

Titre: Contribution à la résolution des programmes mathématiques avec contraintes d'équilibre et des programmes mathématiques à deux niveaux à contraintes linéaires
Title:

Auteur: Jean Bosco Etoa Etoa
Author:

Date: 2005

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Etoa Etoa, J. B. (2005). Contribution à la résolution des programmes mathématiques avec contraintes d'équilibre et des programmes mathématiques à deux niveaux à contraintes linéaires [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/7545/>
Citation:

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7545/>
PolyPublie URL:

Directeurs de recherche: Gilles Savard
Advisors:

Programme: Non spécifié
Program:

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

iv

This reproduction is the best copy available.

UMI[®]

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

CONTRIBUTION À LA RÉOLUTION DES PROGRAMMES MATHÉMATIQUES AVEC
CONTRAINTES D'ÉQUILIBRE ET DES PROGRAMMES MATHÉMATIQUES À DEUX NIVEAUX
À CONTRAINTES LINÉAIRES

JEAN BOSCO ETOA ETOA
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THESE PRÉSENTÉE EN VUE DE L'OBTENTION DU DIPLÔME
DE PHILOSOPHIAE DOCTOR (Ph.D.)
(MATHÉMATIQUES DE L'INGÉNIEUR)
DÉCEMBRE 2005

©Jean Bosco Etoa Etoa, 2005.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-16993-3
Our file *Notre référence*
ISBN: 978-0-494-16993-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

CONTRIBUTION À LA RÉOLUTION DES PROGRAMMES MATHÉMATIQUES AVEC
CONTRAINTES D'ÉQUILIBRE ET DES PROGRAMMES MATHÉMATIQUES À DEUX NIVEAUX
À CONTRAINTES LINÉAIRES

présentée par : ETOA ETOA Jean Bosco

en vue de l'obtention du grade de: Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. AUDET Charles, Ph.D., président

M. SAVARD Gilles, Ph.D., membre et directeur de recherche

M. MARCOTTE Patrice, Ph.D., membre

M. ANDJIGA Nicolas Gabriel, Docteur d'État, membre externe

DÉDICACES

Je dédie cette thèse à mon père défunt, Joseph Etoa Eboko, qui en 1993, quelques jours avant sa disparition, m'avait posé avec insistance la question de savoir ce que le Doctorat d'État que je lui avais promis était devenu. Cette thèse est une réalisation de cette promesse.

REMERCIEMENTS

Je souhaite remercier en premier lieu mon directeur de recherche, Gilles Savard qui avec amitié et une rigueur intellectuelle louable, ses nombreux encouragements, ses conseils et grâce aux facilités qu'il m'a accordées, m'a permis de réaliser la présente thèse.

Je remercie Walid Zghal, étudiant en doctorat au département de mathématiques et de génie industriel de l'École Polytechnique de Montréal qui a bien voulu mettre à ma disposition quelques problèmes tests afin d'éprouver les performances d'un des algorithmes que j'ai développé dans cette thèse.

Je remercie l'ancien Haut Commissaire pour la République du Cameroun au Canada, le Ministre Philémon Yang, ainsi que le Ministre Jean Marie Atangana Mebara qui m'ont encouragé à me remettre aux études afin de valoriser le potentiel qu'ils avaient décelé en moi.

Je remercie également mon ami Ndongo Mebometa, étudiant aux cycles d'études supérieures dans les universités montréalaises, qui grâce à ses conseils pratiques, a contribué à mon insertion académique dans le système universitaire québécois.

Enfin, mes remerciements les plus sincères sont dédiées à mon épouse Susane qui, avec amour a su m'apporter un appui constant, du soutien et les encouragements nécessaires qui m'ont permis de retrouver une seconde jeunesse pour que soit réalisé ce travail. Que mes enfants, Yannick, Victoire, Laurine, Jean Bosco Jr et Wendy me pardonnent pour l'affection dont ils ont été sevrés pendant toutes ces longues années consacrées à ma thèse; Jean Bosco Jr, Wendy et Carl n'ont ils pas vu le jour pendant ces années ?!!!

RÉSUMÉ

La programmation mathématique avec contraintes d'équilibre (MPEC) d'une manière générale et la programmation mathématique à deux niveaux (PDN) en particulier constituent d'excellents outils pour modéliser certains systèmes de décisions hiérarchisés. Cette thèse contribue à une analyse théorique et au développement d'algorithmes de résolution de ces problèmes.

Dans un premier temps, nous faisons une revue de littérature portant sur les problèmes de PDN. Nous passons notamment en revue, les concepts théoriques liés aux problèmes de PDN ainsi qu'aux principaux algorithmes de résolution de ces problèmes. Le même exercice est fait, pour ce qui a trait aux problèmes de MPEC, en insistant toutefois sur les problèmes de MPEC qui s'apparentent à une formulation du problème de PDN dans laquelle le problème de second niveau est remplacé par ses conditions d'optimalité de Karush, Kuhn et Tucker (KKT). Le problème de MPEC qui en résulte est le problème (PDN_{KKT}).

Dans un second temps, partant des caractéristiques des solutions du problème de MPEC ou du problème défini par une inégalité variationnelle, nous proposons des propriétés portant sur les solutions du problème de PDN qui peuvent faciliter les analyses théoriques liées à la conception des algorithmes de résolution du problème de PDN. Nous considérons le concept de décomposition en branches d'un problème de MPEC du type (PDN_{KKT}) afin d'établir une condition d'optimalité globale pour un problème de PDN. Nous montrons que sous des hypothèses appropriées, le problème (PDN_{KKT}) peut être transformé en une suite de programmes mathématiques localement réguliers; nous établissons les conditions d'optimalité de la classe des problèmes qui en résulte.

Pour terminer, nous développons deux algorithmes basés sur le concept de programmation linéaire séquentielle et nous établissons la propriété de convergence linéaire de

ce type d'algorithme dans un contexte de programmation mathématique classique. Le premier algorithme permet de déterminer un point stationnaire d'un problème de MPEC; il fait appel à la notion de fonction de mérite paramétrique. En fonction des seules données du problème résolu, nous développons une méthode originale d'actualisation du paramètre qui intervient dans la fonction de mérite en question. Le second algorithme original que nous développons permet de résoudre un problème de PDN à contraintes linéaires; c'est une méthode de descente appelée algorithme de programmation linéaire séquentielle par énumération implicite (*ESLP*). Cette procédure est basée sur le concept de programmation linéaire séquentielle: l'énumération porte sur les variables et les contraintes de second niveau de manière à déterminer une face S_T de l'ensemble des solutions réalisables susceptible de contenir une solution admissible qui améliore la valeur de la fonction économique du premier niveau. Les principes de monotonie à partir desquels nous proposons des concepts originaux de graphes ou de cheminement de monotonie, permettent de déterminer à une étape de l'algorithme *ESLP*, le choix de la variable ou de la contrainte de second niveau permettant de déterminer la face S_T . Les tests numériques que nous avons effectués démontrent que nos algorithmes sont prometteurs.

ABSTRACT

Mathematical programming with equilibrium constraints problems (MPEC) as well as bi-level programming problems (BPP) constitute adequate tool for modeling hierarchical decision systems. This thesis contributes to a theoretical and algorithmic development of BPP and MPEC problems.

First, we work out a review of literature carrying on the fundamental concepts associated with BPP and the algorithms for solving these problems. We use the same frame carrying on the MPEC problems; focus is on piecewise formulation of MPEC related to BPP problems, where the lower level problem is formulated in terms of Karush, Kuhn and Tucker conditions, called BPP_{KKT} . Algorithms that we developed in this thesis are based on piecewise formulation tool of BPP_{KKT} problems. Next, we exploit the theory of variational inequalities or MPEC, to discuss some properties of the BPP solutions that may make easy theoretical analyses concerning algorithms for BPP. The problem (BPP_{KKT}) can be transform in to a sequence of smooth regular problems, and under suitable assumptions, some optimality conditions are established.

Finally, using linearization and sequential programming techniques, we introduce new sequential linear programming algorithms (*SLP*) and we deal with the rate of convergence of this algorithm. The first algorithm is a smoothing method based on the reformulation of the complementary condition by mean of a perturbed version of the Fisher-Burmeister function, combined with a new update method of that function; this algorithm computes stationary points of MPEC problems. The second algorithm we introduce is called enumeration sequential linear programming algorithm (*ESLP*). The *ESLP* algorithm uses monotonicity properties of the second level objective function and constraints within an enumeration frame work. Depending on the feasibility of the constraints of the BPP and the improvement of the upper level objective function, some nonzero slaked and second level variables may be constraint to be equal to zero. To find these variables, or in order to find a face of the set of feasible solutions of the linear BPP that

contains an optimal solution, we introduce for the first time monotonicity networks. The *ESLP* computes an optimal solution of a linear BPP and some computational results are reported.

TABLE DES MATIÈRES

DÉDICACES.....	v
REMERCIEMENTS	vi
RÉSUMÉ.....	vii
ABSTRACT	ix
TABLE DES MATIÈRES	xi
LISTE DES FIGURES.....	xiv
LISTE DES TABLEAUX.....	xv
LISTE DES SIGLES ET ABRÉVIATIONS.....	xvi
CHAPITRE 1 - INTRODUCTION	1
CHAPITRE 2 – REVUE DE LA LITTÉRATURE SUR LA PROGRAMMATION MATHÉMATIQUE À DEUX NIVEAUX (PDN).....	9
2.1 Définitions et notations	9
2.2. Propriétés des problèmes de PDN	13
2.2.1 Nature de la complexité du problème de PDN	13
2.2.2 Conditions d’optimalité du problème de PDN	14
2.3 Quelques applications de la programmation mathématique à deux niveaux (PDN)...	21
2.4 Les algorithmes de résolution des problèmes de PDN	22
CHAPITRE 3 – REVUE DE LA LITTÉRATURE SUR LES PROGRAMMES MATHÉMATIQUES AVEC CONTRAINTES D’ÉQUILIBRES (MPEC)	33
3.1 Formulation générale du problème de MPEC: définitions et notations	34
3.2 Le problème de complémentarité.....	37
3.2.1 Définitions.....	38
3.2.2 Propriétés du problème de complémentarité.....	40

3.3 Les fonctions de mérite	41
3.4 Le problème de MPEC sous forme KKT	44
3.4.1 Formulation et définitions	44
3.4.2 Les hypothèses de qualification des contraintes	48
3.4.3 Caractérisation des solutions du problème de MPEC sous forme KKT	50
3.4.4 Propriétés du problème de MPEC sous forme KKT	53
3.5 Contribution à la définition des propriétés des solutions d'un problème de PDN	58
3.6 Les algorithmes de résolution des problèmes de MPEC	65
3.6.1 Les méthodes d'optimisation séquentielle	67
3.6.2 Les méthodes d'optimisation classique	72
3.6.3 Une méthode d'optimisation globale: l'algorithme de Thoai et al. [2005].....	74
CHAPITRE 4 – RÉOLUTION DES PROBLÈMES DE MPEC PAR UN ALGORITHME DE PROGRAMMATION LINÉAIRE SÉQUENTIELLE (SLP)	76
4.1 Cadre conceptuel d'un algorithme de programmation linéaire séquentielle:	77
4.1.1 Généralités	77
4.1.2 Convergence linéaire de l'algorithme SLP	81
4.2 Approximation du problème de PDN sous forme KKT par un programme mathématique localement régulier	94
4.2.1 Méthode d'approximation des contraintes d'équilibre.....	94
4.2.2 Conditions d'optimalité de l'approximation de la formulation KKT d'un problème de PDN	99
4.3 Méthode d'actualisation du paramètre utilisé dans une fonction de mérite	103
4.4 Résolution d'un problème de MPEC ou de complémentarité par l'algorithme SLP .	107
4.4.1 Résolution d'un problème de MPEC	107
4.4.2 Résolution d'un problème de complémentarité	114
4.5. Les expérimentations numériques.....	118
4.5.1 Les problèmes traités (Fukushima et al. [1998]).....	118
4.5.2 Les résultats des expérimentations numériques	120

CHAPITRE 5 – RÉOLUTION DU PROBLÈME DE PDN PAR UN ALGORITHME DE PROGRAMMATION LINÉAIRE SÉQUENTIELLE PAR ÉNUMÉRATION IMPLICITE	126
5.1 Un algorithme de programmation linéaire par pièces (PSLP)	127
5.1.1 Décomposition en branches d'un problème de PDN sous forme KKT	127
5.1.2 L'algorithme de programmation linéaire par pièces	130
5.2 Résolution d'un problème de PLDN par un algorithme de programmation linéaire séquentielle par énumération implicite (ESLP).....	138
5.2.1 Fondements théoriques de l'algorithme ESLP	138
5.2.2 Cadre conceptuel de l'algorithme ESLP	144
5.2.3 L'algorithme de programmation linéaire séquentielle par énumération implicite	162
5.2.4 Expérimentations numériques de l'algorithme ESLP	167
CONCLUSION	182
BIBLIOGRAPHIE	184

LISTE DES FIGURES

Figure 2. 1: Illustration du domaine induit	17
Figure 4. 1: Évolutions de la valeur de μ_k (*) et de $\ \Phi_k\ _2$ (+) en fonction de k (Problème 3*) ..	122
Figure 4. 2: Évolutions de la valeur de μ_k (*) et de $\ \Phi_k\ _2$ (+) en fonction de k (Problème 3**) ..	122
Figure 5. 1: Exemple 5.4 - arborescence de monotonie.....	157
Figure 5. 2: Problème 5.5 – 1ère arborescence de monotonie.	172
Figure 5. 3: Problème 5.5 –arborescences de monotonie du 2 nd graphe de monotonie.	174

LISTE DES TABLEAUX

Tableau 4. 1: Tests numériques - algorithme <i>SLP</i> (problèmes de MPEC).....	121
Tableau 4. 2: Tests numériques - algorithme <i>SLP</i> (problèmes de complémentarité).....	123
Tableau 4. 3: Tests numériques - algorithme de Fukushima et al. (problèmes de MPEC)	124
Tableau 5. 1: Matrice d'adjacence du graphe de monotonie de l'exemple 5.4.....	157
Tableau 5. 2: Matrice d'adjacence du graphe de monotonie initial - problème 5.5.....	172
Tableau 5. 3: Matrice d'adjacence du second graphe de monotonie - problème 5.5.	173
Tableau 5. 4: Algorithme <i>ESLP</i> – Problèmes de petite taille	175
Tableau 5. 5: Algorithme <i>Bard et Moore</i> – Problèmes de petite taille	175
Tableau 5. 6: Algorithme <i>ESLP</i> – Problèmes Audet et al. [1999].....	176
Tableau 5. 7: Algorithme <i>Bard et Moore</i> – Problèmes Audet et al. [1999]	177
Tableau 5. 8: Tests comparatifs - Algorithmes <i>ESLP</i> et <i>CBB</i>	178
Tableau 5. 9: Tests comparatifs – Algorithmes <i>Bard et Moore</i> , <i>ESLP</i> et <i>CBB</i>	179
Tableau 5. 10: Algorithme <i>ESLP</i> - Problèmes Calamai et Vicente [1994].....	180

LISTE DES SIGLES ET ABRÉVIATIONS

CSO: Condition de Second Ordre;

ESLP: Programmation Linéaire Séquentielle par Énumération implicite;

GCP: Problème de Complémentarité Généralisé;

KKT: Karush, Kuhn et Tucker;

LKT: Lagrange, Kuhn et Tucker;

LCP: Problème de Complémentarité Linéaire;

LICQ: hypothèse de Qualification des Contraintes d'Indépendance Linéaire;

MFCQ: hypothèse de Qualification des Contraintes de Mangasarian-Fromovitz;

MPEC: Programme Mathématique avec Contraintes d'Équilibre;

NCP: Problème de Complémentarité Non linéaire;

PCP: Pivot Paramétrique Complémentaire;

PDN: Programmation Mathématique à Deux Niveaux;

PDNL: Programme Mathématique à Deux Niveaux à contraintes Linéaires;

PDNP: Programmation Mathématique à Deux Niveaux sous forme Particulière;

PLDN: Programmation Linéaire à Deux Niveaux;

PM: Programme Mathématique;

PSLP: Programmation Linéaire Séquentielle par Pièce;

PSQP: Programmation Quadratique Séquentielle par Pièce;

PVI: Problème défini par une Inégalité Variationnelle;

S.C.S: Semi Continue Supérieurement;

SBCQ: hypothèse de Qualification des Contraintes Séquentiellement Bornées;

SLP: Programmation Linéaire Séquentielle ;

SQP: Programmation Quadratique Séquentielle;

VI: Inégalité Variationnelle.

CHAPITRE 1 - INTRODUCTION

L'utilisation de la programmation mathématique dans différents processus décisionnels est restée pendant de nombreuses années consacrée aux problèmes pour lesquels un décideur unique (gouvernement, politicien, institution, organisation...) avait un contrôle unilatéral sur le niveau d'activités à assigner à tous les objectifs ou variables de décision. Depuis les années 70, l'introduction de la programmation mathématique à plusieurs niveaux consacre la décentralisation du niveau de prise de décision en tenant compte de la réaction des autres décideurs. Souvent, les décideurs interviennent dans un système hiérarchisé où ils peuvent agir soit de façon coopérative, soit de façon non coopérative. La programmation mathématique à plusieurs niveaux résout le problème de la coordination du processus de prise de décision dans un système décentralisé, notamment lorsque les décideurs agissent de façon non coopérative et que les décisions sont prises de manière séquentielle.

Motivés par la théorie des jeux de Stackelberg [1952], plusieurs auteurs se sont intéressés à la programmation mathématique à deux niveaux. Stackelberg a introduit un modèle en concurrence parfaite dans le contexte d'un duopole. Ce modèle est caractérisé par l'existence d'une précedence dans l'annonce des décisions de deux firmes: un producteur se voit assigner le rôle de meneur (premier niveau de décision) et annonce son niveau de production au suiveur (second niveau de décision) dans la détermination de sa propre production. Le paradigme de Stackelberg fait l'hypothèse que le meneur agit en connaissant la réaction optimale du suiveur à son annonce.

Anandalingam et Friez [1992], Ben-Ayed [1993] puis, Vicente et Calamai [1994] ou Dempe [2003] présentent dans des revues bibliographiques, les applications de la programmation mathématique à deux niveaux (PDN), les propriétés théoriques liées à la PDN et les différentes classes d'algorithmes de résolution. Migdalas et Pardalos [1996] procèdent à une revue de

littérature en compilant sept articles qui couvrent les différents aspects de la PDN. La revue de bibliographie la plus significative est contenue dans une monographie de Bard [1998]; cet ouvrage présente une excellente introduction à la théorie de la PDN. Il en est de même de la monographie de Dempe [2002]. Les toutes premières formulations liées au problème de PDN sont l'œuvre d'auteurs tels que Kornaj et Liptack [1965], Ross [1973], Simaan et Cruz [1973] ou Braken et McGill [1973]; toutefois, Candler et Norton [1977a], [1977b] furent les premiers à utiliser la terminologie programmation à deux niveaux ou à plusieurs niveaux dans un rapport de la Banque Mondiale.

La formulation générale d'un programme mathématique à deux niveaux est:

(PDN)

$$\begin{aligned} & \text{Minimiser}_{x,y} f^1(x,y) \\ & \left\{ \begin{array}{l} g^1(x,y) \leq 0, \\ x \in X \subset \mathbb{R}^{n_x}, \end{array} \right. \\ & \text{s.c.} \left\{ \begin{array}{l} \text{Minimiser}_y f^2(x,y) \\ \left\{ \begin{array}{l} g^2(x,y) \leq 0, \\ y \in Y \subset \mathbb{R}^{n_y}, \end{array} \right. \end{array} \right. \end{array} \quad (1.1.1)$$

où $f^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $g^i = (g_1^i, g_2^i, \dots, g_{l_i}^i) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{l_i}$, $i = 1, 2$.

La fonction f^1 et les contraintes $g^1(x,y) \leq 0$, $x \in X$ (resp. la fonction f^2 et les contraintes $g^2(x,y) \leq 0$, $y \in Y$) définissent le problème du premier niveau (resp. le problème de second niveau (PM2)). Pour un vecteur x donné, le vecteur y peut être défini de façon implicite à partir du problème de second niveau par la relation:

$$\begin{aligned} & y \in \operatorname{argmin}_{\xi} f^2(x, \xi) \\ & \text{s.c.} \left\{ \begin{array}{l} g^2(x, \xi) \leq 0, \\ \xi \in Y. \end{array} \right. \end{array} \quad (1.1.2)$$

Cette formulation modélise la coopération entre les deux niveaux dans le cas où le second niveau est indifférent. Les relations définies par (1.1.2) impliquent que le vecteur y est déterminé après que le vecteur x ait été fixé. C'est ce qui explique la désignation du vecteur x comme vecteur de premier niveau et le vecteur y comme vecteur du second niveau. Un cadre d'application économique peut être lié à l'étude d'impacts des décisions économiques ou politiques: le premier niveau représente les décideurs publics et le second niveau simule la réaction d'un marché spécifique. Par exemple, un gouvernement peut s'intéresser à l'impact sur ses revenus de l'introduction d'une nouvelle taxe de vente en fonction d'un ajustement optimal du marché à cette distorsion. On peut trouver des exemples d'application de la PDN dans Candler et Norton [1977a] (marché laitier hollandais), Candler et Norton [1977b] (marché agricole mexicain) et Savard [1989] (cogénération électrique aux Etats-Unis). Abdou-Kandil et Bertrand [1987] ou encore Leleno et Sherali [1992] décrivent des applications de la PDN en économie, comme l'analyse des réseaux d'oligopoles.

Lorsque les fonctions f^1 et f^2 sont linéaires, les fonctions g^1 et g^2 sont affines puis $X \subseteq \mathbb{R}^{n_x}$ et $Y \subseteq \mathbb{R}^{n_y}$ des polyèdres, on a un programme linéaire à deux niveaux (PLDN). De nombreux travaux de recherche ont été consacrés au cas linéaire des PDN et la grande majorité des résultats ont été obtenus pour la version particulière de programmes mathématiques à deux niveaux linéaires ne contenant pas de contraintes dites couplantes de premier niveau $g^1(x, y) \leq 0$.

Dans le cas général d'un PDN, moyennant des hypothèses de régularité sur le problème de second niveau, on peut remplacer ce dernier par ses conditions d'optimalité de Karush, Kuhn et Tucker (KKT); on obtient alors la formulation KKT du problème de PDN qui en fait, est un cas particulier d'un programme mathématique avec contraintes d'équilibre (MPEC) (mathematical program with equilibrium constraints). La formulation générale d'un MPEC est la suivante:

$$\begin{aligned} & \text{Minimiser}_{x,y} f(x,y) \\ & \text{s.c.} \begin{cases} H(x,y) = 0, \\ y \geq 0, \end{cases} \end{aligned} \tag{1.1.3}$$

où $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ et $H : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^l$. L'équation $H(x,y) = 0$ représente un problème d'équilibre incluant des équations du type $F(x)^t y = 0$ avec $F(x) \in \mathbb{R}^{n_y}$. La détermination d'une solution réalisable pour un programme mathématique avec contraintes d'équilibre est un problème difficile et de nombreux travaux ont été consacrés à la résolution de ce problème (Chen et Harker, 1993 ; Kanzow, 1996; Fukushima et al., 1998; Andreani et Martinez, 1998; Xu et Burke, 1999; Engelke et Kanzow, 2002).

Comme le souligne Moré [1996], les programmes mathématiques avec contraintes d'équilibre (MPEC) proviennent de la recherche d'une solution d'équilibre en économie. Les travaux de Ahn [1979] sur le système d'évaluation de l'indépendance des projets sont à l'origine de l'intérêt porté à la formulation et la recherche des solutions des problèmes d'équilibre économique comme des problèmes de complémentarité. Ferris et Kanzow [1998] présentent des exemples liés à la dérégulation des marchés de l'électricité en Amérique du Nord qui sont devenus populaires, ainsi que le problème classique d'équilibre d'Arrow et Debreu. Les revues bibliographiques de Harker et Pang [1990], Ferris et Pang [1997] ou de Ferris et Kanzow [1998] font le point sur les derniers développements des travaux consacrés aux problèmes de complémentarité avec leurs principales applications. Dempe [2003] a publié une revue bibliographique qui fait le point sur la programmation mathématique à deux niveaux et les programmes mathématiques avec contraintes d'équilibre. S'appuyant sur un modèle de péage dans un réseau de transport, Colson et al. [2005] proposent une revue de littérature portant à la fois sur les problèmes de PDN et les problèmes de MPEC. Un exemple d'application du problème de MPEC en mécanique des structures peut être trouvé dans Luo et al. [1996a]. Outrata et al. [1998] présentent des applications importantes des problèmes de MPEC en ingénierie. Schumacher [2004] illustre l'utilisation des problèmes de complémentarité en programmation mathématique avec des applications en ingénierie.

Sous une hypothèse de régularité, tout problème de PDN est équivalent à un problème de MPEC; Ye [1999] a d'ailleurs reconnu que ces deux problèmes partagent des caractéristiques communes. Luo et al. [1996] en introduisant une hypothèse de qualification des contraintes relative au concept de multiplicateurs de KKT séquentiellement bornés, en ont déduit une série de conditions d'optimalité relatives au problème de MPEC. Marcotte et Zhu [1996], Ye et Ye [1997] puis Marcotte et al. [2001] ont défini le problème de MPEC comme un problème de PDN généralisé: les contraintes du problème de MPEC en question incluent une inégalité variationnelle. Outrata [1995], Ye et Ye [1997], Ye [1999] ont fait appel à l'analyse non différentiable de Clarke [1983] pour établir des conditions d'optimalité du problème de PDN généralisé. Pour établir des conditions suffisantes d'optimalité du problème de MPEC, Hu et Ralph [2004] associent une méthode de pénalité à une technique de transformation des contraintes d'équilibre. Considérant les problèmes de MPEC qui comportent parmi leurs contraintes des contraintes d'équilibre ou des problèmes de complémentarité, Kočvara et Outrata [2004] procèdent à une revue de littérature sur la question, puis proposent un nouveau cadre d'analyse pour ce type de problème de MPEC. Le concept de qualification des contraintes du type Abadie introduit par Flegel et Kanzow [2005] débouche sur des propriétés des problèmes de MPEC, précisément sur des nouvelles conditions nécessaires d'optimalité d'un problème de MPEC.

Andreani et Martinez [2001] font remarquer qu'une méthode de résolution des programmes mathématiques ordinaires peut être utilisée pour résoudre un problème de MPEC, à condition de procéder à une bonne reformulation de ses contraintes. La plupart des algorithmes qui ont été développés pour trouver une solution réalisable aux problèmes de complémentarité font appel à une transformation analytique: les contraintes d'équilibre qui rendent difficile la résolution du problème initial sont transformées en contraintes régulières par le biais de fonctions de mérite (Harker et Pang, 1988; Andreani et Martinez, 1998; Burke et Xu, 2000; Hotta et Yoshise, 1999). D'après Fisher et Jiang [2000], les fonctions de mérite (merit functions) sont devenues d'importants outils utilisés pour résoudre des problèmes variés (en optimisation, en

ingénierie, dans les systèmes économiques). Les fonctions de mérite jouent un rôle important dans l'analyse des propriétés théoriques de ces problèmes et dans la conception de leurs algorithmes de résolution: Facchinei et al. [1999], en procédant à une reformulation d'un problème de MPEC par l'intermédiaire d'une fonction de mérite, ont établi des conditions d'optimalité des MPEC.

Les publications consacrées à la conception des algorithmes de résolution des MPEC n'ont pas connu la même évolution, en nombre, que celles consacrées aux propriétés des MPEC. La difficulté de construire un algorithme de résolution d'un problème de MPEC est liée à sa structure particulière. Quelques algorithmes performants ont été proposés pour trouver un point stationnaire pour un problème de MPEC (Luo et al., 1996b; Luo et al., 1998; Jiang et Ralph, 2000; Facchinei et al., 1999; Fukushima et al., 1998; Zhang et Liu, 2001). Sous des hypothèses de qualification des contraintes moins restrictives que celles habituellement utilisées, Liu et Sun [2004] proposent un algorithme globalement convergent de résolution des problèmes de MPEC. Cet algorithme s'appuie sur une méthode primale-duale de point intérieur, combinée à l'utilisation d'une fonction de mérite. Pour résoudre les problèmes de MPEC à variables mixtes, Figueiredo et al. [2005] proposent un algorithme basé sur une exploitation des contraintes de complémentarité par un algorithme du gradient projeté. Pour résoudre des problèmes de MPEC à contraintes linéaires, Thoai et al. [2005] quant à eux proposent un algorithme qui s'inspire des méthodes d'optimisation globale. Ils mettent en exergue les relations qui existent entre les différents critères d'optimalité pour un problème de MPEC.

Nous présentons dans cette thèse une nouvelle approche de résolution du problème de programmation mathématique avec contraintes d'équilibre (MPEC), que nous adaptions à la résolution des programmes mathématiques à deux niveaux à contraintes linéaires (PDNL), moyennant une exploitation judicieuse des contraintes d'équilibre à l'aide des principes de monotonie. L'algorithme de programmation linéaire séquentielle par énumération implicite (*ESLP*) qui en résulte s'avère efficace et simple de mise en œuvre; il permet de déterminer un optimum local (global la plus part du temps) d'un problème de programmation linéaire à deux niveaux

(PLDN). Sur un plan théorique, nous exploitons les propriétés des solutions du problème de MPEC ou du problème défini par une inégalité variationnelle afin de proposer une caractérisation analytique de l'ensemble des solutions du problème de PDN, susceptible de clarifier certaines analyses théoriques liées aux algorithmes de résolution de ce problème. Nous montrons que sous des hypothèses de régularité appropriées, la formulation KKT d'un problème de PDN peut être transformée en une suite de programmes mathématiques localement réguliers; nous établissons des conditions suffisantes d'optimalité du problème de PDN à partir des problèmes qui en résultent.

Le plan de ce travail est organisé comme suit: le second chapitre est consacré à une synthèse bibliographique de la programmation mathématique à deux niveaux. Nous énonçons les principales propriétés, puis nous passons en revue les principaux algorithmes de résolution. Tout comme au second chapitre, nous présentons dans le troisième chapitre une synthèse bibliographique des programmes mathématiques avec contraintes d'équilibre en insistant sur le rôle joué par le concept des fonctions de mérite qui sont devenues d'importants outils pour la résolution des problèmes de complémentarité ou des problèmes de MPEC. Nous présentons une première contribution qui a trait à la compréhension de quelques propriétés relatives à l'ensemble des solutions d'un problème de PDN; ces propriétés sont généralement utilisées dans les analyses théoriques des algorithmes de résolution de ce problème. Dans le quatrième chapitre, nous présentons une seconde contribution à la résolution des programmes mathématiques avec contraintes d'équilibre, lorsque les contraintes du MPEC en question constituent un problème de complémentarité généralisé linéaire. Nous proposons un algorithme de programmation linéaire séquentielle (*SLP*) susceptible de déterminer un point stationnaire d'un problème de MPEC. Les contraintes d'équilibre sont approximées par une fonction de mérite paramétrique de Fisher-Burmeister; en fonction des seules données du problème résolu, nous proposons une méthode originale d'actualisation du paramètre qui caractérise ladite fonction de mérite. Nous établissons la propriété de convergence linéaire de l'algorithme *SLP*, en mettant en exergue une erreur commise jadis dans l'établissement d'un résultat similaire. Nous terminons ce chapitre par une description des résultats de quelques expérimentations numériques de l'algorithme *SLP* appliqué

à la résolution des problèmes de complémentarité généralisé et des problèmes de MPEC. Le cinquième chapitre est consacré à la présentation d'une contribution originale pour la résolution des problèmes de programmation mathématique linéaire à deux niveaux; nous y décrivons un algorithme de programmation linéaire séquentielle par énumération: l'algorithme *ESLP* (Enumeration Sequential Linear Programming) génère au cours de chaque étape une direction de descente en exploitant de manière judicieuse les indices des variables ou des contraintes du problème de second niveau, à l'aide des principes de monotonie desquels nous proposons des concepts originaux de graphes ou de cheminement de monotonie. Le chapitre se termine avec une description des expérimentations numériques liées à la mise en œuvre cet algorithme.

CHAPITRE 2 – REVUE DE LA LITTÉRATURE SUR LA PROGRAMMATION MATHÉMATIQUE À DEUX NIVEAUX (PDN)

Ce chapitre est consacré à la présentation d'une synthèse bibliographique de la programmation mathématique à deux niveaux. Nous énonçons les principales propriétés, puis nous passons en revue les principaux algorithmes de résolution. Le chapitre est organisé comme suit. La section 2.1 présente les notations et définitions d'un programme mathématique à deux niveaux (PDN). La section 2.2 est consacrée à la description des propriétés liées aux concepts des PDN sur la base de la littérature actuelle. Nous faisons une illustration de certaines de ces propriétés avec deux exemples et à la section 2.3, nous passons en revue quelques applications de la PDN. Enfin, dans la section 2.4, nous décrivons les principales classes d'algorithmes de résolution des PND.

2.1 Définitions et notations

Nous rappelons que la formulation générale d'un problème de PDN est:

$$\begin{array}{l}
 \text{(PDN)} \\
 \text{Minimiser}_{x,y} f^1(x,y) \\
 \left\{ \begin{array}{l} g^1(x,y) \leq 0, \\ x \in X \subset \mathbb{R}^{n_x}, \end{array} \right. \\
 \text{s.c.} \left\{ \begin{array}{l} \text{Minimiser}_y f^2(x,y) \\ \text{s.c.} \left\{ \begin{array}{l} g^2(x,y) \leq 0, \\ y \in Y \subset \mathbb{R}^{n_y}, \end{array} \right. \end{array} \right.
 \end{array} \quad (2.1.1)$$

où $f^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $g^i = (g_1^i, g_2^i, \dots, g_{m_i}^i) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{m_i}$, $i = 1, 2$.

La présence des contraintes couplantes de premier niveau $g^1(x,y) \leq 0$ confère à cette formulation un caractère plus général, qui est souvent nécessaire à une modélisation réaliste. En l'absence de ces contraintes, on obtient une formulation particulière du problème de PDN notée

PDNP qui a été utilisée par la plupart des auteurs (Bard, 1983; Bard, 1984; Bialas et Karwan, 1982; Candler et Townsley, 1982). Savard [1989] illustre par un exemple la différence importante qui existe entre les deux formulations.

Dans le cas général d'un problème de PDN, moyennant des hypothèses de régularité sur le problème de second niveau, on peut remplacer ce dernier par ses conditions d'optimalité de Karush, Kuhn et Tucker (KKT). Il en résulte un programme mathématique non linéaire à un niveau dont la formulation est:

(PDN_{KKT})

$$\begin{aligned} & \text{Minimiser}_{x,y} f^1(x,y) \\ & \text{s.c.} \begin{cases} g^1(x,y) \leq 0, \\ g^2(x,y) \leq 0, \\ \nabla_y f^2(x,y) + \lambda^t \nabla_y g^2(x,y) = 0, \\ \lambda^t g^2(x,y) = 0, \\ \lambda \geq 0, (x,y) \in X \times Y. \end{cases} \end{aligned} \quad (2.1.2)$$

Le problème de programmation linéaire à deux niveaux ($PLDN$) (sous forme de maximisation) possède les propriétés les plus riches; sa formulation est la suivante:

($PLDN$)

$$\begin{aligned} & \text{Maximiser}_{x,y} c_1^t x + d_1^t y \\ & \text{s.c.} \begin{cases} A_1 x + B_1 y \leq b_1, \\ x \geq 0, \\ \text{Maximiser}_y d_2^t y \\ \text{s.c.} \begin{cases} A_2 x + B_2 y \leq b_2, \\ y \geq 0, \end{cases} \end{cases} \end{aligned} \quad (2.1.3)$$

où $x \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^{n_y}$, $c_1 \in \mathbb{R}^{n_x}$; $d_1 \in \mathbb{R}^{n_y}$, $b_i \in \mathbb{R}^{m_i}$, $A_i \in \mathbb{R}^{m_i} \times \mathbb{R}^{n_x}$, $B_i \in \mathbb{R}^{m_i} \times \mathbb{R}^{n_y}$, $i = 1, 2$.

À partir de la formulation (PLDN), on peut remplacer le problème de second niveau par ses conditions d'optimalité qui relèvent de la théorie de la dualité en programmation linéaire pour obtenir le problème LKT ci-après:

(LKT)

$$\begin{aligned} & \text{Maximiser}_{x,y} \quad c_1^t x + d_1^t y \\ & \text{s.c.} \quad \begin{cases} A_1 x + B_1 y \leq b_1, \\ A_2 x + B_2 y \leq b_2, \\ u^t (b_2 - A_2 x - B_2 y) = 0, \\ u^t B_2 - \mu = d_2, \\ \mu^t y = 0, \\ x, y, u, \mu \geq 0. \end{cases} \end{aligned} \quad (2.1.4)$$

Nous présentons à présent les principales définitions relatives aux concepts de la programmation mathématique à deux niveaux. Ces définitions font référence à la formulation générale du problème de PDN que l'on pourrait facilement adapter au cas linéaire.

Définitions 2.1

- a) L'ensemble $\Omega = \{(x, y) \in X \times Y : g^i(x, y) \leq 0, i = 1, 2\}$ définit l'**ensemble des solutions réalisables** du problème (PDN).
- b) Pour chaque valeur de x du premier niveau, les contraintes de second niveau permettent de définir le **domaine réalisable du problème de second niveau**:

$$Y(x) = \{y \in Y : g^2(x, y) \leq 0\}.$$

- c) L'ensemble $M(x)$ des solutions optimales de second niveau est:

$$M(x) = \arg \min_{\xi \in Y(x)} f^2(x, \xi)$$

- d) La trace de Ω sur l'ensemble des variables de premier niveau est

$$S(x) = \{x \in X : Y(x) \neq \emptyset\}.$$

- e) La fonction valeur optimale du problème de second niveau pour $x \in S(x)$ fixé est

$$V(x) = f^2(x, y), y \in M(x).$$

Dans le cas linéaire, $V(x)$ est une fonction polyédrale en ce sens que le graphe de V ,

$$\Gamma_V = \{(x, y) \in X \times Y : x \in S(x), f^2(x, y) = V(x)\}$$

est l'union d'un nombre fini de polyèdres.

f) L'ensemble réalisable: $Df = \{(x, y) \in \Omega : y \in M(x)\}$ est appelé **domaine induit**.

Avec ces différentes définitions, il est à présent possible de définir un concept de solution optimale du problème de PND. Soit $(x, y) \in X \times Y$.

Définition 2.2: (x, y) est un **point réalisable** si $(x, y) \in \Omega$.

Définition 2.3: (x, y) est un **point admissible** si (x, y) est un point réalisable et $y \in M(x)$.

Définition 2.4: Un point $(x^*, y^*) \in X \times Y$ est une **solution optimale** de PND si (x^*, y^*) est admissible et pour tout point admissible (x, y) on a: $f^1(x^*, y^*) \leq f^1(x, y)$.

Sous l'hypothèse d'univocité de l'application $M(x)$ et celle de compacité du domaine réalisable, le problème à deux niveaux se reformule comme un problème à un seul niveau en x :

$$\begin{aligned} & \text{Minimiser}_x f^1(x, y(x)) \\ & \text{s.c.} \begin{cases} x \in \Omega_x^2, \\ g^1(x, y(x)) \leq 0, \end{cases} \end{aligned}$$

où $\Omega_x^2 = \{x \in \mathbb{R}^{n_x} / \exists y \in \mathbb{R}^{n_y} \text{ avec } g^2(x, y) \leq 0\}$ désigne la trace de l'ensemble Ω des variables de premier niveau x . Lorsque l'application $M(x)$ n'est pas univoque, on obtient la formulation:

$$\begin{aligned} & \text{Minimiser}_x f^1(x, y) \\ & \text{s.c.} \begin{cases} (x, y) \in \Omega, \\ y \in M(x). \end{cases} \end{aligned} \tag{2.1.5}$$

Les algorithmes de descente présentés dans la section 2.3, utilisent cette formulation et ne convergent pas toujours vers un optimum global puisque l'ensemble $M(x)$ n'est pas convexe. La résolution de ces problèmes est compliquée en raison de la présence des fonctions et/ou des ensembles définis implicitement.

2.2. Propriétés des problèmes de PDN

Dans cette section, nous énonçons les principales propriétés des problèmes de PDN et de PLDN sous leur forme générale. L'exploitation de ces propriétés dans certains cas, permet de reformuler le problème de PDN en un problème à un seul niveau, puis de donner une caractéristique d'une solution optimale dans le cas du problème de PLDN.

2.2.1 Nature de la complexité du problème de PDN

On dit qu'une classe de problèmes est NP-difficile (resp. fortement NP-difficile) dans le cas où il ne peut exister un algorithme de résolution pour un problème de cette classe en temps borné par une fonction polynomiale de la taille du problème (resp. fonction polynomiale de la taille et de la valeur maximale des données du problème), que si tous les problèmes de cette classe peuvent être résolus en temps polynomial.

Tout comme Jeroslow [1985], Ben-Ayed et Blair [1990] ont établi que le problème de PLDN est un problème NP-difficile; pour le démontrer, Ben-Ayed et Blair [1990] utilisent le fait que le problème du sac à dos en variables binaires (dont la résolution est NP-difficile) peut être reformulé comme un problème de PLDN. Hansen et al. [1992] ont montré que ce résultat pouvait être renforcé en considérant le problème de max-min linéaire qui est un cas particulier du problème de PDN; Audet et al. [1997] ont ensuite montré que les problèmes bien connus en optimisation combinatoire, comme le modèle général de programmation quadratique concave à contraintes linéaires, ou alors des instances de programmation à variables mixtes peuvent être réduits de façon polynomiale en problèmes de PLDN. Le chapitre 1 de la monographie de

Marcotte et Savard [2005] est consacrée à l'étude de la complexité des PDN. On a donc le résultat suivant:

Propriété 2.1 (corollaire 3.2, Hansen et al. 1992): Le problème de PDN est fortement NP-difficile.

Vicente et al. [1994] démontrent que le problème qui consiste à établir qu'une solution satisfait aux conditions d'optimalité locale pour un PLDN est fortement NP-difficile. Pour résoudre un problème de PDN non linéaire, Savard et Gauvin [1990] ont proposé, (1) des conditions nécessaires d'optimalité et, (2) ont montré qu'une direction de descente de plus forte pente est obtenue en résolvant un programme quadratique à deux niveaux. Au pire des cas, leur méthode appliquée à un PLDN a une complexité exponentielle même si l'ensemble des solutions réalisables est non dégénéré.

2.2.2 Conditions d'optimalité du problème de PDN

Depuis 1980, des efforts significatifs ont été consacrés à la compréhension des concepts associés à la programmation à deux niveaux. En reformulant le problème PLDN particulier comme un problème de programmation semi infinie, Bard [1983a], [1984] a proposé des conditions d'optimalité au problème PLDN. Savard [1989] a montré que ces conditions n'étaient pas valables de sorte que les algorithmes de résolution qui prennent appui sur les dites conditions d'optimalité ne sont pas convergents; Clark et Westerberg [1988] puis, Haurie et al. [1990] aboutissent aux mêmes conclusions. Pour la formulation particulière du problème de PDN, lorsque l'ensemble des solutions optimales du problème de second niveau est un singleton, ($M(x) = \{y(x)\}$), Dempe [1992] et Outrata [1993] ont proposé des conditions d'optimalité en considérant le problème d'optimisation:

$$\text{Minimiser}_{x \in X} f^1(x, y(x)).$$

Seulement, cette approche suppose que l'application $x \rightarrow M(x)$ est univoque et que la fonction $x \rightarrow y(x)$ est différentiable presque partout. Par ailleurs, moyennant des conditions de régularité

assez restrictive sur le problème de PDN, Dempe [1992] a utilisé le concept de différentielle dans une direction de la fonction $x \rightarrow y(x)$; cette différentielle est obtenue en résolvant un programme quadratique. L'auteur en déduit des conditions nécessaires et suffisantes d'optimalité qui sont à notre avis, laborieuse à établir en pratique. Faisant appel à l'optimisation non différentiable, Ye et Zhu [1992] proposent des conditions d'optimalité pour la formulation particulière du problème de PDN. Dans la perspective de procéder à une analyse théorique des propriétés des PLDN par une méthode des pénalités, Xu [1999] établit (i) des conditions d'existence non seulement d'une solution, mais aussi d'une fonction de pénalité exacte, puis (ii) la convergence vers une solution optimale d'un algorithme basé sur cette méthode des pénalités. Moyennant des hypothèses de régularité sur la formulation KKT d'un problème de PDN, Liu et al. [2001] proposent une fonction de pénalité exacte pour le problème de PDN, puis établissent des conditions sous lesquelles les problèmes de PDN et sa formulation KKT ont même ensemble de solutions.

2.2.2.1 Cas particulier du problème de PLDN

La grande majorité des résultats ont été obtenus pour la version particulière du problème de PLDN ne contenant pas des contraintes couplantes de premier niveau. Pour cette version particulière, Candler et Townsley [1982], Bard et Falk [1982], Bialas et Karwan [1982] ont reconnu la non convexité du problème de PLDN et l'existence d'une solution optimale en un point extrême du domaine réalisable Ω ; Bialas et Karwan [1984] démontrent ce résultat en faisant l'hypothèse que Ω est borné. Savard [1989] a montré que ce résultat reste valable en présence des contraintes couplantes de premier niveau. Pour le problème de PLDN, bien que le domaine induit soit non convexe en général, il possède certaines propriétés des ensembles convexes (Bard, 1983a). Les plus importantes étant les suivantes:

Propriété 2.2 (Corollaire 3.1, Bard 1983a): Tout point extrême du domaine induit Df est aussi un point extrême du domaine réalisable Ω du problème de PLDN.

Propriété 2.3 (Falk, 1973; Bard, 1988: corollaire 1): Lorsqu'un problème de PLDN admet une solution optimale, cette solution est atteinte en un point extrême du domaine induit DI .

Calvete et Galé [1998] ont généralisé ce résultat lorsque les fonctions économiques du premier et du second niveau sous la forme d'une minimisation sont quasi concaves. Comme nous le verrons dans la section 2.3, plusieurs algorithmes de résolution utilisent ces résultats. L'application $x \rightarrow M(x)$ est une application polyédrale (Savard, 1989: théorème 2.5) et son graphe représentant le domaine induit est une réunion finie de polyèdres qui peut être non convexe, ou non connexe. Les exemples suivants illustrent les définitions précédentes.

Exemple 2.1 (Savard et Gauvin 1994): Considérons le problème de PDN dont la formulation est:

$$\begin{array}{l} \text{Minimiser}_{x,y} (x-1)^2 + 2y_1^2 - 2x \\ \text{s.c.} \left\{ \begin{array}{l} \text{Minimiser}_y (2y_1 - x)^2 + (2y_2 - 1)^2 + xy_1 \\ \left\{ \begin{array}{l} 4x + 5y_1 + 4y_2 \leq 12, \\ 4x + 5y_1 - 4y_2 \geq 4, \\ 4x - 4y_1 + 5y_2 \leq 4, \\ -4x + 4y_1 + 5y_2 \leq 4, \\ x, y_1, y_2 \geq 0. \end{array} \right. \end{array} \right. \end{array}$$

La solution optimale se trouve sur le point extrême P de coordonnées $(17/9, 8/9, 0)$ (figure 2.1). Le domaine réalisable Ω est délimité par la frontière de la figure 2.1, alors que le domaine induit est représenté par les segments en gras n'est pas convexe.

$$M(x) = \begin{cases} 5 - x/2 & \text{si } x \in [0,8], \\ -3 + x/2 & \text{si } x \in [8,12], \\ -21 + 2x & \text{si } x \in [12,16]. \end{cases}$$

Le domaine induit D_I , qui est égal à l'ensemble défini par

$$\{(x, 5 - x/2) : x \in [0,8]\} \cup \{(x, -3 + x/2) : x \in [8,12]\} \cup \{(x, -21 + x) : x \in [12,16]\},$$

est constitué de segments non qui ne sont pas adjacents dans \mathbb{R}^2 . Il n'est par conséquent non connexe.

2.2.2.2 Cas général du problème de PDN

Des études significatives ont été consacrées à l'étude des conditions d'optimalité du modèle général du problème de PDN par des auteurs comme Bard [1983a], Luo et al. [1998], Liu et al. [2001]. Chen et Florian [1995] ou encore Ye et Zhu [1995], [1997] utilisent les éléments du calcul non différentiable de Clarke [1983] pour établir des conditions d'optimalité du problème de PDN. Étant donné un problème de PDN fractionnaire (chacune des fonctions économiques se met sous la forme d'un rapport de fonctions linéaires), Calvete et Galé [2004] montrent que l'optimum d'un tel PDN est atteint sur une face du domaine induit.

Nous passons en revue les résultats qui de notre point de vue sont les plus significatifs.

Pour énoncer les conditions d'optimalité de premier ordre pour le problème de PDN, on suppose généralement que lorsque la variable de premier niveau x est fixée, l'on a:

- a) une hypothèse de qualification des contraintes est satisfaite pour le problème de second niveau (PM2) dans un voisinage d'une solution optimale (x, y^*) ;
- b) (Falk et Liu, 1995): les relations de complémentarité sont non dégénérées $(\lambda_i + g_i^2(x, y^*) \neq 0 \forall i)$ et les conditions d'optimalité de second ordre sont satisfaites au voisinage de y^* pour le problème de second niveau:

$$d^t \nabla^2 L^0(x, y^*, \lambda^*) d > 0 \text{ pour tout } d \neq 0 \text{ tel que } d \in T((x, y^*), \Omega) \text{ et } \lambda_i^* > 0 \forall i,$$

où $L^0(x, y, \lambda) = f^2(x, y) + \lambda^t g^2(x, y)$ est le lagrangien du problème de second niveau et $T((x, y^*), \Omega)$, le sous-espace tangent en (x, y^*) à Ω pour x fixé.

Chen et Florian [1995] proposent d'utiliser la fonction valeur pour reformuler le problème (*PDN*) comme un programme mathématique à un niveau:

$$(PMF) \quad \begin{array}{l} \text{Minimiser}_{x,y} f^1(x, y) \\ \text{s.c.} \left\{ \begin{array}{l} g^1(x, y) \leq 0, \\ g^2(x, y) \leq 0, \\ f^2(x, y) - V(x) = 0, \\ (x, y) \in X \times Y. \end{array} \right. \end{array} \quad (2.2.1)$$

La contrainte $f^2(x, y) - V(x) = 0$ signifie que $(x, y) \in DI$ et que les contraintes du problème (*PMF*) décrivent le domaine induit *DI*. Cette formulation pourrait laisser croire que le premier niveau d'optimisation permet de contrôler totalement la valeur y du second niveau d'optimisation, contrairement à la formulation initiale de *PDN*. Chen et Florian [1995] ont montré que, lorsque le problème (*PDN*) a une solution, alors (x^*, y^*) est une solution optimale de (*PDN*) si et seulement si (x^*, y^*) est une solution optimale du problème (*PMF*); de plus, lorsque le problème (*PND*) est convexe, alors les formulations (PDN_{KKT}) et (*PMF*) ont même ensemble de solutions. C'est l'objet de la propriété ci-après.

Propriété 2.4 (Chen et Florian, 1995: lemme 2.2): Pour une valeur donnée de la variable de premier niveau x du problème de *PDN*, si les fonctions $f^i(x, \cdot)$ et $g^i(x, \cdot)$ $i = 1, 2$ sont convexes en y , alors la formulation (*PMF*) est équivalente à la formulation (PDN_{KKT}) du problème de *PDN*.

De la formulation (*PMF*), comme la fonction $V(x)$ est non différentiable et non convexe, il est possible de déduire des propriétés similaires pour le problème (*PDN*); il en est de même pour

les propriétés géométriques du domaine induit du problème de PDN. À l'instar de tout problème d'optimisation, pour énoncer les conditions de régularité du problème (*PMF*) en un point $z = (x, y)$, il faudrait déterminer son ensemble de directions admissibles, puis donner une approximation linéaire de l'ensemble des solutions réalisables Ω en z , et enfin énoncer les conditions nécessaires et suffisantes de qualification des contraintes de Kuhn et Tucker. Chen et Florian [1995] procèdent de cette façon pour énoncer les conditions de régularité du problème (*PDN*): à partir du lagrangien du problème (*PMF*), ces auteurs déduisent les conditions d'optimalité du problème (*PDN*) en termes d'existence des multiplicateurs de Kuhn et Tucker relatifs au problème (*PMF*).

Tuy et al. [1993], remplacent la contrainte $f^2(x, y) - V(x) = 0$ par $f^2(x, y) \leq V(x)$ puis font appel aux concepts de l'optimisation globale pour déterminer une approximation de la solution optimale de PLDN en utilisant des coupes de concavité.

Sous des hypothèses de qualification des contraintes appropriées, un problème de PDN est partiellement équivalent à sa formulation de KKT, à savoir (PDN_{KKT}). La propriété ci-après énonce ce résultat.

Propriété 2.5 (Bard, 1983): Si (x^*, y^*) est une solution optimale du problème (*PDN*) et si y^* est une solution optimale régulière du programme de second niveau pour x fixé, alors il existe un vecteur de multiplicateurs de KKT λ^* tel que (x^*, y^*, λ^*) est une solution optimale du problème (PDN_{KKT}).

Nous désignons par \mathfrak{F}^{KKT} l'ensemble des solutions du problème (PDN_{KKT}) alors que l'ensemble des multiplicateurs de KKT du problème de second niveau d'un problème de PDN est:

$$\mathfrak{M}(x, y) = \{ \lambda \in \mathbb{R}_+^l : g^2(x, y) \leq 0; \nabla_y L^0(x, y, \lambda) = 0; \lambda^t g^2(x, y) = 0 \}.$$

L'ensemble défini par $\mathfrak{F}_r^{KKT} = \{ (x, y, \lambda) \in X \times Y \times \mathbb{R}_+^l : g^2(x, y) \leq 0; \nabla_y L^0(x, y, \lambda) = 0 \}$ est l'ensemble des solutions réalisables du problème (PDN_{KKT}) relaxé des contraintes de

complémentarité. Pour le problème de PDN généralisé (en fait un MPEC), Luo et al. [2001] (ou encore Marcotte et Zhu [1996]) proposent de reformuler le problème (PDN_{KKT}) en pénalisant la fonction économique du premier niveau par la relation de complémentarité; le programme pénalisé $P(\mu)$ qui en résulte a pour formulation:

$$\begin{aligned}
 P(\mu) \\
 \text{Min}_{x,y,\lambda} f^1(x,y) - \mu \lambda^t g^2(x,y) \\
 \text{s.c. } (x,y,\lambda) \in \mathfrak{F}_r^{KKT}.
 \end{aligned}
 \tag{2.2.2}$$

En supposant que pour toute solution $(x^*, y^*, \lambda^*) \in \mathfrak{F}^{KKT}$, on a $\lambda_i^* + g_i^2(x^*, y^*) \neq 0 \forall i$ et que les conditions de qualification des contraintes de Mangasarian-Fromovitz (MFCQ) sont satisfaites en (x^*, y^*, λ^*) pour le problème (PDN_{KKT}) relaxé des contraintes de complémentarité, Liu et al. [2001]) montrent qu'il existe $\mu^* > 0$ tel que, pour tout $\mu \geq \mu^*$, l'ensemble des solutions optimales du problème (PDN_{KKT}) est égal à l'ensemble des solutions du problème d'optimisation $P(\mu)$. Les résultats relatifs à la formulation (PDN_{KKT}) du problème PDN seront enrichis dans le chapitre suivant, dans la perspective du traitement du problème (PDN_{KKT}) comme un programme mathématique avec contraintes d'équilibre.

2.3 Quelques applications de la programmation mathématique à deux niveaux (PDN)

La capacité de la programmation mathématique à deux niveaux à modéliser des systèmes hiérarchisés de prise de décision est à l'origine de l'abondante recherche qui a été jusqu'à ce jour consacrée à l'étude des PDN ainsi que de leurs applications. Dans sa revue bibliographique consacrée à la PDN, Ben-Ayed [1993] fait un tour d'horizon de quelques unes des principales applications de la PDN. Labbé et al. [1998], Brotcorne et al. [2000] et Lavigne et al. [2000] présentent d'intéressantes applications du problème de PDN dans la détermination optimale des prix aussi bien dans les péages routiers que dans le domaine de l'électricité. En

rapport avec la détermination optimale des taxes sur les crédits liés à la production du fuel biologique, Bard et al. [1998], Bard et al. [2000] puis Dempe et Bard [2000] présentent des contributions dignes d'intérêt. Marcotte et Savard [2001] ou encore Dempe [2003] font également un tour d'horizon des principales applications du problème de PDN.

2.4 Les algorithmes de résolution des problèmes de PDN

Parallèlement à l'étude théorique du problème de PDN, de nombreux algorithmes ont été mis au point pour le résoudre, majoritairement dans le cas linéaire. Ces algorithmes peuvent être classés en trois catégories: la première catégorie s'applique au cas linéaire seulement, en exploitant la propriété selon laquelle une solution optimale de tout problème de PLDN est atteinte en un point extrême de l'ensemble des solutions réalisables Ω ; la deuxième catégorie s'applique à la formulation (PDN_{KKT}) , dans laquelle le problème de second niveau est remplacé par ses conditions d'optimalité de KKT. Ce programme mathématique à un niveau est non convexe et la propriété 2.5 assure, sous des conditions de régularité, que le problème (PDN_{KKT}) est équivalent au problème (PDN) . Enfin, la troisième catégorie concerne les méthodes de descente qui exploitent (i) la formulation (2.1.5) ou (ii) la formulation utilisant une fonction de pénalité.

a) Les algorithmes d'exploration des points extrêmes

Parmi les algorithmes de cette catégorie, citons ceux de Bialas et Karwan [1984], Dempe [1987], Chen et al. [1992] qui ont été conçus pour la formulation particulière des problèmes PLDN. Ces algorithmes, à l'instar de celui proposé par Candler et Townsley [1982], du fait de l'accumulation des bases à traiter peuvent causer des problèmes d'espace mémoire ou rendre le temps de résolution élevé en raison de la nécessité d'explorer un grand nombre de bases optimales (Bard 1983a).

Pour un programme quadratique à deux niveaux, Vicente et al. [1994] ont introduit les notions de points extrêmes du domaine induit et de directions admissibles: en fonction de la

nature de la fonction économique du premier niveau, ces auteurs ont proposé une procédure d'exploration des points extrêmes susceptible de déterminer un optimum local.

b) Les algorithmes basés sur les conditions d'optimalité

Les algorithmes basés sur ce principe exploitent les propriétés du problème (PDN_{KKT}). Ces algorithmes, exception faite de celui de Fortune-Amat et McCarl [1981] ont été développés pour les programmes linéaires à deux niveaux (PLDN) sans contraintes couplantes de premier niveau. La plupart de ces algorithmes ont fait l'objet de tests numériques. Il faudrait noter que, pour la variable de premier niveau x fixée, la formulation (PDN_{KKT}) ne donne pas la garantie de trouver, un optimum global du problème de PDN. De sorte que les méthodes de résolution basées sur les conditions de KKT ne fournissent généralement qu'un optimum local. Citons dans cette catégorie l'algorithme " Parametric Complementary Pivot Algorithm " (PCP) de Bialas et al. [1980] qui fut très populaire et de nombreuses publications lui ont fait référence lors d'études comparatives où il se révélait être performant. Cet algorithme s'inspire d'une part, d'une technique utilisée par Wolfe [1959] pour résoudre un problème convexe de programmation quadratique, où un critère restrictif d'entrée en base est utilisé à l'intérieur de la procédure du simplexe afin d'assurer la complémentarité, et d'autre part fait appel à une coupe de type $c^t x + d^t y > \alpha$ afin d'améliorer la fonction économique du premier niveau; Ben-Ayed et Blair [1990] ont montré que cet algorithme ne converge pas toujours vers une solution optimale.

Nous décrivons à présent sommairement l'algorithme de Gümüs et Floudas [2001].

L'algorithme de Gümüs et Floudas [2001]

Cet algorithme basé sur une technique d'optimisation globale, se propose de résoudre un problème général de PDN. On suppose que toutes les fonctions qui interviennent dans sa formulation sont de classe C^2 , et que les hypothèses générales de régularité sont satisfaites pour le problème de second niveau pour tout x fixé. Si on considère la formulation (PDN_{KKT}) du problème (PDN), le principe de l'algorithme est le suivant:

i) l'ensemble des solutions réalisables \mathcal{F}^{KKT} est relaxé: on élargit le domaine \mathcal{F}^{KKT} en construisant des sous-estimations de toutes les expressions qui apparaissent sous forme de produit ou de fraction dans le problème (PDN_{KKT}) , de manière à rendre convexe toutes les contraintes d'inégalité de la nouvelle formulation du problème de second niveau;

ii) on construit ensuite les bornes de chacune des variables y_j et s_j , s_j désignant la variable d'écart dans la contrainte de second niveau, à savoir: $g_j^2(x, y) + s_j = 0$.

iii) les contraintes d'égalité du problème de second niveau du type $h(x, y) = 0$ sont transformées en deux contraintes d'inégalités:

$$\begin{cases} h(x, y) \leq 0, \\ -h(x, y) \leq 0; \end{cases} \quad (2.4.1)$$

iv) si U désigne la borne supérieure des variables d'écart, chacune des relations de complémentarité $\lambda_j s_j = 0$ est transformée en:

$$\begin{cases} \lambda_j^c - U Y_j \leq 0, \\ s_j^c - U(1 - Y_j) \leq 0, \\ \lambda_j^c, s_j^c \geq 0, \\ (x, y) \in X \times Y, Y_j \in \{0, 1\}. \end{cases} \quad (2.4.2)$$

Le problème qui en résulte est un programme mathématique non linéaire à variables mixtes. Les auteurs proposent d'utiliser un logiciel de programmation mixte pour le résoudre. L'algorithme a été testé sur cinq problèmes de petite taille.

Les transformations que Gümüs et Floudas utilisent pour résoudre un problème de PDN posent trois problèmes:

- La transformation d'une contrainte d'égalité $h(x, y) = 0$ en deux contraintes d'inégalité (2.4.2) peut briser la stabilité du programme initial.

- Il peut être difficile d'identifier dans (2.4.2) une valeur de U qui soit suffisamment grande.

- La transformation des relations de complémentarité $\lambda_j s_j = 0$ en deux contraintes du type (2.4.3) transforme le problème initial en un problème combinatoire aussi complexe à résoudre que le problème initial. Signalons que des transformations similaires avaient déjà été proposées par Fortuney-Amat et McCarl [1981]; il est connu que le programme qui résulte de ce type de formulation est généralement difficile à résoudre.

Les algorithmes basés sur une méthode de séparation et évaluation font également partie de cette catégorie. Ils suscitent l'espoir de disposer d'une méthode de résolution du problème de PDN de grande taille, d'autant plus qu'ils sont à même de déterminer un optimum global. Les algorithmes de Savard [1989], Hansen et al. [1992] utilisent les principes de monotonie sur les contraintes du problème de second niveau, et s'articulent sur (1) les méthodes par séparation et évaluation, (2) l'élimination du problème de certaines variables en fonction de la nature (serrée ou non) des contraintes du problème de second niveau. Les autres algorithmes significatifs de cette catégorie à notre sens, sont ceux de Bard et Falk [1982], Papavassilopoulos [1982], Bard et Moore [1990]. Pour l'algorithme de Bard et Moore, la séparation est faite sur les relations de complémentarité du problème (PDN_{KKT}): on résout le problème résultant de (PDN_{KKT}) relaxé des contraintes d'équilibre, en fixant successivement à zéro chacune des variables non nulles qui intervient dans une relation de complémentarité. On retient la solution obtenue si elle est réalisable pour le problème (PDN_{KKT}) et si elle améliore la valeur de la fonction économique du premier niveau. De nombreux travaux ont fait référence à cet algorithme. Cependant, il reste moins performant que l'algorithme MAYBIL de Savard [1989], Hansen et al. [1992].

L'algorithme proposé par Jùdice et Faustino [1992] fait appel à une stratégie de pivot basée sur les relations de complémentarité entre les multiplicateurs de KKT et les variables d'écart. D'après les expérimentations numériques faites par les auteurs, cet algorithme s'est avéré performant sur les problèmes de tailles moyennes, mais n'a pas été concluant sur les problèmes de grandes dimensions. Ceci pourrait s'expliquer par le nombre des branches de l'arbre à explorer qui croît exponentiellement avec la taille des variables du problème.

Pour résoudre un problème de programmation mathématique quadratique à deux niveaux, Muu et Quy [2003] proposent de transformer ce problème en un programme

mathématique à un niveau en définissant une fonction de mérite, puis utilisent une méthode par séparation et évaluation.

c) Les algorithmes de descente

Nous ferons ici une distinction entre les algorithmes utilisant une fonction de pénalité et les algorithmes utilisant une méthode de descente.

En programmation mathématique standard, les méthodes du gradient constituent un des outils privilégiés. Quelques auteurs, à l'instar de Aiyoshi et Shimizu [1981], Kolstad et Lasdon [1990], Falk et Liu [1995] ont proposé d'utiliser une méthode du gradient pour résoudre des programmes à deux niveaux. Ces méthodes ont été peu utilisées en raison de deux problèmes majeurs: (1) la difficulté de calculer la fonction implicite $y(x)$ qui n'est pas différentiable partout, (2) la non convexité inhérente au problème à deux niveaux qui conduit à l'existence d'optima locaux et la convergence vers l'optimum global dépend du point initial choisi.

Les algorithmes utilisant les fonctions de pénalité quant à eux incorporent habituellement des fonctions de pénalité exacte, qui sont déduites du saut de dualité du problème de second niveau dans le cas linéaire. Ces fonctions jouent un rôle important pour les problèmes de PDN, autant pour établir des conditions d'optimalité que pour élaborer des algorithmes de résolution. Marcotte et Zhu [1996], puis Liu et al. [2001] proposent une série de fonctions de pénalité exacte susceptibles d'être utilisées pour reformuler un problème de PDN.

i) Les algorithmes utilisant une méthode de descente

Nous présentons brièvement les méthodes proposées par Kolstad et Lasdon [1990], Falk et Liu [1995] et par de Still [2002].

L'algorithme de Kolstad et Lasdon [1990]

Le principe de cet algorithme consiste à appliquer une méthode de gradient sur le problème à un seul niveau (2.1.5) de la section 2.1. Le vecteur y s'exprime comme une fonction de x de sorte que la fonction économique de ce problème est

$$\text{Maximiser}_x c_1^t x + d_1^t y(x). \quad (2.4.3)$$

En pratique, $y(x)$ n'est pas une fonction, mais une application multivoque. Les auteurs font l'hypothèse que $y(x)$ est une fonction univoque différentiable. Kolstad et Lasdon ont développé une méthode efficace pour le calcul du gradient de la fonction $y(x)$ lorsque le problème de second niveau possède plusieurs contraintes non actives dans la solution optimale; cette méthode est une adaptation de la méthode proposée par Fiacco [1976].

Sous l'hypothèse de complémentarité stricte, la variable $y(x)$ est partitionnée en deux catégories de variables: celles qui sont contenues dans leur borne initiale et celles qui ne le sont pas; ce qui permet de calculer le jacobien $\nabla y(x)$ de la fonction implicite $y(x)$ en ne considérant que les variables qui ne sont pas à leur borne. Cette méthode s'est avérée efficace dans un premier temps pour les problèmes où la majorité des variables de la solution optimale du problème (2.1.5) sont hors base.

Kolstad et Lasdon [1990] ont ensuite procédé à des expérimentations numériques significatives sur des problèmes ayant jusqu'à 230 variables et 30 contraintes. Si la méthode s'est avérée performante sur la quasi totalité des exemples traités, il convient toutefois de signaler que l'algorithme peut être pris au piège lorsque le processus itératif conduit à un point col. Les auteurs n'ont pas prévu un mécanisme pour contourner cette difficulté.

L'algorithme de Falk et Liu [1995]

Falk et Liu proposent un algorithme de résolution du problème PDN mis sous la forme (2.1.5) à savoir:

$$\text{Minimiser}_{x \in M} P f^1(x) \equiv f^1(x, y(x)). \quad (2.4.4)$$

Le principe de l'algorithme que les auteurs ont baptisé algorithme du leader prédominant (leader predominate algorithm) est le suivant:

On suppose qu'à l'itération k , la stratégie du leader est x^k , la réaction optimale du suiveur étant y^k . Partant de x^k , le leader se déplace dans la direction d^k que l'on détermine en résolvant le programme ci-après qui peut prendre la forme d'un programme quadratique

$$\text{Minimiser}_{\|d\|=1} DPf^1(x^k; d) = \frac{\partial f^1}{\partial x} d + \frac{\partial f^1}{\partial y} Dy(x^k; d) \quad (2.4.5)$$

où $Dh(a; d)$ est la dérivée de la fonction h en a dans la direction d .

Cependant, la fonction Pf^1 n'étant pas toujours différentiable, les auteurs suggèrent d'exploiter les itérations antérieures afin de calculer un sous gradient $\xi \in \partial Pf^1$.

La mise en oeuvre de l'algorithme suppose que l'on dispose (a) d'un logiciel de résolution d'un programme mathématique standard, notamment pour trouver une solution réalisable au problème (PDN_{KKT}) relaxé des contraintes d'équilibre, (b) d'une procédure permettant de calculer un sous gradient ξ .

Dans une autre variante de cet algorithme (the adaptive leader predominate algorithm), la direction de déplacement d^k est calculée par une méthode de quasi-Newton si la fonction Pf^1 est différentiable en x^k . Les auteurs montrent que cet algorithme est localement convergent, mais ils n'ont pas procédé à des expérimentations numériques significatives.

L'algorithme de Still [2002]

Cet algorithme cherche à déterminer un optimum local du problème de type (PLDN) que Still met sous la forme:

$$\begin{aligned} & (PLDN0) \\ & \text{Minimiser}_{x, y \geq 0} a_1^t x + b_1^t y \equiv d_1^t z \\ & \text{s.c.} \left\{ Q(x) \right\} \begin{cases} \text{Minimiser}_{y, d_2^t y} \\ \text{s.c.} \{ Ax + By \leq c (\equiv Cz \leq c) \} \end{cases} \end{aligned} \quad (2.4.6)$$

Quelques définitions sont requises.

Soit J l'ensemble des indices des contraintes de ce problème; l'ensemble des solutions réalisables peut se mettre sous la forme $\Omega = \{z \in \mathbb{R}^{n_x+n_y} : C_j z \leq c_j, j \in J\}$ où C_j désigne la ligne d'indice j de la matrice C . Soit $J_0 \subset J$; considérons l'ensemble défini par

$$f_0 = f(J_0) = \{z \in \Omega : C_j z = c_j, j \in J_0\}.$$

Pour $z \in \Omega$, l'ensemble des **indices des contraintes actives** est défini par:

$$J(z) = \{j \in J : C_j z = c_j\}.$$

Pour l'ensemble $J_k \subset J$ de l'itération k , le sous espace $S(J_k)$ est défini par

$$S(J_k) = \{z \in \mathbb{R}^{n_x+n_y} : C_j z = 0, j \in J_k\}.$$

À chaque étape de l'algorithme, il faut calculer la projection de la fonction économique $-d_1$ sur le sous espace $S(J_k)$.

Le cadre conceptuel général de l'algorithme se présente ainsi:

On calcule une solution réalisable z_k du problème (PLDNO) en résolvant le programme linéaire:

$$\begin{aligned} & \text{Minimiser}_z d_1^t z \\ & \text{s.c.} \{Cz \leq c. \end{aligned}$$

Soit $\bar{z} = (\bar{x}, \bar{y})$ sa solution optimale.

Si \bar{y} est une solution optimale de $Q(\bar{x})$, on a trouvé une solution du problème (PLDNO).

Sinon, on calcule à l'itération k la solution optimale y_k ; on a: $z_k = (\bar{x}, y_k)$.

Soit $J_k = J(z_k)$ et s_k la projection de $-d_1$ sur $S(J_k)$. z_k est la solution courante et s_k la direction de déplacement.

Si la direction s_k est telle que $d_1^t s_k < 0$, alors il n'existe pas de direction de descente.

Dans ces conditions, z_k est un minimum local.

Sinon, déterminer le pas de déplacement t_k solution du problème:

$$\text{Minimiser}_\tau \{d_1^t z_{k+1} = d_1^t (z_k + \tau s_k) : \tau > 0 \text{ et } z_{k+1} \in \Omega\}.$$

Si on ne peut pas déterminer t_k , alors z_k est un minimum local.

D'après l'auteur, les résultats des tests numériques qu'il a effectués montrent de manière expérimentale que son algorithme a un comportement polynomial. Signalons que cet algorithme ne permet de déterminer qu'un optimum local du PLDN résolu.

ii) Les algorithmes utilisant une fonction de pénalité

Nous présentons l'algorithme proposé par White et Anandalingam [1993]. Il s'applique aux problèmes de PLDN de type:

$$\begin{aligned} & \text{Minimiser}_{x,y} F(x,y) = a^t x + b^t y \\ & \text{s.c.} \left\{ \begin{array}{l} \text{Minimiser}_y d^t y \\ \text{s.c.} \left\{ \begin{array}{l} A_2 x + B_2 y \leq p, \\ x, y \geq 0. \end{array} \right. \end{array} \right. \end{array} \quad (2.4.7) \end{aligned}$$

Les ensembles $Z = \{(x,y) \geq 0 : A_2 x + B_2 y \leq p\}$ et $U = \{u \geq 0 : u^t B_2 \geq d\}$ représentent respectivement le domaine réalisable et l'ensemble des variables duales du problème du second niveau. Anadalingam et White, considèrent cette formulation particulière du problème de PLDN, puis font appel au saut de dualité du programme linéaire de second niveau défini en tout point $(x,y,u) \in Z \times U$ par: $\Pi(x,y,u) = u^t(p - A_2 x) - d^t y$. On obtient la formulation du problème (PLDN) pénalisé:

$$\begin{aligned} & P(K) \\ & \text{Maximiser}_{x,y,u} \hat{F}(x,y,u,K) \equiv c_1^t x + d_1^t y - K \Pi(x,y,u) \\ & \text{s.c.} \left\{ \begin{array}{l} (x,y) \in Z, \\ u \in U, \end{array} \right. \end{array} \quad (2.4.8)$$

où K est arbitrairement grand. Soit $(x,y,u) \in Z \times U$. Si $\Pi(x,y,u) = 0$, alors (x,y) est une solution réalisable pour la formulation particulière du PLDN. Soit Z_v , U_v et $(Z \times U)_v$ respectivement

l'ensemble des points extrêmes des polyèdres Z , U et $Z \times U$. Anadalingam et White ont fait les hypothèses A1 et A2 ci-après dans la perspective de développer un algorithme de résolution du problème $P(K)$. Signalons toutefois que Campêlo et al. [2000] établissent par le biais de la propriété ci-dessous que l'hypothèse A2 ne peut jamais être satisfaite.

Hypothèse A1: Lorsque x^* est la solution optimale du problème du premier niveau, alors l'application $M(x^*)$ est univoque;

Hypothèse A2: Les polyèdres Z et U sont simultanément non vides et bornés;

Propriété 2.6 (Campêlo et al., 2000): Si l'ensemble $Y = \{y : B_2 y \leq q, y \geq 0\}$ est non vide et borné, alors l'ensemble $U = \{u \geq 0 : u^t B_2 \geq d_2\}$ est non vide et non borné. De même si l'ensemble U est non vide et borné, alors l'ensemble Y est non vide et non borné.

Pour obtenir un algorithme bien défini, Campêlo et al. [2000] font les hypothèses selon lesquelles on a:

(A): $U \neq \emptyset$ et le programme linéaire $PL(P): \text{Max}\{a^t x + b^t y : (x, y) \in Z\}$ a une solution optimale;

(B): U est un ensemble compact.

Les conditions d'optimalité du problème pénalisé $P(K)$ sont indiquées dans les résultats ci-après; elles ont été établies par Anadalingam et White.

Propriété 2.7 (Anadalingam et White, 1993): Pour $K \in \mathbb{R}_+$ fixé, une solution optimale du problème $P(K)$ lorsqu'elle existe est atteinte en un point appartenant à $Z_v \times U_v$ et $Z_v \times U_v = (Z \times U)_v$.

On déduit de cette propriété qu'une solution optimale du problème pénalisé est atteinte sur les sommets des domaines réalisables des programmes linéaires primal et dual du problème

de second niveau du PDN. Le programme pénalisé $P(K)$ est alors résolu à chaque itération. À partir d'un point extrême u^1 de U , qui permet de déterminer une solution optimale $(x(\bar{u}^1), y(\bar{u}^1), \bar{u}^1)$ de $P(K)$, on procède à une exploration de l'ensemble (de cardinal $N(u^1)$) des points extrêmes u^{1s} qui lui sont adjacents, lorsqu'une solution optimale n'est pas atteinte en u^1 . Une procédure similaire à la méthode des coupes de Tuy et al. [1993] permet d'éliminer de l'ensemble U le point u^1 . Les auteurs démontrent qu'il existe une valeur finie $K^* \in \mathbb{R}_+$ pour laquelle une solution optimale du problème $P(K)$ est atteinte en un point extrême u^1 de U , solution optimale du problème (2.4.8) pour tout $K \geq K^*$. Ce résultat est consigné dans la propriété ci-dessous qui indique aussi les conditions d'optimalité du problème de PLDN.

Propriété 2.8 (Anadalingam et White, 1993): Il existe une valeur $K^* \in \mathbb{R}_+$ pour laquelle une solution optimale du problème pénalisé $P(K)$ conduit à une solution optimale pour le problème de PLDN dans sa formulation particulière, pour tout $K \geq K^*$.

Notons cependant que le temps de calcul consacré à l'exploration des points extrêmes adjacents à u^1 peut être considérable et rendre l'algorithme inefficace. Les expérimentations numériques faites par les auteurs ont montré que cet algorithme est moins performant que la méthode proposée par Bard et Moore [1990].

Enfin, du fait de la similitude de plusieurs propriétés structurelles des problèmes de PDN sous forme KKT et des problèmes d'optimisation discrète, les auteurs comme Anandaligam et al. [1983], Friesz et al. [1987], Gendreau et al. [1996] ont proposé des méthodes heuristiques pour résoudre les problèmes de PDN.

En définitive, les algorithmes basés sur une méthode de séparation et d'évaluation, du fait de la nature combinatoire du problème de PDN, sont ceux qui ont fourni les meilleurs résultats jusqu'à présent.

CHAPITRE 3 – REVUE DE LA LITTÉRATURE SUR LES PROGRAMMES MATHÉMATIQUES AVEC CONTRAINTES D'ÉQUILIBRES (MPEC)

Ce chapitre est consacré à la présentation d'une synthèse bibliographique des programmes mathématiques avec contraintes d'équilibre. Nous énonçons les principales propriétés, puis nous passons en revue les principaux algorithmes de résolution. Nous introduisons une étude à la compréhension de quelques propriétés relatives à l'ensemble des solutions d'un problème de programmation mathématique à deux niveaux (PDN), en nous inspirant à cet effet des éléments de théorie liés aux solutions du problème de MPEC ou du problème défini par une inégalité variationnelle. Les problèmes de MPEC auxquels nous nous intéressons sont en fait les problèmes de PDN sous forme KKT. En effet, certains algorithmes de résolution des PDN font généralement référence aux caractéristiques de l'application qui à la variable de premier niveau x , associe l'ensemble des solutions optimales du problème de second niveau $M(x)$: dans l'algorithme MAYBIL développé par Savard [1989], l'un des principes de monotonie sur lequel repose cet algorithme porte sur l'hypothèse d'univocité de l'application $M(x)$; il en est de même pour l'algorithme de descente proposé par Savard et Gauvin [1990] ou encore de l'algorithme développé par White et Anandalingam [1993].

Le chapitre est organisé comme suit: dans la section 3.1, nous présentons les notations et les définitions relatives aux problèmes de MPEC d'une manière générale. La section 3.2 est consacrée aux définitions et aux propriétés liées aux problèmes de complémentarité. Dans la section 3.3, nous faisons un tour d'horizon du concept des fonctions de mérite qui sont devenues d'importants outils pour la résolution des problèmes de complémentarité ou des problèmes de MPEC. Dans la section 3.4, nous parlons du problème de MPEC sous forme KKT qui présente des similitudes avec la formulation (PDN_{KKT}) d'un problème de programmation mathématique à deux niveaux; nous passons en revue les principales définitions et propriétés liées à cette catégorie de problèmes de MPEC. Dans la section 3.5, nous proposons une première contribution théorique consacrée à l'étude de propriétés qui caractérisent l'univocité de l'application $M(x)$ en

programmation mathématique à deux niveaux. Dans la section 3.6, nous présentons les principaux algorithmes de résolution du problème de MPEC.

3.1 Formulation générale du problème de MPEC: définitions et notations

Le problème de MPEC est une généralisation du problème de programmation mathématique à deux niveaux (PDN). Marcotte et Zhu [1996] puis Marcotte et al. [2001] définissent le problème de MPEC comme un problème de PDN généralisé: c'est un programme mathématique dont les contraintes incluent un problème défini par une inégalité variationnelle. Avant de donner une formulation du problème de MPEC dans le cas général, les définitions et les notations relatives à un problème de complémentarité ou à une inégalité variationnelle s'avèrent utiles. Pour cela, considérons la fonction $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_y}$ et l'application multivoque $C(x)$ définie par:

$$\forall x \in X \subseteq \mathbb{R}^{n_x}, C(x) \neq \emptyset \text{ où } C(x) = \{y \in \mathbb{R}^{n_y} : g^2(x, y) \leq 0 \text{ et } h(x, y) = 0\},$$

$g^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^l$ et $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^q$. Lorsque l'ensemble $C(x)$ représente un cône convexe ou un polyèdre, on désigne par $C^*(x)$ le dual de $C(x)$ au sens de l'analyse convexe.

Soit $X \subseteq \mathbb{R}^{n_x}$ et $x \in X$ fixé tels que l'ensemble $C(x)$ est un cône convexe ou un polyèdre borné non vide. Si l'ensemble $M(x)$ défini au chapitre précédent représente l'ensemble des solutions de l'**inégalité variationnelle** (VI) définie par la paire $(F(x, \cdot), C(x))$, c'est-à-dire, $y \in M(x)$ si et seulement si $y \in C(x)$ et vérifie l'inégalité:

$$(v - y)^t F(x, y) \geq 0, \forall v \in C(x). \quad (3.1.0)$$

Cette inégalité variationnelle est notée $VI(F(x, \cdot), C(x))$.

Définition 3.1: Un problème défini par l'inégalité variationnelle $VI(F(x, \cdot), C(x))$ noté (PVI) consiste à déterminer un vecteur $y^* \in C(x)$ qui satisfait (3.1.0).

Définition 3.2: Pour chaque élément $x \in X$ fixé et étant donné un cône $C(x)$, un **problème de complémentarité généralisé** noté $GCP(F(x, \cdot), C(x))$ sur $C(x)$ est l'ensemble des solutions $y \in C(x)$ qui satisfont aux relations:

$$F(x, y) \in C^*(x) \text{ et } y^t F(x, y) = 0. \quad (3.1.1)$$

La formulation d'un problème de complémentarité non linéaire est la suivante:

(NCP): Trouver $(x, y) \in X \times Y \subseteq \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ tel que $y \in C(x)$ et $0 \leq F(x, y) \perp y \geq 0$.

La notation $F(x, y) \perp y$ signifie que l'on a $y^t F(x, y) = 0$. Dans le cas linéaire, $C(x)$ est un polyèdre convexe et la fonction $F(x, y)$ se met sous la forme:

$$F(x, y) = My + Nx + q \text{ où } q \in \mathbb{R}^n, M \in \mathbb{R}^m \times \mathbb{R}^{n_y}, N \in \mathbb{R}^m \times \mathbb{R}^{n_x}.$$

Le problème de complémentarité linéaire est noté $LCP(q, M, N)$. Lorsque $N = 0$, on le note $LCP(q, M)$. La fonction $F(x, \cdot)$ que l'on note F_x représente quelquefois le gradient (par rapport à la variable y) d'une fonction réelle de classe C^1 , $f^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ c'est-à-dire $F(x, y) = \nabla_y f^2(x, y)$. À la lumière de ce qui précède, le problème de complémentarité généralisé $GCP(F(x, \cdot), C(x))$ représente l'ensemble des points stationnaires y du problème d'optimisation ci-après, que l'on pourrait assimiler au problème de second niveau d'un problème de PDN:

$$(PM2) \quad \begin{array}{l} \text{Minimiser}_y f^2(x, y) \\ \text{s.c.} \begin{cases} g^2(x, y) \leq 0, \\ y \in Y. \end{cases} \end{array}$$

Pour un vecteur $x \in X$ donné, on peut écrire $M(x) = \operatorname{argmin}_y \{f^2(x, y), y \in C(x)\}$.

Luo et al. [1996a], faisant référence à la théorie des inégalités variationnelles font remarquer que lorsque l'ensemble $C(x)$ est un cône convexe, le problème défini par l'inégalité variationnelle

$$y \in C(x) \text{ et } (v - y)^t F(x, y) \geq 0, \forall v \in C(x)$$

est un problème de complémentarité généralisé. Sous la forme la plus générale, la formulation d'un problème de MPEC est la suivante:

$$\begin{aligned}
 & \text{(MPEC)} \\
 & \text{Minimiser}_{x,y} f^1(x, y) \\
 & \text{s.c.} \begin{cases} h(x, y) = 0, \\ g^2(x, y) \leq 0, \\ 0 \leq F(x, y) \perp y \geq 0, \\ (x, y) \in Z = X \times Y, \end{cases} \quad (3.1.2)
 \end{aligned}$$

où $f^1 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $Z = X \times Y \subset \mathbb{R}^{n_x + n_y}$ est un sous-ensemble fermé non vide. Dans le cas linéaire, la formulation générale du problème de MPEC est:

$$\begin{aligned}
 & \text{(MPEC}(M, N, q)) \\
 & \text{Minimiser}_{x,y} f^1(x, y) \\
 & \text{s.c.} \begin{cases} w = My + Nx + q, \\ 0 \leq w \perp y \geq 0, \\ (x, y, w) \in X \times Y \times \mathbb{R}^n, \end{cases} \quad (3.1.3)
 \end{aligned}$$

où $f^1 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ est une fonction affine. Lorsque le sous ensemble $M(x)$ de \mathbb{R}^{n_y} représente l'ensemble des solutions d'un problème de complémentarité ou d'une inégalité variationnelle, ou alors l'ensemble des solutions du problème de second niveau de type (PM2) d'un problème de PND, un problème de MPEC peut être formulé comme un problème de programmation mathématique à un niveau, à savoir:

$$\begin{aligned} & \text{Minimiser}_{x,y} f^2(x,y) \\ & \text{s.c.} \begin{cases} y \in M(x), \\ (x,y) \in Z. \end{cases} \end{aligned} \quad (3.1.4)$$

Les caractéristiques du problème de MPEC que nous présentons dans la section suivante, dépendent des propriétés de la fonction F ou de la matrice M des contraintes (3.1.3).

Remarque 3.1: Avec les notations utilisées dans le chapitre 2 pour la formulation (PLDN), posons $f^1(x,y) = c_1^t x + d_1^t y$, $C(x) = \{y \geq 0 : A_2 x + B_2 y \leq b_2\}$ dont le sous espace dual est $C^*(x) = \{z \in \mathbb{R}_+^m : z^t B_2 \geq 0\}$, $F(x,y) = d_2$. Une formulation du type (3.1.3) s'obtient en posant:

$$y \equiv \begin{pmatrix} y' \\ z \end{pmatrix}, q = \begin{pmatrix} -d_2 \\ b_2 \end{pmatrix}, N = \begin{bmatrix} 0 \\ -A_2 \end{bmatrix} \text{ et } M = \begin{bmatrix} 0 & B_2^t \\ -B_2^t & 0 \end{bmatrix}.$$

3.2 Le problème de complémentarité

Nous présentons à présent les principales définitions relatives au problème de complémentarité; en effet, déterminer une solution réalisable d'un problème de MPEC revient à trouver une solution au problème de complémentarité dont les équations constituent les contraintes de ce MPEC. De plus, les propriétés de la matrice M (respectivement de la fonction F) sont à l'origine des conditions d'existence des solutions des problèmes de type $LCP(q,M,N)$ (respectivement NCP) et du comportement des algorithmes qui conduisent à ces solutions. Les caractéristiques du domaine induit d'un problème de PDN peuvent également dépendre des propriétés de la matrice M ou de la fonction F . Nous commencerons cette section en rappelant brièvement les principales définitions qui se rapportent au problème de complémentarité, puis nous présenterons les principales propriétés de ce problème.

3.2.1 Définitions

Les définitions qui suivent se rapportent aux matrices ou fonctions qui sont utilisées dans une relation de complémentarité ou une inégalité variationnelle.

Définitions 3.3: Une matrice M est:

- i) une **P₀-matrice** si $\forall y \in \mathbb{R}^n, y \neq 0, \exists i \in I : y_i \neq 0$ et $y_i [My]_i \geq 0$ (tous les mineurs principaux de M sont positifs ou nuls);
- ii) une **P-matrice** si $\forall y \in \mathbb{R}^n, y \neq 0, \exists i \in I : y_i \neq 0$ et $y_i [My]_i > 0$ (tous les mineurs principaux de M sont strictement positifs);
- iii) une **R₀-matrice** si $LCP(0, M)$ a pour unique solution $z^* = (0, 0) \in \mathbb{R}^n$.

On peut établir que toute P-matrice est à la fois une P₀-matrice et une R₀-matrice.

Soient $X \subseteq \mathbb{R}^{n_x}$ un ensemble non vide, pour $x \in X$ fixé et la fonction $F_x : Y \rightarrow \mathbb{R}^{n_y}$ définie par: $y \mapsto F(x, y)$. Le concept de P₀-fonction qui se rattache à la fonction F_x est une extension naturelle de la notion de P₀-matrice (Huang, 2002); donnons en la définition.

Définition 3.4: La fonction F_x est une **P₀-fonction** si pour tous $u, v \in Y$ avec $u \neq v$, il existe un indice i tel que $(u_i - v_i)(F_x(u) - F_x(v))_i \geq 0$.

La propriété de monotonie gouverne l'existence de solutions du problème de complémentarité non linéaire et le comportement des algorithmes qui permettent de trouver ces solutions. Nous définissons les concepts de monotonie et de coercivité d'une fonction qui confèrent au problème de complémentarité des propriétés qui peuvent faciliter sa résolution.

Définitions 3.5: On dit qu'une fonction F est:

- i) **monotone** sur Y si $(t - y)^t (F(t) - F(y)) \geq 0$ pour tout $t, y \in Y$;

- ii) **strictement monotone** sur Y si $(t - y)^t (F(t) - F(y)) > 0$ pour tout $t, y \in Y$ avec $t \neq y$;
- iii) **fortement monotone** sur Y s'il existe une constante $\mu > 0$ (appelée module) telle que $(t - y)^t (F(t) - F(y)) \geq \mu \|t - y\|^2 \quad \forall t, y \in Y$;
- iv) **coercive** sur Y s'il existe $y^0 \in Y$ tel que $\lim_{y \in Y, \|y\| \rightarrow \infty} F(y)^T (y - y^0) / \|y\| = \infty$;
- v) **co-coercive** sur Y s'il existe $\alpha > 0$ tel que $(t - y)^T (F(t) - F(y)) \geq \alpha \|F(t) - F(y)\|^2$ pour tout t, y dans Y .

$\|\cdot\|$ désigne une norme sur \mathbb{R}^{n_y} . Il faudrait noter que toute fonction co-coercive est monotone, mais pas nécessairement fortement monotone (par exemple, une fonction constante). Si la fonction F est fortement monotone et continue lipschitzienne sur Y , alors elle est co-coercive sur Y . De plus, si F est une fonction affine du type $F(y) = q + My$ et $Y = \mathbb{R}^{n_y}$, alors F est coercive si et seulement F est fortement monotone, ce qui revient à dire que la matrice M est définie positive.

Il est facile de montrer que si Y est un ensemble fermé de \mathbb{R}^{n_y} et F est une fonction coercive et continue sur Y , alors elle atteint son minimum sur Y . Plus généralement, si l'ensemble Y est fermé, si F est continue et s'il existe un vecteur $y^0 \in Y$ tel que $\{y \in Y : F(y) \leq F(y^0)\}$ est borné (donc compact), alors F atteint son minimum sur Y . Harker et Pang [1990], puis Huang [2002] passent en revue les propriétés des fonctions F ainsi définies lorsqu'elles interviennent dans la modélisation des problèmes définis par une inégalité variationnelle ou des problèmes de complémentarité non linéaires. Zhu et Marcotte [1996] pour leur part, consacrent une étude au rôle joué par la propriété de co-coercivité dans la convergence d'algorithmes itératifs pour résoudre les problèmes définis par une inégalité variationnelle.

Comme nous l'avons rappelé, une solution réalisable d'un problème de MPEC est en fait une solution du problème de complémentarité défini par les contraintes de ce MPEC. Plus

précisément, une solution réalisable ou encore le domaine réalisable d'un problème de MPEC se définit comme suit:

Définitions 3.6

i) Un point $z = (x, y)$ est une **solution réalisable** d'un problème de MPEC si z est une solution du problème de complémentarité défini par les contraintes de ce MPEC.

ii) Le **domaine réalisable** \mathfrak{F} d'un problème de MPEC est constitué de l'ensemble des solutions réalisables de ce MPEC.

Par exemple pour la formulation générale du problème de MPEC linéaire,

$$\mathfrak{F} = \{(x, y, w) \in X \times \mathbb{R}^{2n_y} : w = Nx + My + q; 0 \leq w \perp y \geq 0\}.$$

D'une manière générale le domaine \mathfrak{F} n'est pas convexe. Ceci peut être illustré avec l'exemple 2.2 du chapitre précédent qui est un problème de programmation mathématique à deux niveaux (PDN); le domaine \mathfrak{F} est en fait le domaine induit de ce PDN.

3.2.2 Propriétés du problème de complémentarité

Cette section contient les principales propriétés du problème de complémentarité. Les deux propriétés ci-dessous garantissent l'existence d'une solution pour un problème de complémentarité. La propriété de co-coercivité de la fonction F_x assure outre l'existence de cette solution, mais aussi la compacité de l'ensemble des solutions du problème (NCP). Ces deux propriétés peuvent être utiles dans l'étude des propriétés de l'ensemble $M(x)$ des solutions optimales du problème de second niveau d'un programme mathématique à deux niveaux (PDN).

Propriétés 3.1

1) (Cottle et al., 1992): Le problème $LCP(q, N, M)$ admet une solution unique (quand elle existe) lorsque la matrice M est une P_0 -matrice et R_0 -matrice.

2) Soit $Y \subseteq \mathbb{R}^m$ un ensemble non vide et soit une fonction $F_x : Y \rightarrow \mathbb{R}^m$. Les assertions suivantes sont vraies:

- a) Si F_x est monotone sur Y , alors l'ensemble des solutions du problème (*NCP*) est convexe (éventuellement vide).
- b) Si F_x est strictement monotone sur Y , alors le problème (*NCP*) admet au plus une solution.
- c) Si F_x est fortement monotone sur Y , alors la solution du problème (*NCP*) est unique.

Propriété 3.2 (Harker et Pang, 1990): Soit Y un ensemble convexe non vide et fermé de \mathbb{R}^m et F_x une fonction continue: $Y \rightarrow \mathbb{R}^m$. Si F_x est coercive par rapport à Y , alors l'ensemble des solutions du problème (*NCP*) est non vide et compact.

3.3 Les fonctions de mérite

Les fonctions de mérite constituent d'excellents outils de modélisation et d'analyse en programmation mathématique. Elles permettent de transformer les relations de complémentarité d'un problème de MPEC en contraintes régulières au voisinage d'une solution optimale. Facchinei et Soares [1997], puis Ferris et Kanzow [1998] proposent les définitions suivantes:

Définitions 3.7:

i) (Facchinei et Soares, 1997) Étant donné un problème d'optimisation (*OP*) défini sur $C \subseteq \mathbb{R}^m$, une fonction de mérite pour le problème (*OP*) est une fonction non négative $F_m : C \rightarrow \mathbb{R}$ telle que l'ensemble des solutions du problème (*OP*) coïncide avec l'ensemble des solutions du programme ci-après:

$$(P) \text{ Minimiser }_x F_m(x), x \in C. \quad (3.3.1)$$

Si le problème (OP) n'a pas de solution, alors le programme (P) a une solution \bar{x} telle que $F_m(\bar{x}) \neq 0$.

ii) (Ferris et Kanzow, 1998) Une fonction $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée fonction de mérite pour le problème (NCP) si elle a les propriétés suivantes:

- a) $\Phi(x) \geq 0$ pour tout $x \in \mathbb{R}^n$;
- b) $\Phi(x) = 0$ si et seulement si x est solution du problème (NCP).

Comme fait remarquer Facchinei et Soares [1997], la détermination d'une fonction de mérite pour un problème (OP) n'est pas difficile, par contre, l'idéal consiste à trouver une fonction de mérite ayant d'excellentes propriétés des points de vue théorique ou algorithmique.

La fonction de Chen-Harker-Kanzow-Smale (CHKS) fait partie des fonctions de mérite; elle est définie par rapport à la relation de complémentarité $ab = 0$ par:

$$\Phi(a, b, \mu) = a + b - \sqrt{(a - b)^2 + 4\mu^2}.$$

Burke et Xu [2000] utilisent cette fonction pour trouver une solution au problème de complémentarité linéaire $LCP(q, M)$ par un algorithme du point non intérieur. Une variante de la fonction de mérite de Fisher-Burmeister [1992] est définie de $\mathbb{R}_+^2 \times \mathbb{R} \rightarrow \mathbb{R}$ par:

$$\Phi_{FB}(a, b, \mu) = a + b - \sqrt{a^2 + b^2 + \mu}.$$

La fonction Φ_{FB} a la propriété intéressante (NCP) établie par Kanzow [1995] (lemme 2.2) selon laquelle

$$\Phi_{FB}(a, b, \mu) = 0 \Leftrightarrow 0 < a, 0 < b \text{ et } ab = \frac{\mu}{2}. \quad (\text{NCP})$$

On dit alors que la fonction de mérite Φ_{FB} est une NCP-fonction, au même titre que la fonction de mérite de CHKS. De plus, cette fonction est continûment différentiable et concave dans $\mathbb{R}^2 \times \mathbb{R}_+$. Fukushima et al. [1998] ont utilisé avec succès cette fonction de mérite pour résoudre des problèmes de MPEC à contraintes linéaires par une méthode d'optimisation séquentielle; Chen et al. [2000] se proposent de résoudre un problème de complémentarité non linéaire consistant à

trouver $x \in \mathbb{R}^n$ tel que $x \geq 0$, $F(x) \geq 0$ et $x^t F(x) = 0$. Plutôt que d'utiliser la fonction de Fisher-Burmeister $\Phi_{FB}(a, b, 0)$ qui n'est pas régulière au voisinage de $(a, b) = (0, 0)$, ils font appel à une fonction pénalisée de Fisher-Burmeister définie pour $\lambda \in [0, 1]$ arbitrairement choisi par:

$$\Phi_{\lambda}(a, b, 0) = \lambda \Phi_{FB}(a, b, 0) + (1 - \lambda) a_+ b_+ \text{ avec } z_+ = \max(0, z).$$

Le terme $a_+ b_+$ pénalise toute violation de la relation de complémentarité $ab = 0$. Seulement, cette fonction de mérite pénalisée suppose que l'on fait appel à l'analyse non différentiable de Clarke [1983].

Un certain nombre de fonctions de mérite ont été proposées pour résoudre le problème (PVI) défini à partir de l'inégalité variationnelle $VI(F(x, \cdot), C(x))$ par algorithme itératif. Auslender [1976] a introduit pour la première fois la fonction g définie sur \mathbb{R}^{n_y} par:

$$g(y) = \text{Maximiser}_v \left\{ (y - v)^t F(x, y) : v \in C(x) \right\}. \quad (3.3.2)$$

Cette fonction a la propriété que son minimum sur $C(x)$ coïncide avec la solution du problème (PVI) que l'on peut mettre sous la forme d'un problème d'optimisation

$$\begin{aligned} & \text{Minimiser}_y g(y) \\ & \text{s.c. } y \in C(x). \end{aligned} \quad (3.3.3)$$

À partir de cette formulation, Marcotte [1985] propose un algorithme de descente pour résoudre un problème (PVI) monotone. Marcotte et Zhu [1996] établissent la convergence d'un algorithme de descente pour résoudre un problème d'inégalité variationnelle non strictement monotone. Fukushima [1992] a ensuite introduit la fonction de mérite définie sur \mathbb{R}^{n_y} par:

$$g(y) = \text{Maximiser}_v \left\{ -(v - y)^t F(x, y) - \frac{1}{2} (v - y)^t G(v - y) : v \in C(x) \right\} \quad (3.3.4)$$

où G est une matrice symétrique d'ordre n_y définie positive. Taji et Fukushima [1996] quant à eux proposent la fonction de mérite qui consiste à résoudre de manière séquentielle un programme quadratique; cette fonction est définie sur \mathbb{R}^{n_y} par:

$$g(y) = \text{Maximiser}_v \left\{ -(v-y)^t F(x,y) - \frac{1}{2}(v-y)^t G(v-y) : v \in T(x,y) \right\} \quad (3.3.5)$$

où $T(x,y)$ est le sous espace tangent en y à $C(x)$. Mastroeni [2003] propose une généralisation des fonctions de mérite précédentes pour résoudre des problèmes de complémentarité par un algorithme de descente.

3.4 Le problème de MPEC sous forme KKT

Dans cette section, nous nous intéressons particulièrement à la formulation KKT d'un programme mathématique avec contraintes d'équilibre (MPEC). La formulation (PDN_{KKT}) du problème de programmation mathématique à deux niveaux (PDN) telle que présentée au chapitre 2 est un problème de MPEC. De plus, la résolution d'un problème de PDN qui fait appel aux conditions d'optimalité du problème de second niveau utilise cette formulation. Nous commencerons par donner les définitions liées à cette formulation, puis nous parlerons des différentes hypothèses de qualification des contraintes qui sont utilisées pour ce type de problème; nous terminerons la section en faisant un tour d'horizon des différentes propriétés de cette famille particulière de problèmes de MPEC.

3.4.1 Formulation et définitions

La formulation KKT d'un problème de MPEC est la suivante:

(MPEC_{KKT})

$$\begin{aligned} & \text{Minimiser}_{x,y,\lambda} f^1(x,y) \\ & \text{s.c.} \begin{cases} (x,y) \in Z, \\ L(x,y,\lambda) = 0, \\ g^2(x,y) \leq 0, \\ \lambda \geq 0, \\ \lambda^t g^2(x,y) = 0, \end{cases} \end{aligned} \quad (3.4.1)$$

où $L(x,y,\lambda) = \nabla_y L^0(x,y,\lambda)$, avec $L^0(x,y,\lambda) = f^2(x,y) + \lambda^t g^2(x,y)$, $Z = X \times Y \subset \mathbb{R}^{n_x+n_y}$,

$f^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $i = 1,2$; $g^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^m$ et $\lambda \in \mathbb{R}^m$.

Considérons la formulation générale du problème de MPEC définie par (3.1.2); on suppose l'ensemble défini par $C(x) = \{y \in Y : h(x,y) = 0 \text{ et } g(x,y) \leq 0\}$ satisfait à une hypothèse de qualification des contraintes. Lorsque $C(x)$ est un ensemble convexe, la formulation KKT de type (3.4.1) est obtenue en remplaçant le problème de complémentarité généralisé $GCP(F(x,\cdot), C(x))$ par ses conditions d'optimalité de KKT.

Pour des raisons de simplicité dans l'exposé, nous avons omis les contraintes de type $h(x,y) = 0$ qui n'ont pas d'influence significative dans les différentes propriétés pour ce type de problème. Soit \mathfrak{S}^{KKT} l'ensemble des solutions réalisables du problème (MPEC_{KKT}). En utilisant les notations du chapitre 2, la formulation (PDN_{KKT}) d'un problème de PDN sous la forme particulière est identique à la formulation ci-dessus où $L(x,y,\lambda)$ représente le gradient par rapport à y du lagrangien $L^0(x,y,\lambda)$ du problème de second (PM2) niveau pour x fixé.

Soit $C : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ une application multivoque telle que

$$\forall x \in \mathbb{R}^{n_x}, C(x) \neq \emptyset \text{ et } C(x) = \{y \in \mathbb{R}^{n_y} : g^2(x,y) \leq 0\}.$$

On suppose qu'on a:

Hypothèse (H1): Chacune des fonctions $g_i^2(x, \cdot)$ est convexe pour tout $x \in X$ et g^2 est de classe C^2 sur un ensemble ouvert contenant \mathfrak{F}^{KKT} .

Les contraintes du problème ($MPEC_{KKT}$) représentent au sens de Luo et al. [1996b], les conditions de KKT en $y \in C(x)$ du problème de complémentarité (NCP) que nous avons définies au paragraphe 3.1.

Considérons l'application multivoque $\mathfrak{M} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^m$, qui à chaque $(x, y) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ associe $\mathfrak{M}(x, y)$ qui est l'ensemble des multiplicateurs de KKT du problème (NCP). Soit $I(x, y) = \{i : 1 \leq i \leq m, g_i^2(x, y) = 0\}$, l'ensemble des indices des contraintes actives pour tout $(x, y) \in \mathfrak{F}$. On a $I(x, y) \subseteq I$ et on peut donc représenter l'ensemble $\mathfrak{M}(x, y)$ des multiplicateurs de KKT du problème de second niveau par:

$$\mathfrak{M}(x, y) = \left\{ \lambda \in \mathbb{R}_+^m : \nabla_y f^2(x, y) + \sum_{i \in I(x, y)} \lambda_i \nabla_y g_i^2(x, y) = 0, \lambda_i = 0 \forall i \notin I(x, y) \right\}. \quad (3.4.2)$$

Le lagrangien du problème ($MPEC_{KKT}$) (au sens de Luo et al., 1996b; Scheel et Scholtes, 2000) est: $L^{MPEC}(x, y, \lambda, \pi, \mu) = f^1(x, y) + \pi^t L(x, y, \lambda) + \eta^t g^2(x, y)$, où $(\pi, \eta) \in \mathbb{R}^m \times \mathbb{R}_+^m$ sont les multiplicateurs de KKT du MPEC. C'est en fait le lagrangien correspondant au problème de MPEC relaxé des contraintes de complémentarité $\lambda^t g^2(x, y) = 0$. Si $\bar{z} = (\bar{x}, \bar{y})$ est une solution optimale locale du problème (MPEC), on sait (Scheel et Scholtes, 2000) qu'il existe un vecteur de multiplicateurs $(\bar{\pi}, \bar{\eta})$ tel que l'on a $\nabla_{z, \lambda} L^{MPEC}(\bar{z}, \bar{\lambda}, \bar{\pi}, \bar{\eta}) = 0$.

Si $\bar{w} = (\bar{z}, \bar{\lambda})$ est une solution réalisable du problème ($MPEC_{KKT}$), on considère les ensembles d'indices contenus dans $I = \{1, 2, \dots, m\}$ définis par:

$$I(\bar{z}) = \{i : g_i^2(\bar{z}) = 0\}, I_0(\bar{z}, \bar{\lambda}) = \{i \in I(\bar{z}) : \bar{\lambda}_i = 0\}, I_+(\bar{z}, \bar{\lambda}) = \{i \in I(\bar{z}) : \bar{\lambda}_i > 0\}.$$

Soit α un sous ensemble de $I_0(\bar{z}, \bar{\lambda})$ constitué d'éléments qu'on appelle **indices dégénérés**, $\bar{\alpha}$ son complémentaire dans $I_0(\bar{z}, \bar{\lambda})$, et $\bar{\alpha} \equiv I_+(\bar{z}, \bar{\lambda}) \cup \alpha$; définissons le sous ensemble J et son complémentaire dans I ainsi qu'il suit: $J \equiv \bar{\alpha} \cup \{i : g_i^2(\bar{z}) < 0\}$ et $J^c \equiv \alpha \cup I_+(\bar{z}, \bar{\lambda})$.

Lorsque $\forall i \in I, \lambda_i + g_i^2(x, y) \neq 0, \lambda \in \mathfrak{M}(x, y)$, on a $I_0(\bar{z}, \bar{\lambda}) = \emptyset$. Si l'ensemble Z est un polyèdre, considérons pour $\bar{w} = (\bar{z}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$, l'ensemble constitué de familles de paires de sous ensemble d'indices $K(\bar{z}, \bar{\alpha}) = \{(J, J^c) : J \equiv \bar{\alpha} \cup \{i : g_i^2(\bar{z}) < 0\}, J^c = I \setminus J\}$. Cette partition de I conduit à une famille de programmes non linéaires en (x, y, λ) :

(NLP_J)

$$\begin{aligned} & \text{Minimiser}_{x, y, \lambda} f^1(x, y) \\ & \left. \begin{array}{l} (x, y) \in X \times Y, \\ L(x, y, \lambda) = 0, \\ \left. \begin{array}{l} g_i^2(x, y) \leq 0 \\ \lambda_i = 0 \end{array} \right\} i \in J, \\ \left. \begin{array}{l} g_i^2(x, y) = 0 \\ \lambda_i \geq 0 \end{array} \right\} i \in J^c. \end{array} \right\} \text{s.c.} \end{aligned} \quad (3.4.3)$$

Définitions 3.8: Chacun des programmes non linéaires (NLP_J) est appelé **programme non linéaire branche** du problème (MPEC_{KKT}) et son ensemble des solutions réalisables est appelé **branche de solutions réalisables** du problème (MPEC_{KKT}) noté \mathfrak{F}^{NLP_J} .

$$\text{On a: } \mathfrak{F}^{KKT} = \bigcup_{J \subset I} \mathfrak{F}^{NLP_J}.$$

Comme en programmation mathématique à un niveau, les conditions d'optimalité du problème (MPEC) en un point réalisable z^0 sont définies à partir du **cône tangent** en

$z^0 = (x^0, y^0)$ à \mathfrak{F} noté $T(z^0, \mathfrak{F})$. Le concept de B-stationnarité défini ci-dessous est un critère d'optimalité relatif au problème de MPEC.

Définition 3.9: Un point réalisable z^* d'un problème de MPEC est dit **stationnaire primal** (ou optimal local) au sens de Luo et al. [1996b] ou **B-stationnaire** (B pour Bouligand; voir Scheel et Scholtes, 2000) si la condition suivante est satisfaite:

$$\nabla f^1(z^*)^t d \geq 0 \quad \forall d \in T(z^*, \mathfrak{F}) \text{ ce qui équivaut à } \nabla f^1(z^*) \in T(z^*, \mathfrak{F}).$$

Remarque 3.2: En d'autres termes, z^* est un optimum local pour le problème qui est obtenu en linéarisant par rapport à z toutes les fonctions qui interviennent dans le problème de MPEC.

La difficulté pratique pour établir la B-stationnarité en un point z^* pour un problème de MPEC réside dans la complexité (en particulier la non convexité) du cône tangent en z^* , $T(z^*, \mathfrak{F})$ qui est constitué de plusieurs cônes et que l'on peut mettre sous la forme:

$$T(z^*, \mathfrak{F}) = \bigcup_{(J, J^c) \in \mathfrak{J}(z^*)} T(z^*, \mathfrak{F}^{NLP_J})$$

où $T(z^*, \mathfrak{F}^{NLP_J})$ est le cône tangent en z^* de la branche (NLP_J) du problème ($MPEC_{KKT}$). Luo et al. [1996b], puis Pang et Fukushima [1999] consacrent une étude à la compréhension de ce cône: comme nous le verrons plus loin, la B-stationnarité en z^* pourrait être décrite sous la forme des conditions d'optimalité de KKT de chacune des branches du problème de MPEC.

3.4.2 Les hypothèses de qualification des contraintes

Tout comme en programmation mathématique classique, nous définissons un concept qui a trait à la régularité d'un problème de MPEC, à savoir les hypothèses de qualification des

contraintes qui ont une incidence importante sur la nature des propriétés relatives au problème de MPEC.

Définition 3.10: On dit que l'hypothèse (H) de qualification des contraintes est satisfaite en un point réalisable z^0 du problème (MPEC) si chacun des programmes (NLP_j) , $(J, J^c) \in K(z^0, \bar{\alpha})$ satisfait à l'hypothèse de qualification des contraintes (H) au point z^0 .

L'hypothèse (H) pourrait être l'hypothèse de l'indépendance linéaire des contraintes actives du problème de MPEC (LICQ) ou l'hypothèse de qualification des contraintes de Mangasarian-Fromovitz (MFCQ).

On peut noter que si le programme mathématique obtenu du problème de MPEC en relaxant ses contraintes de complémentarité satisfait à une hypothèse de qualification des contraintes (H), nécessairement chaque programme branche (NLP_j) de ce MPEC satisfait la même hypothèse (H).

Pour établir que l'ensemble $\mathfrak{M}(x, y)$ est uniformément borné, Luo et al. [1996b] ont introduit l'hypothèse de qualification des contraintes SBCQ (Sequentially Bounded Constraint Qualification) pour le problème (MPEC) qui se définit comme suit:

Définition 3.11: On dit que le problème de MPEC satisfait à l'**hypothèse de qualification des contraintes SBCQ** si pour toute suite de point $(x^k, y^k) \in \mathfrak{F}$, il existe un vecteur des multiplicateurs de KKT, $\lambda^k \in \mathfrak{M}(x^k, y^k)$ et si de plus, l'ensemble $\{\lambda^k\}$ est borné.

L'hypothèse de qualification des contraintes SBCQ joue un rôle important dans la caractérisation des solutions des problèmes de MPEC. Nous y reviendrons plus loin.

Remarque 3.3: La plupart des propriétés relatives aux problèmes de MPEC qui existent dans la littérature font référence à la décomposition d'un MPEC en une famille de programmes

mathématiques (PM) de type (NLP_J) , $(J, J^C) \in K(z^0, \bar{\alpha}_0)$ au voisinage d'une solution réalisable z^0 . Les conditions de régularité ou les conditions d'optimalité de cette famille de PM sont alors induites au problème de MPEC de départ. Les conditions d'optimalité d'un problème de MPEC proposées par Scheel et Scholtes [2000], notamment la B-stationnarité s'expriment en terme d'existence des multiplicateurs de KKT que l'on obtient à partir des sous problèmes (NLP_J) , $(J, J^C) \in K(z^0, \bar{\alpha}_0)$. Pour ce qui est précisément de l'hypothèse $(LICQ)$ utilisée comme hypothèse de qualification des contraintes du problème de MPEC, Sholtes et Stöhr [2001] font remarquer que les problèmes de MPEC pour lesquels l'hypothèse $LICQ$ n'est pas satisfaite sont des cas pathologiques rares en pratique; de plus l'hypothèse $LICQ$ peut être établie localement pour tout programme mathématique, moyennant une perturbation des données au voisinage d'une solution réalisable z . Nous émettons quelques réserves par rapport à cette assertion. En effet, lorsque l'hypothèse $LICQ$ n'est pas vérifiée pour le problème de départ, une perturbation des données ne change rien à cet état de fait si on fait appel aux méthodes d'analyse des erreurs d'arrondi ou de troncature (qui permettent de déterminer la notion de zéro informatique). Cette façon de faire poserait des problèmes de stabilité numérique et conduirait à des résultats erronés au cours des traitements numériques de ce type de problème (Laporte et Vignes, 1974 ; Etoa Etoa, 1987).

3.4.3 Caractérisation des solutions du problème de MPEC sous forme KKT

On pourrait être tenté de déterminer, à partir de chacun des programmes non linéaires branches du problème $(MPEC_{KKT})$ tel que formulé précédemment, les conditions d'optimalité de premier ou de second ordre, par exemple le cône tangent en \bar{z} pour chaque branche. Cependant, le cardinal de $\mathfrak{J}(\bar{z})$ peut être très grand et le dénombrement de ces branches exponentiel. Dans cette partie, nous décrivons une méthode pratique qui permet de donner une caractérisation d'un optimum local pour problème de MPEC sous forme KKT à partir de ses branches. Nous ferons appel à cette méthode dans un des algorithmes que nous allons développer pour résoudre un problème de type $(MPEC_{KKT})$ dans les chapitres 4 et 5.

Soit $T(\bar{z}, Z)$ le cône tangent en \bar{z} à Z . Luo et al. [1996b] ont proposé une condition nécessaire d'optimalité d'un problème de type $(MPEC_{KKT})$ en termes de stationnarité primale-duale à savoir: sous une hypothèse de qualification des contraintes sur le problème $(MPEC_{KKT})$ en $\bar{w} = (\bar{z}, \bar{\lambda})$, pour tous sous-ensembles $\alpha, \tilde{\alpha}$ et $\bar{\alpha}$ d'indices tels que définis plus haut, on a:

$$\left. \begin{aligned} dz &\in T(\bar{z}, Z), \\ d\lambda_i &\geq 0 \text{ pour } i \in \alpha, \\ \nabla_z L(\bar{z}, \bar{\lambda})^t dz + \sum_{i \in \tilde{\alpha}} d\lambda_i \nabla_y g_i^2(\bar{z}) &= 0, \\ \nabla g_i^2(\bar{z})^t dz &= 0 \text{ pour } i \in \bar{\alpha}, \\ \nabla g_i^2(\bar{z})^t dz &= 0 \text{ pour } i \in \tilde{\alpha}, \end{aligned} \right\} \Rightarrow \nabla f^1(\bar{z}) dz \geq 0. \quad (3.4.4)$$

En utilisant le théorème des alternatives de Farkas, Luo et al. en déduisent qu'il existe un vecteur des multiplicateurs (π, η) tel que l'on ait:

$$\begin{aligned} \nabla_x f^1(\bar{z}) + \nabla_x g^2(\bar{z})^t \eta &= \pi^t \nabla_x L(\bar{z}, \bar{\lambda}), \\ \nabla_y f^1(\bar{z}) + \nabla_y g^2(\bar{z})^t \eta &= \pi^t \nabla_y L(\bar{z}, \bar{\lambda}), \\ \pi^t \nabla_y g_i^2(\bar{z}) &\leq 0 \text{ pour } i \in \alpha, \\ \pi^t \nabla_y g_i^2(\bar{z}) &= 0 \text{ pour } i \in I_+(\bar{z}, \bar{\lambda}), \\ \eta_i &\geq 0 \text{ pour tout } i \in \bar{\alpha}, \\ \eta_i &= 0 \text{ pour tout } i \notin I(\bar{z}). \end{aligned} \quad (3.4.5)$$

En fait le système (3.4.5) ci-dessus est une formulation des conditions d'optimalité de KKT du programme branche (NLP_J) relatif à la partition d'indices $(\alpha, \bar{\alpha})$.

Définition 3.12: Le point $(\bar{x}, \bar{y}, \bar{\lambda})$ est **B-stationnaire** pour le problème $(MPEC_{KKT})$, si $d\mathbf{w} = 0$ est une solution pour le problème linéarisé dérivé du programme branche (NLP_J) pour tout sous ensemble d'indices J tel que $J \supseteq \{i : g_i^2(\bar{z}) < 0\}$ et $J^c \supseteq I_+(\bar{z}, \bar{\lambda})$:

$$\begin{aligned}
& \text{Min}_{dz, d\lambda} f^1(\bar{z}) + \nabla_z f^1(\bar{z})^t dz \\
& \left\{ \begin{array}{l} \bar{z} + dz \in Z, \\ L(\bar{z}, \bar{\lambda}) + \nabla_{(z, \lambda)} L(\bar{z}, \bar{\lambda})^t \begin{pmatrix} dz \\ d\lambda \end{pmatrix} = 0, \end{array} \right. \\
& \text{s.c.} \left\{ \begin{array}{l} g_i^2(\bar{z}) + \nabla_z g_i^2(\bar{z})^t dz \leq 0 \\ \bar{\lambda}_i + d\lambda_i = 0 \end{array} \right\} \forall i \in J, \\
& \left\{ \begin{array}{l} g_i^2(\bar{z}) + \nabla_z g_i^2(\bar{z})^t dz = 0 \\ \bar{\lambda}_i + d\lambda_i \geq 0 \end{array} \right\} \forall i \in J^c.
\end{aligned} \tag{3.4.6}$$

À l'instar de Scheel et Scholtes [2000] ou de Luo et al. [1996b] (proposition 3.3.9) nous pouvons remarquer que si l'hypothèse de qualification des contraintes *LICQ* est satisfaite au point $\bar{w} = (\bar{x}, \bar{y}, \bar{\lambda})$ ($\mathfrak{M}(\bar{z}) = \{\bar{\lambda}\}$), et si \bar{z} est une solution non dégénérée ($\lambda_i + g_i^2(\bar{z}) \neq 0 \forall i$), alors \bar{z} est stationnaire pour le problème (*MPEC_{KKT}*) si et seulement si les multiplicateurs (π, η) satisfont au système ci-après:

$$\begin{aligned}
& \nabla_x f^1(\bar{z}) - \nabla_x g^2(\bar{z})^t \eta = \nabla_x L(\bar{z}, \bar{\lambda})^t \pi, \\
& \nabla_y f^1(\bar{z}) - \nabla_y g^2(\bar{z})^t \eta = \nabla_y L(\bar{z}, \bar{\lambda})^t \pi, \\
& \eta_i = 0 \text{ si } i \notin I(\bar{z}), \\
& \pi^t \nabla_y g_i^2(\bar{z}) = 0 \text{ si } i \in I(\bar{z}).
\end{aligned} \tag{3.4.7}$$

La relaxation du problème (*MPEC_{KKT}*) que nous notons (*MPEC_{RKT}*) joue un rôle important dans la détermination des conditions d'optimalité du problème (*MPEC_{KKT}*) lorsqu'on a une solution dégénérée, c'est-à-dire lorsque l'ensemble des indices $I_0(\bar{z}, \bar{\lambda}) \neq \emptyset$. On fait appel aux conditions d'optimalité de cette relaxation pour mettre en œuvre un critère d'arrêt efficace pour toute procédure de résolution du problème (*MPEC_{KKT}*) qui utilise sa décomposition en branches (*NLP_J*). Nous y reviendrons dans les développements à venir. La formulation de la relaxation en question est:

$$\begin{aligned}
& (MPEC_{RKT}) \\
& \text{Minimiser}_{(x,y,\lambda) \in Z \times \mathfrak{M}(x,y)} f^1(x,y) \\
& \left\{ \begin{array}{l} (x,y) \in X \times Y, L(x,y,\lambda) = 0, \\ g_i^2(x,y) \leq 0 \\ \lambda_i = 0 \end{array} \right\} \forall i \notin I(\bar{z}), \\
& \text{s.c.} \left\{ \begin{array}{l} g_i^2(x,y) \leq 0 \\ \lambda_i = 0 \end{array} \right\} \forall i \in I_0(\bar{z}, \bar{\lambda}), \\
& \left\{ \begin{array}{l} g_i^2(x,y) = 0 \\ \lambda_i \geq 0 \end{array} \right\} \forall i \in I_+(\bar{z}, \bar{\lambda}).
\end{aligned} \tag{3.4.8}$$

Une solution réalisable (x,y,λ) pour le problème $(MPEC_{RKT})$ ne satisfait pas nécessairement le système des conditions KKT défini par (3.4.5) qui sont relatifs à la branche (NLP_j) , puisque aucune relation de complémentarité n'est imposée dans la formulation (3.4.8) pour λ_i et $g_i^2(x,y)$, $i \in I_0(\bar{z}, \bar{\lambda})$ (Luo et al., 1996b p.214). Toutefois, toute solution réalisable de la branche (NLP_j) définie par (3.4.3), suffisamment voisine de $(\bar{x}, \bar{y}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$ doit être réalisable pour le problème $(MPEC_{RKT})$. Donc $(MPEC_{RKT})$ est une relaxation du problème $(MPEC_{KKT})$ au voisinage de $(\bar{x}, \bar{y}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$.

3.4.4 Propriétés du problème de MPEC sous forme KKT

Nous énonçons présent les principales propriétés du problème de MPEC formulé sous la forme KKT dont nous nous servons dans les développements à venir.

Sous les hypothèses (H1) et SBCQ portant sur l'ensemble des solutions réalisables \mathfrak{F} du problème $(MPEC)$, on a $(x,y) \in \mathfrak{F}$ si et seulement si il existe $\lambda \in \mathfrak{M}(x,y)$ tel que $(x,y,\lambda) \in \mathfrak{F}^{KKT}$. De manière formelle, un point stationnaire du problème $(MPEC)$ est un vecteur $\bar{z} = (\bar{x}, \bar{y}) \in \mathfrak{F}$ tel que $\nabla_x f^1(\bar{z})dx + \nabla_y f^1(\bar{z})dy \geq 0$ pour tout $(dx, dy) \in T(\bar{z}, \mathfrak{F})$ où $T(\bar{z}, \mathfrak{F})$ désigne le cône tangent en \bar{z} à \mathfrak{F} .

La formulation KKT d'un problème de MPEC peut être considérée comme une formulation équivalente d'un programme mathématique à deux niveaux (PDN). La première propriété que nous énonçons est relative à l'existence d'une solution optimale pour un problème de MPEC sous forme KKT. Elle fait référence à la définition relative à la co-coercivité de la fonction f^1 .

Propriété 3.3 (Luo et al., 1996b; théorème 1.4.1): Soit Z un ensemble fermé, $f^1, g^2, \nabla_y f^2, \nabla_y g_i^2$ des fonctions continues. Soit $\mathfrak{F}^{KKT} \subset \mathbb{R}^{n_x+n_y} \times \mathbb{R}_+^m$ l'ensemble des solutions réalisables du problème $(MPEC_{KKT})$. S'il existe $(x^0, y^0, \lambda^0) \in \mathfrak{F}^{KKT}$ tel que l'ensemble $\{(x, y, \lambda) \in \mathfrak{F}^{KKT} : f^1(x, y) \leq f^1(x^0, y^0)\}$ est borné, alors le problème $(MPEC_{KKT})$ admet une solution optimale.

- a) Si l'hypothèse de qualification des contraintes SBCQ est satisfaite en \mathfrak{F} , chacune des fonctions $g_i^2(x, \cdot)$ étant convexe et s'il existe $(x^0, y^0) \in \mathfrak{F}$ tel que l'ensemble $\{(x, y) \in \mathfrak{F} : f^1(x, y) \leq f^1(x^0, y^0)\}$ est borné, alors le problème $(MPEC)$ admet une solution optimale.

Soit $\mathfrak{M}(x, y)$ l'ensemble des multiplicateurs de KKT du problème $(MPEC)$. Si on suppose que l'hypothèse de qualification des contraintes SBCQ et l'hypothèse (H1) ci-dessus sont satisfaites, alors la propriété ci-après donne une condition nécessaire et suffisante pour qu'un point de coordonnées (x, y) appartienne au domaine induit du problème (PDN) , et une condition nécessaire pour que le domaine des solutions réalisables soit un ensemble fermé.

Propriété 3.4 (Luo et al., 1996b): Considérons la formulation générale d'un problème de MPEC définie par (3.1.2); on suppose que F et g^2 sont des fonctions continues et que Z un ensemble fermé de $\mathbb{R}^{n_x+n_y}$; sous l'hypothèse (H1) et sous l'hypothèse de qualification des contraintes SBCQ sur le domaine des solutions réalisables \mathfrak{F} du problème $(MPEC)$ on a:

- a) Pour $z = (x, y) \in Z$, $y \in M(x)$ si et seulement si $\mathfrak{M}(x, y) \neq \emptyset$.

b) Si Z est un ensemble fermé, il en est de même pour l'ensemble \mathfrak{F} .

Lorsque l'hypothèse (SBCQ) n'est pas satisfaite, il peut en résulter la non fermeture du domaine \mathfrak{F} . Illustrons ceci par un exemple.

Exemple 3.2: Dans \mathbb{R}^2 , définissons: $F(x,y) = 1$, $Z = \{(x,y) : |x| < 1\}$ est convexe pour chaque x fixé. On peut aisément vérifier que

$$M(x) = \begin{cases} \{-1\} & \text{si } -1 \leq x \leq 0 \\ \{0\} & \text{si } -1 < x < 0 \end{cases}$$

et on a $\mathfrak{F} = \{(x,-1) : -1 \leq x \leq 0\} \cup \{(x,0) : -1 < x < 0\}$ n'est pas un ensemble fermé.

Lorsque $x < 0$, $y = 0$ est l'unique solution du problème (NCP), on peut montrer facilement qu'au point $(x', 0)$, avec $x' < 0$, le multiplicateur correspondant à la contrainte $xy \leq 0$ est égal à $-\frac{1}{x'}$ qui tend vers $-\infty$ lorsque $x' \rightarrow 0$.

La propriété suivante établit l'équivalence entre le problème (MPEC) et sa formulation (MPEC_{KKT}), sous les hypothèses de la propriété 3.4.

Propriété 3.5 (Luo et al., 1996b: théorème 1.3.5): Sous les hypothèses de la propriété 3.4, les problèmes (MPEC) et (MPEC_{KKT}) partagent le même ensemble de solutions.

Cette propriété nous donne une indication importante sur la relation qui existe entre l'ensemble des solutions d'un problème de PDN et sa formulation KKT.

Afin de formuler une condition nécessaire et suffisante d'optimalité de second ordre du problème (MPEC_{KKT}), nous supposons que les hypothèses (H1) et de qualification des contraintes SBCQ sont satisfaites pour le problème (MPEC). Si \bar{z} est un minimum local du problème (MPEC), $(\bar{z}, \bar{\lambda})$ est un point stationnaire du problème (MPEC_{KKT}) pour tout $\bar{\lambda} \in \mathfrak{M}(\bar{x}, \bar{y})$ d'après la propriété 3.3. Par conséquent, pour toute partition $(\alpha, \bar{\alpha})$ de $I_0(\bar{z}, \bar{\lambda})$, $(\bar{z}, \bar{\lambda})$ est un minimum

local du problème branche (NLP_J) défini par (3.4.3), avec $J \equiv \bar{\alpha} \cup \{i : g_i^2(\bar{z}) < 0\}$ et $J^c \equiv \alpha \cup I_+(\bar{z}, \bar{\lambda})$.

Une expression du cône critique du problème (NLP_J) en $\bar{w} = (\bar{x}, \bar{y}, \bar{\lambda})$ est la suivante:

$$\begin{aligned}
C_\alpha(\bar{w}) \equiv & \left\{ (dx, dy, d\lambda) \in T(\bar{z}, Z) \times \mathbb{R}^m : \nabla_x f^1(\bar{z})dx + \nabla_y f^1(\bar{z})^t dy \geq 0, \right. \\
& \nabla_x L(\bar{w})dx + \nabla_y L(\bar{w})dy + \sum_{i \in J^c} d\lambda_i \nabla g_i^2(\bar{z}) = 0, \\
& \nabla_x g_i^2(\bar{z})dx + \nabla_y g_i^2(\bar{z})dy = 0 \text{ pour } i \in J^c, \\
& \nabla_x g_i^2(\bar{z})dx + \nabla_y g_i^2(\bar{z})dy \leq 0 \text{ pour } i \in \bar{\alpha}, \\
& d\lambda_i = 0 \text{ pour } i \in J, \\
& \left. d\lambda_i = 0 \text{ pour } i \in \alpha \right\}.
\end{aligned} \tag{3.4.9}$$

Sous des hypothèses de qualification des contraintes standards, la propriétés 3.6 ci-dessous est une formulation des conditions d'optimalité de second ordre pour le sous problème (NLP_J).

Propriété 3.6 (Luo et al., 1996b: théorèmes 5.6.4 et 5.6.5): Soit $\bar{z} = (\bar{x}, \bar{y}) \in \mathfrak{F}$ et le vecteur des multiplicateurs correspondant $\lambda \in \mathfrak{M}(\bar{x}, \bar{y})$; soit α et $\bar{\alpha}$ une partition quelconque de l'ensemble des indices $I_0(\bar{w})$ et les ensembles complémentaires d'indices J et J^c de la définition 3.10. Supposons que l'hypothèse de qualification des contraintes $MFCQ$ est satisfaite en \bar{w} pour le problème (NLP_J).

1) Si \bar{w} est un optimum local pour le problème ($MPEC_{KKT}$), alors pour chaque $dw = (dx, dy, d\lambda) \in C_\alpha(\bar{w})$, il existe des multiplicateurs π et η qui satisfont le système (3.4.5) tels que l'on a $dw^t \nabla_{ww}^2 L^{MPEC}(\bar{w}, \pi, \eta) dw \geq 0$.

2) Si de plus on a $dw^t \nabla_{ww}^2 L^{MPEC}(\bar{w}, \pi, \eta) dw > 0$, alors il existe un voisinage $V \subseteq \mathbb{R}^{n_x + n_y + l}$ de \bar{w} et une constante $\gamma > 0$ tels que pour chaque $w \equiv (z, \lambda) \in \mathfrak{F}^{KKT}$, on a:

$$f'(z) \geq f'(\bar{z}) + \gamma \|w - \bar{w}\|^2 \geq f'(\bar{z}) + \gamma \|z - \bar{z}\|^2.$$

Les propriétés qui suivent se rapportent à la relaxation ($MPEC_{RKT}$) du problème ($MPEC_{KKT}$). Ces deux problèmes partagent le même ensemble de multiplicateurs de KKT; de plus, sous l'hypothèse de qualification des contraintes $MFCQ$ au voisinage d'une solution optimale de la relaxation ($MPEC_{RKT}$), ces deux problèmes ont le même ensemble des solutions. C'est en substance ce que dit la propriété ci-après.

Propriété 3.7 (Luo et al., 1996b: proposition 4.3.5): Un vecteur (π, η) satisfait aux conditions KKT du problème ($MPEC_{RKT}$) si et seulement si il satisfait au système (3.4.5) relatif aux conditions de KKT du problème ($MPEC_{KKT}$) pour tout ensemble d'indices $\alpha \subseteq I_0(\bar{z}, \bar{\lambda})$. Par conséquent, si l'hypothèse de qualification des contraintes $MFCQ$ est vérifiée au point $(\bar{z}, \bar{\lambda}, \bar{\eta}, \bar{\pi})$ qui satisfait aux conditions de KKT de la relaxation ($MPEC_{RKT}$) définies par (3.4.9), alors $(\bar{\eta}, \bar{\pi})$ est l'unique vecteur des multiplicateurs KKT qui vérifie les conditions KKT (3.4.5) du problème ($MPEC_{KKT}$) pour tout ensemble d'indices $\alpha \subseteq I_0(\bar{z}, \bar{\lambda})$.

La propriété ci-dessous renforce la propriété précédente lorsqu'on suppose l'hypothèse de qualification des contraintes $LICQ$ satisfaite en $(\bar{z}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$ pour la relaxation ($MPEC_{RKT}$) de chaque branche (NLP_j). Outre on a le fait que chaque branche (NLP_j) satisfait également à l'hypothèse $LICQ$, mais aussi les programmes ($MPEC_{RKT}$) et (NLP_j) ont même ensemble de solution.

Propriété 3.8 (Luo et al., 1996b: proposition 4.3.7): Si on suppose que l'hypothèse de qualification des contraintes $LICQ$ est satisfaite pour le problème ($MPEC_{RKT}$) en $(\bar{z}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$, alors la même hypothèse $LICQ$ est satisfaite pour chaque branche de ($MPEC_{KKT}$) en $(\bar{z}, \bar{\lambda})$ pour toute paire de sous ensembles d'indices $(J, J^c) \in K(\bar{z}, \bar{\alpha})$, étant donné un sous-ensemble

quelconque d'indices $\alpha \subseteq I_0(\bar{z}, \bar{\lambda})$. De plus, s'il existe (η, π) satisfaisant au système (3.4.9), alors $(\bar{z}, \bar{\lambda})$ est un point stationnaire du problème $(MPEC_{KKT})$.

3.5 Contribution à la définition des propriétés des solutions d'un problème de PDN

Considérons un problème de PDN; à titre de rappel, la variable du premier niveau x étant fixée, $M(x)$ désigne l'ensemble des solutions optimales pour le problème de second niveau. Dans cette section, nous présentons une contribution consacrée à l'étude des conditions sous lesquelles l'application $M(x)$ est univoque.

Une hypothèse d'univocité sur l'application $x \mapsto M(x)$ confère au problème de PDN des propriétés qui peuvent rendre aisée les analyses théoriques relatives à la conception d'algorithmes de résolution dudit problème. En effet, la convergence des algorithmes proposés par Bard et Moore [1990] ou par Muu et Quy [2003] s'appuient sur cette hypothèse. Il en est de même de la deuxième propriété de monotonie de l'algorithme MAYBIL de Savard [1989] ou encore de la méthode de pénalité de White et Anandalingam [1993]. Lorsque l'application $x \mapsto M(x)$ n'est pas univoque, Dempe [1996] propose une méthode d'approximation tangentielle du problème de second niveau afin de calculer une estimation d'une solution admissible d'un problème de PDN dans le cas général.

Considérons le problème de PDN ci-après:

$$\begin{aligned}
 (PDN) \quad & \text{Minimiser}_{x,y} f^1(x,y) \\
 & \text{s.c.} \begin{cases} x \in X, \\ \text{Minimiser}_y f^2(x,y) \\ \text{s.c.} \{y \in C(x), \end{cases} \quad (3.5.1)
 \end{aligned}$$

où

$$C(x) = \{y \in Y : g_i^2(x,y) \leq 0, i = 1, 2, \dots, m\} \text{ avec } X = \mathbb{R}^{n_x} \text{ et } Y = \mathbb{R}^{n_y}. \quad (3.5.2)$$

Pour $x \in X$ fixé, on suppose que chacune des fonctions $y \mapsto g_i^2(x, y)$, $i = 1, 2, \dots, m$ et $f^j: \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $j = 1, 2$ est convexe et de classe C^2 .

Supposons que $C(x)$ est un cône convexe de \mathbb{R}^{n_y} , $F(x, \cdot): \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_y}$ une application. Considérons le problème de complémentarité généralisé $GCP(C(x), F(x, \cdot))$ au sens de la définition 3.2. Karamardian [1971] a été le premier à établir la relation suivante entre le problème de complémentarité généralisé et le problème défini par l'inégalité variationnelle $VI(C(x), F(x, \cdot))$. Nous rappelons la dite relation à travers la proposition suivante.

Proposition 3.1 (Karamardian 1971): Soit $C(x)$ un cône convexe et $x \in X$ fixé; $y^* \in C(x)$ est une solution du problème défini par l'inégalité variationnelle $VI(C(x), F(x, \cdot))$ si et seulement si y^* est une solution du problème $GCP(C(x), F(x, \cdot))$.

Par conséquent, on peut affirmer que tout problème de complémentarité est un cas particulier d'un problème d'inégalité variationnelle. La réciproque n'est pas vraie d'une manière générale. Lorsque l'ensemble $C(x)$ définie par (3.5.2) satisfait à une hypothèse de qualification des contraintes et que chacune des fonctions $y \mapsto g_i^2(x, y)$, $i = 1, 2, \dots, m$ est affine (l'ensemble $C(x)$ est dans ce cas un polyèdre); l'inégalité variationnelle $VI(C(x), F(x, \cdot))$ peut être transformée en un problème de complémentarité généralisé. Le résultat ci-après inspiré de la proposition 2.2 de Harker et Pan [1990], résume dans quelles conditions cette transformation peut être effectuée. Ladite transformation a été utilisée dans différents contextes par de nombreux auteurs comme Eaves [1971], [1978], Tobin [1986], Qi et Magnanti [1989].

Proposition 3.2: Soit $x \in X$ fixé et $g^2(x, \cdot): \mathbb{R}^{n_y} \rightarrow \mathbb{R}^m$ de classe C^1 ; considérons l'ensemble $C(x)$ tel que défini par (3.5.2).

- a) Si y^* est une solution du problème défini par l'inégalité variationnelle $VI(C(x), F(x, \cdot))$ et si une hypothèse de qualification des contraintes est satisfaite pour l'ensemble $C(x)$ en y^* , alors il existe $\lambda^* \in \mathbb{R}^m$ tel que (y^*, λ^*) est une solution du problème de

complémentarité généralisé $GCP(\mathbb{R}^{n_y} \times \mathbb{R}_+^m, H_x)$ où $H_x : \mathbb{R}^{n_y+m} \rightarrow \mathbb{R}^{n_y+m}$ est définie par:

$$H_x(y, \lambda) = \begin{bmatrix} F(x, y) + \lambda^t \nabla_y g^2(x, y) \\ g^2(x, y) \end{bmatrix}. \quad (3.5.3)$$

b) Réciproquement, si chacune des fonctions $y \mapsto g_i^2(x, y)$, $i = 1, 2, \dots, m$ est convexe et si

(y^*, λ^*) est une solution du problème $GCP(\mathbb{R}^{n_y} \times \mathbb{R}_+^m, H_x)$, alors y^* est solution du problème défini par l'inégalité variationnelle $VI(C(x), F(x, \cdot))$.

La propriété ci-après quant à elle énonce les conditions d'existence et d'unicité d'une solution au problème défini par $VI(C(x), F(x, \cdot))$.

Propriété 3.9 (Harker et Pang, 1990: corollaire 3.2): Soit $x \in X$; supposons que l'ensemble $C(x)$ est non vide, convexe et fermé dans \mathbb{R}^{n_y} . Considérons l'application $F(x, \cdot) : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_y}$ définie par $y \mapsto F(x, y)$. Si l'application $F(x, \cdot)$ est fortement monotone sur $C(x)$, alors il existe une solution unique au problème défini par l'inégalité variationnelle $VI(C(x), F(x, \cdot))$.

Supposons que dans la formulation (PDN) ci-dessus, le problème de second niveau satisfait à une hypothèse de qualification des contraintes. Considérons dans ce cas la formulation KKT de ce problème, à savoir:

(PDKKT)

$$\begin{aligned} & \text{Minimiser}_{x, y, \lambda} f^1(x, y) \\ & \text{s.c.} \begin{cases} F(x, y) + \lambda^t \nabla_y g^2(x, y) = 0, \\ \lambda^t g^2(x, y) = 0, \\ g^2(x, y) \leq 0, \\ \lambda \geq 0, (x, y) \in X \times Y, \end{cases} \end{aligned} \quad (3.5.4)$$

où $\lambda \in \mathbb{R}_+^l$, $F(x, y) = \nabla_y f^2(x, y)$ pour tout $(x, y) \in X \times C(x)$.

Soit $x \in X$ fixé; trouver une solution réalisable (x, y^*, λ^*) au problème (PDN_{KKT}) telle que $(y^*, \lambda^*) \in C(x) \times \mathbb{R}_+^m$ revient à résoudre le problème de complémentarité généralisé $GCP(\mathbb{R}^{n_y} \times \mathbb{R}_+^m, H_x)$. Si $M(x)$ désigne l'ensemble des solutions optimales du problème de second niveau du problème (PDN), le théorème ci-après énonce les conditions sous lesquelles l'application $M(x)$ est univoque.

Théorème 3.1: Considérons le problème (PDN) dont la formulation est donnée par (3.5.1) et soit $x \in X$ fixé; supposons que l'on a: (i) $C(x)$ est un ensemble convexe non vide et fermé de \mathbb{R}^{n_y} qui satisfait à une hypothèse de qualification des contraintes en chacun de ses points, (ii) la fonction f^2 est de classe C^1 et la fonction $y \mapsto \nabla_y f^2(x, y)$ est fortement monotone sur $C(x)$. Alors l'application $M(x)$ est univoque.

Preuve: Soit $x \in X$; considérons la formulation KKT du problème (PDN) ainsi que le problème de complémentarité généralisé $GCP(\mathbb{R}^{n_y} \times \mathbb{R}_+^m, H_x)$. Soit y^* une solution du problème défini par une inégalité variationnelle $VI(C(x), F(x, \cdot))$. Comme l'ensemble $C(x)$ satisfait à une hypothèse de qualification des contraintes en y^* , l'hypothèse (ii) entraîne qu'il existe $\lambda^* \in \mathbb{R}_+^m$ tel que (y^*, λ^*) est une solution du problème de complémentarité généralisé $GCP(\mathbb{R}^{n_y} \times \mathbb{R}_+^m, H_x)$ d'après la proposition 3.2. Le vecteur (x, y^*, λ^*) est une solution réalisable du problème (PDN_{KKT}) et on a $(x, y^*) \in DI$; d'après la propriété 3.6, y^* est l'unique vecteur tel que $y^* \in M(x)$. ■

Soit $L(x, y, \lambda) = F(x, y) + \lambda^t \nabla_y g^2(x, y)$ le lagrangien du problème $VI(C(x), F(x, \cdot))$.

Kyparisis [1987] s'appuie sur le concept de solution perturbée d'un problème défini par une inégalité variationnelle pour établir des conditions suffisantes d'existence d'une solution continûment différentiable pour un problème $VI(C(x), F(x, \cdot))$. Le théorème ci-après est une adaptation au contexte des PDN du résultat établi par Kyparisis [1987] (théorème 2.1) dont la

démonstration est similaire aux théorèmes proposés par Fiacco [1987] (théorème 2.1) et par Tobin [1986] (théorème 3.1).

Théorème 3.2: Étant donné le problème (PDM), soit $x \in X$ tel que les fonctions $y \mapsto \nabla_y f^2(x, y)$ et $y \mapsto g_i^2(x, y), i = 1, 2, \dots, m$ sont de classe C^1 au voisinage de (x, y^*) , et que le problème de second niveau admet un vecteur des multiplicateurs de KKT $\lambda^* \in \mathbb{R}_+^m$. On suppose en outre que l'ensemble $C(x)$ satisfait à l'hypothèse d'indépendance linéaire des contraintes actives (LICQ) en (x, y^*) et que:

- (i) chacune des fonctions $y \mapsto g_i^2(x, y), i = 1, 2, \dots, m$ est convexe;
- (ii) la condition de second ordre (SOC) ci-après est satisfaite en $(x, y^*) \in X \times C(x)$ avec un vecteur des multiplicateurs de KKT $\lambda^* \in \mathbb{R}_+^m$:

$$d^t \nabla_y L(x, y^*, \lambda^*) d > 0 \text{ pour tout } d \neq 0 \text{ tel que}$$

$$\nabla_y g_i^2(x, y^*) d \leq 0 \text{ si } g_i^2(x, y^*) = 0,$$

$$\nabla_y g_i^2(x, y^*) d = 0 \text{ si } \lambda_i^* > 0;$$

- (iii) les relations de complémentarité sont strictes (RCS), c'est-à-dire

$$\lambda_i^* - g_i^2(x, y^*) \neq 0 \quad \forall i.$$

Alors, il existe localement une unique solution $y(x) \in C(x)$ et un vecteur des multiplicateurs de KKT $\lambda(x) \in \mathbb{R}_+^m$ tels que $(x, y(x), \lambda(x))$ est une solution réalisable du problème (PDN_{KKT}). De plus $y(x)$ est localement l'unique élément de l'ensemble $M(x)$.

Contrairement au théorème 3.1, l'univocité de l'application $M(x)$ est établie localement, et dépend essentiellement des propriétés (LICQ), (SOC) et (RCS) du problème de second niveau qui sont également locales.

Remarque 3.4: Si le problème (PDN_{KKT}) représente la formulation KKT d'un problème général de PDN linéaire, alors en utilisant la remarque 3.1 on a:

$$M \equiv \begin{bmatrix} 0 & B_2^t \\ -B_2 & 0 \end{bmatrix}$$

et la propriété 3.1 permet d'affirmer que, lorsque la matrice M est une P_0 -matrice, alors l'application $M(x)$ est univoque.

Étant donné une formulation particulière d'un problème de programmation mathématique linéaire à deux niveaux (PLDN), nous examinons à présent dans quelles autres conditions l'application $M(x)$ est univoque. Lorsque la variable de premier niveau x est fixée, considérons le problème de second niveau ci-après:

$$\begin{aligned} & PM2(x) \\ & \text{Maximiser}_y d_2^t y \\ & \text{s.c.} \begin{cases} A_2 x + B_2 y \leq b_2, \\ y \geq 0. \end{cases} \end{aligned} \quad (3.5.5)$$

Soit $C(x)$ l'ensemble des solutions réalisables de $PM2(x)$ que l'on suppose non vide. On a:

$$C(x) = \left\{ y \in Y = \mathbb{R}_+^{n_y} : B_2 y \leq b_2(x) \right\} \text{ où } b_2(x) = b_2 - A_2 x.$$

$M(x) = \arg \min_{\xi \in C(x)} d_2^t \xi$ désigne l'ensemble des solutions optimales du problème $PM2(x)$ et la fonction valeur $V : \mathbb{R}_+^{n_x} \rightarrow \mathbb{R}$ est définie par $x \mapsto V(x) = d_2^t y$, $y \in M(x)$.

Nous montrons que pour toute valeur de la variable de premier niveau x fixée, si l'ensemble $C(x)$ est non vide et borné, et que le programme $PM2(x)$ admet une solution optimale primale duale non dégénérée, alors l'application $M(x)$ est univoque. Le théorème ci-après énonce cette propriété.

Théorème 3.3: Considérons un problème de PLDN dont le problème de second niveau est un programme linéaire de type $PM2(x)$. La variable de premier niveau x étant fixée, on suppose que

l'ensemble $C(x)$ des solutions réalisables du problème $PM2(x)$ est non vide et borné. Une condition suffisante pour que l'application $M(x)$ soit univoque est que le programme linéaire $PM2(x)$ admette une solution optimale primale-duale non dégénérée.

Preuve: Soit $x \geq 0$ tel que $C(x) \neq \emptyset$ et J l'ensemble des indices de la variable $y(x)$, et B une base optimale. On peut décomposer l'ensemble J en $J = J_B \cup J_N$; J_B (respectivement J_N) représente l'ensemble des indices des variables en base (respectivement hors base). De la même manière, on peut décomposer la matrice des contraintes B_2 , la fonction économique d_2 et la variable $y(x)$ de $PM2(x)$ en:

$$B_2 = [B, N]; d_2 = [d_B \ d_N]; y(x) = \begin{bmatrix} y_B(x) \\ y_N(x) \end{bmatrix}.$$

Le vecteur des coûts réduits des variables hors base est donné par l'expression:

$$\Delta_N = d_N - d_B B^{-1} N$$

et la solution optimale s'écrit: $y_B(x) = B^{-1} b_2(x)$ et $V(x) = d_B^t y_B(x)$. Supposons que la base B soit primale duale dégénérée; on a alors l'une des situations suivantes:

$$\begin{aligned} i) \exists i_0 \in J_B \text{ tel que } y_B^{i_0}(x) &= 0, \\ ii) \exists j_0 \in J_N \text{ tel que } \Delta_N^{j_0}(x) &= 0. \end{aligned} \tag{3.5.6}$$

Dans le cas ii), supposons qu'il existe un indice $i_s \in \{1, 2, \dots, m\}$ tel que

$$\bar{N}_{i_s j_0} > 0 \text{ où } B^{-1} N = \bar{N} = (\bar{N}_{ij}).$$

Simulons une itération de la méthode du simplexe au cours de laquelle la variable d'indice j_0 entre dans la base alors que la variable d'indice i_s en sort de manière à obtenir une base $B1$.

La valeur de la fonction économique s'écrit:

$$V(x) = d_2^t y(x) = d_B^t y_B(x) + \sum_{j \in J_N} \Delta_N^j y_N^j(x) = d_B^t y_B(x) + \Delta_N^{j_0} y_N^{j_0}(x).$$

Une autre solution de base $B1$ avec $y_N^{j_0}(x) > 0$ est telle que $V(x) = d_2^t y'(x) = d_{B1}^t y'_{B1}(x)$ comme $\Delta_N^{j_0} = 0$. On a $\{y(x), y'(x)\} \subset M(x)$ et l'application $M(x)$ n'est pas univoque.

On aboutit à la même conclusion pour le cas i) en simulant une itération de l'algorithme dual du simplexe, ceci achève notre démonstration. ■

3.6 Les algorithmes de résolution des problèmes de MPEC

Les algorithmes de résolution des problèmes de MPEC, faudrait-il le souligner n'ont pas connu à l'état actuel de la recherche, la même avancée que la théorie relative aux conditions d'optimalité du problème de MPEC (Luo et al., 1996b). La formulation compacte (3.1.2) d'un MPEC se rapporte aux problèmes d'optimisation non différentiable. Outrata et Zowe [1995], dans l'algorithme qu'ils proposent pour résoudre ce type de MPEC, font appel aux concepts de l'analyse non différentiable. Mais, Jiang et Ralph [2000] relèvent que cette formulation n'a pas encore fait l'objet d'études crédibles. Il faudrait mentionner que, même si la convergence des algorithmes de résolution des problèmes de MPEC a été établie par leurs différents concepteurs, la solution obtenue par ces algorithmes est un optimum local dans la plupart des cas. Un traitement spécifique est parfois réservé à la contrainte de complémentarité selon les deux méthodes suivantes:

- a) La contrainte de complémentarité est pénalisée dans la fonction économique qui prend alors la forme $P_\alpha(x, y, \alpha) \equiv f(x, y) + \alpha \sqrt{F(x, y)^t y}$ où $\alpha > 0$ est un coefficient de pénalité. L'algorithme *PIPA* de Luo et al. [1996b] (Penalty Interior Point Algorithm) fait appel à cette transformation. Les auteurs proposent ensuite de remplacer la contrainte de complémentarité $0 \leq F(x, y) \perp y \geq 0$ par $F_i(x, y) y_i = 2\mu$, $i=1, 2, \dots, m$, μ étant un paramètre positif qui tend vers zéro pour le processus itératif généré par l'algorithme.

b) La contrainte $0 \leq F(x, y) \perp y \geq 0$ est remplacée par une équation du type $\Psi(w, y, \mu) = 0 \in \mathbb{R}^{n_y}$ qui est construite à partir d'une fonction de mérite Φ , où $w = F(x, y)$. Dans ce cas, la fonction Ψ est définie de $\mathbb{R}^{n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}_+ \rightarrow \mathbb{R}^{n_y}$ par:

$$\Psi(w, y, \mu) = \begin{bmatrix} \Phi(w_1, y_1, \mu) \\ \Phi(w_2, y_2, \mu) \\ \dots \\ \Phi(w_{n_y}, y_{n_y}, \mu) \end{bmatrix}. \quad (3.6.1)$$

Pour $i = 1, 2, \dots, n_y$, $\Phi(w_i, y_i, \mu)$ désigne l'approximation de la relation de complémentarité $w_i y_i = 0$ par la fonction de mérite Φ . Le problème qui en résulte a pour formulation:

($MPEC_{\Phi_\mu}$)

$$\begin{aligned} & \text{Minimiser } f(x, y) \\ & \text{s.c. } \begin{cases} g(x, y) \leq 0, \\ h(x, y) = 0, \\ F(x, y) - w = 0, \\ \Psi(w, y, \mu) = 0, \\ w, y \geq 0. \end{cases} \end{aligned} \quad (3.6.2)$$

Les algorithmes proposés par Fukushima et al. [1998], Jian et Ralph [1999] ou Facchinei et Qi [1999] font appel à ce type de transformation: le problème ($MPEC_{\Phi_\mu}$) est un programme mathématique localement régulier qui, au voisinage d'une solution optimale, a même solution que le problème ($MPEC$). Le processus itératif est mis en oeuvre de manière à faire tendre vers zéro la suite relative au paramètre μ .

Notons cependant que ces méthodes ne fonctionnent qu'au voisinage d'une solution optimale. Mais on peut relever une contrainte majeure pour ce type de transformation: le choix judicieux d'une fonction de mérite qui ne détruirait pas la convergence locale.

La quasi-totalité des algorithmes qui ont été développés pour résoudre les problèmes de complémentarité font appel au type de transformation analytique ci-dessus (Harker et Pang, 1988; Andreani et Martinez, 1998; Burke et Xu, 2000; Hotta et Yoshise, 1999; Qi et Jiang, 1997). On peut classer les algorithmes de résolution des MPEC en deux catégories: les méthodes d'optimisation séquentielle et les méthodes d'optimisation classique.

3.6.1 Les méthodes d'optimisation séquentielle

Le principe des méthodes d'optimisation séquentielle est le suivant: à chaque itération k de solution z^k , une direction de déplacement est obtenue à partir du point z^k en résolvant un programme mathématique dérivé d'une approximation tangentielle du problème d'origine au voisinage de z^k . Illustrons le cadre général d'un algorithme d'optimisation quadratique séquentielle (SQP) en considérant la formulation générale du problème de MPEC linéaire. Soit $z^k = (x^k, y^k, w^k)$ une solution qui vérifie: $w^k = Nx^k + My^k + q$ et $(x^k, y^k, w^k) \in X \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}$. Par ailleurs, soit μ_k et B_k une matrice symétrique définie positive de dimension $(n_x + 2n_y)$. On résout alors à chaque itération le programme quadratique convexe ci-après de variables (dx, dy, dw) que l'on note $QP(z^k, \mu_k)$:

$$(QP(z^k, \mu_k))$$

$$\begin{aligned} & \text{Minimiser}_{dx, dy, dw} \nabla f(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} + \frac{1}{2} (dx, dy, dw)^t B_k \begin{pmatrix} dx \\ dy \\ dw \end{pmatrix} \\ & \text{s.c.} \begin{cases} x^k + dx \in X, \\ dw = Ndx + Mdy, \\ \Psi(y^k, w^k, \mu_k) + \nabla \Psi(y^k, w^k, \mu_k)^t \begin{pmatrix} dy \\ dw \end{pmatrix} = 0. \end{cases} \end{aligned} \quad (3.6.3)$$

La matrice B_k de l'algorithme PSQP proposé par Luo et al. [1998] est:

$$B_k = \frac{1}{2} \nabla_{ww}^2 L^{MPEC}(x^k, y^k, w^k, \lambda^k).$$

C'est en fait la matrice hessienne du lagrangien du problème (*MPEC*) constituée des dérivées secondes de son lagrangien L^{MPEC} par rapport à w , et λ^k est un vecteur des multiplicateurs de KKT du problème (*MPEC*) au point (x^k, y^k, w^k) . Le choix de la matrice B_k est crucial, car la convergence de l'algorithme en dépend. À chaque itération, il faut calculer λ^k le vecteur des multiplicateurs de KKT afin de déterminer la matrice B_k , ce qui peut augmenter le temps des calculs. Parmi les algorithmes de cette classe, nous décrivons brièvement les algorithmes de Luo et al. [1996a], Fukushima et al. [1998], Luo et al. [1998] et Zang et Liu [2001]. Les algorithmes qui intègrent une méthode d'optimisation séquentielle ont généralement une convergence locale super linéaire et parfois quadratique. Cependant, on peut relever une difficulté pour les algorithmes de cette classe: certains sous problèmes traités pourraient être sans solution.

L'algorithme *PIPA* (Luo et al., 1996a)

Avant de donner une description de l'algorithme *PIPA*, nous commençons par décrire certaines de ses spécificités, notamment pour ce qui est de l'actualisation du pas de déplacement à chaque itération. Considérons la formulation générale du problème de *MPEC*; la fonction objectif pénalisée ici est $P_\alpha(x, y, w)$ définie par (3.6.1). L'actualisation de la pénalité α et le calcul du pas de déplacement τ_k à chaque itération sont faits en appliquant la règle d'Armijo à la fonction objectif pénalisée de manière à satisfaire les conditions suivantes: (i) positivité des variables, (ii) décroissance de la relation de complémentarité et (iii) cheminement intérieur.

L'algorithme *PIPA* se présente ainsi qu'il suit:

- Dans la phase d'initialisation, on attribue une valeur arbitraire fixe aux différentes variables structurelles et techniques qui interviennent dans le programme, et particulièrement à la matrice symétrique définie positive B_0 ;
- On détermine ensuite une direction de déplacement en trouvant une solution optimale (dx^k, dy^k, dw^k) au sous problème $QP(z^k, \mu_k)$;

- Si le critère d'optimalité préalablement défini n'est pas satisfait, on procède à une actualisation de la pénalité et au calcul (a) du pas de déplacement (b) de la solution de l'itération suivante, puis on actualise la matrice symétrique définie positive B_{k+1} .

L'algorithme *PIPA* a généralement une convergence linéaire locale; cette convergence est quadratique si la matrice hessienne $\nabla^2 f^1(z^*)$ est définie positive au voisinage d'un point stationnaire z^* .

L'algorithme de Fukushima et al. [1998]

Cet algorithme a été élaboré pour résoudre les problèmes de MPEC linéaires; cet algorithme a une conception similaire à l'algorithme *PIPA*. Les légères différences observées entre les deux algorithmes sont tributaires du traitement spécifique réservé aux contraintes d'équilibre. Les auteurs proposent d'approximer les contraintes d'équilibre par la fonction de mérite de Fisher Burmeister que nous avons définie dans la section 3.3, et de pénaliser la fonction économique par les contraintes d'équilibres ainsi approximées; la fonction économique qui en résulte est définie par $\theta(x, y, \alpha, \mu) = f^1(x, y) + \alpha \sum_i |\Phi(y_i, w_i, \mu)|$ où α est le coefficient de pénalité. Les hypothèses suivantes sont faites:

Hypothèse A1: La suite $\{z^k = (x^k, y^k)\}$ générée par l'algorithme est bornée;

Hypothèse A2: La matrice M est une P_0 -Matrice.

Pour calculer une direction de déplacement (dx, dy, dw) à chaque itération k , il faut résoudre le programme quadratique défini par $QP(z^k, \mu_k)$, dont les multiplicateurs KKT sont $(\lambda^k, u^k, v^k) \in \mathbb{R}^{m_p + 2n_y}$. Fukushima et al. ont mis en œuvre leur algorithme en utilisant entre autres la matrice symétrique semi définie positive de la fonction économique du programme $QP(z^k, \mu_k)$ définie par $B_k = \begin{pmatrix} B_{xx} & 0 \\ 0 & 0 \end{pmatrix}$, où $B_{xx} = I_{n_x}$ est la matrice identité d'ordre n_x . Le pas de déplacement est calculé à partir de la fonction économique pénalisée $\theta(x^k, y^k, \alpha_k, \mu_k)$ en

utilisant la règle d'Armijo. Le coefficient de pénalité α_k et le paramètre μ_k sont actualisés en posant:

$$\mu_{k+1} = \beta \mu_k, \beta \in]0,1[\text{ et } \alpha_{k+1} = \begin{cases} \alpha_k & \text{si } \alpha_k \geq \|v^k\|_\infty + 1 \\ \max\{\|v^k\|_\infty + 1, \alpha_k + 2\} & \text{sinon} \end{cases}.$$

Le processus itératif s'arrête si l'on a: $\|\Phi(y^k, w^k, \mu_k)\|_\infty + \|dx\|_\infty \leq \varepsilon$.

Les auteurs montrent que le problème $QP(z^k, \mu_k)$ a une solution optimale unique. De plus, si $\bar{z} = (\bar{x}, \bar{y})$, limite de la suite des points z^k est une solution réalisable fortement non dégénérée ($\bar{y}_i + \bar{w}_i \neq 0$ pour tout i), alors \bar{z} est un point stationnaire du problème (MPEC).

Des programmes mathématiques à deux niveaux de petites tailles et des problèmes de MPEC de grande taille ont été résolus par cet algorithme. Les auteurs ont également procédé à une étude comparative de leur algorithme avec l'algorithme *PIPA*; il en ressort que ces deux méthodes de résolution des MPEC ont des comportements similaires. Nous reviendrons dans le chapitre 4 sur les expérimentations numériques portant sur l'algorithme de Fukushima et al.

L'algorithme *PSQP* de Luo et al. [1998]

L'algorithme *PSQP* (Piecewise Sequential Quadratic Programming) de Luo et al. au même titre que l'algorithme de Hansen et al. [1992] est de nature combinatoire et permet de déterminer un optimum global du problème de MPEC; il s'appuie sur la décomposition du problème de type (MPEC_{KKT}) en branches (NLP_J): à chaque itération k , le programme défini par la branche (NLP_{J_k}) est résolu par l'algorithme *SQP* où (J_k, J_k^C) est une partition de l'ensemble des indices $K((z^k, \lambda^k), \bar{\alpha}_k)$. Soit $I_0(z^k, \lambda^k) = \{j : \lambda_j^k = g_j^2(z^k) = 0\}$. Le processus itératif s'arrête à une itération k quelconque de solution réalisable $w^k = (z^k, \lambda^k)$ lorsqu'on a $dw = 0$ et :

- (1) soit $K(w^k, \bar{\alpha}_k) \neq \emptyset$; on examine alors les conditions d'optimalité pour chaque branche (NLP_J) et il y en a 2^β , β étant le cardinal de l'ensemble $I_0(w^k)$;

(2) soit on a $K(w^k, \bar{\alpha}_k) = \emptyset$ ou alors $w^k = (z^k, \lambda^k)$ est un point stationnaire de la relaxation ($MPEC_{RKT}$) du problème ($MPEC_{KKT}$).

Les auteurs ont établi que cet algorithme a une convergence locale super linéaire et parfois quadratique lorsque les conditions suffisantes d'optimalité du second ordre sont satisfaites en $\bar{w} = (\bar{z}, \bar{\lambda})$, même lorsqu'on utilise la formulation implicite d'un problème de MPEC de type (3.1.2). Cet algorithme se révèle performant d'après les auteurs. Les expérimentations numériques faites par Jiang et Ralph [1999] ont montré que les algorithmes du type *PSQP* ont une convergence rapide.

L'algorithme de *PSQP* de Zhang et Liu [2001]

Faisant référence à l'algorithme de résolution de programme linéaire quadratique à deux niveaux de Vicente et al. [1994b], Zhang et Liu proposent un algorithme d'exploration des points extrêmes du domaine réalisable \mathfrak{F} d'un problème de MPEC à contraintes linéaires, dont la fonction objectif est supposée convexe et de classe C^1 . Les auteurs supposent en outre que chacune des faces non vide du domaine \mathfrak{F} contient au moins un point extrême.

Le principe de l'algorithme est le suivant: à chaque itération k de l'algorithme, soit $z^k = (x^k, y^k)$ la solution réalisable courante; on détermine une direction de descente extrême ou alors un point extrême \bar{z} adjacent à z^k tel que $\bar{z} \in S(z^k)$ qui est la plus petite face de \mathfrak{F} contenant z^k et telle que $f(\bar{z}) \leq f(z^k)$. Dans tous les cas, on a $f(z^{k+1}) \leq f(z^k)$ si on trouve une direction de descente extrême.

Lorsque le problème (*MPEC*) linéaire a une solution, l'algorithme s'arrête si on ne peut pas trouver une direction de descente ou le point extrême \bar{z} . On trouve la solution de l'itération suivante en résolvant par une méthode *SQP* le problème d'optimisation (*P*) défini par:

$$\begin{aligned} & \text{Minimiser}_{x,y} f(x,y) \\ & \text{s.c. } (x,y) \in S(z^k). \end{aligned}$$

Les auteurs qui ont effectué une série de tests numériques prometteurs, ont montré que leur algorithme converge en un nombre fini d'itérations vers un optimum local.

3.6.2 Les méthodes d'optimisation classique

Parmi les algorithmes de résolution d'un problème de MPEC qui utilisent une méthode d'optimisation classique, on peut citer l'algorithme de Facchinei et Qi [1999] et l'algorithme BITrust de Marcotte et al. [2001]; nous donnons une description sommaire de ces deux algorithmes.

L'algorithme de Facchinei et Qi [1999]

L'algorithme de Facchinei et Qi s'applique aux problèmes de MPEC non linéaires qui ont une formulation du type (PDN_{KKT}) définie dans le chapitre précédent. Le problème de MPEC de départ qui est non différentiable en raison de la contrainte de complémentarité, aura été préalablement transformé en un problème d'optimisation continue de type $(MPEC_{\Phi, \mu})$ qu'on résout par un algorithme d'optimisation non linéaire approprié. Facchinei et Qi montrent que si le problème $(MPEC)$ a une solution optimale, chacun des sous problèmes $(MPEC_{\Phi, \mu})$ a une solution optimale qui satisfait aux conditions KKT. Soit $w^k = (x^k, y^k)$ la séquence des points générés par l'algorithme. Cet algorithme se présente ainsi qu'il suit:

- Considérer $\{\mu^k\}$ une séquence de nombres non nuls telle que $\lim_{k \rightarrow \infty} \mu^k = 0$ Dans la phase initiale, choisir $w^0 = (x^0, y^0) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$.
- Trouver ensuite une solution optimale (locale ou globale) au programme $(MPEC_{\Phi, \mu})$ et actualiser le paramètre μ^k .
- L'algorithme s'arrête lorsque que μ^k est voisin de zéro.

Les auteurs supposent que l'hypothèse de qualification des contraintes LICQ est satisfaite en tout point w^* , solution optimale du problème $(MPEC)$, et que son domaine réalisable est compact; lorsque la fonction F est fortement monotone, les auteurs montrent que la séquence

des solutions $\{w^k\}$ converge vers une approximation de la solution optimale du problème (MPEC).

Le critère d'arrêt que les auteurs utilisent s'appuie sur la décroissance de la séquence $\{\mu^k\}$ et la norme du lagrangien du problème $(MPEC_{\Phi_{\mu}})$.

Le seul inconvénient que l'on peut relever pour cet algorithme est qu'il faut résoudre à chaque itération k , un problème aussi difficile à résoudre que le problème initial. Cependant, la solution de départ de chacun des sous problèmes $(MPEC_{\Phi_{\mu}})$ lorsque k devient grand se rapproche davantage de la solution optimale w^* , puisqu'une solution de départ du problème courant est la solution optimale du problème précédent, de sorte que le nombre d'itérations nécessaires à la résolution de chaque sous problème décroît.

L'algorithme BITrust de Marcotte et al. [2001]

Il s'applique aux problèmes de MPEC de la forme:

(MPECF)

$$\begin{aligned} & \text{Minimiser}_{x \in X, y \in Y(x)} f(x, y) \\ & \text{s.c. } \{ \langle F(x, y), y - y' \rangle \leq 0 \quad \forall y' \in Y(x), \end{aligned}$$

où F est une fonction fortement monotone par rapport à la variable du second niveau y ; les ensembles X et $Y(x) = \{y : Ax + By \geq b\}$ sont des polyèdres. La notation $\langle a, b \rangle$ désigne le produit scalaire de a par b . Le principe de l'algorithme consiste à transformer le problème de MPEC en une suite de problèmes de PDN voisins faciles à résoudre: on procède à une approximation tangentielle des contraintes et de la fonction économique en obligeant la variable du premier niveau x à se situer à une distance d'au plus ε du point courant \bar{x} . À La première étape de l'algorithme, l'inégalité variationnelle qui définit le système des contraintes est remplacée par ses conditions de KKT; c'est un programme mathématique à variables mixtes qui peut être résolu par un algorithme d'énumération implicite. Les expérimentations numériques de cet algorithme ont été réalisées par Colson et al. [2002].

3.6.3 Une méthode d'optimisation globale: l'algorithme de Thoai et al. [2005]

L'algorithme GOA (Global Optimization Algorithm) proposé par Thoai et al. s'inspire des techniques de l'optimisation globale et s'applique aux problèmes de MPEC du type:

(P)

$$\begin{aligned} & \text{Minimiser } f(z, \lambda) \\ & \text{s.c. } \begin{cases} z \in Z, \\ (z, \lambda) \in \Omega, \\ Bz + w + b = 0, \\ \lambda^t w = 0, \\ \lambda \geq 0, w \in W \subset \mathbb{R}^p, \end{cases} \end{aligned}$$

où $w = C\lambda$, C étant une P_0 -matrice, Z un polyèdre borné de \mathbb{R}^n , Ω un ensemble convexe fermé de \mathbb{R}^{n+p} , $f: \mathbb{R}^{n+p} \rightarrow \mathbb{R}$ une fonction continue, $B \in \mathbb{R}^{p \times n}$ et $b \in \mathbb{R}^p$. L'algorithme GOA génère une séquence de points réalisables $\{(z^k, \lambda^k, w^k)\}_{k=1,2,\dots}$ du problème (P) telle que, pour chaque k , $f(z^k, \lambda^k)$ est une borne inférieure de la solution optimale du problème (P). Soit S un polyèdre contenant W dont l'ensemble des points extrêmes est $V(S) = \{s^1, s^2, \dots, s^\pi\}$. Les

variables λ et $w = Cz$ sont remplacées par $\lambda = \sum_{i=1}^{\pi} \lambda^i$, et par $w = \sum_{i=1}^{\pi} \beta_i s^i$, avec

$\sum_{i=1}^{\pi} \beta_i = 1$, $\beta_i \geq 0$; ce qui conduit au problème relaxé P(S) qui a pour formulation:

P(S)

$$\begin{aligned} & \text{Minimiser } f(z, \sum_{i=1}^{\pi} \lambda^i) \\ & \text{s.c. } \begin{cases} z \in Z, (z, \sum_{i=1}^{\pi} \lambda^i) \in \Omega, \\ Bz + \sum_{i=1}^{\pi} \beta_i s^i + b = 0, \\ \lambda^i \left(\sum_{i=1}^{\pi} s^i \right)^t = 0, \\ \lambda^i \geq 0, i = 1, 2, \dots, \pi, \beta \geq 0. \end{cases} \end{aligned}$$

Le principe de l'algorithme GOA consiste à résoudre une suite de programmes relaxés $P(S^k)$ définie à partir d'une séquence de polytopes $\{S^k\}_{k=1,2,\dots}$ dont les solutions sont telles que

$$\{(z^k, \lambda^k, w^k)\}_{k=1,2,\dots} = \{(z(s^k), \lambda(s^k), w(s^k))\}_{k=1,2,\dots}$$

avec:

- (i) soit (z^k, λ^k, w^k) est un optimum global du problème (P) pour un certain k ,
- (ii) soit chaque point d'accumulation de $\{(z^k, \lambda^k, w^k)\}_{k=1,2,\dots}$ est une estimation de l'optimum global du problème (P) .

Les tests numériques effectués par les auteurs ont porté sur des problèmes de PDN générés aléatoirement. Mais aucune comparaison des résultats obtenus par les auteurs n'a été faite avec des résultats d'autres méthodes de résolution des problèmes de PDN ou de MPEC.

Jusqu'à présent, les algorithmes de résolution des problèmes de MPEC conduisent généralement à un optimum local. Une méthode efficace qui exploiterait de manière énumérative toute l'information fournie par les contraintes d'équilibre, pourrait conduire à un optimum global.

CHAPITRE 4 – RÉOLUTION DES PROBLÈMES DE MPEC PAR UN ALGORITHME DE PROGRAMMATION LINÉAIRE SÉQUENTIELLE (*SLP*)

Dans ce chapitre, nous présentons une contribution à la résolution des programmes mathématiques avec contraintes d'équilibre (MPEC), lorsque les contraintes du MPEC en question sont linéaires et constituent un problème de complémentarité généralisé. Nous proposons un algorithme de programmation linéaire séquentielle (*SLP*) capable de déterminer un point stationnaire d'un problème de MPEC à contraintes linéaires. Nous intégrons à l'algorithme *SLP* une fonction de mérite de Fisher Burmeister de paramètre μ , qui transforme analytiquement le problème de MPEC en un programme mathématique classique localement régulier; une méthode originale d'actualisation dudit paramètre est proposée, en fonction des seules données du problème résolu. La formulation KKT de certains problèmes de programmation mathématique à deux niveaux (PDN) peut également être transformée en un programme mathématique classique localement régulier dont une solution réalisable peut être déterminée par un algorithme de type *SLP*.

Le chapitre est organisé comme suit: la section 4.1 est consacrée aux fondements d'une méthode de programmation linéaire séquentielle. En nous inspirant des travaux de Pironneau et Polak [1973], nous montrons que l'algorithme *SLP* a une convergence linéaire; nous mettons en exergue une erreur qui s'est glissée dans le résultat de Pironneau et Polak [1973]. Dans la section 4.2, nous présentons une méthode de transformation d'une classe de problèmes de PDN sous forme KKT en programme mathématique classique localement régulier, puis nous établissons les conditions d'optimalité d'un tel problème. Dans la section 4.3, en référence à la fonction de mérite de Fisher Burmeister qui est utilisée dans notre modélisation, nous montrons que le paramètre μ qui intervient dans cette fonction peut être actualisé en utilisant exclusivement les données du problème. Dans la section 4.4, nous adaptons l'algorithme *SLP* à la résolution du problème de complémentarité linéaire. Nous terminons le chapitre avec une description des résultats des expérimentations numériques de l'algorithme *SLP* qui portent sur les problèmes de MPEC, puis sur les problèmes de complémentarité généralisés linéaires. Nous comparons les performances

de l'algorithme *SLP* à celles de l'algorithme proposé par Fukushima et al. [1998], en procédant à une illustration graphique des conséquences du choix de l'actualisation du paramètre qui intervient dans la fonction de mérite utilisée.

4.1 Cadre conceptuel d'un algorithme de programmation linéaire séquentielle:

4.1.1 Généralités

Considérons le programme non linéaire ci-après:

(NLP)

$$\begin{aligned} & \text{Minimiser}_x f^1(x) \\ & \text{s.c.} \begin{cases} x \in \mathbb{R}^n, \\ g_i^2(x) \leq 0 \quad i \in I, \\ g_i^2(x) = 0 \quad i = I_1, \end{cases} \end{aligned} \quad (4.1.1)$$

où $|I_1 \cup I| = m$, f^1 et $g^2 = (g_1^2, g_2^2, \dots, g_m^2)$ sont des fonctions convexes de classe C^2 ; son ensemble des solutions réalisables est noté \mathfrak{F}^{NLP} . Soit $\bar{x} \in \mathfrak{F}^{NLP}$ une solution réalisable du problème (4.1.1), $\mathfrak{M}(\bar{x})$ l'ensemble $\eta \in \mathbb{R}^m$ des multiplicateurs de Lagrange Kuhn et Tucker (LKT), et $I(\bar{x}) \cup I_1$ l'ensemble des contraintes actives en \bar{x} , avec $I(\bar{x}) = \{i \in I : g_i^2(\bar{x}) = 0\}$. Le lagrangien du problème (4.1.1) est:

$$\mathfrak{L}^{NLP}(x, \eta) \equiv f^1(x) + \eta^t g^2(x). \quad (4.1.2)$$

Dans le cas où les gradients de l'ensemble des contraintes actives en \bar{x} sont linéairement indépendants, on a $\mathfrak{M}(\bar{x}) = \{\bar{\eta}\}$. Lorsque le sous ensemble des indices I_1 est vide dans (4.1.1), Zoutendijk [1960] a proposé un algorithme de descente pour résoudre ce problème; dans ce cas, à partir du point x^k , une direction de descente dx est obtenue en résolvant le problème:

(TNLP)

$$\begin{aligned} & \text{Minimiser}_{dx} \nabla f^1(x^k)^t dx \\ & \text{s.c.} \begin{cases} \nabla g_i^2(x^k)^t dx \leq 0 \quad i \in I(x^k), \\ \|dx\| = 1. \end{cases} \end{aligned} \quad (4.1.3)$$

Seulement, d'une manière générale, la direction dx ainsi calculée peut être telle qu'un petit déplacement infinitésimal dans cette direction fasse sortir immédiatement de l'ensemble des solutions réalisables \mathfrak{F}^{NLP} (Minoux 1983). Tout comme, Topkis et Veinott [1967], nous suggérons de remplacer le problème du type (4.1.3) par le problème (très voisin) de programmation linéaire ci-dessous, adapté la formulation (4.1.1):

$$\begin{aligned} & PL^0(x^k) \\ & \text{Maximiser}_{dx, \xi} \xi \\ & \text{s.c.} \begin{cases} \nabla f^1(x^k)^t dx + \xi \leq 0, \\ g_i^2(x^k) + \nabla g_i^2(x^k)^t dx + u_i \xi \leq 0, \quad i \in I, \\ \nabla g_i^2(x^k)^t dx = 0, \quad i \in I_1, \end{cases} \end{aligned} \quad (4.1.4)$$

dans lequel les inconnues sont dx et ξ . Les coefficients $u_i (i \in I)$ sont choisis théoriquement en fonction des valeurs propres des hessiens des fonctions g_i^2 en x^k . En l'absence de cette information, on choisit $u_i = 1 (\forall i \in I)$. La formulation qui en résulte est la suivante:

$$\begin{aligned} & PL(x^k) \\ & \text{Maximiser}_{dx, \xi} \xi \\ & \text{s.c.} \begin{cases} \nabla f^1(x^k)^t dx + \xi \leq 0, \\ g_i^2(x^k) + \nabla g_i^2(x^k)^t dx + \xi \leq 0, \quad i \in I, \\ \nabla g_i^2(x^k)^t dx = 0, \quad i \in I_1, \\ \xi \geq 0. \end{cases} \end{aligned} \quad (4.1.5)$$

Toutes les contraintes du problème (4.1.1) sont prises en compte pour calculer une direction de déplacement dx . Dans le cas général, l'algorithme *SLP* adapté à la résolution d'un problème de type (4.1.1) est le suivant:

ALGORITHME *SLP* (forme générale)

Étape 0 (initialisation): déterminer $x^0 \in \mathfrak{F}^{NLP}$; poser $k = 0$.

Étape 1 (direction de déplacement dx^k): résoudre le programme linéaire $PL(x^k)$ en (dx, ξ) dont la valeur optimale est $\xi^* \equiv \xi^*(x^k)$.

Étape 2 (actualisation et test d'arrêt): arrêter la procédure si un critère d'arrêt est satisfait;

sinon déterminer le pas de déplacement $\alpha \in]0,1]$ tel que $x^{k+1} = x^k + \alpha dx^k \in \mathfrak{F}^{NLP}$ et $f^1(x^{k+1}) < f^1(x^k)$. Poser $k = k + 1$, $x^{k+1} = x^k + \alpha dx^k$ et aller à l'**Étape 1**.

Le programme (4.1.1) est convexe. On peut aisément montrer que l'application $D: x^k \in \mathfrak{F}^{NLP} \rightarrow dx \in D(x^k)$ est fermée (lemme 3, p.196 Minoux 1983), elle engendre des directions de descente uniformément réalisables. Pour établir la convergence globale de l'algorithme *SLP*, il suffit de mettre en évidence ses multiplicateurs de LKT. Comme dx^k est une direction de descente tant que l'optimum de $PL(x^k)$ est tel que $\xi^* > 0$, il suffit donc de vérifier que lorsque $\xi^* = 0$, le point courant est un point stationnaire. Ce point résulte du théorème ci-dessous.

Théorème 4.1: Si en un point régulier $x^k \in \mathfrak{F}^{NLP}$, la valeur du programme linéaire $PL(x^k)$ est telle que $\xi^* = 0$, alors les conditions de LKT sont satisfaites en x^k .

Preuve (inspirée de Minoux 1983, théorème 7): En associant aux contraintes du programme linéaire $PL(x^k)$ des variables duales $\eta_0 \geq 0, \eta_i \geq 0 \ i \in I \cup I_1$, le dual du programme linéaire $PL(x^k)$ s'écrit:

(DPL(x^k))

$$\begin{aligned} & \text{Minimiser}_{\eta_i} \sum_{i \in I} -\eta_i g_i^2(x^k) \\ & \text{s.c.} \begin{cases} \eta_0 \nabla_x f^1(x^k) + \sum_{i \in I \cup I_1} \eta_i \nabla_x g_i^2(x^k) = 0, \\ \eta_0 + \sum_{i \in I \cup I_1} \eta_i \geq 1, \\ \eta_0 \geq 0, \eta_i \geq 0, i \in I \cup I_1. \end{cases} \end{aligned} \quad (4.1.6)$$

Du théorème de la dualité en programmation linéaire, on a: $\xi^* = \sum_{i \in I} -\eta_i g_i^2(x^k) = 0$.

Puisque $\forall i \in I$, on a $\eta_i \geq 0$ et $g_i^2(x^k) \leq 0 \Rightarrow \forall i \in I, \eta_i g_i^2(x^k) = 0$. Ceci prouve que les multiplicateurs η_i correspondants aux contraintes non serrées sont nuls.

Montrons que $\eta_0 \neq 0$. Si $\eta_0 = 0$, on a: $0 = \sum_{i \in I(x^k) \cup I_1} \eta_i \nabla_x g_i^2(x^k)$.

Mais x^k est un point régulier; les gradients correspondants aux contraintes serrées en x^k sont linéairement indépendants. On en déduit que $\eta_i = 0 \forall i \in I(x^k) \cup I_1$; ce qui est impossible en raison de la contrainte $\eta_0 + \sum_{i \in I(x^k) \cup I_1} \eta_i \geq 1$. Donc $\eta_0 \neq 0$.

En posant $\eta_i' = \frac{\eta_i}{\eta_0} \geq 0$, on obtient $(\eta_i')_{i \in I}$ qui est le vecteur des multiplicateurs LKT du

programme linéaire $PL(x^k)$ ou encore son vecteur des variables duales tel que:

$$\begin{cases} \nabla_x f^1(x^k) + \sum_{i \in I \cup I_1} \eta_i' \nabla_x g_i^2(x^k) = 0, \\ \eta_i' g_i^2(x^k) = 0 \text{ pour tout } i \in I, \\ \eta_i' \geq 0 \text{ pour tout } i \in I \cup I_1, \end{cases} \quad (4.1.7)$$

qui correspond aux conditions d'optimalité de premier ordre de LKT. Par conséquent, sous les hypothèses de régularité l'algorithme *SLP* se termine lorsque $\xi^* = 0$ par la mise en évidence d'une solution qui est primale duale réalisable. ■

4.1.2 Convergence linéaire de l'algorithme *SLP*

Les algorithmes de programmation quadratique séquentielle dont nous avons rappelé le principe au chapitre 3 doivent leur popularité à la propriété de convergence quadratique. Ces algorithmes présentent tout de même un inconvénient: l'obligation de calculer les multiplicateurs de KKT à chaque itération afin d'actualiser la fonction économique de chaque programme quadratique qui permet de déterminer une direction de déplacement, ce qui peut augmenter le temps de calcul.

Denel et Huard [1974] ont établi la convergence linéaire d'une méthode de programmation linéaire séquentielle dans laquelle, une direction de déplacement est obtenue en résolvant un programme linéaire de type $PL'(x^k)$ ci-dessous pour lequel $I_1 = \emptyset$. Les programmes linéaires $PL'(x^k)$ et $PL(x^k)$ possèdent les mêmes variables duales.

$$\begin{aligned}
 & PL'(x^k) \\
 & \text{Min}_{dx} \nabla f^1(x^k)^t dx \\
 & \text{s.c.} \begin{cases} g_i^2(x^k) + \nabla g_i^2(x^k)^t dx \leq 0, i \in I, \\ \nabla g_i^2(x^k)^t dx = 0, i \in I_1, \\ dx \in \mathbb{R}^n. \end{cases} \quad (4.1.8)
 \end{aligned}$$

L'algorithme *SLP* a les mêmes fondements théoriques que l'algorithme de Zoutendijk [1960]. Pour établir la propriété de convergence linéaire de l'algorithme *SLP*, nous nous inspirons des travaux de Pironneau et Polak [1973] qui ont montré que l'algorithme modifié de Zoutendijk [1960] a une convergence linéaire. Lorsque l'on a $I_1 = \emptyset$, la direction de descente dans l'algorithme étudié par Pironneau et Polak [1973] est donnée par l'expression

$$dx^k = -\eta_0 \nabla_x f^1(x^k) - \sum_{i \in I} \eta_i \nabla_x g_i^2(x^k)$$

Tout au long des démonstrations des différents résultats qui vont suivre, notre contribution consiste à mettre en exergue l'implication de toutes les contraintes du problème (4.1.1) pour déterminer une direction de descente dx^k lorsque l'on a notamment $I_1 \neq \emptyset$.

Considérons le programme linéaire $PL(x^k)$, puis $S \subset \mathbb{R}^n$ un sous-ensemble normé, convexe, compact de \mathbb{R}^n et voisin de l'origine. Le sous-ensemble des directions admissibles S , qui confère à l'algorithme de programmation linéaire séquentielle développé par Zoutendijk [1960] la propriété de convergence linéaire est défini par:

$$S = \{dx \in \mathbb{R}^n : g_i^2(x^k) + \nabla g_i^2(x^k)^t dx \leq 0, i \in I; \nabla g_i^2(x^k)^t dx = 0, i \in I_1, |dx_i| \leq 1, i = 1, 2, \dots, n\}. \quad (4.1.9)$$

En utilisant le sous-ensemble ci-dessus, Polak [1971] a établi une borne supérieure de la vitesse de convergence de l'algorithme de Zoutendijk [1960].

Si l'on ajoute l'indice 0 à l'ensemble des indices des contraintes, on obtient $I^0 = \{0\} \cup I \cup I_1$. Pour des raisons de simplification dans l'exposé, la fonction économique du programme (4.1.1) sera désignée par g_0^2 ; on a donc $f^1 \equiv g_0^2$ et on désigne le sous ensemble des indices des contraintes non actives en x^k par $I_n^k = I \setminus I(x^k)$ et on pose $I_0^k = \{0\} \cup I_n^k$.

Le pas de déplacement τ que l'on calcule à l'étape 2 de l'algorithme *SLP* est solution du programme unidimensionnel ci-après:

$$\text{Minimiser}_{\tau} \{f^1(x^k + \tau dx^k) : g_i^2(x^k + \tau dx^k) \leq 0, i \in I; g_i^2(x^k + \tau dx^k) = 0, i \in I_1\}. \quad (4.1.10)$$

Soit $C(x^k)$ l'ensemble des solutions réalisables du programme dual (4.1.6). De ce qui précède, on peut calculer $\xi^*(x^k)$, valeur du programme $PL(x^k)$ en posant:

$$\begin{aligned}\xi^*(x^k) &= -\text{Min}_{d \in S} \text{Max} \{ \nabla f^1(x^k)^t d; g_i^2(x^k) + \nabla g_i^2(x^k)^t d, i \in I \} \\ &= \text{Min}_{\eta \in C(x^k)} \sum_{j \in I \cup I_1} -\eta_j g_j^2(x^k).\end{aligned}\tag{4.1.11}$$

Tout comme Pironneau et Polak [1973], nous faisons les hypothèses suivantes, qui en outre prennent en compte le fait que l'on a $I_1 \neq \emptyset$ dans (4.1.1) afin d'établir la convergence de l'algorithme *SLP*.

Hypothèses 4.1:

- i) Le programme (NLP) a une solution unique \hat{x} .
- ii) L'ensemble $C' = \{x \in \mathbb{R}^n : g_j^2(x) < 0, j \in I\}$ est non vide.
- iii) Il existe des constantes $t > 0$ et $\rho > 0$ telles que

$$t \|d\|^2 \leq d^t \nabla^2 f^1(x) d \tag{4.1.12}$$

pour $d \in \mathbb{R}^m$, pour tout $x \in B(\hat{x}, \rho) \cap \{x : \|x - \hat{x}\| \leq \rho\}$.

L'inégalité (4.1.12) signifie que la fonction économique f^1 est localement fortement convexe. Sous les hypothèses 4.1, lorsque l'on a $I_1 = \emptyset$, Pironneau et Polak [1973] (lemme 3.2) ont montré que la convergence linéaire de la fonction économique du problème (4.1.1) implique celle de la suite des solutions $\{x^k\}_{k \in K}$ générée par l'algorithme *SLP*. En complément aux hypothèses 4.1, nous formulons l'hypothèse ci-après:

Hypothèse 4.2: Soit \hat{x} l'unique solution du problème (4.1.1). Alors, il existe $\hat{\rho} > 0$ tel que si $J \subset I^0$ satisfait $0 \in J$ et $0 \in \text{co}\{\nabla g_j^2(\hat{x})\}_{j \in J}$, alors $0 \in \text{co}\{\nabla g_j^2(x)\}_{j \in J}$ pour tout $x \in \{x : \|x - \hat{x}\| \leq \hat{\rho}; x \in \mathcal{F}^{NLP}\}$, où $\text{co}\{E\}$ désigne l'enveloppe convexe d'un ensemble E et 0 est le vecteur nul de \mathbb{R}^m .

L'hypothèse 4.2 stipule que s'il existe une constante $\hat{\rho} > 0$ telle que $\sum_{j \in J} \hat{\eta}_j \nabla g_j^2(\hat{x}) = 0$ avec $\hat{\eta}_j \geq 0$ et $\sum_{j \in J} \hat{\eta}_j = 1$, alors pour toute solution réalisable x qui satisfait $\|x - \hat{x}\| \leq \hat{\rho}$, il existe $\eta_j(x) \geq 0, j \in J$ avec $\sum_{j \in J} \eta_j(x) = 1$ tel que $\sum_{j \in J} \eta_j \nabla g_j^2(x) = 0$. Puisque \hat{x} est l'unique solution du problème (4.1.1), l'hypothèse 4.2 entraîne que $\nabla f^1(\hat{x}) \neq 0$. L'hypothèse 4.2 est satisfaite sous l'hypothèse d'indépendance linéaire des contraintes actives du problème (4.1.1) en \hat{x} .

Soit $\xi^*(x^k)$ la solution optimale du programme linéaire $PL(x^k)$. Le lemme ci-dessous est une adaptation à l'algorithme *SLP* du lemme 3.2 formulé par Pironneau et Polak [1973]. Il permet de déterminer une borne supérieure de l'opposé de la valeur du programme linéaire $PL(x^k)$ à chaque itération k .

Tout au long des développements à venir, la notation $\langle a, b \rangle$ désigne le produit scalaire de a par b .

Lemme 4.1: Soient $\{x^k\}_{k \in \mathbb{N}}$ et $\{\xi^*(x^k)\}_{k \in \mathbb{N}}$ deux séquences générées par l'algorithme *SLP*.

Sous les hypothèses 4.1 et 4.2, étant donné $\alpha \in (0,1)$, il existe un entier $k_0(\alpha)$ tel que

$$-\xi^*(x^k) \leq \eta(\hat{x})\alpha < 0 \text{ pour tout } k \geq k_0(\alpha) \quad (4.1.13)$$

où $\xi^*(x^k)$ est la valeur du programme linéaire $PL(x^k)$ calculée à l'étape 1 de l'algorithme *SLP* et

$$\eta(\hat{x}) = \max \left\{ \eta_J(\hat{x}) : J \subset I^0, 0 \in J, 0 \notin \text{co} \left\{ \nabla g_j^2(\hat{x}) \right\}_{j \in J} \right\}, \quad (4.1.14)$$

avec

$$\eta_J(\hat{x}) = \text{Min}_{d \in S} \text{Max}_{j \in J} \langle \nabla g_j^2(\hat{x}), d \rangle. \quad (4.1.15)$$

Preuve: Nous commençons par montrer que $\eta(\hat{x}) < 0$. Pour tout $x \in \mathfrak{F}^{NLP}$, soit

$$\mathfrak{J}(x) = \left\{ J \subset I^0 : 0 \notin \text{co} \left\{ \nabla g_j^2(x) \right\}_{j \in J}, 0 \in J \right\}. \quad (4.1.16)$$

Soit $J \in \mathfrak{J}(\hat{x})$ arbitrairement choisi et supposons que nous avons $\eta_J(\hat{x}) = 0$. Montrons que ceci est absurde. Reformulons (4.1.15) en faisant appel à l'enveloppe convexe de \mathfrak{F}^{NLP} et en utilisant le théorème du minimax de Von Neumann, on obtient

$$\eta_J(\hat{x}) = \text{Min}_{d \in S} \text{Max}_{u \in U_J} \sum_{j=0}^m u^j \langle \nabla g_j^2(\hat{x}), d \rangle = \text{Max}_{u \in U_J} \text{Min}_{d \in S} \sum_{j=0}^m u^j \langle \nabla g_j^2(\hat{x}), d \rangle \quad (4.1.17)$$

où

$$U_J = \left\{ (u^0, u^1, \dots, u^m) : u^j \geq 0, j = 0, 1, \dots, m, \sum_{j \in J} u^j = 1 \text{ et } u^j = 0 \ \forall j \notin J \right\}. \quad (4.1.18)$$

Donc, pour un certain $\bar{u} \in U_J$, on doit avoir

$$0 = \eta_J(\hat{x}) = \text{Min}_{d \in S} \sum_{j=1}^m \bar{u}^j \langle \nabla g_j^2(\hat{x}), d \rangle. \quad (4.1.19)$$

Mais S est un voisinage de l'origine et par conséquent, (4.1.19) entraîne que $\sum_{j \in J} \bar{u}^j \nabla g_j^2(\hat{x}) = 0$,

ce qui signifie que $0 \in \text{co} \left\{ \nabla g_j^2(\hat{x}) \right\}_{j \in J}$, qui contredit notre hypothèse que $J \in \mathfrak{J}(\hat{x})$. Par conséquent, on doit avoir $\eta_J(\hat{x}) < 0$ pour tout $J \in \mathfrak{J}(\hat{x})$, i.e., $\eta(\hat{x}) < 0$.

Puisque chacune des fonctions $\eta_J : \mathfrak{F}^{NLP} \rightarrow \mathbb{R}^m$ définie par (4.1.15) est continue, et d'après l'hypothèse 4.2, il existe $\hat{\rho} > 0$ tel que $\mathfrak{J}(\hat{x}) \supset \mathfrak{J}(x)$ pour tout $x \in \mathfrak{F}^{NLP}$, $\|x - \hat{x}\| \leq \hat{\rho}$; on en déduit que $\eta : \mathfrak{F}^{NLP} \rightarrow \mathbb{R}$, définie par (4.1.14) est semi continue supérieurement (s.c.s) en \hat{x} . Par construction de l'algorithme *SLP*, pour $k = 0, 1, 2, \dots$, on a $\xi^*(x^k) > 0$ et par conséquent, en comparant (4.1.11) avec (4.1.17), on doit avoir $I_0^k \in \mathfrak{J}(x^k)$. On a donc: $-\xi^*(x^k) \leq \eta(x^k)$ pour tout k et comme η est s.c.s., pour un certain $\alpha \in (0, 1)$, il existe un entier $k_0(\alpha)$ tel que pour tout $k \geq k_0(\alpha)$,

$$-\xi^*(x^k) \leq \eta(x^k) \leq \eta(\hat{x})\alpha < 0, \quad (4.1.20)$$

ce qui complète la démonstration. ■

Lemme 4.2: Soient $\{x^k\}_{k \in \mathbb{N}}$ et $\{\xi^*(x^k)\}_{k \in \mathbb{N}}$ deux séquences générées par l'algorithme *SLP*.

On suppose que les hypothèses 4.1 et 4.2 sont satisfaites; soit K un ensemble infini d'indices.

Alors, pour un certain $\alpha \in (0,1)$, il existe $k_1(\alpha)$ et $\beta > 0$ tel que

$$\xi^*(x^k) \leq \beta \alpha \bar{\lambda}^0 [f^1(x^k) - f^1(\hat{x})] \text{ pour } k \in K, k \geq k_1(\alpha), \quad (4.1.21)$$

où \hat{x} est l'unique solution du problème (4.1.1) et

$$\bar{\lambda}^0 = \text{Min} \left\{ \lambda^0 : \sum_{j=0}^m \lambda^j \nabla g_j^2(\hat{x}) = 0, \sum_{j=1}^m \lambda^j \nabla g_j^2(\hat{x}) = 0, \right. \\ \left. \sum_{j=0}^m \lambda^j = 1, \lambda^j \geq 0, j = 0,1,2,\dots,m \right\} > 0. \quad (4.1.22)$$

Preuve: Il faudrait d'abord noter qu'en raison de l'hypothèse 4.2, $\bar{\lambda}^0 > 0$ (lemme 1.12, Pironneau et Polak 1971). On a également par définition de I_n^k ,

$$\text{Max}_{j \in I_n^k} g_j^2(x^k) \leq -\xi^*(x^k). \quad (4.1.23)$$

Comme f^1 et g_j^2 , $j = 1,2,\dots,m$ sont des fonctions convexes, et $g_j^2(x^k) \leq 0, j \in I$; $g_j^2(x^k) = 0, j \in I_1$, on doit avoir:

$$f^1(x^k) - f^1(\hat{x}) \leq \langle \nabla f^1(x^k), x^k - \hat{x} \rangle; \\ g_j^2(x^k) - g_j^2(\hat{x}) \leq \langle \nabla g_j^2(x^k), x^k - \hat{x} \rangle, j \in I, \langle \nabla g_j^2(x^k), x^k - \hat{x} \rangle = 0, j \in I_1;$$

on a donc

$$\begin{aligned}
\text{Max}_{j \in I_n^k} g_j^2(x^k) &= \text{Max}_{u \in U_{I_n^k}} \sum_{j \in I_n^k} u^j g_j^2(x^k) \\
&= \text{Max}_{u \in U_{I_0^k}} \left\{ \sum_{j \in I_n^k} u^j g_j^2(x^k) + u^0 [f^1(x^k) - f^1(\hat{x})] - u^0 [f^1(x^k) - f^1(\hat{x})] \right\} \quad (4.1.24) \\
&\leq \text{Max}_{u \in U_{I_0^k}} \left\{ \sum_{j \in I_n^k} u^j \langle g_j^2(x^k), x^k - \hat{x} \rangle - u^0 [f^1(x^k) - f^1(\hat{x})] \right\},
\end{aligned}$$

$U_{I_n^k}$ et $U_{I_0^k}$ sont définis comme dans (4.1.18). D'après l'hypothèse 4.2, il existe un $\bar{u}_k \in U_{I_0^k}$ tel que $\bar{u}_k^j \geq 0$ avec $\sum_{j \in I_0^k} \bar{u}_k^j = 1$ et $\sum_{j \in I_0^k} \bar{u}_k^j \nabla g_j^2(x^k) = 0$.

En reprenant (4.1.24), on a: étant donné $\alpha \in (0,1)$, il existe $k_1'(\alpha)$ tel que

$$\text{Max}_{j \in I_n^k} g_j^2(x^k) \leq -\bar{u}_k^0 [f^1(x^k) - f^1(\hat{x})] \text{ pour tout } k \geq k_1'(\alpha). \quad (4.1.25)$$

où \bar{u}_k^0 est la première composante d'un vecteur \bar{u}_k tel que

$$\bar{u}_k \in U_{I_0^k}, \sum_{j \in I_0^k} \bar{u}_k^j \nabla g_j^2(x^k) = 0. \quad (4.1.26)$$

À présent, considérons $\{\bar{u}_k\}_{k \in K}$ tel que (4.1.23) et (4.1.24) soient satisfaites et soit K' un sous ensemble infini de K tel que $\lim_{k \in K', k \rightarrow \infty} \bar{u}_k^0 = \liminf \bar{u}_k$.

Soit K'' un sous ensemble infini de K' et \hat{I}^0 un sous ensemble de $I(\hat{x}) \cup I_1 \cup \{0\}$ tel que $I_0^k = \hat{I}^0$ pour tout $k \in K''$. Il n'est pas difficile d'établir l'existence de K'' . Notons que l'application multivoque $\Gamma_{I_0^k} : \mathfrak{F}^{NLP} \rightarrow 2^{U_{I_0^k}}$ définie par

$$\Gamma_{I_0^k}(x^k) = \left\{ u \in U_{I_0^k} : \sum_{j=0}^m u^j \nabla g_j^2(x^k) = 0 \right\} \quad (4.1.27)$$

est fermée (Berge 1963, p.111) et est non vide pour tout $k \in K^n$. Donc, puisque $\hat{I}^0 \subset I(\hat{x}) \cup I_1 \cup \{0\}$, $x^k \rightarrow \hat{x}$ quand $k \rightarrow \infty$, $k \in K^n$, l'ensemble $\Gamma_{\hat{I}^0}(\hat{x})$ est un sous ensemble non vide des multiplicateurs en \hat{x} , i.e.

$$\Gamma_{\hat{I}^0}(\hat{x}) \subset \Lambda(\hat{x}) \cap \left\{ \lambda \in U_{\hat{I}^0(\hat{x})} : \sum_{j=0}^m \lambda^j \nabla g_j^2(\hat{x}) = 0 \right\}. \quad (4.1.28)$$

Nous en déduisons que, pour un certain $\alpha \in (0,1)$, il existe $k_1(\alpha) \geq k_1'(\alpha)$ tel que

$$\bar{u}_k^0 \geq \bar{\lambda}^0 \alpha \text{ pour tout } k \geq k_1(\alpha), \quad (4.1.29)$$

avec $k_1(\alpha)$ indépendant de la séquence particulière $\{\bar{u}_k\}$ choisie. En combinant (4.1.25) et (4.1.29), on obtient:

$$\text{Max}_{j \in J_h^k} g_j^2(x^k) \leq -\bar{\lambda}^0 \alpha [f^1(x^k) - f^1(\hat{x})] \quad (4.1.30)$$

Considérons (4.1.25) et (4.1.29); comme $\xi^*(x^k) > 0$ et $\bar{\lambda}^0 \alpha [f^1(x^k) - f^1(\hat{x})] > 0$, deux cas sont envisageables pour tout $k \in K^n$:

- i) $\xi^*(x^k) \leq \bar{\lambda}^0 \alpha [f^1(x^k) - f^1(\hat{x})]$. On choisit $\beta = 1$.
- ii) $\xi^*(x^k) > \bar{\lambda}^0 \alpha [f^1(x^k) - f^1(\hat{x})]$. On peut trouver un entier $k_2(\alpha)$ tel que $\beta = (1 + \alpha)^{k_2(\alpha)}$ et $\xi^*(x^k) \leq \beta \bar{\lambda}^0 \alpha [f^1(x^k) - f^1(\hat{x})]$. Ce qui achève la démonstration. ■

À travers le théorème ci-après, nous établissons la propriété de convergence linéaire de l'algorithme *SLP*. Les éléments de démonstration sont empruntés aux concepts utilisés dans la preuve du théorème 3.24 plus général de Pironneau et Polak [1973].

Théorème 4.2: Soient $\{x^k\}_{k \in K}$ et $\{\xi^*(x^k)\}_{k \in K}$ deux séquences générées par l'algorithme *SLP* afin de résoudre le problème (4.1.1); on suppose que les hypothèses 4.1 et 4.2 sont satisfaites. Soit K un sous-ensemble infini d'indices tel que

$$\xi^*(x^k) < \xi^*(x^{k-1}) \text{ pour tout } k \in K. \quad (4.1.31)$$

Alors, pour un certain $\alpha \in (0,1)$, il existe un entier ν , (qui peut dépendre de α) tel que pour tout $j \geq 0$,

$$f^1(x^{\nu+j}) - f^1(\hat{x}) \leq \left(1 - \frac{\beta \bar{\lambda}^0 |\eta(\hat{x})| \alpha^4}{LW}\right)^j [f^1(x^\nu) - f^1(\hat{x})] \quad (4.1.32)$$

où

$$W = \text{Max}\{l \times |d_x^i| : i = 1, 2, \dots, m, d_x \in S\}, \quad (4.1.33)$$

$$L = (1 + \beta) \times \text{Max}\{\|\nabla f^1(\hat{x})\|; \|\nabla g_j^2(\hat{x})\| \mid j = 1, 2, \dots, m; 1\}, \quad (4.1.34)$$

et $\bar{\lambda}^0$, $\eta(\hat{x})$ sont définis respectivement par (4.1.22) et (4.1.14), alors que \hat{x} est l'unique solution du programme (4.1.1).

Preuve: Soient $\{x^k\}_{k \in N}$ et $\{\xi^*(x^k)\}_{k \in N}$ les deux séquences générées par l'algorithme *SLP*, $\xi^*(x^k)$ étant la valeur optimale du programme linéaire $PL(x^k)$. Soit $dx^k \equiv d(x^k, \xi^*(x^k))$ la direction de déplacement telle calculée par l'algorithme *SLP* pour $k = 0, 1, 2, \dots$ à l'étape 1. En raison de la convexité du programme (4.4.1), pour tout $\tau \geq 0$ et $k = 0, 1, 2, \dots$, on a

$$g_j^2(x^k + \tau dx^k) \leq g_j^2(x^k) + \tau \langle \nabla g_j^2(x^k + \tau dx^k), dx^k \rangle, \quad j = 0, 1, 2, \dots \quad (4.1.35)$$

Par la suite, puisque $x^k \rightarrow \hat{x}$, l'unique solution de (4.1.1) et $\xi^*(x^k) \rightarrow 0$, étant donné $\alpha \in (0,1)$ ¹ arbitrairement fixé, il existe par continuité un entier $k_2(\alpha)$ tel que

¹ Tout au long de cette démonstration, nous supposons que $\alpha \in (0,1)$ est arbitraire, mais fixé.

$$\|\nabla g_j^2(x^k + \tau dx^k)\| \leq L/\alpha, j = 1, 2, \dots, m, \quad (4.1.36)$$

pour tout $k \geq k_2(\alpha)$ et pour tout $\tau \in [0, \xi^*(x^k)/(LW)]$. À présent, considérons certains $k \geq k_2(\alpha)$ et $j \in I_n^k$. Alors, $g_j^2(x^k) \leq -\xi^*(x^k)$ et par conséquent, (4.1.35) et (4.1.36) combinées avec l'inégalité de Schwarz donnent pour tous k et j ,

$$g_j^2(x^k + \tau dx^k) \leq -\xi^*(x^k) + \frac{\tau L}{\alpha} \|dx^k\|, \quad \text{pour tout } \tau \in [0, \frac{\xi^*(x^k)}{LW}]. \quad (4.1.37)$$

De (4.1.10) et de (4.1.37), on déduit que pour tout $k \geq k_2(\alpha)$,

$$\begin{aligned} f^1(x^{k+1}) - f^1(x^k) &\leq \text{Min} \left\{ f^1(x^k + \tau dx^k) - f^1(x^k) : g_j^2(x^k + \tau dx^k) \leq 0, \right. \\ &\quad \left. j \in I_n^k; \tau \in [0, \alpha \xi^*(x^k)/L \|dx^k\|] \right\} \\ &\leq \text{Min}_{0 \leq \tau \leq \alpha \xi^*(x^k)/L \|dx^k\|} \text{Max}_j \left\{ f^1(x^k + \tau dx^k) - f^1(x^k); g_j^2(x^k + \tau dx^k), j \in I_n^k \right\} \quad (4.1.38) \\ &\leq \text{Min}_{0 \leq \tau \leq \alpha \xi^*(x^k)/L \|dx^k\|} \text{Max}_{u \in U_{I_0^k}^k} \left\{ \sum_{j=0}^m u^j g_j^2(x^k + \tau dx^k) - u^0 f^1(x^k) \right\} \end{aligned}$$

où $U_{I_0^k}^k$ est défini par (4.1.18) pour $J = I_0^k$. Soit

$$M = \text{Max} \left\{ \left\| \frac{\partial^2 g_j^2(x^k + \bar{\tau}_k dx^k)}{\partial x^2} \right\| : j = 0, 1, 2, \dots, m, k \geq k_2(\alpha), \bar{\tau}_k \in [0, \frac{\alpha \xi^*(x^k)}{L \|dx^k\|}] \right\} \quad (4.1.39)$$

Alors, en développant au second ordre la dernière expression de (4.1.38) et en faisant appel à (4.1.10) et (4.1.11), on obtient pour tout $k \geq k_2(\alpha)$,

$$\begin{aligned}
f^1(x^{k+1}) - f^1(x^k) &\leq \underset{0 \leq \tau \leq \alpha \xi^*(x^k) / (L \|dx^k\|)}{\text{Min}} \underset{u \in U_0^k}{\text{Max}} \left\{ \sum_{j=1}^m u^j g_j^2(x^k) + \right. \\
&\quad \left. \tau \sum_{j=0}^m u^j \langle g_j^2(x^k), dx^k \rangle + \frac{\tau^2}{2} M \|dx^k\|^2 \right\} \quad (4.1.40) \\
&\leq \underset{0 \leq \tau \leq \alpha \xi^*(x^k) / (L \|dx^k\|)}{\text{Min}} \left\{ -\tau \xi^*(x^k) + \frac{\tau^2}{2} M \|dx^k\|^2 \right\},
\end{aligned}$$

puisque $\sum_{j=1}^m u^j g_j^2(x^k) \leq 0$ pour tout $k \geq 0$ et pour tout $u \in U_0^k$.

Ensuite, soit $k_0(\alpha) \geq k_2(\alpha)$ tel que (4.1.13) soit satisfaite. Alors, (4.1.40), (4.1.33), (4.1.39) et (4.1.13) ensemble donnent pour tout $k \geq k_0(\alpha)$

$$f^1(x^{k+1}) - f^1(x^k) \leq \underset{0 \leq \tau \leq \alpha \xi^*(x^k) / (LW)}{\text{Min}} \left\{ \tau \eta(\hat{x}) \alpha + \frac{\tau^2}{2} MW^2 \right\}, \quad (4.1.41)$$

Puisque $\xi^*(x^k) \rightarrow 0$ quand $k \rightarrow \infty$, il existe un entier $k_3(\alpha) \geq k_0(\alpha)$ tel que le Min sur (4.1.41) est atteint en $\tau = \alpha \xi^*(x^k) / (LW)$ pour tout $k \geq k_3(\alpha)$, par conséquent pour tout $k \geq k_3(\alpha)$, on a:

$$f^1(x^{k+1}) - f^1(x^k) \leq \frac{\xi^*(x^k) \eta(\hat{x}) \alpha^2}{LW} + \frac{\xi^*(x^k)^2 \alpha^2 M}{L^2}. \quad (4.1.42)$$

À nouveau, puisque $\xi^*(x^k) \rightarrow 0$ quand $k \rightarrow \infty$, on déduit de (4.1.42) qu'il existe un $k_4(\alpha) \geq k_3(\alpha)$ tel que pour tout $k \geq k_4(\alpha)$, on a

$$f^1(x^{k+1}) - f^1(x^k) \leq \frac{\xi^*(x^k) \eta(\hat{x}) \alpha^3}{LW}. \quad (4.1.43)$$

Nous faisons appel à présent aux hypothèses du théorème et au lemme 4.2, qui établissent l'existence de $k_1(\alpha) \geq k_4(\alpha)$ pour $k = 0, 1, 2, \dots$; soit $i(k)$ l'entier satisfaisant

$$\xi^*(x^k) < \xi^*(x^{k-1}) < \dots < \xi^*(x^{k-i(k)+1}) < \xi^*(x^{k-i(k)}) \quad (4.1.44)$$

Alors, en combinant (4.1.43) et (4.1.21) (des expressions sont ajoutées et retranchées), on a:

$$\begin{aligned} f^1(x^{k+1}) - f^1(x^{k-i(k)}) &\leq \frac{\eta(\hat{x})\alpha^3}{LW} (\xi^*(x^k) + \xi^*(x^{k-1}) + \dots + \xi^*(x^{k-i(k)})) \\ &\leq \frac{\eta(\hat{x})\alpha^3}{LW} (1+i(k))\xi^*(x^{k-i(k)}) \\ &\leq \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} (1+i(k)) [f^1(x^{k-i(k)}) - f^1(\hat{x})] \end{aligned} \quad (4.1.45)$$

pour tout $k \geq k_1(\alpha)$. En réorganisant (4.1.45), on obtient pour tout $k \geq k_1(\alpha)$,

$$f^1(x^{k+1}) - f^1(\hat{x}) \leq \left[1 + \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} (1+i(k)) \right] [f^1(x^{k-i(k)}) - f^1(\hat{x})] \quad (4.1.46)$$

Il faudrait noter que, puisque $f^1(x^{k+1}) - f^1(\hat{x}) > 0$, (4.1.46) entraîne que $i(k)$ est bornée. On a

$$0 < \left| \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} \right| < 1 \text{ en raison de (4.1.33) et (4.1.34); on doit avoir}$$

$$1 + \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} (1+i(k)) \leq \left(1 + \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} \right)^{(1+i(k))}. \quad (4.1.47)$$

Mais (4.1.46) et (4.1.47) entraînent (4.1.32). Ce qui établit la convergence linéaire de l'algorithme SLP. ■

Remarque (Erreurs sur la preuve du théorème 3.24 de Pironneau et Polak [1973]): Pour établir la convergence linéaire de la variante de l'algorithme de Zoutendijk pour laquelle une direction de descente est obtenue en résolvant à chaque itération un programme linéaire du type $PL(x^k)$ avec $I_1 = \emptyset$, Pironneau et Polak [1973], dans le lemme 3.11, considèrent un ensemble infini

d'indices K pour lequel on a $\varepsilon_k < \varepsilon_{k-1} \forall k \in K$ et montrent que pour un certain $\alpha \in (0,1)$, il existe $k_1(\alpha)$ et $\bar{\lambda}_0$ tel que

$$\varepsilon_k \geq \beta \alpha \bar{\lambda}^0 [f^1(x^k) - f^1(\hat{x})] \text{ pour } k \in K, k \geq k_1(\alpha), \quad (4.1.48)$$

où $\varepsilon_k > 0$ et $\beta > 0$ sont calculés à l'intérieur de l'algorithme sur lequel porte l'analyse. Une première erreur s'est glissée dans la preuve du lemme 3.11 en question. En effet Pironneau et Polak [1973] établissent l'existence d'un entier $i_1(\alpha) \geq i_1'(\alpha)$ tel que $\bar{u}_i^0 \geq \bar{\lambda}^0(1-\alpha)$ pour tout $i \geq i_1(\alpha)$ afin d'obtenir (4.1.48) à partir des inégalités suivantes:

$$\text{Min}_{j \in I_k} g_j^2(x^k) \geq -\varepsilon_k / \beta \text{ où } I_k = \{j \in I : g_j^2(x^k) \geq -\varepsilon_k\},$$

$$\text{Min}_{j \in I_k} g_j^2(x^k) \leq -\bar{u}_0 [f^1(x^k) - f^1(\hat{x})],$$

$$\bar{u}_0 \geq \bar{\lambda}_0(1-\alpha).$$

C'est plutôt l'inégalité $\bar{u}_i^0 \geq \bar{\lambda}^0 \alpha$ qui permet d'établir (4.1.48). Dans le théorème 3.24 mentionné plus haut, considérant $\varepsilon_k = \varepsilon_{k-1} = \dots = \varepsilon_{k-i(k)} < \varepsilon_{k-i(k)-1}$, $k \in K$, ces auteurs en utilisant le raisonnement de la démonstration ci-dessus (il faudrait alors remplacer $\xi^*(x^k)$ par ε_k), établissent à partir de (4.1.48) que l'on a:

$$\begin{aligned} f^1(x^{k+1}) - f^1(x^k) &\leq \frac{\eta(\hat{x})\alpha^3}{LW} (\varepsilon_k + \varepsilon_{k-1} + \dots + \varepsilon_{k-i(k)}) \\ &= \frac{\eta(\hat{x})\alpha^3}{LW} (1+i(k))\varepsilon_{k-i(k)} \\ &\leq \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} (1+i(k)) [f^1(x^{k-i(k)}) - f^1(x^k)] \end{aligned} \quad (4.1.49)$$

d'après (4.1.48), ce qui n'est pas correct. On a plutôt:

$$\frac{\eta(\hat{x})\alpha^3}{LW} (1+i(k))\varepsilon_{k-i(k)} \geq \frac{\beta\eta(\hat{x})\alpha^4\bar{\lambda}^0}{LW} (1+i(k)) [f^1(x^{k-i(k)}) - f^1(x^k)]. \quad (4.1.50)$$

La conclusion de la preuve de ce théorème 3.24, en s'appuyant sur (4.1.49) et (4.1.48) ne peut donc pas être correcte. Cette conclusion aurait pu être correcte si dans le lemme 3.11 énoncé par ces auteurs, l'on avait une inégalité du type $\varepsilon_k \leq \beta \alpha \bar{\lambda}^0 [f^1(x^k) - f^1(\hat{x})]$ pour $k \in K$, $k \geq k_1(\alpha)$, à la place de (4.1.48).

4.2 Approximation du problème de PDN sous forme KKT par un programme mathématique localement régulier

Dans cette section, nous montrons qu'un problème de PDN sous forme KKT peut être transformé en une suite de programmes mathématiques classiques localement réguliers. Nous rappelons le principe d'une méthode d'approximation des contraintes d'équilibre par une fonction de mérite, puis nous établissons les conditions d'optimalité du premier ordre du problème de PDN sous forme KKT ainsi approximé.

4.2.1 Méthode d'approximation des contraintes d'équilibre

Considérons le problème de PDN

(PDN)

$$\begin{aligned} & \text{Minimiser}_{x,y} f^1(x,y) \\ & \text{s.c.} \left\{ \begin{array}{l} x \in P, \\ \text{Minimiser}_y f^2(x,y), \\ g_j^2(x,y) \leq 0, j \in J, \\ y \in \mathbb{R}^{n_y}, \end{array} \right. \end{aligned} \quad (4.2.1)$$

dont la formulation KKT est la suivante:

$$\begin{aligned}
 & (PDN_{KKT}) \\
 & \text{Minimiser}_{x,y} f^1(x,y) \\
 & \text{s.c.} \begin{cases} H(x,y,w,v) = 0, \\ w^t v = 0, \\ (x,y) \in P \times \mathbb{R}^{n_y}, w, v \geq 0, \end{cases} \quad (4.2.2)
 \end{aligned}$$

avec

$$H(x,y,w,v) = \begin{pmatrix} \nabla_y f^2(x,y) + w^t \nabla_y g_j^2 \\ g_j^2(x,y) + v \end{pmatrix}, \quad (4.2.3)$$

$f^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ fonctions de classe C^2 , $i = 1, 2$; $g_j^2 : \mathbb{R}^{n_x + n_y} \rightarrow \mathbb{R}$, $j = 1, 2, \dots, m$, de classe C^2 et convexes, $P \subset \mathbb{R}^{n_x}$ est un ensemble compact supposé non vide et $v, w \in \mathbb{R}_+^m$. Par ailleurs, le domaine réalisable du problème de second niveau défini par l'ensemble $C(x) = \{y \in \mathbb{R}^{n_y} : g_j^2(x,y) \leq 0, j = 1, \dots, m\}$ est supposé non vide et borné pour tout $x \in P$. Soit $I(x,y) = \{j \in J : g_j^2(x,y) = 0\}$ l'ensemble des contraintes actives en une solution admissible du problème (PDN). Pour des besoins d'analyse, nous formulons les hypothèses ci-après:

Hypothèse 4.3: La fonction $y \rightarrow \nabla_y f^2(x,y)$ est fortement monotone pour tout $x \in P$.

Hypothèse 4.4: Pour tout $x \in P$, les gradients $\nabla_y g_j^2(x,y)$, $j \in I(x,y)$ des contraintes actives en tout point admissible (x,y) sont linéairement indépendants.

Hypothèse 4.5: Toute solution optimale du problème (PDN_{KKT}) est non dégénérée ($w_i^* + v_i^* \neq 0 \forall i \in J$).

Examinons les conséquences de ces hypothèses sur les propriétés de la formulation KKT du problème (PDN). L'hypothèse 4.3 garantit que le domaine induit DI est compact; de plus l'application $M(x)$ est univoque d'après le théorème 3.1 du chapitre 3. Sous les hypothèses 4.3 et 4.4, pour tout $x \in P$, la fonction $H(x, \dots) : \mathbb{R}^{n_y + 2m} \rightarrow \mathbb{R}^{n_y + 2m}$ est monotone (proposition 3, Facchinei et al., 1999).

En raison de la présence de la contrainte d'équilibre $w^t v = 0$, il est connu que les hypothèses de régularité ne sont pas satisfaites pour le programme non linéaire (PDN_{KKT}). Pour avoir un problème localement régulier, nous utilisons une des approximations obtenues en faisant subir aux contraintes d'équilibre une transformation localement continue à l'aide d'une variante de la fonction de mérite de Fisher-Burmeister. À titre de rappel, c'est la fonction $\Phi : \mathbb{R}_+^2 \times [0, +\infty[\rightarrow \mathbb{R}$ définie par:

$$\Phi(a, b, \mu) = a + b - \sqrt{a^2 + b^2 + 2\mu}. \quad (4.2.4)$$

Si $\mu = 0$, Φ est la fonction de mérite de Fisher-Burmeister [1992]. Pour résoudre un problème de complémentarité linéaire, Kanzow [1996] a développé un algorithme continu en utilisant une fonction de mérite de ce type; il en est de même de Chen et Harker [1993] qui ont proposé un algorithme de points non intérieurs. Cette fonction de mérite a également été abondamment utilisée dans le contexte des problèmes de MPEC linéaire par des auteurs comme Faccheinei et Soares [1997] et Fukushima et al. [1998].

Kanzow [1995] (lemme 2.2) a montré que la fonction de mérite Φ a la propriété (NCP) que nous avons énoncée au chapitre 3, à savoir que

$$\forall \mu > 0, \Phi(a, b, \mu) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = \mu. \quad (4.2.5)$$

On a également

$$ab \leq \mu \Rightarrow \Phi(a, b, \mu) \leq 0. \quad (4.2.6)$$

pour $\mu \neq 0$, la fonction $\Phi(\cdot, \cdot, \mu) : \mathbb{R}^2 \rightarrow \mathbb{R}$ est différentiable partout. Cependant, lorsque $\mu = 0$ cette fonction n'est pas différentiable en $(a, b) = (0, 0)$. Pour tout $(a, b, \mu) \in \mathbb{R}^2 \times [0, +\infty[$ tel que $(a, b, \mu) \neq (0, 0, 0)$ on a:

$$\nabla_{(a,b)} \Phi(a,b,\mu) \equiv \begin{pmatrix} \frac{\partial \Phi(a,b,\mu)}{\partial a} \\ \frac{\partial \Phi(a,b,\mu)}{\partial b} \end{pmatrix} = \begin{pmatrix} 1 - \frac{a}{\sqrt{a^2 + b^2 + 2\mu}} \\ 1 - \frac{b}{\sqrt{a^2 + b^2 + 2\mu}} \end{pmatrix}. \quad (4.2.7)$$

Rappelons que les fonctions de mérite d'une manière générale, sont utilisées entre autres pour établir la convergence d'algorithmes en programmation mathématique. Considérons la formulation KKT du problème de PDN donnée par (4.2.2). Pour $x \in P$ fixé, nous remplaçons la contrainte d'équilibre $w^i v = 0$ par une approximation du type $\Psi(w,v,\mu) = 0$ qui a pour composantes $\Phi(w_i, v_i, \mu)$, $i \in J = \{1, 2, \dots, m\}$ où Φ est la fonction de mérite paramétrique de Fisher-Burmeister. On obtient le programme non linéaire paramétré suivant:

$$\begin{aligned} & PDN_{KKT}(\mu) \\ & \text{Minimiser } f^1(x, y) \\ & \text{s.c. } \begin{cases} H(x, y, w, v) = 0, \\ \Phi(w_i, v_i, \mu) = 0, \quad i \in J, \\ (x, y) \in P \times \mathbb{R}^{n_y}, w, v \geq 0, \end{cases} \end{aligned} \quad (4.2.9)$$

L'ensemble des solutions réalisables de $PDN_{KKT}(\mu)$ est noté \mathfrak{F}^μ . Sous les hypothèses ci-dessus, Facchinei et al. [1999] ont montré que l'ensemble \mathfrak{F}^μ est non vide et compact (corollaire 1) et que si un problème du type (PDN) a une solution, alors tous les problèmes perturbés $PDN_{KKT}(\mu)$ ont chacun une solution (corollaire 2) et cette solution est unique (lemme 3).

L'omission de l'hypothèse 4.5 pourrait compromettre une hypothèse de qualification des contraintes pour le problème $PDN_{KKT}(\mu)$. En effet, considérons les matrices diagonales ci-après

$$\begin{aligned} D_w^k &= \text{diag}\left(\frac{\partial \Phi(w_1, v_1, \mu_k)}{\partial w_1}, \dots, \frac{\partial \Phi(w_m, v_m, \mu_k)}{\partial w_m}\right), \\ D_v^k &= \text{diag}\left(\frac{\partial \Phi(w_1, v_1, \mu_k)}{\partial v_1}, \dots, \frac{\partial \Phi(w_m, v_m, \mu_k)}{\partial v_m}\right), \end{aligned} \quad (4.2.10)$$

L'hypothèse 4.5 qui est relative aux relations de complémentarité du problème (PDN_{KKT}), garantit que les matrices diagonales D_v^k et D_w^k sont inversibles.

La proposition ci-après est, entre autres une conséquence de la propriété (4.2.5) relative à la fonction de mérite Φ .

Proposition 4.1: La variable de premier niveau $x^* \in P$ étant fixée, considérons la formulation KKT du problème de PDN pour laquelle on suppose que les hypothèses 4.3 - 4.5 sont satisfaites. Étant donné $\mu \equiv 0$ fixé, soit $z(\mu)$ une solution réalisable du problème $PDN_{KKT}(\mu)$ telle que $z(\mu) = (x^*, y(x, \mu), w(x, \mu), v(x, \mu))$. Alors, il existe une unique solution réalisable $z^* = (x^*, y^*, w^*, v^*) = \lim_{\mu \rightarrow 0} z(\mu)$ du problème $PDN_{KKT}(\mu)$ telle que $(x^*, y^*) \in DI$.

Preuve: Lorsque x est fixé, considérons la fonction

$$(y, w, v) \rightarrow H(x, y, w, v) = \begin{pmatrix} \nabla_y f^2(x, y) + w^t \nabla_y g_J^2 \\ g_J^2(x, y) + v \end{pmatrix};$$

elle est de classe C^1 puisque les fonctions $y \rightarrow f^2(x, y)$ et $y \rightarrow g_J^2(x, y)$ sont de classe C^2 . De la proposition 3 de Facchinei et al. [1999], on a pour tout $\mu > 0$, la fonction

$$(y, w, v) \rightarrow \begin{pmatrix} H(x, y, w, v) \\ \Phi(w_i, v_i, \mu), i \in J \end{pmatrix}$$

est localement régulière (1), en raison des hypothèses 4.3 - 4.5. Soit $(x^*, y^*, w^*, v^*, \mu^*)$ avec $\mu^* \equiv 0$ tels que

$$\begin{cases} H(x^*, y^*, w^*, v^*) = 0, \\ \Phi(w_i^*, v_i^*, \mu^*) = 0, i \in J. \end{cases} \quad (4.2.11)$$

On déduit du théorème des fonctions implicites qu'il existe un voisinage \mathbb{V} de (x^*, μ^*) et une fonction de classe C^1 et inversible en raison de (1) de $(y, w, v) : \mathbb{V} \rightarrow \mathbb{R}^{n_y + n_w + n_v}$ telle que pour tout $(x, \mu) \in \mathbb{V}$, on a

$$\begin{cases} H(x, y(x, \mu), w(x, \mu), v(x, \mu)) = 0, \\ \Phi(w_i(x, \mu), v(x, \mu)_i, \mu) = 0, i \in J. \end{cases}$$

Soit $(x^*, \mu^*) \in \mathbb{V}$; l'unicité et l'existence du point $(x^*, y^*, w^*, v^*, \mu^*)$ tel que l'on ait (4.2.11), avec $(y^*, w^*, v^*) = (y(x^*, \mu^*), w(x^*, \mu^*), v(x^*, \mu^*))$ découlent des hypothèses 4.3 – 4.5 et du théorème des fonctions inverses. D'après la propriété 4.2.5, $\Phi(w_i^*, v_i^*, 0) = 0 \Rightarrow w_i^* v_i^* = 0 \forall i \in J$. Ce qui signifie que (x, y^*, w^*) est une solution réalisable du problème $PDN_{KKT}(0) \equiv PDN_{KKT}$. Par conséquent, $(x^*, y^*) \in DI$. ■

4.2.2 Conditions d'optimalité de l'approximation de la formulation KKT d'un problème de PDN

Le problème $PDN_{KKT}(\mu)$ est considéré comme une approximation du problème initial (PDN_{KKT}) lorsque μ est voisin de zéro. Il est connu que ces deux problèmes sont difficiles à résoudre. Lorsque $\mu = 0$, les problèmes (PDN_{KKT}) et $PDN_{KKT}(0)$ sont équivalents. Toutefois, le problème $PDN_{KKT}(\mu)$ est un problème classique d'optimisation localement régulier, les contraintes d'équilibre étant approximée par une fonction de mérite. Pour toute valeur de $\mu_k \geq 0$, toute suite de solutions réalisables de $PDN_{KKT}(\mu_k)$ possède un point d'accumulation du fait de la compacité de l'ensemble des solutions réalisables \mathfrak{F}^μ . Pour Facchinei et al. [1999], la suite des problèmes $PDN_{KKT}(\mu_k)$ peut être résolue par un algorithme classique de programmation mathématique. Pour illustrer cette assertion, ces auteurs ont effectué des expérimentations numériques concluantes portant sur des problèmes bi niveaux quadratiques de petites tailles connus de la littérature.

Soit $\bar{z} = (\bar{x}, \bar{y}, \bar{w}, \bar{v}) \in \mathfrak{F}^\mu$. Lorsque $\mu \rightarrow 0$, un vecteur tangent à \mathfrak{F}^μ en \bar{z} est la limite dz de n'importe quelle séquence $\{(z^k - \bar{z})/\tau_k\}$ où $z^k \in \mathfrak{F}^\mu$ et $\{\tau_k\}$ est une suite de scalaires positifs. Un point stationnaire du problème (PDN_{KKT}) tel que défini par Luo et al. [1996b] est un vecteur $\bar{z} \in \mathfrak{F}^0$ tel que $\nabla_{(x,y)} f^1(\bar{x}, \bar{y})^t \begin{pmatrix} dx \\ dy \end{pmatrix} \geq 0$ pour chaque $dz = (dx, dy, dw, dv)$ tangent à \mathfrak{F}^0 en \bar{z} . La propriété (4.2.5) relative à la fonction de mérite Φ nous permet d'établir qu'on a $\mathfrak{F}^0 = \mathfrak{F}^{KKT}$. Un point stationnaire pour le problème $PDN_{KKT}(\mu)$ est un point qui satisfait une condition nécessaire d'optimalité du premier ordre de ce problème.

Dans le développement qui suit, nous énonçons une propriété d'optimalité locale relative à l'approximation $PDN_{KKT}(\mu)$ du problème PDN_{KKT} lorsque $\mu \equiv 0$, en faisant appel au sous espace tangent en \bar{z} pour chaque domaine \mathfrak{F}^μ contenant \bar{z} , en lieu et place du cône tangent à \mathfrak{F}^0 . Cette propriété induit des conditions d'optimalité pour le problème PDN_{KKT} . Des hypothèses 4.3-4.5, on déduit que les gradients des contraintes actives en \bar{z} du problème $PDN_{KKT}(\mu)$ sont linéairement indépendants. Luo et al. [1996a] ou encore Fukushima et al. [1998], ont proposé des algorithmes de programmation quadratique séquentielle pour résoudre un problème de MPEC de type (PDN_{KKT}). Les contraintes de ce programme sont obtenues à partir d'une approximation tangentielle des contraintes du problème $PDN_{KKT}(\mu_k)$. Par contre, le principe d'un algorithme du type *SLP* appliqué à la résolution de chaque problème $PDN_{KKT}(\mu_k)$ consiste à déterminer une direction de descente $dz = (dx, dw, dv)$ (avec dz suffisamment petit) à partir d'une solution réalisable $z^k \in \mathfrak{F}^{\mu_k}$, en résolvant à chaque itération le programme linéaire $PL(z^k, \mu_k)$ ci-dessous obtenu à partir d'une approximation tangentielle du problème $PDN_{KKT}(\mu_k)$ au voisinage de la solution courante $z^k = (x^k, w^k, v^k)$. Soit $H_i^k = H_i(z^k)$; une direction de descente est telle qu'entre autre on a $w^k + \tau dw \geq 0$ et $v^k + \tau dv \geq 0$, où $\tau \in \mathbb{R}_+^*$ est le pas de déplacement. On a donc:

$PL(z^k, \mu_k)$

$$\begin{aligned}
 & \text{Maximiser}_{\xi, dx, dy, dw, dv} \xi \\
 & \left\{ \begin{array}{l}
 \nabla_{(x,y)} f^1(x^k, y^k)^t \begin{pmatrix} dx \\ dy \end{pmatrix} + \xi \leq 0, \quad (C1) \\
 \nabla_z H_i^k(dz)^t = 0, \quad i \in J, \quad (C2) \\
 \Phi(w_i^k, v_i^k, \mu_k) + \frac{\partial \Phi(w_i^k, v_i^k, \mu_k)}{\partial w_i} dw_i + \frac{\partial \Phi(w_i^k, v_i^k, \mu_k)}{\partial v_i} dv_i = 0, \quad i \in J, \quad (C3) \\
 -\tau(dw_i, dv_i) \leq (w_i^k, v_i^k), \quad i \in J, \quad (C4) \\
 (x^k + \tau dx, y^k + \tau dy, w^k + \tau dw) \in P \times \mathbb{R}^{2n_y}, \quad \xi \geq 0.
 \end{array} \right. \quad (4.2.12)
 \end{aligned}$$

Mais en pratique, ne connaissant pas la valeur du pas de déplacement τ , on peut soit omettre la contrainte (C4), soit choisir $\tau=1$ dans la formulation du programme linéaire $PDN_{KKT}(\mu_k)$. Sa résolution revient à déterminer une direction de descente $dz = (dx, dw, dv)$ pour la fonction économique f^1 , mais en imposant à cette direction de ne pas s'écarter, au moins localement de la frontière du domaine des solutions réalisables \mathfrak{F}^{μ_k} du problème $PDN_{KKT}(\mu_k)$ par l'intermédiaire de la contrainte (C1) et de la variable ξ . En effet, si l'optimum du problème $PL(z^k, \mu_k)$ correspond à $\xi^* > 0$, on voit que $\nabla_{(x,y)} f^1(x^k, y^k)^t \begin{pmatrix} dx \\ dy \end{pmatrix} < 0$ et $\begin{pmatrix} dx \\ dy \end{pmatrix}$ est bel et bien une direction de descente.

La variation $dz = (dx, dy, dw, dv)$ est une direction de descente pour la fonction f^1 tant que $\xi^* > 0$; lorsque $\mu_k \equiv 0$, nous montrons à travers le corollaire ci-après que si $\xi^* = 0$, il existe un vecteur des multiplicateurs de KKT $(\alpha_0, \alpha, \gamma, \theta) \in \mathbb{R}_+ \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}_+^{2m}$ du problème $PDN_{KKT}(\mu_k)$ tels que $\theta_w^t w^k + \theta_v^t v^k = 0$; de plus, la solution courante $z^k = (x^k, y^k, w^k, v^k)$ est une solution réalisable du problème (PDN_{KKT}) . Un résultat similaire a été établi par Facchinei et al. [1999] (théorème 2) pour des problèmes de MPEC dont les contraintes sont définies par une inégalité variationnelle.

Corollaire 4.1: Sous les hypothèses 4.3 - 4.5, si $\lim_{k \rightarrow \infty} \mu_k = 0$ lorsque, en $z^k \in \mathfrak{F}^{\mu_k}$ la valeur optimale de la fonction économique du problème $PL(z^k, \mu_k)$ est telle que $\xi^* = 0$, alors il existe un vecteur des multiplicateurs de KKT $(\alpha_0, \alpha, \gamma, \theta) \in \mathbb{R}_+ \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}_+^{2m}$ du problème $PDN_{KKT}(\mu_k)$ tel que $\theta_w^t w^k + \theta_v^t v^k = 0$; de plus $z^k = (x^k, y^k, w^k, v^k)$ est une solution réalisable du problème (PDN_{KKT}) .

Preuve: En associant aux contraintes du programme linéaire $PL(z^k, \mu_k)$ (lorsqu'il a une solution) des variables duales $(\alpha_0, \alpha, \gamma, \theta) \in \mathbb{R}_+ \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}_+^{2m}$, le dual de $PL(z^k, \mu_k)$ s'écrit

$DPL(z^k, \mu_k)$

$$\begin{aligned} & \text{Minimiser } \sum_{i \in J} -\gamma_i \Phi(w_i^k, v_i^k, \mu_k) + \theta_w^t w^k + \theta_v^t v^k \\ & \text{s.c. } \begin{cases} \alpha_0 \geq 1, \\ \alpha_0 \nabla_{(x,y)} f^1(x^k, y^k) + \sum_{i \in J} \alpha_i \nabla_{(x,y)} H_i^k = 0, \\ \sum_{i \in J} \alpha_i \nabla_{(w_i, v_i)} H_i^k + \sum_{i \in J} \gamma_i \nabla_{(w_i, v_i)} \Phi(w_i^k, v_i^k, \mu_k) - \tau(\theta_w, \theta_v) = 0, \\ \alpha_0, \theta_w, \theta_v \geq 0, \alpha_i \in \mathbb{R}, \gamma_i \in \mathbb{R}, i \in J. \end{cases} \end{aligned} \quad (4.2.13)$$

Sans nuire à la généralité, on peut supposer que $\alpha_0 = 1$. Des hypothèses 4.3 - 4.5, on déduit que le système des contraintes (4.2.13) admet une solution réalisable unique. Ce qui établit l'existence d'un vecteur des multiplicateurs de KKT du problème $PDN_{KKT}(\mu_k)$, qui coïncide avec le vecteur des variables duales du problème $PL(z^k, \mu_k)$ qui est tel que

$$\begin{cases} \nabla_{(x,y)} f^1(x^k, y^k) + \sum_{i \in J} \alpha_i \nabla_{(x,y)} H_i^k = 0, \\ \sum_{i \in J} \alpha_i \nabla_{(w_i, v_i)} H_i^k + \sum_{i \in J} \gamma_i \nabla_{(w_i, v_i)} \Phi(w_i^k, v_i^k, 0) \geq 0, \\ \alpha_i \in \mathbb{R}, \gamma_i \in \mathbb{R}, i \in J, \end{cases} \quad (4.2.14)$$

puisque l'on a $\tau(\theta_w, \theta_v) \geq 0$. Comme par hypothèse $\Phi(w_i^k, v_i^k, \mu_k) = 0$ pour tout $i \in J$, du théorème de la dualité en programmation linéaire, on a:

$$\xi^* = \sum_{i \in J} -\gamma_i \Phi(w_i^k, v_i^k, 0) + \theta_w^t w^k + \theta_v^t v^k = \theta_w^t w^k + \theta_v^t v^k = 0.$$

Lorsque $\mu_k \rightarrow 0$, z^k est une solution réalisable du problème (PDN_{KKT}) d'après la proposition 4.1. ■

Appliqué à la résolution d'un problème de MPEC du type (PDN_{KKT}) , le principe de l'algorithme *SLP* consiste à résoudre la suite des problèmes localement réguliers $PDN_{KKT}(\mu_k)$ en s'assurant que μ_k converge vers zéro. La suite des solutions $z^k = (x^k, y^k, w^k, v^k)$ converge alors vers le point d'accumulation $\bar{z} = (\bar{x}, \bar{y}, \bar{w}, \bar{v})$ telle que $(\bar{x}, \bar{y}) \in DI$.

4.3 Méthode d'actualisation du paramètre utilisé dans une fonction de mérite

En utilisant le développement limité du premier ordre de la fonction définie de $\mathbb{R}_+^2 \rightarrow \mathbb{R}$ qui à $(w, v) \mapsto wv$, nous proposons une méthode qui permet d'actualiser à chaque itération le paramètre μ qui est utilisé dans la fonction de mérite de Fisher Burmeister, sur la base des seules données du problème résolu. Nous illustrons la méthode proposée en appliquant un algorithme *SLP* à la résolution d'un programme de type $PDN_{KKT}(\mu_k)$.

Étant donnée une suite de points $\{w^k v^k\}_{k \in K}$ telle que pour tout $k \in K$, $w^k \geq 0$ et $v^k \geq 0$, nous énonçons à travers la proposition suivante une condition nécessaire et suffisante relative aux variations dw et dv pour que la suite $\{w^k v^k\}_{k \in K}$ soit décroissante.

Proposition 4.2: Considérons une suite de réels $\{w^k v^k\}_{k \in K}$ telle que pour tout $k \in K$, $w^k \geq 0$, $v^k \geq 0$, il existe $\tau > 0$ suffisamment petit tel que $w^{k+1} = w^k + \tau dw$,

$v^{k+1} = v^k + \tau dv$. Une condition nécessaire et suffisante pour que la suite $\{w^k v^k\}_{k \in K}$ soit décroissante est que l'on ait: $w^k dv + v^k dw \leq 0$.

Preuve: Comme $\tau > 0$ est voisin de zéro, il suffit d'écrire que l'on a:

$$w^{k+1} v^{k+1} = w^k v^k + \tau(w^k dv + v^k dw) + o(\tau). \quad \blacksquare$$

Étant donné un ensemble d'indices $I = \{1, 2, \dots, n_y\}$, soit $i \in I$; considérons la fonction de mérite de Fisher Burmeister Φ . Nous montrons que si la suite $\{w_i^k v_i^k\}_{k \in K}$ est décroissante et converge vers zéro, et si $\mu_k = \max_{i \in I} (w_i^k v_i^k)$ pour tout $k \in K$, alors la suite $\{\Phi(w_i^k, v_i^k, \mu_k)\}_{k \in K}$ est croissante à valeurs négatives ou nulles et converge vers zéro. La proposition ci-après énonce ce résultat.

Proposition 4.3: Soit $i \in I$ et considérons la suite $\{w_i^k v_i^k\}_{k \in K}$ avec $w_i^k \geq 0$ et $v_i^k \geq 0$. Soit $\Phi_i^k = \Phi(w_i^k, v_i^k, \mu_k)$, $(dw_i, dv_i, d\mu)$ et $\tau > 0$ voisin de zéro tels que

$$(w_i^{k+1}, v_i^{k+1}, \mu_{k+1}) = (w_i^k + \tau dw_i, v_i^k + \tau dv_i, \mu_k + \tau d\mu). \quad (4.3.1)$$

On suppose que l'on a:

- i) $\mu_k = \max_{i \in I} (w_i^k v_i^k)$ pour tout $k \in K$;
- ii) la suite $\{w_i^k v_i^k\}_{k \in K}$ est décroissante et converge vers zéro.

Alors on a:

- a) Pour tout $i \in I$, la variation $d\mu$ vérifie $w_i^k dv_i + v_i^k dw_i - d\mu \geq 0$.
- b) Pour tout $i \in I$, la suite $\{\Phi_i^k\}_{k \in K}$ est croissante et converge vers zéro.

Preuve:

a) Etant donné un indice $i \in I$, considérons la suite $\{w_i^k v_i^k\}_{k \in K}$ telle que

$(w_i^{k+1}, v_i^{k+1}, \mu_{k+1}) = (w_i^k + \tau dw_i, v_i^k + \tau dv_i, \mu_k + \tau d\mu)$ où $\tau > 0$ voisin de zéro. On peut écrire:

$$w_i^{k+1} v_i^{k+1} = w_i^k v_i^k + \tau(w_i^k dv_i + v_i^k dw_i) + o(\tau). \text{ Comme } w_i^{k+1} v_i^{k+1} \leq w_i^k v_i^k, \text{ on a}$$

$$w_i^k dv_i + v_i^k dw_i \leq 0.$$

De même, puisque $\mu_{k+1} \leq \mu_k$, on a $d\mu \leq 0$. On peut trouver $i_0 \in I$ et $j_0 \in I$ tels que

$$\begin{aligned} \mu_{k+1} &= \max_{i \in I} w_i^{k+1} v_i^{k+1} = w_{i_0}^k v_{i_0}^k + \tau(w_{j_0}^k dv_{j_0} + v_{j_0}^k dw_{j_0}) \\ &= \max_{i \in I} w_i^k v_i^k + \tau \min_{j \in I} (w_j^k dv_j + v_j^k dw_j) \\ &= \mu_k + \tau d\mu. \end{aligned} \quad (4.3.2)$$

On en déduit que $d\mu = \min_{j \in I} w_j^k dv_j + v_j^k dw_j$. D'où le résultat.

b) Soit $k \in K$; comme $\mu_k = \max_{i \in I} (w_i^k v_i^k)$, on a $\Phi_i^k \leq 0$ pour tout $i \in I$. Le développement

limité d'ordre un de Φ_i au voisinage de (w_i^k, v_i^k, μ_k) s'écrit:

$$\begin{aligned} \Phi_i^{k+1} &= \Phi_i^k + \tau \left(\frac{\partial \Phi_i^k}{\partial w_i} dw_i + \frac{\partial \Phi_i^k}{\partial v_i} dv_i + \frac{\partial \Phi_i^k}{\partial \mu} d\mu \right) + o(\tau) \\ &= \Phi_i^k + \tau \left(\left(1 - \frac{w_i^k}{\sqrt{(w_i^k)^2 + (v_i^k)^2 + 2\mu_k}}\right) dw_i + \left(1 - \frac{v_i^k}{\sqrt{(w_i^k)^2 + (v_i^k)^2 + 2\mu_k}}\right) dv_i \right. \\ &\quad \left. - \frac{d\mu}{\sqrt{(w_i^k)^2 + (v_i^k)^2 + 2\mu_k}} \right) + o(\tau). \end{aligned} \quad (4.3.3)$$

En remplaçant μ_k par $w_i^k v_i^k$, puisque $\mu_k \geq w_i^k v_i^k$, pour tout $i \in I$, on a

$$\Phi_i^{k+1} \geq \Phi_i^k + \tau \frac{w_j^k dv_j + v_j^k dw_j - d\mu}{w_i^k + v_i^k}. \quad (4.3.4)$$

Mais $w_j^k dv_j + v_j^k dw_j - d\mu \geq 0$ d'après a). Donc $\Phi_i^{k+1} \geq \Phi_i^k$.

La suite $\{\Phi_i^k\}_{k \in K}$ est croissante et on a $\Phi_i^k \leq 0 \forall i$. Comme la suite $\{w_i^k y_i^k\}_{k \in K}$ est décroissante et converge vers zéro, il en est de même pour la suite $\{\mu_k\}_{k \in K}$; Φ_i étant une NCP fonction, la suite $\{\Phi_i^k\}_{k \in K}$ converge vers zéro. ■

Le résultat ci-dessus constitue une innovation pour ce qui est de l'utilisation des fonctions de mérite de paramètre μ du type Fisher Burmeister. Seules les données du problème résolu conditionnent l'actualisation du paramètre μ . C'est la première fois à notre connaissance que l'on propose de procéder ainsi. L'actualisation du paramètre qui intervient dans la fonction de mérite de Fisher, Kanzow et Smale que nous avons définie dans le chapitre 3 pourrait se faire de manière similaire. Nous implémentons cette méthode d'actualisation du paramètre μ avec l'algorithme *SLP* dans la sous-section 4.4.

Pour résoudre les problèmes de MPEC ou des problèmes de complémentarité, les auteurs qui ont fait appel aux fonctions de mérite paramétrées actualisent ce paramètre de manière arbitraire. À titre d'illustration, afin de garantir la décroissance de la suite $\{\mu_k\}_{k \in K}$, Fukushima et al. [1998] ou encore Facchinei et al. [1999] proposent de choisir un coefficient $\beta \in [0, 1[$ en posant $\mu_{k+1} = \beta \mu_k$. Cette méthode d'actualisation du paramètre μ permet d'établir la convergence des algorithmes proposés par ces auteurs. Puisque $\beta^k \mu_0 = \mu_k$, il faudrait noter que cette méthode impose à l'algorithme d'exécuter un nombre minimal d'itérations $It \min = \ln(\frac{\varepsilon}{\mu_0}) / \ln(\beta)$ où ε est le seuil de la précision des calculs. Pour actualiser μ , Fukushima et al. [1998] ont envisagé comme possibilité d'extension de recherche, l'usage d'une suite bornée $\{\beta_k\}_{k \in K}$ telle que $\underline{\beta} < \beta_k < \bar{\beta}$ avec $0 < \underline{\beta} < \bar{\beta} < 1$; ils suggèrent d'actualiser alors μ en posant $\mu_{k+1} = \beta_k \mu_k$. On peut noter que si $\mu_k < \max_{i \in I} (w_i^{k-1}, v_i^{k-1})$, on peut trouver au moins deux indices $i_+ \in I$ tel que $\Phi(w_{i_+}^k, v_{i_+}^k, \mu_k) \geq 0$ et $i_- \in I$ tel que $\Phi(w_{i_-}^k, v_{i_-}^k, \mu_k) \leq 0$. Le fait de poser pour tout $i \in I$

$$\Phi^{k+1} = \Phi(w_i^k, v_i^k, \mu_k) + \nabla_{w_i, v_i} \Phi(w_i^k, v_i^k, \mu_k) \begin{pmatrix} dw_i \\ dv_i \end{pmatrix} = 0$$

à chaque itération n'est pas rigoureusement correct.

4.4 Résolution d'un problème de MPEC ou de complémentarité par l'algorithme SLP

L'algorithme SLP peut constituer un excellent outil pour résoudre un problème de complémentarité ou un problème de MPEC dont les contraintes sont linéaires.

4.4.1 Résolution d'un problème de MPEC

Considérons le problème de MPEC ci-après:

$MPEC(M, N, q)$

$$\begin{aligned} & \text{minimiser}_{x, y, w} f^1(x, y) \\ & \text{s.c.} \begin{cases} w = q + Nx + My, \\ 0 \leq w \perp y \geq 0, \\ (x, y, w) \in X \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}, \end{cases} \end{aligned} \quad (4.4.1)$$

où $f^1 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ est de classe C^1 , $N \in \mathbb{R}^{n_y \times n_x}$, $M \in \mathbb{R}^{n_y \times n_y}$, $q \in \mathbb{R}^{n_y}$, l'ensemble X est un polyèdre défini par $X \equiv \{x \in \mathbb{R}_+^{n_x} : Ax \leq b\}$, avec $A \in \mathbb{R}^{m \times n_x}$ et $b \in \mathbb{R}^m$. Soit \mathfrak{F} son ensemble des solutions réalisables et $T(z^*, \mathfrak{F})$ le cône tangent en z^* à \mathfrak{F} . $T(z^*, \mathfrak{F})$ faudrait-il le rappeler, est un cône non convexe et il se met sous la forme d'une union finie de cônes polyédraux. La plupart des propriétés théoriques relatives à un tel problème ont été établies par Fukushima et al. [1998]. On suppose que la matrice M est une P_0 -matrice et de l'hypothèse 4.5, on déduit que toute solution $(\bar{x}, \bar{y}, \bar{w})$ réalisable de ce problème est non dégénérée c'est-à-dire $\bar{y}_i + \bar{w}_i \neq 0 \forall i$.

Nous avons vu au chapitre précédent qu'une solution réalisable $z^* = (x^*, y^*, w^*) \in \mathfrak{F}$ est un point stationnaire pour le problème $MPEC(M, N, q)$ si on a

$$(dx, dy, dw) \in T(z^*, \mathfrak{F}) \Rightarrow \nabla f^1(x^*, y^*) \begin{pmatrix} dx \\ dy \end{pmatrix} \geq 0.$$

Sous l'hypothèse de non dégénérescence 4.5, la propriété ci-après donne un résumé des conditions d'optimalité locale du problème $MPEC(M, N, q)$.

Propriété 4.1 (Corollaire 6.1.3, Luo et al. 1996b): Considérons $z^* = (x^*, y^*, w^*) \in \mathfrak{F}$ une solution réalisable du problème $MPEC(M, N, q)$ telle que l'hypothèse de non dégénérescence 4.5 est satisfaite, et M est une P_0 -matrice. Alors, z^* est un point stationnaire du problème $MPEC(M, N, q)$ si et seulement si il existe des multiplicateurs $(\eta^*, \nu^*, \lambda^*) \in \mathbb{R}^{2n_y + m}$ tels que:

$$\begin{pmatrix} \nabla f^1(x^*, y^*) \\ 0 \end{pmatrix} + \begin{pmatrix} N^t \\ M^t \\ -I_{n_y} \end{pmatrix} \eta^* + \begin{pmatrix} 0 \\ W^* \\ Y^* \end{pmatrix} \nu^* + \begin{pmatrix} A^t \\ 0 \\ 0 \end{pmatrix} \lambda^* = 0,$$

$$(\lambda^*)^t (Ax - b) = 0,$$

où W^* et Y^* sont des matrices diagonales dont les coefficients diagonaux respectifs sont w_i^* et y_i^* , $i = 1, \dots, n_y$.

Dans l'algorithme *SLP*, la contrainte d'équilibre $w^t y = 0$ est approximée par une fonction de mérite; ce qui conduit à un problème voisin paramétré qui a pour formulation:

$MPEC(M, N, q, \mu)$

Minimiser $_{x, y, w} f^1(x, y)$

$$\text{s.c.} \begin{cases} w = q + Nx + My, \\ \Phi(w_i, y_i, \mu) = 0, \quad i = 1, 2, \dots, n_y, \\ (x, y, w) \in X \times \mathbb{R}_+^{n_y} \times \mathbb{R}_+^{n_y}. \end{cases} \quad (4.4.2)$$

Soit (x^k, y^k, w^k) un point qui satisfait l'égalité ci-après:

$$w^k = q + Nx^k + My^k \text{ et } (x^k, y^k, w^k) \in X \times \mathbb{R}_+^{n_x} \times \mathbb{R}_+^{n_y}$$

Pour $\mu_k > 0$, v étant une variable d'écart, afin de mettre en œuvre l'algorithme SLP, nous proposons de résoudre le programme linéaire suivant:

$$\begin{aligned}
 & PL(\mu_k) \\
 & \text{Maximiser}_{v, dx, dy, dw, \xi} \xi \\
 & \text{s.c.} \begin{cases} \nabla_{(x,y)} f^1(x^k, y^k)^t \begin{bmatrix} dx \\ dy \end{bmatrix} + \xi + v = 0, \\ dw = Ndx + Mdy, \\ \Phi(w_i^k, y_i^k, \mu_k) + \nabla_{(w_i, y_i)} \Phi(w_i^k, y_i^k, \mu_k) \begin{bmatrix} dw_i \\ dy_i \end{bmatrix} = 0, i \in I, \\ (x^k + dx, y^k + dy, w^k + dw) \in X \times \mathbb{R}_+^{n_x} \times \mathbb{R}_+^{n_y}, \xi, v \geq 0, \end{cases} \quad (4.4.3)
 \end{aligned}$$

où $\nabla_{(w_i, y_i)} \Phi(w_i^k, y_i^k, \mu_k)$ $i \in I$ sont les composantes de la matrice jacobienne définie par

$$\nabla_{(w,y)} \Psi(w^k, y^k, \mu_k) = \begin{pmatrix} D_w(w^k, y^k, \mu_k) \\ D_y(w^k, y^k, \mu_k) \end{pmatrix} \equiv \begin{pmatrix} D_{w_i}^k \\ D_{y_i}^k \end{pmatrix}_{i \in I}.$$

Comme dans la formulation du programme linéarisé (4.2.12), les contraintes de positivité des variables sont prises en compte. Les contraintes du programme linéaire $PL(\mu_k)$ peuvent encore se mettre sous la forme

$$\begin{aligned}
 & \begin{bmatrix} \nabla_x f^1(x^k, y^k) & \nabla_y f^1(x^k, y^k) & 0 & 1 \\ N & M & -I_m & 0 \\ 0 & D_w^k & D_y^k & 0 \end{bmatrix} \begin{pmatrix} dx \\ dy \\ dw \\ \xi \end{pmatrix} = - \begin{pmatrix} v \\ 0 \\ \Psi(w^k, y^k, \mu_k) \end{pmatrix} \\
 & \Leftrightarrow A_k \begin{pmatrix} dy \\ dw \\ \xi \end{pmatrix} = - \begin{pmatrix} \nabla_x f^1(x^k, y^k) dx + v \\ Ndx \\ \Psi(w^k, y^k, \mu_k) \end{pmatrix},
 \end{aligned}$$

la direction de déplacement (dx, dy, dw) étant telle que $(x^k + dx, y^k + dy, w^k + dw) \in P \times \mathbb{R}_+^{2n_y}$, et

$$A_k = \begin{bmatrix} \nabla_y f^1(x^k, y^k) & 0 & 1 \\ M & -I_m & 0 \\ D_w^k & D_y^k & 0 \end{bmatrix}. \quad (4.4.4)$$

Sous l'hypothèse 4.5, la matrice A_k est inversible M étant une P_0 -matrice, ce qui assure l'unicité de la solution du problème $PL(\mu_k)$. Nous le démontrons à partir de la proposition suivante:

Proposition 4.4: Soit $z^k = (x^k, y^k, w^k)$ une solution du problème $MPEC(M, N, q, \mu_k)$ telle que M est une P_0 -matrice, $\mu_k > 0$ étant donné. Sous l'hypothèse 4.5, lorsque le problème $PL(\mu_k)$ admet une solution, celle-ci est unique.

Preuve: Supposons que le programme linéaire $PL(\mu_k)$ a une solution (ξ, dx, dy, dw) telle que $(x^k + dx, y^k + dy, w^k + dw) \in X \times \mathbb{R}_+^{n_y} \times \mathbb{R}_+^{n_y}$. Puisque la matrice M est une P_0 -matrice et comme D_w^k et D_y^k sont des matrices diagonales à coefficients non nuls, les relations de complémentarité étant non dégénérées, Fukushima et al. [1998] (proposition 3.1) ont établi que la matrice

$$\begin{bmatrix} M & -I_m \\ D_w^k & D_y^k \end{bmatrix}$$

est régulière. Il en est de même de la matrice A_k . En utilisant la substitution

$$\begin{pmatrix} dy \\ dw \\ \xi \end{pmatrix} = -A_k^{-1} \begin{pmatrix} \nabla_x f^1(x^k, y^k) dx + v \\ N dx \\ \Psi(w^k, y^k, \mu_k) \end{pmatrix}, \quad (4.4.5)$$

on peut réduire le problème $PL(\mu_k)$ en un programme linéaire équivalent de la seule variable dx avec la contrainte $x^k + dx \in X$. Comme $dx = 0$ satisfait de manière triviale la dernière contrainte du programme linéaire $PL(\mu_k)$, puisque $x^k \in X$, on en déduit que le problème $PL(\mu_k)$ a une solution optimale unique. ■

Le problème $PL(\mu_k)$ permet de déterminer une direction de descente pour le problème $MPEC(M, N, q, \mu)$ à partir d'une solution réalisable $z^k = (x^k, y^k, w^k)$. En pratique, le calcul d'une direction de descente $dz = (dx, dy, dw)^t$ peut se faire en résolvant le programme linéaire ci-après :

$$\begin{array}{l}
 PL(z^k, \mu_k) \\
 \text{Maximiser}_{dx, dy, dw, \xi} \xi \\
 \left. \begin{array}{l}
 \nabla_{(x,y)} f^1(x^k, y^k) \begin{bmatrix} dx \\ dy \end{bmatrix} + \xi \leq 0, \\
 dw = Ndx + Mdy, \\
 \Psi(w^k, y^k, \mu_k) + \nabla_{(w,y)} \Psi(w^k, y^k, \mu_k) \begin{bmatrix} dw \\ dy \end{bmatrix} = 0, \\
 (x^k + dx, y^k + dy, w^k + dw) \in X \times \mathbb{R}_+^{n_y} \times \mathbb{R}_+^{n_w}, \quad \xi \geq 0.
 \end{array} \right\} \text{s.c.}
 \end{array}$$

D'après le corollaire 4.1, lorsqu'une solution optimale $(\xi^*, dx^*, dy^*, dw^*)$ du programme linéaire $PL(z^k, \mu_k)$ est telle que $\xi^* = 0$ avec $\mu_k \equiv 0$, les variables duales de $PL(z^k, 0)$ sont les multiplicateurs de KKT du problème $MPEC(M, N, q, \mu)$. En définitive, l'algorithme *SLP* adapté à la résolution du problème $MPEC(M, N, q)$ se présente ainsi qu'il suit :

Algorithme SPL: Résolution du problème ($MPEC(M, N, q)$)

Étape 0 (initialisation)

Soient $\varepsilon > 0$ (seuil de tolérance) et $\beta > 0$ deux constantes suffisamment petites, et

$z^0 = (x^0, y^0, w^0)$ une solution réalisable de la relaxation du problème $MPEC(M, N, q)$ telle

que $\mu_0 = \max_i \{w_i^0 y_i^0\}$. Poser $k = 0$ et aller à l'Étape 1.

Étape 1 (Pas et direction de déplacement)

- i) Calculer une direction de descente d^k pour la fonction f^1 à partir du point z^k en résolvant le programme linéaire $PL(z^k, \mu_k)$ dont la valeur optimale est ξ^* . Si $(\xi^* \leq \varepsilon$ ou $dz = 0)$ et $\mu_k > \varepsilon$, aller à l'étape 2.
- ii) En utilisant la règle d'Armijo, calculer le pas de déplacement τ tel que $z^k + \tau d^k \in \mathcal{F}^{\mu_k}$ et $f^1(z^k + \tau d^k) \leq f^1(z^k)$. Poser $z^{k+1} = z^k + \tau d^k$, puis actualiser $\mu_{k+1} = \max_i \{w_i^k y_i^k\}$, poser $k = k + 1$ et aller à l'Étape 3.

Étape 2 (pas de déplacement nul)

Poser $\mu_{k+1} = \beta \mu_k$, $k = k + 1$ et aller à l'Étape 1.

Étape 3 (Critère d'arrêt)

Si $\mu_k + \xi^* \leq \varepsilon$ ou si $\|\Psi(w^k, y^k, \mu_k)\|_2 + \xi^* \leq \varepsilon$, Stop; z^k est un point stationnaire du problème $MPEC(M, N, q, \mu_k)$ avec $\mu_k \equiv 0$. Sinon aller à l'Étape 1.

Contrairement à l'algorithme générique de descente que nous avons présenté dans la section précédente, le critère d'arrêt de l'algorithme *SLP* ci-dessus prend en compte, outre la convergence vers zéro de la suite $\{\mu_k\}$ relative à la valeur du paramètre μ , mais aussi la valeur optimale de la fonction économique du programme linéaire $PL(z^k, \mu_k)$. Nous établissons à travers le théorème ci-après, la convergence de l'algorithme *SLP* précédent.

Théorème 4.2: Considérons le problème $MPEC(M, N, q)$. On suppose que M est une P_0 -matrice, que la suite des points $\{z^k\}$ générée par l'algorithme *SLP* est bornée et que l'hypothèse de non dégénérescence 4.5 est satisfaite. Soit $z^* = (x^*, y^*, w^*) = \lim_{k \rightarrow \infty} z^k$ tel que $\lim_{k \rightarrow \infty} \mu_k = 0$. Si la valeur optimale du programme linéaire $PL(z^*, 0)$ est telle que $\xi^* = 0$, alors z^* est un point stationnaire du problème $MPEC(M, N, q)$.

Preuve: D'après la proposition 4.4, la suite $\{z^k\}$ converge vers l'unique point d'accumulation $z(0) = \lim_{k \rightarrow \infty} z(\mu_k)$, comme $z^k = z(\mu_k)$ et $\lim_{k \rightarrow \infty} \mu_k = 0$. Le dual du programme linéaire $PL(z^k, \mu_k)$ a pour formulation:

$$\begin{aligned}
 & DPL(z^k, \mu_k) \\
 & \text{Minimiser } \sum_{i \in I} -v^i \Psi(w^k, v^k, \mu_k) + \theta_x^i x^k + \theta_y^i y^k + \theta_w^i w^k + \lambda^t (b - Ax^k) \\
 & \text{s.c. } \left\{ \begin{array}{l} \alpha_0 \geq 1, \\ \alpha_0 \begin{pmatrix} \nabla f^t(x^k, y^k) \\ 0 \end{pmatrix} + \begin{pmatrix} N^t \\ M^t \\ -I_{n_y} \end{pmatrix} \eta + \begin{pmatrix} 0 \\ Y^k \\ W^k \end{pmatrix} v + \begin{pmatrix} A^t \\ 0 \\ 0 \end{pmatrix} \lambda = \begin{pmatrix} I_{n_x} \theta_x \\ I_{n_y} \theta_y \\ I_{n_w} \theta_w \end{pmatrix}, \\ (\theta_x, \theta_y, \theta_w) \in \mathbb{R}_+^{n_x} \times \mathbb{R}_+^{n_y} \times \mathbb{R}_+^{n_w} \geq 0, (\lambda, \eta, v) \in \mathbb{R}_+^m \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_w}. \end{array} \right.
 \end{aligned}$$

où W^k et Y^k sont des matrices diagonales dont les coefficients diagonaux respectifs sont w_i^k et y_i^k , $i = 1, \dots, n_y$, $(\theta_x, \theta_y, \theta_w)$ sont les multiplicateurs liés aux bornes des variables (dx, dy, dw) respectivement.

Sans nuire à la généralité, on peut supposer que $\alpha_0 = 1$. À l'optimalité, si on désigne par $(\eta^*, v^*, \lambda^*, \theta_x^*, \theta_y^*, \theta_w^*)$ le vecteur des variables duales du programme linéaire $PL(z^*, \mu_k)$, on a nécessairement $dz = 0$. Ce qui entraîne que l'on a, d'après les relations de complémentarité $\theta_y^{*t} y^* + \theta_y^{*t} y^* + \theta_w^{*t} w^* = 0$ et $\lambda^{*t} (b - Ax^*) = 0$. Comme $\mu_k \rightarrow 0$, du théorème de la dualité en programmation linéaire et de la propriété 4.2.5, on a pour tout $i \in I$, $\Phi(w_i^k, y_i^k, 0) = 0 = \xi^*$, où $\Phi(w_i^k, y_i^k, 0)$ représente une des composantes de $\Psi(w^k, y^k, 0)$. Donc $w_i^* \cdot y_i^* = 0$ ($\forall i \in I$).

Les multiplicateurs de KKT du problème $MPEC(M, N, q)$ sont approximés par les variables duales du programme linéaire $PL(z^*, 0)$, à savoir le vecteur (η^*, v^*, λ^*) . z^* est donc un point stationnaire du problème $MPEC(M, N, q)$. ■

Contrairement à l'algorithme SQP de Fukushima et al. [1998], les contraintes de positivité des variables sont prises en compte dans le calcul d'une direction de descente à chaque itération de l'algorithme SLP. Les tests numériques relatifs à cet algorithme sont présentés dans la section 4.5.

4.4.2 Résolution d'un problème de complémentarité

Nous considérons à présent le problème général de complémentarité qui consiste à trouver une solution réalisable au système suivant:

$$(GCP) \quad \begin{cases} 0 \leq y \perp H(x, y) \geq 0, \\ (x, y) \in X \times Y, \end{cases} \quad (4.4.6)$$

où $H : \mathbb{R}_+^{n_x} \times \mathbb{R}_+^{n_y} \rightarrow \mathbb{R}_+^{n_y}$, $X \equiv \{x \in \mathbb{R}_+^{n_x} : Ax \leq b\}$ avec $A \in \mathbb{R}^{m \times n_x}$, $b \in \mathbb{R}^m$, $Y \equiv \mathbb{R}_+^{n_y}$.

On suppose que pour $x \in X$ fixé, la fonction $y \rightarrow H(x, y)$ est une P_0 -fonction. L'ensemble des solutions réalisables du problème (GCP), noté \mathfrak{F}^{GCP} est supposé non vide, et nous désignons par \mathfrak{F}_r^{GCP} , l'ensemble des solutions réalisables du problème (GCP) relaxé des contraintes de complémentarité. Lorsque la fonction $H(x, y)$ est affine, on peut la mettre sous la forme $H(x, y) = Nx + My - q$ où $N \in \mathbb{R}^{n_y \times n_x}$, $M \in \mathbb{R}^{n_y \times n_y}$, $q \in \mathbb{R}_+^{n_y}$ et M est une P_0 -matrice. On obtient ainsi un problème de complémentarité linéaire généralisé. Si les relations de complémentarité induites par la relation $y^t H(x, y) = 0$ sont non dégénérées ($y_i + H_i(x, y) \neq 0$ pour tout $1 \leq i \leq n_y$) d'après l'hypothèse 4.5, l'algorithme SLP est un excellent outil pour résoudre le problème (GCP). Pour cela, considérons le problème de MPEC suivant:

$$(QP) \quad \begin{aligned} & \text{Minimiser } f(y, z) = \frac{1}{2} y^t z \\ & \text{s.c. } \begin{cases} Ax \leq b, \\ H(x, y) - z = 0, \\ y^t z = 0, \\ x, y, z \geq 0, \end{cases} \end{aligned} \quad (4.4.7)$$

où z est une variable d'écart. Si la valeur de la solution optimale du programme (QP) est nulle, alors, la solution correspondante est une solution du problème de complémentarité (GCP).

Supposons que le problème (GCP) admet une solution réalisable $\bar{w} = (\bar{x}, \bar{y}, \bar{z})$ et que les contraintes actives en \bar{w} satisfont à l'hypothèse d'indépendance linéaire; le résultat ci-après, similaire au théorème 4.2 énonce des conditions nécessaires d'optimalité pour le problème (QP).

Proposition 4.5: Soit $\bar{w} = (\bar{x}, \bar{y}, \bar{z}) \in \mathfrak{F}^{GCP}$ tel que la relation de complémentarité $0 \leq \bar{y} \perp \bar{z} \geq 0$ est non dégénérée ($\bar{y}_i + \bar{z}_i \neq 0, i = 1, \dots, n_y$). \bar{w} est un point stationnaire pour le problème (QP) s'il existe des multiplicateurs de KKT $(\lambda, \mu, \eta) \in \mathbb{R}^{m_p + 2n_y}$ tels que:

$$\begin{pmatrix} 0 \\ \bar{z} \\ \bar{y} \end{pmatrix} + \begin{pmatrix} A^t \\ 0 \\ 0 \end{pmatrix} \lambda + \begin{pmatrix} \nabla_x H(\bar{x}, \bar{y}) \\ \nabla_y H(\bar{x}, \bar{y}) \\ -I_{n_y} \end{pmatrix} \mu + \begin{pmatrix} 0 \\ \bar{Y} \\ \bar{Z} \end{pmatrix} \eta = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (4.4.8)$$

$$\lambda \geq 0, \lambda^t (A\bar{x} - b) = 0,$$

où \bar{Y} et \bar{Z} sont des matrices diagonales dont les coefficients diagonaux respectifs sont \bar{y}_i et $\bar{z}_i, i = 1, \dots, n_y$.

Nous proposons de résoudre le programme (QP) par un algorithme de type SLP, en procédant à une approximation de la relation de complémentarité $0 \leq y \perp z \geq 0$ par une la fonction de mérite paramétrique de Fisher Burmeister. On obtient donc le système des contraintes suivant:

$$\begin{cases} Ax \leq b, \\ H(x, y) - z = 0, \\ \Psi(y, z, \mu) = 0, \\ x, y, z \geq 0, \end{cases} \quad (4.4.9)$$

où $\Psi : \mathbb{R}_+^{n_y} \times \mathbb{R}_+^{n_y} \times \mathbb{R}_+ \rightarrow \mathbb{R}^{n_y}$ dont chacune des composantes est constituée de l'approximation de la relation de complémentarité $z_i y_i = 0$ par la fonction de mérite paramétrique de Fisher Burmeister, à savoir $\Phi(y_i, z_i, \mu) = 0$. Soit (w^k, μ_k) une solution réalisable du système (4.4.9), avec $w^k = (x^k, y^k, z^k)$. Nous utilisons les notations suivantes:

$H^k = H(x^k, y^k)$ et $\nabla_y \Psi(y^k, z^k, \mu_k) \equiv D_y^k$ et $\nabla_z \Psi(y^k, z^k, \mu_k) \equiv D_z^k$ sont les matrices diagonales dont les éléments diagonaux sont respectivement

$$1 - \frac{z_i^k}{\sqrt{(z_i^k)^2 + (y_i^k)^2 + 2\mu_k}} \text{ et } 1 - \frac{y_i^k}{\sqrt{(z_i^k)^2 + (y_i^k)^2 + 2\mu_k}}, \quad i = 1, \dots, n_y.$$

Tout comme pour la résolution d'un problème de MPEC, une direction de descente pour le problème (QP) est calculée à chaque itération k à l'aide de l'algorithme SLP en résolvant le programme linéaire ci-après:

$$\begin{aligned}
 & PL(w^k, \mu_k) \\
 & \text{Maximiser}_{dx, dy, dz, \xi} \xi \\
 & \text{s.c.} \begin{cases} (z^k)^t dy + (y^k)^t dz + \xi \leq 0, \\ Adx \leq b - Ax^k, \\ \begin{bmatrix} \nabla_x H^k & \nabla_y H^k & -I_{n_y} \\ 0 & D_y^k & D_z^k \end{bmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{bmatrix} -H^k + z^k \\ -\Psi(z^k, y^k, \mu_k) \end{bmatrix}, \\ (x^k + dx, y^k + dy, z^k + dz) \in X \times Y \times \mathbb{R}_+^{n_y}. \end{cases} \quad (4.4.10)
 \end{aligned}$$

Supposons que la fonction $y \rightarrow H(x, y)$ est une P_0 -fonction (la matrice $\nabla_y H^k$ est donc une P_0 -matrice). À travers la proposition ci-dessous qui se démontre de la même façon que la proposition 4.4, nous établissons que sous l'hypothèse 4.5, chaque programme linéaire $PL(w^k, \mu_k)$ admet toujours une solution réalisable et cette solution est unique.

Proposition 4.6: Étant donnée le problème de MPEC (QP), considérons $w^k = (x^k, y^k, z^k) \in \mathfrak{F}_r^{GCP}$. Si la fonction $y \rightarrow H(x, y)$ est une P_0 -fonction (donc monotone), alors le programme linéaire $PL(w^k, \mu_k)$ admet une solution optimale unique.

Soit (ξ^*, dx, dy, dz) une solution optimale du programme linéaire $PL(z^k, \mu_k)$; ses multiplicateurs du simplexe $(\theta_w, \lambda, \mu, \eta) \in \mathbb{R}_+^{n_x+2n_y} \times \mathbb{R}_+^m \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}$ sont obtenus à partir du programme linéaire dual de $PL(w^k, \mu_k)$ dont la formulation est donnée par:

$$\begin{aligned}
 & DPL(w^k, \mu_k) \\
 & \text{Minimiser}_{\alpha, \lambda, \mu, \eta, \theta_w} \lambda^t (b - Ax^k) - \mu^t (H^k - z^k) - \eta^t \Psi(z^k, y^k, \mu_k) + \theta_w^t w^k \\
 & \text{s.c.} \left\{ \begin{array}{l} \alpha \begin{pmatrix} 0 \\ z^k \\ y^k \end{pmatrix} + \begin{pmatrix} A^t \\ 0 \\ 0 \end{pmatrix} \lambda + \begin{pmatrix} \nabla_x H^k \\ \nabla_y H^k \\ -I_{n_y} \end{pmatrix} \mu + \begin{pmatrix} 0 \\ D_y^k \\ D_z^k \end{pmatrix} \eta = \begin{pmatrix} I_{n_x} \\ I_{n_y} \\ I_{n_y} \end{pmatrix} \theta_w, \\ \alpha \geq 1, (\theta_w, \lambda, \mu, \eta) \in \mathbb{R}_+^{n_x+2n_y} \times \mathbb{R}_+^m \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}. \end{array} \right. \quad (4.4.11)
 \end{aligned}$$

Sans nuire à la généralité, on peut supposer que $\alpha = 1$.

Remarque 4.2: En s'inspirant des développements antérieurs, notamment le corollaire 4.1 et le théorème 4.2, on peut observer que:

- 1) Si $\xi^* > 0$, on a $\sum_{j=1}^{n_y} (y_j^k dz_j + z_j^k dy_j) \leq 0$ et nécessairement $y_j^k dz_j + z_j^k dy_j \leq 0$ pour tout j .

D'après la proposition 4.3, (dx, dy, dz) est une direction de descente.

- 2) Si $\xi^* = 0$ et $\lim_{k \rightarrow \infty} \mu_k = 0$, d'après le théorème 4.2, (x^k, y^k) est un point stationnaire du problème (QP). $(\theta_w, \lambda, \mu, \eta) \in \mathbb{R}_+^{n_x+2n_y} \times \mathbb{R}_+^m \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}$ est le vecteur des multiplicateurs de KKT du problème (QP). On a:

$$\lambda^t (b - Ax^k) = 0; \mu^t (H^k - z^k) = 0; \theta_w^t w^k = 0 \text{ et } \eta^t \Psi(z^k, y^k, \mu_k) = 0.$$

Nécessairement, on a $f(x^k, y^k) = 0$.

L'approche de résolution du problème (CGP) que nous proposons pourrait être adaptée à la détermination d'une solution admissible d'un problème de PDNL.

4.5. Les expérimentations numériques

Les expérimentations numériques que nous avons faites sont inspirées des exemples de problèmes de MPEC tirés de Fukushima et al. [1998].

4.5.1 Les problèmes traités (Fukushima et al. [1998])

Nous avons testé l'algorithme *SLP* sur les problèmes de MPEC à contraintes linéaires et sur les problèmes de complémentarité généralisé de type (4.4.6). Pour tous les problèmes résolus, la matrice M est une P_0 -matrice.

Les données des problèmes de grande taille sont générées aléatoirement (générateur de l'interface MATLAB) et satisfont quelques contraintes techniques afin d'assurer l'existence des solutions; ainsi:

- La matrice A et le vecteur b sont tels que la solution $x^0 = (1, \dots, 1)^t \in \mathbb{R}^{n_x}$ vérifie $Ax^0 \leq b$.
- La matrice M est à diagonale positive dominante et tous les autres éléments de M valent 0 ou 1.
- les matrices A , M et N ont une proportion de 30% de coefficients non nuls. Pour les matrices A et N , les coefficients non nuls sont des entiers compris entre -50 et 50 alors que les éléments diagonaux de la matrice M , les coefficients des vecteurs b et q sont générés aléatoirement dans l'intervalle $[0, 50]$.

Les problèmes sur lesquels ont portés les tests sont donc les suivants:

Problème 1: Ce problème de MPEC comporte une seule variable de premier niveau et une seule variable de second niveau.

$$\begin{aligned} & \text{Minimiser } \frac{1}{2}x^2 - \frac{1}{2}xy - 95 \\ & \text{s.c. } \begin{cases} 0 \leq x \leq 100, \\ 0 \leq y \perp (2y + \frac{1}{2}x - 100) \geq 0. \end{cases} \end{aligned}$$

Problème 2: Les données relatives à ce problème de MPEC qui comporte une inégalité variationnelle comme contrainte de second niveau sont les suivantes:

$$\begin{aligned} X &= [0,10] \times [0,10], \quad f(x,y) = \frac{1}{2}[(x_1 + x_2 + y_1 - 15)^2 + (x_1 + x_2 + y_2 - 15)^2] \\ q &= \begin{pmatrix} 36 \\ 25 \end{pmatrix}, \quad N = \begin{pmatrix} 8/3 & 2 \\ 2 & 5/4 \end{pmatrix}, \quad M = \begin{pmatrix} 2 & 8/3 \\ 5/4 & 2 \end{pmatrix}. \end{aligned}$$

Problème 3: Nous avons ici un ensemble de problèmes de MPEC dont les données sont générées aléatoirement selon la méthode décrite ci-dessus. La taille des problèmes est caractérisée par un triplet (n_x, n_y, m_p) avec $n = n_x + n_y$. La fonction économique est définie par:

$f(x,y) = \frac{1}{2}x^t x + 1^t y$ où $1^t = (1, \dots, 1)^t \in \mathbb{R}^{n_y}$. L'optimum global de ce problème est connu et vaut zéro, conformément au choix des données.

La solution de départ est déterminée en fonction du type de problème résolu. Partant d'une formulation des contraintes de type (4.4.1). La solution de départ pour un problème de MPEC est:

$$\text{i) } x^0 = (0, \dots, 0)^t \in \mathbb{R}^{n_x}, \quad y^0 = (1, \dots, 1)^t \in \mathbb{R}^{n_y};$$

$$\text{ii) } x^0 = (1, \dots, 1)^t \in \mathbb{R}^{n_x}, \quad y^0 = (0, \dots, 0)^t \in \mathbb{R}^{n_y};$$

$$\text{iii) } x^0 = (1, \dots, 1)^t \in \mathbb{R}^{n_x}, \quad y^0 = (1, \dots, 1)^t \in \mathbb{R}^{n_y} \quad \text{et } z^0 = \begin{pmatrix} b \\ q \end{pmatrix} - \begin{pmatrix} A & 0 \\ N & M \end{pmatrix} \begin{pmatrix} x^0 \\ y^0 \end{pmatrix},$$

alors que la solution de départ pour le problème de complémentarité généralisé est:

$$x^0 = (0, \dots, 0)^t \text{ (ou } x^0 = (1, \dots, 1)^t), x^0 \in \mathbb{R}^{n_x}, y^0 = (1, \dots, 1)^t \in \mathbb{R}^{n_y} \text{ et } z^0 = \begin{pmatrix} b \\ q \end{pmatrix} - \begin{pmatrix} A & 0 \\ N & M \end{pmatrix} \begin{pmatrix} x^0 \\ y^0 \end{pmatrix}.$$

4.5.2 Les résultats des expérimentations numériques

Les algorithmes *SLP* et *SQP* de Fukushima et al. [1998] ont été implémentés dans une interface MATLAB, puis exécutés sur un micro ordinateur PC Pentium 4, CPU 3.20 GHz et 512 MB de RAM.

a. Mise en œuvre de l'algorithme *SLP*

Les résultats numériques sont résumés dans les tableaux ci-dessous; le tableau 4.1 représente les résultats relatifs à la résolution des problèmes de MPEC par l'algorithme *SLP*. Le processus itératif s'arrête lorsqu'on trouve une valeur de $\xi^* + \mu_k \leq \varepsilon$, avec $\varepsilon = 10^{-10}$. Implicitement, on a $\xi^* \leq \varepsilon$ et $\mu_k \leq \varepsilon$. Ce qui signifie que l'algorithme *SLP* a trouvé la solution $\bar{z} = (\bar{x}, \bar{y}, \bar{w}) = \lim_{k \rightarrow \infty} (z^k = (x^k, y^k, w^k))$ qui est la meilleure approximation d'un point stationnaire pour le problème $MPEC(M, N, q)$.

Pour le problème 3, les données sont générées aléatoirement: pour (n_x, n_y, m_p) fixé, nous avons testé l'algorithme *SLP* sur 10 séries de problèmes. Le nombre moyen d'itérations est calculé à partir de chacune de ces 10 séries de problèmes; il en est de même pour le temps d'exécution du programme représenté par CPU. La solution de départ est donnée par le vecteur (x^0, y^0) : $(x^0, y^0) = (0, 1)$ signifie que toutes les composantes de x^0 valent 0, et celles de y^0 valent 1. Il en est de même pour $(x^0, y^0) = (1, 1)$ ou $(x^0, y^0) = (1, 0)$.

Partant de $(x^0, y^0) = (1, 0)$ avec les séries du problème 3 comme Fukushima et al. [1998], l'algorithme *SLP* a donné la même solution optimale que la méthode de Fukushima et al., avec un nombre d'itérations moins élevé que cette méthode. Cependant, pour les séries relatives au problème 3, partant de $(x^0, y^0) = (1, 1)$ ou de $(x^0, y^0) = (0, 1)$, l'algorithme a certes conduit à $\mu^* \leq \varepsilon$, mais avec $\xi^* \neq 0$ huit fois sur dix. La solution ainsi trouvée est une solution réalisable du problème $MPEC(M, N, q)$ et non un point stationnaire. Les figures 4.1 et 4.2 représentent les

courbes de l'évolution de μ_k (courbe des points (*)), et de $\|\Phi(y^k, z^k, \mu_k)\|_2$ (courbe des points (+)) en fonction de k . On peut observer sur ces courbes que les fonctions $(z^k, y^k) \rightarrow \|\Phi(z^k, y^k, \mu_k)\|_2$ et $(z^k, y^k) \rightarrow \mu_k = \max_{i \in I} (z_i^k y_i^k)$ varient de manière identique; ceci confirme de manière expérimentale la justesse du choix de l'actualisation du paramètre μ_k que nous avons proposé.

Tableau 4. 1: Tests numériques - algorithme SLP (problèmes de MPEC)

Nom du problème	Taille: (n, n_x, n_y, m_p)	Nombre moyen d'itérations	(x^0, y^0)	CPU
Problème 1	(4,2,2,2)	6	(0,1)	0.90
Problème 1	(4,2,2,2)	6	(1,1)	0.50
Problème 2	(2,1,1,1)	10	(0,1)	1.13
Problème 2	(2,1,1,1)	10	(1,1)	1.13
Problème 3	(50,25,25,8)	10	(0,1)	1.25
Problème 3	(50,25,25,8)	9	(1,1)	1.14
Problème 3	(80,40,40,13)	8	(1,1)	1.75
Problème 3	(80,40,40,13)	12	(0,1)	2.05
Problème 3	(100,50,50,17)	10	(0,1)	2.20
Problème 3*	(100,50,50,17)	13	(1,1)	2.37
Problème 3**	(150,75,75,25)	14	(0,1)	5.73
Problème 3	(150,75,75,25)	12	(1,1)	4.87

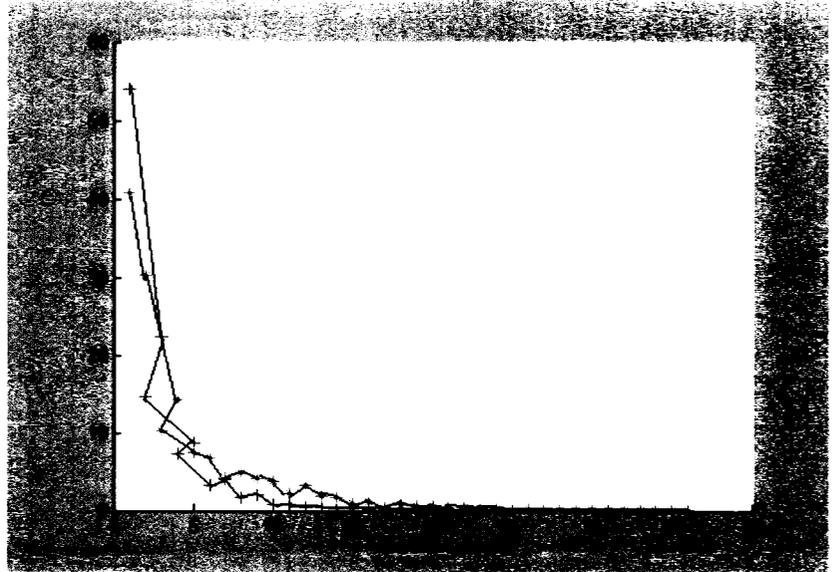


Figure 4. 1: Évolutions de la valeur de μ_k (*) et de $\|\Phi_k\|_2$ (+) en fonction de k (Problème 3*)

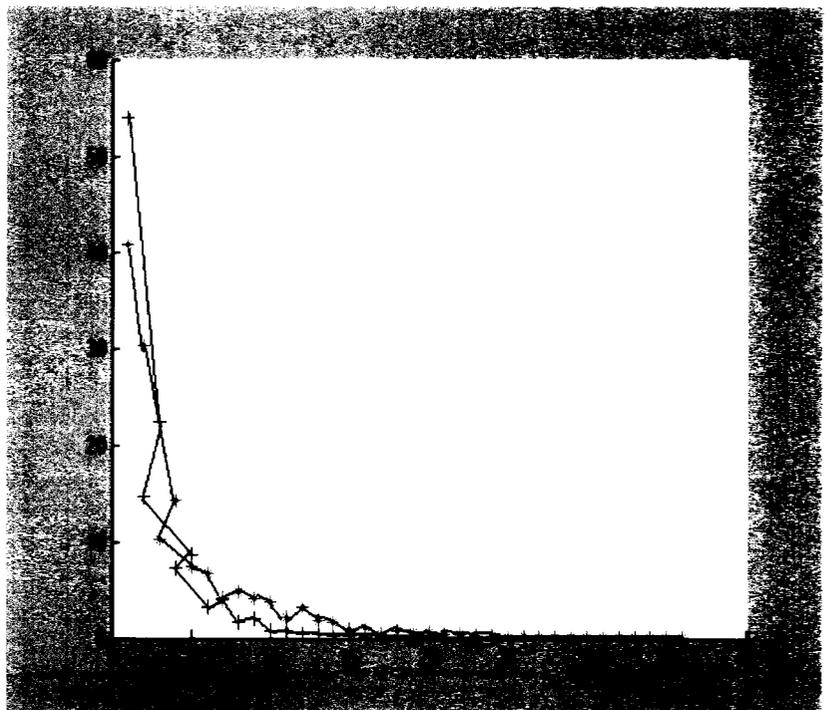


Figure 4. 2: Évolutions de la valeur de μ_k (*) et de $\|\Phi_k\|_2$ (+) en fonction de k (Problème 3**)

L'algorithme *SLP* s'est avéré particulièrement efficace pour résoudre le problème de complémentarité linéaire; les résultats des expérimentations numériques figurent dans le tableau 4.2 ci-dessous. Les données relatives à chacun des problèmes traités de taille (n, n_x, n_y, m_p) sont identiques à celles du problème 3 ci-dessus. La taille (n, n_x, n_y, m_p) du problème résolu étant fixée et les données générées de manière aléatoire, nous avons comme précédemment résolu des séries de 10 problèmes. On a trouvé systématiquement une valeur de la fonction économique du problème (*QP*) voisine de zéro au même titre que les valeurs μ^* du paramètre μ et ξ^* du programme linéaire ($PL(w^k, \mu_k)$).

Tableau 4. 2: Tests numériques - algorithme *SLP* (problèmes de complémentarité)

Taille: (n, n_x, n_y, m_p)	Nombre moyen d'itérations	(x^0, y^0)
(50,25,25,8)	7	(0,1)
(50,25,25,8)	4	(1,1)
(80,40,40,13)	8	(1,1)
(80,40,40,13)	5	(0,1)
(100,50,50,17)	9	(1,1)
(100,50,50,17)	4	(0,1)
(150,75,75,25)	10	(1,1)
(150,75,75,25)	6	(0,1)

b. Mise en œuvre de l'algorithme *SQP* de Fukushima et al. [1998]

Le tableau 4.3 représente les résultats liés à la mise en œuvre de l'algorithme de Fukushima et al.. Nous avons écrit le code MATLAB relatif à la version initiale de l'algorithme *SQP* proposé par Fukushima et al.. Nous avons constaté que, quelque soit la taille des données utilisées pour résoudre chacune des séries relatives au problème 3, le nombre minimal d'itérations dépend de la valeur de ε utilisée comme valeur de la précision dans le critère d'arrêt de l'algorithme. Partant de la solution initiale $(x^0, y^0) = (0,1)$, l'algorithme Fukushima et al. n'a pas

été en mesure de résoudre le problème 3 neuf fois sur dix, l'interface MATLAB n'a pas pu trouver de solution de départ à la résolution du programme quadratique qui calcule une direction de déplacement. Partant de la solution de départ $(x^0, y^0) = (1, 1)$, l'algorithme de Fukushima et al. n'a pu déterminer qu'un point stationnaire pour chacune des séries relatives au problème 3 ($f^1(x^*, y^*) \neq 0$), tout comme l'algorithme SLP. Partant de la solution de départ $(x^0, y^0) = (1, 0)$, les algorithmes SLP et de Fukushima et al. ont pu déterminer systématiquement l'optimum global pour les séries du problème 3.

Pour les problèmes de petite taille 1 et 2, nous avons fixé $\mu_0 = 0.6$ et $\beta = 0.2$; ceci a permis d'accélérer le processus itératif. Cependant, l'algorithme de Fukushima et al. n'a pas pu résoudre ces problèmes, partant des solutions de départ $(x^0, y^0) = (1, 1)$ ou $(x^0, y^0) = (0, 1)$. Pour la série (100,50,50,17) (respectivement (150,75,75,25)) relative au problème 3, le nombre moyen d'itérations était de 40, et on a observé que la solution optimale était atteinte dès la 20^{ème} itération (respectivement 30^{ème} itération) avec certains tests de la série. Ceci confirme le caractère arbitraire de l'actualisation du paramètre μ utilisé dans la fonction de mérite.

Tableau 4. 3: Tests numériques - algorithme de Fukushima et al. (problèmes de MPEC)

Nom du problème	Taille: $n = n_x + n_y$ (n, n_x, n_y, m_p)	Nombre Moyen d'itérations	(x^0, y^0)	(μ_0, β)	CPU
Problème 1	(4,2,2,2)	17	(0,0)	(0.6,0.2)	0.25
Problème 2	(2,1,1,1)	23	(0,0)	(0.6,0.2)	0.26
Problème 1	(4,2,2,2)	17	(1,0)	(0.6,0.5)	0.17
Problème 2	(2,1,1,1)	23	(1,0)	(0.6,0.5)	0.21
Problème 3	(50,25,25,8)	35	(1,0)	(0.6,0.5)	3.11
Problème 3	(50,25,25,8)	31	(1,1)	(0.6,0.5)	3.51
Problème 3	(80,40,40,13)	37	(1,0)	(0.6,0.5)	9.46
Problème 3	(80,40,40,13)	31	(1,1)	(0.6,0.5)	8.93
Problème 3	(100,50,50,17)	38	(1,0)	(0.6,0.5)	14.17
Problème 3	(100,50,50,17)	41	(1,1)	(0.6,0.5)	24.73
Problème 3	(150,75,75,25)	35	(1,0)	(0.6,0.5)	39.40
Problème 3	(150,75,75,25)	40	(1,1)	(0.6,0.5)	80.70

Le comportement de l'algorithme proposé par Fukushima et al. [1998] dépend entre autres (i) du choix de la méthode d'actualisation du paramètre μ qui intervient dans la fonction de mérite que les auteurs utilisent dans leurs analyses, qui impose à l'algorithme d'exécuter un nombre minimal d'itérations (égal à $it_{min} = \ln(\frac{\varepsilon}{\mu_0}) / \ln(\beta)$) qui conduisent à $\mu^* \leq \varepsilon$, (ii) du choix de la matrice qui détermine la fonction économique du programme quadratique qui permet de calculer une direction de déplacement à chaque itération. À la lumière de nos expérimentations numériques, nous avons noté une efficacité relative de l'algorithme *SLP* par rapport à l'algorithme de Fukushima et al., qui faudrait-il le rappeler, donne des résultats similaires à l'algorithme PIPA de Luo et al. [1996a]. De plus l'algorithme *SLP* s'est avéré comme un excellent outil pour résoudre les problèmes de complémentarité généralisés linéaires.

En incorporant une méthode énumérative efficace d'exploitation des contraintes d'équilibre, l'algorithme *SLP* appliqué à la résolution des problèmes de MPEC peut conduire à un optimum global.

CHAPITRE 5 – RÉOLUTION DU PROBLÈME DE PDN PAR UN ALGORITHME DE PROGRAMMATION LINÉAIRE SÉQUENTIELLE PAR ÉNUMÉRATION IMPLICITE

Dans ce chapitre, nous présentons une contribution originale à la résolution des problèmes de programmation mathématique à deux niveaux à contraintes linéaires (PLDN) en procédant à une énumération partielle des faces de l'ensemble des solutions réalisables Ω du problème à résoudre, à partir de l'analyse de la monotonie et des conditions d'optimalité proposées par Hansen et al. [1992]. Nous introduisons pour la première fois dans l'analyse de la monotonie les concepts de graphes ou d'arborescences de monotonie à partir desquels nous dérivons des faces du domaine induit contenant une solution admissible. L'algorithme de programmation linéaire séquentielle (*SLP*) que nous avons décrit dans le chapitre 4 s'appuie sur la propriété de non dégénérescence d'un point stationnaire du problème de programmation mathématique avec contraintes d'équilibre (MPEC). Cet algorithme, appliqué à la résolution d'un MPEC de type (*PDN_{KKT}*) correspondant à la formulation KKT d'un problème de PLDN, permet de ne déterminer qu'une solution admissible de ce problème de PLDN, ce qui le rend théoriquement assez restrictif. Pour contourner cette lacune, nous proposons une autre version de l'algorithme *SLP*: c'est l'algorithme de programmation linéaire par pièces (*PSLP*) qui s'applique à la décomposition en branches du problème (*PDN_{KKT}*). Une exploitation judicieuse des contraintes d'équilibre de ce problème nous permet de transformer l'algorithme *PSLP* (Piecewise Sequential Linear Programming) en un algorithme de programmation linéaire séquentielle par énumération implicite: l'algorithme *ESLP* (Enumeration Sequential Linear Programming) est en fait l'algorithme *PSLP*, dans lequel on intègre à partir des principes de monotonie, une méthode d'énumération implicite portant sur des variables de second niveau. Cet algorithme génère au cours de chaque étape une direction de descente en exploitant de manière judicieuse les indices des contraintes et des variables du problème de second niveau. Le problème de second niveau étant mis sous forme standard, le traitement que l'algorithme *ESLP* consacre à certaines des variables de second niveau ou des variables d'écart a trait à deux états possibles de ces variables

à partir d'une solution admissible: chacune des variables en question est contrainte à prendre une valeur nulle. Il est question d'examiner si ces contraintes supplémentaires sont compatibles avec l'ensemble des contraintes initiales, et contribuent à améliorer la valeur de la solution courante.

L'organisation du chapitre est la suivante: dans la section 5.1, nous définissons le cadre théorique de l'algorithme *PSLP*; nous rappelons le principe de décomposition du problème (PDN_{KKT}) en branches, puis nous décrivons l'algorithme *PSLP* en illustrant son fonctionnement par un exemple de petite taille. Dans la section 5.2, nous introduisons l'algorithme *ESLP* avec, dans la sous-section 5.2.1, la présentation des fondements théoriques de cet algorithme. Nous proposons une condition suffisante d'optimalité pour un problème de programmation linéaire à deux niveaux (PLDN). Dans la sous-section 5.2.2, nous présentons le cadre conceptuel de l'algorithme *ESLP*. Nous faisons un bref rappel de l'énoncé des principes de monotonie, ainsi qu'une description du principe d'une méthode d'énumération implicite. Nous introduisons à partir des propriétés de monotonie du problème de second niveau, les concepts de graphes ou d'arborescences de monotonie qui permettent d'éliminer implicitement du problème de second niveau certaines variables. La sous-section 5.2.3 est consacrée à une présentation exhaustive de l'algorithme *ESLP* qui génère une suite de points contenus dans une face du domaine induit à chacune de ses étapes. Dans la sous-section 5.2.4, nous procédons à une description des expérimentations numériques liées à la mise en œuvre de l'algorithme *ESLP*.

5.1 Un algorithme de programmation linéaire par pièces (*PSLP*)

Nous rappelons pour commencer le principe de décomposition en branches de la formulation KKT d'un problème de PDN.

5.1.1 Décomposition en branches d'un problème de PDN sous forme KKT

Considérons dans un cas général, le problème (PDN_{KKT}) ci-dessous qui est en fait la formulation KKT d'un problème de programmation mathématique à deux niveaux (PDN):

(PDN_{KKT})

$$\begin{aligned} & \text{Minimiser}_{x,y} f^1(x,y) \\ & \text{s.c.} \begin{cases} (x,y) \in Z = X \times Y, \\ L(x,y,\lambda) = 0, \\ g^2(x,y) \leq 0, \lambda \geq 0, \\ \lambda^t g^2(x,y) = 0, \end{cases} \end{aligned} \quad (5.1.1)$$

où

$L(x,y,\lambda) = \nabla_y f^2(x,y) + \lambda^t \nabla_y g^2(x,y)$, $f^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $i=1,2$, $g^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^m$, $\lambda \in \mathbb{R}_+^m$ et $X \times Y = \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$. Soit \mathfrak{F}^{KKT} son ensemble des solutions réalisables et $\bar{w} = (\bar{z}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$ avec $\bar{z} = (\bar{x}, \bar{y})$. Nous faisons référence au paragraphe 3.4 du chapitre 3 afin de procéder à la décomposition des indices $I = \{1,2,\dots,m\}$ relatifs aux relations de complémentarité $\bar{\lambda}^t g^2(\bar{z}) = 0$.

À cet effet et à titre de rappel, nous considérons les sous-ensembles de I suivants:

$$I(\bar{z}) = \{i : g_i^2(\bar{z}) = 0\}, I_0(\bar{z}, \bar{\lambda}) = \{i \in I(\bar{z}) : \bar{\lambda}_i = 0\}, I_+(\bar{z}, \bar{\lambda}) = \{i \in I(\bar{z}) : \bar{\lambda}_i > 0\}.$$

Soient α et $\bar{\alpha}$ deux sous-ensembles complémentaires de $I_0(\bar{z}, \bar{\lambda})$ constitué du sous-ensembles des indices dits dégénérés. Les sous-ensembles $J \equiv \bar{\alpha} \cup \{i : g_i^2(\bar{z}) < 0\}$ et $J^c \equiv \alpha \cup I_+(\bar{z}, \bar{\lambda})$ forment une partition de I ; ils comportent à la fois des indices dégénérés et des indices i pour lesquels on a respectivement $\bar{\lambda}_i = 0$ et $\bar{\lambda}_i > 0$.

Pour $\bar{w} = (\bar{z}, \bar{\lambda}) \in \mathfrak{F}^{KKT}$ et $\bar{\alpha}$ donné, considérons l'ensemble constitué de familles de paires de partitions de I défini par

$$K(\bar{w}, \bar{\alpha}) = \{(J, J^c) : J \equiv \bar{\alpha} \cup \{i : g_i^2(\bar{z}) < 0\}, J^c = I \setminus J\}. \quad (5.1.2)$$

Soit (NLP_J) un programme branche (au sens de la définition 3.8 du chapitre 3) du problème (PDN_{KKT}) dont l'ensemble des solutions réalisables est noté \mathfrak{F}_J^{KKT} ; la définition 3.12 du

chapitre 3 relative à la B-stationnarité du problème de type (PDN_{KKT}) est définie à partir d'un ensemble de type $K(\bar{w}, \bar{\alpha})$. La formulation de (NLP_J) est la suivante :

$$\begin{aligned}
 & (NLP_J) \\
 & \text{Minimiser}_{x,y,\lambda} f^1(x,y) \\
 & \left. \begin{array}{l} (x,y) \in X \times Y, \\ L(x,y,\lambda) = 0, \\ g_i^2(x,y) \leq 0 \\ \lambda_i = 0 \end{array} \right\} i \in J, \\
 & \left. \begin{array}{l} g_i^2(x,y) = 0 \\ \lambda_i \geq 0 \end{array} \right\} i \in J^c.
 \end{aligned} \tag{5.1.3}$$

Partant d'une solution réalisable $w^k \in \mathfrak{F}^{KKT}$, on peut déterminer une partition $(J_k, J_k^c) \in K(w^k, \bar{\alpha}_k)$ à partir de laquelle on définit le programme branche (NLP_{J_k}) dont une solution peut être meilleure que la solution connue $w^k \in \mathfrak{F}^{KKT}$. L'exemple suivant illustre le calcul de l'ensemble $K(w^k, \bar{\alpha}_k)$.

Exemple 5.1: Considérons le problème de MPEC ci-après:

$$\begin{aligned}
 & \text{Minimiser}_{x,y,\lambda} f^1(x,y) = x - 2y \\
 & \left. \begin{array}{l} (x,y,\lambda) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+^3, \\ -2x + 2\lambda_1 x + \lambda_2 + 2x\lambda_3 = 0, \\ 2y + 3\lambda_2 + 2\lambda_3 = 0, \\ \lambda_1(x^2 - 1) = \lambda_2(x + 3y - 2) = \lambda_3(x^2 + 2y - 1) = 0, \\ g_1^2(x,y) = x^2 - 1 \leq 0, \\ g_2^2(x,y) = x + 3y - 2 \leq 0, \\ g_3^2(x,y) = x^2 + 2y - 1 \leq 0, \end{array} \right\} \begin{array}{l} (1) \\ (2) \\ (3) \end{array}
 \end{aligned}$$

dont une solution réalisable est $w^0 = (z^0, \lambda^0)$ avec $z^0 = (1,0)$ et $\lambda^0 = (1,0,0)$. On a pour $\alpha_0 = \{3\}$,

$$\bar{\alpha}_0 = \emptyset, l_0(\mathbf{w}^0) = \{3\}, l(z^0) = \{1,3\}, l_+(\mathbf{w}^0) = \{1\};$$

$$K(\mathbf{w}^0, \bar{\alpha}) = \{(\{2\}, \{1,3\}), (\{1,2\}, \{3\}), (\{2,3\}, \{1\}), (\{1,2,3\}, \emptyset)\}.$$

Le lagrangien du problème (NLP_j) est $L^{MPEC}(z, \lambda, \pi, \eta) = f^1(z) + \pi^t L(z, \lambda) + \eta^t g^2(z)$. La matrice hessienne relative au lagrangien $L^{MPEC}(z, \lambda, \pi, \eta)$ en $\mathbf{w} = (z, \lambda)$ est donnée par l'expression $H = \nabla_{\mathbf{w}\mathbf{w}}^2 L^{MPEC}(\mathbf{w}, \pi, \eta) = \nabla_{zz}^2 f^1(z) + \pi^t \nabla_{\mathbf{w}\mathbf{w}}^2 L(z, \lambda) + \sum_i \eta_i \nabla_{zz}^2 g_i^2(z)$.

L'utilisation d'un algorithme de programmation quadratique séquentielle pour résoudre le problème (PDN_{KKT}) suppose que la matrice H est définie positive et qu'il faut calculer le vecteur (π, η) des multiplicateurs de KKT à chaque itération, ce qui allongerait inexorablement le temps des calculs. Nous contournons cette exigence en proposant d'adapter l'algorithme SLP à la résolution de chacune des branches (NLP_j) du programme (PDN_{KKT}): c'est la méthode de programmation linéaire séquentielle par pièces ($PSLP$) que décrivons dans les détails dans la sous section suivante. Rappelons toutefois que les algorithmes de programmation quadratique séquentielle présentent l'avantage d'avoir une propriété de convergence quadratique, contrairement aux algorithmes de programmation linéaire séquentielle qui ont une convergence linéaire.

5.1.2 L'algorithme de programmation linéaire par pièces

Considérons un problème de PDN convexe dont la formulation KKT est donnée par (5.1.1). L'algorithme $PSLP$ consiste à utiliser un algorithme de programmation linéaire séquentielle pour résoudre à chaque itération un programme non linéaire de type (NLP_{j_k}): il faut déterminer une direction de descente à partir d'une solution réalisable $\mathbf{w}^k \in \mathfrak{F}_{j_k}^{KKT}$ avec $(J_k, J_k^C) \in K(\mathbf{w}^k, \bar{\alpha}_k)$, en résolvant le programme linéaire $PL(\mathbf{w}^k, J_k)$ ci-dessous qui est en fait la linéarisation de la branche (NLP_{j_k}) au voisinage de \mathbf{w}^k .

$PL(w^k, J_k)$

$$\begin{aligned}
 & \text{Maximiser}_{\xi, dz, d\lambda} \xi \\
 & \left\{ \begin{array}{l} \nabla_z f^1(z^k)^t dz + \xi \leq 0, \\ L(w^k) + \nabla L(w^k) dw = 0, \\ g_i^2(z^k) + \nabla g_i^2(z^k)^t dz + \xi \leq 0 \\ (\lambda^k + d\lambda)_i = 0 \end{array} \right\} \text{ pour } i \in J_k, \\
 & \left\{ \begin{array}{l} g_i^2(z^k) + \nabla g_i^2(z^k)^t dz \leq 0 \\ (\lambda^k + d\lambda)_i \geq 0 \end{array} \right\} \text{ pour } i \in J_k^c, \\
 & z^k + dz \in Z, \\
 & \xi \geq 0.
 \end{aligned} \tag{5.1.4}$$

Lorsque $w^k \in \mathfrak{F}^{KKT}$, si la valeur optimale de la fonction économique du programme linéaire $PL(w^k, J_k)$ est telle que $\xi^* = 0$ pour tout $(J_k, J_k^c) \in K(w^k, \bar{\alpha}_k)$, la solution courante $w^k = (z^k, \lambda_k)$ est un point stationnaire du problème (PDN_{KKT}) ; nous le démontrons à travers le corollaire ci-après.

Corollaire 5.1: Supposons que les fonctions f^1 et g sont de classe C^2 et que pour tout $(J_k, J_k^c) \in K(w^k, \bar{\alpha}_k)$, les contraintes actives en $\bar{w} = (\bar{z}, \bar{\lambda}) \in \mathfrak{F}_J^{KKT}$ sont linéairement indépendantes. Soit $(J_k, J_k^c) \in K(w^k, \bar{\alpha}_k)$ tel que le programme linéaire $PL(w^k, J_k)$ admet une solution. Si l'optimum du programme linéaire $PL(w^k, J_k)$ est tel que $\xi^* = 0$ pour tout $(J_k, J_k^c) \in K(w^k, \bar{\alpha}_k)$, alors la solution courante $w^* = w^k$ est un point stationnaire du problème (PDN_{KKT}) . Si $\xi^* > 0$, une solution optimale $(\xi^*, dz, d\lambda)$ du problème $PL(w^k, J_k)$ est telle que $dw = (dz, d\lambda)$ est une direction de descente pour la fonction f^1 .

Preuve: Soit $(J_k, J_k^C) \in K(w^k, \bar{\alpha}_k)$; il suffit de donner une formulation du programme linéaire dual de $PL(w^k, J_k)$ et de recourir à la démarche que nous avons utilisée pour démontrer le corollaire 4.1 du chapitre 4. Lorsque $\xi^* = 0$, on peut aisément montrer que les variables duales (π, η) du programme linéaire $PL(w^k, J_k)$ sont les multiplicateurs de KKT du programme non linéaire (NLP). On a:

$$\begin{aligned} \nabla_x f^1(z^k) + \nabla_x g^2(z^k)^t \eta &= -\nabla_x L(z^k, \lambda^k)^t \pi, \\ \nabla_y f^1(z^k) + \nabla_y g^2(z^k)^t \eta &= -\nabla_y L(z^k, \lambda^k)^t \pi, \\ \eta_i &= 0 \text{ si } i \in J_k, \\ \pi^t \nabla_y g_i^2(z^k) &= 0 \text{ si } i \in J_k^C. \end{aligned}$$

Par conséquent, les conditions d'optimalité du premier ordre sont satisfaites. D'après Luo et al. [1996b] (proposition 3.3.9), w^k est un point stationnaire pour le problème (PDN_{KKT}) et $(x^k, y^k) \in DI$.

Montrons que la direction de déplacement $dw = (dz, d\lambda)$ est une direction de descente pour la fonction économique f^1 lorsque $\xi^* > 0$. Soit $z^{k+1} = z^k + \tau dz$;

$$f^1(z^{k+1}) = f^1(z^k) + \tau \nabla f^1(z^k)^t dz + o(dz) \text{ avec } o(dz) \rightarrow 0 \text{ lorsque } dz \rightarrow 0.$$

Puisque $\nabla f^1(z^k)^t dz \leq -\xi^* < 0$, on a $\nabla f^1(z^k)^t dz \leq 0$, donc $f^1(z^{k+1}) \leq f^1(z^k)$. ■

Afin d'utiliser le principe de programmation linéaire séquentielle pour résoudre le problème (PDN_{KKT}) en faisant appel à sa décomposition en branches (NLP), nous pouvons remarquer que l'on a

$$\{i : g_i^2(\bar{z}) < 0 = \bar{\lambda}_i\} = \{i : g_i^2(\bar{z}) + \bar{\lambda}_i < 0\} \text{ et } \{i : \bar{\lambda}_i > 0 = g_i^2(\bar{z})\} = \{i : g_i^2(\bar{z}) + \bar{\lambda}_i > 0\}.$$

Considérons alors une solution $w^k \equiv (z^k, \lambda^k) \in \mathbb{R}^{n_x+n_y} \times \mathbb{R}_+^m$ relative à une itération quelconque de l'algorithme PSLP; w^k n'est pas forcément réalisable pour le problème (PDN_{KKT}) .

En pratique, il faudrait choisir arbitrairement une partition (J_k, J_k^C) de l'ensemble des indices I telle que $J_k \supset \{i : g_i^2(z^k) + \lambda_i^k < 0\}$ et $J_k^C \supset \{i : g_i^2(z^k) + \lambda_i^k > 0\}$ comme le suggèrent Luo et al. [1996b].

En relation avec la partition ci-dessus, il faudrait résoudre à chacune des étapes d'une itération k le programme linéaire $PL(w^k, J_k)$. L'algorithme *PSLP* se présente donc comme suit:

Algorithme *PSPL*: Résolution du problème (PDN_{KKT})

Étape 0: (Initialisation)

Poser $k = 0$. Soit $w^k = (x^k, y^k, \lambda^k) = (z^k, \lambda^k) \in \mathfrak{F}^{KKT}$; poser $O^k = K(w^k, \bar{\alpha}_k)$.

Étape 1: (Direction et pas de déplacement)

Choisir une paire d'ensemble d'indices $(J_k, J_k^C) \in O^k$.

Si le problème $PL(w^k, J_k)$ est non réalisable ou alors non borné, aller à l'étape 3.

Sinon, soit (ξ^*, dw) la solution optimale du programme linéaire $PL(w^k, J_k)$.

Si $dw = 0$, aller à l'étape 3.

Étape 2: (Actualisation)

Déterminer un pas de déplacement τ tel que pour $x = x^k + \tau dx$, le problème de second niveau admet comme solution optimale le vecteur $y^{k+1} = y^k + \tau dy$.

Si on a pu calculer le pas de déplacement τ , alors $w^{k+1} = w^k + \tau dw$ et

$O^{k+1} = K(w^{k+1}, \bar{\alpha}_{k+1})$. Poser $k = k + 1$ et aller à l'étape 1.

Sinon aller à l'étape 3.

Étape 3: (direction de déplacement nulle ou choix d'une autre paire d'ensemble d'indices)

Soit $w^{k+1} = w^k$, $O^{k+1} = O^k \setminus \{(J_k, J_k^C)\}$ et $k = k + 1$.

Étape 4: (Critère d'arrêt)

Si un critère d'arrêt est satisfait, stop.

Sinon, aller à l'étape 1.

À l'étape 3 de l'algorithme *PSLP*, on sélectionne un programme non linéaire d'indices (J_k, J_k^C) et on actualise w^k en w^{k+1} en appliquant une étape de l'algorithme *SLP* à ce sous-problème. L'idée de sélection de l'un des multiples sous-problèmes à chaque itération est inspirée de l'algorithme *PSQP* de Luo et al. [1998] qui font appel à une procédure similaire. De plus, tout comme la méthode *PSQP* de Luo et al., l'algorithme *PSLP* pourrait être considéré comme une autre version des méthodes disjonctives qui permettent de trouver un optimum du problème de PDN sous forme KKT, à l'instar de l'algorithme proposé par Hansen et al. [1992], à condition de disposer d'une procédure efficace qui permet de déterminer à chaque itération une branche (NLP_{j_k}) susceptible d'améliorer la fonction économique f^1 .

Afin de déterminer le critère d'arrêt évoqué à l'étape 4 de l'algorithme *PSLP*, nous pouvons noter que si toutes les branches (NLP_j) (ayant une solution) admettent w^k comme point stationnaire ($\xi^* = 0$ dans ce cas), alors w^k est un point B-stationnaire pour le problème (PDN_{KKT}) . Deux critères d'arrêts sont alors envisageables.

Critère d'arrêt A: $K(w^k, \bar{\alpha}) \neq \emptyset$;

Le critère d'arrêt A nécessite un examen exhaustif de la stationnarité de w^k pour chaque branche (NLP_j) (ayant une solution) où $(J_k, J_k^C) \in K(w^k, \bar{\alpha}_k)$. Ainsi, lorsque w^k est un minimum local, établir la stationnarité en w^k consiste à vérifier cette stationnarité en ce point pour $2^{|I_0(w^k)|}$ programmes non linéaires. Comme le font remarquer Luo et al. [1996b], ce critère d'arrêt peut être laborieux si le nombre $|I_0(w^k)|$ est grand.

Il faudrait faire appel à la relaxation du problème (PDN_{KKT}) pour définir un critère d'arrêt plus efficace. Ce critère d'arrêt s'appuie sur les conditions de KKT de la relaxation de type $(MPEC_{RKT})$ définie par le système (3.4.11) et les propriétés 3.7 et 3.8 du chapitre 3. Pour cela, considérons l'ensemble des indices $I_+(w^k)$. Le problème relaxé qui nous intéresse a pour formulation:

$$\begin{aligned}
 & (PDN_{RKT}) \\
 & \text{Minimiser}_{x,y,\lambda} f^1(x,y) \\
 & \text{s.c.} \begin{cases} (x,y,\lambda) \in Z \times \mathfrak{M}(x,y), \\ L(x,y,\lambda) = 0, \\ g_i^2(x,y) \leq 0 = \lambda_i, \quad \forall i \in I(z^k), \\ g_i^2(x,y) \leq 0 \leq \lambda_i, \quad \forall i \in I_0(w^k), \\ g_i^2(x,y) = 0 \leq \lambda_i, \quad \forall i \in I_+(w^k). \end{cases} \quad (5.1.5)
 \end{aligned}$$

Critère d'arrêt B: Soit $K(w^k, \bar{\alpha}_k) = \emptyset$ ou $\xi^* = 0$ et w^k est un point stationnaire du problème (PDN_{RKT}) .

Remarques 5.1:

1) Les multiplicateurs de KKT du problème (PDN_{KKT}) de chacune de ses branches ou du problème relaxé (PDN_{RKT}) sont obtenus à partir des variables duales des programmes linéaires $PL(w^k, J_k)$.

2) Lorsque $dw = (dz, d\lambda) = (0, 0)$, si pour chaque indice $i \in I_0(w^k)$, on a $\eta_i \geq 0$ et $\pi^t \nabla_y g_i^2(z^k) \leq 0$, alors $w^k = (z^k, \lambda^k)$ est un point B-stationnaire du problème (PDN_{KKT}) , et on arrête le processus itératif. Un critère d'arrêt similaire a été mis en œuvre avec succès par Jiang et Ralph [2000].

3) Si la partition (J_k, J_k^C) demeure inchangée tout au long des itérations, ce qui est le cas lorsque $K(w^k, \bar{\alpha}_k) = \emptyset$, alors il existe au moins une branche (NLP_J) ($J \subseteq I$) du problème (PDN_{KKT}) , telle que $\bar{w} \in \mathfrak{F}_J^{KKT}$ et l'algorithme $PSLP$ se réduit alors à l'algorithme SLP .

4) La contrainte $\nabla_z f^1(z^k)^t dz + \xi \leq 0$ est équivalente à la contrainte $f^1(z^{k+1}) + \xi \leq f^1(z^k)$; la contrainte $\nabla_z f^1(z^k)^t dz + \xi \leq 0$ est donc en quelque sorte une coupe.

Par conséquent, les algorithmes basés sur une méthode de programmation linéaire séquentielle que nous proposons ici s'apparentent aux méthodes des coupes.

La convergence de l'algorithme *PSLP* s'établit de la même manière que celle de l'algorithme *SLP* et est tributaire du corollaire 5.1 précédent.

Exemple 5.2 (illustration de l'algorithme *PSLP*)

Considérons le problème MPEC ci-après:

$$\begin{aligned} & \text{Minimiser}_{x,y} f(x,y) = \frac{1}{2}(x^2 + y^2) \\ & \text{s.c.} \begin{cases} g(x,y) = x - y - 2 \leq 0, y \geq 0, \\ x \geq 1, g(x,y)y = 0. \end{cases} \end{aligned}$$

Pour tout x , il est facile de voir que l'unique solution du problème d'équilibre relatif aux contraintes du MPEC ci-dessus est $y = \max\{x - 2, 0\}$. L'ensemble des solutions réalisables de ce problème est: $\mathfrak{F} = \{(x, 0) : 1 \leq x \leq 2\} \cup \{(x, x - 2) : x \geq 2\}$. On en déduit que son unique minimum (global) est: $(\bar{x}, \bar{y}) = (1, 0)$. Pour une itération k , soit $(J_k, J_k^C) \in \mathfrak{J}(z^k)$.

Itération 1: Prenons comme solution de départ: $z^0 = (x^0, y^0) = (4, 2)$; $\bar{\alpha} = \emptyset$ $f(z^0) = 10$, $g(z^0) = 0$. Donc $K(z^0, \bar{\alpha}_0) = \{(\{1\}, \emptyset), (\emptyset, \{1\})\}$. Au voisinage de z^0 , une branche linéarisée du problème ci-dessus est notée $PL(z^0, J_0)$ et on a:

$$\begin{aligned} & PL(z^0, \{1\}) \\ & \text{Maximiser}_{\xi, dx, dy} \xi \\ & \text{s.c.} \begin{cases} 4dx + 2dy + \xi \leq 0, \\ 4 + dx \geq 1, \\ dx - dy = 0, \\ 2 + dy \geq 0. \end{cases} \end{aligned}$$

En prenant en compte la contrainte $dx = dy$, la solution optimale de $PL(z^0, \{1\})$ est $(dx, dy) = (-2, -2)$; $\xi^* = 12$. $z^1 = z^0 + dz = (2, 0)$ est le minimum de f sur le domaine $\mathfrak{F}_{\{1\}}$.

Itération 2: On a $z^1 = (x^1, y^1) = (2, 0)$, $f(z^1) = 2$, $g(z^1) = 0 = y$ et $K(z^1, \bar{\alpha}_1) = K(z^0, \bar{\alpha}_0)$. Puisque z^1 est déjà le minimum de f sur $\mathfrak{F}_{\{1\}}$, on trouvera $dz = (0, 0)$ si on choisit $J_1 = \{1\}$. Supposons donc que $J_1 = \emptyset$. Le programme linéarisé à résoudre est:

$$\begin{array}{l}
 PL(z^1, \emptyset) \\
 \text{Maximiser}_{\xi, dx, dy} \xi \\
 \text{s.c.} \begin{cases} 2dx + \xi \leq 0, \\ 2 + dx \geq 1, \\ dx - dy \leq 0, \quad dy = 0. \end{cases}
 \end{array}$$

Mais $dy = 0$, la solution optimale de $PL(z^1, \emptyset)$ est $dz = (dx, dy) = (-1, 0)$; $\xi^* = 2$;

$z^2 = z^1 + dz = (1, 0)$ est le minimum de f sur le domaine \mathfrak{F}_{\emptyset} .

Itération 3: $z^2 = (x^2, y^2) = (1, 0)$; $f(z^2) = \frac{1}{2}$, $g(z^2) = -1$. $K(z^2, \bar{\alpha}_2) = K(z^1, \bar{\alpha}_1)$; la formulation de $PL(z^2, \emptyset)$ est:

$$\begin{array}{l}
 PL(z^2, \emptyset) \\
 \text{Maximiser}_{\xi, dx, dy} \xi \\
 \text{s.c.} \begin{cases} dx + \xi \leq 0, \\ dx - dy \leq 1, \\ dx \geq 0, \quad dy = 0. \end{cases}
 \end{array}$$

Sa solution optimale est $dz = (dx, dy) = (0, 0)$; $\xi^* = 0$, il en est de même pour $PL(z^2, \{1\})$. Ce qui prouve que z^2 est un point stationnaire de $\text{Min}\{f(z) : z \in \mathcal{F}_{J_2}\}$. On a montré que z^2 est un point stationnaire du problème de MPEC départ.

5.2 Résolution d'un problème de PLDN par un algorithme de programmation linéaire séquentielle par énumération implicite (ESLP)

L'algorithme de programmation linéaire séquentielle par énumération implicite que nous introduisons pour la première fois, permet de déterminer un optimum local pour un problème de programmation linéaire à deux niveaux (PLDN). Nous donnons dans la sous-section qui suit les fondements théoriques de cet algorithme.

5.2.1 Fondements théoriques de l'algorithme ESLP

Considérons la formulation particulière d'un problème programmation linéaire à deux niveaux (PLDN):

$$\begin{aligned}
 & (PLDN) \\
 & \text{Minimiser}_{x,y} f^1(x,y) \\
 & \text{s.c.} \left\{ \begin{array}{l} x \in P, \\ \text{Maximiser}_y d_2^t y \\ \text{s.c.} \left\{ \begin{array}{l} Ax + B_e y \leq b, \\ x, y \geq 0, \end{array} \right. \end{array} \right. \quad (5.2.1)
 \end{aligned}$$

où $(x,y) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$, $A \in \mathbb{R}^m \times \mathbb{R}^{n_x}$, $B_e \in \mathbb{R}^m \times \mathbb{R}^{n_y}$, $b \in \mathbb{R}^m$, $f^1 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ est linéaire et P un polyèdre de $\mathbb{R}_+^{n_x}$. Nous faisons l'hypothèse suivante afin de nous assurer que le problème (PLDN) est bien posé:

Hypothèse (H1): L'ensemble Ω des solutions réalisables du problème (PLDN) est non vide, fermé et borné. De plus, pour toute variable de premier niveau x fixée, toute solution optimale du problème de second niveau est primale duale non dégénérée.

La formulation KKT du problème (PLDN) est la suivante:

$$\begin{aligned}
 & \text{(LKKT)} \\
 & \text{Minimiser}_{x,y} f^1(x,y) \\
 & \text{s.c.} \begin{cases} Ax + B_e y \leq b, \\ \lambda^t B_e \geq d_2, \\ \lambda^t (Ax + B_e y - b) = 0, \\ (d_2 - \lambda^t B_e) y = 0, \\ x \in P, y, \lambda \geq 0. \end{cases} \quad (5.2.2)
 \end{aligned}$$

Nous notons \mathfrak{F}^{LKT} l'ensemble des solutions réalisables du problème (LKKT). Soit $w = (x, y, \lambda) \in \mathfrak{F}^{LKT}$, et considérons les ensembles $S(w)$ et $T(w)$ des partitions respectives des ensembles d'indices $S = \{1, 2, \dots, n_y\}$ et $T = \{1, 2, \dots, m\}$ définis par:

$$\begin{aligned}
 S(w) &= \{(I, I^c) : y_j = 0 \text{ pour } j \in I; y_j \geq 0 \text{ pour } j \in I^c\}, \\
 T(w) &= \{(J, J^c) : (Ax + B_e y)_i \leq b_i \text{ si } i \in J^c; (Ax + B_e y)_i = b_i \text{ si } i \in J\}.
 \end{aligned}$$

Pour $(I, I^c) \in S(w)$, on a $(d_2 - \lambda^t B_e)_j \leq 0$ si $j \in I$ et $(d_2 - \lambda^t B_e)_j = 0$ si $j \in I^c$. De même, pour $(J, J^c) \in T(w)$, on a $\lambda_i = 0$ si $i \in J$ et $\lambda_i \geq 0$ si $i \in J^c$. Ces partitions de S et T conduisent à une famille de programmes mathématiques à contraintes linéaires, qui sont des relaxations du problème LKKT, à savoir:

$$\text{(LP}_{(I,J)})$$

$$\begin{array}{l}
 \text{Minimiser}_{x,y,\lambda} f^1(x,y) \\
 \left. \begin{array}{l} (Ax + B_e y)_i \leq b_i \\ \lambda_i = 0 \end{array} \right\}, i \in J, \\
 \left. \begin{array}{l} (Ax + B_e y)_i = b_i \\ \lambda_i \geq 0 \end{array} \right\}, i \in J^c, \\
 \text{s.c.} \left. \begin{array}{l} (d_2 - \lambda^t B_e)_j \leq 0 \\ y_j = 0 \end{array} \right\}, j \in I, \\
 \left. \begin{array}{l} (d_2 - \lambda^t B_e)_j = 0 \\ y_j \geq 0 \end{array} \right\}, j \in I^c, \\
 x \in P.
 \end{array} \tag{5.2.3}$$

Soit $w^k \in \mathfrak{F}^{KKT}$ et $dw = (dx, dy, d\lambda)$ tels que $w^k + dw \in \mathfrak{F}^{KKT}$. Posons $b^k = b - Ax^k + B_e y^k$ et $d_2^k = d_2 - (\lambda^k)^t B_e$; on peut mettre les ensembles $S(w^k)$ et $T(w^k)$ sous la forme suivante:

$$\begin{aligned}
 S(w^k) &= \{(I, I^c) : y_j^k + dy_j = 0 \text{ pour } j \in I; y_j^k + dy_j \geq 0 \text{ pour } j \in I^c\} \text{ et} \\
 T(w^k) &= \{(J, J^c) : (Adx + B_e dy)_i \leq b_i^k \text{ si } i \in J^c; (Adx + B_e dy)_i = b_i^k \text{ si } i \in J\}.
 \end{aligned}$$

À partir de ces partitions de S et T , on définit une famille de programmes linéarisés:

$$DLP(w^k, I, J)$$

$$\begin{aligned}
& \text{Minimiser}_{dx, dy, d\lambda} \nabla f^1(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} \\
& \text{s.c.} \left\{ \begin{array}{l} (Adx + B_e dy)_i \leq b_i^k \\ \lambda_i^k + d\lambda_i = 0 \end{array} \right\}, i \in J, \\
& \left\{ \begin{array}{l} (Adx + B_e dy)_i = b_i^k \\ \lambda_i^k + d\lambda_i \geq 0 \end{array} \right\}, i \in J^c, \\
& \left\{ \begin{array}{l} (d_2^k - (d\lambda)^t B_e)_j \leq 0 \\ y_j^k + dy_j = 0 \end{array} \right\}, j \in I, \\
& \left\{ \begin{array}{l} (d_2^k - (d\lambda)^t B_e)_j = 0 \\ y_j^k + dy_j \geq 0 \end{array} \right\}, j \in I^c, \\
& x^k + dx \in P,
\end{aligned} \tag{5.2.4}$$

dont l'ensemble des solutions réalisables est noté $S_{i,J}(w^k)$.

Remarque 5.2: Les différentes partitions $S(w^k)$ et $T(w^k)$ relatives aux indices des variables et des contraintes de la formulation (LKKT) ne sont pas définies en fonction d'un ensemble de type $I_0(w^k)$ constitué de l'ensemble des indices dits dégénérés en w^k (qui ici est vide conformément à l'hypothèse H1), contrairement aux développements faits par des auteurs comme Luo et al. [1996a], [1998]. Ces partitions sont fonction de la valeur des variables ou de la nature des contraintes de second niveau du problème de PLDN résolu.

On peut noter que si on a $dw \in S_{i,J}(w^k)$, les contraintes

$$\left\{ \begin{array}{l} \lambda_i^k + d\lambda_i = 0 \text{ si } i \in J \\ \lambda_i^k + d\lambda_i \geq 0 \text{ si } i \in J^c \end{array} \right. \text{ et } \left\{ \begin{array}{l} (d_2^k - (d\lambda)^t B_e)_j = 0 \text{ si } dy_j + y_j^k \geq 0 \text{ (} j \in I^c \text{)} \\ (d_2^k - (d\lambda)^t B_e)_j \leq 0 \text{ si } dy_j + y_j^k = 0 \text{ (} j \in I \text{)} \end{array} \right. \tag{5.2.5}$$

sont satisfaites si $\lambda^k + d\lambda$ est le vecteur des multiplicateurs du simplexe pour le problème de second niveau relatif au problème (PLDN), lorsque la variable du premier niveau est fixée en prenant la valeur $x^k + dx$. La proposition suivante établit ce résultat.

Proposition 5.1: Soient w^k une solution réalisable du problème (LKKT) telle que $dw = (dx, dy, d\lambda) \in \mathcal{S}_{i,j}(w^k)$, $(I, I^C) \in S(w^k)$ et $(J, J^C) \in T(w^k)$. Posons $b^{k+1} = b - A(x^k + dx)$ et $d_2^k = d_2 - (\lambda^k)^t B_e$. Pour que les contraintes (5.2.5) ci-dessus soient satisfaites, il suffit que le vecteur $y^{k+1} = y^k + dy$ soit solution optimale du programme linéaire:

$$\begin{aligned}
 & PL_2(x^{k+1}) \\
 & \text{Maximiser } d_2^t y \\
 & \text{s.c. } \begin{cases} B_e y \leq b^{k+1}, \\ y \geq 0. \end{cases}
 \end{aligned}$$

Preuve: Il suffit de considérer le vecteur des multiplicateurs du simplexe du programme linéaire $PL_2(x^{k+1})$ que l'on peut mettre sous la forme $\lambda^{k+1} = \lambda^k + d\lambda$, comme $b^{k+1} = b - A(x^k + dx)$ et d'utiliser le théorème de la dualité en programmation linéaire pour établir le résultat. ■

Soit $w^k = (x^k, y^k, \lambda^k) \in \mathcal{F}^{LKT}$, $S(w^k)$ et $T(w^k)$ les partitions des ensembles d'indices S et T telles que $(I, I^C) \in S(w^k)$ et $(J, J^C) \in T(w^k)$. Le programme linéarisé que l'on devrait résoudre à chaque itération a pour formulation:

$$PL(w^k, I, J)$$

$$\begin{aligned}
 & \text{Maximiser}_{dx, dy, d\lambda, \xi} \xi \\
 & \text{s.c.} \left\{ \begin{array}{l} \nabla f^1(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} + \xi \leq 0, \\ (A dx + B_e dy)_i \leq b_i^k \\ \lambda_i^k + d\lambda_i = 0 \end{array} \right\}, i \in J, \\
 & \left\{ \begin{array}{l} (A dx + B_e dy)_i = b_i^k \\ \lambda_i^k + d\lambda_i \geq 0 \end{array} \right\}, i \in J^c, \\
 & \left\{ \begin{array}{l} (d_2^k - (d\lambda)^t B_e)_j \leq 0 \\ y_j^k + dy_j = 0 \end{array} \right\}, j \in I, \\
 & \left\{ \begin{array}{l} (d_2^k - (d\lambda)^t B_e)_j = 0 \\ y_j^k + dy_j \geq 0 \end{array} \right\}, j \in I^c, \\
 & x^k + dx \in P, \xi \geq 0.
 \end{aligned} \tag{5.2.6}$$

D'après la proposition ci-dessus, une condition nécessaire pour que le programme linéaire $PL(w^k, I, J)$ ait une solution est que l'on ait:

a) (dx, dy) est une solution du programme linéaire:

$PRL(z^k, J)$

$$\begin{aligned}
 & \text{Maximiser}_{dx, dy, \xi} \xi \\
 & \text{s.c.} \left\{ \begin{array}{l} \nabla f^1(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} + \xi \leq 0, \\ (A dx + B_e dy)_i \leq b_i^k, \text{ si } i \in J, \\ (A dx + B_e dy)_i = b_i^k, \text{ si } i \in J^c, \\ \xi \geq 0, x^k + dx \in P, y^k + dy \geq 0, \end{array} \right.
 \end{aligned} \tag{5.2.7}$$

et

b) $(y^{k+1}, \lambda^{k+1}) = (y^k + dy, \lambda^k + d\lambda)$ est un couple de solutions optimales primale et duale du programme linéaire $PL_2(x^{k+1})$.

La proposition précédente donne une indication qui pourrait simplifier la résolution du problème (LKKT) par l'algorithme $PSLP$ que l'on applique à une décomposition du type $PL(w^k, I, J)$.

5.2.2 Cadre conceptuel de l'algorithme *ESLP*

Avec l'algorithme *PSLP* de la section précédente, on pourrait être amené à résoudre $2^{|I|}$ programmes linéaires du type $PRL(z^k, T)$ afin de s'assurer que l'on a déterminé la meilleure direction de descente (dx, dy) , ce qui n'est pas efficace en pratique. Il faudrait donc disposer d'une procédure qui, à partir des conditions nécessaires d'optimalité et de l'information contenue dans la solution admissible courante, exploite efficacement l'ensemble des indices définis par les relations de complémentarité, en affectant à chacune des variables impliquées dans ces relations une valeur, ou à chacune des contraintes un état susceptible d'améliorer la valeur de la fonction économique du premier niveau. Dans cette sous-section, nous nous proposons d'atteindre cet objectif en introduisant un algorithme de programmation linéaire séquentielle par énumération. Nous décrivons dans un premier temps les fondements de cet algorithme, puis nous faisons un bref rappel des principes de monotonie sur lesquels repose la procédure d'énumération des variables de second niveau du problème de programmation mathématique à deux niveaux linéaires (PLDN) à résoudre. Nous rappelons enfin le principe d'une méthode d'énumération implicite.

5.2.2.1 Prémisses et motivations

Considérons la formulation particulière d'un problème de PLDN définie par (5.2.1). La définition de la notion de direction de descente du domaine induit est nécessaire, puisque l'algorithme *ESLP* génère une suite de points contenus dans le domaine induit.

Définition 5.1: Considérons un problème de PDN sous sa forme générale et soit $z = (x^*, y^*) \in DI$; $d = (dx, dy) \in \mathbb{R}^{n_x + n_y}$ est appelée **direction du domaine induit** s'il existe $\alpha_0 > 0$ tel que l'on a:

$$(x^* + \alpha dx, y^* + \alpha dy) \in DI \text{ pour tout } \alpha \in [0, \alpha_0] \text{ et } \nabla f^1(x^*, y^*) \begin{pmatrix} dx \\ dy \end{pmatrix} < 0. \quad (5.2.8)$$

Dans l'algorithme *HJS* proposé par Hansen et al. [1992], la nature des contraintes (actives ou non actives) du problème à résoudre est déterminée selon un ensemble de variables booléennes α_i susceptibles de prendre les valeurs 0 ou 1. Nous reprenons implicitement cette démarche en introduisant les concepts de graphes, d'arborescences et de cheminements de monotonie; un cheminement de monotonie nous permet de déterminer la nature (active ou non active) de certaines contraintes à partir d'une solution admissible connue. Le problème à résoudre est en fait le problème (*PLDN*) dont la formulation est donnée par (5.2.1), le problème de second niveau étant mis sous forme standard; il a donc pour formulation:

$$\begin{aligned}
 & \text{(PLDNS)} \\
 & \text{Minimiser}_{x,y} f^1(x,y) \\
 & \text{s.c.} \left\{ \begin{array}{l} x \in P, \\ \text{Maximiser}_y d_2^t y \\ \text{s.c.} \left\{ \begin{array}{l} Ax + By = b, \\ x, y \geq 0, \end{array} \right. \end{array} \right. \quad (5.2.9)
 \end{aligned}$$

où $(x,y) \in \mathbb{R}_+^{n_x} \times \mathbb{R}_+^{n_y+m}$, $A \in \mathbb{R}^{m \times n_y}$, $B \in \mathbb{R}^{n_x \times (n_y+m)}$; la matrice B est de la forme $B = [B_e \ I_m]$ et $y = (y_1, y_2, \dots, y_{n_y}, y_{\bar{1}}, \dots, y_{\bar{m}})^t$, où $y_{\bar{1}}, \dots, y_{\bar{m}}$ sont les variables d'écart et I_m la matrice identité d'ordre m .

Remarque 5.3: Lorsqu'un problème de programmation linéaire à deux niveaux ne comporte pas de contraintes de premier niveau, les auteurs comme Bialas et Karwan [1982], Candler et Townsley [1982] ou encore Benson [1989] ont montré que le domaine induit s'écrit comme une union connexe de faces du polyèdre défini par l'ensemble des solutions réalisables Ω . Calvete et Gale [1998] ont établi un résultat similaire pour un problème de programmation quadratique quasi convexe à deux niveaux dont les contraintes du premier niveau ne comportent pas de variable de second niveau. Sous l'hypothèse (H1), la frontière du polyèdre Ω est constituée d'un

nombre fini de facettes F_t , $t \in H = \{1, 2, \dots, m + n_y\}$ et Ω comporte un nombre fini de faces non vides S_1, S_2, \dots, S_r . De plus, il existe un sous-ensemble de faces non vides de Ω telles que $Dl = \bigcup_{i \in J} S_i$ où $J \subset \{1, 2, \dots, r\}$ (Calvette et Gale, 1998: lemme 2.5). Les contraintes de second niveau $Ax + B_e y \leq b, y \geq 0$ peuvent se mettre sous la forme $A'x + B'y \leq b'$, avec

$$A' = \begin{bmatrix} A \\ O_{n_y \times n_x} \end{bmatrix}, B' = \begin{bmatrix} B_e \\ -I_{n_y} \end{bmatrix} \text{ et } b' = \begin{bmatrix} b \\ O_{n_y \times 1} \end{bmatrix}, \quad (5.2.10)$$

où $O_{k \times l} \in \mathbb{R}^{k \times l}$ est la matrice dont tous les coefficients sont nuls, alors que I_{n_y} est la matrice identité d'ordre n_y . La variable de premier niveau x étant fixée, on peut représenter une facette F_t ou une face S_T du polyèdre Ω par $F_t = \{y \in \mathbb{R}_+^{n_y} : B'_t y = (b' - A'x)_t\}$ et $S_T = \bigcap_{t \in T} F_t$ avec $T \subset H$.

Il faudrait rappeler qu'une solution admissible $z^k = (x^k, y^k)$ d'un problème de PLDN est atteinte sur un des sommets de l'ensemble des solutions réalisables Ω . À l'optimalité, si chacune des variables d'écart $y_{\bar{t}}, t \in T$ a une valeur nulle, alors les contraintes d'indices $t \in T$ du problème de second niveau sont actives et la face S_T définie par les contraintes en question contient la solution optimale. Une solution admissible (x^{k+1}, y^{k+1}) susceptible d'être calculée par l'algorithme *ESLP* est dite **solution améliorante** pour l'algorithme *ESLP* si on a $f^1(x^{k+1}, y^{k+1}) < f^1(x^k, y^k)$. À partir d'une solution admissible $z^k = (x^k, y^k)$, le principe de l'algorithme *ESLP* consiste à déterminer de manière énumérative une face $S_T \subset Dl$ qui contient solution améliorante pour l'algorithme *ESLP*.

Une direction de descente du domaine induit $d = (dx, dy)$ est calculée par une méthode de programmation linéaire séquentielle. Sakarovitch [1984] montre comment on peut caractériser les facettes (ou des familles de facettes) des polyèdres associés à certains problèmes d'optimisation combinatoire et comment on peut résoudre ces problèmes en utilisant ces facettes.

Nous montrerons plus loin que si l'algorithme *ESLP* ne peut pas déterminer une face $S_T \subset D_I$ contenant une solution améliorante, la solution actuelle (x^k, y^k) est une solution optimale locale (qui peut être globale) du problème (PLDN), puisque l'usage que nous faisons des principes de monotonie consiste en une énumération implicite des bases primales duales pour le problème de second niveau lorsque la variable du premier niveau est fixée.

La formulation KKT qui correspond au problème (PLDNS) est la suivante:

$$\begin{aligned}
 & \text{(LKTS)} \\
 & \text{Minimiser}_{x,y,\lambda} f^1(x,y) \\
 & \text{s.c.} \begin{cases} Ax + By = b, \\ \lambda^t B \geq d_2, \\ (\lambda^t B - d_2)y = 0, \\ x \in P, y \geq 0. \end{cases} \quad (5.2.11)
 \end{aligned}$$

Soit \mathfrak{F}^{LKS} l'ensemble des solutions réalisables du problème (LKTS), $w^k = (x^k, y^k, \lambda^k) \in \mathfrak{F}^{LKS}$ et $S(w^k)$ une partition de l'ensemble des indices $H = \{1, 2, \dots, n_y + m\}$ avec $(I, I^c) \in S(w^k)$; on a:

$$\begin{cases} (d_2^k - (d\lambda)^t B)_j = 0 \text{ si } dy_j + y_j^k \geq 0 \text{ pour } j \in I^c, \\ (d_2^k - (d\lambda)^t B)_j \leq 0 \text{ si } dy_j + y_j^k = 0 \text{ pour } j \in I. \end{cases}$$

Pour calculer une direction de descente, le programme linéarisé qui devrait être résolu à chaque itération de l'algorithme *ESLP* a pour formulation:

$$PLL(w^k, I)$$

$$\begin{array}{l}
\text{Maximiser}_{dx,dy,d\lambda,\xi} \xi \\
\left\{ \begin{array}{l}
\nabla_{(x,y)} f^1(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} + \xi \leq 0, \\
A dx + B dy = 0, \\
\left. \begin{array}{l}
y_j^k + dy_j = 0 \\
((d\lambda)^t B - d_2^k)_j \geq 0
\end{array} \right\} j \in I, \\
\left. \begin{array}{l}
y_j^k + dy_j \geq 0 \\
((d\lambda)^t B - d_2^k)_j = 0
\end{array} \right\} j \in I^c, \\
x^k + dx \in P, \xi \geq 0,
\end{array} \right.
\end{array} \quad (5.2.12)$$

où $d_2^k = d_2 - \lambda_k^t B$.

Pour $j \in I^c$, la relation $y_j^k + dy_j = 0$ signifie que si $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \mathfrak{F}^{LKS}$, alors $y_j^{k+1} = 0$ et $dy_j = -y_j^k$. D'après la proposition 5.1, pour trouver une solution optimale $dw = (dx, dy, d\lambda)$ du problème $PLL(w^k, I)$ telle que $w^{k+1} = w^k + dw \in \mathfrak{F}^{LKS}$, il suffit de résoudre le programme linéaire relaxé:

$PLR(z^k, I)$

$$\begin{array}{l}
\text{Maximiser}_{dx,dy,\xi} \xi \\
\left\{ \begin{array}{l}
\nabla_z f^1(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} + \xi \leq 0, \\
A dx + B dy = 0, \\
\left. \begin{array}{l}
y_j^k + dy_j = 0, j \in I, \\
y_j^k + dy_j \geq 0, j \in I^c, \\
x^k + dx \in P, \xi \geq 0,
\end{array} \right.
\end{array} \right.
\end{array} \quad (5.2.13)$$

qui comporte $n_x + n_y + m$ variables et $m+1$ contraintes avec $z^k = (x^k, y^k)$; une solution optimale (ξ^*, dx, dy) du programme $PLR(z^k, I)$ digne d'intérêt est telle que le programme linéaire

$$\begin{aligned}
 & PL_2(k) \\
 & \text{Maximiser}_y d_2^t y \\
 & \text{s.c.} \begin{cases} By = b - A(x^k + dx), \\ y \geq 0, \end{cases} \quad (5.2.14)
 \end{aligned}$$

a pour solution optimale $y^{k+1} = y^k + dy$, $\lambda^{k+1} = \lambda^k + d\lambda$ étant son vecteur des multiplicateurs du simplexe. Le problème $PL_2(k)$ comporte $n_y + m$ variables et m contraintes.

Nous commençons par établir une condition suffisante d'optimalité d'un problème de PDN du type (PLDNS). Le théorème ci-après énonce ce résultat.

Théorème 5.1: Étant donné un problème de programmation linéaire à deux niveaux de type (PLDNS) dont l'ensemble des solutions réalisables Ω est supposé non vide et borné, soit DI son domaine induit. Pour $k \geq 0$, considérons une solution $(x^k, y^k) \in DI$, l'ensemble des indices $H = \{1, 2, \dots, n_y\} \cup \{\bar{1}, \bar{2}, \dots, \bar{m}\}$ et la famille de programmes linéaires $PLR(z^k, I)$, $I \subset H$. Pour que (x^k, y^k) soit un optimum global du problème (PLDNS), il suffit que l'on ait pour tout $I \subset H$, l'une des deux situations suivantes:

- i) Le programme linéaire $PLR(z^k, I)$ n'a pas de solution.
- ii) Soit (ξ^*, dx, dy) est une solution optimale du programme linéaire $PLR(z^k, I)$. On a soit (a) $\xi^* = 0$, soit (b) (dx, dy) n'est pas une direction de descente du domaine induit, soit (c) $(x^{k+1}, y^{k+1}) \in DI$ et $f^1(x^{k+1}, y^{k+1}) \geq f^1(x^k, y^k)$ avec $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$.

Preuve: Soient $k \geq 0$, (x^k, y^k) une solution admissible du problème (PLDNS) et $I \subset H$. Une solution améliorante $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$ satisfait aux deux conditions $(x^{k+1}, y^{k+1}) \in DI$ et $f^1(x^{k+1}, y^{k+1}) < f^1(x^k, y^k)$, où (ξ^*, dx, dy) une solution optimale du programme linéaire $PLR(z^k, I)$ telle que $\xi^* > 0$. Si pour tout sous-ensemble d'indices

$I \subset H$, on a i) ou ii), nécessairement (x^k, y^k) est une solution optimale globale du problème (PLDNS) puisqu'il n'existe pas dans ce cas de solution améliorante. ■

Nous utilisons les principes de monotonie pour gérer au mieux l'exploration des indices de l'ensemble des partitions $S(w^k)$ à chaque itération k , afin de déterminer une direction de descente du domaine induit. Le problème (PLDNS) est obtenu à partir du problème (PLDN) défini par (5.2.1) en ajoutant des variables d'écart aux contraintes du problème de second niveau. Lorsque la variable de premier niveau x est fixée à \bar{x} , la formulation du problème de second niveau est:

$$\begin{aligned}
 & PL_2(\bar{x}) \\
 & \text{Maximiser}_y d_2^t y \\
 & \text{s.c.} \begin{cases} By = b - A\bar{x}, \\ y \geq 0. \end{cases} \quad (5.2.15)
 \end{aligned}$$

Considérons la solution admissible (\bar{x}, \bar{y}) et $T \subseteq \{1, 2, \dots, m\}$. Si chacune des variables d'écart $y_{\bar{i}}, i \in T$ a une valeur nulle, $(dy_{\bar{i}} = -\bar{y}_{\bar{i}}, i \in T)$, toutes les contraintes d'indices $i \in T$ sont serrées au point (\bar{x}, \bar{y}) et il existe une facette $F_{\bar{T}}$ de Ω (lorsque $|T| = 1$) ou une face $S_{\bar{T}}$ du domaine induit contenant (\bar{x}, \bar{y}) .

Une variable structurelle y_j peut également prendre une valeur nulle à l'optimalité. Détecter des variables structurelles ou des variables d'écart de second niveau qui prennent une valeur nulle à l'optimalité est un problème difficile. Nous l'abordons à l'aide des principes de monotonie. Ces principes ont d'abord été introduits par Wilde [1975], puis généralisés par Hansen et al. [1989]. L'algorithme HJS de Hansen et al. [1992] s'inspire de ces principes en procédant à une analyse de la monotonie, qui consiste à exploiter de manière combinée des principes de monotonie et d'élimination de variables de second niveau.

5.2.2.2 Les principes de monotonie

Les deux principes de monotonie que nous présentons s'appliquent au problème de second niveau, lorsque celui-ci est mis sous forme standard. Nous désignons par B_{ij} les coefficients de la matrice B sur la ligne d'indice i et la colonne d'indice j , et par d_2^j ceux de la fonction économique de second niveau d_2 du problème (*PLDM*). Le programme à résoudre est sous forme de maximisation.

Le premier principe de monotonie affirme que si un coefficient d_2^j de la fonction économique du second niveau est négatif, alors à l'optimalité, soit la variable y_j est nulle, soit au moins une contrainte i contenant un coefficient B_{ij} négatif est serrée, ce qui signifie que la variable d'écart $y_{\bar{i}}$ est nulle. Si le coefficient d_2^j est positif, alors nécessairement au moins une contrainte k contenant un coefficient B_{kj} positif est serrée, ce qui revient à dire que la variable d'écart $y_{\bar{k}}$ a une valeur nulle à l'optimalité. Le second principe s'applique dans le cas où l'application $M(x)$ est univoque. Il affirme que si le coefficient d_2^j est nul, alors soit la conclusion du premier principe pour le coefficient positif s'applique, soit celle avec un coefficient négatif s'applique.

Si le problème de second niveau a été mis sous forme standard après avoir ajouté une variable d'écart $y_{\bar{i}}$ à chaque contrainte d'indice i , nous donnons une formulation des principes de monotonie ci-dessus adaptée à ce cas particulier. Pour cela, considérons le problème de second niveau dont la formulation est la suivante:

$$\begin{aligned}
 & PL_2(\bar{x}) \\
 & \text{Maximiser}_y \sum_{j=1}^{n_y} d_2^j y_j \\
 & \text{s.c.} \left\{ \begin{array}{l} \sum_{j=1}^{n_y} B_{ij} y_j + y_{\bar{i}} = b_i - \sum_{k=1}^{n_x} A_{ik} \bar{x}_k, \\ y_j \geq 0, j = 1, 2, \dots, n_y; y_{\bar{i}} \geq 0, i = 1, 2, \dots, m. \end{array} \right. \quad (5.2.16)
 \end{aligned}$$

Pour chaque variable structurelle y_j , avec $j \in J_{n_y} = \{1, 2, \dots, n_y\}$, on définit les ensembles d'indices suivants:

$$I_j^+ = \{i \in \{1, 2, \dots, m\} : B_{ij} > 0\},$$

$$I_j^- = \{i \in \{1, 2, \dots, m\} : B_{ij} < 0\}.$$

Les propriétés ci-dessous-précisent la formulation des deux principes de monotonie lorsque le problème de second niveau est sous la forme standard.

Propriété 5.1 (premier principe de monotonie): Soit (x^*, y^*) une solution optimale du problème de PLDN. Alors, pour tout $j \in J_{n_y}$ tel que $d_2^j < 0$, il existe au moins un indice $i \in I_j^- \cup \{j\}$ tel que l'on a: a) $y_{\bar{i}}^* = 0$ ($dy_{\bar{i}} = -y_{\bar{i}}^*$) si $i \in I_j^-$, b) $y_j^* = 0$ ($dy_j = -y_j^*$) si $i = j$; pour tout $j \in J_{n_y}$ tel que $d_2^j > 0$, il existe au moins un indice $i \in I_j^+$ tel que $y_{\bar{i}}^* = 0$ ($dy_{\bar{i}} = -y_{\bar{i}}^*$).

Cette propriété est une conséquence du corollaire 4.2 établi par Hansen et al. [1992]. Illustrons ce premier principe de monotonie avec l'exemple suivant.

Exemple 5.4: Considérons le problème de PLDN ci-après

$$\begin{array}{l} \text{Maximiser}_{x,y \geq 0} \quad 8x_1 + 4x_2 - 4y_1 + 40y_2 - 4y_3 \\ \left\{ \begin{array}{l} x_1 + x_2 - y_3 + y_{\bar{1}} = 1.3, \\ \text{Minimiser}_y \quad 2y_1 + y_2 + y_3 \\ \text{s.c.} \left\{ \begin{array}{l} -y_1 + y_2 + y_3 + y_{\bar{2}} = 1, \quad (\alpha_1) \\ 4x_1 - 2y_1 + 4y_2 - y_3 + y_{\bar{3}} = 2, \quad (\alpha_2) \\ 4x_2 + 4y_1 - 2y_2 - y_3 + y_{\bar{4}} = 2, \quad (\alpha_3) \\ -y_1 \leq 0, \quad (\alpha_4) \\ -y_2 \leq 0, \quad (\alpha_5) \\ -y_3 \leq 0, \quad (\alpha_6) \end{array} \right. \end{array} \right. \end{array}$$

À l'optimalité du PLDN, le premier principe de monotonie affirme que:

- Par rapport à $j = 1$, il existe au moins un indice $t \in \{1, \bar{2}, \bar{3}\}$ tel que $y_t^* = 0$.
- Par rapport à $j = 2$, il existe au moins un indice $t \in \{2, \bar{4}\}$ tel que $y_t^* = 0$.
- Par rapport à $j = 3$, il existe au moins un indice $t \in \{3, \bar{3}, \bar{4}\}$ tel que $y_t^* = 0$.

Le deuxième principe de monotonie s'applique au cas où la fonction économique est indépendante d'une variable. Il n'est valable que sous l'hypothèse d'univocité de l'application $M(x)$ définie par l'ensemble des solutions optimales du problème de second niveau, c'est-à-dire lorsque toute solution optimale du problème de second niveau est primale duale non dégénérée.

Propriété 5.2 (Deuxième principe de monotonie): Supposons que l'application $M(x)$ est univoque conformément à l'hypothèse (H1). Soit (x^*, y^*) une solution optimale du PLDN. Alors pour tout $j \in J_{n_j}$ tel que $d_2^j = 0$, il existe au moins un indice $i \in I_j^+ \cup I_j^-$ tel que $y_i^* = 0$.

Cette propriété n'est autre qu'une reformulation du théorème 3.2 que l'on trouve dans Savard [1989]. L'application de ce second principe de monotonie sur l'exemple 5.4 précédent entraîne qu'à l'optimalité, au moins une variable d'écart $y_i^* = 0$, $1 \leq i \leq 4$.

Nous faisons à présent appel aux conditions nécessaires d'optimalité introduites par Hansen et al. [1992]. Ces conditions, qui s'expriment en termes de contraintes actives du problème de second niveau à l'optimalité, sont matérialisées par des relations logiques ainsi qu'il suit: à chacune des contraintes de second niveau d'indice i , on associe une variable booléenne α_i qui est égale à 1 si la contrainte est active ($y_i^* = 0$), et à 0 sinon. La propriété ci-après énonce les conditions en question.

Propriété 5.3 (Hansen et al., 1992: théorème 4.1 et corollaire 4.2): Supposons que l'application $M(x)$ est univoque conformément à l'hypothèse (H1). Pour toute solution admissible du problème (PLDN), la nature active des contraintes de second niveau est telle que:

$$\sum_{i:B_{ij}>0} \alpha_i \geq 1 \text{ si } d_j^2 > 0, \quad (\text{m1})$$

$$\sum_{i:B_{ij}<0} \alpha_i + \alpha_{m+j} \geq 1 \text{ si } d_j^2 < 0, \quad (\text{m2}) \quad (5.2.17)$$

$$\sum_{i:B_{ij} \neq 0} \alpha_i \geq 1 \text{ si } d_j^2 = 0. \quad (\text{m3})$$

Pour une solution admissible (x^k, y^k) , si une variable booléenne α_i (respectivement α_{m+j}) est égale à 1, nécessairement, la facette $F_{\bar{i}}$ (respectivement F_j) contient (x^k, y^k) . Si l'ensemble des variables booléennes α_i d'indices dans $T \subset H$ prennent la valeur 1, alors la face du domaine induit $S_T = \bigcap_{i \in T} F_i$ contient la solution (x^k, y^k) .

À partir des relations logiques ci-dessus, nous définissons pour chacune des valeurs d_j^2 de la fonction économique de second niveau un **sous-ensemble des indices de monotonie** par:

$$J_j = \begin{cases} \{\bar{i} : B_{ij} > 0\} & \text{si } d_j^2 > 0, \\ \{\bar{i} : B_{ij} < 0\} \cup \{j\} & \text{si } d_j^2 < 0, \\ \{\bar{i} : B_{ij} \neq 0\} & \text{si } d_j^2 = 0. \end{cases} \quad (5.2.18)$$

Il faudrait noter que lorsqu'un sous-ensemble J_j est tel que $|J_j| = 1$, alors, on a nécessairement à l'optimalité $y_t = 0$, pour $\{t\} = J_j$.

Les relations logiques de l'exemple 5.4 sont les suivantes:

$$\begin{cases} \alpha_1 + \alpha_2 + \alpha_4 \geq 1, \\ \alpha_3 + \alpha_5 \geq 1, \\ \alpha_2 + \alpha_3 + \alpha_6 \geq 1. \end{cases}$$

$J_1 = \{1, \bar{2}, \bar{3}\}$, $J_2 = \{2, \bar{4}\}$, $J_3 = \{3, \bar{3}, \bar{4}\}$ sont les sous-ensembles des indices de monotonie.

5.2.2.3 Utilisation des principes de monotonie dans l'algorithme *ESLP*

Considérons un problème de PLDN, puis les sous-ensembles distincts des indices de monotonie $J_j, j = 1, 2, \dots, n_y$, ainsi que la propriété de monotonie $\sum_{i \in J_j} \alpha_i \geq 1$ qui est relative à chaque sous-ensemble J_j . L'énumération portant sur les indices de monotonie est conforme à une méthode de **branchement dichotomique**: cette méthode consiste à choisir de manière successive pour chaque sous-ensemble J_j , un seul indice $i_0 \in J_j$ et à fixer à 1 la variable booléenne α_{i_0} correspondante, les autres variables $\alpha_i, i \neq i_0$ ayant une valeur positive ou nulle. Ceci signifie que dans le programme de second niveau du problème (*PLDN*), on impose à la variable y_{i_0} de prendre une valeur nulle à l'optimalité, en s'assurant que cette contrainte supplémentaire est compatible avec les contraintes initiales du PLDN et contribue éventuellement à améliorer la valeur de la fonction économique du premier niveau. Si y_{i_0} est une variable d'écart, la contrainte correspondante est active à l'optimalité. Pour le problème de second niveau du PLDN, on déduit de la remarque 5.3 que les contraintes actives à l'optimalité coïncident avec au moins un sous-ensemble des indices de monotonie $T = \{i_1, i_2, \dots, i_{n_y}\}$, où $i_j \in J_j : j = 1, 2, \dots, n_y$. Il est question de déterminer à partir des contraintes de monotonie, un tel sous-ensemble T duquel on déduit une face du domaine induit S_T contenant une solution optimale.

Considérons à cet effet $z^k = (x^k, y^k)$ une solution admissible de l'itération k . L'usage que nous proposons de faire des principes de monotonie dans l'algorithme *ESLP* est le suivant. Soit

J_{j_0} un sous-ensemble des indices de monotonie de cardinalité minimale et non nulle; on a: $|J_{j_0}| = \min_{1 \leq j \leq n_y} |J_j|$. Considérons la numérotation initiale des sous-ensembles des indices de monotonie $J_{j_0}, J_{j_1}, J_{j_2}, \dots, J_{j_{R_0}}, J_{j_{R_0+1}}, \dots, J_{j_{n_y-1}}$; chacun des éléments de J_{j_t} constitue un sommet de niveau $t \in \{0, 1, 2, \dots, n_y - 1\}$ d'un graphe dit **graphe de monotonie** que nous désignons par G_M^0 . Les arcs de ce graphe sont les liaisons qui existent entre les indices de deux sous-ensembles de monotonie consécutifs J_{j_t} et $J_{j_{t+1}}$. Les arcs en question sont construits comme suit: un indice $i_0 \in J_{j_t}$ (sommet de niveau t) est relié à tous les indices de $J_{j_{t+1}}$ (sommets de niveau $t+1$), à l'exception d'un éventuel $i_0 \in J_{j_{t+1}}$. Une **arborescence de monotonie** est constituée de tous les sommets de G_M^0 que l'on peut atteindre à partir d'un sommet $i_0 \in J_{j_0}$ de niveau 1 dans le graphe G_M^0 . L'ensemble des arborescences de monotonie notée A_M^0 est tel que les indices de tout cheminement dans une arborescence A_m de A_M^0 sont contenus dans un sous-ensemble appelé **cheminement de monotonie** que l'on désigne par $T = \{i_0, i_1, \dots, i_r\}$ avec $i_0 \in J_{j_0}$ et pour $r > 0$, on a $i_r \in J_{j_r}$.

Un cheminement de monotonie T est tel que $\alpha_i = 1 \forall i \in T$ et $\alpha_i \geq 0$ si $i \notin T$; ce qui signifie qu'à une étape k de l'algorithme *ESLP*, les composantes de la variable de second niveau sont telle que $y_i^k = 0$ ($dy_i = -y_i^{k-1}$) $\forall i \in T$, et $y_i^k \geq 0$ ($dy_i \geq -y_i^{k-1}$) si $i \notin T$. En d'autres termes, les contraintes de second niveau dont les variables d'écart ont un indice $i \in T$ sont actives en z^k . De plus, on a par construction du graphe de monotonie, pour tout $r \geq 0$, si $i_r \in J_{j_r} \cap T$, $\alpha_{i_r} = 1$ et $\alpha_i \geq 0$ pour tout $i \in J_{j_r}$ tel que $i \neq i_r$.

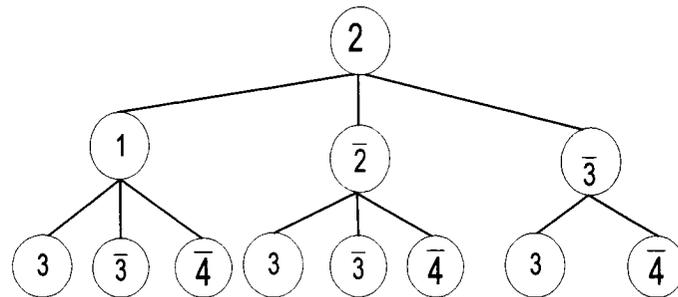
À partir de la suite ordonnée des sous-ensembles des indices de monotonie (J_2, J_1, J_3) de l'exemple 5.4, la matrice d'adjacence du graphe de monotonie G_M^0 est la suivante:

Tableau 5. 1: Matrice d'adjacence du graphe de monotonie de l'exemple 5.4

	2	$\bar{4}$	1	$\bar{2}$	$\bar{3}$	3	$\bar{3}$	$\bar{4}$
2	0	0	1	1	1	0	0	0
$\bar{4}$	0	0	1	1	1	0	0	0
1	0	0	0	0	0	1	1	1
$\bar{2}$	0	0	0	0	0	1	1	1
$\bar{3}$	0	0	0	0	0	1	0	1
3	0	0	0	0	0	0	0	0
$\bar{3}$	0	0	0	0	0	0	0	0
$\bar{4}$	0	0	0	0	0	0	0	0

On en déduit une des arborescences de monotonie de la figure 5.1 ci-dessous.

$T_1 = \{2, 1, \bar{3}\}$, $T_2 = \{2, \bar{2}, 3\}$, $T_3 = \{2, \bar{3}, \bar{4}\}$ sont des exemples de cheminements de monotonie.

**Figure 5. 1:** Exemple 5.4 - arborescence de monotonie

Nous proposons de parcourir chacune des arborescences de monotonie en profondeur d'abord. À cet effet, considérons une arborescence de monotonie A_m et un cheminement T composé des sommets d'un sous-ensemble S tel que $S = \{i_0, i_1, i_2, \dots\}$ avec $i_0 \in J_{j_0}$ et $i_r \in J_{j_r}$, $r \geq 1$. On résout successivement les programmes linéaires $PRL(z^k, \{i_0\})$, $PRL(z^k, \{i_0, i_1\})$, ..., $PRL(z^k, S)$ dont une solution optimale lorsqu'elle existe est (dx, dy) , tant que l'une des trois conditions suivantes dites **conditions de retour en arrière** n'est pas satisfaite:

- i) On a trouvé soit une solution améliorante $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$;

- ii) Un de ces programmes linéaires est irréalisable;
- iii) Pour un de ces programmes linéaires, la valeur de la fonction économique du premier niveau correspondant à une solution réalisable est supérieure à celle qui correspond à (x^k, y^k) .

Si une des conditions du retour en arrière est satisfaite sur sommet i_s contenu dans un cheminement de monotonie T , on arrête le parcours de l'arborescence de monotonie contenant T . À partir du sommet i_s , on remonte les indices du cheminement T jusqu'à l'indice du premier sommet rencontré $i_n \in T$ ayant des successeurs dans l'arborescence qui n'ont pas encore été examinés. L'examen de chacun desdits successeurs de i_n permet de construire d'autres cheminements de monotonie. Le parcours d'une arborescence de monotonie est considéré comme achevé lorsque du fait d'un retour en arrière, aucun successeur de la racine de l'arborescence ne peut plus être examiné. Au cours de l'exploration du graphe G_M^0 , on désigne par n_{i_s} le niveau du sommet i_s évoqué ci-dessus dans le graphe G_M^0 . Si on considère l'ensemble des arborescences de G_M^0 , Ro désigne la plus petite valeur de l'ensemble des niveaux des sommets de type i_s sur lequel on a effectué un retour en arrière. On a donc:

$$Ro = \min_{A_m \in A_M^0} \{n_{i_s} : i_s \in A_m\}. \quad (5.2.19)$$

Nous verrons plus loin comment utiliser ce nombre dans la construction des différents graphes de monotonie que nous désignons par G_M^t , $t = 1, 2, \dots$. Si la solution du sous-problème $PRL(z^k, T)$, $T \subset S$, à savoir (dx, dy) est telle que $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$ est une solution améliorante, alors la face du domaine induit $S_T = \bigcap_{t \in T} F_t$ contient (x^{k+1}, y^{k+1}) ; on choisit d'explorer un autre cheminement dans l'arborescence par un retour en arrière éventuellement.

Remarque 5.4: Étant donnée une contrainte de monotonie $\sum_{i \in J_j} \alpha_i \geq 1$, une méthode de branchement dichotomique consiste à imposer à une seule variable de monotonie d'indice $i_0 \in J_j$ de prendre la valeur $\alpha_{i_0} = 1$; on n'examine pas explicitement les cheminements de monotonie pour lesquels on a $\alpha_{i_0} = 1, \forall i_0 \in K$, avec $K \subseteq J_j$ et $|K| \geq 2$. Pour combler partiellement cette lacune, nous proposons de considérer des cheminements de monotonie dont les indices sont contenus dans les sous-ensembles des indices de monotonie $J_{j_{Ro+1}}, J_{j_{Ro+2}}, \dots, J_{j_{ny-1}}$. Certaines combinaisons d'indices n'ont pas été examinées du fait d'un retour en arrière dans les cas ii) et iii). Il est trivial qu'un nouvel examen des sous arborescences de monotonie dont les sommets sont contenus dans le sous graphe de monotonie d'indices dans $J_{j_0}, J_{j_1}, J_{j_2}, \dots, J_{j_{Ro}}$ ne peut pas contribuer à améliorer la valeur de la solution courante par construction du nombre Ro . L'exploration d'une arborescence de monotonie à partir des sommets contenus dans les sous-ensembles $J_{j_{Ro+1}}, J_{j_{Ro+2}}, \dots, J_{j_{ny-1}}$ quant à elle peut contribuer à améliorer la valeur de la solution.

À la lumière de ce qui précède, afin de générer de nouveaux cheminements de monotonie dignes d'intérêt, nous proposons de construire une suite de graphes de monotonie $G_M^t, t = 1, 2, \dots, n_y - Ro$ en procédant à une renumérotation des niveaux des sommets des sous-ensembles $J_{j_i}, i = 0, 1, 2, \dots, n_y - 1$. Ainsi, le graphe de monotonie G_M^1 est obtenu à partir des sous-ensembles des indices de monotonie placés dans l'ordre $J_{j_{Ro+1}}, J_{j_0}, J_{j_1}, J_{j_2}, \dots, J_{j_{Ro}}, J_{j_{Ro+2}}, \dots, J_{j_{ny-1}}$. Tous les sommets du graphe G_M^0 de niveau $Ro + 1$ deviennent les sommets de niveau 1 dans le graphe G_M^1 et les sommets de niveaux $2, 3, \dots, Ro$ dans le graphe G_M^1 sont les sommets de niveau $1, 2, 3, \dots, Ro - 1$ dans le graphe G_M^0 . De la même façon on obtient à partir du graphe G_M^1 , le graphe G_M^2 en considérant l'ordre suivant des sous-ensembles $J_{j_{Ro+2}}, J_{j_{Ro+1}}, J_{j_0}, J_{j_1}, J_{j_2}, \dots, J_{j_{Ro}}, J_{j_{Ro+3}}, \dots, J_{j_{ny-1}}$, ainsi de suite pour les graphes $G_M^t, t = 3, 4, \dots, n_y - Ro$.

5.2.2.4 Principe d'un algorithme par énumération implicite

D'une manière générale, l'énumération implicite pourrait caractériser les procédures d'exploration arborescente. Un algorithme d'optimisation par séparation (AOS) est une procédure d'énumération implicite portant sur l'ensemble des solutions du problème résolu. Les AOS sont souvent utilisés pour résoudre certains problèmes de programmation mathématique en variables binaires, entières ou mixtes. On attribue généralement le principe des algorithmes de séparation à Little et al. [1963] puis à Bertier et Roy [1964]. Hansen [1974] et Jaumard [1986] ont proposé des exposés de synthèse. Une méthode d'énumération implicite sert à définir un schéma d'exploration des ensembles des solutions d'un problème d'optimisation et peut être illustré par une arborescence. Nous utilisons une méthode d'énumération implicite portant sur des variables continues pour résoudre le programme linéarisé relaxé $PLR(z^k, T)$.

Le principe général de l'algorithme *ESLP* est conforme au parcours des cheminements de monotonie que nous avons décrit dans la sous section précédente. Plus précisément, considérons un cheminement S dans une arborescence de monotonie A_m et une solution admissible $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$ telle que (dx, dy) est une solution optimale d'un programme linéaire $PLR(z^k, T)$, $T \subset S$. Le sous-ensemble T est construit graduellement en parcourant les indices de l'ensemble S . L'évaluation sur l'état que l'on impose aux variables $y_i, i \in T$ consiste à comparer la solution du problème de second niveau $PL_2(k)$ avec la solution (x^{k+1}, y^{k+1}) . Si (x^{k+1}, y^{k+1}) est dans le domaine induit et améliore la valeur de la fonction économique f^1 , les variables $y_i, i \in T$ gardent provisoirement l'état correspondant à la direction de déplacement $dy_i, i \in T$ et la fonction économique prend la valeur correspondant à la solution pour laquelle $y_i^{k+1} = 0, i \in T$. Nous disons dans ce cas que l'évaluation $(dy_i^k = -y_i^k, i \in T)$ portant sur l'ensemble des variables $y_i, i \in T$ est concluante. On sélectionne un autre cheminement si l'une des conditions de retour en arrière de la sous section précédente est satisfaite, ou alors une autre arborescence de monotonie (s'il en existe) qui n'a pas encore été

explorée. Chacune des étapes de l'algorithme *ESLP* consiste alors à déterminer une face $S_T = \bigcap_{i \in T} F_i$ du domaine induit contenant la solution améliorante (x^{k+1}, y^{k+1}) .

Sur un sommet d'une arborescence de monotonie, on porte l'indice d'une variable de second niveau ou une variable d'écart, puis un triplet $(T, z^k, f^1(z^k))$ où T représente le sous-ensemble utilisé pour résoudre le programme $PLR(z^k, T)$. Un arc (i, j) porte la lettre *S* si on ne peut plus cheminer dans l'arborescence de monotonie à partir du sommet i . Ce qui est le cas lorsqu'une condition de retour en arrière est vérifiée sur le sommet i . La construction d'une arborescence de monotonie sera illustrée par deux exemples numériques à la sous-section 5.2.4.

Remarques 5.5:

1) Chacun des sous-ensembles des indices de monotonie comporte au plus $m+1$ éléments et un problème de PLDN défini par (5.2.9) comporte au plus $n_y + m$ sous-ensembles des indices de monotonie.

2) Par construction, un graphe de monotonie comporte au plus $N = n_y(m+1)$ sommets tous non distincts. Un cheminement dans une arborescence de monotonie est un chemin élémentaire de longueur p ($p = 1, 2, \dots, n_y$) reliant deux sommets distincts quelconque du graphe de monotonie. On déduit des propriétés d'un graphe d'ordre N , que le nombre de cheminements du graphe de monotonie de longueur p est au plus la combinatoire du choix d'une suite de $p+1$ sommets distinguables parmi N ; il est égal à $A_N^{p+1} = \frac{N!}{(N-p-1)!}$, et est en $O(N!)$.

3) Considérons une arborescence de monotonie A_j . Le degré d'un sommet $i \in A_j$, que l'on note $d_j(i)$ est le nombre de sommets de l'arborescence que l'on peut atteindre à partir du sommet i . Par construction du graphe de monotonie, on a $d_j(i) \leq m$.

Le nombre N_{AM} des arborescences de monotonie dont les sommets sont dans $J = \{1, 2, \dots, n_y\} \cup \{\bar{1}, \bar{2}, \dots, \bar{m}\}$ telles que $d_i(i) = k$ est donné par: $N_{AM} = (n-1)^{(n-k-1)} C_{k-1}^{n-2}$ (Clarke, 1963).

5.2.3 L'algorithme de programmation linéaire séquentielle par énumération implicite

Considérons le problème (PLDN); val^k désigne la valeur actuelle de la fonction économique du premier niveau, alors que $z^k = (x^k, y^k)$ désigne la solution admissible courante. Avec les notations de la sous section précédente, nous présentons dans un premier temps la procédure $PARCOURGM(G_M^t, z^k)$ utilisée par l'algorithme $ESLP$ afin de parcourir le graphe de monotonie G_M^t dont l'ensemble des arborescences de monotonie est noté A_M^t . Elle est conforme à la méthode d'exploitation des principes de monotonie que nous avons décrite dans la sous section précédente. Elle se présente ainsi qu'il suit:

Procédure $PARCOURGM(G_M^t, z^k)$

Si $t = 0$, poser $R_0 = n_y$. Poser $H = A_M^t$ (ensemble des arborescences); tant que $H \neq \emptyset$, exécuter la procédure A.

Procédure A:

Choisir une arborescence de monotonie $A_m \in H$, puis poser $H = A_M^t \setminus A_m$ et

$CH_T = \{Ch \in A_m : Ch \text{ est un chemin élémentaire}\}$. Tant $CH_T \neq \emptyset$, exécuter la procédure B.

Procédure B:

Choisir un cheminement Ch dans CH_T , puis poser $S = Ch$, $CH_T = CH_T \setminus \{Ch\}$ et $T = \emptyset$.

Tant que l'on a (i) $S \neq \emptyset$ ou (ii) une condition de retour en arrière n'est pas satisfaite, exécuter la procédure C.

Procédure C:

Choisir un indice $i \in S$ de niveau n_i dans G_M^t , puis poser $S = S \setminus \{i\}$ et $T = T \cup \{i\}$.

Soit (dx, dy) une solution de $PLR(z^k, T)$ (si elle existe)

et $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$.

Procéder à un retour en arrière sur un sommet i_0 du cheminement Ch si on a:

(i) $PLR(z^k, T)$ n'a pas de solution.

(ii) $(x^{k+1}, y^{k+1}) \in DI$ avec $f^1(x^{k+1}, y^{k+1}) < valf$. Poser dans ce cas $k = k + 1$ et

$valf = f^1(x^k, y^k)$.

(iii) $f^1(x^{k+1}, y^{k+1}) > f^1(x^k, y^k)$.

Éliminer de Ch le sommet i_0 et tous ses successeurs, puis sortir de procédure C.

Fin procédure C.

Fin procédure B.

Si $t = 0$, poser $Ro = \min(Ro, n_{i_0})$.

Fin procédure A. (sortie avec une solution admissible $z^k = (x^k, y^k)$ et le nombre Ro si $t = 0$).

Nous présentons à présent l'algorithme *ESLP* dans les détails.

Algorithme *ESLP*: Résolution du problème (*PLDN*)

Etape 0 (initialisation).

Poser $k = 0$, déterminer une solution admissible $z^0 = (x^0, y^0)$ de *PLDN* et poser $valf = f^1(x^0, y^0)$. Pour $j = 1, 2, \dots, n_y$, déterminer la liste L_M de tous les sous-ensembles des indices de monotonie J_j .

Pour tout sous-ensemble J_j tel que $|J_j| = 1$, poser $y_t = 0$, avec $\{t\} = J_j$; supprimer de la liste L_M le sous-ensemble J_j et construire le graphe de monotonie G_M^0 .

Exécuter la procédure *PARCOURGM*(G_M^0, z^0) afin de calculer une solution admissible $z^k = (x^k, y^k)$ et le nombre Ro .

Pour $t = 1, 2, \dots, n_y - Ro$, construire chaque graphe de monotonie G_M^t .

Étape 1 (application des principes de monotonie: recherche des faces du domaine induit contenant une solution admissible améliorante).

Pour $t = 1, 2, \dots, n_y - Ro$, exécuter la procédure $PARCOURGM(G_M^t, z^k)$.

Étape 2: (optimalité).

Stop (x^k, y^k) est une solution optimale du problème (PLDN) au terme de l'Étape 1.

Remarques 5.6:

1) L'algorithme *ESLP* pourrait être interprété comme une méthode de descente pour résoudre un problème de PLDN: pour chacune de ses itérations $k \geq 0$ et à partir d'une solution admissible (x^k, y^k) , l'algorithme *ESLP* permet de déterminer une direction de descente du domaine induit (dx, dy) afin de calculer une autre solution admissible (x^{k+1}, y^{k+1}) qui, lorsqu'elle existe est telle que $f^1(x^{k+1}, y^{k+1}) < f^1(x^k, y^k)$.

2) Si on met en œuvre l'algorithme *ESLP* dans une interface qui permet de résoudre un programme linéaire en lui imposant une solution de départ contrairement à l'interface MATLAB, le programme $PLR(z^k, T)$ a pour solution de départ z^k ; sa formulation est la suivante:

$PL(z^k, T)$

$$\begin{aligned} & \text{Minimiser}_{x,y} f^1(x,y) \\ & \text{s.c.} \begin{cases} (A'x + B'y)_i = b'_i, & i \in T, \\ (A'x + B'y)_i \leq b'_i, & \forall i \notin T, \\ x \in P. \end{cases} \end{aligned} \quad (5.2.20)$$

Il sera alors résolu à chaque étape de l'algorithme *ESLP* à la place du programme linéarisé $PLR(z^k, T)$ (avec $T \subset \bigcup_{1 \leq j \leq n_y} J_j$). Sa solution optimale lorsqu'elle existe est z^{k+1} . En fait, le

cheminement T conduit à une relaxation du problème de PLDN qui est aussi difficile à résoudre que le problème de départ. Sa formulation est la suivante:

$PLDN(z^k, T)$

$$\begin{array}{l} \text{Minimiser}_{x,y} f^1(x,y) \\ \text{s.c.} \left\{ \begin{array}{l} x \in P, \\ \text{Maximiser}_y d_2^t y \\ \text{s.c.} \left\{ \begin{array}{l} (A'x + B'y)_i = b'_i, \quad i \in T, \\ (A'x + B'y)_i \leq b'_i, \quad \forall i \notin T. \end{array} \right. \end{array} \right. \end{array}$$

Soit $S = \{1, 2, \dots, n_y + m\}$ et z^k une solution admissible du problème ($PLDN$), nous montrons à travers la proposition ci-après que pour trouver une solution améliorante z^{k+1} , l'algorithme $ESLP$ calcule une direction de descente du domaine induit en résolvant un nombre fini de programmes linéaires.

Proposition 5.3: Considérons le problème ($PLDN$) pour lequel l'hypothèse (H1) est satisfaite. Soit $(x^k, y^k) \in DI$ une solution admissible d'une itération k ; on suppose qu'il existe une solution améliorante (x^{k+1}, y^{k+1}) pour l'algorithme $ESLP$. Pour la calculer, l'algorithme $ESLP$ détermine une direction de descente du domaine induit en résolvant un nombre fini de programmes linéaires à chacune de ses étapes.

Preuve: Soient (x^k, y^k) une solution courante du problème ($PLDN$), (x^{k+1}, y^{k+1}) une solution améliorante (pour l'algorithme $ESLP$) à déterminer et la suite des graphes de monotonie (G_M^t) , $t = 0, 1, 2, \dots$. Soit S un cheminement de monotonie et considérons le programme linéaire $PLR(z^k, T)$, $T \subset S$ dont la formulation est la suivante:

$PLR(z^k, T)$

$$\begin{aligned}
 & \text{Maximiser}_{dx, dy, \xi} \xi \\
 & \left\{ \begin{array}{l}
 \nabla_z f^1(x^k, y^k) \begin{pmatrix} dx \\ dy \end{pmatrix} + \xi \leq 0, \\
 Adx + Bdy = 0, \\
 s.c. \left\{ \begin{array}{l}
 y_i^k + dy_i \geq 0 \quad \forall i \notin T, \\
 y_i^k + dy_i = 0 \quad \forall i \in T, \\
 x^k + dx \in P, \quad \xi \geq 0.
 \end{array} \right.
 \end{array} \right. \quad (5.2.21)
 \end{aligned}$$

Par hypothèse, il existe un cheminement T d'une arborescence de monotonie tel que (ξ^*, dx, dy) est une solution optimale du programme linéaire $PLR(z^k, T)$ avec $x^{k+1} = x^k + dx$, $y^{k+1} = y^k + dy$ et $f^1(x^{k+1}, y^{k+1}) < f^1(x^k, y^k)$.

D'après 2) remarques 5.5, chaque graphe de monotonie G_M^f comporte un nombre fini d'arborescences de monotonie, donc un nombre fini de cheminements de monotonie. Sur chaque cheminement, on résout un nombre fini de fois un programme linéaire de type $PLR(z^k, T)$ afin de déterminer une direction de descente du domaine induit. Lorsque ce dernier programme linéaire a une solution, on résout ensuite le programme linéaire $PL_2(k)$ pour vérifier que la solution actualisée $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy)$ est dans le domaine induit. Par conséquent, pour calculer une solution améliorante (x^{k+1}, y^{k+1}) , l'algorithme *ESLP* détermine la direction de descente (dx, dy) en résolvant un nombre fini de programmes linéaires. ■

Lorsque le problème (*PLDN*) a une solution, le processus itératif s'arrête lorsqu'on a exploré toutes les arborescences de monotonie. À partir d'une solution admissible $(x^k, y^k) \in DI$, $k \geq 0$, l'algorithme *ESLP* permet de déterminer un optimum local (qui peut être global) du problème (*PLDNS*) un nombre fini d'itérations. En effet, d'après la proposition ci-dessus, la solution (ξ^*, dx, dy) est obtenue en résolvant un nombre fini de programmes linéaires

de type $PLR(z^k, T)$; de plus on a $(x^{k+1}, y^{k+1}) = (x^k + dx, y^k + dy) \in DI$ lorsque (dx, dy) est une direction de descente du domaine induit. L'ensemble des solutions réalisables Ω est par hypothèse un polyèdre fermé et borné. On a $DI = \bigcup_{i \in J} S_i$ où $J \subset \{1, 2, \dots, r\}$ (remarque 5.3) et Ω contient un nombre fini de points extrêmes. Chacune des itérations de l'algorithme *ESLP*, et implicitement le parcours des différents cheminements de monotonie consiste à examiner partiellement les faces S_τ du domaine induit contenant (x^{k+1}, y^{k+1}) . De plus, (x^{k+1}, y^{k+1}) est un point extrême de Ω . La résolution du problème (PLDNS) se fait par conséquent en un nombre fini d'itérations.

5.2.4 Expérimentations numériques de l'algorithme *ESLP*

Nous avons implémenté l'algorithme *ESLP* dans une interface MATLAB sur un ordinateur PC Pentium 4 (processeur 3.2 GHZ, 1.24 GB de RAM). Dans cette interface, la résolution des programmes linéaires est faite par la méthode du point intérieur de Mehrotra [1992]. Nous illustrons le fonctionnement de l'algorithme *ESLP* sur la formulation particulière du PLDN de deux manières:

- Nous résolvons cinq problèmes de petite taille connus de la littérature pour certains.
- Pour résoudre les problèmes de grande taille, les données ont été générées i) aléatoirement selon la méthode de Audet et al. [1999], et ii) par une méthode proposée par Calamai et Vicente [1994]. Nous comparons les résultats obtenus par l'algorithme *ESLP* avec ceux obtenus par les algorithmes de Bard et Moore [1990] (*BM*) ou *CBB* développé par Audet et al. [2004].

Rappelons que l'algorithme *ESLP* résout deux types de programmes linéaires à chaque étape:

- La résolution des programmes linéaires relaxés $PLR(z^k, T)$, $T \subseteq S$ ($S \subset \bigcup_{1 \leq j \leq n_y} J_j$)

permet de déterminer une direction de déplacement (dx, dy) de l'itération courante k . Pour un problème donné, le nombre total de directions de déplacement calculé est désigné par *Neudd*;

cette quantité représente également le nombre de programmes linéaires relaxés du type $PLDN(z^k, T)$ résolus par l'algorithme BM .

- Si l'un des programmes linéaires relaxés $PLR(z^k, T)$ a une solution (dx, dy) , il faudrait vérifier que la solution $(x^k + dx, y^k + dy)$ est dans le domaine induit. On résout le problème de second niveau $PL_2(k+1)$. $Neud$ représente le nombre de fois que ce type de vérification a été faite pour un problème donné. $Neuda$ quant à lui représente le nombre des solutions admissibles qui ont contribué à améliorer la valeur de la fonction économique du premier niveau du problème de PDN résolu. Les colonnes des tableaux qui représentent les valeurs de $Neud$ et $Neudd$ comportent deux nombres: la plus petite valeur pour chacun de ces nombres correspond à l'étape de l'algorithme qui a conduit à la solution optimale, alors le nombre le plus grand désigne la valeur totale de ces nombres pour le problème résolu. Il en est de même pour le temps de résolution d'un problème de taille donnée qui est désigné par Cpu .

Pour Bard et Moore [1990], la solution optimale calculée par leur algorithme ne peut être atteinte tant que l'on n'a pas exploré 60 à 70% de l'arborescence du problème résolu. Savard [1989], dans la mise en œuvre de l'algorithme BM ou de l'algorithme MAYBIL, propose d'arrêter le processus itératif lorsque le temps d'exécution Cpu atteint la valeur 10000. Nous avons opté pour ce critère d'arrêt, qui faudrait-il le souligner est arbitraire; nous mettrons en exergue plus loin dans l'analyse des résultats, les insuffisances empiriques que nous avons observées. Ces insuffisances, liées probablement à ce critère d'arrêt peuvent avoir une influence sur la qualité de la solution.

5.2.4.1 Illustration de l'algorithme $ESLP$ sur les problèmes de PLDN de petites tailles

Considérons la formulation du problème de PLDN dans le cas général, dans laquelle on suppose que la matrice B_1 dans les contraintes du premier niveau est nulle. La solution de départ est obtenue à partir des contraintes du premier et du second niveau en prenant comme fonction objectif $f(x, y) = c_1^t x + d_2^t y$. Savard [1989] a montré que la solution ainsi obtenue est dans le

domaine induit. Pour toute itération $k \geq 0$, $z^k = (x^k, y^k)$ désigne une solution admissible. Les problèmes sur lesquels nous avons testé l'algorithme *ESLP* sont les suivants:

Problème 5.1 (Bard et Falk, 1982):

$$\begin{array}{l} \text{Maximiser}_{x,y \geq 0} \quad 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3 \\ \text{s.c.} \left\{ \begin{array}{l} \text{Minimiser}_y \quad 2y_1 + y_2 + 2y_3 \\ \text{s.c.} \left\{ \begin{array}{l} -y_1 + y_2 + y_3 \leq 1, \\ 2x_1 - y_1 + 2y_2 - 0.5y_3 \leq 1, \\ 2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1, \\ x_1, x_2, y_i \geq 0, \quad i = 1, 2, 3. \end{array} \right. \end{array} \right. \end{array}$$

Problème 5.2 (Savard, 1989):

$$\begin{array}{l} \text{Maximiser}_{x,y \geq 0} \quad 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3 \\ 1. \text{s.c.} \left\{ \begin{array}{l} x_1 + x_2 - y_3 \leq 1.3, \\ \text{Minimiser}_y \quad 2y_1 + y_2 + y_3 \\ \text{s.c.} \left\{ \begin{array}{l} -y_1 + y_2 + y_3 \leq 1, \\ 4x_1 - 2y_1 + 4y_2 - y_3 \leq 2, \\ 4x_2 + 4y_1 - 2y_2 - y_3 \leq 2, \\ x_1, x_2, y_i \geq 0, \quad i = 1, 2, 3. \end{array} \right. \end{array} \right. \end{array}$$

Problème 5.3 (Savard, 1989):

$$\begin{array}{l} \text{Minimiser}_{x,y \geq 0} \quad x + 4y \\ \text{s.c.} \left\{ \begin{array}{l} \text{Minimiser}_y \quad -y \\ \text{s.c.} \left\{ \begin{array}{l} x + y \geq 8, \\ -3x + 2y \leq 6, \\ 3x + 4y \leq 48, \\ 2x - 5y \leq 9, \\ x, y \geq 0. \end{array} \right. \end{array} \right. \end{array}$$

Problème 5.4:

$$\begin{array}{l}
 \text{Minimiser}_{x,y \geq 0} y_1 + 8y_2 \\
 \text{s.c. Maximiser}_y y_2 \\
 2. \quad \text{s.c.} \left\{ \begin{array}{l}
 x + y_1 + y_2 \leq 3, \\
 x - y_1 + y_2 \leq 1, \\
 x + y_1 - y_2 \geq 1, \\
 -x + y_1 + y_2 \leq 1, \\
 x - 3y_2 \leq 1, \\
 x + 3y_2 \geq 1, \\
 y_2 \leq 0.5, \\
 x, y_1, y_2 \geq 0.
 \end{array} \right.
 \end{array}$$

Problème 5.5: (Données générées aléatoirement selon la méthode Audet et al. [1999])

$$\begin{array}{l}
 \text{Minimiser } -10x_1 - 10x_2 - 7x_3 - 10x_4 + 5x_5 + 3x_6 + 2y_1 + 3y_2 + 4y_3 - 11y_4 - 3y_5 + 9y_6 \\
 \text{s.c.} \left\{ \begin{array}{l}
 \text{Minimiser } -y_1 + y_2 - 6y_3 + 5y_4 - 12y_5 - 10y_6 \\
 \left\{ \begin{array}{l}
 6x_4 - y_5 + 15y_6 \leq 9, \\
 -9x_1 + 3y_4 + 6y_6 \leq 11, \\
 18x_3 - 2x_4 + 5y_3 \leq 7, \\
 -3x_3 \leq 14, \\
 x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 6, \\
 x_i, y_i \geq 0, i = 1, \dots, 6.
 \end{array} \right.
 \end{array} \right.
 \end{array}$$

Examinons dans les détails la résolution du problème 5.5 par l'algorithme *ESLP*.

Résolution du problème 5.5

Après avoir mis le problème de second niveau sous forme standard, le système (5.2.22) représente le système des contraintes qui en résultent.

$$\left\{ \begin{array}{l}
 6x_4 - y_5 + 15y_6 + y_{\bar{1}} = 9, \quad (\alpha_1) \\
 -9x_1 + 3y_4 + 6y_6 + y_{\bar{2}} = 11, \quad (\alpha_2) \\
 18x_3 - 2x_4 + 5y_3 + y_{\bar{3}} = 7, \quad (\alpha_3) \\
 -3x_3 + y_{\bar{4}} = 14, \quad (\alpha_4) \\
 x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_{\bar{5}} = 6, \quad (\alpha_5) \\
 -y_1 \leq 0, \quad (\alpha_6) \\
 -y_2 \leq 0, \quad (\alpha_7) \\
 -y_3 \leq 0, \quad (\alpha_8) \\
 -y_4 \leq 0, \quad (\alpha_9) \\
 -y_5 \leq 0, \quad (\alpha_{10}) \\
 -y_6 \leq 0, \quad (\alpha_{11}) \\
 x_i \geq 0, i = 1, \dots, 6.
 \end{array} \right. \quad (5.2.22)$$

Les contraintes de monotonie sont les suivantes:

$$\begin{cases}
 \alpha_5 \geq 1; \alpha_7 \geq 1; \alpha_3 + \alpha_5 \geq 1; \\
 \alpha_1 + \alpha_9 \geq 1; \alpha_5 \geq 1; \alpha_1 + \alpha_2 + \alpha_3 \geq 1.
 \end{cases}$$

On en déduit les sous-ensembles des indices de monotonie correspondants:

$$J_1 = \{\bar{5}\}, J_2 = \{2\}, J_3 = \{\bar{3}, \bar{5}\}, J_4 = \{4, \bar{1}\}, J_5 = \{\bar{5}\}, J_6 = \{\bar{1}, \bar{2}, \bar{5}\}.$$

À l'optimalité, on a à priori: $y_2 = y_{\bar{5}} = 0$. Mais, afin d'illustrer le fonctionnement de l'algorithme *ESLP*, nous ne tenons pas compte de cette implication. La solution de départ est $z^0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0)$ et $valf^0 = -18$. Les valeurs prises par les variables d'écart ne figurent pas dans cette solution. Le graphe de monotonie G_M^0 est construit à partir de la suite ordonnée des sous-ensembles des indices de monotonie $(J_1, J_2, J_5, J_3, J_4, J_6)$. Ce qui conduit à la matrice d'adjacence du tableau 5.2. La figure 5.2 représente l'arborescence de monotonie correspondante.

Tableau 5. 2: Matrice d'adjacence du graphe de monotonie initial - problème 5.5.

	$\bar{5}$	2	$\bar{5}$	$\bar{3}$	$\bar{5}$	4	$\bar{1}$	$\bar{1}$	$\bar{2}$	$\bar{5}$
$\bar{5}$	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
$\bar{5}$	0	0	0	1	0	0	0	0	0	0
$\bar{3}$	0	0	0	0	0	1	1	0	0	0
$\bar{5}$	0	0	0	0	0	1	1	0	0	0
4	0	0	0	0	0	0	0	1	1	1
$\bar{1}$	0	0	0	0	0	0	0	0	1	1
$\bar{1}$	0	0	0	0	0	0	0	0	0	0
$\bar{2}$	0	0	0	0	0	0	0	0	0	0
$\bar{5}$	0	0	0	0	0	0	0	0	0	0

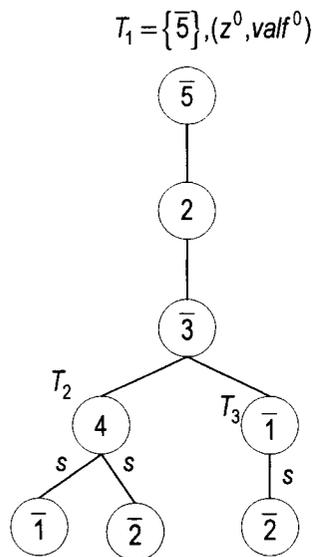


Figure 5. 2: Problème 5.5 – 1ère arborescence de monotonie.

1) Cheminements sur la première arborescence de monotonie: sur le cheminement $T_2 = \{\bar{5}, 2, \bar{3}, 4\}$, la résolution du programme $PLR(z^0, T_2)$ conduit à la solution admissible suivante:

$$z^1 = (1.9918, 3.6194, 0.3889, 0, 0, 0, 0, 0, 0, 0, 0) \text{ et } valf^1 = -58.8333.$$

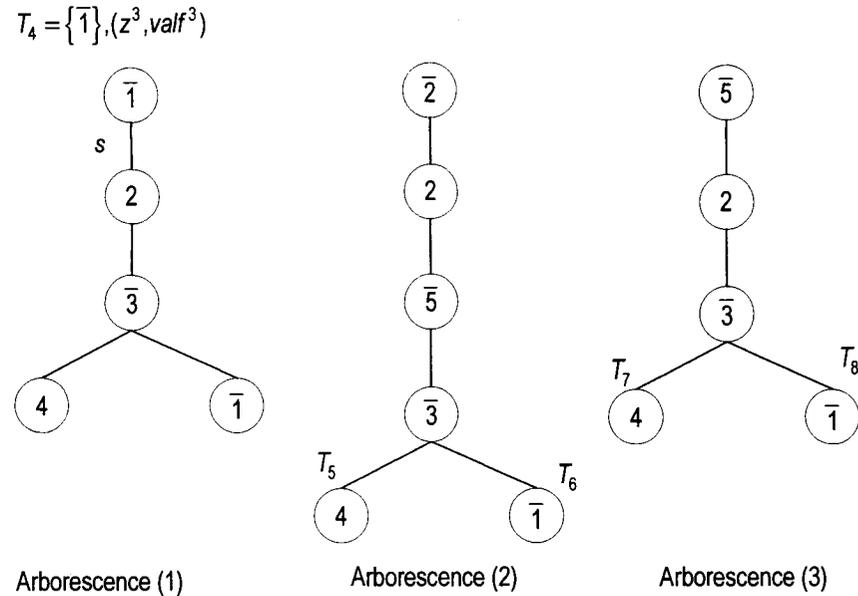


Figure 5.3: Problème 5.5 –arborescences de monotonie du 2nd graphe de monotonie.

2) Cheminement sur les arborescences de monotonie du second graphe de monotonie:

Sur l'arborescence (1), la résolution du programme $PLR(z^2, \{\bar{1}\})$ conduit à la solution admissible

$$z^3 = (0.0494, 0, 0, 2.1358, 0, 0, 0, 0, 0, 3.8148, 0, 0) \text{ et } valf^3 = -63.8148.$$

On arrête l'exploration de cette arborescence. Sur l'arborescence (2), les programmes $PLR(z^3, T_5)$ et $PLR(z^3, T_6)$ ne sont pas réalisables, avec $T_5 = \{\bar{2}, 2, \bar{5}, \bar{3}, 4\}$ et $T_6 = \{\bar{2}, 2, \bar{5}, \bar{3}, \bar{1}\}$.

On arrête l'exploration de cette arborescence. Sur l'arborescence (3), les programmes $PLR(z^3, T_7)$ et $PLR(z^3, T_8)$ ne sont pas réalisables, avec $T_7 = \{\bar{5}, 2, \bar{3}, 4\}$ et $T_8 = \{\bar{5}, 2, \bar{3}, \bar{1}\}$. On arrête également l'exploration de cette arborescence. L'exploration des arborescences de monotonie du graphe G_M^1 est terminée. z^3 est par conséquent la solution optimale.

Les tableaux 5.4 et 5.5 contiennent les résultats des problèmes de petite taille que nous avons résolus respectivement par les algorithmes *ESLP* et *BM*. (x^{opt}, y^{opt}) représente la solution

optimale, alors que f_{opt}^1 et f_{opt}^2 représentent les valeurs optimales des fonctions économiques du premier et du second niveau respectivement.

Tableau 5. 4: Algorithme *ESLP* – Problèmes de petite taille

Problèmes	(x^{opt}, y^{opt})	f_{opt}^1	f_{opt}^2	Ro	$Neud$	$Neuda$	$Neudd$
Problème 5.1	(0,0.9,0,0.6,0.4)	29.2	1.4	1	2	2	2
					20		41
Problème 5.2	(0.5,0.8,0,0.2,0.8)	18.4	1.8	2	5	3	9
					20		36
Problème 5.3	(12,3)	24	-3	1	3	3	3
					3		3
Problème 5.4	(1.75,1,0.25)	3	-0.25	1	3	2	3
					6		7
Problème 5.5	(0.3103,0,0,0,0.5, 0.2598,0.0601,0.1029,0,0)	-6.38	19.07	5	7	4	7
					14		18

Tableau 5. 5: Algorithme *Bard et Moore* – Problèmes de petite taille

Problèmes	(x^{opt}, y^{opt})	f_{opt}^1	f_{opt}^2	$Neud$	$Neuda$	$Neudd$
Problème 5.1	(0,0.9,0,0.6,0.4)	29.2	1.4	9	3	14
				14		25
Problème 5.2	(0.5,0.8,0,0.2,0.8)	18.4	1.8	9	3	12
				21		39
Problème 5.3	(12,3)	24	-3	7	3	8
				15		27
Problème 5.4	(1.75,1,0.25)	3	-0.25	14	3	19
				231		459
Problème 5.5	(0.3103,0,0,0,0.5, 0.2598,0.0601,0.1029,0,0)	-6.38	19.07	9	3	12
				41		79

Pour les problèmes de petites tailles 5.1 à 5.4, nous avons constaté que l'algorithme *ESLP* a fourni l'optimum global à partir du seul graphe de monotonie initial G_M^0 .

5.2.4.2 Résolution des problèmes de grande taille

Nous avons traité trois types de problèmes:

A) Les données relatives aux problèmes de PLDN ont été générées aléatoirement selon la méthode utilisée par Audet et al. [1999] : pour les coefficients des matrices des contraintes, ceux de la matrice B varient uniformément entre -20 et 20, alors que ceux de la matrice A varient de 0 à 20 avec une densité de 8%. Les coefficients des vecteurs c^1 et d^1 varient uniformément dans les intervalles respectifs [-10 10] et [10 20], et ceux de d^2 sont dans l'intervalle [0 10]. Les coefficients des seconds membres des contraintes sont uniformément distribués entre -40 et 40.

Enfin, la contrainte additionnelle $\sum_{j=1}^{n_x} x_j + \sum_{j=1}^{n_y} y_j \leq n_x + n_y$ est ajoutée aux contraintes de second niveau afin d'obtenir une relaxation bornée. Les résultats sont présentés dans les tableaux 5.6 et 5.7. Le triplet (n, n_y, m) caractérise la taille de chacun des problèmes résolus.

Tableau 5. 6: Algorithme *ESLP* – Problèmes Audet et al. [1999]

n	n_y	m	Ro	$Neud$	$Neuda$	$Neudd$	Cpu	F_{ESLP}^{opt}
70	35	30	7	8534	12	13398	1139	418.92
				11176		17887	1517	
90	45	30	28	48	6	60	6.17	563.78
				2459		6661	459	
100	33	50	8	63	10	8871	682	97.85
				63		10001	10001	
100	30	45	11	118	5	259	26	328.49
				3631		6413	800	
110	40	40	1	1084	10	9983	797	480.66
				7685		62232	5297	
120	33	33	1	3981	9	5128	452	665.25
				14378		19327	1750	
120	60	30	32	9966	8	171515	9341	671.2
				10219		182987	10001	

Tableau 5. 7: Algorithme *Bard et Moore* – Problèmes Audet et al. [1999]

n	n_y	m	<i>Neud</i>	<i>Neuda</i>	<i>Neudd</i>	<i>Cpu</i>	F_{BB}^{opt}	F_{ESLP}^{opt}
70	35	30	139606	10	279164	14827	411.05	418.92
			188599		3777153	20001		
90	45	30	460	5	884	80	558.4	563.78
			55698		111360	10001		
100	33	50	76	3	104	18	97.85	97.85
			76		149	22		
100	30	45	70	2	95	12	324	328.49
			41813		83586	10001		
110	40	40	1120	5	2210	337	480.66	480.66
			36240		72442	10001		
120	33	33	7006	5	13974	1084	641.38	665.25
			67496		134957	10001		
120	60	30	26630	8	53221	6846	671.20	671.2
			40131		80210	10001		

Pour les problèmes résolus ici, nous constatons que les solutions trouvées par l'algorithme *BM* sont moins bonnes que celles trouvées par l'algorithme *ESLP*: le processus itératif de l'algorithme *BM* a été interrompu du fait du critère d'arrêt $Cpu \geq 10000$. En utilisant le critère d'arrêt $Cpu \geq 20000$ pour le problème de taille $(n, n_y, m) = (70, 35, 30)$, la solution trouvée par l'algorithme *BM* reste toujours moins bonne que celle trouvée par l'algorithme *ESLP*.

B) Sur des problèmes de tailles moyennes générés selon la méthode ci-dessus, nous avons procédé à une comparaison des résultats obtenus par l'algorithme *ESLP* à ceux trouvés par les algorithmes *BM* ou *CBB* de Audet et al. [2004]. Cette comparaison a porté sur sept problèmes de taille $(n, n_y, m) = (70, 35, 20)$ avec $n = n_y + n_x$. Les tableaux 5.8 et 5.9 ci-dessous-présentent les résultats de cette étude; F_{ESLP}^{opt} , F_{CBB}^{opt} et F_{BM}^{opt} désignent respectivement la valeur de la fonction économique du premier niveau obtenue par les algorithmes *ESLP*, *CBB* et *BM*.

Sur tous les problèmes traités, l'étude comparative des tableaux 5.8 et 5.9 montrent que les résultats obtenus par l'algorithme *ESLP* sont relativement au moins aussi bons que ceux

obtenus par les algorithmes *CBB* ou *BM*. Le résultat trouvé par l'algorithme *ESLP* sur l'exemple 6 est meilleur que celui fourni par l'algorithme *CBB*; la même observation a été faite sur le problème de taille $(n, n_y, m) = (70, 35, 30)$ du tableau 5.5 pour lequel on a trouvé $F_{ESLP}^{opt} = 418.92$ et $F_{CBB}^{opt} = 396.11$. Avec le critère d'arrêt $Cpu \geq 10000$ pour l'Exemple4, la solution trouvée par l'algorithme *ESLP* a pour valeur $F_{ESLP}^{opt} = 530.81$, alors que l'on a $F_{CBB}^{opt} = F_{BM}^{opt} = 532.23$. En obligeant l'algorithme *ESLP* à calculer une 7^{ème} solution admissible améliorante pour ce problème, on a trouvé $F_{ESLP}^{opt} = 531.31$ au bout de $Cpu = 83794$, $Neud = 100522$ et $Neudd = 438779$. Il faudrait souligner que l'algorithme *ESLP* mis en œuvre sur un ordinateur PC a été développé dans l'interface MATLAB qui, pour résoudre un programme linéaire, fait appel à une méthode du point intérieur. Quant à l'algorithme *CBB*, il a été écrit en langage C et mis en œuvre sur un ordinateur *ULTRA 60 Solaris 2.7-05*, la résolution d'un programme linéaire étant faite dans l'interface *CPLEX8.1*. Des problèmes numériques liés à la précision des calculs propres à chacun des environnements de mise en œuvre pourraient constituer une des explications liées à ces résultats, outre celui du critère d'arrêt $Cpu \geq 10000$ pour l'Exemple4.

Tableau 5. 8: Tests comparatifs - Algorithmes *ESLP* et *CBB*

Problèmes	Ro	Neud	Neuda	Neudd	Cpu	F_{ESLP}^{opt}	F_{CBB}^{opt}
Exemple1	12	968	3	1423	75	574.68	576.63
		10709		16000	836		
Exemple2	17	6086	4	19024	970	656.55	656.55
		62181		214452	10001		
Exemple3	13	27	2	28	20.7	599.93	599.93
		61136		179918	9419		
Exemple4	21	15949	6	67683	7882	530.83	532.23
		21078		88090	10001		
Exemple5	12	93	2	112	20.7	479.17	479.17
		65431		77253	10001		
Exemple6	18	4561	11	7960	1197	492.69	486.27
		448323		76244	10001		
Exemple7	14	50	2	80	12.5	632.46	632.46
		6154		13704	1933		

Nous avons constaté d'une manière générale, que l'algorithme *ESLP* est arrivé plus rapidement à l'itération ayant conduit à une solution optimale que l'algorithme *BM*, du moins 9 fois sur 14. De plus, l'algorithme *BM* n'a pas été en mesure de résoudre l'Exemple7, même avec le critère d'arrêt $Cpu \geq 20000$.

Tableau 5. 9: Tests comparatifs – Algorithmes *Bard et Moore*, *ESLP* et *CBB*

Problèmes	Neud	Neuda	Neudd	Cpu	F_{BM}^{opt}	F_{ESLP}^{opt}	F_{CBB}^{opt}
Exemple1	25601	6	51173	2615	574.68	574.68	576.63
	66219		232411	10001			
Exemple2	100	3	177	177	656.55	656.55	656.55
	130977		261936	10001			
Exemple3	172	4	320	14	599.93	599.93	599.93
	83365		166725	7601			
Exemple4	4494	4	8964	393	532.23	531.31	532.23
	96329		192639	10001			
Exemple5	1580	4	3136	136	479.17	479.17	479.17
	77128		154234	10001			
Exemple6	137	3	251	251	492.69	492.69	486.27
	84142		168264	10001			
Exemple7	1	1	1	0	613.95	632.46	632.46
	52258		1	10001			

C) Les problèmes de PLDN (sous forme de minimiser) ont été construits selon la méthode proposée par Calamai et Vicente [1994]; la valeur des coefficients du paramètre ρ du second membre des contraintes de second niveau a pour composantes $\rho_i = \text{modulo}(i,5) + 4$,

$i = 1, 2, \dots, \frac{m}{3}$. Pour ce type de problème, la solution optimale est connue d'avance. Le tableau

5.10 présente les résultats obtenus par l'algorithme *ESLP*: F_{THEO}^{opt} désigne la valeur optimale théorique de la fonction économique du premier niveau. Ces problèmes se sont avérés les plus

difficiles à résoudre en terme de temps de calcul et du nombre de programmes linéaires résolus pour chaque sous-problème.

Tableau 5. 10: Algorithme *ESLP* - Problèmes Calamai et Vicente [1994]

n	n_y	m	Ro	<i>Neud</i>	<i>Neuda</i>	<i>Neudd</i>	<i>Cpu</i>	F_{ESLP}^{opt}	F_{THEO}^{opt}
20	10	40	10	46	4	85	4.5	28	28
				48		87	4.7		
30	15	60	15	632	5	1461	83	42	42
				635		1464	84		
40	20	80	20	11192	6	26749	1236	56	56
				11197		26754	1237		
50	24	98	24	159443	5	390754	19024	69	69
				160345		401360	20001		
60	29	118	29	195567	6	473466	26857	84	83
				220543		535772	30001		
70	23	116	23	65917	6	164674	8205	65	65
				65923		164680	8206		
70	35	140	35	11485	5	27187	3193	101	98
				36455		87060	10001		
80	27	134	27	28617	6	69845	8240	76	75
				72330		176141	20001		
80	39	158	39	187668	7	386724	25612	114	111
				186068		455341	30001		

Pour $n \geq 50$, le tableau 5.10 comporte les résultats obtenus par l'algorithme *ESLP* en utilisant les critères d'arrêt $Cpu \geq 10000$, $Cpu \geq 20000$ ou $Cpu \geq 30000$. Il faudrait noter que la valeur de la solution trouvée par l'algorithme *ESLP* avec le critère d'arrêt $Cpu \geq 30000$ a été améliorée, comparée à la solution trouvée avec le critère d'arrêt $Cpu \geq 10000$ ou $Cpu \geq 20000$. Par exemple, le critère d'arrêt $Cpu \geq 10000$ conduit pour $n = 50$ à $F_{ESLP}^{opt} = 70$ au terme de $Cpu = 1906$. Par contre avec le critère d'arrêt $Cpu \geq 20000$, on trouve $F_{ESLP}^{opt} = F_{THEO}^{opt} = 69$ au terme de $Cpu = 19024$.

Le fait de ne pas disposer d'un critère d'arrêt efficace pour les algorithmes *BM* ou *ESLP* fait en sorte que ces algorithmes peuvent être arrêtés prématurément avant que la solution optimale globale n'ait été trouvée. De plus, à la lumière des expérimentations numériques que nous avons réalisées, on effectue pour certains problèmes un nombre considérable d'itérations, alors que la solution optimale a déjà été trouvée.

En parcourant toutes les arborescences de monotonie, il n'est pas exclu que l'algorithme *ESLP* arrive à déterminer un optimum global de chaque problème résolu. Seulement, il faudrait arriver à un compromis entre le temps d'exécution et la meilleure solution trouvée. Avec le critère d'arrêt $Cpu \geq 10000$, la capacité de l'algorithme *ESLP* à calculer rapidement plusieurs solutions admissibles ou à trouver au moins 96% de la valeur d'un optimum global pour tous les problèmes que nous avons résolus démontre que cet algorithme est non seulement digne d'intérêt, mais aussi perfectible, notamment pour ce qui est de la procédure de parcours des arborescences de monotonie. La fiabilité de l'algorithme *ESLP* dépend en effet de la finesse avec laquelle la suite des graphes de monotonie G_M^t est construite, puis explorée. De plus, nous avons constaté que lorsque $\max(n,m) \leq 100$, l'algorithme *ESLP* a permis de calculer un optimum global sur la quasi totalité des problèmes résolus.

On constate enfin que l'algorithme *ESLP* présente quelques similitudes avec l'algorithme *HJS* de Hansen et al. [1992]: dans l'algorithme *ESLP*, l'élimination de certaines variables de second niveau se fait implicitement grâce aux cheminements dans les arborescences de monotonie. Notons que la vitesse d'exécution de l'algorithme *HJS* pourrait être améliorée si la résolution des différents programmes linéaires relaxés prend en compte la meilleure solution admissible connue en la considérant comme solution de départ de ces programmes. De plus, à chaque étape, la taille des programmes linéaires résolus par l'algorithme *ESLP* est inférieure à la taille de ceux résolus par l'algorithme *BM*.

En définitive, l'algorithme *ESLP* dans le développement actuel constitue un compromis qui permet de trouver une bonne estimation de la solution optimale d'un problème de PLDN à moindre coût.

CONCLUSION

Dans cette thèse, nous avons étudié le problème de programmation mathématique avec contraintes d'équilibres (MPEC) d'une manière générale, et le problème de programmation mathématique à deux niveaux (PDN) en particulier, des points de vue théorique et algorithmique. Après avoir fait une revue de littérature portant sur ces deux classes de problèmes, nous avons établi des propriétés relatives à l'ensemble des solutions du problème de PDN, qui peuvent contribuer à rendre aisées les analyses théoriques consacrées aux algorithmes qui calculent ces solutions. Ces propriétés ont été établies à partir des caractéristiques des solutions du problème de MPEC ou du problème défini par une inégalité variationnelle. Nous avons utilisé la décomposition en branches de la formulation KKT du problème de programmation linéaire à deux niveaux (PDN_{KKT}) pour établir une condition suffisante d'optimalité de ce problème. Sous des hypothèses appropriées, nous avons montré que le problème (PDN_{KKT}) peut être transformé en une suite de programmes mathématiques localement réguliers, et nous avons établi des conditions d'optimalité de la catégorie de problèmes qui en résulte.

Nous avons par la suite développé une classe d'algorithmes basés sur une méthode de programmation linéaire séquentielle. La convergence linéaire de cette classe d'algorithmes a été établie dans le cas particulier de la programmation mathématique classique. Le premier de ces algorithmes, qui s'applique à la résolution du problème de MPEC fait appel à une méthode d'approximation des contraintes d'équilibre par une fonction de mérite. Une méthode originale a été développée pour actualiser, à partir des seules données du problème à résoudre, le paramètre qui caractérise la fonction de mérite utilisée.

Pour développer le second algorithme, à savoir l'algorithme de programmation linéaire séquentielle par énumération (*ESLP*), nous avons fait appel à la formulation KKT d'un problème de PDN à contraintes linéaires (PLDN), et à une exploitation judicieuse des contraintes d'équilibre à l'aide des principes de monotonie. Nous avons introduit pour la première fois les concepts d'arborescences ou de cheminements de monotonie qui nous ont permis de résoudre les

problèmes de PLDN, en déterminant à chaque itération une face du domaine induit contenant une solution admissible.

Cependant, certaines questions restent ouvertes. Au niveau théorique, il faudrait rechercher des conditions qui confèrent à l'algorithme *ESLP* la propriété de convergence globale à la lumière de ses performances prometteuses actuelles; pour cela, il faudrait déterminer un ordre de priorité optimal sur le choix des variables à examiner lors du parcours des différents cheminements de monotonie.

Au niveau algorithmique, si l'objectif théorique ci-dessus énoncé est atteint, on devrait pouvoir mettre en œuvre une procédure efficace d'exploration des contraintes de monotonie par un algorithme de type *ESLP* qui serait alors globalement convergent. Une généralisation aux cas non linéaires des algorithmes de programmation linéaire séquentielle développés dans cette thèse représenterait également une perspective intéressante. On pourrait alors envisager dans un premier temps la conception d'un algorithme de programmation quadratique séquentielle par énumération qui utilise les principes de monotonie pour résoudre la formulation KKT d'un problème de PDN quadratique quasi convexe, puis dans un cas plus général où les contraintes du problème de PDN sont convexes, en faisant appel aux méthodes d'approximation tangentielles des contraintes. Il faudrait dans ce cas inclure une méthode d'analyse de la monotonie similaire à celle que nous avons développée dans cette thèse. Une autre perspective intéressante consisterait à proposer une méthode de résolution d'un problème de MPEC par un algorithme de programmation linéaire séquentielle par énumération, en exploitant efficacement la nature disjonctive des contraintes de ce problème.

Enfin, la prise en compte dans le traitement des données de l'analyse statistique des erreurs d'arrondi ou de troncature permettrait de développer des algorithmes de résolution des problèmes de PDN fiables et numériquement stables. En effet, sous une interface comme MATLAB par exemple, le type numérique ne permet que de manipuler des entiers avec 15 chiffres significatifs (au plus). Les résultats de certaines opérations impliquant à la fois des nombres très grands ou très petits peuvent être entachées d'erreurs susceptibles de conduire à des conclusions erronées.

BIBLIOGRAPHIE

ABOU-KANDIL et BERTRAND, P. (1987). Government private sector relations as a Stackelberg game: A degenerate case. *Journal of Economical Dynamics and Control*, 11, 513-517.

AHN, B.H. (1979). "Computation of market equilibria for policy analysis. The project independence evaluation study PIES approach." Garland Publishing, New York and London.

ANANDALINGAM, G., MATHIEU, R., PITTARD, L. et SINHA, N. (1983). Artificial Intelligence based Approaches for solving Hierarchical Optimization Problems. In R. Sharda, B. Golden, E. Wasil, O. Balci and W. Stewart, editor, *Impacts of Recent Computer Advances on Operations Research*, 289–301, Elsevier Science Publishing Co., Inc.

ANANDALINGAM, G. et FRIEZ, T.L. (1992). Hierarchical optimization. An introduction. *Annals of Operations Research*, 3, 1-11.

ANDREANI, R. et MARTINEZ, J.M. (1998). On the solution of the extended linear complementarity problems. *Linear Algebra and its Applications*, 281, 247-257.

ANDREANI, R. et MARTINEZ, J.M. (2001). On the solution of mathematical programming problems with equilibrium constraints. *Math. Methods of Operations Research*, 54, 345-358.

ARROW, K. et G. DEBREU, G. (1954). Existence of equilibrium for a competitive economy. *Econometrica*, 22, 265-290.

AUDET, C., HANSEN, P., JAUMARD, B. et SAVARD, G. (1997). Links between linear bilevel and mixed 0-1 programming problems. *Journal of Optimization Theory and Applications*, 93:2, 273-300.

AUDET, C., HANSEN, P., JAUMARD, B. et SAVARD, G. (1999). A symmetrical linear max-min approach to disjoint bilinear programming. *Mathematical Programming*, 85, 573-592.

AUDET, C., SAVARD, G. et ZGHAL, W. (2004). Gomory-like cuts for linear bilevel programming. *Technical Report G-2004-13, Les Cahiers du GERAD. Soumis à Journal of Optimization Theory and Applications*.

AUSLENDER, A. (1976). "Optimisation. Méthodes Numériques". Paris. Masson.

BARD, J.F. et FALK, J.E. (1982). An explicit Solution to the multi-level programming problem. *Computers and Operations Research*, 9:1, 77-100.

BARD, J.F. (1983a). An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, 31, 670-684.

BARD, J.F. (1983b). An algorithm for solving the general bilevel programming problem. *Mathematics of Operations Research*, 8:2, 260-272.

BARD, J.F. (1984). Optimality conditions for the bilevel programming problem. *Naval Research Logistics Quarterly*, 31, 13-26.

BARD, J.F. et MOORE, J.T. (1990). A Branch and Bound Algorithm for the Bilevel Programming Problem. *SIAM J. Sci. Stat. Comput*, 11, 281-292.

BARD, J.F. (1998). "Practical Bilevel Optimization. Algorithms and Applications." Dordrecht. Kluwer Academic Publishers.

BARD, J.F., PLUMMER, J.C. et SOURIE, J.C. (2000). A bilevel programming approach to determine tax credits for biofuel production. *European Journal of Operational Research*, 120, 30-46.

BEN-AYED, O et BLAIR, C. (1990). Computational difficulties of bilevel linear programming. *Operations Research*, 38, 556-560.

BEN-AYED, O. (1993). Bilevel linear programming. *Computers and Operations Research*, 20:5, 485-501.

BENSON, H.P. (1989). On the structure and properties of a linear multilevel programming problem. *Journal of Optimization Theory and Applications*, 60, 353-373.

BERGE, C. (1963). "Topological spaces." Macmillan, New York.

BERTIER, P. et ROY, B. (1964). Procédures de résolution pour une classe de problème pouvant avoir un caractère combinatoire. *Cahier du Centre d'Études de Recherche Opérationnelle*, 6.

BIALAS, W.F., KARWAN, M.H. et SHAW, J. (1980). A parametric complementary pivot approach for two level linear programming. *Technical Report 80-2, Operations Research Program, State University of New York at Buffalo*.

BIALAS, W.F. et KARWAN, M.H. (1982). On two-level optimization. *IEEE Transactions on Automatic Control*, AC-27:1.

BIALAS, W.F. et KARWAN, M.H. (1984). Two-level linear programming. *Management Science*, 30:8, 1004-1020.

BURKE, J. et XU, S. (2000). A non-interior predictor-corrector path following algorithm for the monotone linear complementarity problem. *Mathematical Programming, Ser. A* 87, 113-130.

BRACKEN, J. et MCGILL, J. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research*, 21, 37-44.

CALAMAI, P.H. et VICENTE, L.N. (1993). Generating linear and linear-quadratic bilevel programming problems. *SIAM J. Scientific and Statistical Computing*, 14, 770-782.

CALVETE, H.I. et GALE, N. (1998). On the quasiconcave bilevel programming problems. *Journal of Optimization Theory and Applications*, 98:3, 613-622.

CALVETE, H.I. et GALE, N. (2004). Solving linear fractional bilevel programs. *Operations Research Letters*, 32, 143-151.

CAMPÊLO, M., DANTAS, S. et SCHEIMBERG, S. (2000). A note on a penalty function approach for solving bilevel linear programs. *Journal of Global Optimization*, 16, 245-255.

CANDLER, W. et NORTON, R. (1977a). Multi-level programming. *World Bank Development Research Center Discussion Paper*, 20, Washington, DC.

CANDLER, W. et NORTON, R. (1977b mai). Multi-level programming and development policy. *World Bank Development Research Center Discussion Paper*, 258, Washington, DC.

CANDLER, W. et TOWNSLEY, R. (1982). A linear two-level programming problem. *Computers and Operations Research*, 9, 57-76.

CHEN, B. et HARKER, P. (1993). A non-interior-point continuation method for linear complementarity problems. *SIAM J. Matrix Anal. Appl*, 14, 1168-1190.

CHEN, B., CHEN, X. et KANZOW, C. (2000). A penalized fisher-Burmeister NCP-function. *Mathematical Programming*, Ser. A 88, 211-216.

CHEN, C. et MANGASARIAN, O.L. (1996). A Class of smoothing functions for nonlinear and mixed complementarity problems. *Comput. Optim. Appl.*, 5, 97-138.

CHEN, Y. et FIORIAN, M. (1995). The nonlinear bilevel programming problem: formulation, regularity and optimality conditions. *Optimization*, 32, 193-309.

CLARKE, L.E. (1963). On Cayley's formula for counting trees. *Proc. Cambridge Phil. Soc.*, 59, 509-517.

CLARKE, F.H. (1983). "*Optimization and Nonsmooth Analysis*." New-York. Wiley-Interscience. 1990, Reprinted by SIAM, Philadelphia.

COLSON, B., MARCOTTE, P. et SAVARD, G. (2002). A trust-region method for nonlinear bilevel programming: algorithm and computational experience. *Technical report, Département de Mathématique, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgique*.

COLSON, B., MARCOTTE, P. et SAVARD, G. (2005). Bilevel programming: a survey. *4OR*, 3, 87-107.

COTTLE, R.W., PANG, J.-S. et STONE, R.E. (1992). "*The linear complementarity problem*." Academic Press: Boston.

DEMPE, S. (1987). A simple algorithm for linear Bilevel programming problem. *Optimization*, 1, 373-385.

DEMPE, S. (1992). A necessary and sufficient optimality for bilevel programming problem. *Optimization*, 25, 341-354.

DEMPE, S. (2000). Bundle algorithm applied to bilevel programming problems with non-unique lower level solutions. *Computational Optimization and Applications*, 15:2, 145-166.

DEMPE, S. et BARD, J.F. (2001). Bundle trust-region algorithm for bilinear bilevel programming. *Journal of Optimization Theory and Applications*, 110, 265-288.

DEMPE, S. (2002). "Foundations of bilevel programming." Kluwer Academic Publishers, Dordrecht et al.

DEMPE, S. (2003). Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52, 333-359.

DENEL, J et HUARD, P. (1974). Programmation non linéaire et linéarisation. 237-260 R. W. Cottle and J. Krarup (eds.), *Optimization Methods for Resource Allocation*, The English Universities Press Ltd.

EAVES, B.C. (1971). On the basic theorem of complementarity. *Mathematical Programming*, 1, 68-75.

EAVES, B.C. (1978). Computing stationary points. *Mathematical Programming Study*, 7, 1-14.

EDMUNDS, T. et BARD, J. F. (1991). Algorithms for nonlinear bilevel mathematical programming. *IEEE Transactions on Systems, Man, and Cybernetics*, 21, 83-89.

EDMUNDS, T. et BARD, J. F. (1992). An algorithm for the mixed-integer nonlinear bilevel programming problems. *Annals of Operations Research*, 34, 149-162.

ENGELKE, S. et KANZOW, C. (2002). Improved smoothing-type methods for the solution of linear programs. *Numer. Math.*, 90, 487-507.

ETOA ETOA, J.B. (1987). *Méthodes simpliciales numériquement stables pour la résolution des programmes linéaires à matrices des contraintes très creuses*. Thèse de doctorat en mathématiques (option recherche opérationnelle), Université Pierre et Marie Curie, Paris 6.

FACCHINEI, F. et SOARES, J. (1997). A new merit function for nonlinear complementarity problems and related algorithm. *SIAM Journal on Optimization*, 7, 227-247.

FACCHINEI, F., JIANG, H. et QI, L. (1999). A smoothing method for mathematical programs with equilibrium constraints. *Mathematical Programming*, 85, 107-134.

FALK, J.E., et LIU, J. (1995). On bilevel programming, Part I. General nonlinear case. *Mathematical Programming*, 70, 47-72.

FERRIS, M.C. et PANG, J.S. (1997). Engineering and economic application of complementarity problems. *SIAM Review*, 19, 669-713.

FERRIS, M.C. et KANZOW, C. (1998). Complementarity and related problems. A survey. *Dipartimento di Informatica e Sistemistica, Università di Roma La Sapienza, Via Buonarroti 12, 00185 Roma, working paper*.

FIACO, A.V. (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10, 287-311.

FIGUEIREDO, I.N., JUDICE, J.J. et SILVÉRIO, S.R. (2005). A class of mathematical programs with equilibrium constraints: A smooth algorithm and applications to contact problems. *Optimization and Engineering*, 6:2, 203-239.

FISHER, A. et JIANG, H. (2000). Merit functions for complementarity and related problems. A survey. *Computational and Applications*, 17, 159-182.

FISHER, A. (1992). A special newton-type optimization method. *Optimization*, 24, 269-284.

FLEGEL, M.L. et KANZOW, C. (2005). Abadie-type constraint qualification for mathematical programs with equilibrium constraints. *Journal of Optimization Theory and Applications*, 124:3, 595-614.

FORTUNY-AMAT, J. et McCARL, B. (1981). A representation and economic interpretation of two-level programming problem. *Journal of the Operational Research Society*, 32, 783-792.

FRIESZ, T., SUWANSIRIKUL, C. et TOBBIN, R. (1987). Equilibrium decomposition optimization: a heuristic for the continuous equilibrium network design problem. *Transportation Science*, 21, 254-263.

FUKUSHIMA, M. (1992). Equivalent difference optimization problems and descent methods for variational inequality problems. *Mathematical Programming*, 53, 99-110.

FUKUSHIMA, M., LUO, Z.Q. et PANG, J.S. (1998). A global convergent sequential quadratic programming algorithm for mathematical programs with linear complementary constraints. *Computational Optimization and Applications*, 10, 5-34.

GAUVIN, J. (1977). A necessary and sufficient regularity condition to have bounded multipliers in non convex programming. *Mathematical Programming*, 12, 136-138.

GENDREAU, M., MARCOTTE, P. et SAVARD, G. (1996). A hybrid tabu-ascent algorithm for the linear bilevel programming problem. *Journal of Global Optimization*, 9, 1-14.

GÜMÜS, Z.H. et FLOUDAS, C.H. (2001). Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization*, 20, 1-31.

HANSEN, P. (1974). Les procédures d'optimisation et d'exploration par séparation et évaluation. *Dordrecht: Combinatorial Programming*, B. Roy éd., Reidel, 19-65.

HANSEN, P., JAUMARD, B et LU, S.H. (1989). A frame work for algorithms in globally optimal design. *ASME, Journal of Mechanisms Automation and Transmissions*, 111, 353-360.

HANSEN, P., JAUMARD, B. et SAVARD, G. (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13, 1194-1217.

HARKER, P.T. et PANG, J.S. (1988). On the existence of optimal solutions to mathematical programs with equilibrium constraints. *Operations Research Letters*, 7:2, 61-64.

HARKER, P.T. et PANG, J.S. (1990). Finite-dimensional variational duality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical Programming*, 48, 161-220.

HAURIE, A., SAVARD, G. et WHITE, D. (1990). A note on an efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, 38, 553-555.

HOGAN, W.W. (1973). Point to set maps in mathematical programming. *SIAM Review*, 15:3.

HOTTA, K. et YOSHIE, A. (1999). Global convergence of a class of non-interior-point algorithms using Chen-Harker-Kanzow function for nonlinear complementarity problems. *Mathematical Programming*, 86, 105-134.

HU, X. et RALPH, D. (2003). A note on sensitivity of value functions of mathematical programs with equilibrium constraints. *Mathematical Programming*, 93, 265-279.

HU, X.M. et RALPH, D. (2004). Convergence of a penalty method for mathematical programming with complementarity constraints. *Journal of Optimization Theory and Applications*, 123:2, 365-390.

HUANG, Z. (2002). Sufficient conditions on non emptiness and boundness of the solution set of the P_0 -function nonlinear complementarity problem. *Operations Research Letters*, 30, 61-64.

JAUMARD, B. (1986). "Extraction et utilisation des relations booléennes pour la résolution des programmes linéaires en variables 0-1." Thèse de doctorat en mathématiques appliquées, École Nationale Supérieure des Télécommunications, Paris.

JIANG, H. RALPH, D. (2000). Smooth SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 10, 779-808.

JEROSLOW, R.G. (1985). The polynomial hierarchy and simple model for competitive analysis. *Math. Programming*, 32, 146-164.

JÙDICE, J. et FAUSTINO, A.M. (1992). A sequential LCP method for bilevel linear programming. *Annals of Operations Research*, 34, 86-106.

KANZOW, C. (1995). Some noninterior continuation methods for linear complementarity problems. *Manuscript, Institut of Applied Mathematics, University of Hamburg Hamburg, Germany.*

KANZOW, C. (1996). Some non interior continuation methods for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 17, 851-868.

KANZOW, C. et JIANG, H. (1998). A continuation method for strongly monotone variational inequalities. *Mathematical Programming*, 81, 103-125.

KARAMARDIAN, S. (1971). Generalized complementarity problems. *Journal of Optimization Theory and Applications*, 8, 131-167.

KOCVARA, M.K. et OUTRATA, J.V. (2003). Optimization problems with equilibrium constraints and their numerical solution. *Mathematical Programming*, 101:1, 119-149.

KOJIMA, M., MEGGIDO, N., NOMA, T. et YOSHISE, A. (1991). "A unified approach to interior point algorithms for linear complementarity problems." Berlin: Springer-Verlag, Heidelberg.

KOLSTAD, C.D. et LASDON, L. (1990). Derivative evaluation and computational experience with large bilevel mathematical programs. *Journal of Optimization Theory and Applications*, 65, 485-499.

KORNAJ, J. et LIPTAK, T. (1965). Two-level planning. *Econometrica*, 33, 141-169.

KYPARISIS, J. (1987). Sensitivity analysis framework for variational inequalities. *Mathematical Programming*, 38, 203-213.

LABBÈ, M., MARCOTTE, P. et SAVARD, G. (1998a). A bilevel model of taxation an its application to optimal highway pricing. *Management Science*, 44, 1608-1622.

LABBÈ, M., MARCOTTE, P. et SAVARD, G. (1998b). On class of bilevel problems. In di Pillo, D. and Gianessi, F., editor *Nonlinear Optimization and Applications*, 2, 1-24.

LAPORTE, M. et VIGNES, J. (1974). "Algorithmes numériques: analyse et mise en oeuvre. Arithmétique des Ordinateurs, Systèmes Linéaires." Éditions Technip.

LAVIGNE, D., LOULOU, R. et SAVARD, G. (2000). Pure competition regulated and Stackelberg equilibria: application to energy system of Quebec. *European Journal of Operational Research*, 125, 1-17.

LELENO, J.M. et SHERALI, H.D. (1992). A leader follower model and analysis for a two-stage network of oligopolies. *Annal of Operations Research*, 34, 37-72.

LIN, G.H. et FUKUSHIMA, M. (2003). New relaxation method for mathematical programs with complementarity constraints. *Journal of Optimization Theory and Applications*, 8:1, 81-116.

LITTLE, J.D.C., MURTY, K.G., SWEENEY, D.W. et KAREL, C. (1963). An algorithm for the traveling salesman problem. *J. OSRA*, 11.

LIU, G.S, HAN, J.Y. et ZHANG, J.Z. (2001). Exact penalty functions for convex bilevel programming problems. *Journal of Global Optimization*, 110:3, 621-643.

LIU, X. et SUN, J. (2004). Generalized stationary points and an interior-point method for mathematical programs with equilibrium constraints. *Mathematical Programming*, 101:1, 231-261.

LUO, Z. Q., PANG, J.S. et RALPH, D. (1996a). Exact penalization and stationary conditions of mathematical programs with equilibrium constraints. *Mathematical Programming*, 76, 19-76.

LUO, Z.Q., J.S. PANG et D. RALPH. (1996b). "Mathematical programs with nonlinear complementary constraints." Cambridge UK, University Press.

LUO, Z.Q., PANG, J.S., RALPH, D. et WU, S.Q. (1998). "Piecewise sequential quadratic programming for mathematical programs with nonlinear complementary constraints." Migdalas, A. et al., ed., Multilevel Optimization: Algorithms, Complexity and Applications. Kluwer Academic Publishers.

MANGASARIAN, O.L. et SOLODOV, M.V. (1993). Nonlinear complementarity as unconstrained and constrained minimization. *Mathematical Programming*, 62, 277-297.

MARCOTTE, P. (1985). A new algorithm for solving variational inequalities, with application to the traffic assignment problem. *Mathematical Programming*, 33, 277-297.

MARCOTTE, P. et DUSSAULT, J. P. (1989). A sequential linear programming algorithm for solving monotone variational inequalities. *SIAM Journal on Control and Optimization*, 27:6, 1260-1278.

MARCOTTE, P., WU, S. et CHEN, Y. (1993). A cutting-plane algorithm for linear bilevel programming problem. *Technical Report CRT-925, Centre de Recherche sur les Transports*.

MARCOTTE, P. et ZHU, D.L. (1996). Exact and inexact penalty methods for the generalized bilevel programming problem. *Mathematical Programming*, A 74, 141-157.

MARCOTTE, P., G. SAVARD et ZHU, D.L. (2001). A trust region algorithm for nonlinear bilevel programming. *Operations Research Letters*, 29, 171-179.

MARCOTTE, P. et SAVARD, G. (2001). "Bilevel programming applications." Dordrecht. encyclopedia of optimization: Kluwer Academic Publishers.

MARCOTTE, P. et SAVARD, G. (2005). "Bilevel programming : A combinatorial perspective, graph theory and combinatorial optimization." Eds. D. Avis, A. Hertz, O. Marcotte, Springer.

MASTROENI, G. (2003). Gap functions for equilibrium problems. *Journal of Global Optimization*, 27, 411-426.

MEHROTRA, S. (1992). On the implementation of a primal-dual interior point method. *SIAM J. Optimization*, 29, 575-601.

MIGDALAS, A. et P.M. PARDALOS. (1996). Editorial: hierarchical and bilevel programming. *Journal of Global Optimization (Historical Archive)*, 8:3, 209-215.

MINOUX, M. (1983). "Programmation mathématique: théorie et algorithmes Tome 1." Paris, Dunod Bordas et C.N.E.T, E.N.S.T.

MORÉ, J.J. (1996). Global methods for nonlinear complementarity problems. *Mathematics of Operations Research*, 21, 589-614.

MOSHIRVAZIRI, K., AMOUZEGAR, M.A. et JACOBSEN, S.E. (1996). Test problems construction for linear bilevel programming problems. *Journal of Global Optimization (Historical Archive)*, 8:3, 235-243.

MUU, L.D. et QUY, N.V. (2003). A global optimization method for solving convex quadratic bilevel programming problems. *Journal of Global Optimization*, 26, 199-219.

OUTRATA, J.V. (1994). On optimization problems with variational inequality constraints. *SIAM Journal on Optimization*, 4:2, 340-357.

OUTRATA, J.V. et ZOWE, J. (1995). A numerical approach to optimization with variational inequality constraints. *Mathematical Programming*, 68, 105-130.

OUTRATA, J.V., KOCVARA, J. et ZOWE, J. (1998). "Nonsmooth approach to optimization problems with equilibrium constraints." Encyclopedia of Optimization, Kluwer Academic Publishers, Dordrecht.

PANG, J.S. et FUKUSHIMA, M. (1999). Complementarity constraint qualification and simplified B-stationarity conditions for mathematical programming with equilibrium constraints. *Computational Optimization and Applications*, 13, 111-136.

PAPAVASSILOPOULOS, G. (1982). Algorithms for static stackelberg games with linear costs and polyhedral constraints. In *Proceedings of 21st IEEE Conference on Decision and Control*, 647-652.

PIRONNEAU, O. et POLAK, E. (1971). On the rate of convergence of certain method of centers. memo. *ERL-M296, University of California, Berkeley*.

PIRONNEAU, O. et POLAK, E. (1973). Rate of convergence of a class of methods of feasible directions. *SIAM Journal on Numerical Analysis*, 10:1, 161-174.

POLAK, E. (1971). "*Computational method in optimization*." Academic Press, New York.

QI, Y. et MAGNANTI, T.L. (1989). Sensitivity analysis for variational inequalities defined on polyhedral sets. *Mathematics of Operations Research*, 14, 410-432.

QI, L. et JIANG, H. (1997). Semismooth Karush-Kuhn-Tucker equations and convergence analysis of Newton and quasi-Newton methods for solving these equations. *Mathematics of Operations Research*, 22:2, 301-324.

ROBINSON, S.M. (1987). Local structure of feasible sets in nonlinear programming, part III: stability and sensitivity. *Mathematical Programming Study*, 30, 45-66.

ROSEN, J. B. (1961). The gradient projection method for nonlinear programming, part 1: linear constraints. *Journal S.I.A.M.*, 8, 181-217.

ROSS, S.A. (1973). The economic theory of agency: The principal's problem. *AER*, 63, 134-139.

SAKAROVITCH, M. (1984). "Optimisation combinatoire, méthodes mathématiques (Programmation discrètes)." Herman, Paris.

SAVARD, G. (1989). "Contribution à la programmation mathématique à deux niveaux." Thèse de doctorat en génie électrique, École Polytechnique de Montréal.

SAVARD, G. et GAUVIN, J. (1994). The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters*. 15. 265-272.

SCHEEL, H. et SCHOLTES, S. (2000). Mathematical programs with complementarity constraints: stationarity, optimality and sensitivity. *Mathematics of Operations Research*. 26:4. 1-22.

SCHOLTES, S. et STÖHR, M. (2001). How stringent is the linear independence assumption for mathematical programs with complementarity. *Mathematics of Operations Research*, 25:1, 851-863.

SCHUMACHER, J.M. (2004). Complementarity systems in optimization. *Mathematical Programming, Ser B*, 101, 263-295.

SIMAAN, M. et CRUZ, JB. (1973). On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11, 533-555.

STACKELBERG, H.V. (1952). "*The Theory of Market Economy*." Oxford University Press.

STILL, G. (2002). Linear bilevel problems: genericity results and efficient method for computing local minima. *Mathematical Methods of Operations Research*, 55, 383-400.

TAJI, K. et FUKUSHIMA, M. (1996). A new merit function and a successive quadratic programming algorithm for variational inequality problems. *SIAM Journal on Optimization*, 6:3, 704-713.

THOAI, N.V., YAMAMOTO, Y. et YOSHIE, A. (2005). Global optimization method for solving mathematical programs with linear complementarity constraints. *Journal of Optimization Theory and Applications*, 124:2, 467-490.

TOBIN, R.L. (1986). Sensitivity analysis for variational inequalities. *Journal of Optimization Theory and Applications*, 48:1, 191-204.

TOPKIS, D.M., J. et VEINOTT, A.F. (1967). On the convergence of some feasible directions algorithms for nonlinear programming. *SIAM Journal on Control*, 268-279.

TUY, H., MIGDLAS, A. et VERBRAND, P. (1993). A global optimization approach for the linear two-level program. *Journal of Global Optimization*, 3, 1-24.

VICENTE, L.N. et CALAMAI, P.H. (1994). Bilevel and multilevel programming. A bibliography review. *Journal of Global Optimization*, 5:3, 291-306.

VICENTE, L.N. et CALAMAI, P.H. (1994). Generating quadratic bilevel programming test problems. *ACM Transactions on Mathematical Software*, 20:1, 103-119.

VICENTE, L.N., SAVARD, G. et JÚDICE, J. (1994). Descent approach for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, 81, 379-399.

WHITE, D. et ANANDALINGAM, G. (1993). A penalty function approach for solving bilevel linear programs. *Journal of Global Optimization*, 3, 397-419.

WILDE, D. (1975). Monotonicity and dominance in optimal hydraulic cylinder design. *Journal of Engineering for Industry*, 97:4,13-26.

WOLFE, P. (1959). Simplex method for quadratic programming. *Econometrica*, 27.

XU, S. et BURKE, J.V. (1999). A polynomial time interior-point path following algorithm based on Chen-Harker-Kanzow smoothing techniques. *Mathematical Programming*, 86, 91-103.

XU, Z.K. (1999). Deriving the properties of linear bilevel programming via a penalty function approach. *Journal of Optimization Theory and Applications*, 103, 2441-456.

YAMASHITA, N. (1998). Properties of restricted NCP functions for nonlinear complementarity problems. *Journal of Optimization Theory and Applications*, 98:3, 701-717.

YE, J.J. (1999). Optimality conditions for optimization problems with complementarity constraints. *Journal on Optimization*, 9:2, 374-387.

YE, J.J. et YE, X.Y. (1997). Necessary optimality conditions for optimization problems with variational inequality constraints. *Mathematics of Operations Research*, 22:4, 977-996.

YE, J.J. et ZHU, D.L. (199). Optimality conditions for bilevel programming problems. *Journal on Optimization*, 33, 9-27.

YE, J.J. et ZHU, D.L. (1997). A note on optimality conditions for bilevel programming problems. *Optimization*, 39, 361-366.

ZANG, J.Z. et LIU, G.S. (2001). A new extreme point algorithm and its applications in PSQP algorithms for solving mathematical programs with linear complementarity constraints. *Journal of Global Optimization*, 345-361.

ZHU, D.L. et MARCOTTE, P (1994). An extended descent framework for monotone variational inequalities. *Journal of Optimization Theory and Applications*, 80, 353-369.

ZHU, D.L. et MARCOTTE, P. (1996). Co-coercivity and its role in the convergence of iterative schemes for solving variational inequalities. *SIAM Journal on Optimization*, 6:3, 714-726.

ZOUTENDIJK, G. (1960). "*Method of feasible directions.*" Amsterdam, Elsevier Publishing Company.