hyperplan h_z contenant tous les points z trouvés sur chacun des rayons extrêmes de ce cône. Il faut alors avoir recours à l'approche généralisée de Carvajal-Moreno (1972) qui permet de construire une γ-coupe de concavité relative à v qui est valide et dont la portion couverte de D est maximale. Cette approche domine celle de Balas (1971) qui détermine une γ-coupe de concavité relative à v pour un sous-ensemble quelconque de n rayons extrêmes du plus petit cône polyédral pointé sur v contenant D et dont la profondeur n'est pas nécessairement maximale. Il est important de noter que l'application de l'approche de Carvajal-Moreno exige de déterminer explicitement tous les rayons extrêmes du plus petit cône polyédral contenant D pointé sur un rayon extrême fortement dégénéré, avec l'aide d'une des algorithmes décrits précédemment.

Soit un point extrême v minimum local tel que $f(v) = \gamma$, après avoir résolu le problème

$$\max_{x \in D} \; h_z^T(x - v)$$

dont la valeur optimale est μ_z, si $\mu_z \leq 1$, alors un minimum global a été identifié tandis que si $\mu_z > 1$, alors une itération de l'algorithme des hyperplans coupants prescrit d'abord d'appliquer la γ-coupe de concavité relative à v sur D, définissant ainsi un polyèdre résiduel D', puis de chercher un minimum local v' sur D' à partir d'une solution optimale du problème de minimisation précédent. L'algorithme des hyperplans coupants exige finalement de construire une γ-coupe de concavité relative à v' définie par $h_{z'}^T(x - v') \geq 1$ pour ensuite commencer la prochaine itération en résolvant le problème

$$\max_{x \in D'} \; h_{z'}^T(x - v').$$

Cependant, si v' fait partie de $D \backslash G_\gamma$, alors v' est le nouveau candidat à titre de minimum global et l'algorithme des hyperplans coupants prescrit simplement de laisser $\gamma = f(v')$ avant de construire une γ-coupe de concavité relative à v'.

2.2.2 *Algorithme d'approximation extérieure*

En s'appuyant sur Hoffman (1981) et Thieu, Tam et Ban (1983), un algorithme d'approximation extérieure consiste à approcher extérieurement D par un polyèdre borné initial $\overline{D}$ possédant peu de points extrêmes, puisqu'il est alors aisé d'y déterminer un minimum global de f. Un choix simple pour former $\overline{D}$ consiste à considérer l'intersection du plus petit cône polyédral pointé contenant D et du demi-espace contenant v engendré par l'hyperplan $h_z^T(x - v) = \mu_z$ résultant de l'application d'une seule itération de l'algorithme des hyperplans coupants. Si le minimum global de f sur $\overline{D}$, dénoté $\overline{v}_f^*$, appartient à D, alors il n'est plus nécessaire de continuer la recherche d'un minimum global de f sur D. Puisque ce n'est généralement pas le cas, l'algorithme d'approximation extérieure prescrit d'ajouter à la description de $\overline{D}$, le demi-espace de D tel que la valeur de $a_i^T \overline{v}_f^* - b_i$ soit minimale, formant ainsi $\overline{D}'$.

Une fois que le demi-espace de D est ajouté à la description de $\overline{D}$ pour former $\overline{D}'$, il est essentiel d'identifier tous les nouveaux points extrêmes de $\overline{D}'$ afin de déterminer un minimum global de f sur $\overline{D}'$. À cette fin, l'utilisation de tout algorithme de balayage d'un graphe est inadéquat, puisque chacun d'eux énumère les points extrêmes de $\overline{D}'$ à partir d'une base de départ sans exploiter l'information relative aux points extrêmes déjà existants. En fait, il est naturel dans ce contexte particulier d'appliquer n'importe lequel des algorithmes constructifs pour l'énumération des points extrêmes de $\overline{D}'$ qui ne requiert pas de savoir à l'avance l'ordre d'insertion des demi-espaces, cet ordre étant déterminé par l'application de l'algorithme d'approximation extérieure.

2.2.3 *Algorithme d'annexation polyédrale*

En s'appuyant sur Vaish et Shetty (1976) et Tuy (1990), l'algorithme d'annexation polyédrale consiste à démontrer $D\backslash G_\gamma = D\backslash\{x \in \Re^n \mid f(x) \geq \gamma\} = \emptyset$, où $f(v) = \gamma$, v étant un point extrême minimum local candidat à titre de minimum global. En translatant v à l'origine pour des fins pratiques et après avoir effectué une seule itération de l'algorithme des hyperplans coupants, si une solution optimale du problème

$$\max_{x \in D} \; h_z^T x,$$

dénotée v_z^*, fait partie de G_γ telle que $\mu_z > 1$, alors l'algorithme d'annexation polyédrale requiert d'abord de couper v_z^* en formant le polyèdre borné $P = conv\{D \cap \{x \in \Re^n | h_z^T x \leq 1\}, \{\hat{v}_z^*\}\}$ où $\hat{v}_z^*$ est un point de $bd(G_\gamma)$ sur le rayon émanant de l'origine qui contient le point v_z^*. Ensuite, l'algorithme d'annexation polyédrale exige pour chaque hyperplan h_P de la représentation de P ne contenant pas l'origine de résoudre le problème

$$\max_{x \in D} \; h_P^T x$$

dont la valeur optimale est μ_P et une solution optimale est v_P^*. Si $\mu_P \leq 1$ pour tout h_P, alors v est un minimum global, mais puisque ce n'est généralement pas le cas, l'algorithme d'annexation polyédrale prescrit de couper le point v_P^* ayant la plus grande valeur μ_P correspondante en formant le polyèdre borné $P' = conv\{P, \{\hat{v}_P^*\}\}$ où $\hat{v}_P^*$ est un point de $bd(G_\gamma)$ sur le rayon émanant de l'origine qui contient le point identifié v_P^*. Cependant, si v_P^* fait partie de $D\backslash G_\gamma$, alors l'algorithme d'annexation polyédrale dans sa forme la plus simple prescrit de chercher un nouveau minimum local sur D à partir de v_P^* et de mettre à jour γ avant de recommencer l'algorithme en translatant à l'origine ce minimum local.

La manière dont est déterminé P' à partir de P requiert d'étendre le concept de dualité entre un hyperplan définissant un cône polyédral pointé et le rayon extrême correspondant sur l'ensemble polaire de ce cône : chaque hyperplan de P ne contenant pas l'origine correspond à un point extrême de $pol(P)$ tandis que chaque hyperplan de P contenant l'origine correspond à un rayon extrême de $pol(P)$. Lorsque le demi-espace engendré par l'hyperplan v_P^* ne contenant pas l'origine est ajouté au cône polyédral $pol(P)$ pointé sur h_z dont tous les rayons extrêmes sont connus, seulement de nouveaux points extrêmes formant $pol(P')$ sont générés et ils correspondent aux nouveaux hyperplans de P ne contenant pas l'origine que requiert l'algorithme d'annexation polyédrale. Pour les mêmes raisons que dans le cas de l'algorithme d'approximation extérieure, il est naturel dans ce contexte particulier d'utiliser n'importe lequel des algorithmes constructifs pour l'énumération des points extrêmes de $pol(P')$ qui n'exigent pas de savoir à l'avance l'ordre d'insertion des demi-espaces, cet ordre étant déterminé par l'application de l'algorithme d'annexation polyédrale.

2.2.4 *Algorithme de division conique*

En s'appuyant sur Tuy (1964), l'algorithme de division conique consiste à démontrer $D\backslash G_\gamma = D\backslash\{x \in \Re^n \mid f(x) \geq \gamma\} = \emptyset$, où $f(v) = \gamma$, v étant un point extrême minimum local candidat à titre de minimum global. Après avoir effectué une seule itération de l'algorithme des hyperplans coupants, si une solution optimale du problème

$$\max_{x \in D} \quad h_z^T(x - v),$$

dénotée v_z^*, fait partie de G_γ telle que $\mu_z > 1$, alors l'algorithme de division conique requiert d'abord de partitionner le plus petit cône polyédral pointé K sur v contenant D par rapport à un rayon r de K émanant de v tel que chaque élément de la partition est aussi un cône polyédral pointé sur v. Si v est non-dégénéré ou faiblement dégénéré, alors l'intersection du plus petit cône polyédral pointé sur v contenant D avec $h_z^T(x - v) = 1$ forme un simplexe de dimension $n - 1$. Pour l'unique point de ce simplexe appartenant aussi à r, l'utilisation de la division radiale introduite par Horst (1976) permet d'obtenir une partition du simplexe dont chaque élément est un simplexe et tel que pour chaque simplexe de la partition, un cône polyédral pointé sur v est défini par sa représentation sous forme de rayons extrêmes correspondant aux demi-lignes émanant de v passant par chaque point extrême du simplexe. L'union de tous les cônes polyédraux pointés sur v ainsi engendrés permet d'obtenir une partition du plus petit cône polyédral pointé sur v contenant D par rapport à un rayon r. Après avoir déterminé les γ-coupes de concavité relative à v sur chaque cône polyédral pointé K' de la partition, l'algorithme de division conique requiert ensuite, pour chaque hyperplan h_z, de résoudre le problème

$$\max_{x \in D \cap K'} \quad h_z^T(x - v).$$

Si $\mu_z \leq 1$, pour tout h_z, alors v est un minimum global. Puisque ce n'est généralement pas le cas, l'algorithme de division conique prescrit de partitionner l'élément de la partition du plus petit cône polyédral pointé sur v contenant D ayant la plus grande valeur μ_z correspondante. Cependant, si v_z^* fait partie de $D\backslash G_\gamma$, alors l'algorithme de division conique dans sa forme la plus simple prescrit de chercher un nouveau minimum local sur D à partir de v_z^* et de mettre à jour γ avant de recommencer une autre itération l'algorithme.

Si v est fortement dégénéré, alors l'intersection du plus petit cône polyédral pointé sur v contenant D avec $h_z^T(x - v) = 1$ ne forme plus un simplexe ce qui invalide l'utilisation de la division radiale. En fait, il n'existe aucun algorithme de division conique pour résoudre *MCB* lorsque la pointe du plus petit cône polyédral contenant D est un point extrême fortement dégénéré. Lorsqu'un tel cas survient, l'ensemble de ces algorithmes prescrivent de chercher un autre point extrême minimum local non-dégénéré ou faiblement dégénéré avant de commencer. Cependant, dans le but d'introduire un algorithme de division conique généralisé capable d'être appliqué sur le plus petit cône polyédral contenant D dont la pointe correspond à

un point extrême fortement dégénéré, il est nécessaire d'introduire une méthode permettant de trouver une partition de ce cône. Il est important de noter qu'une telle méthode repose lourdement sur l'énumération des hyperplans de chaque cône de la partition, effectuée à l'aide d'un des algorithmes décrits précédemment.

Le premier algorithme de division conique est celui de Tuy (1964), qui diffère principalement de l'algorithme décrit ci-dessus au niveau de la résolution du problème de maximisation de la γ-coupe de concavité, qui est défini sur D et non sur $D \cap K'$ et où chaque r est tel qu'il contient v_z^*. Néanmoins, Zwart (1973) démontre que cet algorithme peut possiblement cycler et il propose (Zwart (1974)) de toujours résoudre le problème de maximisation de la γ-coupe de concavité sur $D \cap K'$ afin d'avoir un algorithme de division conique convergent pour un ε-minimum global au sens où il peut exister un point extrême v de D et un point v' tel que $f(v') \leq f(v)$ et $\|v - v'\| \leq \varepsilon$. Cependant, Tuy (1990) démontre que l'algorithme de division conique de Zwart (1974) mène parfois à une solution incorrecte. Thoai and Tuy (1980) introduisent un premier algorithme de division conique convergent vers un ε'-minimum global, $v_{f,\varepsilon'}^*$, au sens où $f\left(v_{f,\varepsilon'}^*\right) - \varepsilon' \leq f(x)$, $\forall\ x \in D$. Cet algorithme est basé sur l'exploitation d'une division conique dite exhaustive, c'est-à-dire que pour toutes les sous-suites infinies de toutes les suites infinies de cônes polyédraux imbriqués, chaque sous-suite converge vers un rayon. Jacobsen (1981) tente sans succès (voir Tuy (1991)) de démontrer la convergence de l'algorithme de division conique, où chaque r est tel qu'il contient v_z^* tandis que Locatelli (1999) et Jaumard et Meyer (2001) donnent tous deux des preuves différentes de ce résultat. Bien que le premier algorithme de division conique convergeant en un temps fini vers un ε'-minimum global lorsque $\varepsilon' > 0$ soit celui de Thoai and Tuy, le premier algorithme de division conique convergent en un temps fini vers un minimum global exact est dû à Hamami and Jacobsen (1988). Ce dernier algorithme exige l'introduction d'une division conique dite exhaustive et non-dégénérée, c'est-à-dire que pour toutes les sous-suites infinies de toutes les suites infinies de cônes polyédraux imbriqués, chaque sous-suite converge vers un rayon et les vecteurs normaux des γ-coupes de concavité correspondantes tendent à être orthogonaux à G_γ au point où le rayon en question croise G_γ. Puisque cette dernière exigence est difficile à satisfaire en pratique, Tuy (1991) propose un algorithme de division conique plus simple et convergeant vers un ε'-minimum global basé sur une division conique normale. Celle-ci consiste à définir une stratégie combinant une division conique exhaustive et une division conique non-dégénérée définie dans un sens plus faible que dans le cas de Hamami et Jacobsen. Tout comme l'algorithme de Thoai et Tuy, cet algorithme est fini seulement lorsque $\varepsilon' > 0$ tandis qu'il peut être infini lorsque $\varepsilon' = 0$.

Bibliographie

D. AVIS (2000), "A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm", *Polytopes - Combinatorics and Computation*, vol. 29, 177-198.

D. AVIS, D. BREMNER et R. SEIDEL (1997), "How Good Are Convex Hull Algorithms?", *Computational Geometry*, vol. 7, 265-301.

D. AVIS et K. FUKUDA (1992), "A Pivoting Algorithm for Convex Hull and Vertex Enumeration of Arrangements and Polyhedra", *Discrete Computational Geometry*, vol. 8, 295-313.

E. BALAS (1971), "Intersection Cuts - a New Type of Cutting Planes for Integer Programming", *Operations Research*, vol. 19, 19-39.

M.L. BALINSKI (1961), "An Algorithm for Finding All Verticies of Convex Polyhedral Sets", *SIAM Journal*, vol. 9, 72-88.

R.G. BLAND (1977), "New Finite Pivoting Rules for the Simplex Method", *Mathematics of Operations Research*, vol. 2, 103-107.

D. BREMNER, K. FUKUDA et A. MARZETTA (1998), "Primal-Dual Methods for Vertex and Facet Enumeration", *Discrete Computational Geometry*, vol. 20, 333-358.

R. CARVAJAL-MORENO (1972), "Minimization of a Concave Functions Subject to Linear Constraints", *Operations Research Center Report*, ORC 72-3.

B. CHAZELLE (1993), "An Optimal Convex Hull Algorithm in any Fixed Dimension", *Discrete Computational Geometry*, vol. 10, 377-409.

P.-C. CHEN, P. HANSEN et B. JAUMARD (1991), "On-line and Off-line Vertex Enumeration by Adjacency Lists", *Operations Research Letters*, vol. 10, 403-409.

N.V. CHERNIKOVA (1964), "Algorithm for Finding a General Formula for the Nonnegative Solutions of a System of Linear Equations", *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 4, 151-156.

N.V. CHERNIKOVA (1965), "Algorithm for Finding a General Formula for the Nonnegative Solutions of a System of Linear Equations", *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 5, 228-233.

K.L. CLARKSON et P.W. SHOR (1988), "Algorithms for Diametral Pairs and Convex Hull that are Optimal, Randomized and Incremental", *Proceedings of 4th Annual ACM Symposium on Computational Geometry*, 12-17.

G.B. DANTZIG, A. ORDEN et P. WOLFE (1955), "The Generalized Simplex Method for Minimizing a Linear Function Form under Linear Inequality Restraints", *Pacific Journal of Mathematics*, vol. 5, 183-195.

M.E. DYER et L.G. PROLL (1977), "An Algorithm for Determining All Extreme Points of a Convex Polytope", *Mathematical Programming*, vol. 12, 81-96.

M.E. DYER et L.G. PROLL (1980), "Eliminating Extraneous Edges in Greenberg's Algorithm", *Mathematical Programming*, vol. 19, 106-110.

J.E. FALK and R.M. SOLAND (1969), "An Algorithm for Separable Nonconvex Programming Problems", *Management Science*, vol. 15, 550-569.

K. FUKUDA et A. PRODON (1996), "The Double Description Method Revisited", *Combinatorics and Computer Science*, vol. 1120, 91-111.

T. GAL (1985), "On the Structure of the Set Bases of a Degenerate Point", *Journal of Optimization Theory and Applications*, vol. 45, p. 577-589.

T. GAL and F. GEUE (1992), "A New Pivoting Rule for Solving Various Degeneracy Problems", *Operations Research Letters*, vol. 11, 23-32.

F. GEUE (1993), "An Improved N-Tree Algorithm for the Enumeration of All Neighbors of a Degenerate Vertex", *Annals of Operations Research*, vol. 47, 361-391.

H. GREENBERG (1975), "An Algorithm for Determining Redundant Inequalities and All Solutions to Convex Polyhedra", *Numerische Mathematik*, vol. 24, 19-26.

M. HAMAMI and S.E. JACOBSEN (1988), "Exhaustive Nondegenerate Conical Processes for Concave Minimization on Convex Polytopes", *Mathematics of Operations Research*, vol. 13, 479-487.

K.L. HOFFMAN (1981), "A Method for Globally Minimizing Concave Functions over Convex Sets", *Mathematical Programming*, vol. 20, 20-32.

R. HORST (1976), "An Algorithm for Nonconvex Programming Problems", *Mathematical Programming*, vol. 10, 312-321.

R. HORST, J. de VRIES et N.V. THOAI (1988), "On Finding New Verticies and Redundant Constraints in Cutting Plane Algorithms for Global Optimization", *Operations Research Letters*, vol. 7, 85-90.

R. HORST and H. TUY (1996), "Global Optimization : Deterministic Approaches", Third edition, Springer-Verlag Editors.

S.E. JACOBSEN (1981), "Convergence of a Tuy-Type-Algorithm for Concave Minimization Subject to Linear Inequality Constraints", *Applied Mathematics and Optimization*, vol. 7, 1-9.

B. JAUMARD and C. MEYER (2001), "On the Convergence of Cone Splitting Algorithms with ϖ-subdivisions", *Journal of Optimization Theory and Applications*, vol. 110, 119-144.

H.-J. KRUSE (1986), "Degeneracy Graphs and the Neighbourhood Problem", *Lecture Notes in Economics and Mathematical Systems*, vol. 260.

M. LOCATELLI (1999), "Finiteness of Conical Algorithms with ϖ-subdivisions", *Mathematical Programming*, vol. 85, 593-616.

M. MANAS et J. NEDOMA (1968), "Finding All Vertices of a Convex Polyhedron", *Numerische Mathematik*, vol. 12, 226-229.

T.H. MATHEISS (1973), "An Algorithm for Determining Irrelevant Constraints and All Vertices in Systems of Linear Inequalities", *Operations Research*, vol. 21, 247-260.

P. McKEOWN (1975), "A Vertex Ranking Procedure for Solving the Linear Fixed Charge Problem", *Operations Research*, vol. 23, 1182-1191.

T.S. MOTZKIN, H. RAIFFA, G.L. THOMPSON et R.M. THRALL (1953), "The Double Description Method", Contribution to the Theory of Games II, *Annals of Mathematics Studies*, 51-73.

K.G. MURTY (1968), "Solving the Fixed Charge Problem by Ranking the Extreme Points", *Operations Research*, vol. 16, 268-279.

B.F. SHERMAN (1977), "A Counterexample to Greenberg's Algorithm for Solving Linear Inequalities", *Numerische Mathematik*, vol. 27, 491-492.

T.V. THIEU, B.T. TAM et V.T. BAN (1983), "An Outer Approximation Method for Globally Minimizing a Concave Function Over a Compact Convex Set", *Acta Mathematica Vietnamica*, vol. 8, 21-40.

N.V. THOAI and H. TUY (1980), "Convergent Algorithm for Minimizing a Concave Function", *Mathematics of Operations Reasearch*, vol. 5, 556-566.

H. TUY (1964), "Concave Minimization under Linear Constraints", *Soviet Mathematics Doklady Academii Nauk SSSR*, vol. 5, 1437-1440.

H. TUY (1990), "On a Polyhedral Annexation Method for Concave Minimization", *Optimization and Mathematical Economics*, 248-260.

H. TUY (1991), "Normal Conical Algorithm for Concave Minimization Over Polytopes", *Mathematical Programming*, vol. 51, 229-245.

H. VAISH and C.M. SHETTY (1976), "The Bilinear Programming Problem", *Naval Research Logistics Quarterly*, vol. 23, 303-309.

H. UZAWA (1958), "A Theorem on Convex Polyhedral Cones", *Studies in Linear and Non-Linear Programming*, Stanford University Press, 23-31.

P.B. ZWART (1973), "Nonlinear Programming : Counterexamples to two Global Optimization Algorithms", *Operations Research*, vol. 21, 1260-1266.

P.B. ZWART (1974), "Global Maximization of a Convex Function with Linear Inequality Constraints", *Operations Research*, vol. 22, 602-609.

Chapitre 3

Extension to the L – DG⁺ Algorithm for the Enumeration of Extreme Rays

La référence de cet article est :
E. SPRINGUEL, B. GENDRON and P. SORIANO (2005), "*Extension to the L – DG⁺ for the Enumeration of Extreme Rays*", soumis à Discrete Applied Mathematics en Octobre 2005.

Extension to the L – DG$^+$ Algorithm for the Enumeration of Extreme Rays

Eric Springuel[1,2], Bernard Gendron[1,2] and Patrick Soriano[2,3]

[1] Department of Computer Science and Operations Research, University of Montréal
2920, Chemin de la Tour, Montréal, Québec, Canada, H3T 1J4

[2] Centre for Research on Transportation
2920, Chemin de la Tour, Montréal, Québec, Canada, H3T 1J4

[3] Department of Management Science, H.E.C. Montréal
3000, Chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada, H3T 2A7

Summary

In this paper, we extend the L – DG$^+$ algorithm of Geue for the enumeration of extreme rays of a polyhedral cone given its hyperplane representation through the exploitation of the concept underlying the LRS algorithm of Avis. The resulting algorithm is more efficient since the searching cost to append any basis to the subgraph of DG$^+$ is totally eliminated. As a by-product, we introduce a generalized LRS algorithm that uses any arbitrary lexicographic basic variable selection rule for the enumeration of the extreme points and extreme rays of a polyhedron given its hyperplane representation via the enumeration of the extreme rays of its homogenization.

3.1 *Introduction*

Consider the hyperplane representation

$$Hx \geq 0$$

of an n-dimensional polyhedral cone K pointed on the origin, where $H \in \Re^p \times \Re^n$ and p is the number of non-redundant hyperplanes (including, if any, the non-redundant non-negativity variable constraints). It is sometimes required to determine the extreme ray representation

$$x = R\lambda, \lambda \geq 0$$

of K where $R \in \Re^n \times \Re^q$ and q is the number of extreme rays of K. Within the context of pivoting enumeration algorithms, the k-th basis matrix B_k of the unique extreme point of K is a subset of p linearly independent columns of the augmented system

$$Hx - Is = 0, s \geq 0,$$

where $I \in \Re^p \times \Re^p$ is the identity matrix and s are surplus variables, while its cobasis matrix N_k is defined as the subset of remaining columns. By letting $y^T = (x^T, s^T)$ and by defining y_{B_k} to be the set of basic variables of B_k having as a subset $y_{B_k}^{\min}$ which are basic s variables and y_{N_k} to be the set of non-basic variables of B_k, if the first n columns of B_k are those of H while the last $p - n$ columns are those of $-I$ such that the columns of B_k are linearly independent, then the augmented system, rewritten as

$$B_k y_{B_k} + N_k y_{N_k} = 0, y_{n+1} \geq 0, \ldots, y_{n+p} \geq 0,$$

is equivalent to

$$y_{B_k} = -(B_k)^{-1}(N_k) y_{N_k}, y_{n+1} \geq 0, \ldots, y_{n+p} \geq 0$$

since B_k is non-singular. According to this last construction, if $p = n$, then there is a unique combination of columns which means that there is only B_0 and all the extreme rays of K are identified in the columns of $-(B_0)^{-1}N_0$. However, if $p > n$, then there are $(p!)/(n!(p-n)!)$ combinations of columns to form a B_k of which not necessarily all satisfy the linear independence condition. An extreme ray of K is then identified by the first n elements of a column of $-(B_k)^{-1}N_k$ if the last $p - n$ elements of this column are positive because, otherwise, a positive scalar of this column violates the non-negativity constraints of $y_{B_k}^{\min}$. Instead of trying all the combinations in order to enumerate all the extreme rays of K given its hyperplane representation, Kruse (1986) introduces an algorithm which uses the standard lexicographic basic variable selection rule of Dantzig, Orden and Wolfe (1955) in order to construct a subgraph of the positive degeneracy graph DG^+. In this graph introduced by Gal (1985), each node corresponds to a y_{B_k} while an edge exists between two nodes if and only if it is possible to move from one to the other by a pivot on a positive element. Whenever this algorithm generates any y_{B_k}, it is appended to the subgraph if and only if it is not found in the current subgraph. To reduce the searching cost to append any y_{B_k}, Geue (1993) proposes the $L - DG^+$ algorithm based on an arbitrary lexicographic basic variable selection rule L which includes as a special case the transition-node-pivoting rule of Gal and Geue (1992) that determines only y_{B_k} such that $-(B_k)^{-1}N_k$ identifies at least one extreme ray of K. Restricted to the enumeration of extreme rays of K given its hyperplane representation, the lexicographic reverse search algorithm of Avis (2000) (*LRS* algorithm for short) is such that the searching cost to append any node to the constructed tree of y_{B_k} is null due to an appropriate initialization and to an intelligent use of the non-ambiguity of the standard lexicographic basic variable selection rule given a non-ambiguous non-basic variable selection rule.

In this paper, we extend the $L - DG^+$ algorithm for the enumeration of the extreme rays of K given its hyperplane representation through the exploitation of the concept underlying the *LRS* algorithm. The resulting algorithm is more efficient since the searching cost to append any y_{B_k} to the subgraph of DG^+ is totally eliminated. As a by-product, we introduce a generalized *LRS* algorithm that uses any arbitrary lexicographic basic variable selection rule for the enumeration of the extreme points and extreme rays of a polyhedron given its hyperplane representation via the enumeration of the extreme rays of its homogenization.

The paper is organized as follows. The next section presents the extension to the $L - DG^+$ algorithm which includes as a special case the transition-node-pivoting rule. The third section describes the generalized *LRS* algorithm. In the fourth section, we analyze numerical results of experiments. Finally, the last section is dedicated to concluding remarks.

3.2 *Extension to the L – DG⁺ Algorithm*

As in the case of the $L - DG^+$ algorithm, it is first necessary to have an initial basis matrix B_0 and an arbitrary non-singular matrix $\overline{B}$ such that the rows of the matrix

$$L(B_0) = \left(0 \ (B_0)^{-1}\overline{B}\right)$$

of $y_{B_0}^{\min}$ are lex-positive vectors (i.e. the first non-zero element of the vector is positive) which means that $L(B_0)$ is said to be a lex-positive matrix. For any choice of $\overline{B}$, there always exists $L(B_0)$ being a lex-positive matrix since K is a full dimensional polyhedral cone defined only by non-redundant constraints. Indeed, it is always possible to perform pivots on the negative element found in the row of the lexicographically negative $\tilde{y}_0 \in y_{B_0}^{\min}$ with minimal index and in the column of $\bar{y}_0 \in y_{N_0}$ with minimal index.

Secondly, the extension of the $L - DG^+$ algorithm requires that the rows of H are permuted such that the last n columns of $-I$ are forming N_0, i.e. $y_{B_0} = (y_1,\ldots,y_p)^T = (x_1,\ldots,x_n,s_1,\ldots,s_{p-n})^T$, $y_{B_0}^{\min} = (y_{n+1},\ldots,y_p)^T = (s_1,\ldots,s_{p-n})^T$ and $y_{N_0} = (y_{p+1},\ldots,y_{p+n})^T = (s_{p-n+1},\ldots,s_p)^T$. For a given $\bar{y}_0 \in y_{N_0}$, which is not identifying an extreme ray of K, $\tilde{y}_0 \in y_{B_0}^{\min}$ is uniquely determined by the arbitrary lexicographic basic variable selection rule as the lexicographically minimum vector of

$$\left\{ \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_0)^{-1}_{\tilde{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}} \text{ such that } (B_0)^{-1}_{\tilde{y}_0\bullet}(N_0)_{\bullet\bar{y}_0} > 0 \right\} \quad \text{(II.1)}$$

since $L(B_0)$ has full row rank. The following proposition extends the proof of Avis (2000) to any arbitrary lexicographic basic variable selection rule.

Proposition

The resulting y_{B_1} having basic variables $y_{B_0} + \bar{y}_0 - \tilde{y}_0$ is such that $L(B_1)$ remains a lex-positive matrix and given $\bar{y}_1 \in N_1$ such that $\bar{y}_1 = \tilde{y}_0$, $\tilde{y}_1 \in y_{B_1}^{\min}$ is uniquely determined by the arbitrary lexicographic basic variable selection rule as the lexicographically minimum vector of

$$\left\{ \frac{L(B_1)_{\bar{y}_1\bullet}}{(B_1)^{-1}_{\tilde{y}_1\bullet}(N_1)_{\bullet\bar{y}_1}} \text{ such that } (B_1)^{-1}_{\tilde{y}_1\bullet}(N_1)_{\bullet\bar{y}_1} > 0 \right\} \quad \text{(II.2)}$$

such that $\tilde{y}_1 = \bar{y}_0$.

Proof

The pivot operation on element $(B_0)^{-1}_{\tilde{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}$ determined according to (II.1) produces $L(B_1)$ defined as

$$L(B_1)_{\bar{y}_0\bullet} = \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_0)^{-1}_{\tilde{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}} \text{ and } L(B_1)_{\hat{y}_0\bullet} = L(B_0)_{\hat{y}_0\bullet} - \frac{\left((B_0)^{-1}_{\tilde{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}\right)L(B_0)_{\bar{y}_0\bullet}}{(B_0)^{-1}_{\tilde{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}} \quad \text{(II.3)}$$

where $\hat{y}_0 \in y_{B_0}$ and $\hat{y}_0 \neq \tilde{y}_0$. In order to prove that $L(B_1)$ remains a lex-positive matrix, it is sufficient to prove that $L(B_1)_{\hat{y}_0\bullet}$ are lex-positive vectors. For each $\hat{y}_0$ such that

$(B_0)^{-1}_{\hat{y}_0\bullet}(N_0)_{\bullet\bar{y}_0} < 0$, $L(B_1)_{\hat{y}_0\bullet}$ is a lex-positive vector because it is equal to the sum of two lex-positive vectors. For each $\hat{y}_0$ such that $(B_0)^{-1}_{\hat{y}_0\bullet}(N_0)_{\bullet\bar{y}_0} = 0$, $L(B_1)_{\hat{y}_0\bullet}$ is obviously a lex-positive vector. For each $\hat{y}_0$ such that $(B_0)^{-1}_{\hat{y}_0\bullet}(N_0)_{\bullet\bar{y}_0} > 0$, $L(B_1)_{\hat{y}_0\bullet}$ is a lex-positive vector because the last equation of (II.3), rewritten as

$$\frac{L(B_1)_{\bar{y}_0\bullet}}{(B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}} = \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}} - \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}},$$

is such that its last part is a lex-positive vector since $L(B_0)_{\bar{y}_0\bullet}$ is the lexicographically minimum vector determined according to (II.1). Since $(B_0)^{-1}(N_0)_{\bullet\bar{y}_0}$ is the column with 1 in position of row of $\bar{y}_0$ while all other elements are 0, the pivot operation on element $(B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}$ determined according to (II.1) produces the resulting column $(B_1)^{-1}(N_1)_{\bullet\bar{y}_0}$ given by

$$(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0} = \frac{1}{(B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}} \text{ and } (B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0} = -\left((B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}\right)\left((B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}\right)$$

(II.4).

Now, it is possible to rewrite the last equation of (II.3) as

$$\frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} = \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} - \frac{\left((B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}\right)L(B_0)_{\bar{y}_0\bullet}}{\left((B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}\right)(B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}},$$

but the insertion of the first equation of (II.3) leads to

$$\frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} = \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} - \frac{\left((B_0)^{-1}_{\bar{y}_0\bullet}(N_0)_{\bullet\bar{y}_0}\right)L(B_1)_{\bar{y}_0\bullet}}{\left((B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}\right)}$$

while the insertion of the last equation of (II.4) gives

$$\frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} = \frac{L(B_0)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} + \frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}}.$$

This implies for $\hat{y}_0$ where $(B_1)^{-1}_{\hat{y}_0\bullet}(N_1)_{\bullet\bar{y}_0} > 0$ that

$$\frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} > \frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}}$$

in the lexicographical sense because

$$\frac{L(B_0)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} = \frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}} - \frac{L(B_1)_{\bar{y}_0\bullet}}{(B_1)^{-1}_{\bar{y}_0\bullet}(N_1)_{\bullet\bar{y}_0}}$$

is a lex-positive vector, which means that $\tilde{y}_1 = \bar{y}_0$ is the minimizer for $L(B_1)$ according to (II.2), given that $\bar{y}_1 \in N_1$ is such that $\bar{y}_1 = \tilde{y}_0$. □

It is important to point out that the above proposition is valid for any couple of neighboring bases induced by the arbitrary lexicographic basic variable selection rule, which means that $L(B_k)$ always remains a lex-positive matrix over this subgraph of DG^+ (assuming $L(B_0)$ is a lex-positive matrix).

A third requirement to extend the $L - DG^+$ algorithm is to define an objective function $c^T y$ to be maximized such that the first p elements of c are 0 while the remaining n elements are -1 in order to y_{B_0} to be the unique optimal basis. Specifically, y_{B_0} is an optimal basis since its

reduced cost vector given by

$$c_{N_0}^T - c_{B_0}^T (B_0)^{-1} N_0$$

has all its elements equal to -1 and is unique since for any $y_{B_k} \neq y_{B_0}$ of the subgraph of DG^+ induced by the arbitrary lexicographic basic variable selection rule, the lexicographical objective function value vector $c_{B_k}^T (B_k)^{-1} \overline{B}$ of y_{B_k} is stricly smaller than the lexicographical objective function value vector $c_{B_0}^T (B_0)^{-1} \overline{B}$ of y_{B_0} because c_{B_0} has all its elements equal to zero while c_{B_k} has some of its elements equal to -1.

Given this framework, the extended $L - DG^+$ algorithm consists in constructing a tree of y_{B_k} where the root is y_{B_0} and the children of a given y_{B_k} in the tree are its neighboring bases $y_{B_{k'}}$ such that from each $y_{B_{k'}}$, the algorithm pivots back to its father y_{B_k} according to the arbitrary lexicographic basic variable selection rule given the non-basic variable with minimal index and having a positive reduced cost. Since the non-basic variable selection rule and the arbitrary lexicographic basic variable selection rule are non-ambiguously determining a unique pivot on a positive element when pivoting back, no y_{B_k} of the tree is reachable by different paths because, otherwise, the uniqueness of the choice of the pivot element would be contradicted. By proceeding in this way, every y_{B_k} of the subgraph of DG^+ induced by the arbitrary lexicographic basic variable selection rule is generated, since there exists a path which connects any y_{B_k} to y_{B_0}. When initialized with y_{B_0} and the same $\overline{B}$, the $L - DG^+$ algorithm covers the same set of y_{B_k} because it also generates the complete subgraph of DG^+ induced by the arbitrary lexicographic basic variable selection rule. This implies that all the extreme rays of K are identified over the set of y_{B_k} generated by the extended $L - DG^+$ algorithm as demonstrated by Geue (1993).

However, it is important that each extreme ray of K is output only once when processing the extended $L - DG^+$ algorithm. Avis (2000) determines a sufficient condition to establish a unique lexicographic minimal $y_{B_k^*}$ that identifies an extreme ray of K. The validity of the condition is guaranteed because $y_{B_k^*}$ is an element of the subgraph of DG^+ induced by the standard lexicographic basic variable selection rule. However, the application of this sufficient condition to the subgraph of DG^+ induced by the arbitrary lexicographic basic variable selection rule leads to the determination of an insufficient number of extreme rays of K thus indicating the invalidity of the condition in the general case. Nevertheless, a slight modification is possible for the extended $L - DG^+$ algorithm : given y_{B_k} identifying an extreme ray of K in the first n elements of the column $-(B_k)^{-1}(N_k)_{\cdot \hat{y}_k}$, where $\hat{y}_k \in y_{N_k}$, it is required to verify if this extreme ray is already output only if for each $\tilde{y}_k \in y_{B_k}^{\min}$ such that $(B_k)^{-1}_{\tilde{y}_k \cdot}(N_k)_{\cdot \hat{y}_k} = 0$, $\nexists\ \bar{y}_k \in y_{N_k}$ with its index of s smaller than the index of s of $\tilde{y}_k$ and $(B_k)^{-1}_{\tilde{y}_k \cdot}(N_k)_{\cdot \bar{y}_k} > 0$. In other words, if no neighboring basis $y_{B_{k'}}$ of the subgraph of DG^+ induced by the arbitrary lexicographic basic variable selection rule is lexicographically smaller

than y_{B_k} and identifies the same extreme ray of K, then it is needed to verify if this extreme ray is already output by searching the key corresponding to the saturated constraint of the extreme ray identified in y_{B_k} within a hash table containing keys corresponding to the saturated constraints for the extreme rays already output.

In order to make easier the statement of the extended $L-DG^+$ algorithm, four basic functions are determined :

1- *pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ performs the pivot on element $(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \bar{y}_k}$ where $\bar{y}_k \in y_{N_k}$ and $\tilde{y}_k \in y^{\min}_{B_k}$ are given and such that $\bar{y}_k$ replaces $\tilde{y}_k$;

2- *select_pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ returns first $\bar{y}_k \in y_{N_k}$ with minimal index and a positive reduced cost (assuming that $y_{B_k} \neq y_{B_0}$) and second $\tilde{y}_k \in y^{\min}_{B_k}$ determined according to the arbitrary lexicographic basic variable selection rule given $\bar{y}_k$;

3- *reverse*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ determines if it is *True* for a given $\bar{y}_k \in y_{N_k}$ and $\tilde{y}_k \in y^{\min}_{B_k}$ that *select_pivot* applied to the basis $y_{B_{k'}}$ having basic variables $y_{B_k} + \bar{y}_k - \tilde{y}_k$ returns the element $\tilde{y}_k \in y_{N_{k'}}$ and $\bar{y}_k \in y^{\min}_{B_{k'}}$ (note that $\tilde{y}_k \in y^{\min}_{B_k}$ is meaningful only when *reverse*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ is *True*);

4- for y_{B_k} representing an extreme ray of K in the first n elements of the column $-(B_k)^{-1}(N_k)_{\bullet \hat{y}_k}$ where $\hat{y}_k \in y_{N_k}$, *verify_output_file*$(y_{B_k}, \hat{y}_k)$ determines if it is *True* for y_{B_k} that for each $\tilde{y}_k \in y^{\min}_{B_k}$ such that $(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \hat{y}_k} = 0$, $\nexists\, \bar{y}_k \in y_{N_k}$ with its index of s smaller than the index of s of $\tilde{y}_k$ and $(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \bar{y}_k} > 0$.

EXTENDED $L-DG^+$ ALGORITHM

Initialization :

Let $\overline{B}$ and B_0 be a given.

While $L(B_0) = \left(0 \;\; (B_0)^{-1}\overline{B}\right)$ is not a lex-positive matrix, do :

Find the lexicographically negative $\tilde{y}_0 \in y^{\min}_{B_0}$ with minimal index.

Find $\bar{y}_0 \in y_{N_0}$ with minimal index and $(B_0)^{-1}_{\tilde{y}_0 \bullet}(N_0)_{\bullet \bar{y}_0} < 0$.

Perform *pivot*$(y_{B_0}, \bar{y}_0, \tilde{y}_0)$ and let y_{B_0} be the resulting basis.

Permute the rows of H such that the last n columns of $-I$ are forming N_0.

According to this permutation, change the rows of B_0, $\overline{B}$ and $L(B_0)$.

Let the first p elements of c be 0 and the remaining n elements be -1 and $i = n+1$.

While $i \leq n+p$ or $y_{B_k} \neq y_{B_0}$, do :

While $i \leq n+p$, do :

Let $\bar{y}_k$ be such that its index of s is i.

If $\bar{y}_k \in y_{N_k}$, then :

If $-(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \bar{y}_k} \geq 0 \;\forall\; \tilde{y}_k \in y^{\min}_{B_k}$, then :

If *verify_output_file*$(y_{B_k}, \bar{y}_k) =$ *True*, then :

If this ray is not in the output file, then write it to the output file.

Else :

If $reverse(y_{B_k}, \bar{y}_k, \tilde{y}_k) = True$, then :

Perform $pivot(y_{B_k}, \bar{y}_k, \tilde{y}_k)$.

Let $i = n + 1$.

Else $i = i + 1$.

Else $i = i + 1$.

Perform $select\_pivot(y_{B_k}, \bar{y}_k, \tilde{y}_k)$.

Let i be the index of s of $\bar{y}_k$ and $i = i + 1$.

Perform $pivot(y_{B_k}, \bar{y}_k, \tilde{y}_k)$.

The most critical function of the extended $L - DG^+$ algorithm is $reverse(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ which obviously returns *True* only if $c_{\bar{y}_k} - c_{B_k}^T (B_k)^{-1}(N_k)_{\bullet \bar{y}_k} < 0$ and for each $\hat{y}_k \in y_{N_k}$ with its index of s smaller than the index of s of $\bar{y}_k$, its resulting reduced cost element

$$\left(c_{\hat{y}_k}^T - c_{B_k}^T (B_k)^{-1}(N_k)_{\bullet \hat{y}_k}\right) - \frac{\left(c_{\bar{y}_k} - c_{B_k}^T (B_k)^{-1}(N_k)_{\bullet \bar{y}_k}\right)}{(B_k)^{-1}_{\bar{y}_k \bullet}(N_k)_{\bullet \bar{y}_k}} (B_k)^{-1}_{\bar{y}_k \bullet}(N_k)_{\bullet \hat{y}_k}$$

is negative if pivoting on $(B_k)^{-1}_{\bar{y}_k \bullet}(N_k)_{\bullet \bar{y}_k}$.

This extended $L - DG^+$ algorithm may identify y_{B_k} which are useless to the determination of any extreme ray of K. This problem is solved by the introduction of the transition-node-pivoting rule which is a special case of $\bar{B}$ as demonstrated by Geue (1993). More specifically, when $\bar{B} = -B_0$ and once the while loop of the initialization phase is completed, B_0 identifies at least one extreme ray of K and all y_{B_k} of the subgraph of DG^+ induced by this particular lexicographic basic variable selection rule are also identifying at least one extreme ray of K (proof in Geue (1993)). If all the extreme rays of K are non-degenerate (i.e. defined only by $n - 1$ hyperplanes), then each y_{B_k} generated by the extended $L - DG^+$ algorithm initialized with $\bar{B} = -B_0$ is such that none of the last n elements of the columns of $-(B_k)^{-1}N_k$ which identify the extreme rays of K are 0 and *verify_output_file* always returns *True*. Note that the number of y_{B_k} generated by the extended $L - DG^+$ algorithm initialized with $\bar{B} = -B_0$ is bounded from above by the number of extreme rays of K if all the extreme rays of K are non-degenerate, as proven by Geue (1993) (the proof is obviously valid in the current context). However, there exists bounded polyhedron families for which the number of y_{B_k} generated by the extended $L - DG^+$ algorithm (for any choice of $\bar{B}$) is not polynomially bounded from above by the number of extreme rays of K since the results of Armand (1993) are still valid here.

3.3 *Generalization of LRS algorithm*

From the extended $L - DG^+$ algorithm, we can *almost* derive a generalization of *LRS* algorithm

to any arbitrary lexicographic basic variable selection rule in order to enumerate the extreme points and extreme rays of an n-dimensional polyhedron given its hyperplane representation $D = \{x \in \Re^n \mid Ax \geq b, x \geq 0\}$ where $A \in \Re^m \times \Re^n$ and $b \in \Re^m$. More specifically, given any y_{B_0} associated to an extreme point of D and $\overline{B} = -B_0$, the application of the extended $L - DG^+$ algorithm leads to the observation that when it is pivoting to a degenerate neighboring extreme point, $L(B_k)$ does not remain a lex-positive matrix. Incidentally, it is necessary to pivot on negative elements as in the while loop of the initialization phase of the extended $L - DG^+$ algorithm in order for $L(B_k)$ to become a lex-positve matrix and to stack the reverse negative pivot sequence which is used when pivoting back to y_{B_0} prior to pivot out of the degenerate extreme point in question. Since all y_{B_k} visited when performing negative pivots are useless to the determination of any extreme point or any extreme ray of D, the generalization of *LRS* algorithm to any arbitrary lexicographic basic variable selection rule is then given by the extended $L - DG^+$ algorithm applied to the enumeration of the extreme rays of the homogenization $D_{Hom} = \{x \in \Re^n, x_n \in \Re \mid Ax - bx_{n+1} \geq 0, x \geq 0, x_{n+1} \geq 0\}$ of D. For every extreme ray of D_{Hom} determined, if $x_{n+1} = 0$, then x is an extreme ray of D, while if $x_{n+1} > 0$, then $\frac{1}{x_{n+1}}x$ is an extreme point of D.

3.4 *Numerical Experiments and Analyses*

The first part of this section presents the numerical experiments comparing the extended $L - DG^+$ algorithm to the $L - DG^+$ algorithm for different choices of $\overline{B}$. The second part of this section shows numerical experiments demonstrating that the generalized *LRS* algorithm initialized with $\overline{B} = -B_0$ generates a smaller number of y_{B_k} when compared to *LRS* algorithm.

Instances of K of various sizes, given p and n, are defined pseudo-randomly in order to test the extended $L - DG^+$ algorithm and the $L - DG^+$ algorithm. The first $n - 1$ coefficients of the first $p - n$ rows of H are pseudo-randomly generated in the interval [-1, 1]. The n^{th} coefficient is pseudo-random number generated in the interval [-2, 0] added to the negative value of the sum of the elements of the row in question. The last n rows of H are the usual non-negativity constraints. Concerning the construction of D, some of them correspond to the bounded instances of unhomogenization of K as previously constructed while the polyhedron of Problem_2.3 of Floudas and Pardalos (1990) and of Problem_A, defined in Appendix, are also used. It has to be pointed out that prior to apply either the extended $L - DG^+$ algorithm, the generalized *LRS* algorithm or the $L - DG^+$ algorithm, a subroutine is performed in order to remove all redundant constraints.

Regarding the implementation, the extended $L - DG^+$ algorithm, the generalized *LRS* algorithm and the $L - DG^+$ algorithm are programmed in C with the integer pivoting rules of Edmonds and Maurras (1997) performed in exact integer arithmetic using GNU Multiple

Precision library (version 4.1.4) because the algorithms are extremely sensitive to numerical errors; GNU hash table library (version 0.5.4) is used to perform searching whenever necessary; Fortran BLAS\LAPACK libraries (version 3.0) are used for linear algebra computations in floating point arithmetic; and ILOG-CPLEX (version 8.1) is used for maintaining polyhedra information and LP optimization. All tests are performed on a Sun Ultra-Sparc 1.2 GHz computer with 2 GB of RAM where the Sun f77 compiler is used to compile the BLAS\LAPACK libraries and the Sun C compiler is used to compile all other programmed algorithms and libraries.

3.4.1 *Comparison of L-DG$^+$ Algorithms*

In order to compare the efficiency of the extended $L-DG^+$ algorithm and the $L-DG^+$ algorithm, recall that it is required for the latter algorithm to verify if a given y_{B_k} is already in the subgraph of DG^+. To this end, a hash table is used where the key is constructed by setting to 1 the bit of long long integer at position y_{B_k} and to 0 the bit of the same long long integer at position y_{N_k} (concatenation is used whenever $p > 64$). We tested two values for the size of the hash table (p^2 and p^3) which are fairly large compared to the expected number of generated B_k of the subgraph of DG^+; these large values tend to reduce the number of collisions. Table 3.1 reports the CPU time in seconds for the application of the two algorithms given the same choice of B_0 and $\overline{B}$ for different instances of K. If $\overline{B} = 1$, then $\overline{B} = B_0$; if $\overline{B} = 2$, then $\overline{B} = I$; and if $\overline{B} = 3$, then $\overline{B} = -B_0$ while n, p, q and B_k correspond respectively to the dimension, the number of constraints, the number of extreme rays and the number of generated bases of the subgraph of DG^+ for the problem in question.

Table 3.1.

K	$\overline{B}$	n	p	q	B_k	$L-DG^+$ (p^2)	$L-DG^+$ (p^3)	Ext. $L-DG^+$
Rand1_10_25	1	10	25	5324	7334	9.47	9.40	8.58
Rand2_10_25	1	10	25	4810	8416	10.97	10.97	9.98
Rand3_10_25	1	10	25	6121	9483	12.57	12.53	11.51
Rand1_10_25	2	10	25	5324	3430	5.11	5.12	5.03
Rand2_10_25	2	10	25	4810	3158	4.89	4.90	4.77
Rand3_10_25	2	10	25	6121	4065	6.17	6.09	6.04
Rand1_10_25	3	10	25	5324	3526	4.66	4.65	4.55
Rand2_10_25	3	10	25	4810	1948	2.93	2.92	2.87
Rand3_10_25	3	10	25	6121	2957	3.93	3.93	3.84
Rand1_13_30	1	13	30	57514	73484	188.25	182.70	165.40
Rand2_13_30	1	13	30	39708	73561	177.88	170.28	154.90
Rand3_13_30	1	13	30	64940	89931	209.70	200.00	180.59
Rand1_13_30	2	13	30	57514	31636	87.98	87.23	84.56
Rand2_13_30	2	13	30	39708	31145	82.69	82.13	80.65
Rand3_13_30	2	13	30	64940	31701	87.90	87.01	84.82
Rand1_13_30	3	13	30	57514	27007	68.48	67.47	64.71
Rand2_13_30	3	13	30	39708	20181	49.32	49.38	48.11
Rand3_13_30	3	13	30	64940	31741	77.29	75.54	72.86

Based on these results, the CPU time required by the extended $L-DG^+$ algorithm on these instances of K is smaller by about 9-10% comparatively to $L-DG^+$ algorithm with the choice of $\overline{B}=B_0$ while for the two other choices of $\overline{B}$ (i.e. $\overline{B}=I$ or $\overline{B}=-B_0$), the extended $L-DG^+$ algorithm perform only slightly better than the $L-DG^+$ algorithm. This phenomenon is due to the rule of thumb for the determination of the size of the hash table which is inadequate when $\overline{B}=B_0$ and acceptable when $\overline{B}=I$ or $\overline{B}=-B_0$ and it is apparent in all these cases that if the size of the hash table diminishes, then the CPU time increases. Since it is hard to know a priori the number of B_k in the subgraph of DG^+ in order to determine the appropriate size of the hash table for the $L-DG^+$ algorithm, the extended $L-DG^+$ algorithm is a good alternative. Moreover, the fact that the CPU time of the extended $L-DG^+$ algorithm is always smaller than the CPU time of the $L-DG^+$ algorithm shows that the extended $L-DG^+$ algorithm is an efficient alternative.

3.4.2 *Comparison of generalized LRS and LRS algorithm*

To demonstrate that the generalized *LRS* algorithm initialized with $\overline{B} = -B_0$ generates a smaller number of B_k when compared to *LRS* algorithm, we use version 4.2 of the computer implementation of *LRS* algorithm (using implicitly $\overline{B} = B_0$) found on the Web site of Avis to perform the comparison for which the redundant constraints are always removed by the function *redund* prior any application of the algorithm. Table 3.2 reports the number of B_k generated by *LRS* algorithm applied to the bounded instances of the unhomogenization of K where n, p and ext. points correspond, respectively, to the dimension, the number of constraints and the number of extreme points for the problem in question.

Table 3.2.

K	n	p	ext. points	B_k
Rand1_10_25	10	25	5324	5324
Rand2_10_25	10	25	4810	4810
Rand3_10_25	10	25	6121	6121
Rand2_13_30	13	30	39708	39708
Rand3_13_30	13	30	64940	64940

At first glance, the number of B_k of bounded instances of the unhomogenization of K reported in Table 3.2 is always greater than or equal to the number of B_k generated by the extended $L - DG^+$ algorithm initialized with $\overline{B} = -B_0$ applied to the corresponding K reported in Table 3.1. This phenomenon is a direct consequence of the result of Geue (1993) since all the extreme points of bounded instances of the unhomogenization of K are non-degenerate because their number is equal to the number of B_k generated by *LRS* algorithm applied to the polyhedron in question. Incidentally, this observation leads to the conclusion that the generalized *LRS* algorithm, initialized with $\overline{B} = -B_0$, will always generate a smaller number of B_k compared to *LRS* algorithm, when applied to non-degenerate bounded polyhedra. Table 3.3 indicates that the number of B_k generated by *LRS* algorithm applied to K is sometimes greater than the number of B_k generated by *LRS* algorithm applied to the unhomogenization of K. This indicates that the above conclusion is not the fortuitous result of the homogeneization underlying the generalized *LRS* algorithm. It has to be pointed out that the initialization of the implementation of *LRS* algorithm is slightly different than the initialization of the implementation for the extended $L - DG^+$ algorithm thus explaining the small difference in the number of B_k generated when comparing results of Table 3.1 and Table 3.3.

Table 3.3.

K	n	p	ext. rays	B_k
Rand1_10_25	10	25	5324	7208
Rand2_10_25	10	25	4810	8486
Rand3_10_25	10	25	6121	9291
Rand2_13_30	13	30	39708	64807
Rand3_13_30	13	30	64940	73760

For bounded instances of D having some degenerate extreme points such as Problem_2.3 and Problem_A, Table 3.4 reports the number of B_k generated by the generalized *LRS* algorithm initialized with $\overline{B} = -B_0$ (Alg. = 1) and the number of B_k generated by *LRS* algorithm (Alg. = 2) both applied to these two instances.

Table 3.4.

D	n	p	Alg.	ext. points	B_k
Problem_2.3	13	32	1	2744	1536
Problem_2.3	13	32	2	2744	4096
Problem_A	10	35	1	552	136
Problem_A	10	35	2	552	603

According to these results, the above conclusion must be qualified even for some bounded instances of D having degenerate extreme points : the generalized *LRS* algorithm initialized with $\overline{B} = -B_0$ generates a smaller number of B_k compared to *LRS* algorithm for some bounded instances of D having degenerate extreme points. This is due to the fact that it is always possible to identify a neighboring extreme point among the columns of $-(B_k)^{-1}N_k$ with the generalized *LRS* algorithm initialized with $\overline{B} = -B_0$ while it is not the case with *LRS* algorithm.

3.5 *Conclusion*

In this paper, we demonstrate that the extended $L - DG^+$ algorithm outperforms the $L - DG^+$ algorithm from a theoretical, as well as a practical point of view. As a by-product, the generalized *LRS* algorithm using the basic variable selection rule initialized with $\overline{B} = -B_0$, always generates a smaller number of B_k compared to the *LRS* algorithm, when all the extreme points of bounded instances of D are non-degenerate. For bounded instances of D having degenerate extreme points, our numerical experiments confirm this result.

3.6 *Appendix*

Problem_A has the respectively the technology matrix and RHS compatible with the problem structure introduced in the first section

$$\begin{bmatrix}
-1 & -2 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1
\end{bmatrix}
\begin{bmatrix}
-0.9968 \\
0.9968 \\
0.9964 \\
0.0004 \\
0 \\
0 \\
-0.0004 \\
0.0004 \\
0 \\
0 \\
-0.0004 \\
0.0004 \\
0 \\
0 \\
-0.0004 \\
0.0004 \\
0.897 \\
-0.9968
\end{bmatrix}$$

where it is necessary to include explicitly the variables' bound constraints $0 \leq x \leq 0.0998$ in the desciption of D in the form of $\geq$ inequality constraints to be compatible with the problem structure introduced in the first section.

Bibliographie

P. ARMAND (1993), "Bounds on the Number of Vertices of Perturbed Polyhedra", *Annals of Operations Research*, vol. 47, 249-269.

D. AVIS (2000), "A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm", *Polytopes - Combinatorics and Computation*, vol. 29, 177-198.

G.B. DANTZIG, A. ORDEN and P. WOLFE (1955), "The Generalized Simplex Method for Minimizing a Linear Function Form under Linear Inequality Restraints", *Pacific Journal of Mathematics*, vol. 5, 183-195.

J. EDMONDS and J.F. MAURRAS (1997), "Note sur les Q-matrices d'Edmonds", *RAIRO*, vol. 31, 203-209.

C.A. FLOUDAS and P.M. PARDALOS (1990), "A Collection of Test Problems for Constrained Global Optimization Algorithms", *Lecture Notes in Computer Science*, vol. 455.

T. GAL (1985), "On the Structure of the Set Bases of a Degenerate Point", *Journal of Optimization Theory and Applications*, vol. 45, 577-589.

T. GAL and F. GEUE (1992), "A New Pivoting Rule for Solving Various Degeneracy Problems", *Operations Research Letters*, vol. 11, 23-32.

F. GEUE (1993), "An Improved N-Tree Algorithm for the Enumeration of All Neighbors of a Degenerate Vertex", *Annals of Operations Research*, vol. 47, 361-391.

H.-J. KRUSE (1986), "Degeneracy Graphs and the Neighbourhood Problem", *Lecture Notes in Economics and Mathematical Systems*, vol. 260.

Chapitre 4

Degeneracy and Enumeration in Basic Concave Programming

La référence de cet article est :
E. SPRINGUEL, B. GENDRON and P. SORIANO (2005), "*Degeneracy and Enumeration in Basic Concave Programming*", soumis à Journal of Global Optimization en Octobre 2005.

Degeneracy and Enumeration in Basic Concave Programming

Eric Springuel[1,2], Bernard Gendron[1,2] and Patrick Soriano[2,3]

[1] Department of Computer Science and Operations Research, University of Montréal
2920, Chemin de la Tour, Montréal, Québec, Canada, H3T 1J4

[2] Centre for Research on Transportation
2920, Chemin de la Tour, Montréal, Québec, Canada, H3T 1J4

[3] Department of Management Science, H.E.C. Montréal
3000, Chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada, H3T 2A7

Summary

In this paper, we numerically compare, as methods for dealing with degeneracy in basic concave programming, the extended $L - DG^+$ algorithm of Springuel, Gendron and Soriano (appropriately initialized in order to replicate the transition-node-pivoting rule of Gal and Geue) and the most efficient double description algorithm, as defined originally by Motzkin, Raiffa, Thompson and Thrall. In addition, we actualize the numerical comparison of constructive enumeration algorithms used for basic concave programming.

4.1 *Introduction*

Minimizing a continuous concave function $f : \Re^n \to \Re$ over a polyhedron $D \subset \Re^n$ is probably the most studied global optimization problem due to the numerous applications found in operations research, engineering and economics. This problem can be written as

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & Ax \geq b \\ & x \geq 0 \end{aligned}$$

where $A \in \Re^m \times \Re^n$ and $b \in \Re^m$. Without loss of generality, we assume that D has a non-empty interior, i.e., that $int(D) = \{x \in \Re^n \mid Ax - b > 0,\ x > 0\} \neq \emptyset$. If this is not the case, then it is always possible to perform variable substitutions from the set of linearly independent constraints that are always saturated for all $x \in D$, in order to obtain a full dimensional polyhedron. When D is compact, Weierstrass's theorem guarantees the existence of at least one global minimum of f over D since f is continuous, while Carathéodory's theorem highlights the fact that one global minimum denoted v_f^* is an extreme point of D. When D is unbounded, if there exists at least one extreme ray r of D for which $f(\alpha r) \to -\infty$ as $\alpha \to \infty$, then f is unbounded from below on D. This means that if f is bounded from below on D, then v_f^* is also an extreme point of D. Frequently denoted in the literature as basic concave programming (*BCP*), this problem is said to be multiextremal since the application of standard convex optimization techniques may lead to local minima which are not global minima (where locality

is defined here by the set of neighboring extreme points).

As outlined in Horst and Tuy (1996), there are numerous deterministic algorithms for *BCP* which are based either on extreme point enumeration, cutting hyperplanes, outer approximation, branch and bound, or any combination of two or more of these basic approaches. With the exception of all simplicial and rectangular branch and bound algorithms based respectively on Horst (1976) or Falk and Soland (1969), deterministic algorithms for *BCP* are heavily dependent upon the enumeration of the extreme rays of the smallest polyhedral cone K pointed on an extreme point v of D such that $D \subset K$ given by the subset of p constraints of D

$$Hx \geq d$$

including explicitly the non-negativity constraints defining v. In the context of pivoting enumeration algorithms, the k-th basis matrix B_k of the unique extreme point of K is a subset of p linearly independent columns of the augmented system

$$Hx - Is = d, s \geq 0$$

where $I \in \Re^p \times \Re^p$ is an identity matrix and s are surplus variables while its cobasis matrix N_k is defined as the subset of remaining columns. By letting $y^T = (x^T, s^T)$ and by defining y_{B_k} to be the set of basic variables of B_k having as a subset $y_{B_k}^{\min}$ which are basic s variables and y_{N_k} to be the set of non-basic variables of B_k, if the first n columns of B_k are those of H while the last $p - n$ are those of $-I$ such that the columns of B_k are linearly independent, then the augmented system, rewritten as

$$B_k y_{B_k} + N_k y_{N_k} = d, y_{n+1} \geq 0, \ldots, y_{n+p} \geq 0$$

is equivalent to

$$y_{B_k} = (B_k)^{-1} d - (B_k)^{-1}(N_k) y_{N_k}, y_{n+1} \geq 0, \ldots, y_{n+p} \geq 0$$

since B_k is non-singular. According to this last construction, if $p = n$, then there is a unique combination of columns to form B_0 and all the extreme rays of K are determined by the columns of $-(B_0)^{-1}N_0$. However, if $p > n$, then there are $(p!)/((p-n)!n!)$ combinations of columns to form B_k of which not necessarily all satisfy the linear independence condition. An extreme ray of K is then determined by the first n elements of a column of $-(B_k)^{-1}(N_k)$ if the last $p - n$ elements of this column are positive because, otherwise, a positive scalar of this column violates the non-negativity constraints of $y_{B_k}^{\min}$. Instead of trying all the basis combinations in order to enumerate all the extreme rays of K given its hyperplane representation, a judicious choice is to use the extended $L - DG^+$ algorithm of Springuel, Gendron and Soriano (2005) since it is one of the most efficient pivoting enumeration algorithms treating degeneracy with an arbitrary lexicographic basic variable selection rule. In the context of constructive enumeration algorithms, a judicious choice is to use the most efficient double description algorithm introduced by Motzkin, Raiffa, Thompson and Thrall (1953) because it is originally treating degeneracy by maintaining at each iteration adjacency information of extreme rays of polyhedral cones outer approximating K. Finally, it has to be

pointed out that enumeration algorithms treating degeneracy with the production of the entire face lattice of K are not appropriate since more information than needed is generated.

In the context of *BCP*, we numerically compare as methods for dealing with degeneracy the extended $L-DG^+$ algorithm of Springuel, Gendron and Soriano (appropriately initialized in order to replicate the transition-node-pivoting rule of Gal and Geue (1992)) and the most efficient double description algorithm introduced by Motzkin, Raiffa, Thompson and Thrall where both are used in a partial enumeration perspective, i.e. to enumerate only the extreme rays of K given its hyperplane representation which is defined by a subset of the constraints of D. Traditionally, they have been used in a total enumeration perspective, i.e. to enumerate all the extreme points and extreme rays of D given its hyperplane representation. In addition, we actualize the numerical comparison of constructive enumeration algorithms used for *BCP* performed by Horst, Thoai and de Vries (1988) by embedding the methods for dealing with degeneracy in an adaptation of their constructive enumeration algorithm and by including also the most efficient double description algorithm in the analysis.

The paper is organized as follows. The next section presents a detailed presentation of the methods for dealing with degeneracy in *BCP*. The third section introduces the major *BCP* algorithms affected by degeneracy, namely the cutting hyperplane, outer approximation and polyhedral annexation algorithms. In the fourth section, we analyze numerical results of experiments. Finally, the last section contains the concluding remarks.

4.2 *Methods for Dealing with Degeneracy*

4.2.1 *Extended $L-DG^+$ Algorithm*

Based on the concept of positive degeneracy graph DG^+ introduced by Gal (1985) where each node corresponds to a y_{B_k} of v while an edge exists between two nodes if and only if it is possible to move from one to the other by a pivot on a positive element, Springuel, Gendron and Soriano (2005) introduce the extended $L-DG^+$ algorithm to construct a subgraph of DG^+ which combines the $L-DG^+$ algorithm of Geue (1993) and the concept underlying the lexicographic reverse search (*LRS* algorithm for short) of Avis (2000).

After having translated v to the origin for convenience purposes and having removed the redundant constraints defining K, for a given starting basis y_{B_0} of the unique extreme point of K and $\overline{B} = -B_0$, it is first required to pivot on the negative element found in the row of the lexicographically negative $\tilde{y}_0 \in y_{B_0}^{\min}$ with minimal index and in the column of $\bar{y}_0 \in y_{N_0}$ with minimal index as long as

$$L(B_0) = \left(0 \ (B_0)^{-1}\overline{B}\right)$$

is not a lex-positive matrix (i.e. each row of $y_{B_0}^{\min}$ is such that its first non-zero element is positive). In fact, such pivots are always possible since K is a full dimensional polyhedral cone defined only by non-redundant constraints.

Secondly, it is required to permutate the rows of H such that the last n columns of $-I$ are forming N_0 which means that $y_{B_0} = (y_1, \ldots, y_p)^T = (x_1, \ldots, x_n, s_1, \ldots, s_p)^T$, $y_{B_k}^{\min} = (y_{n+1}, \ldots, y_p)^T = (s_1, \ldots, s_{p-n})^T$ and $y_{N_0} = (y_{p+1}, \ldots, y_{p+n})^T = (s_{p-n+1}, \ldots, s_p)^T$. For a given $\bar{y}_0 \in y_{N_0}$ which is not identifying an extreme ray of K, $\tilde{y}_0 \in y_{B_0}^{\min}$ is uniquely determined by the arbitrary lexicographic basic variable selection rule as the lexicographically minimum vector of

$$\left\{ \frac{L(B_0)_{\bar{y}_0 \bullet}}{(B_0)_{\bar{y}_0 \bullet}^{-1}(N_0)_{\bullet \bar{y}_0}} \text{ such that } (B_0)_{\bar{y}_0 \bullet}^{-1}(N_0)_{\bullet \bar{y}_0} > 0 \right\}$$

since $L(B_0)$ has full row rank. In addition, the resulting y_{B_1} having basic variables $y_{B_0} + \bar{y}_0 - \tilde{y}_0$ is such that $L(B_1)$ remains a lex-positive matrix and for $\tilde{y}_1 \in y_{N_1}$ such that $\bar{y}_1 = \tilde{y}_0$, $\bar{y}_1 \in y_{B_1}^{\min}$ is uniquely determined by the arbitrary lexicographic basic variable selection rule as the lexicographically minimum vector of

$$\left\{ \frac{L(B_1)_{\bar{y}_1 \bullet}}{(B_1)_{\bar{y}_1 \bullet}^{-1}(N_1)_{\bullet \tilde{y}_1}} \text{ such that } (B_1)_{\bar{y}_1 \bullet}^{-1}(N_1)_{\bullet \tilde{y}_1} > 0 \right\}$$

such that $\tilde{y}_1 = \bar{y}_0$, as shown by Springuel, Gendron and Soriano (2005). Since this property is valid for any couple of neighboring bases induced by the arbitrary lexicographic basic variable selection rule, it means that $L(B_k)$ always remains a lex-positive matrix over this subgraph of DG^+ (assuming that it is initialized with $L(B_0)$ being a lex-positive matrix). It is important to point out that Geue (1993) demonstrates that this choice of $\overline{B}$ replicates the transition-node-pivoting rule which identifies only y_{B_k} of v determining at least one extreme ray of K.

A third requirement is to define an objective function $c^T y$ to be maximized such that the first p elements of c_v are 0 while the remaining n elements are -1 in order to y_{B_0} to be the unique optimal basis. Specifically, y_{B_0} is an optimal basis since its reduced cost vector given by

$$c_{N_0}^T - c_{B_0}^T (B_0)^{-1} N_0$$

has all its elements equal to -1 and is unique since for any $y_{B_k} \neq y_{B_0}$ of the subgraph of DG^+ induced by the arbitrary lexicographic basic variable selection rule, the lexicographical objective function value vector $c_{B_k}^T (B_k)^{-1} \overline{B}$ of y_{B_k} is stricly smaller than the lexicographical objective function value vector $c_{B_0}^T (B_0)^{-1} \overline{B}$ of y_{B_0} because c_{B_0} has all its elements equal to zero while c_{B_k} has some of its elements equal to -1.

The concept which underlies the extended $L - DG^+$ algorithm consists in constructing a tree of y_{B_k} where the root is y_{B_0} and the children of a given y_{B_k} in the tree are its neighboring bases $y_{B_{k'}}$ such that from each $y_{B_{k'}}$, the algorithm pivots back to its father y_{B_k} according to the

arbitrary lexicographic basic variable selection rule given the non-basic variable with minimal index having a positive reduced cost. Springuel, Gendron and Soriano (2004) indicate that the extended $L - DG^+$ algorithm is such that no y_{B_k} is appended twice to the sugbraph of DG^+ and all the extreme rays of K are identified.

However, it is important that each extreme ray of K is output only once when processing the extended $L - DG^+$ algorithm. Springuel, Gendron and Soriano (2004) determine a condition which states that for a given y_{B_k} identifying an extreme ray of K in the first n elements of the column $-(B_k)^{-1}(N_k)_{\cdot\hat{y}_k}$ where $\hat{y}_k \in y_{N_k}$, it is required to verify if this extreme ray is already output only if, for each $\tilde{y}_k \in y_{B_k}^{\min}$ such that $(B_k)^{-1}_{\tilde{y}_k\cdot}(N_k)_{\cdot\hat{y}_k} = 0$, $\nexists\, \bar{y}_k \in y_{N_k}$ with its index of s smaller than the index of s of $\tilde{y}_k$ and $(B_k)^{-1}_{\tilde{y}_k\cdot}(N_k)_{\cdot\bar{y}_k} > 0$. It is important to point out that if all the extreme rays of K are non-degenerate (i.e. defined only by $n-1$ hyperplanes), then each y_{B_k} generated by the extended $L - DG^+$ algorithm (appropriately initialized in order to replicate the transition-node-pivoting rule) is such that none of the last n elements of the columns of $-(B_k)^{-1}(N_k)_{\cdot\hat{y}_k}$ are 0.

As in Springuel, Gendron and Soriano (2004), four basic functions which make easier the statement of the extended $L - DG^+$ algorithm are determined :

1- *pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ performs the pivot on element $(B_k)^{-1}_{\tilde{y}_k\cdot}(N_k)_{\cdot\bar{y}_k}$ where $\bar{y}_k \in y_{N_k}$ and $\tilde{y}_k \in y_{B_k}^{\min}$ are given and such that $\bar{y}_k$ replaces $\tilde{y}_k$;

2- *select_pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ returns first $\bar{y}_k \in y_{N_k}$ with minimal index and a positive reduced cost (assuming that $y_{B_k} \neq y_{B_0}$) and second $\tilde{y}_k \in y_{B_k}^{\min}$ determined according to the arbitrary lexicographic basic variable selection rule given $\bar{y}_k$;

3- *reverse*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ determines if it is *True* for a given $\bar{y}_k \in y_{N_k}$ and $\tilde{y}_k \in y_{B_k}^{\min}$ that *select_pivot* applied to the basis $y_{B_{k'}}$ having basic variables $y_{B_k} + \bar{y}_k - \tilde{y}_k$ returns the element $\tilde{y}_k \in y_{N_{k'}}$ and $\bar{y}_k \in y_{B_{k'}}^{\min}$ (note that $\tilde{y}_k \in y_{B_k}^{\min}$ is meaningful only when *reverse*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ is *True*);

4- for y_{B_k} representing an extreme ray of K in the first n elements of the column $-(B_k)^{-1}(N_k)_{\cdot\hat{y}_k}$ where $\hat{y}_k \in y_{N_k}$, *verify_output_file*$(y_{B_k}, \hat{y}_k)$ determines if it is *True* for y_{B_k} that for each $\tilde{y}_k \in y_{B_k}^{\min}$ such that $(B_k)^{-1}_{\tilde{y}_k\cdot}(N_k)_{\cdot\hat{y}_k} = 0$, $\nexists\, \bar{y}_k \in y_{N_k}$ with its index of s smaller than the index of s of $\tilde{y}_k$ and $(B_k)^{-1}_{\tilde{y}_k\cdot}(N_k)_{\cdot\bar{y}_k} > 0$.

EXTENDED $L - DG^+$ ALGORITHM

Initialization :

Translate v to the origin and remove the redundant constraints defining K.

Let B_0 be given and let $\overline{B} = -B_0$.

While $L(B_0) = \left(0\ (B_0)^{-1}\overline{B}\right)$ is not a lex-positive matrix, do :

Find the lexicographically negative $\tilde{y}_0 \in y_{B_0}^{\min}$ with minimal index.

Find $\bar{y}_0 \in y_{N_0}$ with minimal index and $(B_0)^{-1}_{\tilde{y}_0 \bullet}(N_0)_{\bullet \bar{y}_0} < 0$.

Perform *pivot*$(y_{B_0}, \bar{y}_0, \tilde{y}_0)$ and let y_{B_0} be the resulting basis.

Permute the rows of H such that the last n columns of $-I$ are forming N_0.

According to this permutation, change the rows of B_0, $\overline{B}$ and $L(B_0)$.

Let the first p elements of c be 0 and the remaining n elements be -1 and $i = n + 1$.

While $i \leq n + p$ or $y_{B_k} \neq y_{B_0}$, do :

While $i \leq n + p$, do :

Let $\bar{y}_k$ be such that its index of s is i.

If $\bar{y}_k \in y_{N_k}$, then :

If $-(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \bar{y}_k} \geq 0 \; \forall \; \tilde{y}_k \in y_{B_k}^{\min}$, then :

If *verify_output_file*$(y_{B_k}, \bar{y}_k) = True$, then :

If this ray is not in the output file, then write it to the output file.

Else :

If *reverse*$(y_{B_k}, \bar{y}_k, \tilde{y}_k) = True$, then :

Perform *pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$.

Let $i = n + 1$.

Else $i = i + 1$.

Else $i = i + 1$.

Perform *select_pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$.

Let i be the index of s of $\bar{y}_k$ and $i = i + 1$.

Perform *pivot*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$.

The most critical function of the extended $L - DG^+$ algorithm is *reverse*$(y_{B_k}, \bar{y}_k, \tilde{y}_k)$ which obviously returns *True* only if $c_{\bar{y}_k} - c^T_{B_k}(B_k)^{-1}(N_k)_{\bullet \bar{y}_k} < 0$ and for each $\hat{y}_k \in y_{N_k}$ with its index of s smaller than the index of s of $\bar{y}_k$, its resulting reduced cost element

$$\left(c^T_{\hat{y}_k} - c^T_{B_k}(B_k)^{-1}(N_k)_{\bullet \hat{y}_k}\right) - \frac{\left(c_{\bar{y}_k} - c^T_{B_k}(B_k)^{-1}(N_k)_{\bullet \bar{y}_k}\right)}{(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \bar{y}_k}} (B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \hat{y}_k}$$

is negative if pivoting on $(B_k)^{-1}_{\tilde{y}_k \bullet}(N_k)_{\bullet \bar{y}_k}$.

The number of y_{B_k} generated by the extended $L - DG^+$ algorithm (appropriately initialized in order to replicate the transition-node-pivoting rule) is bounded from above by the number of extreme rays of K if all the extreme rays of K are non-degenerate as proven by Geue (1993). However, there exist bounded polyhedron families for which the number of y_{B_k} generated by the extended $L - DG^+$ algorithm is not polynomially bounded from above by the number of extreme rays of K, according to Armand (1993).

4.2.2 *Double Description Algorithm*

Based on the dual concept of Fourier (1826) variable elimination procedure (refer to Dantzig and Eaves (1973) for an explanation concerning the dual relationship), Motzkin, Raiffa, Thompson and Thrall (1953) introduce the double description algorithm by characterizing a homogeneous system of linear inequalities by the matrix of its generating rays.

After having translated v to the origin for convenience purposes, there exists at least one subset of n linearly independent constraints among the p constraints defining K since D is full dimensional. If the lines of the submatrix $\overline{H}$ of H corresponds to the normal vectors of each one of these n constraints, then the inverse of $\overline{H}$ determines the starting generating rays matrix of the initial polyhedral cone $\overline{K}_0$ pointed on the origin outer approximating K since

$$\overline{H}x = \lambda \geq 0 \Leftrightarrow x = (\overline{H})^{-1}\lambda \text{ and } \lambda \geq 0.$$

Without loss of generality, suppose that the first n constraints are chosen to form $\overline{H}$ and define

$$(NR_0)^T = (\underbrace{1,\ldots,1}_{n \text{ times}},\underbrace{0,\ldots,0}_{p-n \text{ times}})$$

as the indicator vector of non-redundant constraints for the description of $\overline{K}_0$. Rewriting these p constraints as a function of the generating rays gives the following matrix

$$\overline{F}_0 = \begin{pmatrix} h_1^T\overline{E}_0 \\ \vdots \\ h_{n+1}^T\overline{E}_0 \\ \vdots \\ h_p^T\overline{E}_0 \end{pmatrix} = \begin{pmatrix} h_1^T\overline{r}_{0,1} & h_1^T\overline{r}_{0,2} & \cdots & h_1^T\overline{r}_{0,n} \\ \vdots & \vdots & \vdots & \vdots \\ h_{n+1}^T\overline{r}_{0,1} & h_{n+1}^T\overline{r}_{0,2} & \cdots & h_{n+1}^T\overline{r}_{0,n} \\ \vdots & \vdots & \vdots & \vdots \\ h_p^T\overline{r}_{0,1} & h_p^T\overline{r}_{0,2} & \cdots & h_p^T\overline{r}_{0,n} \end{pmatrix}$$

where $\overline{E}_0$ corresponds to the inverse of $\overline{H}$, h_i is the normal vector of the i^{th} contraints of H and $\overline{r}_{0,j}$ is the vector representing the generating ray found at the j^{th} column of $\overline{E}_0$.

The intersection of $\overline{K}_0$ and a half-space generated by a hyperplane passing through the origin belongs to one of the three following cases : $\overline{K}_0$ if and only if it is contained entirely in the half-space (case I); a smaller polyhedral cone $\overline{K}_1$ pointed on the origin if and only if $\overline{K}_0$ is not contained entirely in the half-space (case II); the origin if and only if the half-space contains only this point in common with $\overline{K}_0$ (case III). From the signs of the coefficients of the $(n+1)^{th}$ line of $\overline{F}_0$, it is possible to characterize each extreme ray of $\overline{E}_0$ relative to $h_{n+1}^T x \geq 0$. If $h_{n+1}^T\overline{r}_{0,j}$ is positive, then $\overline{r}_{0,j}$ is stricly feasible relative to $h_{n+1}^T x \geq 0$. When $h_{n+1}^T\overline{r}_{0,j}$ is zero, then it implies that $\overline{r}_{0,j}$ is feasible and lies entirely on $h_{n+1}^T x = 0$. Finally, if $h_{n+1}^T\overline{r}_{0,j}$ is negative, then it means that $\overline{r}_{0,j}$ is strictly infeasible relative to $h_{n+1}^T x \geq 0$. By defining the following sets

$$N_{0,m} = \left\{j \in \{1,2,\ldots,n\} \mid h_{n+1}^T\overline{r}_{0,j} < 0\right\},$$
$$N_{0,p} = \left\{j \in \{1,2,\ldots,n\} \mid h_{n+1}^T\overline{r}_{0,j} > 0\right\},$$
$$N_{0,z} = \left\{j \in \{1,2,\ldots,n\} \mid h_{n+1}^T\overline{r}_{0,j} = 0\right\},$$

it is clear that case I is possible only when $N_{0,p} \cup N_{0,z} = \{1,2,\ldots,n\}$ whereas case II is possible only when $N_{0,m} \neq \emptyset$ and $N_{0,p} \neq \emptyset$ while case III occurs only when $N_{0,m} = \{1,2,\ldots,n\}$. It is important to point out that case III and the case $N_{0,m} \cup N_{0,z} = \{1,2,\ldots,n\}$ are not possible since D is full dimensional.

More specifically, if case I occurs, then $h_{n+1}^T x \geq 0$ is redundant for the description of $\overline{K}_1$, i.e. $NR_1 = NR_0$ and all the extreme rays of $\overline{K}_0$ remain feasible. If case II occurs, then $h_{n+1}^T x \geq 0$ is non-redundant for the description of $\overline{K}_1$, i.e. $NR_1 = NR_0$ and $NR_1(n+1) = 1$, but only the $\bar{r}_{0,j}$ such that $j \in N_{0,p} \cup N_{0,z}$ remain feasible while the $\bar{r}_{0,j}$ such that $j \in N_{0,m}$ are not feasible anymore. However, new extreme rays of $\overline{K}_1$ are generated from the intersection of $h_{n+1}^T x = 0$ and a two dimensional face of $\overline{K}_0$ containing a couple of adjacent extreme rays $(\bar{r}_{0,j}, \bar{r}_{0,j'})$ where $j \in N_{0,p}$ and $j' \in N_{0,m}$ if $\nexists\, \bar{r}_{0,j''}$ such that $Z(\bar{r}_{0,j}) \cap Z(\bar{r}_{0,j'}) \cap NR_0 \subseteq Z(\bar{r}_{0,j''}) \cap NR_0$ where $Z(\bar{r})$ corresponds to the set of constraints of $\overline{K}_0$ for a given $\bar{r}$. In addition, it is possible that some constraints non-redundant for the description of $\overline{K}_0$ are no longer necessary to the description of $\overline{K}_1$ and these constraints are identified by the necessary and sufficient condition introduced by Greenberg (1975) : if there exists one extreme ray lying entirely on $h_{i'}^T x = 0$ which is stricly feasible relative to $h_{n+1}^T x \geq 0$ where i' is such that $NR_0(i') = 1$, then the constraint remains non-redundant for the description of $\overline{K}_1$. In other words, $NR_1(i') = 1$ if and only if there exists a $j \in N_{0,p}$ such that $h_{i'}^T \bar{r}_{0,j} = 0$. In fact, when all the extreme rays lying entirely on $h_{i'}^T x = 0$ are stricly infeasible relative to $h_{n+1}^T x \geq 0$, then $h_{i'}^T x \geq 0$ possesses only the origin in common with $h_{n+1}^T x \geq 0$. When only some extreme rays lying entirely on $h_{i'}^T x = 0$ are stricly infeasible relative to $h_{n+1}^T x \geq 0$ and all the others are lying on $h_{n+1}^T x = 0$, then $h_{i'}^T x \geq 0$ possesses only a face of a smaller dimension in common with $h_{n+1}^T x \geq 0$ where this face is generated by all the extreme rays lying entirely on $h_{i'}^T x = 0$ and on $h_{n+1}^T x = 0$.

Fukuda and Prodon (1996) introduce the notion of inherited adjacency and a stronger necessary condition to the adjacency of two extreme rays comparatively to the necessary condition of Greenberg (1975) which states that if $Card(Z(\bar{r}_{0,j}) \cap Z(\bar{r}_{0,j'}) \cap NR_0) \geq n-2$, then the two extreme rays might be adjacent. Specifically for the k^{th} iteration of the double description algorithm ($k \geq 0$), two extreme rays are adjacent by inheritance if and only if one of the two rays is the father of the other where Fukuda and Prodon (1996) defined $B(\bar{r}_{k,j})$ and $F(\bar{r}_{k,j})$ to be respectively the iteration of birth of $\bar{r}_{k,j}$ and the father of $\bar{r}_{k,j}$ defined as the feasible extreme ray of the couple of adjacent extreme rays from which $\bar{r}_{k,j}$ is generated. These authors also indicate that a necessary condition for two extreme rays $\bar{r}_{k,j}$ and $\bar{r}_{k,j'}$ that are non-adjacent by inheritance at the beginning of iteration $k' = \max\{B(\bar{r}_{k,j}), B(\bar{r}_{k,j'})\}$, $j \in N_{k,p}$ and $j' \in N_{k,m}$, to become adjacent at the beginning of the k^{th} iteration ($k > k'$) is that there exists at least one hyperplane introduced at the l^{th} iteration ($k' \leq l < k$) non-redundant for the description of the polyhedral cone pointed at the origin at the beginning of the k^{th} iteration which contains $\bar{r}_{k,j}$ and $\bar{r}_{k,j'}$ or, in other words, it requires that $Z(\bar{r}_{k,j}) \cap Z(\bar{r}_{k,j'}) \cap (NR_k - NR_{k'}) \neq \emptyset$. It is important

to point out that when $k' = 0$, any two extreme rays are adjacent by construction.

DOUBLE DESCRIPTION ALGORITHM :

Initialization :

Translate v to the origin.

Form $\overline{H}$ with n linearly independent constraints among the p constraints of K.

Let $\overline{E}_0$ be the inverse of $\overline{H}$ and construct the matrix $\overline{F}_0$.

Let $B(\overline{r}_{0,j}) = 0$ and $F(\overline{r}_{0,j}) = 0 \; \forall \; j = 1,2,\ldots,n$.

Let NR_0 be defined as previously, $q = n$ and $q' = 0$.

Iteration $k = 0,1,\ldots,p-n-1$:

Select among the $p-n-k$ last lines of $\overline{F}_k$ the line according to an insertion rule.

Interchange this line with $h_{n+k+1}^T \overline{E}_k$.

Determine the set $N_{k,m}$, $N_{k,p}$, $N_{k,z}$ of $h_{n+k+1}^T \overline{E}_k$.

If $Card(N_{k,m}) = 0$, then $k = k+1$.

Else :

$NR_{k+1} = NR_k$ and $NR_{k+1}(n+k+1) = 1$.

For $i' \leq n+k$, do :

If $NR_k(i') = 1$ and $\nexists \; j \in N_{k,p}$ such that $h_{i'}^T \overline{r}_{k,j} = 0$, then $NR_{k+1}(i') = 0$.

Let $\overline{E}_{k+1} = \left\{ \overline{r}_{k,j} \mid j \in N_{k,p} \cup N_{k,z} \right\}$ and $q' = Card(N_{k,p}) + Card(N_{k,z})$.

For each pair $\overline{r}_{k,j}, \overline{r}_{k,j'}$ where $j \in N_{k,p}, j' \in N_{k,m}$, do :

Let $k' = \max\{B(\overline{r}_{k,j}), B(\overline{r}_{k,j'})\}$.

If $k' = 0$ or $F(\overline{r}_{k,j}) = \overline{r}_{k,j'}$ or $F(\overline{r}_{k,j'}) = \overline{r}_{k,j}$, then :

Add $\overline{r}_{(k+1),jj'} = (h_{n+k+1}^T \overline{r}_{k,j})\overline{r}_{k,j'} - (h_{n+k+1}^T \overline{r}_{k,j'})\overline{r}_{k,j}$ to $\overline{E}_{k+1}$.

Let $q' = q' + 1$, $B(\overline{r}_{(k+1),jj'}) = k+1$ and $F(\overline{r}_{(k+1),jj'}) = \overline{r}_{k,j}$.

If $k' > 0$ and $F(\overline{r}_{k,j}) \neq \overline{r}_{k,j'}$ and $F(\overline{r}_{k,j'}) \neq \overline{r}_{k,j}$, then :

If $Z(\overline{r}_{k,j}) \cap Z(\overline{r}_{k,j'}) \cap (NR_k - NR_{k'}) \neq \emptyset$, then :

If $Card(Z(\overline{r}_{k,j}) \cap Z(\overline{r}_{k,j'}) \cap NR_k) \geq n-2$, then :

If $\nexists \; \overline{r}_{k,j''}$ such that $Z(\overline{r}_{k,j}) \cap Z(\overline{r}_{k,j'}) \cap NR_k \subseteq Z(\overline{r}_{k,j''}) \cap NR_k$, then :

Add $\overline{r}_{(k+1),jj'} = (h_{n+k+1}^T \overline{r}_{k,j})\overline{r}_{k,j'} - (h_{n+k+1}^T \overline{r}_{k,j'})\overline{r}_{k,j}$ to $\overline{E}_{k+1}$.

Let $q' = q' + 1$, $B(\overline{r}_{(k+1),jj'}) = k+1$ and $F(\overline{r}_{(k+1),jj'}) = \overline{r}_{k,j}$.

Rewrite $\overline{F}_{k+1}$ as a function of $\overline{E}_{k+1}$, let $k = k+1$, $q = q'$ and $q' = 0$.

Several static and dynamic insertion rules are proposed in the literature. The usual static insertion rules of the constraints are either the min-index which selects the constraint not yet processed having the smallest index, or the max-index which selects the constraint not yet processed having the largest index. The usual dynamic insertion rules of the constraints are either the min-column which selects the constraint not yet processed that generates the potentially smallest number of extreme rays for $\overline{K}_k$ at the end of the k^{th} iteration, the min-cutoff

which selects the constraint not yet processed that cuts the smallest number of extreme rays of $\overline{K}_k$ or the max-cutoff which selects the constraint not yet processed that cuts the largest number of extreme rays of $\overline{K}_k$.

Avis, Bremner and Seidel (1997) present some families of bounded polyhedra for each insertion rule of the constraints such that the application of the double description algorithm requires a computing time not polynomially bounded from above. Consequently, if the constraints defining a degenerate extreme point v being translated to the origin correspond to the homogenization of a polyhedron (refer to Subsection 4.3.2 for the definition of homogenization of a polyhedron) belonging to one of these pathological families of bounded polyhedra, then the double description algorithm might require a computing time not polynomially bounded from above.

4.3 *Major BCP Algorithms Affected by Degeneracy*

4.3.1 *Cutting Hyperplane Algorithm*

Based on Tuy (1964), the cutting hyperplane algorithm consists in demonstrating $D \backslash G_\gamma = D \backslash \{x \in \Re^n \mid f(x) \geq \gamma\} = \emptyset$ where $f(v) = \gamma$ and $\gamma' = \gamma - \varepsilon$ $(\varepsilon \geq 0)$, v being a candidate as a global minimum. If v is a non-degenerate or a weak degenerate extreme point of D (i.e. degeneracy is only due to the presence of redundant constraints defining v), then the smallest polyhedral cone K pointed on v containing D has only n extreme rays denoted r_j which are easily determined by one of the previous technique. On each extreme ray of K, there exists a point $z_j \in bd(G_{\gamma'}) = \{x \in \Re^n \mid f(x) = \gamma'\}$ such that $z_j = v + \alpha_j r_j$ where $\alpha_j = \sup\{\theta \in \Re_+ \mid f(v + \theta r_j) = \gamma'\}$. It is important to point out that it may be possible to have $v = z_j$ if $\varepsilon = 0$ for any given j, but suppose for the moment that it is not the case. Since the extreme rays of K are linearly independent, then there is only one hyperplane h_z containing all the z_j which can be written as $h_z^T(x - v) = 1$ where the j^{th} component of h_z is equal to $1/\alpha_j$. The simplex S determined by the convex envelope of all the z_j and v is such that it corresponds to the intersection of K and the half-space containing v generated by h_z. But since all the z_j and v are elements of $G_{\gamma'}$, then $S \subset G_{\gamma'}$ which implies that h_z is a linear constraint that cuts v from the set $D \cap G_{\gamma'}$; h_z is called the γ'-concavity cut relative to v. It has to be pointed out that if one $\alpha_j \to \infty$, then the corresponding r_j is a recession direction of $G_{\gamma'}$, the j^{th} component of h_z is zero which means that the γ'-concavity cut is parallel to this direction and S is a generalized simplex while if all $\alpha_j \to \infty$, then $S = K \subset G_{\gamma'}$ and therefore $D \subset G_{\gamma'}$ because $D \subset K$ which implies that all the solutions of D are such that their value of f is greater than γ', i.e. v is an ε-global minimum of f over D (i.e. $f(v) - \varepsilon \leq f(x)$ $\forall$ $x \in D$). By defining v_z^* and μ_z to be respectively one optimal solution and the optimal value of

$$\max_{x \in D} \; h_z^T(x - v),$$

if $\mu_z \leq 1$, then the cutting hyperplane algorithm is stopped since v is a global minimum while if $1 < \mu_z < \infty$, then v_z^* belongs either to $D \backslash G_{\gamma'}$ or to $G_{\gamma'} \backslash S$. Even if the first case is preferable since a new candidate as being a global minimum is determined, the second case is beneficial since a γ'-concavity cut relative to v reduces $G_{\gamma'} \cap D$. If $\mu_z = \infty$, then an extreme ray of D is identified and if it does not belong to the recession cone of $G_{\gamma'}$ denoted $rec(G_{\gamma'})$, then it crosses $bd(G_{\gamma'})$ which implies that $f \to -\infty$ in this direction and f is therefore unbounded from below on D.

If v is a strong degenerate extreme point of D (i.e. degeneracy is not only due to the presence of redundant constraints defining v), then K possesses p extreme rays where $p > n$ and a direct application of the previous construction generates p points z_j. It is important to point out that S is not a simplex anymore (or a generalized simplex) and it may be impossible to find a h_z containing all the z_j. To solve this problem, the approach of Carvajal-Moreno (1972) consists in finding a h_z of a γ'-concavity cut relative to v by determining a basic solution to

$$h_z^T r_j \geq 1/\alpha_j \text{ for } j = 1, 2..., p.$$

For the $p - n$ extreme rays of K not constituting a basic solution, the points $\bar{z}_j = \{x \in \{\theta_j \in \Re_+ \mid v + \theta_j r_j\} \mid h_z^T(x - v) = 1\}$ corresponding to the intersection of these extreme rays and h_z are such that $0 < \theta_j < \alpha_j$, in order for the γ'-concavity cut relative to v to remain valid. Since many are available, it is preferable to generate the γ'-concavity cut relative to v that covers the largest portion of D, i.e. one that minimizes the sum of the difference between α_j and θ_j which is given by an optimal solution of

$$\min \sum_{j=1,2,\ldots,p} (h_z^T r_j - 1/\alpha_j)$$

$$\text{s.t. } h_z^T r_j \geq 1/\alpha_j \text{ for } j = 1, 2..., p.$$

In such cases, the γ'-concavity cut relative to v is said to be of maximum depth where depth is measured by the Euclidean distance between v and h_z, i.e. by $1/(\|h_z\|^2)$.

CUTTING HYPERPLANE ALGORITHM

Choose $\varepsilon \geq 0$.

Initialization :

Determine a local minimum v of f over D.

Let $\gamma = f(v)$, $\gamma' = \gamma - \varepsilon$ and let $v_{f,\varepsilon}^* = v$.

Iteration $k = 0, 1, \ldots$:

Determine the smallest polyhedral cone K pointed on v that contains D.

Construct the γ'-concavity cut relative to v where its normal vector is h_z.

Solve $\max_{x \in D} h_z^T(x - v)$ of which the optimal value is μ_z.

If $\mu_z \leq 1$, then STOP because v is an ε-global minimum of f over D.
Else :
  If $\mu_z < \infty$, then let v_z^* be an optimal solution of this problem.
  Else :
    Verify if the identified extreme ray belongs to $rec(G_{\gamma'})$.
    If no, then STOP because f is unbounded from below on D.
    If yes, then let v_z^* be the extreme point from which emanates this extreme ray.
  Let $D = D \cap \{x \in \Re^n \mid h_z^T(x - v) \geq 1\}$.
  From v_z^*, find a new local minimum v of f over D.
  If $f(v) < \gamma'$, then let $\gamma = f(v)$, $\gamma' = \gamma - \varepsilon$, $v_{f,\varepsilon}^* = v$ and set $k = 0$.
  Otherwise, let $k = k + 1$.

It is important to point out that this algorithm always requires determining the γ'-concavity cut relative to v which are local minima of f over D. A first reason for this is that an extreme point which is not a local minimum cannot be a global minimum. A second and more important reason is that it is required to have $v \neq z_j \; \forall \; j$ to ensure that the depth of each γ'-concavity cut relative to v is non-null. When $\varepsilon = 0$, it is possible that $v \in bd(G_\gamma)$, but $v \neq z_j \; \forall \; j$ since v is a local minimum of f over D for which all its neighboring extreme points and all its emanating extreme rays are contained in G_γ while if $\varepsilon > 0$, then $v \in int(G_{\gamma'})$ by construction which implies that $v \neq z_j \; \forall \; j$ (it is not necessary to determine the γ'-concavity cut relatively to local minima in this case).

When the depth of each γ'-concavity cut relative to v is non-negligible, then the cutting hyperplane algorithm is finite because a finite number of γ'-concavity cut relative to v are necessary to cover entirely D and it finds an ε-global minimum of f over D. More specifically, the algorithm is composed of a finite number of cycles of iterations since D has only a finite number of extreme points not satisfying to equality any of the previous γ'-concavity cut relative to v which start a new cycle. In addition, each of these cycles is finite by construction since the depth of each γ'-concavity cut relative to v is non-negligible, but it is hard in pratice to ensure such a property which implies that the cutting hyperplane algorithm may be infinite in certain cases.

4.3.2 *Outer Approximation Algorithm*

Based on Hoffman (1981) and Thieu, Tam and Ban (1983), the outer approximation algorithm consists first in outer approximating D by a polyhedron $\overline{D}$ which possesses only few extreme points and few extreme rays in order to efficiently find a global minimum $\overline{v}_f^*$ of f over $\overline{D}$. After the application of the cutting hyperplane algorithm until the γ-concavity cut relative to v is constructed and the corresponding linear program is solved, $\overline{D}$ is defined by the intersection of

K and the half-space containing v generated by the hyperplane $h_z^T(x - v) = \mu_z$ when $\mu_z < \infty$ or else, $\overline{D}$ is simply defined by K when $\mu_z = \infty$. Once $\overline{D}$ is determined, if $f \to -\infty$ on an extreme ray $\bar{r}$ of $\overline{D}$ and $\bar{r} \in rec(D)$, then f is unbounded from below on D since there exists at least one extreme ray of D on which $f \to -\infty$. However, if $\bar{r} \notin rec(D)$, then $\bar{r}$ violates one of the constraints defining $rec(D)$ expressed by $a_i^T x \geq 0$ $(i = 1, ..., m)$. By choosing

$$i' \in \arg \min_{i=1,...,m} a_i^T \bar{r},$$

the set $\overline{D}' = \overline{D} \cap \{x \in \Re^n \mid a_{i'}^T x \geq b_{i'}\}$ is formed such that $\overline{D}'$ does not possess $\bar{r}$ anymore since $a_{i'}^T \bar{r} < 0$. If each of the extreme rays of $\overline{D}$ is such that f is bounded from below on it, then $\bar{v}_f^*$ is determined by taking the minimal value of f on the set of extreme points of $\overline{D}$. If $\bar{v}_f^* \in D$ then stop, but since it is not generally the case, there is at least one of the m constraints of D which is violated. By choosing

$$i' \in \arg \min_{i=1,...,m} a_i^T \bar{v}_f^* - b_i,$$

the set $\overline{D}' = \overline{D} \cap \{x \in \Re^n \mid a_{i'}^T x \geq b_{i'}\}$ is formed.

Once the affine cut is added to the description of $\overline{D}$ to form $\overline{D}'$, it is essential to identify all the new extreme points and new extreme rays of $\overline{D}'$ comparatively to $\overline{D}$ in order to minimize efficiently f over $\overline{D}'$. Consequently, using any non-constructive enumeration algorithms such as the extended $L - DG^+$ algorithm of Springuel, Gendron and Soriano (2005) is inadequate since it is generating at each iteration of the outer approximation algorithm all the extreme points and extreme rays of $\overline{D}$ from one starting basis without using the information relative to the extreme points and extreme rays already available. Based on this information, the double description algorithm of Subsection 4.2.2 can be applied directly on the homogenization D_{Hom} of D which corresponds to

$$Ax - bx_{n+1} \geq 0 \text{ and } (x, x_{n+1}) \geq 0$$

such that for any extreme ray $(x, x_{n+1}) \in \Re^{n+1}$ of D_{Hom}, if $x_{n+1} = 0$, then x corresponds to an extreme ray of D while if $x_{n+1} > 0$, then $\frac{1}{x_{n+1}} x$ is an extreme point of D. Horst, Thoai and de Vries (1988) introduce another constructive enumeration algorithm which first characterizes each extreme point of $\overline{D}$ with respect to the half space generated by $a_{i'}^T x = b_{i'}$ through

$$N_m = \{j \in \{1, 2, \ldots\} \mid a_{i'}^T \bar{x}_j < b_{i'}\},$$
$$N_p = \{j \in \{1, 2, \ldots\} \mid a_{i'}^T \bar{x}_j > b_{i'}\},$$

where $\bar{x}_j$ denotes the j^{th} extreme point of $\overline{D}$ and in order to determine which of these two sets of indices possesses the least number of elements. Assuming that $Card(N_p) < Card(N_m)$, for each index j of N_p, the algorithm then requires to determine all the extreme rays of the polyhedral cone defining $\bar{x}_j$ and all the neighboring extreme points of $\bar{x}_j$ for $\overline{D}$ such that for each one, if it is infeasible with respect to $a_{i'}^T x \geq b_{i'}$, then a new extreme point of $\overline{D}'$ is determined by the intersection of the hyperplane $a_{i'}^T x = b_{i'}$ with the edge connecting $\bar{x}_j$ to the neighboring extreme point in question. For each extreme ray $\bar{r}_j$ of $\overline{D}$ emanating from $\bar{x}_j$, if

there exists $0 < \alpha < \infty$ such that $a_{i'}^T(\bar{x}_j + \alpha\bar{r}_j) < b_{i'}$, then a new extreme point of $\overline{D}'$ is determined by the intersection of the hyperplane $a_{i'}^T x = b_{i'}$ with $\bar{x}_j + \alpha\bar{r}_j$. Concerning the case of new extreme rays generated by the introduction of the half-space, they are determined once $\{x \in \Re^n \mid a_{i'}^T x \geq 0\}$ is added to $rec(\overline{D})$.

OUTER APPROXIMATION ALGORITHM

Initialization :

Determine a local minimum v of f over D.

Let $\gamma = f(v)$, I be the set of constraints defining v, $I_0 = \{1, ..., m\} \backslash I$.

Determine the smallest polyhedral cone K pointed on v that contains D.

Construct the γ-concavity cut relative to v where its normal vector is h_z.

Solve $\max_{x \in D} h_z^T(x - v)$ of which the optimal value is μ_z.

If $\mu_z < \infty$, then let $\overline{D} = K \cap \{x \in \Re^n \mid h_z^T(x - v) \leq \mu_z\}$.

Else let $\overline{D} = K$.

Let $\overline{EP}$ and $\overline{ER}$ be respectively the set of extreme points and extreme rays of $\overline{D}$.

Iteration $k = 0, 1, \ldots$:

If for one $\bar{r} \in \overline{ER}$ there exists a $\alpha > 0$ such that $f(v + \alpha\bar{r}) < \gamma$, then :

If $a_i^T\bar{r} \geq 0$ for all $i \in I_k$, then STOP since f is unbounded from below on D.

If $a_i^T\bar{r} < 0$ for one $i \in I_k$, then determine $i' \in \arg \min_{i \in I_k} a_i^T\bar{r}$.

Else :

Determine $\bar{v}_f^* \in \arg \min_{x \in \overline{EP}} f(x)$.

If $\bar{v}_f^* \in D$, then STOP because $\bar{v}_f^*$ is a global minimum of f over D.

If $\bar{v}_f^* \notin D$, then determine $i' \in \arg \min_{i \in I_k} a_i^T\bar{v}_f^* - b_i$.

Let $\overline{D} = \overline{D} \cap \{x \in \Re^n \mid a_{i'}^T x \geq b_{i'}\}$.

Determine $\overline{EP}$ and $\overline{ER}$ of $\overline{D}$.

Let $I_{k+1} = I_k \backslash \{i(k)\}$ and $k = k + 1$.

The outer approximation algorithm is finite by construction since there is only a finite number of elements belonging to I_0 from which one element is deleted at each iteration of the algorithm.

4.3.3 *Polyhedral Annexation Algorithm*

Based on Vaish and Shetty (1976) and Tuy (1990), the polyhedral annexation algorithm consists in demonstrating $D \backslash G_\gamma = D \backslash \{x \in \Re^n \mid f(x) \geq \gamma\} = \emptyset$ where $f(v) = \gamma$ and $\gamma' = \gamma - \varepsilon$ ($\varepsilon \geq 0$), v being a local minimum candidate as a global minimum. Translating v to the origin

for convenience purposes, determining the γ-concavity cut relative to the origin and defining as previously v_z^* and μ_z to be respectively one optimal solution and the optimal value of

$$\max_{x \in D} \; h_z^T x,$$

if $1 < \mu_z < \infty$, then v_z^* is an extreme point of D either belonging to $D \backslash G_\gamma$ or to $G_\gamma \backslash S$. The polyhedral annexation algorithm prescribes for the second case to cut v_z^* by forming $P = conv\{D \cap \{x \in \Re^n | h_z^T x \leq 1\}, \{\hat{v}_z^*\}\}$ where $\hat{v}_z^*$ is either a point of $bd(G(\gamma))$ on the ray emanating from the origin passing through v_z^* or a direction of $rec(G_\gamma)$. If $\mu_z = \infty$, then an extreme ray r of D is identified and if it belongs to $rec(G_\gamma)$, then $P = conv\{D \cap \{x \in \Re^n | h_z^T x \leq 1\}, \{r\}\}$ is formed to cut this direction. For each new hyperplane h_P of the representation of P not containing the origin, the polyhedral annexation algorithm then requires to solve

$$\max_{x \in D} \; h_P^T x$$

of which an optimal solution is v_P^* and the optimal value is μ_P. If $\mu_P \leq 1$ for each new hyperplane h_P of the representation of P not containing the origin, then the polyhedral annexation algorithm is stopped since v is a global minimum, but since it is not generally the case, v_P^* or r identifed by the hyperplane h_P of the representation of P not containing the origin having the highest μ_P is cut as previously.

The manner by which P is constructed requires the introduction of concepts from polyhedral geometry. Namely, an hyperplane of the representation of P is said to be transversal if it does not possess the origin. When P is a full dimensional polyhedron such that $0 \in P$, each transversal hyperplane of the representation of P corresponds to an extreme point of $pol(P) = \{x \in \Re^n \mid w^T x \leq 1 \;\; \forall w \in P\}$ while each non-transversal hyperplane of the representation of P corresponds to an extreme ray of $pol(P)$ and vice-versa. Therefore, since K is a full dimensional polyhedral cone pointed on the origin for which all the transversal hyperplanes of its representation are known, then $pol(K)$ is also a full dimensional polyhedral cone pointed on the origin, but for which all its extreme rays are known. If $\hat{v}_z^*$ is a point added to form P, then $pol(P)$ is now a full dimensional polyhedral cone pointed on h_z having its extreme rays corresponding to the extreme rays of $pol(K)$. For each following iteration of the polyhedral annexation algorithm, if $1 < \mu_P < \infty$, $f(\hat{v}_P^*) \geq \gamma$ and $\hat{v}_P^*$ is a point, then a transversal hyperplane is added to the representation of $pol(P)$, i.e. $pol(P') = pol(P) \cap \{x \in \Re^n \mid (\hat{v}_P^*)^T x \leq 1\}$, thus creating only new extreme points which correspond to new transversal hyperplane of the representation of P'. If $1 < \mu_P < \infty, f(\hat{v}_P^*) \geq \gamma$ and $\hat{v}_P^*$ is a direction, then one extreme ray belonging to $rec(G_\gamma)$ is identified and a non-transversal hyperplane is added to the representation of $pol(P)$, i.e. $pol(P') = pol(P) \cap \{x \in \Re^n \mid (\hat{v}_P^*)^T x \leq 0\}$, generating only new extreme points which correspond to new transversal hyperplanes of the representation of P'. Finally, if $\mu_P = \infty$ and r belongs to $rec(G_\gamma)$, then a non-transversal hyperplane is added to the representation of $pol(P)$,

i.e. $pol(P') = pol(P) \cap \{x \in \Re^n \mid r^T x \leq 0\}$, generating only new extreme points which correspond to new transversal hyperplanes of the representation of P'. In all these cases, all new extreme points are determined by an iteration of any constructive enumeration algorithm presented above.

POLYHEDRAL ANNEXATION ALGORITHM

Initialization :

Determine a local minimum v of f over D.

Let $\gamma = f(v)$ and translate v to the origin.

Determine the smallest polyhedral cone K pointed on the origin containing D.

Construct the γ-concavity cut relative to the origin where its normal vector is h_z.

Let $\overline{EP} = \{h_P\} = \{h_z\}$, $\overline{EP}^* = \overline{EP}$, $P = D \cap \{x \in \Re^n | h_z^T x \leq 1\}$ and $k = 0$.

($\overline{EP}^*$ is the set of new extreme points of $pol(P)$).

Iteration $k = 0, 1, \ldots$:

For each $h_P \in \overline{EP}^*$:

Solve $\max_{x \in D} h_P^T x$ of which the optimal value is μ_P.

If $\mu_P < \infty$, then :

Let v_P^* be the optimal solution of this problem.

If $f(v_P^*) < \gamma$, then $D = D \cap \{x \in \Re^n \mid h_z^T x \geq 1\}$ and go to initialization phase.

Else :

Verify if the identified extreme ray of D belongs to $rec(G_\gamma)$.

If no, then STOP because f is unbounded from below on D.

If yes, then let r be the direction of this extreme ray of D.

Determine $h_P \in \arg \max_{v \in \overline{EP}} \mu_P$.

If $\mu_P \leq 1$, then STOP since v is a global minimum of f over D.

Else :

If $\mu_P < \infty$, then :

Determine $\hat{v}_P^*$.

If $\hat{v}_P^*$ is a point, then let $pol(P) = pol(P) \cap \{x \in \Re^n \mid (\hat{v}_P^*)^T x \leq 1\}$.

If $\hat{v}_P^*$ is a direction, then let $pol(P) = pol(P) \cap \{x \in \Re^n \mid (\hat{v}_P^*)^T x \leq 0\}$.

Else let $pol(P) = pol(P) \cap \{x \in \Re^n \mid r^T x \leq 0\}$.

Determine $\overline{EP}^*$ of the representation of $pol(P)$.

Let $\overline{EP} = \overline{EP} \cup \overline{EP}^*$ and $k = k + 1$.

It is important to point out that when a better extreme point v_P^* is found, v is cut by the γ-concavity cut relative to v. The polyhedral annexation algorithm is finite and finds a global minimum since it consists of a finite number of cycles because D has only a finite number of

extreme points not satisfying at equality any of the previous γ-concavity cut relative to v and each cycle of iterations is finite by construction.

4.4 *Numerical Experiments and Analyses*

The first part of this section consists in the numerical experimentations and analyses of the methods for dealing with degeneracy in *BCP*. Even if the outer approximation algorithm and the polyhedral annexation algorithm are also directly affected by degeneracy when the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries is used, the comparison of the methods for dealing with degeneracy is carried out in the context of the cutting hyperplane algorithm. The second part of this section consists in the numerical experimentations and analyses of the constructive enumeration algorithms used in the outer approximation algorithm and in the polyhedral annexation algorithm including, when necessary, the best method for dealing with degeneracy previously identified.

The objective function used to perform the numerical experimentations and analyses is

$$-\|x\|^2 \ln(1 + \|x\|^2)$$

which is not a monotone decreasing function of a linear nonnegative function on $\Re^n_+$ (refer to Thuan (1997) for more details). It has to be pointed out that prior to applying any of the *BCP* algorithms, a subroutine is performed in order to ensure that D is full dimensional and to rewrite the resulting polyhedron in a form compatible with the problem structure introduced in the first section. Instances of D of various sizes, given m and n, are defined pseudo-randomly in order to test the various *BCP* algorithms. The coefficients of A are pseudo-randomly generated in the interval [-1, 1] and the coefficients of b are the sum of the elements of the row in question to which a pseudo-random number generated in the interval [0, 2] is added. Moreover, other instances of D are used : Problem_2.3 to Problem_2.7 of Floudas and Pardalos (1990) and Problem_A1 to Problem_A3 defined in Appendix. Finally, it has to be pointed out also that all the numerical results reported in the following tables were obtained with the following stopping criterion : the *BCP* algorithm converged or was stopped as soon as it finished the iteration during which the 300 seconds of total CPU time is attained.

Regarding the implementation, all the algorithms are programmed in C using floating point arithmetic with the exception of the extended $L - DG^+$ algorithm which is based on the integer pivoting rules of Edmonds and Maurras (1997) performed in exact integer arithmetic using GNU Multiple Precision library (version 4.1.4) because it is extremely sensitive to numerical errors; GNU hash table library (version 0.5.4) is used to perform searching whenever necessary; Fortran BLAS\LAPACK libraries (version 3.0) are used for linear algebra computations in floating point arithmetic; and ILOG-CPLEX (version 8.1) is used for maintaining polyhedra information and LP optimization. It has to be pointed out that the

initialization of the computer implementation of the extended $L - DG^+$ algorithm requires to truncate double precision number thus creating slighlty different results compared to the computer implementation of the extended $L - DG^+$ algorithm using floating point arithmetic (this last approach cannot be used here since full refactorisation is performed for each B_k in order to minimize numerical errors and is therefore much slower). All tests are performed on a Sun Ultra-Sparc 1.2 GHz computer with 2 GB of RAM where the Sun f77 compiler is used to compile the BLAS\LAPACK libraries and the Sun C compiler is used to compile all other programmed algorithms and libraries.

4.4.1 *Comparison of Methods to Deal with Degeneracy*

To compare numerically the methods for dealing with degeneracy in the context of the cutting hyperplane algorithm, Table 4.1 reports the total CPU time in seconds for the enumeration of the extreme rays of polyhedral cones pointed on extreme points encountered over the application of the local descent subroutine. Specifically, the total CPU time for all extreme points of degeneracy degree $\sigma_i = i$ defined as the number of saturated constraints at the point in question minus n is reported under heading T_{σ_i}. Either the double description algorithm based on the min-index insertion order (Alg. = 1) or the extended $L - DG^+$ algorithm (Alg. = 2) is used to deal with degeneracy where we applied each one successively on each strong degenerate extreme points.

Table 4.1.

D	m	n	Alg.	T_{σ_1}	T_{σ_2}	T_{σ_3}	T_{σ_4}	T_{σ_5}	T_{σ_6}	T_{σ_7}	T_{σ_8}	T_{σ_9}	$T_{\sigma_{10+}}$
Problem_2.3	19	12	1	10.83	7.90	3.41	0	0	0	0	0	0	0
Problem_2.3	19	12	2	10.51	8.35	3.86	0	0	0	0	0	0	0
Problem_2.6	15	10	1	0.49	0	0	0	0	0	0	0	0	0
Problem_2.6	15	10	2	0.19	0	0	0	0	0	0	0	0	0
Problem_A1	33	16	1	0.07	0.01	0	0	0	0	0	0	0	0
Problem_A1	33	16	2	0.31	0.03	0	0	0	0	0	0	0	0
Problem_A2	81	32	1	0.18	0.33	1.14	1.51	2.46	4.36	1.93	5.55	0	0
Problem_A2	81	32	2	0.36	0.83	2.71	4.05	7.41	16.35	8.88	95.29	0	0
Problem_A3	145	64	1	0	0	0	0	0	0	0	0	0	1.56
Problem_A3	145	64	2	0	0	0	0	0	0	0	0	0	273.52

These results indicate that the double description algorithm is in general faster than the extended $L - DG^+$ algorithm for these instances of D. More specifically, it is observed for Problem_2.3 and Problem_2.6 that the double description algorithm performs slightly better compared to the extended $L - DG^+$ algorithm while for Problem_A2 and Problem_A3, this

advantage is significant. This last phenomenon may be due to the fact that the polyhedral cones pointed on strong degenerate extreme points encountered over the application of the cutting hyperplane algorithm might belong to one of the pathological families of polyhedra determined by Avis, Bremner and Seidel (1997) for which the application of any pivoting enumeration algorithm based on perturbation is problematic. The dominance of one algorithm for treating degeneracy in *BCP* over the other as observed in the above numerical results is not necessarily the result of a more efficient algorithm and one must qualify the previous conclusion. More precisely, it would be false to conclude that the double description algorithm is in general faster than the extended $L - DG^+$ algorithm for any instance of D encountered in *BCP* since each algorithm exhibits a pathological behavior for specific families of polyhedra. However, according to the above numerical experimentation, it is possible to conclude that the occurence of pathological behavior for the double description algorithm is less likely than for the extended $L - DG^+$ algorithm.

4.4.2 *Comparison of Constructive Enumeration Algorithms*

In order to compare numerically the double description algorithm and the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries based on the double description algorithm used to deal with degeneracy, recall that it is required for the latter algorithm to verify if a given extreme point is already generated. To this end, a hash table is used where the key is constructed by setting to 1 the bit of long long integer of the index corresponding to saturated constraints for the extreme point in question and to 0 the bit of the index of all other constraints (concatenation is used whenever $m + n > 64$). We tested only one value for the size of the hash table (n^3) which is fairly large compared to an expectation of the number of new extreme points generated at this iteration of the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries; this large value tends to decrease the number of collisions. The total CPU time in seconds for each constructive enumeration algorithm is reported in Table 4.2 under respective heading DD and HTV for the outer approximation algorithm (Alg. = 1) and the polyhedral annexation algorithm (Alg. = 2). Heading ext. points corresponds to the number of extreme points of the last iteration of the *BCP* algorithm while HTV (DD) corresponds to the total CPU time in seconds for the enumeration of the extreme rays of polyhedral cones pointed on strong degenerate extreme points encountered over the application of the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries.

Table 4.2.

D	m	n	Alg.	ext. points	DD	HTV	HTV (DD)
Problem_2.7	10	20	1	177310	1127.99	62.70	0
Problem_2.7	10	20	2	62818	384.78	N/A	N/A
Rand1_16_24	24	16	1	89038	387.08	36.18	0
Rand1_16_24	24	16	2	52861	386.11	N/A	N/A
Rand2_16_24	24	16	1	8609	2.39	2.05	0
Rand2_16_24	24	16	2	37052	316.71	N/A	N/A
Rand3_16_24	24	16	1	14640	7.77	3.95	0
Rand3_16_24	24	16	2	73949	358.08	N/A	N/A
Problem_A1	33	16	1	552	0.06	0.17	0.05
Problem_A1	33	16	2	17782	313.49	N/A	N/A
Problem_A2	81	32	1	39383	358.10	118.95	112.05
Problem_A2	81	32	2	89475	336.36	N/A	N/A
Problem_A3	145	64	1	43534	120.45	504.58	488.24
Problem_A3	145	64	2	7722	305.04	N/A	N/A

Prior to discussing the results reported in Table 4.2, two remarks have to be made. First, the results for some instances of D tested with the cutting hyperplane algorithm are not reported in Table 4.2 due to a rapid convergence of either the outer approximation or the polyhedral annexation algorithm. Second, no numerical results are available for the polyhedral annexation algorithm based on the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries for these instances of D since many numerical errors due to floating point arithmetic occured while constructing the key for the extreme points of the polar set of the growing polyhedron inner approximating G_γ. From the figures appearing in Table 4.2, it is clear that the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries based on the double description algorithm used to deal with degeneracy generally outperforms the double description algorithm for all these instances except for Problem_A3. This is due to the fact that searching for an extreme point in the hash table of adequate size for the former algorithm is much more efficient compared to the adjacency test of the latter algorithm which could not be implemented efficiently using a hash table. However for Problem_A3, the fact that the double description algorithm is faster than the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries is due to the presence of many strong degenerate extreme points for which it is necessary to determine the extreme rays of polyhedral cones pointed on these strong degenerate extreme points explaining approximately 95% of total CPU time. Consequently, it is true to conclude, according to these results, that the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries outperforms the double description algorithm for non-degenerate D, while for D having

strong degenerate extreme points, it is not possible to establish the clear dominance of one constructive enumeration algorithm over the other.

4.5 *Conclusion*

In this paper, we numerically compare the double description algorithm and the extended $L-DG^{+}$ algorithm as methods for dealing with degeneracy in *BCP*. Based on the tested instances of D, we can conclude that the former algorithm outperforms the latter. Moreover, the adaptation of the constructive enumeration algorithm of Horst, Thoai and de Vries outperforms the double description algorithm when D is such that all its extreme points are non-degenerate, while if it is not the case, then it is not possible to establish a clear dominance of one constructive enumeration algorithm over the other. Finally, it has to be pointed out that the conical splitting algorithm often used in *BCP* are somehow affected by strong degeneracy; this will be a topic of a future paper.

4.6 *Appendix*

4.6.1 *Problem_A1*

$$\sum_{i=1,\ldots,16} x_i = 1;$$

$$x_{i+2} \le x_i \le x_{i+1} \qquad \text{for } i = 1,5,9,13;$$

$$x_i \le 2x_{i+2} + x_{i+3} \le x_{i+1} \quad \text{for } i = 1,5,9,13;$$

$$0.0002 \le x_i \le 0.1 \qquad \text{for } i = 1,\ldots,16.$$

4.6.2 *Problem_A2*

$$\sum_{i=1,\ldots,32} x_i = 1;$$

$$x_{i+2} \le x_i \le x_{i+1} \qquad \text{for } i = 1,5,9,13,17,21,25,29;$$

$$x_i \le 2x_{i+2} + x_{i+3} \le x_{i+1} \quad \text{for } i = 1,5,9,13,17,21,25,29;$$

$$x_i \le x_{i+16} \qquad \text{for } i = 1,\ldots,16;$$

$$0.0002 \le x_i \le 0.1 \qquad \text{for } i = 1,\ldots,32.$$

4.6.3 *Problem_A3*

$$\sum_{i=1,\ldots,64} x_i = 1;$$

$$x_{i+2} \le x_i \le x_{i+1} \qquad \text{for } i = 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61;$$

$$x_i \le 2x_{i+2} + x_{i+3} \le x_{i+1} \qquad \text{for } i = 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61;$$

$$x_{i+48} \le x_{i+32} \le x_i \le x_{i+16} \qquad \text{for } i = 1,\ldots,16;$$

$$0.0002 \le x_i \le 0.1 \qquad \text{for } i = 1,\ldots,64.$$

Bibliographie

P. ARMAND (1993), "Bounds on the Number of Vertices of Perturbed Polyhedra", *Annals of Operations Research*, vol. 47, 249-269.

D. AVIS (2000), "A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm", *Polytopes - Combinatorics and Computation*, vol. 29, 177-198.

D. AVIS, D. BREMNER and R. SEIDEL (1997), "How Good Are Convex Hull Algorithms?", *Computational Geometry*, vol. 7, 265-301.

R. CARVAJAL-MORENO (1972), "Minimization of a Concave Functions Subject to Linear Constraints", *Operations Research Center Report*, ORC 72-3.

G.B. DANTZIG and C. EAVES (1973), "Fourier-Motzkin Elimination and its Dual", *Journal of Combinatorial Theory*, vol. 14, 288-297.

J. EDMONDS and J.-F. MAURRAS (1997), "Note sur les Q-matrices d'Edmonds", *RAIRO*, vol. 31, 203-209.

J.B.J. FOURIER (1826), "Solution d'une question particulière du calcul des inégalités", *Oeuvres II*, 317-328.

J.E. FALK and R.M. SOLAND (1969), "An Algorithm for Separable Nonconvex Programming Problems", *Management Science*, vol. 15, 550-569.

C.A. FLOUDAS and P.M. PARDALOS (1990), "A Collection of Test Problems for Constrained Global Optimization Algorithms", *Lecture Notes in Computer Science*, vol. 455.

K. FUKUDA and A. PRODON (1996), "The Double Description Method Revisited", *Combinatorics and Computer Science*, vol. 1120, 91-111.

T. GAL (1985), "On the Structure of the Set Bases of a Degenerate Point", *Journal of Optimization Theory and Applications*, vol. 45, 577-589.

T. GAL and F. GEUE (1992), "A New Pivoting Rule for Solving Various Degeneracy Problems", *Operations Research Letters*, vol. 11, 23-32.

F. GEUE (1993), "An Improved N-Tree Algorithm for the Enumeration of All Neighbors of a Degenerate Vertex", *Annals of Operations Research*, vol. 47, 361-391.

H. GREENBERG (1975), "An Algorithm for Determining Redundant Inequalities and All Solutions to Convex Polyhedra", *Numerische Mathematik*, vol. 24, 19-26.

K.L. HOFFMAN (1981), "A Method for Globally Minimizing Concave Functions over Convex Sets", *Mathematical Programming*, vol. 20, 20-32.

R. HORST (1976), "An Algorithm for Nonconvex Programming Problems", *Mathematical Programming*, vol. 10, 312-321.

R. HORST, N.V. THOAI and J. de VRIES (1988), "On Finding New Verticies and Redundant Constraints in Cutting Plane Algorithms for Global Optimization", *Operations Research Letters*, vol. 7, 85-90.

R. HORST and H. TUY (1996), "Global Optimization : Deterministic Approaches", Third edition, Springer-Verlag Editors.

T.S. MOTZKIN, H. RAIFFA, G.L. THOMPSON and R.M. THRALL (1953), "The Double Description Method", Contribution to the Theory of Games II, *Annals of Mathematics Studies*, 51-73.

E. SPRINGUEL, B. GENDRON and P. SORIANO (2005), "Extension to the $L - DG^+$ for the Enumeration of Extreme Rays", to appear.

T.V. THIEU, B.T. TAM and V.T. BAN (1983), "An Outer Approximation Method for Globally Minimizing a Concave Function Over a Compact Convex Set", *Acta Mathematica Vietnamica*, vol. 8, 21-40.

H.D. THUAN (1997), "Can Linear Programs Be Used to Test Global Optimization Algorithms?", *Computing*, vol. 59, 91-93.

H. TUY (1964), "Concave Minimization under Linear Constraints", *Soviet Mathematics Doklady Academii Nauk SSSR*, vol. 5, 1437-1440.

H. TUY (1990), "On a Polyhedral Annexation Method for Concave Minimization", *Optimization and Mathematical Economics*, 248-260.

H. VAISH and C.M. SHETTY (1976), "The Bilinear Programming Problem", *Naval Research Logistics Quarterly*, vol. 23, 303-309.

Chapitre 5

A Simple Finite Conical Splitting Algorithm for Basic Concave Programming

La référence de cet article est :
E. SPRINGUEL, B. GENDRON and P. SORIANO (2005), "*A Simple Finite Conical Splitting Algorithm for Basic Concave Programming*", soumis à Mathematical Programming en Octobre 2005.

A Simple Finite Conical Splitting Algorithm for Basic Concave Programming

Eric Springuel[1,2], Bernard Gendron[1,2] and Patrick Soriano[2,3]

[1] Department of Computer Science and Operations Research, University of Montréal
2920, Chemin de la Tour, Montréal, Québec, Canada, H3T 1J4

[2] Centre for Research on Transportation
2920, Chemin de la Tour, Montréal, Québec, Canada, H3T 1J4

[3] Department of Management Science, H.E.C. Montréal
3000, Chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada, H3T 2A7

Summary

In the context of basic concave programming, we introduce a simple conical splitting algorithm finitely converging to a global minimum based on the use of the extreme point ranking algorithm of Murty.

5.1 *Introduction*

In the context of basic concave programming (*BCP*) which consists in minimizing a continuous concave function $f: \Re^n \to \Re$ over a bounded polyhedron $D \subset \Re^n$, a global minimum v_f^* of f over D is one of its extreme points. The application of standard convex optimization techniques may lead to local minima (locality is defined here by the set of neighboring extreme points) which are not global minima thus requiring to define algorithms overcoming this difficulty. Written as

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } Ax \geq b \\ &\quad x \geq 0 \end{aligned}$$

where $A \in \Re^m \times \Re^n$, $b \in \Re^m$ and assuming that D has a non-empty interior, i.e. that $int(D) = \{x \in \Re^n \mid Ax - b > 0,\ x > 0\} \neq \emptyset$, Tuy (1964) lays down the foundations for most of the *BCP* algorithms with the introduction of the first version of the conical splitting algorithm. Zwart (1973) demonstrates that this version might be prone to cycling and then Zwart (1974) proposes a slightly modified version of the method in order to (supposedly) ensure the convergence to an ε-global optimum in the sense that there might exist an extreme point v of D and a point v' such that $f(v') \leq f(v)$ where $\|v - v'\| \leq \varepsilon$. However, it is shown in Tuy (1990) that this version sometimes leads to an incorrect solution. Thoai and Tuy (1980) introduce the first version of the conical splitting algorithm converging to an ε'-global minimum $v_{f,\varepsilon'}^*$ in the sense that $f(v_{f,\varepsilon'}^*) - \varepsilon' \leq f(x) \ \ \forall \ x \in D$. This algorithm is based on an exhaustive conical splitting, i.e. for all the infinite decreasing subsequences of polyhedral

cones of every infinite sequence of embedded polyhedral cones, each subsequence converges to a ray. Even if the version of Thoai and Tuy (1980) is finitely converging to an ε'-global optimum only when $\varepsilon' > 0$, the first version of a conical splitting algorithm finitely converging to a global minimum when $\varepsilon' = 0$ is due to Hamami and Jacobsen (1988). However, this algorithm requires the introduction of an exhaustive non-degenerate conical splitting, i.e. for all the infinite decreasing subsequences of polyhedral cones of every infinite sequence of embedded polyhedral cones, each subsequence converges to a ray and normal vectors of corresponding concavity cuts tend to be orthogonal to $G_\gamma = \{x \in \Re^n | f(x) \geq f(v_f^*) = \gamma\}$ at the point where the ray in question crosses G_γ (refer to the third section for the definition of a concavity cut). Since this last requirement is hard to satisfy in practice, Tuy (1991) proposes a broader class of conical splitting algorithms converging to an ε'-global minimum based on a normal conical splitting which consists in a combined strategy of an exhaustive conical splitting and a non-degenerate conical splitting defined in a weaker sense when compared to the definition of Hamami and Jacobsen (1988). As in the case of the version of Thoai and Tuy (1980), each algorithm of this class is finitely converging when $\varepsilon' > 0$ while it may be infinite when $\varepsilon' = 0$.

In this paper, we introduce a simple conical splitting algorithm finitely converging to a global minimum when $\varepsilon' = 0$. The finiteness of the method is guaranteed by the fact that there is a finite number of conical splittings to be perfomed with respect to rays emanating from the extreme point on which all polyhedral cones are pointed through extreme points of D determined via the use of the extreme point ranking algorithm of Murty (1968). Contrary to the polyhedral annexation algorithm of Vaish and Shetty (1976) or Tuy (1990), both finitely converging to a global minimum when $\varepsilon' = 0$, this algorithm does not require the enumeration of all the extreme points of the polar set of the growing polyhedron that inner approximates G_γ via the application of any constructive enumeration algorithm. Instead, the simple finite conical splitting algorithm presented here is such that it sometimes applies the extreme point ranking algorithm to partially enumerate the extreme points of a portion of D.

The paper is organized as follows. The next section describes the extreme point ranking algorithm. The third section introduces the simple finite conical splitting algorithm. In the fourth section, we analyze numerical results of experiments. Finally, the last section is devoted to concluding remarks.

5.2 *Extreme Point Ranking Algorithm*

Given an objective function $\bar{c}$ to be maximized over a bounded polyhedron $\overline{D}$ and an optimal solution $v_{\bar{c}}^*$, determined, for example by the simplex algorithm, such that $\bar{c}^T v_{\bar{c}}^* < \infty$, the extreme point ranking algorithm consists in determining the sequence of all the extreme points

of $\overline{D}$ in a non-increasing order with respect to $\bar{c}$ where its first point is $v_{1,\bar{c}} = v_{\bar{c}}^*$, the second point $v_{2,\bar{c}}$ is found among the neighboring extreme points of $v_{1,\bar{c}}$ while the third point $v_{3,\bar{c}}$ is identified over the neighboring extreme points of $v_{1,\bar{c}}$ or those of $v_{2,\bar{c}}$ and so on. In this fashion, every extreme point of $\overline{D}$ are generated since there exists a path which connects any of them to $v_{1,\bar{c}}$. Moreover, the extreme point ranking algorithm can naturally be implemented with the usage of a max-heap data structure as shown below.

EXTREME POINT RANKING ALGORITHM

Initialization

Determine a minimum $v_{\bar{c}}^*$ for a given $\bar{c}$ over $\overline{D}$ and let $v_{1,\bar{c}} = v_{\bar{c}}^*$.

Let H be the max-heap initialized with the couple $(c^T v_{1,\bar{c}}, v_{1,\bar{c}})$.

Let $L = \{v_{1,\bar{c}}\}$ be the set of extreme points of $\overline{D}$ visited so far.

While $H \neq \emptyset$, do :

Let $(c^T v_{k,\bar{c}}, v_{k,\bar{c}})$ be the root of H.

Determine the neighboring extreme points of $v_{k,\bar{c}}$.

For each neighboring extreme point $v_{k',\bar{c}}$ of $v_{k,\bar{c}}$, do :

If $\bar{c}^T v_{k',\bar{c}} \leq \bar{c}^T v_{k,\bar{c}}$, then :

If $v_{k',\bar{c}} \notin L$, then insert $(c^T v_{k',\bar{c}}, v_{k',\bar{c}})$ in H.

Remove the root of H.

Write $(c^T v_{k,\bar{c}}, v_{k,\bar{c}})$ to the output file.

One critical aspect of this algorithm consists in its sensitivity to degeneracy when the enumeration of the neighboring extreme points of $v_{k,\bar{c}}$ has to be performed, but it is important to point out that either the extended arbitrary lexicographic positive degeneracy graph algorithm of Springuel, Gendron and Soriano (2005a) or the double description algorithm of Motzkin, Raiffa, Thompson and Thrall (1953) both described and analyzed in Springuel, Gendron and Soriano (2005b) are applicable in this context.

5.3 *Simple Finite Conical Splitting Algorithm*

Based on Tuy (1964), a conical splitting algorithm consists in demonstrating $D \backslash G_\gamma = D \backslash \{x \in \Re^n \mid f(x) \geq \gamma\} = \emptyset$ where $f(v) = \gamma$ and $\gamma' = \gamma - \varepsilon$ ($\varepsilon \geq 0$), v being a local minimum candidate as a global minimum. If v is a non-degenerate or a weak degenerate extreme point of D (i.e. degeneracy is only due to the presence of redundant constraints defining v), then the smallest polyhedral cone K pointed on v containing D has only n extreme rays denoted r_j and on each one, there exists a point $z_j = v + \alpha_j r_j$ where $\alpha_j = \sup\{\theta \in \Re_+ \mid f(v + \theta r_j) = \gamma\}$. It is important to point out that it is not possible to have $v = z_j$ for any given j since v is a local minimum of f over D for which all its neighboring

extreme points are contained in G_γ. By construction, there is only one hyperplane h_z called the γ-concavity cut relative to v containing all the z_j written as $h_z^T(x-v)=1$ where the j^{th} component of h_z is equal to $1/\alpha_j$. The simplex S determined by the convex envelope of all the z_j and v is such that it corresponds to the intersection of K and the half-space containing v generated by h_z. By defining v_z^* and μ_z to be respectively one optimal solution and the optimal value of

$$\max_{x\in D} \; h_z^T(x-v),$$

if $\mu_z \leq 1$, then $D \subset S$ and therefore $D \subset G_\gamma$ which means that v is a global minimum while if $1 < \mu_z$, then v_z^* belongs either to $D\backslash G_\gamma$ or to $G_\gamma\backslash S$. The conical splitting algorithm prescribes for the second case to split K with respect to a ray r of K emanating from v such that each element of the partition $\mathsf{P}(K)$ of K is also a polyhedral cone pointed on v. If no $\alpha_j \to \infty$, then $S' = K \cap \{x \in \Re^n \mid h_z^T(x-v)=1\}$ is a simplex possessing n extreme points z_j such that any point $w \in S'\backslash\{z_1,\ldots,z_n\}$ corresponding to the intersection of $v+\alpha r$ and S' is uniquely expressed as

$$w = \sum_{j=1,\ldots,n} \lambda_j z_j,\ \lambda_j \geq 0 \text{ and } \sum_{j=1,\ldots,n} \lambda_j = 1.$$

The simplicial partition $\mathsf{P}(S')$ of S' with respect to w (radial subdivision introduced by Horst (1976)) is defined by

$$\bigcup_{\{j=1,\ldots,n \mid \lambda_j>0\}} S'(j,w) = conv\{z_1,\ldots,z_{j-1},w,z_{j+1},\ldots,z_n\}$$

and such that for each simplex $S'(j,w)$, a polyhedral cone $K(j,r)$ pointed on v has its extreme rays determined by the half-lines emanating from v and passing through each of the extreme points of $S'(j,w)$, thus forming $\mathsf{P}(K)$. Once the γ-concavity cut relative to v for each polyhedral cone K' of $\mathsf{P}(K)$, the conical splitting algorithm then requires for each concavity h_z to solve

$$\max_{x\in D\cap K'} \; h_z^T(x-v)$$

of which an optimal solution is v_z^* and the optimal value is μ_z. If $\mu_z \leq 1$ for each h_z, the conical splitting algorithm is stopped since v is a global minimum, but since it is not generally the case, the polyhedral cone K' of $\mathsf{P}(K)$ having the highest μ_P is splitted.

Whenever v is a strong degenerate extreme point (i.e. degeneracy is not only due to the presence of redundant constraints defining v), then it might be impossible to find a h_z containing all the p points z_j where $p > n$ through the construction of the concavity cut relative to v. To solve this problem, Carvajal-Moreno (1972) proposes to find a h_z corresponding to a concavity cut by determining a basic solution to

$$h_z^T r_j \geq 1/\alpha_j \text{ for } j = 1,2\ldots,p.$$

For the $p-n$ extreme rays of K not constituting a basic solution, the points $\bar{z}_j = \{x \in \{\theta_j \in \Re_+ \mid v+\theta_j r_j\} \mid h_z^T(x-v)=1\}$ corresponding to the intersection of these extreme rays and h_z are such that $0 < \theta_j < \alpha_j$, in order for the γ-concavity cut relative to v to

remain valid. Since many are available, it is preferable to generate the γ-concavity cut relative to v that covers the largest portion of D, i.e. one that minimizes the sum of the difference between α_j and θ_j which is given by an optimal solution of

$$\min \sum_{j=1,2,\ldots,p} (h_z^T r_j - 1/\alpha_j)$$
$$\text{s.t. } h_z^T r_j \geq 1/\alpha_j \text{ for } j = 1,2\ldots,p.$$

In addition, the preceeding conical subdivision cannot be applied since S' is not a simplex anymore. However, if for each hyperplane of the representation of K that does not contain r, the convex envelope of the extreme rays belonging to the hyperplane in question and r is determined, then the union of all these convex envelopes forms $\mathsf{P}(K)$ as shown below.

Proposition

Let $K \subset \Re^n$ be an n-dimensional pointed polyhedral cone and let $ER(K) = \{r_0,\ldots,r_p\}$ where $p \geq n$ be its set of generating rays. For a given $r \in K$, a partition $\mathsf{P}(K)$ of K is

$$\mathsf{P}(K) = \bigcup_{i=1,\ldots,p} conv\{r, \{r_j \in ER(K) \mid r_j \in h_i\}\}$$

where h_i is its i^{th} hyperplane of the representation of K not containing r.

Proof

First, it is trivial to observe that if $r' \in \mathsf{P}(K)$, then $r' \in K$. Second, if $r' \in K$ lies on one of its l hyperplane h_i of its representation, then it is also trivial to see that $r' \in \mathsf{P}(K)$. However, if $r' \in int(K)$, then the half-plane emanating from r containing r' intersects at least one h_i and r' belongs, by construction, to the convex envelope of the extreme rays of one of these h_i and r. Third, if there exists a $r' \in int\big(conv\{r, \{r_j \in ER(K) \mid r_j \in h_i\}\}\big) \cap int\big(conv\{r, \{r_j \in ER(K) \mid r_j \in h_{i'}\}\}\big)$ for $i \neq i'$, then the half-plane emanating from r passing through r' intersects both the $rel\_int(h_i)$ and the $rel\_int(h_{i'})$ which is impossible. □

It has to be pointed out that the above construction has to be applied if one or more $\alpha_j \to \infty$. Moreover, the hyperplanes of the representation of each polyhedral cone of $\mathsf{P}(K)$ are found by enumerating the extreme rays of the polyhedral cone corresponding to the polar set of each element of $\mathsf{P}(K)$ of which all the hyperplanes of its representation are known (all the extreme rays of each element of $\mathsf{P}(K)$ are determined by the proposition).

Within this framework, Thoai and Tuy (1980) demonstrate that the conical splitting algorithm is converging to a global minimum if it is performed with respect to the ray emanating from v through the mid-point of one of the longest edge of S'. But this exhaustive conical splitting provides a slow convergence since the problem characteristics are not exploited. In order to remedy this problem, Tuy (1991) determines a broader class of conical splitting algorithms converging to a global minimum based on a normal conical splitting which consists in a

combined strategy of an exhaustive conical splitting and a non-degenerate conical splitting defined as following : for every infinite sequence of embedded polyhedral cones $\{K, K', ...\}$ where the conical splitting is performed with respect to a ray emanating from v through v_z^* for all polyhedral cones of the sequence for all but finite number of times, there exists a subsequence Δ and a constant η such that $\|h_z\| \leq \eta$ for all polyhedral cones of the subsequence. Jacobsen (1981) attempts unsuccessfully (see Tuy (1991)) to prove the convergence of the conical splitting algorithm based only on conical splittings performed with respect to a ray emanating from v through v_z^*. Locatelli (1999) and Jaumard and Meyer (2001) both provide different proofs of this result. Nevertheless, the first conical splitting algorithm finitely converging to a global minimum is due to Hamami and Jacobsen (1988), but their exhaustive non-degenerate conical splitting is hard to use in pratice. The use of a simple sufficient condition to ensure finiteness of the conical splitting algorithm is then imperative : perform conical splitting only with respect to r which passes through extreme point of D as in the case of the polyhedral annexation algorithm. According to the following proposition, a deliberate choice of an extreme point of D other than v which belongs to K in order to implement such a conical splitting is v_z' which maximizes the value of $h_z^T(x - v)$ over $D \cap K$ as determined by the extreme point ranking algorithm where $\bar{c} = h_z$ and which is stopped as soon as v_z' is found.

Proposition

If $h_z^T(v_z' - v) \leq 1$ over $D \cap K$, then fathom K.

Proof

Only the extreme points of D need to be considered as candidates of being a global minimum and if all of them belonging to $D \cap K$ are such that they are in G_γ as asserted by $h_z^T(v_z' - v) \leq 1$, then no remaining extreme point of D in K are of interest. □

Lemma

If the extreme point ranking algorithm is such that no extreme point of D are found so far over $D \cap K$ and $h_z^T(w - v) \leq 1$ where w is the root of the heap data structure used in the extreme point ranking algorithm, then fathom K.

SIMPLE CONICAL SPLITTING ALGORITHM

Initialization :

Determine a local minimum v of f over D.

Let $\gamma = f(v)$.

Determine the smallest polyhedral cone K pointed on v that contains D.

Construct the γ-concavity cut relative to v where its normal vector is h_z.

Let $\mathsf{M} = \mathsf{P}(K) = \{K\}$ and $\bar{h}_z = h_z$.

Iteration $k = 0, 1, \ldots$:

For each cone K of $\mathsf{P}(K)$:

Solve $\max_{x \in D \cap K} h_z^T(x - v)$ of which the optimal value is μ_z.

Let v_z^* be an optimal solution of this problem.

If $f(v_z^*) < \gamma$, then $D = D \cap \left\{x \in \Re^n \mid \bar{h}_z^T(x - v) \geq 1\right\}$ and go to the initialization.

If $1 < \mu_z$, then :

Use the extreme point ranking algorithm to solve

$$\max_{x \in D \cap K \cap EP(D)} h_z^T(x - v)$$
$$\text{s.t. } h_z^T(x - v) \geq 1$$

If there is a solution, then let v_z' be it and $\mu_z = h_z^T(v_z' - v)$.

Else let $\mu_z = 0$.

Fathom any cone $K \in \mathsf{P}(K)$ such that $\mu_z \leq 1$ and update correspondingly M.

If $\mathsf{M} = \emptyset$, then STOP since v is a global minimum of f over D.

Else :

Determine $K^* \in \arg \max_{K \in \mathsf{M}} \mu_z$.

Apply the conical splitting with respect to the ray emanating from v through v_z'.

Let $\mathsf{P}(K)$ be the resulting conical partition of K^*.

$\forall\ K \in \mathsf{P}(K)$, construct the γ-concavity cut relative to v having normal vector h_z.

Let $\mathsf{M} = (\mathsf{M} \backslash K^*) \cup \mathsf{P}(K)$ and $k = k + 1$.

It is important to point out that when a better extreme point v_z' is found, v is cut by the γ-concavity cut of K relative to v. The simple conical splitting algorithm is finite and finds a global minimum since it consists of a finite number of cycles : 1) D has only a finite number of extreme points not satisfying at equality any of the previous γ-concavity cut of K relative to v; and 2) each cycle of iterations is finite by construction since there is only a finite number of extreme points of D used to perform the conical splitting with respect to a ray emanating from v passing through the extreme point in question.

The simple finite conical splitting algorithm has one advantage when compared to the polyhedral annexation algorithm which also performs conical splitting only with respect to a ray emanating from v passing through the extreme points of D. Specifically, the polyhedral annexation algorithm requires the enumeration of all the extreme points of the polar set of the growing polyhedron that inner approximates G_γ via the application of any constructive enumeration algorithm, which can be quite cumbersome. The simple finite conical splitting algorithm is such that during the first iteration of the first cycle of iterations, only one iteration of the extreme point ranking algorithm is necessary to identify an extreme point of D in order to perform the conical splitting and as long as the number of iterations of any cycle of

iterations of the simple finite conical splitting algorithm is increasing, the number of iterations of the extreme point ranking algorithm might increase, but the portion of D of interest is shrinking simultaneously since the K are smaller thus leading to believe that the number of iterations of the extreme point ranking algorithm will not be too large.

5.4 *Numerical Experiments and Analyses*

The objective function used to perform the numerical experimentations and analyses is

$$-\|x\|^2 \ln(1 + \|x\|^2)$$

which is not a monotone decreasing function of a linear nonnegative function on $\Re^n_+$ (refer to Thuan (1997) for more details). It has to be pointed out that prior to apply any of the *BCP* algorithms, a subroutine is performed in order to ensure that D is full dimensional and to rewrite the resulting polyhedron in a form compatible with the problem structure introduced in the first section. Bounded instances of D of size m and n are defined pseudo-randomly in order to test the *BCP* algorithms. The coefficients of A are pseudo-randomly generated in the interval [-1, 1] and the coefficients of b are the sum of the elements of the row in question to which a pseudo-random number generated in the interval [0, 2] is added. Two other instances of bounded D are used : Problem_2.5 and Problem_2.6 of Floudas and Pardalos (1990). Finally, it has to be pointed out also that all the numerical results reported in the following tables were obtained with the following stopping criterion : the *BCP* algorithm converged or stopped as soon as the iteration is finished during which the 7200 seconds of total CPU time is attained.

Regarding the implementation, all the algorithms are programmed in C using floating point arithmetic; GNU hash table library (version 0.5.4) is used to perform searching whenever necessary; Fortran BLAS\LAPACK libraries (version 3.0) are used for linear algebra computations in floating point arithmetic; and ILOG-CPLEX (version 8.1) is used for maintaining polyhedra information and LP optimization. All the tests are performed on a Sun Ultra-Sparc 1.2 GHz computer with 2 GB of RAM where the Sun f77 compiler is used to compile the BLAS\LAPACK libraries while the Sun C compiler is used to compile all other programmed algorithms and libraries.

To verify if a given extreme point of $D \cap K$ is already generated by the extreme point ranking algorithm, a hash table is used where the key is constructed by setting to 1 the bit of long long integer of the index corresponding to saturated constraints for the extreme point in question and setting to 0 the bit of the index of all other constraints (concatenation is used whenever $m + n > 64$) while its size is set to be large extremely large (n^3 slots) compared to an expectation of the number of extreme points of $D \cap K$ generated by the extreme point ranking algorithm. The following table reports the CPU time in seconds for the application of the simple finite conical splitting algorithm where the double description algorithm is used

whenever necessary and the normal conical splitting algorithm on instances of D described above where for both, the generalized conical splitting is always used.

Table 5.1.

D	m	n	Normal C.S.A.	Simple Finite C.S.A.
Problem_2.5	18	10	CYCLED	5061.3
Problem_2.6	15	10	CYCLED	450.16
Rand1_11_12	12	11	CYCLED	525.31
Rand1_10_15	15	10	2463.42	2232.41
Rand1_16_24	24	16	CYCLED	7200.00

These results first indicate that for Problem_2.5, Problem_2.6 and Rand1_11_12, the simple finite conical splitting algorithm converged while it is not the case for the normal conical splitting algorithm which cycled as revealed through a careful analysis of the behavior of the algorithm on these instances. It is suspected that this phenomenon is due to numerical errors since floating point arithmetic is used. Secondly, the analysis of the behavior of the simple finite conical splitting algorithm applied on Problem_2.5, Problem_2.6 and Rand1_11_12 indicates that the total number of cones that remains unfathomed grows less rapidly compared to the normal conical splitting algorithm thus indicating that the new fathoming rule is contributing to the improvement of the efficiency (i.e. enhancing the speed). However, it has to be pointed out that each iteration of the simple finite conical splitting algorithm is more expensive due to the extreme point ranking algorithm. Nevertheless, the results of Rand1_10_15 seem to indicate that this algorithm is still about 9-10% faster when compared to the normal conical splitting algorithm. Finally, the simple finite conical splitting algorithm applied on Rand1_16_24 did not converge within the 7200 seconds limit of total CPU time, but the algorithm was still running normally.

5.5 *Conclusion*

In this paper, we present a simple conical splitting algorithm finitely converging to a global minimum even when $\varepsilon' = 0$. The finiteness of the method is guaranteed by the fact that there is a finite number of conical splittings to be perfomed with respect to rays emanating from the extreme point on which all polyhedral cones are pointed through extreme points of D, determined via the use of the extreme point ranking algorithm of Murty (1968). According to the numerical experimentations carried out, this algorithm is also practical from a computational point of view.

Bibliographie

R. CARVAJAL-MORENO (1972), "Minimization of a Concave Functions Subject to Linear Constraints", *Operations Research Center Report*, OR 72-3.

C.A. FLOUDAS and P.M. PARDALOS (1990), "A Collection of Test Problems for Constrained Global Optimization Algorithms", *Lecture Notes in Computer Science*, vol. 455.

M. HAMAMI and S.E. JACOBSEN (1988), "Exhaustive Nondegenerate Conical Processes for Concave Minimization on Convex Polytopes", *Mathematics of Operations Research*, vol. 13, 479-487.

R. HORST (1976), "An Algorithm for Nonconvex Programming Problems", *Mathematical Programming*, vol. 10, 312-321.

S.E. JACOBSEN (1981), "Convergence of a Tuy-Type-Algorithm for Concave Minimization Subject to Linear Inequality Constraints", *Applied Mathematics and Optimization*, vol. 7, 1-9.

B. JAUMARD and C. MEYER (2001), "On the Convergence of Cone Splitting Algorithms with ϖ-subdivisions", *Journal of Optimization Theory and Applications*, vol. 110, 119-144.

M. LOCATELLI (1999), "Finiteness of Conical Algorithms with ϖ-subdivisions", *Mathematical Programming*, vol. 85, 593-616.

T.S. MOTZKIN, H. RAIFFA, G.L. THOMPSON and R.M. THRALL (1953), "The Double Description Method", Contribution to the Theory of Games II, *Annals of Mathematics Studies*, 51-73.

K.G. MURTY (1968), "Solving the Fixed Charge Problem by Ranking the Extreme Points", *Operations Research*, vol. 16, 268-279.

E. SPRINGUEL, B. GENDRON and P. SORIANO (2005a), "Extension of the $L-DG^{+}$ Algorithm for the Enumeration of Extreme Rays", to appear.

E. SPRINGUEL, B. GENDRON and P. SORIANO (2005b), "Degeneracy and Enumeration in Basic Concave Programming", to appear.

N.V. THOAI and H. TUY (1980), "Convergent Algorithm for Minimizing a Concave Function", *Mathematics of Operations Reasearch*, vol. 5, 556-566.

H.D. THUAN (1997), "Can Linear Programs Be Used to Test Global Optimization Algorithms?", *Computing*, vol. 59, 91-93.

H. TUY (1964), "Concave Minimization under Linear Constraints", *Soviet Mathematics Doklady Academii Nauk SSSR*, vol. 5, 1437-1440.

H. TUY (1990), "On a Polyhedral Annexation Method for Concave Minimization", *Optimization and Mathematical Economics*, 248-260.

H. TUY (1991), "Normal Conical Algorithm for Concave Minimization Over Polytopes", *Mathematical Programming*, vol. 51, 229-245.

H. VAISH and C.M. SHETTY (1976), "The Bilinear Programming Problem", *Naval Research Logistics Quarterly*, vol. 23, 303-309.

P.B. ZWART (1973), "Nonlinear Programming : Counterexamples to two Global Optimization Algorithms", *Operations Research*, vol. 21, 1260-1266.

P.B. ZWART (1974), "Global Maximization of a Convex Function with Linear Inequality Constraints", *Operations Research*, vol. 22, 602-609.

Chapitre 6

Conclusion

Dans cette thèse, nous avons utilisé dans un premier temps des principes complémentaires développés dans des contextes différents dans le but de résoudre plus efficacement le problème de l'énumération des rayons extrêmes d'un cône polyédral pointé à partir de sa représentation sous forme d'hyperplans. En effet, nous avons présenté un nouvel algorithme de balayage d'un graphe perturbant la représentation sous forme d'hyperplans, en se basant sur des concepts complémentaires issus de la programmation mathématique et de la géométrie computationnelle. Par la suite, nous avons identifé dans le contexte de certains algorithmes déterministes de minimisation concave sur un polyèdre, l'algorithme le plus performant pour traiter la dégénérescence forte lors de l'énumération des rayons extrêmes d'un cône polyédral pointé. Plus précisément, nous avons comparé les performances de notre nouvel algorithme de balayage d'un graphe à celles du plus efficace des algorithmes constructifs de type double description dû à Motzkin, Raiffa, Thompson et Thrall (1953) pour traiter la dégénérescence forte. Finalement, nous avons développé un algorithme déterministe de minimisation concave sur un polyèdre, qui exploite l'algorithme le plus efficace identifié précédemment pour énumérer les rayons extrêmes d'un cône polyédral pointé. Plus spécifiquement, nous avons présenté un nouvel algorithme de division conique simple et facile à implanter qui détermine un minimum global exact en un temps fini et qui utilise l'algorithme d'ordonnancement des points extrêmes de Murty (1968) d'une manière novatrice.

Ainsi au chapitre 3, nous étendons l'algorithme $L - DG^+$ de Geue (1993) pour le problème d'énumération des rayons extrêmes d'un cône polyédral pointé à partir de sa représentation sous forme d'hyperplans en utilisant le principe sous-jacent à l'algorithme *LRS* d'Avis (2000). Ce dernier prescrit de ne pas vérifier, lors du balayage du graphe, la présence du noeud courant pour l'ajouter à ce graphe s'il n'y est pas puisque tous les noeuds ne sont générés qu'une seule fois. En plus d'être une amélioration théorique, il nous est aussi possible de conclure d'après nos expériences numériques qu'il s'agit d'une amélioration pratique, permettant de réduire le temps de calcul. Du même souffle, nous présentons l'algorithme *LRS* généralisé pour l'énumération des points extrêmes d'un polyèdre borné, étant donné sa représentation sous forme d'hyperplans. Cet algorithme utilise une perturbation lexicographique généralisée, dont

un cas particulier fort intéressant est celui induit par la règle de pivotage de Gal et Geue (1992) balayant uniquement des noeuds qui identifient toujours au moins un point extrême du polyèdre borné. Si cette règle de pivotage est utilisée, alors il est démontré théoriquement que l'algorithme *LRS* généralisé génère toujours un plus petit nombre de noeuds comparativement à l'algorithme *LRS* lorsque tous les points extrêmes du polyèdre borné sont non-dégénérés. Nos expériences numériques nous permettent également de confirmer la validité de cette affirmation pour certains polyèdres bornés possédant des points extrêmes fortement dégénérés. Au chapitre 4, nous déterminons dans le contexte particulier de l'algorithme d'hyperplans coupants pour résoudre un problème de minimisation concave sur un polyèdre que l'extension de l'algorithme $L-DG^{+}$ est moins rapide pour traiter la dégénérescence forte comparativement à l'algorithme le plus efficace du type double description de Motzkin, Raiffa, Thompson et Thrall. Dans le contexte particulier de l'algorithme d'approximation extérieure pour résoudre un problème de minimisation concave sur un polyèdre, nous démontrons que notre adaptation de l'algorithme de Horst, Thoai et de Vries (1991) domine l'algorithme le plus efficace du type double description, lorsque tous les points extrêmes du polyèdre sont non-dégénérés, tandis qu'il n'est pas possible d'établir une dominance claire lorsque le polyèdre possède des points extrêmes fortement dégénérés. Finalement, nous introduisons au Chapitre 5, dans le contexte de la minimisation concave sur un domaine polyédral borné, un nouvel algorithme de division conique simple et facile à implanter qui détermine un minimum global exact en un temps fini. La finitude de cet algorithme provient du fait que les divisions coniques ne s'effectuent que par rapport aux rayons extrêmes émanant d'un point extrême du polyèdre sur lequel tous les cônes polyédraux sont pointés. Ceux-ci possèdent toujours un autre point extrême du polyèdre, qui est dans notre algorithme déterminé par l'algorithme d'ordonnacement des points extrêmes de Murty (1968).

Comme avenues de recherche possibles suite aux travaux de cette thèse, mentionnons qu'il est possible d'utiliser l'énumération des rayons extrêmes d'un cône polyédral pointé avec une métaheuristique de type recherche avec tabous ou recherche à voisinages variables, afin de résoudre un problème de minimisation concave sur un domaine polyédral. En effet, il est possible de définir un algorithme d'hyperplans coupants qui utilise le principe de mémoire de la recherche avec tabous pour guider la recherche de plusieurs minima locaux en partageant l'information qui les caractérisent pour ensuite couper simultanément tous ces minima locaux. De plus, il est possible de définir un algorithme de division conique qui utilise les principes des métaheuristiques dans le but de déterminer le plus rapidement possible un candidat à titre de minimum global, afin que les coupes de concavité soient les plus profondes possible.

Bibliographie

P. ARMAND (1993), "Bounds on the Number of Vertices of Perturbed Polyhedra", *Annals of Operations Research*, vol. 47, 249-269.

D. AVIS (2000), "A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm", *Polytopes - Combinatorics and Computation*, vol. 29, 177-198.

D. AVIS, D. BREMNER et R. SEIDEL (1997), "How Good Are Convex Hull Algorithms?", *Computational Geometry*, vol. 7, 265-301.

D. AVIS et K. FUKUDA (1992), "A Pivoting Algorithm for Convex Hull and Vertex Enumeration of Arrangements and Polyhedra", *Discrete Computational Geometry*, vol. 8, 295-313.

E. BALAS (1971), "Intersection Cuts - a New Type of Cutting Planes for Integer Programming", *Operations Research*, vol. 19, 19-39.

M.L. BALINSKI (1961), "An Algorithm for Finding All Verticies of Convex Polyhedral Sets", *SIAM Journal*, vol. 9, 72-88.

R.G. BLAND (1977), "New Finite Pivoting Rules for the Simplex Method", *Mathematics of Operations Research*, vol. 2, 103-107.

D. BREMNER, K. FUKUDA et A. MARZETTA (1998), "Primal-Dual Methods for Vertex and Facet Enumeration", *Discrete Computational Geometry*, vol. 20, 333-358.

R. CARVAJAL-MORENO (1972), "Minimization of a Concave Functions Subject to Linear Constraints", *Operations Research Center Report*, ORC 72-3.

B. CHAZELLE (1993), "An Optimal Convex Hull Algorithm in any Fixed Dimension", *Discrete Computational Geometry*, vol. 10, 377-409.

P.-C. CHEN, P. HANSEN et B. JAUMARD (1991), "On-line and Off-line Vertex Enumeration by Adjacency Lists", *Operations Research Letters*, vol. 10, 403-409.

N.V. CHERNIKOVA (1964), "Algorithm for Finding a General Formula for the Nonnegative Solutions of a System of Linear Equations", *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 4, 151-156.

N.V. CHERNIKOVA (1965), "Algorithm for Finding a General Formula for the Nonnegative Solutions of a System of Linear Equations", *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 5, 228-233.

K.L. CLARKSON et P.W. SHOR (1988), "Algorithms for Diametral Pairs and Convex Hull that are Optimal, Randomized and Incremental", *Proceedings of 4th Annual ACM Symposium on Computational Geometry*, 12-17.

G.B. DANTZIG and C. EAVES (1973), "Fourier-Motzkin Elimination and its Dual", *Journal of Combinatorial Theory*, vol. 14, 288-297.

G.B. DANTZIG, A. ORDEN et P. WOLFE (1955), "The Generalized Simplex Method for Minimizing a Linear Function Form under Linear Inequality Restraints", *Pacific Journal of Mathematics*, vol. 5, 183-195.

M.E. DYER et L.G. PROLL (1977), "An Algorithm for Determining All Extreme Points of a Convex Polytope", *Mathematical Programming*, vol. 12, 81-96.

M.E. DYER et L.G. PROLL (1980), "Eliminating Extraneous Edges in Greenberg's Algorithm", *Mathematical Programming*, vol. 19, 106-110.

J. EDMONDS and J.F. MAURRAS (1997), "Note sur les Q-matrices d'Edmonds", *RAIRO*, vol. 31, 203-209.

J.E. FALK and R.M. SOLAND (1969), "An Algorithm for Separable Nonconvex Programming Problems", *Management Science*, vol. 15, 550-569.

C.A. FLOUDAS and P.M. PARDALOS (1990), "A Collection of Test Problems for Constrained Global Optimization Algorithms", *Lecture Notes in Computer Science*, vol. 455.

J.B.J. FOURIER (1826), "Solution d'une question particulière du calcul des inégalités", *Oeuvres II*, 317-328.

K. FUKUDA et A. PRODON (1996), "The Double Description Method Revisited", *Combinatorics and Computer Science*, vol. 1120, 91-111.

T. GAL (1985), "On the Structure of the Set Bases of a Degenerate Point", *Journal of Optimization Theory and Applications*, vol. 45, p. 577-589.

T. GAL and F. GEUE (1992), "A New Pivoting Rule for Solving Various Degeneracy Problems", *Operations Research Letters*, vol. 11, 23-32.

F. GEUE (1993), "An Improved N-Tree Algorithm for the Enumeration of All Neighbors of a Degenerate Vertex", *Annals of Operations Research*, vol. 47, 361-391.

H. GREENBERG (1975), "An Algorithm for Determining Redundant Inequalities and All Solutions to Convex Polyhedra", *Numerische Mathematik*, vol. 24, 19-26.

M. HAMAMI and S.E. JACOBSEN (1988), "Exhaustive Nondegenerate Conical Processes for Concave Minimization on Convex Polytopes", *Mathematics of Operations Research*, vol. 13, 479-487.

K.L. HOFFMAN (1981), "A Method for Globally Minimizing Concave Functions over Convex Sets", *Mathematical Programming*, vol. 20, 20-32.

R. HORST (1976), "An Algorithm for Nonconvex Programming Problems", *Mathematical Programming*, vol. 10, 312-321.

R. HORST, N.V. THOAI et J. de VRIES (1988), "On Finding New Verticies and Redundant Constraints in Cutting Plane Algorithms for Global Optimization", *Operations Research Letters*, vol. 7, 85-90.

R. HORST and H. TUY (1996), "Global Optimization : Deterministic Approaches", Third edition, Springer-Verlag Editors.

S.E. JACOBSEN (1981), "Convergence of a Tuy-Type-Algorithm for Concave Minimization Subject to Linear Inequality Constraints", *Applied Mathematics and Optimization*, vol. 7, 1-9.

B. JAUMARD and C. MEYER (2001), "On the Convergence of Cone Splitting Algorithms with ϖ-subdivisions", *Journal of Optimization Theory and Applications*, vol. 110, 119-144.

H.-J. KRUSE (1986), "Degeneracy Graphs and the Neighbourhood Problem", *Lecture Notes in Economics and Mathematical Systems*, vol. 260.

M. LOCATELLI (1999), "Finiteness of Conical Algorithms with ϖ-subdivisions", *Mathematical Programming*, vol. 85, 593-616.

M. MANAS et J. NEDOMA (1968), "Finding All Vertices of a Convex Polyhedron", *Numerische Mathematik*, vol. 12, 226-229.

T.H. MATHEISS (1973), "An Algorithm for Determining Irrelevant Constraints and All Vertices in Systems of Linear Inequalities", *Operations Research*, vol. 21, 247-260.

P. McKEOWN (1975), "A Vertex Ranking Procedure for Solving the Linear Fixed Charge Problem", *Operations Research*, vol. 23, 1182-1191.

T.S. MOTZKIN, H. RAIFFA, G.L. THOMPSON et R.M. THRALL (1953), "The Double Description Method", Contribution to the Theory of Games II, *Annals of Mathematics Studies*, 51-73.

K.G. MURTY (1968), "Solving the Fixed Charge Problem by Ranking the Extreme Points", *Operations Research*, vol. 16, 268-279.

B.F. SHERMAN (1977), "A Counterexample to Greenberg's Algorithm for Solving Linear Inequalities", *Numerische Mathematik*, vol. 27, 491-492.

N.V. THOAI and H. TUY (1980), "Convergent Algorithm for Minimizing a Concave Function", *Mathematics of Operations Reasearch*, vol. 5, 556-566.

H.D. THUAN (1997), "Can Linear Programs Be Used to Test Global Optimization Algorithms?", *Computing*, vol. 59, 91-93.

T.V. THIEU, B.T. TAM et V.T. BAN (1983), "An Outer Approximation Method for Globally Minimizing a Concave Function Over a Compact Convex Set", *Acta Mathematica Vietnamica*, vol. 8, 21-40.

H. TUY (1964), "Concave Minimization under Linear Constraints", *Soviet Mathematics Doklady Academii Nauk SSSR*, vol. 5, 1437-1440.

H. TUY (1990), "On a Polyhedral Annexation Method for Concave Minimization", *Optimization and Mathematical Economics*, 248-260.

H. TUY (1991), "Normal Conical Algorithm for Concave Minimization Over Polytopes", *Mathematical Programming*, vol. 51, 229-245.

H. UZAWA (1958), "A Theorem on Convex Polyhedral Cones", *Studies in Linear and Non-Linear Programming*, Stanford University Press, 23-31.

H. VAISH and C.M. SHETTY (1976), "The Bilinear Programming Problem", *Naval Research Logistics Quarterly*, vol. 23, 303-309.

P.B. ZWART (1973), "Nonlinear Programming : Counterexamples to two Global Optimization Algorithms", *Operations Research*, vol. 21, 1260-1266.

P.B. ZWART (1974), "Global Maximization of a Convex Function with Linear Inequality Constraints", *Operations Research*, vol. 22, 602-609.