

Université de Montréal

**Étude des propriétés polyédrales du problème
de conception de réseaux multiproduits,
avec coût fixe et capacité**

par

Mervat Chouman

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

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en informatique et recherche opérationnelle

Avril, 2003

©Mervat Chouman, 2003



QA

76

U54

2003

v.034

Direction des bibliothèques

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée :

**Étude des propriétés polyédrales du problème
de conception de réseaux multiproduits,
avec coût fixe et capacité**

présentée par :
Mervat Chouman

a été évaluée par un jury composé des personnes suivantes :

Professeur Jacques Ferland	Président-rapporteur
Professeur Teodor Gabriel Crainic	Directeur de recherche
Professeur Bernard Gendron	Co-directeur de recherche
Professeur Patrice Marcotte	Membre du jury
Professeur Nelson Maculan	Examineur externe
Professeur Jean-Yves Potvin	Représentant de l'examineur externe
Professeur Jacques Ferland	Représentant du doyen de la FES

Thèse acceptée le : 25 août 2003

Sommaire

Cette thèse porte sur le problème de conception de réseaux multiproduits, avec coût fixe et capacité, ou PCMC. Ce dernier représente un cas particulier d'une grande classe de problèmes combinatoires, les problèmes de conception de réseaux. Cette classe s'applique à une vaste variété d'applications dans différents domaines, notamment le transport et les télécommunications. Sommairement, le PCMC consiste à concevoir un réseau permettant de transporter, à un coût minimal, plusieurs types de produits afin de satisfaire la demande. Dans cette thèse, nous nous intéressons aux méthodes polyédrales qui, de nos jours, semblent figurer parmi les méthodes les plus appropriées pour résoudre des problèmes combinatoires de grande taille. Ces méthodes visent essentiellement à dériver de meilleures formulations du problème moyennant l'ajout d'informations supplémentaires, représentées par des inégalités valides (ou IV). Nous distinguons deux difficultés majeures de ces méthodes. La première réside dans l'identification théorique des formulations mathématiques qui définissent des ensembles d'IV. À cette fin, une étude poussée des propriétés polyédrales du problème est nécessaire. La deuxième difficulté concerne le problème de génération des IV et leur ajout au modèle. Ainsi, cette thèse s'oriente dans cette direction de recherche et trouve sa motivation dans la volonté d'élaborer des méthodes efficaces de résolution pour résoudre des exemplaires de grande taille du PCMC.

Les principales contributions de cette recherche se résument comme suit. Nous présentons, dans un premier temps, le problème de conception de réseaux multiproduits, avec coût fixe et capacité et nous passons en revue les principaux travaux qui lui ont été dédiés. Nous présentons également une revue de la littérature portant sur les IV pertinentes au PCMC, qui ont été étudiées pour des problèmes apparentés à ce dernier. Cette revue constitue le premier jalon vers l'élaboration d'une étude poussée des propriétés polyédrales du PCMC. Nous démontrons la nécessité d'effectuer une telle étude en présentant des résultats expérimentaux utilisant les implantations générales des IV présentées dans un logiciel sophistiqué et nous soulignons quelques questions importantes donnant naissance à des voies de recherche.

Nous entreprenons une étude des propriétés polyédrales du PCMC. Nous adaptons au problème deux classes d'IV connues dans la littérature. Nous développons particulièrement des procédures innovatrices pour résoudre les problèmes de séparation et d'élévation séquentielle de ces deux classes d'IV. Nous identifions également de nouvelles familles d'IV. Notons que toutes les IV étudiées sont valides pour la structure induite par des coupes du réseau. Dans ce contexte, nous développons une procédure de génération de coupes du réseau. Nous incorporons ces IV dans une méthode de coupes qui, à la lumière des résultats expérimentaux, s'avère efficace à obtenir de meilleures reformulations du problème en un temps raisonnable.

Nous proposons une méthode de résolution exacte pour le PCMC. Il s'agit d'un algorithme de "Branch-and-Cut", qui intègre une méthode de coupes à un algorithme de séparation et d'évaluation progressive. Dans cet algorithme, nous proposons une nouvelle stratégie de branchement basée sur des décisions polytomiques. Nous proposons également une heuristique pour le calcul des bornes supérieures sur l'objectif. Des résultats expérimentaux montrent un comportement prometteur de l'algorithme. En effet, l'algorithme offre des solutions réalisables de bonne qualité pour des instances de grande taille du problème.

Finalement, nous dégagons quelques avenues de recherche futures visant, dans un premier temps, l'amélioration de la performance de l'algorithme de Branch-and-Cut proposé. Ces recherches futures concernent le développement des stratégies mixtes de branchement ainsi que des heuristiques sophistiquées de recherche locale pour le calcul des bornes supérieures et pour l'identification des coupes du réseau. Nous suggérons également des avenues de recherche s'ouvrant sur des concepts plus généralisés. Dans ce qui suit, nous citons quelques-unes.

- L'étude de la dimension des nouvelles familles d'inégalités et la possibilité de les améliorer moyennant la technique d'élévation séquentielle.
- Une meilleure compréhension des propriétés polyédrales du PCMC.
- L'utilité des familles d'inégalités étudiées dans cette thèse à améliorer d'autres problèmes de conception de réseaux.
- Le développement d'une approche algorithmique qui intègre les coupes violées dans le cadre d'une méthode de relaxation lagrangienne combinée avec un al-

Mots-clés : Conception de réseaux multiproduits avec coût fixe et capacité, méthodes polyédrales, propriétés polyédrales, inégalités valides, Branch-and-Cut, méthode de coupes.

Abstract

This thesis addresses the multicommodity capacitated fixed charge network design problem, or MCND. This problem represents a particular case of a wide class of combinatorial problems, the network design problems. This class is used to model a wide variety of applications, most notably in the fields of transportation and telecommunications. Briefly, the MCND aims to design a network which will be able to satisfy the demand requirements of several commodities at minimum cost. In this thesis, we are interested in the polyhedral approaches which are considered, actually, as the most common tools to solve large scale combinatorial problems. The polyhedral approaches aim essentially to derive strengthened formulations of the problem by adding additional informations in the form of valid inequalities (or VI). We distinguish two major challenges of these approaches. The first consists of the theoretical identification of mathematical formulations that represent sets of VI. To this aim, a profound study of the polyhedral structure of the MCND is necessary. The second challenge consists in the generation and the addition of these VI to the model. Thus, this thesis points towards this research direction and is motivated by the elaboration of efficient resolution approaches to solve large scale instances of the MCND.

The main contributions of this research are summarized next. We present first the multicommodity capacitated fixed charge network design problem and a relevant literature survey on the principal works that addressed this problem. We also present a literature review of relevant VI for the MCND that have been studied for related problems. This review constitutes the first step towards the elaboration of profound polyhedral studies of the MCND. We show the necessity of such studies by presenting experimental results on the general implementations of some of the presented VI in a state-of-the-art commercial software and we point out several important questions that introduce some research avenues.

We investigate the polyhedral structure of the MCND. We adapt to the problem two families of VI proposed in the literature. We develop efficient separation heuristics and lifting procedures for these families of VI. We identify new families of VI that, to the best of our knowledge, are introduced for the first time. Note that, all VI studied in

this paper are of cutset type structure. In this context, we develop heuristic procedures to identify cutsets of the network. We incorporate these VI within a cutting-plane method which, in the light of the experimental results, seems to be efficient to improve the formulation of the problem using a reasonable computational time.

We propose an exact resolution method for the MCND. It is a Branch-and-Cut approach which incorporates a cutting-plane procedure into a Branch-and-Bound algorithm. In this approach, we propose a new branching strategy based on a polytomic decisions. We also propose a heuristic approach to compute upper bounds on the objective. The experimental results show a promising behavior of the algorithm. In fact, the algorithm offers feasible solutions of good quality for large scale instances of the problem.

Finally, we suggest some research avenues that aim first to improve the performance of the proposed Branch-and-Cut algorithm. These future researches consist on the development of mixed branching strategies and more sophisticated local search heuristics to compute upper bounds and to identify cutsets of the network. We propose also some research avenues dedicated to more general concepts. In the following, we summarize some of these avenues.

- The theoretical study of the dimension of the new families of VI and the possibility to improve them using the lifting technic.
- Better understanding of the polyhedral structure of the MCND.
- The usefulness of the VI studied in this thesis to improve other network design problems.
- The development of an algorithmic approach that uses the violated cuts identified within a Lagrangian relaxation method combined with a Branch-and-Bound algorithm.

Key words : multicommodity capacitated fixed charge network design, polyhedral approaches, polyhedral studies, valid inequalities, Branch-and-Cut, cutting-plane.

Table des matières

Sommaire	i
Abstract	iv
Table des matières	vi
Liste des figures	viii
Liste des tableaux	ix
Remerciements	xiii
Dédicace	xiv
1 Introduction	1
2 Revue des inégalités valides pertinentes aux problèmes de conception de réseaux	5
1 Introduction	8
2 Le problème de conception de réseaux multiproduits avec coût fixe et capacité (PCMC)	10
2.1 Formulation du PCMC	11
2.2 Méthodes de résolution pour le PCMC	13
2.3 Modèles apparentés au PCMC	16
3 Approches polyédrales	21
4 Revue des inégalités valides pertinentes au PCMC	24
4.1 Inégalités de couverture	24
4.2 Inégalités de couverture de flot	31

4.3	Inégalités basées sur des coupes	35
5	Résultats expérimentaux	40
6	Conclusion	45
3	A cutting-plane algorithm based on cutset inequalities for multicom-	
	modity capacitated fixed charge network design	58
1	Introduction	61
2	Formulation and Basic Valid Inequalities	64
3	Cover Inequalities	67
	3.1 Separation	68
	3.2 Lifting	69
4	Minimum Cardinality Inequalities	73
	4.1 Lifting	74
5	Network Cutset Inequalities	77
	5.1 Separation	80
6	Algorithm Description	82
	6.1 Cutset Generation Algorithm	82
	6.2 Cutting-Plane Algorithm	86
7	Computational Results	89
8	Conclusion	95
4	A Branch-and-Cut algorithm for multicommodity capacitated fixed	
	charge network design	120
1	Introduction	123
2	Problem Formulation and Algorithmic Framework	126
3	Cut Generation	129
	3.1 Strong Inequalities	130
	3.2 Cutset Type Inequalities	130
	3.3 Cutting-Plane Algorithm	134
4	Branching	137
5	Upper Bound Heuristic	139
6	Computational Results	143
7	Conclusion	150

5 Conclusion	161
1 Principales contributions	161
2 Nouvelles avenues de recherche	163

Table des figures

Chapitre 2	8
1 L'enveloppe convexe $\text{conv}(P)$ et une relaxation R_P	22
2 Structure induite par le sac alpin.	25
3 Problème à un seul noeud avec uniquement des arcs entrants.	31
4 Problème à un seul noeud avec des arcs entrants et sortants.	33
5 La coupe (X, \bar{X})	36
Chapitre 3	61
1 Cutset structure	66
2 Network cutset structure	77
3 Add-drop example	84
4 Add-drop example	85
Chapitre 4	123
1 Network cutset structure	133

Liste des tableaux

Chapitre 2	8
1 Résultats d'instances difficiles.	42
2 Comparaison du B&B et du B&C.	44
3 Distribution des solutions réalisables.	44
 Chapitre 3	 61
1 Detailed results for 13 selected instances	92
2 Gap and improvement distribution according to problem characteristics, C problems	93
3 CPU reduction and gap improvement according to problem dimensions, large C problems	93
4 Gap distribution according to fixed cost and capacity level, R problems	94
5 CPU reduction and gap improvement according to problem dimensions, R problems	94
6 Distribution of relative improvement for the 68 unsolved problems . .	95
7 Computational Results Simple and Advanced Cutting-Plane-NCI, C problems	106
8 Computational Results Simple and Advanced Cutting-Plane-NCI, C problems	107
9 Computational Results Simple and Advanced Cutting-Plane-NCI, R10	108
10 Computational Results Simple and Advanced Cutting-Plane-NCI, R11	108
11 Computational Results Simple and Advanced Cutting-Plane-NCI, R12	109
12 Computational Results Simple and Advanced Cutting-Plane-NCI, R13	109
13 Computational Results Simple and Advanced Cutting-Plane-NCI, R14	110

14	Computational Results Simple and Advanced Cutting-Plane-NCI, R15	110
15	Computational Results Simple and Advanced Cutting-Plane-NCI, R16	111
16	Computational Results Simple and Advanced Cutting-Plane-NCI, R17	111
17	Computational Results Simple and Advanced Cutting-Plane-NCI, R18	112
18	Computational Results Simple and Advanced Cutting-Plane-GNCI, C problems	113
19	Computational Results Simple and Advanced Cutting-Plane-GNCI, C problems	114
20	Computational Results Simple and Advanced Cutting-Plane-GNCI, R10	115
21	Computational Results Simple and Advanced Cutting-Plane-GNCI, R11	115
22	Computational Results Simple and Advanced Cutting-Plane-GNCI, R12	116
23	Computational Results Simple and Advanced Cutting-Plane-GNCI, R13	116
24	Computational Results Simple and Advanced Cutting-Plane-GNCI, R14	117
25	Computational Results Simple and Advanced Cutting-Plane-GNCI, R15	117
26	Computational Results Simple and Advanced Cutting-Plane-GNCI, R16	118
27	Computational Results Simple and Advanced Cutting-Plane-GNCI, R17	118
28	Computational Results Simple and Advanced Cutting-Plane-GNCI, R18	119
Chapitre 4		123
1	Computational Results, C problems	146
2	Computational Results, C problems	147
3	Gap distribution according to problem characteristics, C problems . .	148
4	Gap distribution according to fixed cost and capacity level, R problems	148
5	Gap distribution according to problem dimension, R problems	149
6	Distribution of relative improvement	149
7	Computational Results Simple Cutting-Plane R10 : 20,100,40 . . .	156
8	Computational Results Simple Cutting-Plane R11 : 20,100,100 . .	156
9	Computational Results Simple Cutting-Plane R12 : 20,100,200 . .	157
10	Computational Results Simple Cutting-Plane R13 : 20,200,40 . . .	157
11	Computational Results Simple Cutting-Plane R14 : 20,200,100 . .	158
12	Computational Results Simple Cutting-Plane R15 : 20,200,200 . .	158

13	Computational Results Simple Cutting-Plane R16 : 20,300,40 . . .	159
14	Computational Results Simple Cutting-Plane R17 : 20,300,100 . . .	159
15	Computational Results Simple Cutting-Plane R18 : 20,300,200 . . .	160

Remerciements

Je voudrais avant tout remercier mes directeurs de recherche, les professeurs Teodor Gabriel Crainic et Bernard Gendron, pour leurs encouragements et pour la confiance qu'ils m'ont témoignée durant toutes ces années de travail. Je désire également les remercier pour leur soutien financier qu'ils m'ont accordé tout au long de cette thèse. Je voudrais exprimer ma profonde estime au professeur Crainic pour son appui constant et surtout pour son inestimable expérience qui m'a été très bénéfique. Je tiens aussi à exprimer ma profonde gratitude au professeur Gendron pour sa disponibilité, son implication et ses remarques pertinentes qu'il m'a prodiguées tout au long de cette recherche. J'espère sincèrement que cette thèse n'est autre que le début de collaborations futures tout aussi bénéfiques.

Je remercie les professeurs, chercheurs, professionnels et étudiants du département d'informatique et de recherche opérationnelle et du Centre de Recherche sur les Transports pour avoir contribué à enrichir mes connaissances. Je remercie particulièrement François Gertin et Geneviève Hernu pour leurs aides judicieuses en programmation. Je remercie aussi les responsables de ressources informatiques et l'équipe de secrétariat pour leurs aides, et leurs nombreuses services rendus.

Je voudrais remercier particulièrement mes amis pour leur soutien moral et leur présence, surtout dans les moments difficiles.

Finalement, tout mon respect et ma gratitude vont à mes parents qui m'ont soutenu avec leurs encouragements tout au long des années et ce malgré les distances qui nous séparent. Je voudrais exprimer ma profonde gratitude à mon mari, Ghassan, et mes deux trésors, Aya et Mira, pour leur présence et leur soutien tout au long de cette thèse. Je tiens à les remercier plus particulièrement de m'avoir supporter durant les moments difficiles qui se sont, d'ailleurs, prolongés durant les derniers mois. Leur patience et leur amour m'ont permis de mener ce travail jusqu'à la fin.

À mes parents

À ma famille

Chapitre 1

Introduction

Une grande variété de situations réelles exigeant la planification ou la construction des infrastructures, que ce soit des réseaux urbains, des réseaux de services, des voies ferrées, des réseaux de télécommunication, des réseaux électriques, des réseaux de canalisation d'eau ou de gaz, etc, sont représentées selon des modèles de conception de réseaux. Cette thèse porte spécifiquement sur le modèle de conception de réseaux multiproduits, avec coût fixe et capacité (ou PCMC). Ce modèle constitue un cas particulier des modèles de conception de réseaux, qui renferme néanmoins une vaste variété d'applications importantes dans différents domaines, notamment le transport et les télécommunications. Sommairement, le problème de conception de réseau consiste à concevoir un réseau à un coût minimal qui assurera, selon le contexte, le transport des biens ou des gens, ou encore l'acheminement ou l'échange des ressources, afin de satisfaire les demandes. Dans ce type de problèmes, outre les coûts opérationnels proportionnels à la quantité transportée, des coûts fixes associés à la conception du réseau, par exemple des coûts de construction de routes, sont aussi présents.

Les modèles de conception de réseaux généraux entrent dans le paradigme de la programmation entière-mixte et sont NP-difficiles. Outre la complexité théorique prouvée, la difficulté à résoudre ces problèmes revient, en partie, à leur nature fortement combinatoire, alliée à la très grande taille des applications réelles. De plus, dans la plupart des cas, ces modèles renferment un flot à plusieurs types de produits et les ressources disponibles ou à concevoir sont munies de capacité limitée. La présence de capacités rend le problème considérablement plus difficile à résoudre. En effet, le

problème de l'arbre sous-tendant minimal, qui représente un cas très simplifié des modèles de conception de réseaux, en est la preuve. La version sans capacité, dans un réseau non orienté, de l'arbre sous-tendant minimal est très facile à résoudre avec des algorithmes voraces simples, alors que la version avec capacité est significativement plus difficile.

En dépit de la complexité du PCMC, certains travaux intéressants ont été dédiés à l'étude et à l'analyse, à la fois théorique et expérimentale, de différentes relaxations du problème. Ces études ont démontré que plusieurs relaxations possibles du problème fournissent théoriquement la même borne et que l'approximation du modèle entier-mixte par sa relaxation linéaire est généralement médiocre. Étant donné la relative mauvaise qualité des bornes inférieures issues de la relaxation linéaire du problème, les méthodes de résolution basées sur les relaxations linéaires, qui sont généralement très utilisées dans le cas des modèles entier-mixte, ne sont plus intéressantes lorsqu'appliquées directement sur la formulation du problème. Une meilleure compréhension de la structure polyédrale du domaine, visant ainsi l'identification de meilleures reformulations du problème, paraît d'une importance cruciale.

Cette conclusion motive essentiellement notre intérêt porté vers les méthodes polyédrales dont le principe de base est d'ajouter des informations supplémentaires au modèle dans le but d'améliorer sa formulation. Plus précisément, l'amélioration de la formulation se traduit par l'ajout des contraintes linéaires (ou inégalités valides, IV), possiblement redondantes dans la formulation originale, qui permettent, néanmoins, d'obtenir de meilleures bornes obtenues à partir de ses relaxations. Nous distinguons deux difficultés majeures de ces méthodes. La première réside dans l'identification théorique des formulations mathématiques qui définissent des ensembles d'IV. À cette fin, une étude poussée des propriétés polyédrales du problème est nécessaire. La deuxième difficulté, qui est d'ordre algorithmique, concerne le problème de génération des IV et de leur ajout au modèle. Notons que plusieurs études polyédrales ont été effectuées pour des problèmes de conception de réseaux apparentés au PCMC. Cependant, ce dernier n'a jamais fait l'objet d'une telle étude. Notons également qu'une inégalité valide pour une relaxation du problème est valide pour le problème lui-même.

Ce fait nourrit notre intérêt porté aux propriétés et études polyédrales des problèmes apparentés au PCMC.

Appuyé par la nécessité d'une meilleure reformulation du problème de conception de réseaux multiproduits, avec coût fixe et capacité, notre thèse s'oriente dans cette direction de recherche et puise sa motivation dans la volonté d'élaborer des méthodes efficaces de résolution pour résoudre des instances de grande taille du problème. Dans ce contexte, nous nous intéressons, dans un premier temps, à effectuer une revue et une synthèse de la littérature portant sur les propriétés polyédrales des modèles qui constituent des cas particulier du PCMC, ou encore qui contiennent des structures pertinentes à ce dernier. Cette revue sert à maîtriser les techniques utilisées dans le développement des inégalités valides et constitue le premier jalon vers l'élaboration d'une étude plus détaillée des propriétés polyédrales du PCMC. Suite à cette revue, nous étudions et analysons des IV existantes dans la littérature et qui sont pertinentes au PCMC. Plus précisément, nous proposons des heuristiques innovatrices de séparation et d'élévation séquentielle qui visent à améliorer la qualité des IV générées. Nous identifions également de nouveaux types d'IV. Ensuite, nous évaluons l'efficacité des IV étudiées et identifiées en les intégrant dans le cadre d'une méthode de coupes. La méthode proposée vise essentiellement à calculer, dans un temps raisonnable, des bornes de meilleure qualité que celles obtenues de la relaxation linéaire du PCMC. Finalement, nous développons une méthode exacte de résolution, un algorithme de "Branch-and-Cut", qui consiste à combiner la méthode de coupes proposée et l'algorithme de séparation et d'évaluation progressive. Dans cet algorithme, nous utilisons plusieurs sources d'informations dans l'élaboration d'une nouvelle stratégie de branchement, ainsi que dans le développement d'une heuristique de calcul de bornes supérieures sur l'objectif. Bien que nous suggérons des recherches futures visant l'amélioration de la performance de l'algorithme de Branch-and-Cut proposé, il constitue néanmoins un pas considérable vers une résolution efficace des problèmes de conception de réseaux de grande taille.

L'organisation de cette thèse reflète cette orientation. Outre le présent chapitre, le corps de la thèse se compose de quatre chapitres, chacun pouvant être lu indépendam-

ment des autres. Le chapitre 2 correspond à un article qui est à paraître dans INFOR, le chapitre 3 correspond à un article qui est soumis à NETWORKS et le chapitre 4 sera soumis dans un très proche délai à une revue scientifique. Ainsi, la bibliographie relative à chacun d'eux se trouve à la fin du chapitre. Le chapitre 2 introduit le modèle de conception de réseaux multiproduits, avec coût fixe et capacité et présente une revue de la littérature portant sur les méthodes polyédrales et sur les familles d'inégalités valides pertinentes au problème, qui ont été étudiées pour des problèmes apparentés. Le chapitre 3 présente un algorithme de génération de coupes qui intègre des inégalités valides basées sur des coupes du réseau. Le chapitre 4 présente un algorithme de Branch-and-Cut, qui utilise une adaptation de la procédure de génération de coupes, présentée dans le chapitre 3, à chaque noeud de l'arbre d'énumération. Finalement, dans le chapitre 5, nous soulignons les principales contributions de cette thèse et nous dégageons des avenues de recherche pertinentes à explorer.

Chapitre 2

Revue des inégalités valides
pertinentes aux problèmes de
conception de réseaux

Revue des inégalités valides pertinentes aux problèmes de conception de réseaux

Mervat Chouman, Teodor Gabriel Crainic
et Bernard Gendron

Résumé

L'objectif de cet article est de présenter une revue de la littérature portant sur les inégalités valides pertinentes au problème de conception de réseaux multiproduits avec coût fixe et capacité. Nous passons d'abord en revue les principaux travaux portant sur des méthodes de relaxations et des heuristiques. Nous constatons que ces méthodes seules ne sont pas suffisantes pour résoudre des exemplaires de grande taille du problème. Une combinaison de ces méthodes avec des approches polyédrales apparaît pourtant prometteuse. Par la suite, nous présentons trois grandes classes d'inégalités valides pertinentes au problème, qui ont été étudiées pour des problèmes apparentés. Nous présentons également des résultats expérimentaux utilisant les implantations générales de ces inégalités dans un logiciel sophistiqué. Ces résultats montrent clairement la nécessité d'entreprendre une étude approfondie des propriétés polyédrales du problème afin de résoudre efficacement des exemplaires de grande taille.

Mots-clés : Inégalités valides, conception de réseaux multiproduits avec capacité, propriétés polyédrales, méthodes de relaxations, heuristiques, méthodes polyédrales.

Abstract

The objective of this paper is to present a survey of relevant valid inequalities for the multicommodity capacitated fixed-charge network design problem. We present first a review of the relevant literature on relaxations and heuristic approaches. We observe that these approaches alone are not sufficient to solve large-scale instances of the problem. However, a combination of these approaches and polyhedral methods appears very promising. Then, we present three classes of relevant valid inequalities

for the problem that have been studied for related problems. We present as well experimental results on the general implementations of some of these inequalities in a state-of-the-art commercial software. These experiments point towards the necessity of profound polyhedral studies in order to solve efficiently large-scale instances of the problem.

Key words : Valid inequalities, multicommodity capacitated network design, polyhedral studies, relaxations, heuristic approaches, polyhedral methods.

1 Introduction

Les modèles de conception de réseaux sont largement utilisés pour représenter une grande variété d'applications dans différents domaines, notamment le transport, les télécommunications, la logistique et les systèmes de production-distribution (Magnanti et Wong 1984, Minoux 1989). Dans le domaine du transport, en particulier, les modèles de conception de réseaux couvrent la hiérarchie complète de la planification, que ce soit au niveau stratégique, tactique ou opérationnel (Crainic et Laporte 1997). Sommairement, le problème de conception de réseaux consiste à construire un réseau permettant de transporter à un coût minimal la demande requise entre des paires origines-destinations. Dépendamment de l'application, afin de satisfaire la demande, la construction du réseau se fait soit en choisissant un sous-ensemble d'arcs (ou de noeuds) dans le graphe sous-jacent au réseau, soit en installant des capacités additionnelles sur les arcs. À chaque décision (choix ou installation) est associée un coût fixe.

Les modèles de conception de réseaux sont généralement représentés par des formulations de programmation linéaire en nombres entiers incluant des variables continues ("mixed integer programming", ou MIP) représentant ainsi à la fois les décisions de choix ou d'installation et le transport de la demande à travers le réseau. Ils sont NP-difficiles. De plus, outre leur caractère fortement combinatoire, ces modèles sont souvent de très grande taille. Dans la plupart des cas, ces modèles incluent également un flot à plusieurs types de produits et les arcs sont munis de capacités, augmentant ainsi le niveau de complexité du problème.

Typiquement, les modèles MIP sont résolus d'une manière exacte par des algorithmes de programmation linéaire (tel le simplexe) combinés avec des méthodes de séparation et évaluation progressive ("branch-and-bound", ou B&B). Toutefois, étant donné la complexité de ces modèles, seulement des instances de petite taille peuvent être résolues efficacement en un temps raisonnable. Étant donné que l'efficacité des méthodes de B&B repose sur la possibilité de générer efficacement des bornes de qualité sur la valeur optimale, les chercheurs se sont intéressés aux méthodes polyédrales,

qui permettent d'améliorer les bornes calculées par les algorithmes de programmation linéaire. La philosophie de ces méthodes est d'ajouter de l'information au modèle dans le but de l'améliorer. L'amélioration du modèle se traduit par l'ajout de contraintes linéaires (*inégalités valides*, ou IV) qui sont redondantes dans le modèle original, mais qui permettent d'améliorer les bornes obtenues à partir de ses relaxations.

La difficulté majeure de l'approche polyédrale réside dans l'identification des ensembles (ou familles) d'IV. À cette fin, une étude poussée des propriétés polyédrales du modèle est nécessaire. Au cours des vingt dernières années, plusieurs études polyédrales ont été effectuées sur des modèles difficiles connus dans la littérature; ainsi, des résultats théoriques et des familles d'IV pour des problèmes spécifiques ont été proposés dans plusieurs articles. Par la suite, ces familles d'IV ont été utilisées pour développer des algorithmes pour le problème initial. Elles ont été également adaptées pour d'autres formulations.

Le but principal de cet article est de présenter une revue de trois grandes familles d'IV proposées dans la littérature, et qui sont pertinentes au *problème de conception de réseaux multiproduits avec coût fixe et capacité* (ou PCMC, que nous présentons à la section 2.1), dont la formulation permet de représenter un grand nombre d'applications. Ainsi, plusieurs familles d'IV ont été étudiées pour des structures particulières que nous pouvons identifier à des relaxations du PCMC. La pertinence de ces familles d'IV pour le PCMC est due au fait qu'une inégalité valide pour une relaxation du problème est également valide pour le problème lui-même. Par ailleurs, il y a eu plusieurs études polyédrales portant sur des problèmes de conception de réseaux apparentés au PCMC. Plus précisément, nous traitons des problèmes de localisation et des problèmes d'installation sur les arcs, ou "network loading" (que nous présentons aux sections 2.3.1 et 2.3.2). De ces contributions, nous dégageons les trois grandes familles d'IV suivantes (que nous étudions à la section 4) : les inégalités de couverture, les inégalités de couverture de flot et les inégalités basées sur des coupes. Mentionnons qu'à notre connaissance, aucune étude des propriétés polyédrales du PCMC n'a été publiée à ce jour. Le présent article, qui se veut essentiellement une revue des propriétés polyédrales des modèles apparentés au PCMC, constitue donc un pre-

mier jalon vers l'élaboration d'une étude plus détaillée des propriétés polyédrales du PCMC.

Remarquons que la plupart des IV présentées dans cet article ont été incorporées dans des logiciels commerciaux de MIP. Dans ce contexte, il est naturel de se poser la question suivante : les implantations générales des IV en question, proposées par ces logiciels commerciaux, sont-elles suffisantes pour résoudre efficacement des exemplaires de grande taille du PCMC ? Nous illustrons par des résultats numériques obtenus avec un logiciel de MIP (CPLEX 2000) que les implantations générales des IV ne semblent pas suffisantes pour résoudre des exemplaires de grande taille du PCMC.

Cet article est organisé comme suit. Dans la section 2, nous présentons une formulation du PCMC, ainsi que les différentes méthodes utilisées pour le résoudre, et nous décrivons des cas particuliers pertinents à notre étude, soit les problèmes de localisation et les problèmes d'installation sur les arcs. La section 3 est consacrée à la présentation des approches polyédrales. Des familles d'IV étudiées dans la littérature sont présentées à la section 4. Dans la section 5, nous montrons des résultats expérimentaux sur les implantations générales de certaines familles d'IV par le logiciel CPLEX. Nous terminons cet article par une conclusion qui soulève quelques questions importantes donnant naissance à des voies de recherche.

2 Le problème de conception de réseaux multiproduits avec coût fixe et capacité (PCMC)

Dans cette section, nous décrivons d'abord la formulation du PCMC, puis nous présentons une revue de la littérature portant sur les méthodes de résolution pour le problème, et enfin nous présentons deux classes de modèles apparentés au PCMC, soit les modèles de localisation et les modèles d'installation sur les arcs, pour lesquels plusieurs études portant sur les IV ont été publiées à ce jour.

2.1 Formulation du PCMC

Le PCMC est défini sur un graphe $G = (N, A)$, où N est l'ensemble des noeuds et A est l'ensemble des arcs, dans lequel circulent plusieurs produits. Pour simplifier la présentation, on considère que chaque produit correspond à une paire origine-destination. Dénotons P l'ensemble des produits, ou paires origine-destination. Pour chaque produit $p \in P$ nous distinguons son origine $O(p)$, sa destination $D(p)$ et la quantité $d^p > 0$ à transporter de $O(p)$ à $D(p)$. Tout noeud différent de l'origine et de la destination est un noeud de transfert : il reçoit et transmet la même quantité de flot du produit p .

À chaque arc $(i, j) \in A$ sont associées plusieurs quantités :

- u_{ij} : capacité globale sur la quantité de flot pouvant circuler sur l'arc ;
- b_{ij}^p : capacité partielle sur la quantité de flot de produit p pouvant circuler sur l'arc (dans beaucoup d'applications, ces capacités sont redondantes, et on pose alors $b_{ij}^p = \min\{u_{ij}, d^p\}$);
- f_{ij} : coût fixe qui est encouru dès qu'un flot d'un produit quelconque traverse l'arc (i, j) ;
- c_{ij}^p : coût de transport par unité de flot de produit p traversant l'arc.

Le problème consiste à satisfaire les demandes de tous les produits de façon à minimiser la somme des coûts de transport et des coûts fixes, tout en respectant les capacités. Nous introduisons deux types de variables afin de modéliser le problème :

- x_{ij}^p : des variables continues, qui représentent la quantité de flot de produit p circulant sur l'arc (i, j) ;
- y_{ij} : des variables binaires, qui représentent les décisions de conception pour chaque arc (i, j) ; ainsi

$$y_{ij} = \begin{cases} 0, & \text{si } x_{ij}^p = 0, \forall p \in P, \\ 1, & \text{sinon.} \end{cases}$$

Le PCMC est alors formulé comme suit (en introduisant la notation $N^+(i) = \{j \in N \mid (i, j) \in A\}$ et $N^-(i) = \{j \in N \mid (j, i) \in A\}$) :

$$\min \sum_{p \in P} \sum_{(i,j) \in A} c_{ij}^p x_{ij}^p + \sum_{(i,j) \in A} f_{ij} y_{ij}, \quad (1)$$

sujet à :

$$\sum_{j \in N^+(i)} x_{ij}^p - \sum_{j \in N^-(i)} x_{ji}^p = \begin{cases} d^p & \text{si } i = O(p), \\ -d^p & \text{si } i = D(p), \\ 0 & \text{sinon,} \end{cases} \quad \forall i \in N, \forall p \in P, \quad (2)$$

$$\sum_{p \in P} x_{ij}^p \leq u_{ij} y_{ij}, \quad \forall (i, j) \in A, \quad (3)$$

$$x_{ij}^p \leq b_{ij}^p y_{ij}, \quad \forall (i, j) \in A, \forall p \in P, \quad (4)$$

$$x_{ij}^p \geq 0, \quad \forall (i, j) \in A, \forall p \in P, \quad (5)$$

$$0 \leq y_{ij} \leq 1, \quad \forall (i, j) \in A, \quad (6)$$

$$y_{ij} \text{ entier, } \quad \forall (i, j) \in A. \quad (7)$$

Les contraintes (3) et (4) sont identifiées comme étant les *contraintes liantes*, parce qu'elles lient entre elles les variables continues et les variables entières pour assurer que le flot soit nul sur l'arc (i, j) dans le cas où $y_{ij} = 0$, c'est-à-dire lorsque l'arc (i, j) n'est pas présent dans la structure du réseau. De plus, les contraintes (3) lient les produits entre eux, assurant que la somme des flots de tous les produits sur chaque arc ne dépasse pas la capacité globale de l'arc. Notons que, dans le cas où les capacités partielles sont redondantes, les contraintes (4) sont inutiles, et peuvent donc être éliminées. Le modèle résultant est appelé *formulation faible*. Toutefois, à cause de la présence des variables binaires au sein des contraintes (4), ces dernières ne sont pas redondantes pour la relaxation linéaire, ce qui permet d'améliorer la qualité des bornes issues de la relaxation linéaire. Par conséquent, le modèle obtenu en conservant les contraintes (4) est appelé *formulation forte*.

Le PCMC est NP-difficile (Krarup et Pruzan 1983 ont démontré que le problème de localisation sans capacité, qui représente un cas particulier du PCMC, est un

problème NP-difficile). Outre la complexité théorique prouvée, la difficulté à résoudre ce problème revient, en partie, à sa nature fortement combinatoire, alliée à la très grande taille des applications réelles. De plus, les contraintes (3) lient entre eux les produits, rendant ainsi le problème non décomposable par produit. Lorsque les capacités globales (3) sont très contraignantes, une autre difficulté s'impose : l'approximation du modèle MIP par sa relaxation linéaire est généralement médiocre. Ceci est dû, en particulier, à la mauvaise représentation des liens entre les capacités et les coûts fixes dans l'objectif de la relaxation linéaire. De plus, le problème de multiflot à coût minimum avec contraintes additionnelles, obtenu de la relaxation linéaire, est généralement difficile à résoudre par des méthodes traditionnelles en programmation linéaire, particulièrement lorsque le nombre de produits est élevé : le problème devient extrêmement dégénéré, lors de l'application de la méthode du simplexe (Crainic, Frangioni, Gendron 2001). Étant donné la relative mauvaise qualité des bornes inférieures issues de la relaxation linéaire du problème avec capacité, les méthodes de résolution basées sur les relaxations linéaires, qui sont généralement très utilisées dans le cas des modèles MIP, ne sont plus intéressantes lorsqu'appliquées directement sur cette formulation. Pour cette raison, une amélioration de la formulation via l'ajout d'IV paraît importante pour résoudre le problème.

2.2 Méthodes de résolution pour le PCMC

Avant de passer en revue les méthodes de résolution pour le PCMC, il convient de souligner que le cas particulier du problème où il n'y a aucune capacité est beaucoup plus simple à résoudre. Dans le cas sans capacité, on peut définir les formulations faible et forte de la manière suivante. La formulation faible est obtenue en posant $u_{ij} = \sum_{p \in P} d^p$ et en éliminant les contraintes (4), alors que la formulation forte revient à poser $b_{ij}^p = d^p$ et à éliminer les contraintes (3). Le problème sans capacité (PCMS) a inspiré quelques travaux portant sur le développement d'algorithmes efficaces de résolution (voir la bibliographie de Balakrishnan, Magnanti et Mirchandani 1997). Le succès de certains algorithmes est dû au fait que la relaxation linéaire forte du PCMS fournit une bonne approximation du problème. Il a été démontré que le

polytope de la relaxation linéaire possède la propriété de quasi intégralité (Hellstrand, Larsson et Migdalas 1992), dans le sens où chaque arête de l'enveloppe convexe des points entiers est aussi une arête du polytope lui-même. Cette propriété ne tient pas pour la formulation faible du problème.

Parmi les méthodes de résolution efficaces qui ont été développées pour le problème sans capacité, on retrouve la procédure duale-ascendante de Balakrishnan, Magnanti et Wong (1989), et l'heuristique lagrangienne proposée par Holmberg et Hellstrand (1998). La procédure de Balakrishnan, Magnanti et Wong (1989) est une généralisation de la procédure duale ascendante, DUALOC, conçue par Erlenkotter (1978) pour le problème de localisation sans capacité, que nous allons présenter dans la section suivante. Elle consiste à ajuster les valeurs des multiplicateurs dans le but d'augmenter la valeur de l'objectif du dual de la relaxation forte du problème. Holmberg et Hellstrand (1998) ont proposé de résoudre une relaxation lagrangienne (par rapport aux contraintes de conservation de flot) avec une heuristique primale pour trouver une solution réalisable du problème. Les résultats expérimentaux dans les deux travaux ont démontré l'efficacité de ces deux méthodes à résoudre des problèmes de grande taille.

Avant 1990, peu de travaux ont porté sur le PCMC. Tout d'abord, il semblait naturel d'adapter la procédure duale-ascendante de Balakrishnan, Magnanti et Wong (1989) pour résoudre le PCMC, étant donné la qualité remarquable des résultats obtenus pour le problème sans capacité (1% à 4% de l'optimum sur des problèmes de grande taille). Toutefois, les auteurs mentionnent avoir obtenu des résultats beaucoup moins performants à cause de la faible qualité de l'approximation fournie par la relaxation continue, et ces efforts ont été abandonnés par la suite (voir la thèse de doctorat de Balakrishnan 1984).

À notre connaissance, le PCMC n'a fait l'objet d'aucune étude polyédrale. Par contre, il y a eu dans la littérature des études polyédrales effectuées pour une structure particulière que nous retrouvons dans des relaxations du PCMC. Cette structure est celle du noeud-seul, ou "single-node", qui a été principalement étudiée par Padberg,

Van Roy et Wolsey (1985). Dans cette étude, les auteurs ont développé une famille d'IV (inégalité de couverture de flot, ou ICF, introduite à la section 4.3) pour la structure du noeud-seul ayant des bornes supérieures sur les flots des arcs. Ensuite, l'ICF a été adaptée par Van Roy et Wolsey (1986a) pour une structure similaire à la première mais ayant en plus des bornes inférieures sur les flots. Par la suite, Wolsey (1989) a généralisé les inégalités trouvées dans les travaux cités précédemment en une seule grande famille, celle des inégalités sous-modulaires. Stallaert (2000) a utilisé les inégalités développées par Padberg, Van Roy et Wolsey (1985) pour résoudre des problèmes de PCMC à un seul produit.

Dans le but d'améliorer la qualité des bornes inférieures sur l'objectif, plusieurs chercheurs ont tenté différentes relaxations lagrangiennes : Gendron et Crainic (1994, 1996), Holmberg et Yuan (2000), Gendron, Crainic et Frangioni (1998), et Crainic, Frangioni et Gendron (2001). Cependant, Gendron et Crainic (1994) ont démontré que différentes relaxations du problème fournissent la même borne, qui est égale à celle de la relaxation linéaire forte. En effet, les sous-problèmes obtenus des différentes relaxations lagrangiennes possèdent la propriété d'intégralité. Comme toutes les bornes sont théoriquement égales, la comparaison entre les différentes approches de relaxation se base donc sur l'efficacité expérimentale à résoudre les sous-problèmes, de même qu'à calculer des directions de modification dans le dual lagrangien.

Gendron, Crainic et Frangioni (1998) ont étudié comment se comparent les relaxations linéaires faible et forte, telles que résolues par un logiciel commercial de programmation linéaire, et différentes approches de relaxations lagrangiennes. De même, ils ont comparé la qualité de la solution réalisable obtenue par une heuristique lagrangienne avec celle obtenue avec une heuristique de recherche avec tabous conçue par Crainic, Gendreau et Farvolden (2000). Les résultats numériques ont confirmé la supériorité de la relaxation forte sur la relaxation faible, ainsi que la supériorité de la solution réalisable obtenue par la recherche avec tabous sur celle obtenue par l'heuristique lagrangienne de décomposition selon les ressources.

Étant donné la difficulté du PCMC, plusieurs chercheurs ont tenté des approches

heuristiques afin de trouver une solution réalisable de qualité fournissant ainsi une borne supérieure sur l'objectif. Crainic, Gendreau et Farvolden (2000) et Ghamlouche, Crainic et Gendreau (2001) ont proposé des méthodes de recherche avec tabous. Hernu (2001) a proposé une heuristique basée sur des méthodes de programmation mathématique. Les résultats numériques dans les trois travaux ont montré une performance relativement bonne pour les problèmes de taille moyenne. Pour les problèmes de grande taille, il est difficile d'évaluer les performances de ces heuristiques en raison du fait que les solutions optimales, ou même des bornes inférieures de qualité, ne sont pas connues pour ces problèmes.

2.3 Modèles apparentés au PCMC

Plusieurs modèles apparentés au PCMC, et qui en constituent des cas particuliers, ont été largement étudiés dans la littérature, notamment le problème de l'arbre sous-tendant minimal, le problème du plus court chemin et le problème de multiflot à coût minimum. D'autres cas apparentés au problème, plus difficiles que ceux que nous venons de citer, ont généré beaucoup de travaux. Pour une revue complète et détaillée, voir Magnanti et Wong (1984), Minoux (1989), Ahuja *et al.* (1995), Nemhauser et Wolsey (1988), ainsi que la bibliographie de Balakrishnan, Magnanti et Mirchandani (1997). Nous présentons dans les prochaines sections deux classes de modèles apparentés : les problèmes de localisation et les problèmes d'installation sur les arcs ("network loading"). Nous nous intéressons particulièrement à ces deux classes parce qu'elles ont généré des études polyédrales, qui pourraient être exploitées pour résoudre le PCMC.

Problèmes de localisation

Dans cette section, nous présentons des problèmes discrets de localisation simple à un niveau, ou "simple plant location problem". Nous nous intéressons précisément à ce type de problèmes parce qu'ils sont des cas particuliers du PCMC. Ces problèmes consistent généralement à déterminer un sous-ensemble optimal de sites, choisis parmi un ensemble fini de sites potentiels pour l'installation de dépôts, permettant de satisfaire la demande de clients. L'objectif dans ce type de problèmes est de minimiser les

coûts totaux de transport de produits et les coûts fixes reliés à l'installation des dépôts. La formulation d'un problème de localisation est définie sur un graphe $G = (N, A)$ où N est l'ensemble de noeuds et A est l'ensemble des arcs. Les noeuds dans le graphe représentent les sites potentiels possibles pour l'implantation des dépôts et les clients à servir. Donc, soient $I \subseteq N$ et $J \subseteq N$ les ensembles des noeuds qui représentent les clients et les sites respectivement (notons que $A \subseteq I \times J$).

Le modèle MIP de localisation multiproduits avec capacité, ou PLC, est le suivant :

$$(PLC) \quad \min \quad \sum_{j \in J} f_j y_j + \sum_{p \in P} \sum_{(i,j) \in A} c_{ij}^p x_{ij}^p,$$

sujet à :

$$x_{ij}^p \leq \min\{u_j, d_i^p\} y_j, \quad \forall (i, j) \in A, \forall p \in P, \quad (8)$$

$$\sum_{j \in J} x_{ij}^p = d_i^p, \quad \forall i \in I, \forall p \in P, \quad (9)$$

$$\sum_{p \in P} \sum_{i \in I} x_{ij}^p \leq u_j y_j, \quad \forall j \in J, \quad (10)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J, \quad (11)$$

$$x_{ij}^p \geq 0, \quad \forall (i, j) \in A, \forall p \in P, \quad (12)$$

où

- f_j : coût d'installation du dépôt j ;
- u_j : capacité du dépôt j ;
- d_i^p : demande du client i pour le produit p ;
- y_j : variable de localisation liée au dépôt j ($y_j = 1$ si un dépôt est installé au site j et $y_j = 0$ sinon) ;
- x_{ij}^p : quantité de flot de produit p desservie par le dépôt j pour le client i .

Le modèle ainsi introduit représente la formulation forte du problème parce qu'il contient les contraintes redondantes (8). Dans le cas où ces contraintes sont éliminées, on obtient la formulation faible du problème. La différence entre les deux formulations est que la relaxation linéaire de la formulation forte donne une meilleure approximation que celle de la formulation faible.

On obtient le modèle de localisation sans capacité, PLS, dans le cas où

$$u_j \geq \sum_{p \in P} \sum_{i \in I} d_i^p, \quad \forall j \in J.$$

Les contraintes (8) sont alors remplacées par les contraintes :

$$x_{ij}^p \leq d_i^p y_j, \quad \forall (i, j) \in A, \forall p \in P,$$

et les contraintes (10) peuvent être éliminées parce qu'elles sont redondantes.

Le modèle de localisation est un cas particulier du modèle de conception de réseaux. Pour s'en convaincre, il suffit d'introduire une super origine S , fournissant au réseau la totalité de la demande et de la relier à chaque dépôt par un arc. Chaque arc de ce type est muni d'un coût fixe de conception égal à f_j , d'un coût de transport par unité de flot de produit p égal à $c_{Sj}^p = 0$ et d'une capacité égale à u_j (bien sûr, dans le cas où les dépôts admettent des capacités). Bien que le problème de localisation soit un cas particulier du problème de conception de réseaux avec capacité, les travaux entrepris sur les deux problèmes sont encore bien souvent peu reliés. Cela est dû surtout à la structure spéciale du premier : pour un ensemble fixe de dépôts choisi, le problème devient un problème de transport.

Les problèmes de localisation ont généré énormément de travaux depuis un certain nombre d'années. Les articles de Krarup et Pruzan (1983), Cornuéjols, Nemhauser et Wolsey (1990), Cornuéjols, Sridharan et Thizy (1991), Conn et Cornuéjols (1990) ainsi que le livre de Mirchandani et Francis (1990) mettent en évidence l'ampleur et la diversité des travaux entrepris pour résoudre ces problèmes. Des algorithmes de résolution très efficaces ont été conçus pour les problèmes de localisation sans capacité. Mentionnons principalement la procédure duale-ascendante, DUALOC, conçue par Erlenkotter en 1978, et qui est similaire à celle utilisée par Bilde et Krarup (1977). DUALOC consiste principalement à approximer la valeur optimale du dual de la relaxation forte du problème, tout en générant des solutions réalisables de qualité. Le succès de cette procédure revient à ce que la relaxation forte fournit une bonne

approximation. Même pour les instances où cette propriété ne tient pas, Conn et Cornuéjols (1990) montrent comment réduire le gap en utilisant des IV.

Malheureusement, pour les problèmes avec capacité, la relaxation forte ne fournit plus une aussi bonne approximation. Dans le but d'améliorer la formulation du problème de localisation avec capacité, plusieurs chercheurs ont étudié les propriétés polyédrales du problème afin de générer des inégalités valides (Leung et Magnanti 1989, et Aardal, Pochet et Wolsey 1995). Les IV étudiées sont principalement des généralisations et des adaptations de l'inégalité de couverture de flot, que nous présentons à la section 4.3. Aardal, dans son article de 1998, explique comment traiter les inégalités valides trouvées avec l'approche des méthodes de coupes, que nous décrivons à la section 3.

Problèmes d'installation sur les arcs

Les problèmes d'installation sur les arcs, ou PIA, consistent à satisfaire à un coût minimal une certaine demande de plusieurs types de produits. Pour satisfaire la demande, on a la possibilité d'installer des capacités de plusieurs types sur les arcs. Soit $L = \{1, 2, \dots, |L|\}$ l'ensemble des types de capacités qu'on peut installer et soit C_l la capacité de chaque type $l \in L$. On suppose que les C_l sont des entiers tels que

$$C_1 < C_2 < C_3 < \dots < C_{|L|}.$$

Chaque type de capacité admet un coût d'installation par arc (i, j) dénoté f_{ij}^l . Tout comme le PCMC, le PIA est défini sur un graphe $G = (N, A)$, où N est l'ensemble des noeuds et A est l'ensemble des arcs. Dénotons P l'ensemble des produits, ou paires origine-destination. Le PIA s'énonce alors comme suit :

$$(PIA) \quad \min \quad \sum_{p \in P} \sum_{(i,j) \in A} c_{ij}^p x_{ij}^p + \sum_{l \in L} \sum_{(i,j) \in A} f_{ij}^l y_{ij}^l,$$

sujet à :

$$(2), (5),$$

$$\sum_{p \in P} x_{ij}^p \leq \sum_{l \in L} C_l y_{ij}^l, \quad \forall (i, j) \in A,$$

$$y_{ij}^l \in \mathbf{N}, \quad \forall (i, j) \in A, \forall l \in L,$$

où

- x_{ij}^p : des variables continues, qui représentent la quantité de flot de produit p circulant sur l'arc (i, j) ;
- y_{ij}^l : des variables entières, qui représentent le nombre d'unités du type de capacité l installé sur l'arc (i, j) ($\mathbf{N} = \{0, 1, \dots\}$).

Le PIA est un cas particulier du PCMC. Pour s'en convaincre, il suffit d'introduire, pour chaque arc (i, j) , $\sum_{l \in L} \lceil \frac{\sum_{p \in P} d^p}{C_l} \rceil$ arcs parallèles correspondant au flot maximum pouvant être desservi par chacun des L types de capacités disponibles. À chaque arc $(i, j)^{lk}$, introduit pour le type l et pour $k \in \{1, \dots, \lceil \frac{\sum_{p \in P} d^p}{C_l} \rceil\}$, sont associés un coût fixe $k \times f_{ij}^l$, une capacité C^l et $|P|$ variables continues (x_{ij}^p)^{lk} représentant le flot de produit p circulant sur l'arc $(i, j)^{lk}$. Le PIA est extrêmement difficile à résoudre. Magnanti et Mirchandani (1993) ont démontré que le PIA ayant un seul type de capacité est un problème NP- difficile et ce, même dans le cas où on dispose d'une seule origine pour tous les produits.

La manière la plus naturelle de traiter ce problème serait de résoudre sa relaxation continue. Cependant, il a été démontré théoriquement (Epstein 1998) et expérimentalement (Magnanti, Mirchandani et Vachani 1995) que la relaxation linéaire ne fournit pas une bonne approximation du problème, ce qui a amené les chercheurs à améliorer la formulation du problème en lui ajoutant des inégalités valides. Afin d'étudier les propriétés polyédrales du problème pour pouvoir générer des inégalités valides, les chercheurs ont travaillé sur des versions simplifiées du problème. La plupart des problèmes étudiés étaient soit à un seul type de capacité ($|L| = 1$), soit à plusieurs types de capacité qui sont des entiers multiples d'une certaine modularité (C_{l+1} est un entier multiple de $C_l, \forall l \in L$). Magnanti et Mirchandani (1993) ont étudié le problème avec un seul type de produit et au plus trois types différents de capacités. Chopra, Gilboa et Sastry (1998) ont dérivé de nouvelles inégalités pour le problème à un seul produit et deux types de capacités. Magnanti, Mirchandani et Vachani (1993, 1995) ont étudié le problème multiproduits avec deux types de capacités. Les inégalités

valides trouvées pour ce problème vont être présentées dans les prochaines sections. Ces inégalités ont été généralisées par la suite par Bienstock et Günlük (1996) et incorporées dans une méthodologie de résolution par Günlük (1999) pour le même problème, mais avec l'existence d'une certaine distribution initiale de capacité sur les arcs, des demandes directionnelles et des coûts de transport. Barahona (1996) montre comment utiliser les IV sur des relaxations du problème PIA multiproduits avec un seul type de capacité. Gabrel, Knippel et Minoux (1999) ont utilisé des IV (les inégalités de coupes qui sont présentées à la section 4.4) dans l'esprit d'une méthode de décomposition de Benders (Benders 1962) pour résoudre d'une manière exacte une généralisation du PIA. Le problème étudié admet toujours la possibilité d'installer différents types de capacités, le coût d'installation sur chaque arc étant donné par une fonction en escalier discontinue croissante.

Récemment, il y a eu des études polyédrales effectuées sur des problèmes plus généraux de PIA comme le travail de Bienstock et Muratore (2000) sur le problème avec des contraintes de fiabilité (ce sont des contraintes supplémentaires utilisées pour s'assurer que la demande va être satisfaite dans le cas où certains liens sont coupés dans le réseau construit), ou encore le travail d'Atamtürk (2000) sur le problème multiproduits sans restriction sur les capacités.

3 Approches polyédrales

Considérons un modèle de programmation linéaire en nombres entiers. Nous savons qu'une solution optimale à ce modèle est un point extrême de l'enveloppe convexe du domaine réalisable (Nemhauser et Wolsey 1988). Si nous connaissions une description linéaire explicite de l'enveloppe convexe du domaine réalisable, le problème serait résolu facilement par un algorithme de programmation linéaire en variables continues. La description linéaire explicite de l'enveloppe convexe se fait par l'identification des contraintes linéaires qui définissent toutes les facettes du domaine réalisable. Cependant, le problème d'identification de toutes les facettes de l'enveloppe convexe est généralement au moins aussi difficile à résoudre que le modèle en nombres entiers lui-même.

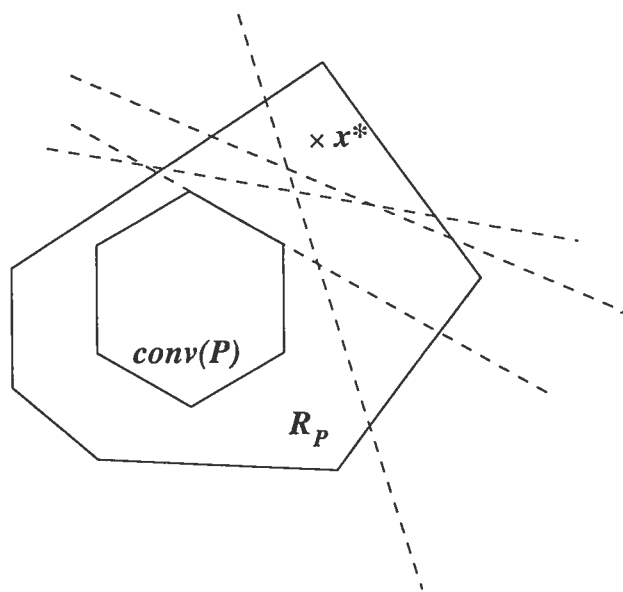


FIG. 1 – L'enveloppe convexe $conv(P)$ et une relaxation R_P .

La philosophie générale des approches polyédrales est d'améliorer la formulation du problème par l'ajout d'IV. À cette fin, nous distinguons deux approches : une approche théorique et une autre algorithmique. L'objectif principal de l'approche théorique consiste à identifier des formulations mathématiques qui définissent des contraintes linéaires valides, appelés des familles d'IV. Ces familles représentent idéalement des facettes ou des faces de dimension élevée de l'enveloppe convexe du domaine réalisable. Cette approche est basée sur une étude approfondie des propriétés polyédrales du problème à résoudre. Après l'identification théorique des familles d'IV vient le problème de la génération de ces IV et de leur ajout au modèle. Souvent l'ajout explicite de toutes les IV au modèle n'est pas efficace, autant à cause du temps nécessaire à cette tâche qu'en raison de la grande taille du modèle résultant. En plus, plusieurs inégalités ne sont d'aucune utilité dans le modèle soit parce qu'elles sont dominées par d'autres et donc redondantes (l'ajout de telles inégalités ne couperait pas davantage de points), soit parce qu'elles définissent des faces de dimension plus faible qu'une facette de l'enveloppe convexe. L'approche algorithmique consiste donc à résoudre une relaxation du problème et si la solution obtenue, x^* , n'est pas réalisable pour le problème original, à générer et à ajouter au modèle un sous-ensemble d'IV violées par la solution courante du problème relaxé. La figure 1 illustre cette situation ; ici,

$\text{conv}(P)$ représente l'enveloppe convexe des points entiers, alors que R_P est le domaine décrit par la relaxation. La solution optimale, x^* , est coupée de l'enveloppe convexe par plusieurs IV, représentées par les droites pointillées. Parmi celles-ci, la plus forte correspond à l'inégalité qui définit une facette de l'enveloppe convexe, qui est illustrée par la droite pointillée qui supporte l'enveloppe convexe $\text{conv}(P)$. Les IV ainsi ajoutées au modèle peuvent être utilisées soit dans le cadre d'une méthode de coupes, soit dans le cadre d'une méthode de relaxation lagrangienne. Dans le premier cas, on ajoute les IV à la formulation du problème et on résout sa relaxation linéaire par le simplexe. Dans le deuxième cas, on va plutôt relaxer les IV trouvées en les introduisant dans l'objectif avec des termes de pénalités. Le choix entre l'une ou l'autre des méthodes dépend essentiellement du problème à résoudre. Ainsi, l'approche lagrangienne peut être plus appropriée dans le cas où le problème relaxé admet une structure particulière qu'il serait souhaitable de préserver.

Le problème d'identifier, parmi une famille ou un ensemble d'IV, au moins une inégalité qui soit violée par un point x^* , s'appelle le problème de séparation. En fait, étant donné une solution optimale x^* (non réalisable) d'une relaxation du modèle MIP, la résolution du problème de séparation consiste à générer au moins une IV qui n'est pas satisfaite par x^* , si une telle IV existe parmi la famille d'IV considérée. L'ajout d'IV à la formulation du problème relaxé sert à séparer la solution courante du domaine réalisable. La difficulté majeure de l'approche algorithmique est que le problème de séparation ne se résout en temps polynômial que si le problème de programmation linéaire, ayant comme contraintes la famille d'IV considérée, se résout lui-même en temps polynômial (Nemhauser et Wolsey 1988). Pour contourner la difficulté du problème de séparation pour certaines familles d'IV, les chercheurs utilisent des heuristiques à la place des méthodes de résolution exacte, la conséquence étant qu'on peut ne pas identifier certaines IV violées par le point courant.

La force des approches polyédrales vis-à-vis des problèmes difficiles réside dans le fait qu'une famille d'IV pour une relaxation du problème constitue une famille d'IV pour le problème lui-même. Cette propriété importante motive les études et les travaux sur les propriétés polyédrales des problèmes qui ont des structures simples, mais

qui constituent des relaxations de problèmes plus complexes. Bien sûr, les IV étudiées ne définissent pas nécessairement des facettes ou des faces de dimensions élevées du problème original. Dans ce cas, la technique d'élévation séquentielle (ou "lifting", présentée en annexe), bien qu'elle soit difficile à appliquer, peut être utilisée pour augmenter la dimension des faces induites par de telles inégalités.

Afin de tirer davantage des méthodes polyédrales et pour développer des algorithmes de résolution efficaces, ces méthodes sont souvent combinées avec des méthodes de type énumératif comme le B&B. Si la génération de coupes se fait juste au niveau de la racine de l'arbre d'énumération alors nous parlons de l'algorithme de Cut-and-Branch (ou C&B). Par contre, si la génération des coupes se fait au niveau de chaque noeud de l'arbre alors nous parlons de l'algorithme de Branch-and-Cut (ou B&C). Cette méthodologie a été présentée pour la première fois par Padberg et Rinaldi (1987). Bien que l'efficacité de cette approche ait été prouvée à maintes reprises (voir la bibliographie de Caprara et Fischetti 1997), ses performances dépendent grandement d'un compromis judicieux entre l'effort de calcul à chaque noeud de l'arbre d'énumération et la taille de l'arbre.

4 Revue des inégalités valides pertinentes au PCMC

Dans cette section, nous présentons trois familles d'IV pertinentes au problème de conception de réseaux : les inégalités de couverture (ou "cover inequalities"), les inégalités de couverture de flot (ou "flow cover inequalities") et des inégalités basées sur les coupes. Pour le lecteur intéressé, des techniques de base utilisées dans la dérivation des IV (Nemhauser et Wolsey 1988) sont rappelées en annexe.

4.1 Inégalités de couverture

Les inégalités de couverture ont été développées pour le problème du sac alpin qui consiste à déterminer un sous-ensemble d'objets optimal parmi un ensemble fini d'objets de telle sorte que le poids total des objets choisis ne dépasse pas une certaine

capacité d . La figure 2 montre la structure induite par ce problème, où le domaine réalisable est donné par $P = \{y \in \{0, 1\}^n : \sum_{j \in N} u_j y_j \leq d\}$ et $N = \{1, 2, \dots, n\}$.

Pour la présentation de cette famille, nous allons introduire quelques définitions.

Définition 1 : Un sous-ensemble $C \subset N$ représente une couverture de P si

$$\sum_{j \in C} u_j > d.$$

Définition 2 : Une couverture C de P est minimale si

$$\sum_{j \in C \setminus \{k\}} u_j \leq d, \quad \forall k \in C.$$

Proposition 1 : (Wolsey 1975) Pour toute couverture C de P , l'inégalité

$$(IC) \quad \sum_{j \in C} y_j \leq |C| - 1 \quad (13)$$

est une IV de P .

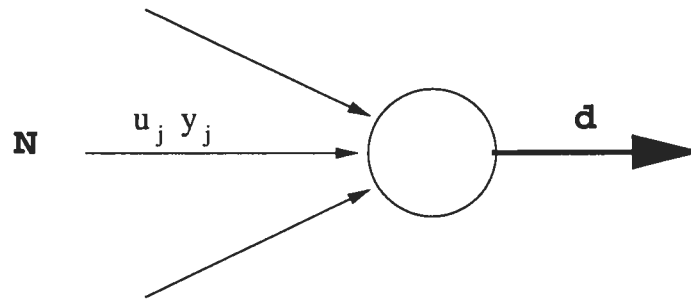


FIG. 2 – Structure induite par le sac alpin.

La démonstration de la validité de (13) est très simple. En effet, si $y_j = 1, \forall j \in C$ alors la solution serait non-réalisable puisqu'on dépasserait la capacité du sac d . De plus, si C est une couverture minimale telle que $C = N$ alors l'inégalité de couverture (13), ou (IC), représente une facette de $\text{conv}(P)$ (Nemhauser et Wolsey 1988). Mais dans le cas où $C \subset N$, l'IC ne représente pas, généralement, une face de dimension élevée de $\text{conv}(P)$. Par contre, elle peut être considérablement renforcée par l'application de la technique d'élévation séquentielle (voir section 4.1.2).

Wolsey (1989) a développé des IV de type sac alpin pour le même problème mais avec des contraintes additionnelles de type

$$\sum_{j \in S_i} y_j \leq l, \text{ pour } i \in I \subset N.$$

Les contraintes additionnelles peuvent être vues comme des contraintes budgétaires dans un problème de conception de réseaux.

L'utilisation de ce type d'IV, comme tout autre type, nécessite la résolution de deux problèmes majeurs : le problème de séparation et le problème de l'élévation séquentielle.

Problème de séparation

Étant donné une solution optimale non-entière y^* d'une relaxation du problème P , le problème de séparation pour l'IC revient à générer une couverture minimale C , de telle sorte que

$$\sum_{j \in C} y_j^* > |C| - 1.$$

À cette fin, introduisons les variables binaires $z_j, j \in N$, où $z_j = 1$ signifie que $j \in C$. Alors, le problème de séparation consiste à résoudre le problème de sac alpin en variables 0-1 suivant :

$$(SepIC) \quad \begin{aligned} Z_{sep} = \min & \quad \sum_{j \in N} (1 - y_j^*) z_j, \\ \text{sujet à :} & \quad \sum_{j \in N} u_j z_j > d, \\ & \quad z_j = 0 \text{ ou } 1, \forall j \in N. \end{aligned}$$

Soit Z_{sep}^* la valeur de l'objectif à l'optimum. Deux cas se présentent :

- si $Z_{sep}^* < 1$, alors nous avons réussi à générer une IC violée par la solution y^* ;
- si $Z_{sep}^* \geq 1$, alors la solution y^* ne viole aucune inégalité de couverture.

Étant donné que le problème (SepIC) est un problème NP-difficile (sac alpin en variables 0-1), on utilise des heuristiques pour le résoudre. Deux conséquences directes découlent de la résolution non exacte du problème :

- Les couvertures générées ne donnent pas les ICs les plus violées par la solution courante, ce qui affecte la qualité (facette ou non) de l'IC.
- Nous pouvons manquer la génération d'une IC violée même lorsqu'elle existe.

L'élévation séquentielle joue un rôle très important dans ces cas. En premier lieu, elle renforce l'inégalité générée en augmentant la dimension de la face par l'ajout des variables qui ne figurent pas dans l'inégalité. En deuxième lieu, même si l'inégalité générée n'est pas violée, il y a une chance qu'elle le soit après l'élévation.

Élévation séquentielle

Padberg, Van Roy et Wolsey (1985) ont développé ce qu'on appelle l'inégalité de couverture élevée (ICE) introduite dans la proposition 2.

Corollaire 1 : Si $C \subset N$ est une couverture minimale de P et si le couple (C_1, C_2) forme une partition quelconque de C telle que $C_1 \neq \emptyset$ alors

$$(IC) \quad \sum_{j \in C_1} y_j \leq |C_1| - 1 \quad (14)$$

représente une facette de $\text{conv}(P(C_1, C_2))$, où

$$P(C_1, C_2) = P \cap \{y \in \{0, 1\}^n : y_j = 0, \forall j \in N \setminus C, y_j = 1 \forall j \in C_2\}.$$

Proposition 2 : Si $C \subset N$ est une couverture minimale de P et si le couple (C_1, C_2) forme une partition quelconque de C telle que $C_1 \neq \emptyset$ alors il existe $\alpha_j \geq 0, \forall j \in N \setminus C$ et $\gamma_j \geq 0, \forall j \in C_2$ tels que l'inégalité

$$(ICE) \quad \sum_{j \in N \setminus C} \alpha_j y_j + \sum_{j \in C_2} \gamma_j y_j + \sum_{j \in C_1} y_j \leq |C_1| - 1 + \sum_{j \in C_2} \gamma_j \quad (15)$$

représente une facette de $\text{conv}(P)$.

La démonstration de la validité de l'inégalité (15) se fait en commençant par l'inégalité (14) et en appliquant la technique d'élévation séquentielle (on applique la Proposition 9, en annexe, pour tout $j \in N \setminus C$ et la Proposition 10, en annexe, pour tout $j \in C_2$).

L'application de la technique d'élévation séquentielle sur l'IC est une tâche difficile. En effet, nous savons que les variables doivent être traitées séquentiellement pour

calculer les coefficients α_j et γ_j de l'inégalité (15). Le calcul de chaque coefficient requiert la résolution d'un problème de sac alpin en variables 0-1, donc en totalité il faut résoudre $|N \setminus C_1|$ problèmes. En plus, l'ordre dans lequel les variables sont traitées affecte énormément les valeurs des coefficients, par conséquent la qualité de la coupe. Nous savons que les variables dans $N \setminus C$ doivent être élevées avant celles dans C_2 , car ces variables contribuent à la violation de la contrainte et que les variables traitées en premier auront des coefficients d'élévation plus grands. Cependant, nous n'avons aucune idée sur le meilleur ordre dans lequel nous devons traiter les variables de C_2 et de $N \setminus C$. En effet, Gu, Nemhauser et Savelsbergh (1999) ont démontré que, étant donné une couverture minimale, le problème qui consiste à identifier l'ordre des variables à élever qui donne l'ICE la plus violée (la meilleure, correspondant à celle qui s'éloigne davantage du point courant) est un problème NP-complet. Normalement, les variables fractionnaires doivent être élevées en premier parce qu'elles contribuent à la violation de l'IV, ensuite les variables qui sont à leur borne supérieure, et à la fin celles qui sont à leur borne inférieure.

Pour élever les variables dans C_2 , nous devons résoudre un problème de sac alpin en variables 0-1 pour chaque coefficient d'élévation. En effet, supposons que nous sommes à une itération intermédiaire de la procédure d'élévation, et soit $L = \{\text{variables déjà élevées}\}$ et $\bar{d} = d - \sum_{j \in C_2 - L} u_j$. Si nous voulons traiter la variable $y_{j_k} \in C_2$, alors nous devons résoudre le problème suivant :

$$\begin{aligned} \gamma_{max} = \max & \quad \sum_{j \in C_1} y_j + \sum_{j \in L} \gamma_j y_j, \\ \text{sujet à :} & \quad \sum_{j \in C_1 \cup L} u_j y_j \leq \bar{d} - u_{j_k}, \\ & \quad y_j = 0 \text{ ou } 1, \forall j \in C_1 \cup L. \end{aligned}$$

Donc, le coefficient de la variable traitée serait $\gamma_{j_k} = \gamma_{max} - |C| + 1 - \sum_{j \in L} \gamma_j$.

De la même façon, pour élever les variables dans $N \setminus C$, nous devons résoudre le

problème de sac alpin suivant :

$$\begin{aligned} \alpha_{max} = \max & \quad \sum_{j \in C_1} y_j + \sum_{j \in L} \alpha_j y_j, \\ \text{sujet à :} & \quad \sum_{j \in C_1 \cup L} u_j y_j \leq \bar{d} - u_{j_k}, \\ & \quad y_j = 0 \text{ ou } 1, \forall j \in C_1 \cup L. \end{aligned}$$

Dans ce cas, le coefficient de la variable y_{j_k} serait $\alpha_{j_k} = |C| - 1 + \sum_{j \in C_2} \gamma_j - \alpha_{max}$.

Mentionnons qu'il n'y a pas une méthode systématique pour déterminer la meilleure partition (C_1, C_2) de C . Les chercheurs utilisent généralement des heuristiques basées sur les informations liées à la solution courante.

Un cas particulier important suivrait lorsque $C_2 = \phi$. Dans ce cas, seulement les variables dans $|N \setminus C|$ sont élevées pour donner l'inégalité de couverture élevée simple (ICES) suivante :

$$\sum_{j \in C} y_j + \sum_{j \in N \setminus C} \alpha_j y_j \leq |C| - 1.$$

L'importance de ce cas particulier revient au théorème de Balas (1975) qui donne une formule pour déterminer de façon approchée tous les coefficients de l'élévation.

Applications antérieures

Depuis les années 70, les ICs existent dans la littérature. Elles ont été utilisées à maintes reprises et avec succès dans des algorithmes de Branch-and-Cut pour résoudre des programmes entiers en variables 0-1 (Aardal et Weismantel 1997). Crowder, Johnson et Padberg (1983) ont été les premiers à utiliser ces inégalités dans des algorithmes de Cut-and-Branch pour résoudre des programmes entiers en variables 0-1.

Motivé par le succès que ces derniers ont rencontré, Van Roy et Wolsey (1987) ont utilisé les ICE, entre autres, pour créer un programme de résolution pour les modèles MIP connu sous le nom de MPSARX. Depuis, ces inégalités ont été incorporées dans des logiciels commerciaux pour résoudre des modèles de MIP, notamment CPLEX (2000), Xpress-MP (1999) et MINTO (Nemhauser, Savelsbergh et Sigismondi 1994).

Bien sûr, il y a eu des adaptations spéciales et des applications de ces IV à des problèmes spécifiques provenant d'applications réelles comme dans Aardal, Pochet et Wolsey (1995) pour le problème de localisation avec capacité, ou encore dans Barnhart, Hane et Vance (2000) pour un problème de flot multiproduits à variables entières. Pour une revue complète, voir les bibliographies de Aardal et Weismantel (1997), Caprara et Fischetti (1997) et Balakrishnan, Magnanti et Mirchandani (1997). Remarquons que, malgré la diversité et l'ampleur des études publiées concernant l'utilisation de cette IV, cette dernière n'a jamais été adaptée spécialement pour le PCMC.

Une recherche exhaustive a été faite par Gu, Nemhauser et Savelsbergh (1998) concernant différentes façons de résoudre le problème de séparation et le problème d'élévation. Pour résoudre le problème de séparation, ils ont utilisé la programmation dynamique pour obtenir une solution exacte, la relaxation linéaire pour dériver une solution approximative et une heuristique basée sur un algorithme vorace pour calculer une solution réalisable. La dernière a donné la meilleure performance, que ce soit pour la qualité ou pour l'effort requis. Pour le problème d'élévation, ils ont essayé plusieurs séquences pour l'élévation des variables.

Leurs contributions ne sont pas seulement expérimentales. Dans un autre article, Gu, Nemhauser et Savelsbergh (1999) ont pu démontrer que, étant donné une solution fractionnaire et une couverture minimale, le problème de déterminer une ICES violée est un problème NP-complet. De même, le problème de déterminer une ICES violée parmi toutes les couvertures minimales, étant donné une solution fractionnaire, est aussi un problème NP-complet.

Il reste une question importante : est-ce que la famille d'IC avec toutes ces variantes décrit entièrement l'enveloppe convexe du polytope du sac alpin ? La réponse est généralement non. Weismantel (1997) a identifié une classe d'inégalités pour des problèmes de sac alpin particuliers. Ces inégalités représentent des facettes de l'enveloppe convexe des problèmes étudiés. Il a prouvé que ces facettes ne peuvent pas être générées comme des ICE.

4.2 Inégalités de couverture de flot

Les IV de couverture de flot ont été développées initialement par Padberg, Van Roy et Wolsey (1985) durant l'étude des propriétés polyédrales de la structure du noeud-seul ("single-node") que nous pouvons identifier dans des relaxations du problème de conception de réseaux. Soit P_{SN} le domaine réalisable d'une telle structure (illustrée à la figure 3) où

$$P_{SN} = \{y \in \{0, 1\}^n, x \in R_+^n : \sum_{j \in N} x_j \leq d, x_j \leq u_j y_j, \forall j \in N\}.$$

Soit $R_{P_{SN}}$ le polytope de la relaxation linéaire de P_{SN} , où

$$R_{P_{SN}} = \{y \in R_+^n, x \in R_+^n : \sum_{j \in N} x_j \leq d, x_j \leq u_j y_j, y_j \leq 1, \forall j \in N\}.$$

Une couverture de flot $C \subset N$ est un sous-ensemble d'arcs tel que $\sum_{j \in C} u_j > d$. Soit

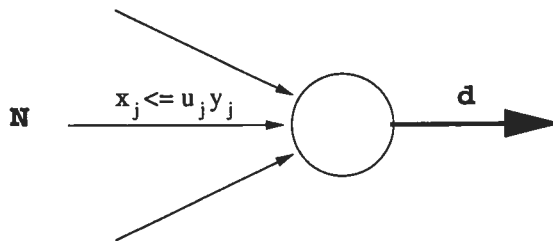


FIG. 3 – Problème à un seul noeud avec uniquement des arcs entrants.

$\lambda = \sum_{j \in C} u_j - d$ l'excès de capacité de l'ensemble C . Pour tout $k' \in C$, la capacité de l'ensemble $C \setminus \{k'\}$ est

$$\min\left\{ \sum_{j \in C \setminus \{k'\}} u_j, d \right\} = d - (u_{k'} - \lambda)^+$$

où $(x)^+ = \max(0, x)$.

Alors pour tout $C' \subseteq C$, la capacité de l'ensemble $C \setminus C'$ est donnée par

$$d - \left(\sum_{j \in C'} u_j - \lambda \right)^+ \leq d - \sum_{j \in C'} (u_j - \lambda)^+.$$

Ce résultat nous amène à introduire l'*inégalité de couverture de flot* ou ICF (Padberg, Van Roy et Wolsey 1985).

Proposition 3 : Si $C \subset N$ est une couverture de flot et $\lambda = \sum_{j \in C} u_j - d$ est l'excès de capacité, alors

$$(ICF) \quad \sum_{j \in C} x_j \leq d - \sum_{j \in C} (u_j - \lambda)^+(1 - y_j) \quad (16)$$

est une IV pour P_{SN} .

Remarquons que la face induite par l'inégalité de couverture de flot (16) détermine un hyperplan qui sépare tout point extrême non réalisable du domaine de $R_{P_{SN}}$ (Nemhauser et Wolsey 1988). En plus, Padberg, Van Roy et Wolsey (1985) démontrent que si $\max_{j \in C} u_j > \lambda$ alors l'ICF (16) définit une facette de $\text{conv}(P_{SN})$.

De la même façon, Van Roy et Wolsey (1986a) ont généralisé l'ICF (16) pour une structure similaire à la première mais ayant des arcs entrants et sortants. Cette structure est illustrée à la figure 4. Soit P_{SNG} le domaine définissant une telle structure :

$$P_{SNG} = \{y \in \{0, 1\}^n, x \in R_+^n : \sum_{j \in N_1} x_j - \sum_{j \in N_2} x_j \leq d, x_j \leq u_j y_j, \forall j \in N_1 \cup N_2\}.$$

L'inégalité dérivée pour cette structure est connue sous le nom d'*inégalité de couverture de flot généralisée* (ICFG) donnée par la proposition 4.

Proposition 4 : Si $C = (C_1, C_2)$ est une couverture de flot généralisée dans le sens où $\sum_{j \in C_1} u_j - \sum_{j \in C_2} u_j = d + \lambda$, avec $\lambda > 0$, $C_1 \subseteq N_1$, et $C_2 \subseteq N_2$, et si $L_2 \subseteq N_2 \setminus C_2$, alors

$$(ICFG) \quad \sum_{j \in C_1} x_j + \sum_{j \in C_1} (u_j - \lambda)^+(1 - y_j) \leq d + \sum_{j \in C_2} u_j + \sum_{j \in L_2} \lambda y_j + \sum_{j \in N_2 \setminus (C_2 \cup L_2)} x_j$$

est une IV pour P_{SNG} .

Les auteurs ont démontré que dans le cas où $C_2 = \emptyset$, $d > 0$, $\max_{j \in C_1} u_j > \lambda$ et $u_j > \lambda, \forall j \in L_2$, l'ICFG définit une facette de $\text{conv}(P_{SNG})$. Mais si $C_2 \neq \emptyset$, l'inégalité ne représente plus une facette, en général, et elle doit être renforcée par l'application de la technique d'élévation séquentielle. Cependant, l'élévation dans ce cas est extrêmement

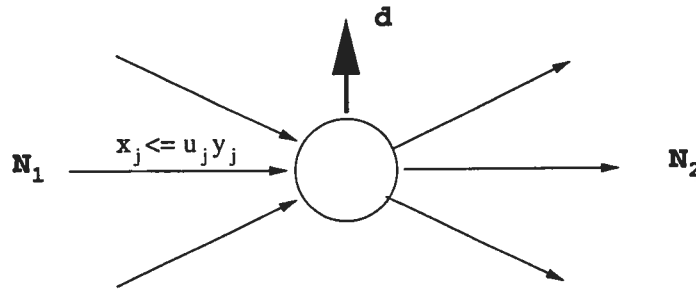


FIG. 4 – Problème à un seul noeud avec des arcs entrants et sortants.

difficile parce que nous devons calculer pour chaque élévation deux coefficients pour le couple (x_j, y_j) en même temps, ce qui fait qu'il faut résoudre pour chaque couple (x_j, y_j) un problème MIP.

Problème de séparation

Le problème de séparation, dont le but est de générer une ICF violée par la solution courante (x^*, y^*) , est difficile à résoudre et coûteux. En effet, pour trouver la couverture de flot $C = (C_1, C_2)$ pour laquelle l'inégalité (ICFG) est la plus violée, définissons Z^1 et Z^2 comme étant les vecteurs caractéristiques de C_1 et C_2 respectivement. Le problème de séparation de l'ICF est le suivant :

$$\begin{aligned}
 Z_{sep} = \quad & \max \quad \sum_{j \in N_1} (x_j^* + (u_j - \lambda)^+(1 - y_j^*)) z_j^1 - d - \sum_{j \in N_2} u_j z_j^2 \\
 & \quad \quad \quad - \sum_{j \in N_2} \min(\lambda y_j^*, x_j^*) (1 - z_j^2), \\
 \text{sujet à :} \quad & \\
 (SepICFG) \quad & \sum_{j \in N_1} u_j z_j^1 - \sum_{j \in N_2} u_j z_j^2 = d + \lambda, \\
 & \quad \quad \quad \lambda > 0, \\
 & \quad \quad \quad z_j^1 = 0 \text{ ou } 1, \forall j \in N_1, \\
 & \quad \quad \quad z_j^2 = 0 \text{ ou } 1, \forall j \in N_2.
 \end{aligned}$$

Dans le but de résoudre un tel problème, les chercheurs utilisent des simplifications et des relaxations comme :

- Poser $L_2 = N_2 \setminus C_2$.
- Remplacer $(u_j - \lambda)^+$ par $(u_j - \lambda)$.
- Remplacer x_j^* par $u_j y_j^*$, $\forall j \in N_1$.
- Substituer d par sa valeur en fonction de λ .
- Substituer $\lambda > 0$ dans la première contrainte.

Après avoir posé ces conditions, le problème de séparation devient un problème de sac alpin en variables 0-1 :

$$\begin{aligned}
 Z_{sep} = \max & \quad \sum_{j \in N_1} (y_j^* - 1) z_j^1 + \sum_{j \in N_2} y_j^* z_j^2, \\
 \text{sujet à :} & \\
 (SepICFG) & \quad \sum_{j \in N_1} u_j z_j^1 - \sum_{j \in N_2} u_j z_j^2 > d, \\
 & \quad z_j^1 = 0 \text{ ou } 1, \forall j \in N_1, \\
 & \quad z_j^2 = 0 \text{ ou } 1, \forall j \in N_2.
 \end{aligned}$$

Bien sûr, le problème de séparation simplifié demeure un problème difficile à résoudre et ne génère pas nécessairement une bonne ICF violée, mais du moins, on dispose de plusieurs algorithmes relativement efficaces pour le résoudre, que ce soit des heuristiques, des relaxations ou des algorithmes de programmation dynamique.

Applications antérieures

Les inégalités de couverture de flot ont été utilisées à maintes reprises dans la littérature. Van Roy et Wolsey (1987) ont démontré l'amélioration effective de la formulation du problème de conception de réseaux sans capacité suite à l'ajout des ICF.

Tout comme les inégalités de couverture, les ICF ont été incorporées dans des logiciels commerciaux, comme MPSARX (Van Roy et Wolsey 1987), CPLEX (2000), Xpress-MP (1999) et MINTO (Nemhauser, Savelsbergh et Sigismondi 1994), pour créer des algorithmes de branch-and-cut efficaces pour résoudre des modèles de MIP.

De même, ces IV ont été adaptées par plusieurs chercheurs pour des problèmes spécifiques. Mentionnons principalement Aardal, Pochet et Wolsey (1995) qui ont utilisé les ICF pour reformuler le problème de localisation avec capacité. Les auteurs ont démontré que ces IV représentent en fait des facettes de l'enveloppe convexe du PLC sous certaines conditions. De plus, les auteurs ont dérivé d'autres IV en se basant sur le même principe (couverture de flot). Également, Magnanti, Mirchandani et Vachani (1993) se sont basés sur le même principe pour dériver des IV, connues sous le nom d'*inégalités de capacités résiduelles*, pour le problème d'installation sur les arcs avec un seul type de capacité. En fait, les auteurs ont travaillé sur le problème lagrangien obtenu après avoir relaxé les contraintes de conservation de flot. En plus,

les auteurs ont démontré que la nouvelle formulation du problème lagrangien, obtenue en lui additionnant des inégalités de capacités résiduelles et des inégalités dérivées à partir des contraintes de conservation de flot, décrit entièrement l'enveloppe convexe du sous-problème étudié.

Gu, Nemhauser et Savelsberg (1999a) ont développé une méthode pour résoudre le problème d'élévation de l'ICF moyennant la technique d'élévation séquentielle indépendante (voir annexe). Cependant, puisque la fonction d'élévation de l'ICF n'est pas superadditive, les auteurs ont dû passer par des simplifications et des relaxations pour trouver une fonction superadditive qui domine la fonction d'élévation. Les auteurs ont prouvé expérimentalement l'amélioration apportée à la formulation, en général, et à la qualité de l'ICF, en particulier, par l'utilisation de la technique d'élévation séquentielle indépendante.

4.3 Inégalités basées sur des coupes

Ces types d'inégalités sont dérivées à partir des coupes du réseau (i.e., les arcs entre deux sous-ensembles de sommets forment une partition) selon certains critères, dont les plus utilisés sont le critère de conservation de flot et le critère de capacité, qui doivent être respectés pour n'importe quelle coupe qui sépare certaines origines de leurs destinations.

Inégalités de réseaux

Van Roy et Wolsey (1986) ont identifié une famille d'IV, les *inégalités de réseaux* ("network inequalities"), pour le PCMS avec un seul produit, une origine et plusieurs destinations (ou PC1S). Une généralisation de cette inégalité, "dicut inequalities", est étudiée dans Rardin et Wolsey (1993). Nous rappelons qu'une formulation du PCMS est donnée à la section 2.1 où les capacités u_{ij} sont remplacées par $\sum_{i \in N} d_i$ (d_i représente la demande du noeud i), N est l'ensemble des noeuds et A est l'ensemble des arcs. L'idée principale de cette famille est basée sur le principe de conservation de flot pour un sous-ensemble d'arcs définissant une coupe qui sépare l'origine de

certaines destinations. Soit $X \subset N$ un sous-ensemble de noeuds et soient

$$E(X) = \{(i, j) \in A : i \in X, j \in X\},$$

$$(X, \bar{X}) = (X, N \setminus X) = \{(i, j) \in A : i \in X, j \in N \setminus X\}.$$

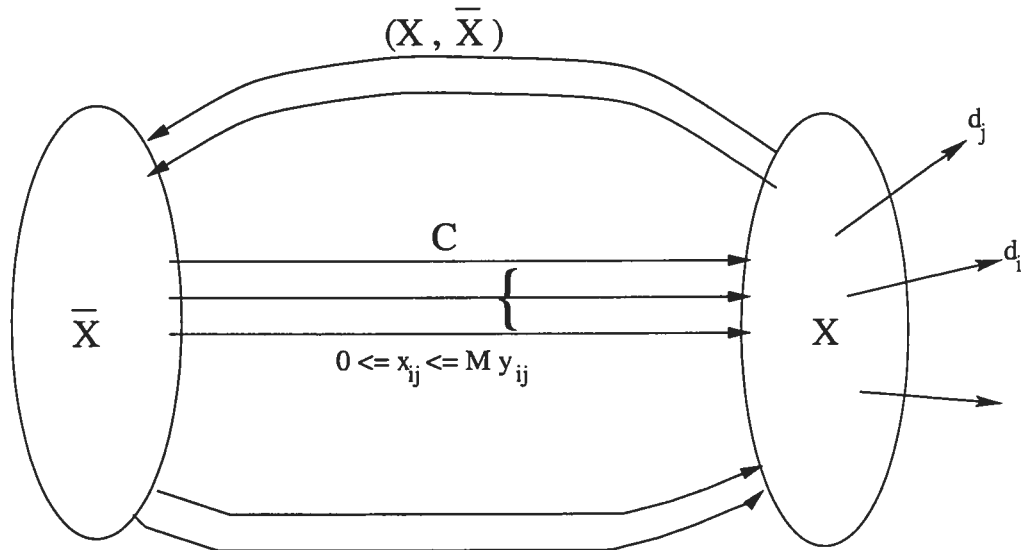


FIG. 5 – La coupe (X, \bar{X}) .

Définition 3 : Le triplet (X, R, C) est compatible si

$$X \subset N, \quad R \subseteq E(X), \quad C \subseteq (\bar{X}, X).$$

Proposition 5 : Étant donné un triplet compatible (X, R, C) (illustré à la figure 5),

alors

$$\sum_{(i,j) \in C} x_{ij} \leq \sum_{(i,j) \in C} \alpha_j^R y_{ij} + \sum_{(i,j) \in E(X) \setminus R} x_{ij} + \sum_{(i,j) \in (X, \bar{X})} x_{ij} \quad (17)$$

est une IV pour le PC1S, où α_j^R est égal à la totalité de la demande pouvant être satisfaite à partir de j , en utilisant seulement les arcs de R .

La démonstration de la validité de (17) est basée sur le fait que tout flot circulant sur les arcs de C doit être

- a) un flot qui circule sur les arcs de R et qui satisfait une certaine demande $d_i, i \in X$,
ou
- b) un flot qui circule sur les arcs de $E(X) \setminus R$, ou
- c) un flot qui circule sur les arcs de (X, \bar{X}) .

Van Roy et Wolsey (1986) démontrent que l'inégalité (17) représente une facette pour certains cas particuliers du PC1S (le problème de "lot-sizing").

Mentionnons simplement que le problème d'élévation de cette inégalité n'a jamais été étudié, ni dans Van Roy et Wolsey (1986), ni dans aucun autre travail publié.

Problème de séparation :

Afin de générer une inégalité de réseau violée, étant donné une solution (x^*, y^*) d'une relaxation du problème qui n'est pas réalisable pour le PCMS (y^* n'est pas entier), il faut déterminer un triplet compatible, le meilleur si possible, (X, R, C) tel que

$$\sum_{(i,j) \in C} x_{ij}^* > \sum_{(i,j) \in C} \alpha_j^R y_{ij}^* + \sum_{(i,j) \in E(x) \setminus R} x_{ij}^* + \sum_{(i,j) \in (X, \bar{X})} x_{ij}^*.$$

Pour pouvoir résoudre le problème de séparation, Van Roy et Wolsey (1986) proposent deux problèmes modifiés, plus simples que le problème de séparation original. Le premier consiste à choisir l'ensemble C étant donné que X et R sont fixes, alors que le deuxième consiste à choisir le meilleur ensemble X étant donné que C est fixe. Dans le deuxième problème, les auteurs supposent que $R = E(X)$.

Applications antérieures :

Tout comme les IC et les ICF, les inégalités de réseaux ont été implantées par Van Roy et Wolsey (1987) dans un système de résolution pour les MIP (MPSARX). Les résultats expérimentaux ont démontré l'efficacité de ces inégalités pour des PC1S. Toutefois, l'utilisation de ces IV a été limitée à des problèmes spécifiques comme le "lot-sizing" (Barany, Van Roy et Wolsey 1984a, 1984b).

Van Roy et Wolsey (1986) ont donné une adaptation de l'inégalité de réseaux pour

traiter des problèmes ayant des capacités sur les arcs de $(E(X) \setminus R \cup (X, \bar{X}))$. Cependant, il n'y a pas eu d'études publiées sur la possibilité d'étendre l'inégalité pour le PCMC.

Inégalités de coupes

Dans une étude des propriétés polyédrales pour le problème d'installation sur les arcs à une origine-destination, Magnanti, Mirchandani et Vachani (1995) ont adapté les inégalités de coupes connues qui sont basées sur le critère du théorème célèbre de flot maximal-coupe minimale. L'idée de base est la suivante : étant donné une certaine installation de capacités sur les arcs, le problème admet une solution réalisable si et seulement si la capacité de chaque coupe séparant l'origine de la destination est plus grande que la demande. Ainsi, soit $(S, \bar{S}) = \{(i, j) \in A : i \in S \subset N, j \in \bar{S} = N \setminus S\}$ une coupe et soient

$D_{(S, \bar{S})} = \sum_{p \in P/O(p) \in S, D(p) \in \bar{S}} d^p$, la demande à travers la coupe (S, \bar{S}) ;

$Y_{(S, \bar{S})}^l = \sum_{(i, j) \in (S, \bar{S})} y_{ij}^l$, le nombre total d'éléments de capacité l installés sur les arcs de la coupe (S, \bar{S}) .

Proposition 6 : L'inégalité

$$Y_{(S, \bar{S})} \geq \lceil D_{(S, \bar{S})} / C \rceil \quad (18)$$

est valide pour le PIA à un type de capacité (PIA1), où C est la capacité de l'unique type disponible à installer. En plus, l'IV (18) définit une facette du PIA1 dans le cas où $D_{(S, \bar{S})} > 0$.

Proposition 7 : L'inégalité

$$Y_{(S, \bar{S})}^1 + r_{(S, \bar{S})} Y_{(S, \bar{S})}^2 \geq r_{(S, \bar{S})} \lceil D_{(S, \bar{S})} / C \rceil$$

est valide pour le PIA à deux types de capacité (PIA2), où

1 est la capacité du premier type,

C est la capacité du deuxième type,

et $r_{(S, \bar{S})} = D_{(S, \bar{S})} \bmod(C)$.

Pour démontrer la validité de l'inégalité, partons du fait que la capacité de la coupe (S, \bar{S}) est plus grande que la demande

$$Y_{(S, \bar{S})}^1 + CY_{(S, \bar{S})}^2 \geq D_{(S, \bar{S})} \quad (19)$$

Le résultat suit en multipliant (19) par $\frac{r_{(S, \bar{S})}}{C}$ et en utilisant le fait que $Y_{(S, \bar{S})}^1 \geq \frac{r_{(S, \bar{S})}}{C} Y_{(S, \bar{S})}^1$.

Proposition 8 : L'inégalité

$$Y_{(S, \bar{S})}^1 + r_{(S, \bar{S})} Y_{(S, \bar{S})}^2 + \lambda r_{(S, \bar{S})} Y_{(S, \bar{S})}^3 \geq r_{(S, \bar{S})} \lceil D_{(S, \bar{S})} / C \rceil$$

est valide pour le PIA à trois types de capacité (PIA3), où

1 est la capacité du premier type,

C est la capacité du deuxième type,

λC est la capacité du troisième type,

et $r_{(S, \bar{S})} = D_{(S, \bar{S})} \bmod(C)$.

La démonstration de la validité de l'inégalité est pareille à celle de la proposition 7. Les auteurs généralisent cette inégalité pour le PIA à plusieurs origines-destinations.

Problème de séparation :

Le problème de séparation pour une inégalité de coupes quelconque consiste à trouver une coupe, la meilleure si possible, (S, \bar{S}) pour laquelle cette inégalité soit violée. Cependant, trouver une telle coupe, même pour un problème de PIA avec une seule origine et une seule destination, est une tâche difficile (problème de flot maximal, voir Mirchandani 1989). En général, afin de générer de bonnes coupes qui engendrent des inégalités de coupes violées, les chercheurs utilisent des heuristiques qui se sont révélées efficaces (Atamtürk 2000).

Applications antérieures :

L'existence des inégalités de coupes remonte loin dans la littérature. Cependant, Magnanti et Mirchandani (1993) et Magnanti, Mirchandani et Vachani (1993, 1995) ont

été les premiers à utiliser ces inégalités pour traiter les PIA à un ou deux types de capacités. La généralisation de cette inégalité pour des PIA plus généraux (multiproduits et plusieurs types de capacités sans aucune restriction) est due à Atamürk (2000). L'auteur a développé des heuristiques pour résoudre les problèmes de séparation pour chaque inégalité étudiée.

Magnanti et Mirchandani (1993) développent deux autres familles d'IV basées sur le même principe en utilisant des techniques d'élévation séquentielle. Les auteurs démontrent que les deux familles d'IV identifiées définissent des facettes de $conv(PIA3)$ sous certaines conditions.

Les inégalités de coupes avec la technique de Chvátal-Gomory (voir annexe) ont été utilisées par Magnanti, Mirchandani et Vachani (1993) pour décrire entièrement l'enveloppe convexe du PIA à trois noeuds, un type de capacité et plusieurs produits.

Magnanti, Mirchandani et Vachani (1995) ont résolu le PIA2 en lui ajoutant des inégalités de coupes, des inégalités de capacité résiduelle et des inégalités redondantes liées à la demande. Les résultats numériques ont montré une amélioration considérable dans l'écart entre la formulation MIP et la relaxation linéaire.

L'efficacité de ces IV à améliorer les formulations des problèmes a été prouvée non seulement pour des cas spécifiques traitant des situations réelles comme dans Barnhart, Jin et Vance (2000), mais surtout lorsqu'incorporées dans des algorithmes de branch-and-cut pour résoudre des PIA ; voir notamment les travaux de Barahona (1996), Bienstock et Günlük (1996), et Günlük (1999). Les travaux de Gabrel, Knipfel et Minoux (1999) et de Bienstock *et al.* (1998) explorent également des stratégies algorithmiques basées sur des généralisations des inégalités de coupes.

5 Résultats expérimentaux

Comme nous l'avons mentionné précédemment, toutes les inégalités présentées dans la section précédente ont été utilisées à maintes reprises dans des algorithmes de B&C. Elles ont notamment été incorporées dans des logiciels commerciaux sophis-

tiqués comme par exemple CPLEX (2000), Xpress-MP (1999), MINTO (Nemhauser, Savelsbergh et Sigismondi 1994), et MPSARX (Van Roy and Wolsey 1987). Ces logiciels représentent actuellement les meilleurs outils généraux de résolution connus pour résoudre des modèles MIP.

Bien que le PCMC n'ait fait l'objet d'aucune étude polyédrale, nous savons qu'il contient des structures particulières cachées dans ses différentes relaxations, comme la structure du sac alpin et la structure de "single-node" étudiée par Van Roy et Wolsey (1986a). En effet, la présence de la structure de "single-node" est évidente dans la formulation du PCMC présentée à la section 2.1. Pour montrer que cette formulation contient des structures du sac alpin, il suffit de combiner les contraintes (2) et (3) et de compléter ensuite les variables binaires y , soit remplacer y_{ij} par $(1 - y_{ij})$. Par conséquent, les inégalités de couverture et de couverture de flot sont des IV pour le PCMC. Étant donné que ces inégalités, entre autres, ont été incorporées dans le logiciel CPLEX, nous avons décidé de tester la méthode de B&C de CPLEX, version 7.0, sur un large ensemble d'instances du PCMC ayant différentes caractéristiques. Rappelons que la formulation forte du PCMC s'est révélée meilleure que la faible (Gendron, Crainic et Frangioni 1998). Malgré ce fait, nous avons décidé de tester l'algorithme de B&C de CPLEX sur les deux formulations et de comparer le nombre d'IV générées et l'amélioration apportée dans les deux cas. Nous voulons ainsi vérifier ou infirmer l'hypothèse selon laquelle CPLEX puisse générer davantage d'IV violées sur la formulation faible du problème, ce qui pourrait permettre d'améliorer la qualité de la solution obtenue, dans le cas où cette dernière n'est pas optimale. Ainsi, nous avons exécuté la méthode de B&C de CPLEX sur les formulations faible et forte de 124 problèmes tests en utilisant un ordinateur Entreprise 10000 (64 Go de Ram, 400MHz). Ces problèmes sont obtenus avec le générateur de réseau décrit dans Crainic, Frangioni et Gendron (2001) et sont caractérisés par un triplet (N, A, P) qui représente le nombre de noeuds, d'arcs et de produits respectivement. Chaque instance (N, A, P) contient $|A|$ variables binaires et $|A| \times |P|$ variables continues. La richesse de ces problèmes tests réside dans le fait que pour chaque triplet, nous disposons de plusieurs instances obtenues en modifiant les niveaux de capacités par

rapport à la demande, et de coûts fixes par rapport aux coûts de transport. Ainsi, pour une même dimension de problème, nous obtenons des instances de niveaux de difficulté différents, dont les moins difficiles correspondent à des problèmes avec des coûts fixes modestes et des capacités larges, et les plus difficiles correspondent à des problèmes avec des coûts fixes élevés et des capacités très contraignantes.

Dans nos expérimentations sur le B&C de CPLEX, nous avons activé plusieurs catégories d'IV disponibles et nous avons imposé une limite de temps CPU de dix heures, dans le but de trouver soit une solution optimale du problème, soit une solution réalisable (donc une borne supérieure sur la valeur optimale) de meilleure qualité.

	N-A-P	B&C Formulation Faible			B&C Formulation Forte		
		SOL	NB-N	COUPES (GUB,IC,ICF)	SOL	NB-N	COUPES (GUB,IC,ICF)
P1	25-100-10	49899* (37.3)	501	(4,7,172)	49899* (40.6)	419	(2,3,116)
P2	25-100-30	85530* (707.6)	4424	(2,11,294)	85530* (534.2)	1350	(1,2,75)
P3	20-230-40	643036* (7226.7)	13704	(0,5,1276)	643036* (671.8)	1618	(0,0,116)
P4	20-300-40	604196* (766.7)	576	(0,7,1053)	604196* (59.9)	92	(0,0,51)
P5	20-230-200	136911 (T)	66	(0,8,1627)	137271 (T)	11	(0,0,0)
P6	20-300-200	109584 (T)	41	(0,0,1508)	108252 (T)	10	(0,0,0)
P7	100-400-10	66680 (T)	2006	(3,14,1069)	66364 (T)	1253	(2,6,610)
P8	30-700-100	57667 (T)	96	(0,3,817)	55082 (T)	118	(0,0,5)
P9	30-520-400	- (T)	0	(0,0,1389)	- (T)	0	(0,0,0)
P10	30-700-400	- (T)	0	(0,0,1547)	- (T)	0	(0,0,0)

TAB. 1 – Résultats d'instances difficiles.

Afin de donner un bref aperçu des résultats obtenus, nous présentons dans le tableau 1 quelques résultats obtenus sur les formulations faible et forte de 10 problèmes, particulièrement difficiles, car ils présentent des coûts fixes élevés relativement aux coûts

de transport et des capacités très contraignantes. Dans la colonne COUPES, à part les inégalités de couverture (IC) et de couverture de flot (ICF), nous avons également généré des inégalités “general upper bound” (GUB), qui représentent une adaptation de l’IC, généralement plus serrée que cette dernière. Les chiffres dans cette colonne représentent le nombre de coupes générées pour les trois types d’inégalités durant l’exécution de l’algorithme. La colonne SOL montre la valeur, soit de la solution optimale (marquée par une étoile), soit de la meilleure solution réalisable trouvée par l’algorithme. Le caractère “-” signifie qu’aucune solution réalisable n’a pu être identifiée même après dix heures. Le temps d’exécution de l’algorithme est donné entre parenthèses sous la solution. La valeur (T) pour le temps d’exécution signifie que l’algorithme s’est arrêté parce qu’il a atteint la limite de temps (soit dix heures) et que la solution rapportée est une solution réalisable. Le nombre de noeuds explorés est donné dans les colonnes NB-N.

Les résultats montrent clairement que la solution obtenue sur la formulation forte est meilleure que celle sur la formulation faible (sauf P5) et ce malgré que le nombre de coupes générées sur la formulation faible soit nettement supérieur à celui sur la forte. Pour le B&C sur la formulation forte nous remarquons que le nombre de coupes générées diminue avec l’augmentation du nombre de produits. Pour des problèmes de grande taille (P5 à P10), CPLEX n’a pu résoudre à l’optimum aucune instance, ni même trouver une solution réalisable pour certains problèmes, même après dix heures de temps de calcul. On remarque également que pour les problèmes à 400 produits, CPLEX a utilisé tout le temps alloué pour traiter la racine de l’arbre d’énumération. En effet, sur la formulation faible de P9 et P10, CPLEX a pris 256.6 sec et 317.6 sec respectivement pour résoudre la relaxation linéaire. Le reste du temps alloué était pris à la fois pour la génération des coupes et pour l’application des heuristiques afin de trouver une solution réalisable.

Afin de mieux qualifier la performance du B&C de CPLEX, nous allons comparer ses résultats avec ceux obtenus par le B&B de CPLEX sur les mêmes problèmes tests et en imposant la même limite de temps de calcul. Le tableau 2 rapporte un sommaire des résultats sur les 124 problèmes tests. Les lignes BBFFA, BBFFO, BCFFA

	OPT	NON-MÉM	SOL-RÉAL	NON-RÉAL
BBFFA	24	39	57	4
BBFFO	54	-	62	8
BCFFA	41	-	77	6
BCFFO	56	-	62	6

TAB. 2 – Comparaison du B&B et du B&C.

et BCFFO correspondent aux résultats du B&B simple sur les formulations faible et forte et du B&C sur les formulations faible et forte respectivement. La colonne OPT donne le nombre de problèmes résolus à l'optimalité. La colonne SOL-RÉAL donne le nombre de problèmes pour lesquels l'algorithme a trouvé une solution réalisable et est arrêté à cause de la limite de temps. Le nombre de problèmes où l'algorithme n'a pu identifier une solution réalisable est donné dans la colonne NON-RÉAL. La colonne NON-MÉM donne le nombre de problèmes où l'algorithme est arrêté parce que CPLEX n'a plus de mémoire disponible.

Les résultats montrent clairement la supériorité de la formulation forte sur la faible que ce soit avec le B&B ou avec le B&C. Remarquons que CPLEX s'est trouvé à court de mémoire uniquement pour le B&B sur la formulation faible. Pour ces problèmes, plusieurs millions de noeuds ont été explorés. Nous constatons la supériorité du B&C sur le B&B, qui est montrée clairement sur la formulation faible. Par contre, cette supériorité n'est pas aussi évidente pour la formulation forte. Le tableau 3 montre la distribution des meilleures solutions réalisables trouvées par les quatre algorithmes pour les 62 problèmes qui n'ont pu être résolus à l'optimalité par BCFFO. Le tableau

BBFFA	BBFFO	BCFFA	BCFFO
3	7	4	48

TAB. 3 – Distribution des solutions réalisables.

3 montre que la meilleure solution de 48 problèmes sur 62 a été trouvée par le B&C sur la formulation forte.

Suite à ces résultats, nous pouvons dégager quelques conclusions importantes :

- 1- Les implantations générales des IV dans CPLEX apportent une certaine amélioration ; toutefois, elles ne sont pas suffisantes pour résoudre des instances de grande taille en des temps raisonnables.
- 2- CPLEX génère rarement des IC sur la formulation forte et très peu sur la formulation faible comparativement aux ICF. Serait-ce parce que les structures de sac alpin sont cachées dans ces formulations ? Dans ce cas, une adaptation spécifique de l'IC au PCMC semble indiquée.
- 3- La formulation forte est supérieure à la formulation faible ; cependant, le problème devient plus lourd à résoudre. Serait-il plus avantageux de traiter les inégalités fortes dans le cadre d'une méthode de coupes, au lieu d'inclure l'ensemble de ces inégalités à priori dans la formulation ?

6 Conclusion

Dans cet article, notre objectif principal était de présenter une revue de littérature sur les IV pertinentes au problème de conception de réseaux multiproduits avec coût fixe et capacité, ou PCMC. Dans un premier temps, nous avons présenté une formulation du PCMC, ainsi que les méthodes utilisées pour le résoudre. Par la suite, nous avons présenté trois grandes classes d'IV pertinentes au PCMC, et qui ont été largement étudiées pour des problèmes apparentés au PCMC.

À la lumière des résultats expérimentaux, nous avons montré que les implantations générales des IV dans un logiciel commercial sophistiqué (CPLEX) ne sont pas suffisantes pour améliorer significativement la qualité des solutions obtenues, surtout pour les instances de grande taille. Cependant, ces résultats confirment l'efficacité des approches polyédrales à améliorer la formulation du PCMC. Rappelons que, puisque les problèmes de séparation et d'élévation doivent être résolus par des heuristiques, une compréhension profonde du problème à résoudre, ainsi que des adaptations particulières des inégalités utilisées, s'avèrent essentielles afin de développer des méthodes efficaces. Dans ce contexte, une étude approfondie des propriétés polyédrales du PCMC paraît nécessaire.

Remerciements

Ce travail a été réalisé grâce au soutien financier du CRSNG (Canada), du Fonds FCAR (Québec) et du Réseau de Calcul et de Modélisation Mathématique (RCM2). Nous remercions M. Dominique Tourillon et Mme Geneviève Hernu pour leur aide concernant l'expérimentation numérique.

Bibliographie

- [1] Aardal K. 1998, *Capacitated Facility Location : Separation Algorithms and Computational Experience*, Mathematical Programming 81, 149-175.
- [2] Aardal K., Pochet Y. et Wolsey L.A. 1995, *Capacitated Facility Location : Valid Inequalities and Facets*, Mathematics of Operations Research 20, 562-582.
- [3] Aardal K. et Weismantel R. 1997, *Polyhedral Combinatorics, Annotated Bibliographies in Combinatorial Optimization*, Dell'Amico M, Maffioli F., Mortello S. (éds.), Wiley-Interscience, 31-44.
- [4] Ahuja R.K., Magnanti T.L., Orlin J.B. et Reddy M.R. 1995, *Applications of Network Optimization*, chapitre 1 du *Network Models*, Handbooks in Operations Research and Management Science 7, 1-83.
- [5] Atamtürk A. 2000, *On Capacitated Network Design Cut-Set Polyhedra*, Technical Report, University of California, Berkeley, California.
- [6] Balakrishnan A. 1984, *Valid Inequalities and Algorithms for the Network Design Problem with an Application to LTL Consolidation*, Ph.D. Dissertation, MIT Sloan School of Management, Cambridge, Mass.
- [7] Balakrishnan A., Magnanti T.L. et Mirchandani P. 1997, *Network Design, Annotated Bibliographies in Combinatorial Optimization*, Dell'Amico M, Maffioli F., Mortello S. (éds.), Wiley-Interscience, 311-334.
- [8] Balakrishnan A., Magnanti T.L. et Wong R.T. 1989, *A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design*, Operations Research 37, 716-740.
- [9] Balas E. 1975, *Facets of the Knapsack Polytope*, Mathematical Programming 8, 146-164.

- [10] Balas E. et Zemel E. 1984, *Lifting and Complementing Yields all the Facets of Positive Zero-One Programming Polytopes*, Mathematical Programming, Proceedings of the International Conference on Mathematical Programming, Cottle R.W. et al. (éds.), 13-24.
- [11] Barahona F. 1996, *Network Design Using Cut Inequalities*, SIAM Journal of Optimization 6, 823-837.
- [12] Barany I, Van Roy T.J. et Wolsey L.A. 1984a, *Strong Formulation for Multi-Item Capacitated Lot-Sizing*, Management Science 30, 1255-1271.
- [13] Barany I, Van Roy T.J. et Wolsey L.A. 1984b, *Uncapacitated Lot Sizing : the Convex Hull of Solutions*, Mathematical Programming Study 22, 32-43.
- [14] Barnhart C., Hane C. et Vance P.H. 2000, *Using Branch-and-Price-and-Cut to solve Origin-Destination Integer Multicommodity Flow Problems*, Operations Research 48, 318-326.
- [15] Barnhart C., Jin H. et Vance P.H. 2000, *Railroad Bolcking : a Network Design Application*, Operations Research 48, 603-614.
- [16] Benders J.F. 1962, *Partitioning Procedures for Solving Mixed-Variables Programming Problems*, Numerische Mathematik 4, 238-252.
- [17] Bienstock D., Chopra S., Günlük O. et Tsai C.Y. 1998, *Minimum Cost Capacity Installation for Multicommodity Network Flows*, Mathematical Programming 81, 177-199.
- [18] Bienstock D. et Günlük O. 1996, *Capacitated Network Design-Polyedral Structure and Computation*, INFORMS Journal on Computing 8, 243-259.
- [19] Bienstock D. et Muratore G. 2000, *Strong Inequalities for Capacitated Survivable Network Design Problems*, Mathematical Programming 89, 127-147.
- [20] Bilde O. et Krarup J. 1977, *Sharp Lower Bounds and Efficient Algorithms for the Simple Plant Location Problem*, Annals of Discrete Mathematics 1, 79-97.
- [21] Caprara A. et Fischetti M. 1997, *Branch-and-Cut Algorithms, Annotated Bibliographies in Combinatorial Optimization*, Dell'Amico M, Maffioli F., Mortello S. (éds.), Wiley-Interscience, 31-44.

- [22] Chopra S., Gilboa I, et Sastry S.T. 1998, *Source Sink Flows with Capacity Installation in Batches*, Discrete Applied Mathematics 85, 165-192.
- [23] Chvátal V. 1973, *Edmonds Polytopes and a Hierarchy of Combinatorial Problems*, Discrete Mathematics 4, 305-337.
- [24] Conn A.R. et Cornuéjols G. 1990, *A projection Method for the Uncapacitated Facility Location Problem*, Mathematical Programming 46, 273-298.
- [25] Cornuéjols G., Nemhauser G.L. et Wolsey L.A. 1990, *The Uncapacitated Facility Location Problem*, in *Discrete Location Theory*, Mirchandani P.B. et Francis R.L. (éds), Wiley-Interscience, 119-171.
- [26] Cornuéjols G., Sridharan R. et Thizy J.M. 1991, *A Comparison of Heuristics and Relaxations for the Capacitated Plant Location Problem*, European Journal of Operations Research 50, 280-297.
- [27] CPLEX Optimization, Inc. 2000, *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, version 7.0.
- [28] Crainic T.G., Frangioni A. et Gendron B. 2001, *Bundle-Based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems*, à paraître dans Discrete Applied Mathematics.
- [29] Crainic T.G., Gendreau M. et Farvolden J. 2000, *A Simplex-Based Tabu Search Method for Capacitated Network Design*, INFORMS Journal on Computing 12, 223-236.
- [30] Crainic T.G. et Laporte G. 1997, *Planning Models for Freight Transportation*, European Journal of Operation Research 97, 409-438.
- [31] Crowder H., Johnson E.L. et Padberg M.W. 1983, *Solving Large Scale Zero-One Linear Programming Problems*, Operations Research 31, 803-834.
- [32] Epstein R. 1998, *Linear Programming and Capacitated Network Loading*, Ph.D. Dissertation, MIT Sloan School of Management, Cambridge, Mass.
- [33] Erlenkotter D. 1978, *A Dual-Based Procedure for Uncapacitated Facility Location*, Operations Research 26, 992-1009.

- [34] Gabrel V., Knippel A. et Minoux M. 1999, *Exact Solution of Multicommodity Network Optimization Problems with General Step Cost Functions*, Operations Research Letters, vol. 25, 15-23.
- [35] Gendron B. et Crainic T.G. 1994, *Relaxations for Multicommodity Capacitated Network Design Problems*, Publication CRT-96-05, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [36] Gendron B. et Crainic T.G. 1996, *Bounding Procedures for Multicommodity Capacitated Fixed Charge Network Design Problems*, Publication CRT-96-06, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [37] Gendron B., Crainic T.G. et Frangioni A. 1998, *Multicommodity Capacitated Network Design, Telecommunications Network Planning*, Sansó B., Soriano P. (éds.), Kluwer Academic Publishers, Norwell MA., 1-19.
- [38] Ghamlouche I., Crainic T.G. et Gendreau M. 2001, *Cycle-Based Neighborhoods for Fixed-Charge Capacitated Multicommodity Network Design*, Publication CRT-2000-01, Centre de Recherche sur les Transports, Université de Montréal, Montréal.
- [39] Gomory R.E. 1958, *Outline of an Algorithm for Integer Solutions to Linear Programs*, Bulletin of the American Mathematical Society 64, 275-278.
- [40] Gomory R.E. 1969, *Some Polyhedra Related to Combinatorial Problems*, Linear Algebra and Its Applications 2, 451-558.
- [41] Gu Z., Nemhauser G.L. et Savelsbergh M.W.P. 1998, *Lifted Cover Inequalities for 0-1 Integer Programs : Computation*, INFORMS Journal on Computing 10, 427-437.
- [42] Gu Z., Nemhauser G.L. et Savelsbergh M.W.P. 1999, *Lifted Covers Inequalities for 0-1 Integer Programs : Complexity*, INFORMS Journal on Computing 11, 117-123.
- [43] Gu Z., Nemhauser G.L. et Savelsbergh M.W.P. 1999a, *Lifted Flow Cover Inequalities for Mixed 0-1 Integer Programs*, Mathematical Programming 85, 439-467.
- [44] Günlük O. 1999, *A Branch-and-Cut Algorithm for Capacitated Network Design Problems*, Mathematical Programming 86, 17-39.

- [45] Hellstrand J., Larsson T. et Migdalas A. 1992, *A Characterization of the Uncapacitated Network Design Polytope*, Operations Research Letters 12, 159-163.
- [46] Hernu G. 2001, *Heuristique basée sur la programmation mathématique pour le problème de conception de réseau avec coût fixe et capacité*, Mémoire de Maitrise, CRT-2001-11, Centre de Recherche sur les Transports, Université de Montréal, Montréal.
- [47] Holmberg K. et Hellstrand J. 1998, *Solving the Uncapacitated Network Design Problem by a Lagrangean Heuristic and Branch-and-Bound*, Operations Research 46, 247-259.
- [48] Holmberg K. et Yuan D. 2000, *A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem*, Operations Research 48, 461-481.
- [49] Krarup J. et Pruzan P.M. 1983, *The Simple Plant Location Problem : Survey and Synthesis*, European Journal of Operation Research 12, 36-81.
- [50] Leung J.M.Y. et Magnanti T.L. 1989, *Valid Inequalities and Facets of the Capacitated Plant Location Problems*, Mathematical Programming 44, 271-291.
- [51] Magnanti T.L. et Mirchandani P.B. 1993, *Shortest Paths, Single Origine- Destination Network Design, and Associated Polyhedra*, Networks 23, 103-121.
- [52] Magnanti T.L., Mirchandani P.B. et Vachani R. 1993, *The Convex Hull of Two Core Capacitated Network Design Problems*, Mathematical Programming 60, 233-250.
- [53] Magnanti T.L., Mirchandani P.B. et Vachani R. 1995, *Modeling and Solving the Two-Facility Capacitated Network Loading Problem*, Operations Research 43, 142-157.
- [54] Magnanti T.L. et Wong R.T. 1984, *Network Design and Transportation Planning : Models and Algorithms*, Transportation Science 18, 1-55.
- [55] Minoux M. 1989, *Network Synthesis and Optimum Network Design Problems : Models, Solution Methods and Applications*, Networks 19, 313-360.

- [56] Mirchandani P. 1989, *Polyhedral Structure of a Capacitated Network Design Problem with an Application to the Telecommunication Industry*, Ph.D. Dissertation, MIT Sloan School of Management, Cambridge, Mass.
- [57] Mirchandani P.B. et Francis R.L. 1990, *Discrete Location Theory*, Wiley-Interscience.
- [58] Nemhauser G.L., Savelsbergh M.W.P. et Sigismondi G. 1994, *MINTO, a Mixed Integer Optimizer*, Operations Research Letters 15, 47-58.
- [59] Nemhauser G.L. et Wolsey L.A. 1988 *Integer and Combinatorial Optimization*, Wiley-Interscience.
- [60] Padberg M.W. 1973, *On the Facial Structure of Set Packing Polyhedra*, Mathematical Programming 5, 199-215.
- [61] Padberg M.W. et Rinaldi G. 1987, *Optimization of 532-city Symmetric Traveling Salesman Problem by Branch-and-Cut*, Operations Research Letters 6, 1-7.
- [62] Padberg M.W., Van Roy T.J. et Wolsey L.A. 1985, *Valid Linear Inequalities for Fixed Charge Problems*, Operations Research 33, 842-861.
- [63] Rardin R.L. et Wolsey L.A. 1993, *Valid Inequalities and Projecting the Multi-commodity Extended Formulation for Uncapacitated Fixed Charge Network Flow Problems*, European Journal of Operation Research 71, 95-109.
- [64] Schrijver A. 1980, *On Cutting Planes*, Annals of Discrete Mathematics 9, 291-296.
- [65] Stallaert J. 2000, *Valid Inequalities and Separation for Capacitated Fixed Charge Flow Problems*, Discrete Applied Mathematics 98, 265-274.
- [66] Van Roy T.J. et Wolsey L.A. 1986, *Valid Inequalities and Separation for Uncapacitated Fixed Charge Networks*, Operations Research Letters 4, 105-112.
- [67] Van Roy T.J. et Wolsey L.A. 1986a, *Valid Inequalities for Mixed 0-1 Programs*, Discrete Applied Mathematics 14, 199-213.
- [68] Van Roy T.J. et Wolsey L.A. 1987, *Solving Mixed Integer Programming Problems Using Automatic Reformulation*, Operations Research 35, 45-57.

- [69] Weismantel R. 1997, *On the 0-1 Knapsack Polytope*, Mathematical Programming 77, 49-68.
- [70] Wolsey L.A. 1975, *Faces of Linear Inequalities in 0-1 variables*, Mathematical Programming 8, 165-178.
- [71] Wolsey L.A. 1976, *Facets and Strong Valid Inequalities for Integer Programs*, Operations Research 24, 367-372.
- [72] Wolsey L.A. 1977, *Valid Inequalities and Superadditivity for 0-1 Integer Programs*, Mathematics of Operations Research 2, 66-77.
- [73] Wolsey L.A. 1989, *Submodularity and Valid Inequalities in Capacitated Fixed Charge Networks*, Operations Research Letters 8, 119-124.
- [74] Xpress-MP 1999, *Xpress-MP Integrated into MIMI*, Modeling and Optimization Software, Version 12, Dash Optimization Inc., New York, NY.
- [75] Zemel E. 1978, *Lifting the Facets of 0-1 Polytopes*, Mathematical Programming 15, 268-277.

Annexe : Techniques de base pour la dérivation d'IV

À l'origine, les études des propriétés polyédrales ont porté sur des problèmes de programmation en nombres entiers seulement ; définissons les ensembles $P = \{y \in \mathbb{Z}_+^n : Ay \leq b\}$ et $R_P = \{y \in \mathbb{R}_+^n : Ay \leq b\}$ (la relaxation continue de P).

Technique de combinaison linéaire : L'idée de base est simple. Si un ensemble de contraintes est valide pour R_P , alors toute combinaison linéaire non négative de ces contraintes est valide pour R_P . Comme $Ay \leq b$ est valide pour R_P alors

$$uAy \leq ub, \quad (20)$$

est encore valide pour R_P pour tout $u \geq 0$. Comme $P \subset R_P$ les inégalités (20) sont valides pour P .

Cette technique n'est pas suffisante pour générer des inégalités qui soient valides pour P mais non pour R_P . Pour ce faire, il faut utiliser la propriété d'intégralité de P .

Technique d'arrondissement : Cette technique est basée sur un principe très simple :

si $a \leq b$ et a est entier alors $a \leq \lfloor b \rfloor$, où $\lfloor b \rfloor$ est le plus grand entier plus petit que b .

En utilisant les techniques que nous venons de présenter, nous pouvons facilement comprendre et démontrer la validité des coupes de Gomory (1958) bien connues en programmation en nombres entiers. Ces coupes sont conçues pour des problèmes ayant des domaines réalisables bornés ainsi que des coefficients entiers dans A et b . La coupe de Gomory est définie sur le modèle standard obtenu de l'original en introduisant les variables d'écart $(y_{n+1}, y_{n+2}, \dots, y_{n+m})$. Donc, soit $P^e = \{y \in \mathbb{Z}_+^{n+m} : (A, I)y = b\}$ la formulation standard du problème. Soit $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ un vecteur de poids. On sait que $\lambda(A, I)y = \lambda b$ est valide pour P^e , alors la coupe de Gomory définie par :

$$\sum_{j \in N} (\bar{a}_j - \lfloor \bar{a}_j \rfloor) y_j + \sum_{i \in M} (\lambda_i - \lfloor \lambda_i \rfloor) y_{n+i} \geq \bar{b} - \lfloor \bar{b} \rfloor,$$

est valide pour P^e , où $\lambda a_j = \bar{a}_j, \forall j \in N$ et $\lambda b = \bar{b}$.

Chvátal (1973), après avoir développé la technique d'arrondissement, a généralisé

la coupe de Gomory pour traiter les cas où les coefficients de A et b sont des nombres réels. L'inégalité obtenue est connue sous le nom d'inégalité de Chvátal-Gomory. Elle est définie par :

$$\sum_{j \in N} \lfloor ua_j \rfloor y_j \leq \lfloor ub \rfloor, \quad \forall u \geq 0.$$

Schrijver (1980) a généralisé l'inégalité de Chvátal-Gomory pour traiter les cas où les coefficients de A et b sont des nombres rationnels et le domaine réalisable n'est pas nécessairement borné.

Technique d'élévation séquentielle (ou "lifting") : Cette technique permet d'aug-

menter la dimension d'une face (IV) de P en essayant d'impliquer dans l'inégalité toutes les variables possibles. L'idée de l'élévation séquentielle est introduite initialement par Gomory (1969) et étudiée expérimentalement par Padberg (1973). Ensuite, plusieurs chercheurs ont généralisé cette approche : Wolsey (1976), Zemel (1978) et Balas et Zemel (1984). Bien que cette approche soit très importante et indispensable pour la qualité de certaines IV connues, elle est très coûteuse et même dans certains cas, très compliquée à mettre en oeuvre. En effet, l'implication des variables qui ne figurent pas dans l'IV se fait d'une façon itérative. Les variables sont traitées séquentiellement dans un certain ordre. Nous présentons dans les propositions 9 et 10 deux approches différentes d'élévation tirées de Nemhauser et Wolsey (1988) pour calculer les coefficients d'élévation nommés α_1 et γ_1 .

Proposition 9 : Soit $P \subset \{0, 1\}^n$ et $P^\delta = P \cap \{y \in \{0, 1\}^n : y_1 = \delta\}$ pour $\delta \in \{0, 1\}$ et soit

$$\sum_{j=2}^n \pi_j y_j \leq \pi_0,$$

une IV de P^0 , alors :

- si $P^1 = \phi$, alors $y_1 \leq 0$ est une IV pour P ;
- si $P^1 \neq \phi$, posons $\zeta = \max\{\sum_{j=2}^n \pi_j y_j : y \in P^1\}$, et alors

$$\alpha_1 y_1 + \sum_{j=2}^n \pi_j y_j \leq \pi_0,$$

est une IV pour P , pour tout $\alpha_1 \leq \pi_0 - \zeta$.

En plus, si $\alpha_1 = \pi_0 - \zeta$ et $\{y \in P : \sum_{j=2}^n \pi_j y_j = \pi_0\}$ est une face de dimension k de $\text{conv}(P^0)$ alors $\{y \in P : \alpha_1 y_1 + \sum_{j=2}^n \pi_j y_j = \pi_0\}$ est une face de dimension $(k + 1)$ de $\text{conv}(P)$.

Proposition 10 : Soit $P \subset \{0, 1\}^n$ et $P^\delta = P \cap \{y \in \{0, 1\}^n : y_1 = \delta\}$ pour $\delta \in \{0, 1\}$ et soit

$$\sum_{j=2}^n \pi_j y_j \leq \pi_0,$$

une IV de P^1 , alors :

- si $P^0 = \phi$, alors $y_1 \geq 1$ est une IV pour P ;
- si $P^0 \neq \phi$, posons $\zeta = \max\{\sum_{j=2}^n \pi_j y_j : y \in P^0\}$, et alors

$$\gamma_1 y_1 + \sum_{j=2}^n \pi_j y_j \leq \pi_0 + \gamma_1,$$

est une IV pour P , pour tout $\gamma_1 \geq \zeta - \pi_0$.

En plus, si $\gamma_1 = \zeta - \pi_0$ et $\{y \in P : \sum_{j=2}^n \pi_j y_j = \pi_0\}$ est une face de dimension k de $\text{conv}(P^1)$ alors $\{y \in P : \gamma_1 y_1 + \sum_{j=2}^n \pi_j y_j = \pi_0 + \gamma_1\}$ est une face de dimension $(k + 1)$ de $\text{conv}(P)$.

La difficulté majeure de cette approche est que le calcul de chaque coefficient d'élévation requiert la résolution d'un problème d'optimisation différent, entier ou MIP dépendamment de l'IV. Ce dernier correspond au problème de maximisation pour calculer la valeur ζ . Dans la plupart des cas, ces problèmes sont extrêmement difficiles à résoudre, ce qui nous oblige à utiliser des heuristiques et des méthodes de relaxation. En plus, l'ordre dans lequel les variables sont traitées affecte énormément la qualité de la coupe obtenue. En effet, les variables traitées en premier auront des coefficients d'élévation plus grands. Ainsi, différents ordres donnent différentes inégalités, et il n'y a pas une méthode systématique simple permettant d'obtenir le meilleur ordre à considérer d'où l'utilisation des heuristiques. Un moyen d'alléger

l'effort demandé pour accomplir l'élévation serait d'utiliser la technique d'élévation séquentielle indépendante (Wolsey 1977). Cette dernière est basée sur le principe suivant : si la fonction d'élévation, f , est superadditive (f bornée sur Y et $f(y_1) + f(y_2) \leq f(y_1 + y_2), \forall y_1, y_2, y_1 + y_2 \in Y$) alors l'élévation peut se faire en résolvant un seul problème d'optimisation pour calculer tous les coefficients de l'élévation (la fonction d'élévation étant la fonction objective dans le problème de maximisation pour calculer les coefficients d'élévation). Bien que cette théorie soit intéressante, le problème est que la plupart des fonctions d'élévation rencontrées en pratique n'ont pas la propriété de superadditivité. Pour contourner cet obstacle, plusieurs auteurs utilisent des fonctions superadditives qui dominent la fonction d'élévation ; ainsi il est possible de calculer des coefficients d'élévation, plus ou moins exacts, sans trop d'effort.

Chapitre 3

**A cutting-plane algorithm based
on cutset inequalities for
multicommodity capacitated fixed
charge network design**

A cutting-plane algorithm based on cutset inequalities for multicommodity capacitated fixed charge network design

Mervat Chouman, Teodor Gabriel Crainic
and Bernard Gendron

Abstract

The multicommodity capacitated fixed charge network design problem is a well-known NP-hard problem which arises in a wide variety of applications, most notably in transportation and telecommunications. In this paper, we propose to improve the mixed integer programming (MIP) formulation of the problem by using a polyhedral approach. We present a cutting-plane algorithm which incorporates valid inequalities (VI) based on cutsets of the network. We identify and study, in the field of cutset type inequalities, three families of VI : the well-known cover inequalities, the minimum cardinality inequalities, and the network cutset inequalities. We develop efficient separation heuristics and lifting procedures, as well as a cutset generation algorithm. We present computational results on a large set of instances of various characteristics. Our results show that the cutting-plane algorithm achieves better lower bounds with competitive computational effort when compared to the state-of-the-art MIP software package CPLEX.

Key words : multicommodity capacitated fixed-charge network design, polyhedral approach, cutting-plane algorithm, valid inequalities, separation, lifting.

Résumé

Le problème de conception de réseaux multiproduits avec coût fixe et capacité est un problème NP-difficile apparaissant dans une grande variété d'applications, notamment en transport et en télécommunications. Dans cet article, nous proposons d'améliorer la formulation de programmation en nombres entiers (PNE) du problème en utilisant une approche polyédrale. Nous présentons un algorithme de coupes qui intègre des

inégalités valides basées sur des coupes du réseau. Parmi ces inégalités, nous identifions et étudions trois familles d'inégalités : les inégalités de couverture, les inégalités de cardinalité minimale et les inégalités de coupes du réseau. Nous développons des heuristiques de séparation et des procédures d'élévation séquentielle, de même qu'un algorithme de génération de coupes du réseau. Nous présentons des résultats expérimentaux obtenus sur un grand nombre de problèmes ayant différentes caractéristiques. Nos résultats montrent que l'algorithme de coupes obtient de meilleures bornes dans des temps de calcul compétitifs lorsqu'on le compare aux résultats obtenus au moyen du logiciel de PNE CPLEX.

Mots-clés : problème de conception de réseaux multiproduits avec coût fixe et capacité, approche polyédrale, algorithme de coupes, inégalités valides, séparation, élévation séquentielle.

1 Introduction

Network design formulations are used to model a wide variety of applications, most notably in the fields of transportation and telecommunications (Magnanti and Wong 1984, Minoux 1989, Crainic 1999). These formulations are often characterized as follows : given a network with arc capacities, it is required to send flows in order to satisfy, at minimum cost, known demands between origin-destination pairs. In doing so, one pays a price not only for transporting flows, but also for using arcs, in terms of fixed costs. The goal of the resulting multicommodity capacitated fixed charge network design problem (MCND) is to determine the optimal amounts of flows to be transported and the arcs to be used.

The MCND is modeled as a mixed integer program (MIP) where continuous variables represent the flows and 0-1 variables are used to model the design decisions (the arcs to be used). The MCND is NP-hard since it contains as a special case the uncapacitated fixed charge network design problem, which is NP-hard as well (Magnanti and Wong 1984). In addition, considerable algorithmic challenges are imposed when solving realistically-sized problem instances. These challenges are not only due to the large size of real applications but also to the trade-off between variable and fixed costs, as well as to the competition among commodities due to the limited capacity on the arcs.

Branch-and-bound (B&B) algorithms based on linear programming (LP) relaxations are the most common tools to solve MIP problems. An effective B&B implementation for the MCND requires to improve the lower bound provided by the LP relaxation, which generally yields a poor approximation of the MIP convex hull of solutions (Gendron, Crainic, and Frangioni 1998). Gendron and Crainic (1994) proposed to improve the lower bounds by adding the so-called *strong inequalities* (presented in Section 2), which are redundant for the MIP formulation but not for its LP relaxation. Unfortunately, solving the resulting strong LP relaxation requires considerable computational time since the formulation is very large and frequently exhibits degeneracy. This seriously impacts the overall performance of the B&B algorithm. For

example, the state-of-the-art MIP software CPLEX (version 7.1, 2002) requires more than 10 hours of computation on a Sun Enterprise 10000 to solve only the strong LP relaxation of instances with 400 commodities, 700 0-1 variables and 280 000 continuous variables.

Alternative relaxation approaches have been devised for solving the MCND, in particular Lagrangean-based procedures (Gendron and Crainic 1994, 1996, Holmberg and Yuan 2000, Gendron, Crainic, and Frangioni 1998, Crainic, Frangioni, and Gendron 2002, and Sellmann, Kliewer, and Koberstein 2002). Several Lagrangean relaxations have been used in these procedures, and all of them yield the same theoretical lower bound as the strong LP relaxation (Gendron and Crainic 1994). Heuristic strategies have also been proposed for computing feasible solutions (Crainic, Gendreau, and Farvolden 2000, Ghamlouche, Crainic, and Gendreau 2001, 2002, Crainic, Gendron, and Hernu 2002, Crainic and Gendreau 2002). To assess the quality of the solutions produced by these heuristics, optimal solutions must be known. However, to this date, even tight lower bounds cannot be computed in a reasonable amount of time for a large set of instances.

Tightening up the formulation of the problem is thus of crucial importance for the development of efficient solution methods. This might be accomplished using the polyhedral approach, which attempts to improve the formulation by adding *valid inequalities* (or *VI*) in the form of linear constraints. This is usually performed within the framework of simplex-based cutting-plane methods. The polyhedral approach has proven effective to solve network design problems related to the MCND, such as the *network loading problem* (Magnanti, Mirchandani and Vachani 1993, 1995, Barahona 1996, Bienstock and Günlük 1996, Bienstock *et al.* 1998, Günlük 1999), and the *capacitated facility location problem* (Leung and Magnanti 1989, Aardal, Pochet and Wolsey 1995, Aardal 1998). To the best of our knowledge, however, no systematic study of the polyhedral structure of the MCND has been performed yet.

Several VI derived for particular structures of the MCND are well-known, especially the *cover* and *flow cover* inequalities (Nemhauser and Wolsey 1988). These inequa-

lities, among others, are incorporated in Branch-and-Cut (B&C) algorithms implemented in standard MIP software packages, such as MPSARX (Van Roy and Wolsey 1987), MINTO (Nemhauser, Savelsbergh and Sigismondi 1994), Xpress-MP (2002) and CPLEX (2002). Chouman, Crainic and Gendron (2001) have observed that the B&C method of CPLEX (version 7.1, 2002) cannot solve large size instances of the problem in a reasonable amount of time. With more than 10 hours of CPU time allowed for each instance on a Sun Enterprise 10000, the authors have noticed that CPLEX could generate flow cover inequalities, but was unable to generate any cover inequality.

In this paper, we present a cutting-plane procedure which incorporates VI based on *cutsets* of the network. These inequalities are based on the following idea : in any feasible solution, the capacity of the arcs of any cutset that cuts off some origins from their destinations must be sufficient to flow the demand across the cutset. We are motivated by previous studies that prove the effectiveness of such inequalities for solving problems related to the MCND, in particular the network loading problem. We identify and study, in the field of cutset type inequalities, three families of VI : the well-known *cover inequalities*, the *minimum cardinality inequalities*, and the *network cutset inequalities*. The last one is, to the best of our knowledge, a new family of VI, introduced for the first time. The study of cover inequalities is motivated by the fact that the general implementation of these inequalities in CPLEX was unable to detect the knapsack structure needed for their generation. To evaluate the performance of the proposed cutting-plane procedure, we perform a computational study on a large set of test problems studied in the literature.

The main contribution of this paper is the development of computationally efficient separation and lifting procedures for these families of VI. We have also developed an efficient heuristic to identify cutsets of the network on which cuts can be generated. Our computational results show that the three families of VI are useful for the MCND. In addition, when compared to the state-of-the-art LP software CPLEX, our cutting-plane procedure shows a remarkable improvement in the time needed to compute the lower bounds.

This paper is organized as follows. In the next section, we present a MIP formulation of the MCND and some basic VI. Sections 3 and 4 are dedicated to the study of two cutset type inequalities, the cover inequalities and the minimum cardinality inequalities, respectively, and to their relationship. The network cutset inequalities are presented in Section 5. Section 6 describes the cutset generation algorithm and the cutting-plane procedure. Computational results are presented in Section 7. We conclude this paper by proposing future research related to this work.

2 Formulation and Basic Valid Inequalities

The multicommodity capacitated fixed charge network design problem (MCND) is characterized using the following notation. Given a directed graph $G = (N, A)$, where N is the set of nodes and A is the set of arcs, and a set of commodities (or origin-destination pairs) K to be routed according to a known demand d^k for each commodity k , the problem is to satisfy the demand at minimum cost. The cost consists of the sum of transportation costs and fixed design costs, the latter being charged whenever an arc is used. The transportation cost per unit of commodity k on arc (i, j) is denoted $c_{ij}^k \geq 0$, while the fixed design cost for arc (i, j) is denoted $f_{ij} \geq 0$. An origin $O(k)$ and a destination $D(k)$ are associated to each commodity k . We introduce continuous flow variables x_{ij}^k , which reflect the amount of flow on each arc (i, j) for each commodity k , and 0-1 design variables y_{ij} , which indicate if arc (i, j) is used or not. With this notation, we write the MIP formulation of the MCND as follows :

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij}, \quad (1)$$

$$\sum_{j \in N_i^+} x_{ij}^k - \sum_{j \in N_i^-} x_{ji}^k = \begin{cases} d^k, & \text{if } i = O(k), \\ -d^k, & \text{if } i = D(k), \\ 0, & \text{otherwise,} \end{cases} \quad \forall i \in N, \forall k \in K, \quad (2)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i, j) \in A, \quad (3)$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K, \quad (4)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A, \quad (5)$$

where $N_i^+ = \{j \in N \mid (i, j) \in A\}$ and $N_i^- = \{j \in N \mid (j, i) \in A\}$.

Relations (2) correspond to flow conservation constraints for each node and each commodity. Relations (3) represent capacity constraints for each arc. They also link together flow and design variables by forbidding any flow to pass through an arc that is not chosen as part of the design.

The LP relaxation is obtained by replacing the integrality constraints (5) by

$$0 \leq y_{ij} \leq 1, \quad \forall (i, j) \in A.$$

Gendron, Crainic, and Frangioni (1998) show that the lower bounds derived from this LP relaxation are generally weak. In order to improve these lower bounds, provided by the so-called *weak LP relaxation*, the following VI, called *strong inequalities*, can be added to obtain the *strong LP relaxation* :

$$x_{ij}^k \leq d^k y_{ij}, \quad \forall (i, j) \in A, k \in K. \quad (6)$$

Adding the strong inequalities to the model significantly improves the quality of the LP lower bounds (Gendron, Crainic, and Frangioni 1998). In order to derive other useful VI for the MCND, we are interested in particular structures that might be hidden in relaxations of the model.

By aggregating the capacity constraints (3) and the flow conservation equations (2), we obtain the following constraints :

$$\sum_{j \in N_i^+} u_{ij} y_{ij} \geq \sum_{k \in K \mid i=O(k)} d^k, \quad \forall i \in N. \quad (7)$$

In order to generalize inequality (7), let $S \subset N$ be any non empty subset of N and its complement $\bar{S} = N \setminus S$, and let $K(S, \bar{S}) \subseteq K$ be the non empty set of commodities having their origin in S and their destination in \bar{S} . Then, summing the inequality (7) over all nodes in S gives

$$\sum_{(i,j) \in (S, \bar{S})} u_{ij} y_{ij} \geq d_{(S, \bar{S})}, \quad (8)$$

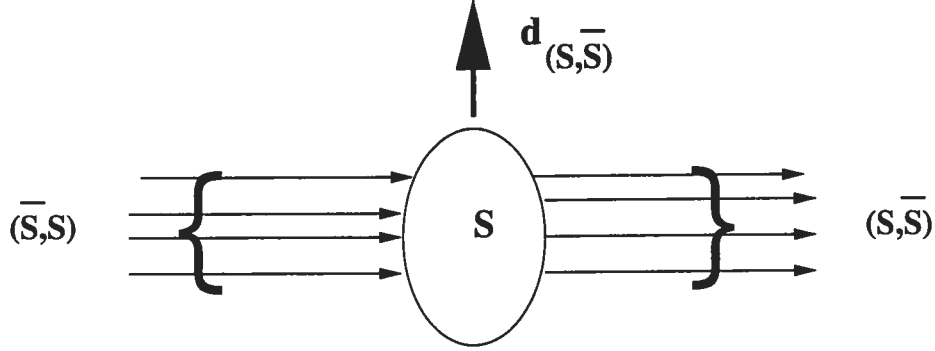


FIG. 1 - Cutset structure

where $d_{(S,\bar{S})} = \sum_{k \in K(S,\bar{S})} d^k > 0$ and (S, \bar{S}) is the set of arcs that connect a node in S to a node in \bar{S} (a *cutset*). The resulting inequality (8) is the well-known cutset inequality which states that the capacity used on the arcs of the cutset (S, \bar{S}) must be sufficient to satisfy the demand across the cutset. Consequently, the cutset inequality is a VI for the MCND for any $S \subset N$. Although the cutset inequality is redundant for the LP relaxation of the MCND, some VI derived from it might improve the LP relaxation.

We can derive useful VI from the cutset structure by using two different ideas. The first idea consists in complementing the y variables (replacing y_{ij} by $1 - y_{ij}$) in the cutset inequality to obtain the knapsack structure given by :

$$\sum_{(i,j) \in (S,\bar{S})} u_{ij} y_{ij} \leq \sum_{(i,j) \in (S,\bar{S})} u_{ij} - d_{(S,\bar{S})}.$$

Then, the well-known *cover inequalities* and the *minimum cardinality inequalities*, derived from this knapsack structure, are VI for any cutset, and thus for the MCND. Note that these two families of VI are not redundant in the LP relaxation of the MCND since they are based on the integrality of the y variables. The second idea, used to derive the *network cutset inequalities*, is based on the insertion of the flow variables x into the cutset inequalities.

These families of VI are described in the next three sections. We assume through these sections that a cutset (S, \bar{S}) is given. Heuristics used to generate such cutsets are presented in Section 6.

3 Cover Inequalities

Consider the set P_S induced by the given cutset (S, \bar{S}) ,

$$P_S = \{y \in \{0, 1\}^m : \sum_{(i,j) \in (S, \bar{S})} u_{ij} y_{ij} \geq d_{(S, \bar{S})}\},$$

where $m = |(S, \bar{S})|$. As mentioned above, by complementing the y variables in P_S , one obtains the 0-1 knapsack set. Then, the cover inequalities, developed independently by Balas (1975) and Wolsey (1975), constitute an interesting family of VI for P_S . We now adapt to P_S the standard definitions related to cover inequalities.

Definition 1 : A set $C \subseteq (S, \bar{S})$ is a cover if the total capacity of arcs in $(S, \bar{S}) \setminus C$ does not cover the demand : $\sum_{(i,j) \in (S, \bar{S}) \setminus C} u_{ij} < d_{(S, \bar{S})}$.

Definition 2 : A cover $C \subseteq (S, \bar{S})$ is minimal if it is sufficient to open any arc in C to cover the demand : $\sum_{(i,j) \in (S, \bar{S}) \setminus C} u_{ij} + u_{pq} \geq d_{(S, \bar{S})}, \forall (p, q) \in C$.

For every cover $C \subseteq (S, \bar{S})$, the *cover inequality* (CI) defined as

$$\sum_{(i,j) \in C} y_{ij} \geq 1 \tag{9}$$

is valid for P_S , for any subset S . Thus, it is valid for the MCND. The basic idea behind this inequality is that one has to open at least one arc from the set C in order to meet the demand. In addition, it has been proven (Nemhauser and Wolsey 1988) that if C is a minimal cover, then the CI is a facet of

$$\text{conv}(P_S \cap \{y \in \{0, 1\}^m : y_{ij} = 1, \forall (i, j) \in (S, \bar{S}) \setminus C\}).$$

However, the CI (9) is not a facet of $\text{conv}(P_S)$ in general and it has to be strengthened using the lifting procedure (see Section 3.2 below). In order to generate a violated CI, given that the current LP solution is fractional, one has to determine a cover C by solving the separation problem for a given cutset (S, \bar{S}) .

3.1 Separation

Let \bar{y} be a fractional optimal solution for some LP relaxation of the MCND. The separation problem of the CI consists in finding the most violated cover C , if there is one, by solving the following optimization problem :

$$Z_{sep} = \min \begin{array}{l} \sum_{(i,j) \in (S, \bar{S})} Z_{ij} \bar{y}_{ij}, \\ \sum_{(i,j) \in (S, \bar{S})} Z_{ij} u_{ij} > \sum_{(i,j) \in (S, \bar{S})} u_{ij} - d_{(S, \bar{S})}, \\ Z_{ij} \in \{0, 1\}, \forall (i, j) \in (S, \bar{S}), \end{array}$$

where Z is the characteristic vector of the set C ($Z_{ij} = 1$ then $(i, j) \in C$). Let Z_{sep} be the optimal objective value. If $Z_{sep} < 1$, then the constructed cover C identifies the most violated cover inequality. Otherwise, the current fractional solution \bar{y} does not violate any cover inequality.

This problem is a 0-1 knapsack problem. Even though there exists a pseudo-polynomial dynamic programming algorithm for solving exactly the 0-1 knapsack problem (Martello and Toth 1990), it is still time consuming especially because such problems must be solved repeatedly. We have implemented a faster heuristic approach that considers the arcs in non-decreasing order of $\frac{\bar{y}_{ij}}{u_{ij}}$, instead of $\frac{\bar{y}_{ij}}{u_{ij}}$, as would be performed by the classical greedy heuristic for the 0-1 knapsack problem. Ties are broken by considering the arcs in non-increasing order of their capacity. Once a cover is obtained with this heuristic, it is easy to extract a minimal cover from it, by removing some of the arcs from the cover until the condition in Definition 2 is satisfied. The basic idea of this heuristic is to try to exclude as much as possible from the set C the arcs with large \bar{y}_{ij} , in order to increase the chance of finding a violated inequality (i.e., $\sum_{(i,j) \in C} \bar{y}_{ij} < 1$). The same heuristic has been used by Gu, Nemhauser, and Savelsbergh (1998, 1999) in their extensive study of cover inequalities. Once the cover C is constructed, the induced inequality might be strengthened by the lifting procedure. Note that, even if the identified cover inequality is not violated, we might find a violated one through the lifting procedure (see Example 1 below).

3.2 Lifting

The principle of lifting is to extend a given VI in a higher dimensional space by including the variables that do not appear in the inequality. By doing so, we associate with each of these variables a so-called lifting coefficient. Lifting the CI can be performed using two different strategies. The first one consists in lifting down all variables in $(S, \bar{S}) \setminus C$ by seeking lifting coefficients, $\gamma_{ij} \geq 0$, such that the lifted cover inequality

$$\sum_{(i,j) \in (S, \bar{S}) \setminus C} \gamma_{ij} y_{ij} + \sum_{(i,j) \in C} y_{ij} \geq 1 + \sum_{(i,j) \in (S, \bar{S}) \setminus C} \gamma_{ij}$$

is a VI (see Example 1 below). This type of lifting contributes to the violation since $\gamma_{ij} y_{ij} \leq \gamma_{ij}$. Moreover, γ_{ij} can be chosen so that this inequality is facet-defining for $\text{conv}(P_S)$ when C is a minimal cover (Wolsey 1998).

We have implemented and tested this lifting strategy on a large set of instances. The experimental results show that this strategy is not useful for our instances. In particular, we have noted that many arcs $(i, j) \in C$ with a large capacity have a fractional \bar{y}_{ij} value close to zero at the current LP solution. In this case, if the corresponding cover inequality is not violated, this lifting strategy will not, in general, produce a violated inequality. We will illustrate this situation in Example 1 below.

Considering this observation, we present a second lifting strategy, which, to the best of our knowledge, is introduced for the first time. The basic idea is to determine, *a priori*, two subsets C_1 (the open arcs) and C_0 (the closed arcs) in (S, \bar{S}) that satisfy the condition

$$\sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} u_{ij} \geq d_{(S, \bar{S})} - \sum_{(i,j) \in C_1} u_{ij} > 0.$$

We then define the restricted set $P_{S_{(C_1, C_0)}}$ induced by C_1 and C_0 as

$$P_{S_{(C_1, C_0)}} = \{y \in \{0, 1\}^m : \sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} u_{ij} y_{ij} \geq d_{(S, \bar{S})} - \sum_{(i,j) \in C_1} u_{ij}\}.$$

We call *restricted CI* a cover inequality derived from this restricted set, with its corresponding cover denoted C . Since this inequality is restricted to open arcs in C_1

and closed arcs in C_0 , lifting (down for the variables in C_1 and up for the variables in C_0) is necessary to ensure global validity. In addition, by lifting down the variables in $(S, \bar{S}) \setminus (C_1 \cup C_0 \cup C)$, as in the first lifting strategy, we obtain the *lifted cover inequality* (LCI) :

$$\sum_{(i,j) \in (S, \bar{S}) \setminus C} \gamma_{ij} y_{ij} + \sum_{(i,j) \in C} y_{ij} \geq 1 + \sum_{(i,j) \in (S, \bar{S}) \setminus (C \cup C_0)} \gamma_{ij}. \quad (10)$$

Again, it is possible to choose the values for the lifting coefficients so that this inequality is facet-defining for $\text{conv}(P_S)$, when C is a minimal cover with respect to the restricted set. Note also that lifting down the variables in C_1 contributes to the violation of the inequality since $\gamma_{ij} y_{ij} \leq \gamma_{ij}$. However, lifting up the variables in C_0 has a negative impact on the violation in the sense that, an inequality that appeared violated prior to this lifting step might become satisfied after. This might happen if some variables in C_0 have positive values ($\bar{y}_{ij} > 0$) at the current LP solution. Thus, the arcs in C_0 must have small values at the current solution (\bar{y}_{ij} close to 0).

The choice of the sets C_1 and C_0 is of crucial importance for the performance of this lifting strategy. We propose the following procedure, called **OpenCloseArcs**, which uses the variables T and D representing, respectively, the residual capacity ($\sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} u_{ij}$) and the residual demand ($d_{(S, \bar{S})} - \sum_{(i,j) \in C_1} u_{ij}$). The procedure attempts to close an arc (i, j) with a small value \bar{y}_{ij} (as measured by a parameter ϵ_0) and such that the residual capacity ($T - u_{ij}$) still covers the residual demand D . Similarly, the procedure attempts to open an arc (i, j) with a large value \bar{y}_{ij} (as measured by a parameter ϵ_1) and such that there is still some residual demand to cover ($D - u_{ij} > 0$). The outline of this procedure is summarized next.

Procedure **OpenCloseArcs** :

- $T \leftarrow \sum_{(i,j) \in (S, \bar{S})} u_{ij}$, $D \leftarrow d_{(S, \bar{S})}$.
- For each arc $(i, j) \in (S, \bar{S})$ (in arbitrary order) :
 - If $(\bar{y}_{ij} < \epsilon_0)$ and $(T - u_{ij} \geq D)$ then
 - Add (i, j) to C_0 ;

- Close (i, j) by setting $T \leftarrow T - u_{ij}$;
- If $(\bar{y}_{ij} > \epsilon_1)$ and $(D - u_{ij} > 0)$ then
 - Add (i, j) to C_1 ;
 - Open (i, j) by setting $D \leftarrow D - u_{ij}$ and $T \leftarrow T - u_{ij}$.

Once the sets C_1 and C_0 are obtained, the lifting procedure is applied sequentially, meaning that the variables are lifted one after the other in some predetermined order. At a given step of the lifting procedure, suppose we are lifting variable y_{rt} . Then, let $L = \{(i, j) \in (S, \bar{S}) \setminus C, y_{ij} \text{ has been lifted}\}$, $\tilde{C}_1 = (S, \bar{S}) \setminus (C \cup C_0)$, $\gamma_{ij} = 1, \forall (i, j) \in C$ and $\bar{d} = d_{(S, \bar{S})} - \sum_{(i, j) \in \tilde{C}_1 \setminus L} u_{ij}$. To *lift down* variable $y_{rt} \in (S, \bar{S}) \setminus (C \cup C_0)$, we have to solve the following 0-1 knapsack problem :

$$Z_{opt} = \min \begin{array}{l} \sum_{(i, j) \in C \cup L} \gamma_{ij} y_{ij}, \\ \sum_{(i, j) \in C \cup L} u_{ij} y_{ij} \geq \bar{d} + u_{rt}, \\ y_{ij} \in \{0, 1\}, \forall (i, j) \in C \cup L. \end{array}$$

If this problem is feasible, the lifting coefficient is given by $\gamma_{rt} = Z_{opt} - 1 - \sum_{(i, j) \in L \setminus C_0} \gamma_{ij}$; otherwise, we set $\gamma_{rt} = \sum_{(i, j) \in C \cup L} \gamma_{ij} - \sum_{(i, j) \in L \setminus C_0} \gamma_{ij}$ (this is equivalent to $y_{rt} = 1$, which follows from the infeasibility of the problem). To *lift up* variable $y_{rt} \in C_0$, we have to solve the following 0-1 knapsack problem :

$$Z_{opt} = \min \begin{array}{l} \sum_{(i, j) \in C \cup L} \gamma_{ij} y_{ij}, \\ \sum_{(i, j) \in C \cup L} u_{ij} y_{ij} \geq \bar{d} - u_{rt}, \\ y_{ij} \in \{0, 1\}, \forall (i, j) \in C \cup L. \end{array}$$

If this problem is feasible, the lifting coefficient is given by $\gamma_{rt} = 1 + \sum_{(i, j) \in L \setminus C_0} \gamma_{ij} - Z_{opt}$; otherwise, we set $\gamma_{rt} = \sum_{(i, j) \in L \setminus C_0} \gamma_{ij} - \sum_{(i, j) \in C \cup L} \gamma_{ij}$ (this is equivalent to $y_{rt} = 0$).

Since the lifting coefficients have small values, the 0-1 knapsack problems are solved efficiently using a dynamic programming algorithm. Also, the quality of the resulting inequality depends on the lifting order. In particular, lifting down the variables in $(S, \bar{S}) \setminus C \cup C_0$ must be accomplished before lifting up the variables in C_0 (since the former contribute to the violation). Also, when lifting down the variables in

$(S, \bar{S}) \setminus C \cup C_0$, those with fractional values must be lifted first, in non-decreasing order of their current value. Ties are broken by considering first the arcs in non-increasing order of their capacity. When lifting up the variables in C_0 , we do the exact opposite. Example 1 illustrates the two lifting strategies just described.

Example 1 : Consider the following cutset inequality :

$$13y_1 + 7y_2 + 6y_3 + 4y_4 + 3y_5 + 11y_6 + 22y_7 \geq 22,$$

and let $\bar{y} = (0, 0.5, 0.5, 0.6, 0.7, 1, 0)$ be the current LP solution, restricted to the arcs in the cutset. Note that the inequality is satisfied at equality by this solution.

First strategy : Generate a CI and then lift down all variables in $(S, \bar{S}) \setminus C$: To generate the most violated cover we have to solve the following separation problem :

$$\begin{aligned} \min \quad & 0.5z_2 + 0.5z_3 + 0.6z_4 + 0.7z_5 + z_6, \\ & 13z_1 + 7z_2 + 6z_3 + 4z_4 + 3z_5 + 11z_6 + 22z_7 > 44, \\ & z_i \in \{0, 1\}, \quad \forall i = 1, \dots, 7. \end{aligned}$$

Using the heuristic described above, we find the following CI :

$$y_1 + y_2 + y_3 + y_7 \geq 1,$$

which is not violated. Note that this inequality corresponds to a minimal cover. Lifting down y_4, y_5 and y_6 in that order, according to the strategy described above, gives the same inequality since we find $\gamma_i = 0, i = 4, 5, 6$. In this case, the lifting does not improve the inequality, although it identifies a facet of $\text{conv}(P_S)$.

Second strategy : Let $\epsilon_1 = 0.6$ and $\epsilon_0 = 0.1$, so that $C_1 = \{5, 6\}$ and $C_0 = \{1, 7\}$; generate an LCI by first lifting down all variables in $S \setminus C \cup C_0$ and then lifting up all variables in C_0 : The resulting restricted set $P_{S(C_1, C_0)}$ is defined by the inequality

$$7y_2 + 6y_3 + 4y_4 \geq 8.$$

Applying the separation heuristic, we find the restricted CI induced by C_1 and C_0 :

$$y_2 + y_3 \geq 1.$$

Note that this inequality corresponds to a minimal cover with respect to the restricted set. Lifting down y_4 , y_5 and y_6 and then lifting up y_1 and y_7 in that order, according to the strategy described above, gives the following LCI :

$$3y_1 + y_2 + y_3 + y_4 + 2y_6 + 4y_7 \geq 4,$$

which is valid for P_S but violated by 0.4 by the current LP solution. This LCI also defines a facet of $\text{conv}(P_S)$.

4 Minimum Cardinality Inequalities

By a simple rounding argument, we can derive from (8) the following inequality :

$$\sum_{(i,j) \in (S, \bar{S})} y_{ij} \geq \left\lceil \frac{d_{(S, \bar{S})}}{u_{max}} \right\rceil \quad (11)$$

where $u_{max} = \max_{(i,j) \in (S, \bar{S})} u_{ij}$. Note that similar, although more complex, rounding arguments have been used to generate effective families of VI for the network loading problem (Magnanti, Mirchandani, and Vachani 1993, 1995, Barahona 1996, Bienstock and Günlük 1996, Bienstock *et al.* 1998). Note also that when the magnitude of the capacities vary widely, (11) provides a weak inequality as illustrated in Example 2.

Example 2 : Consider the following constraint :

$$15y_1 + 5y_2 + 3y_3 + 2y_4 + y_5 + 5y_6 \geq 30.$$

For this example $\left\lceil \frac{d_{(S, \bar{S})}}{u_{max}} \right\rceil = 2$, while, in fact, we have to open at least 5 arcs ($y_1 = y_2 = y_3 = y_4 = y_6 = 1$) to satisfy the constraint.

In order to obtain a stronger inequality of a form similar to (11), we define the *minimum cardinality inequality* (MCI) as follows :

$$\sum_{(i,j) \in (S, \bar{S})} y_{ij} \geq l_S,$$

where $l_S = \max \{h : \sum_{t=1, \dots, h} u_{ij(t)} < d_{(S, \bar{S})}\} + 1$ and $u_{ij(t)} \geq u_{ij(t+1)}$, $t = 1, \dots, m - 1$.

In this inequality, l_S is the least number of arcs in (S, \bar{S}) that must be open in every feasible solution. This inequality has been used by Martello and Toth (1997) to strengthen relaxation bounds for the 0-1 knapsack problem.

Note that, the separation problem for the MCI is trivial : for a given cutset (S, \bar{S}) , l_S is computed efficiently by a greedy algorithm. However, the MCI is not a strong inequality in general and it has to be strengthened using a lifting procedure.

4.1 Lifting

The lifting procedure for the MCI is similar to that for the LCI. We first apply procedure **OpenCloseArcs** to determine the sets C_1 and C_0 . We then derive a *restricted MCI* valid for the restricted set $P_{S_{(C_1, C_0)}}$, and we denote by $l_{S_{(C_1, C_0)}}$ the least number of arcs that must be open in the restricted set. To obtain a VI for P_S , it is then necessary to lift down the variables in C_1 and to lift up the variables in C_0 . Consequently, we seek lifting coefficients γ_{ij} such that the *lifted minimum cardinality inequality (LMCI)*,

$$\sum_{(i,j) \in C_1 \cup C_0} \gamma_{ij} y_{ij} + \sum_{(i,j) \in (S, \bar{S}) \setminus C_1 \cup C_0} y_{ij} \geq l_{S_{(C_1, C_0)}} + \sum_{(i,j) \in C_1} \gamma_{ij}, \quad (12)$$

is valid for P_S .

The lifting procedure is similar to that used to generate the LCI. More specifically, we introduce the notation $\tilde{C} = (S, \bar{S}) \setminus (C_1 \cup C_0)$, $\gamma_{ij} = 1, \forall (i, j) \in \tilde{C}$ and $\bar{d} = d_{(S, \bar{S})} - \sum_{(i,j) \in C_1 \setminus L} u_{ij}$. Then, to *lift down* variable $y_{rt} \in (S, \bar{S}) \setminus C_0$, we have to solve the following 0-1 knapsack problem :

$$\begin{aligned} Z_{opt} = \min \quad & \sum_{(i,j) \in \tilde{C} \cup L} \gamma_{ij} y_{ij}, \\ & \sum_{(i,j) \in \tilde{C} \cup L} u_{ij} y_{ij} \geq \bar{d} + u_{rt}, \\ & y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in C \cup L. \end{aligned}$$

If this problem is feasible, the lifting coefficient is given by $\gamma_{rt} = Z_{opt} - l_{S_{(C_1, C_0)}} - \sum_{(i,j) \in L \setminus C_0} \gamma_{ij}$; otherwise, we set $\gamma_{rt} = \sum_{(i,j) \in \tilde{C} \cup L} \gamma_{ij} - l_{S_{(C_1, C_0)}} - \sum_{(i,j) \in L \setminus C_0} \gamma_{ij} + 1$ (this

is equivalent to $y_{rt} = 1$). To *lift up* variable $y_{rt} \in C_0$, we have to solve the following 0-1 knapsack problem :

$$Z_{opt} = \min \sum_{(i,j) \in \tilde{C}_{UL}} \gamma_{ij} y_{ij},$$

$$\sum_{(i,j) \in \tilde{C}_{UL}} u_{ij} y_{ij} \geq \bar{d} - u_{rt},$$

$$y_{ij} \in \{0, 1\}, \forall (i, j) \in C \cup L.$$

If this problem is feasible, the lifting coefficient is given by $\gamma_{rt} = l_{S_{(C_1, C_0)}} + \sum_{(i,j) \in L \setminus C_0} \gamma_{ij} - Z_{opt}$; otherwise, we set $\gamma_{rt} = l_{S_{(C_1, C_0)}} + \sum_{(i,j) \in L \setminus C_0} \gamma_{ij} - \sum_{(i,j) \in \tilde{C}_{UL}} \gamma_{ij} - 1$ (this is equivalent to $y_{rt} = 0$).

In the same way as for the LCI, the 0-1 knapsack problems are solved using a dynamic programming algorithm. Also, lifting down the variables in C_1 must be accomplished before lifting up the variables in C_0 . When lifting down the variables in C_1 , those with fractional values must be lifted first, in non-decreasing order of their current value. Ties are broken by considering first the arcs in non-increasing order of their capacity. When lifting up the variables in C_0 , we do the opposite. Example 3 illustrates the impact of this lifting strategy.

Example 3 : As in Example 1, consider the following cutset inequality :

$$13y_1 + 7y_2 + 6y_3 + 4y_4 + 3y_5 + 11y_6 + 22y_7 \geq 22$$

and let $\bar{y} = (0, 0.5, 0.5, 0.6, 0.7, 1, 0)$ be the current LP solution. In every feasible solution, we have to open at least one arc (y_7) to satisfy the demand. The MCI is thus

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 \geq 1.$$

This inequality is not violated by the current LP solution.

To generate the LMCI, let $\epsilon_1 = 0.6$ and $\epsilon_0 = 0.1$, as in Example 1. With Procedure **OpenCloseArcs**, we find $C_1 = \{5, 6\}$ and $C_0 = \{1, 7\}$. The cutset inequality restricted to $P_{S_{(C_1, C_0)}}$ is

$$7y_2 + 6y_3 + 4y_4 \geq 8.$$

The least number of arcs to open in $P_{S_{(C_1, C_0)}}$ is $l_{S_{(C_1, C_0)}} = 2$, so the restricted MCI is

$$y_2 + y_3 + y_4 \geq 2.$$

To obtain an inequality that is valid for P_S , we lift down y_5 and y_6 and then we lift up y_1 and y_7 , according to the lifting order strategy described above. We then obtain the following LMCI :

$$3y_1 + y_2 + y_3 + y_4 + 2y_6 + 4y_7 \geq 4.$$

Note that, we have generated the same inequality as in Example 1. It is thus natural to look more deeply into the relationship between LCI and LMCI. In particular, given that we apply the same lifting order strategy with the same sets C_1 and C_0 , one might ask if there are cases where one family of VI provides a cut, while the other does not. The following example illustrates these situations.

Example 4 : Consider the following cutset inequality :

$$3y_1 + 3y_2 + 3y_3 + 4y_4 + 5y_5 \geq 10.$$

Let $\bar{y}^1 = (0.1, 0.9, 0.9, 0.1, 0.9)$ and $\bar{y}^2 = (0.1, 0.99, 0.99, 0.1, 0.85)$ be two LP solutions. It is easy to verify that both solutions satisfy this cutset inequality. We will assume that $C_1 = C_0 = \emptyset$, a result that would be obtained if Procedure **OpenCloseArcs** is performed with $\epsilon_1 = 1$ and $\epsilon_0 = 0$.

When we apply the CI separation heuristic for any of the two LP solutions, we derive the minimal cover $C = \{4, 5\}$, since we include successively arcs 4, 1, and 5, thus obtaining a cover, and then we remove arc 1 to ensure the cover is minimal. The corresponding CI is :

$$y_4 + y_5 \geq 1.$$

When we apply the lifting procedure, we lift down successively y_1 , y_2 , and y_3 , but the inequality is not improved, since we obtain $\gamma_i = 0, i = 1, 2, 3$. We note that this inequality is not violated by \bar{y}^1 , but is violated by \bar{y}^2 .

The least number of arcs to open in P_S is $l_S = 3$. Thus, the MCI is :

$$y_1 + y_2 + y_3 + y_4 + y_5 \geq 3.$$

This inequality is violated by \bar{y}^1 , but not by \bar{y}^2 .

As shown by this example, with the same lifting strategy, it is possible to generate two different inequalities, one being violated by the current LP solution, while the other is not. Motivated by this remark, we have incorporated the LCI and the LMCI into our cutting-plane procedure.

5 Network Cutset Inequalities

Any feasible solution of the MCND is defined by the continuous flow variables, x_{ij}^k , and the 0-1 design variables, y_{ij} . Yet, the VI presented in the previous sections involve only the design variables. In order to derive VI that involve both types of variables, let $x_{ij} = \sum_{k \in K} x_{ij}^k$. We characterize any feasible solution of the MCND by Proposition 1.

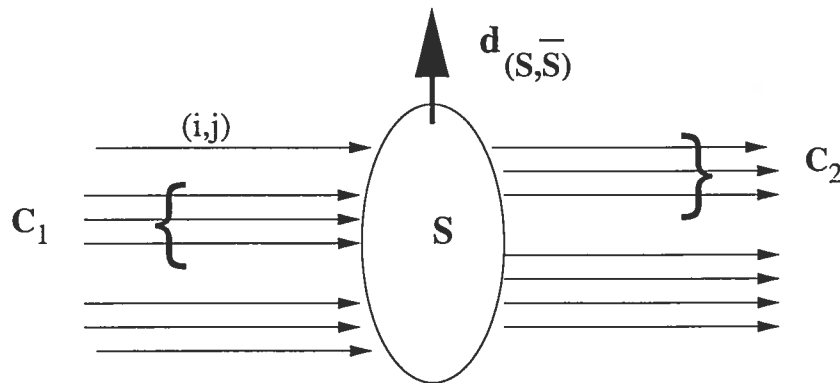


FIG. 2 – Network cutset structure

Proposition 1 : Let $(i, j) \in (\bar{S}, S)$ and $C_2 \subseteq (S, \bar{S})$. Then,

$$x_{ij} \leq \left(\sum_{(r,t) \in C_2} u_{rt} + d_{(S, \bar{S})} \right) y_{ij} + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}, \quad (13)$$

is a VI for the MCND.

Proof : Let (x, y) be a feasible solution of the MCND. Then,

- if $x_{ij} = 0$, inequality (13) is satisfied ;

- if $x_{ij} > 0$,

$$\begin{aligned}
x_{ij} &\leq d_{(S,\bar{S})} + \sum_{(r,t) \in (S,\bar{S})} x_{rt}, && \text{(flow conservation)} \\
&\leq d_{(S,\bar{S})} + \sum_{(r,t) \in C_2} x_{rt} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}, \\
&\leq d_{(S,\bar{S})} + \sum_{(r,t) \in C_2} u_{rt} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}, && \text{(capacity constraint)} \\
&\leq (\sum_{(r,t) \in C_2} u_{rt} + d_{(S,\bar{S})}) y_{ij} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}. && \text{(since } y_{ij} = 1)
\end{aligned}$$

Then, (13) is a VI for the MCND for any cutset (S, \bar{S}) . In fact, inequality (13) is redundant for the original formulation of the MCND but might be violated by a fractional LP solution, as shown in the following example.

Example 5 : Consider a set $S \subset N$ such that $(\bar{S}, S) = \{1, 2\}$, $(S, \bar{S}) = \{3, 4\}$, and $d(S, \bar{S}) = 2$. Let the capacities of the arcs 1, 2, 3, and 4 be given by 8, 4, 4, and 1, respectively. Also, let $\bar{x} = (1, 2, 0, 1)$ and $\bar{y} = (0.125, 0.5, 0, 1)$ be the current LP solution. Given that $C_2 = \{4\}$, we obtain the following inequalities :

$$x_1 - x_3 - 3y_1 \leq 0,$$

derived for arc 1 and violated by 0.625 ;

$$x_2 - x_3 - 3y_2 \leq 0,$$

derived for arc 2 and violated by 0.5.

Based on Proposition 1, we introduce the *network cutset inequality (NCI)*, defined by Proposition 2.

Proposition 2 : Let $(i, j) \in (\bar{S}, S)$, $C_1 \subseteq (\bar{S}, S) \setminus \{(i, j)\}$ and $C_2 \subseteq (S, \bar{S})$. Then,

$$\sum_{(r,t) \in C_1} x_{rt} + x_{ij} \leq \left(\sum_{(r,t) \in C_2} u_{rt} + d_{(S,\bar{S})} \right) y_{ij} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt} + (1 - y_{ij}) \sum_{(r,t) \in C_1} u_{rt} \quad (14)$$

is a VI for the MCND.

Proof : Let (x, y) be a feasible solution of the MCND. Then,

- if $y_{ij} = 0$, then $x_{ij} = 0$ and (14) is valid since $\sum_{(r,t) \in C_1} x_{rt} \leq \sum_{(r,t) \in C_1} u_{rt}$;

- if $y_{ij} = 1$,

$$\begin{aligned}
\sum_{(r,t) \in C_1} x_{rt} + x_{ij} &\leq \sum_{(r,t) \in C_2} x_{rt} + d_{(S,\bar{S})} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}, \\
&\quad \text{(flow conservation)} \\
&\leq \sum_{(r,t) \in C_2} u_{rt} + d_{(S,\bar{S})} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}, \\
&\quad \text{(capacity constraint)} \\
&\leq (\sum_{(r,t) \in C_2} u_{rt} + d_{(S,\bar{S})}) y_{ij} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}. \\
&\quad \text{(since } y_{ij} = 1)
\end{aligned}$$

Note that, the NCI (14) is stronger than inequality (13) when $\sum_{(r,t) \in C_1} x_{rt} > (1 - y_{ij}) \sum_{(r,t) \in C_1} u_{rt}$. The next example illustrates this situation.

Example 6 : Consider a set $S \subset N$ such that $(\bar{S}, S) = \{1, 2\}$, $(S, \bar{S}) = \{3, 4\}$, and $d(S, \bar{S}) = 1$. Let the capacities of the arcs 1, 2, 3, and 4 be given by 2, 3, 3, and 4, respectively. Also, let $\bar{x} = (2, 1, 2, 0)$ and $\bar{y} = (1, 1/3, 2/3, 0)$ be the current LP solution. Given that $C_2 = \{3\}$, generating the NCI (14) for arc 2, with $C_1 = \{1\}$, gives the inequality

$$x_1 + x_2 - x_4 - 2y_2 \leq 2,$$

which is violated by $1/3$. But, generating the inequality (13) for arc 2 gives

$$x_2 - x_4 - 4y_2 \leq 0,$$

which is not violated.

The NCI considers each arc $(i, j) \in (\bar{S}, S)$ separately. We can improve it by considering multiple arcs in a single inequality, called the *generalized network cutset inequality (GNCI)*, introduced in the following proposition.

Proposition 3 : Let $C \subseteq (\bar{S}, S)$, $C_1 \subseteq (\bar{S}, S)$ ($C \cap C_1 = \emptyset$), $C_2 \subseteq (S, \bar{S})$ and $Y_C = \max_{(r,t) \in C} y_{rt}$. Then,

$$\sum_{(r,t) \in C_1} x_{rt} + \sum_{(r,t) \in C} x_{rt} \leq \left(\sum_{(r,t) \in C_2} u_{rt} + d_{(S,\bar{S})} \right) Y_C + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt} + (1 - Y_C) \sum_{(r,t) \in C_1} u_{rt}, \quad (15)$$

is a VI for the MCND.

Proof : Let (x, y) be a feasible solution of the MCND. Then,

- if $y_{rt} = 0$, $\forall (r, t) \in C$, then $Y_C = 0$ and the GNCI is valid since

$$\sum_{(r,t) \in C_1} x_{rt} \leq \sum_{(r,t) \in C_1} u_{rt};$$

- if there exists $(i, j) \in C$ such that $y_{ij} = 1$, then $Y_C = 1$ and

$$\begin{aligned} \sum_{(r,t) \in C_1} x_{rt} + \sum_{(r,t) \in C} x_{rt} &\leq \sum_{(r,t) \in C_2} x_{rt} + d_{(S, \bar{S})} + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}, \\ &\quad \text{(flow conservation)} \\ &\leq \sum_{(r,t) \in C_2} u_{rt} + d_{(S, \bar{S})} + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}, \\ &\quad \text{(capacity constraint)} \\ &\leq (\sum_{(r,t) \in C_2} u_{rt} + d_{(S, \bar{S})}) Y_C + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}. \\ &\quad \text{(since } Y_C = 1) \end{aligned}$$

5.1 Separation

Given a set $S \subset N$ and an arc $(i, j) \in (\bar{S}, S)$, the separation problem for the NCI consists in finding two sets $C_1 \subseteq (\bar{S}, S)$ and $C_2 \subseteq (S, \bar{S})$ such that the violation with respect to the current LP solution, (\bar{x}, \bar{y}) , is maximal :

$$\begin{aligned} \max \sum_{(r,t) \in (\bar{S}, S) \setminus \{(i,j)\}} (\bar{x}_{rt} - (1 - \bar{y}_{ij})u_{rt}) Z_{rt}^1 - \sum_{(r,t) \in (S, \bar{S})} (u_{rt} \bar{y}_{ij} - \bar{x}_{rt}) Z_{rt}^2, \\ Z_{rt}^1 = 0 \text{ or } 1, \quad \forall (r, t) \in (\bar{S}, S), \\ Z_{rt}^2 = 0 \text{ or } 1, \quad \forall (r, t) \in (S, \bar{S}), \end{aligned}$$

where Z^1 and Z^2 are the characteristic vectors of C_1 and C_2 , respectively. This problem is easy to solve. Its optimal solution (Z^{1*}, Z^{2*}) is :

$$\begin{aligned} Z_{rt}^{1*} &= \begin{cases} 1, & \text{if } \bar{x}_{rt} > (1 - \bar{y}_{ij})u_{rt}, \\ 0, & \text{otherwise,} \end{cases} \\ Z_{rt}^{2*} &= \begin{cases} 1, & \text{if } u_{rt} \bar{y}_{ij} < \bar{x}_{rt}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Note that, with this choice of C_1 , the NCI (14) always dominates inequality (13).

The separation problem for the GNCI is more difficult. It consists in finding three sets, $C \subseteq (\bar{S}, S)$, $C_1 \subseteq (\bar{S}, S)$ ($C \cap C_1 = \emptyset$), and $C_2 \subseteq (S, \bar{S})$, such that the violation with respect to the current LP solution, (\bar{x}, \bar{y}) , is maximal. Note that, once we have

determined C , we can efficiently construct the sets C_1 and C_2 as in the generation of the NCI. To identify the set C , we consider the following MIP problem :

$$\begin{aligned} \max \quad & \sum_{(i,j) \in (\bar{S}, S)} \bar{x}_{ij} Z_{ij} - d_{(S, \bar{S})} Y, \\ & Y \geq \bar{y}_{ij} Z_{ij}, \quad \forall (i, j) \in (\bar{S}, S), \\ & Z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in (\bar{S}, S), \\ & 0 \leq Y \leq 1, \end{aligned}$$

where Z is the characteristic vector of C . To solve this problem, we have devised a heuristic procedure, **Gen-GNCI**, based on the fact that, if Y is fixed, the problem becomes easy to solve. Next, we summarize the outline of this procedure, which accepts as input the values of two user-supplied parameters, \bar{Y} and δ .

Procedure Gen-GNCI :

- 1- Fix Y_C at some small value, $Y_C \leftarrow \bar{Y}, 0 < \bar{Y} < 1$.
- 2- Let $C \leftarrow \{(i, j) \in (\bar{S}, S) \mid \bar{y}_{ij} \leq Y_C\}$ and readjust Y_C by setting $Y_C \leftarrow \max_{(i,j) \in C} \bar{y}_{ij}$.
- 3- Construct C_1 and C_2 as follows :
 - $C_1 \leftarrow \{(i, j) \in (\bar{S}, S) \mid u_{ij}(1 - Y_C) < \bar{x}_{ij}\}$,
 - $C_2 \leftarrow \{(i, j) \in (S, \bar{S}) \mid u_{ij}Y_C < \bar{x}_{ij}\}$.
- 4- If the GNCI is violated, then add it to the formulation, along with the inequalities
 - $Y_C \geq y_{ij}, \quad \forall (i, j) \in C$,
 - $Y_C \leq \sum_{(i,j) \in C} y_{ij}$,
 - $0 \leq Y_C \leq 1$.
- 5- Otherwise, give Y_C a smaller value, $Y_C \leftarrow Y_C - \delta$;
 - if $Y_C > 0$, return to 2,
 - else STOP.

Note that, if a violated GNCI is generated at step 4, one has to add the violated inequality, but also a new continuous variable Y_C and a set of linear inequalities equivalent to $Y_C = \max_{(i,j) \in C} y_{ij}$.

We present in Appendix 1 variants of the NCI and the GNCI that give, from a theoretical point of view, stronger inequalities than the originals. The investigation and study of the usefulness of these variants still under developments.

6 Algorithm Description

Recall that, all VI presented and studied in the previous sections are based on the assumption that a cutset (S, \bar{S}) is given. In the following subsections, we address the issues of how to determine “good” cutsets and how to use these cutsets within a cutting-plane procedure. In Section 6.1, we describe heuristics to generate “good” cutsets, while in Section 6.2, we present the outline of the cutting-plane procedure.

6.1 Cutset Generation Algorithm

Given that each cutset might allow to generate several types of cuts (LCI, LMCI, NCI or GNCI), the performance of the cutting-plane algorithm depends on how quickly it can identify “good” cutsets. We propose to identify and maintain simultaneously multiple cutsets, called a *family*. Such a family of cutsets is obtained by partitioning the set of nodes N into L subsets $S_l, l = 1, \dots, L,$, such that $S_l \cap S_k = \emptyset$, for all $l \neq k$, and $\cup_{l=1, \dots, L} S_l = N$. Then, each cutset (S_l, \bar{S}_l) induced by subset $S_l, l = 1, \dots, L$, determines a family of cutsets $(S_l, \bar{S}_l)_{l=1, \dots, L}$ available for the generation of violated VI.

An initial family of cutsets, for which we verify the violation of VI, is obtained by partitioning N into $|N|$ subsets, each containing a single node (i.e., $S_l = \{i\}, \forall i \in N, l = 1, \dots, |N|$). To partition N into subsets of cardinalities greater than 1, we introduce a procedure called **CreateMultiSet(M)**, which constructs iteratively multiple subsets S_l , each with a maximum cardinality of M . Since all types of VI (LCI, LMCI, NCI and GNCI) have a better chance of being violated when the arcs in (S_l, \bar{S}_l) show small fractional values \bar{y}_{ij} , the procedure attempts to construct the sets S_l with the objective of minimizing $\sum_{(i,j) \in (S_l, \bar{S}_l)} \bar{y}_{ij}$. At any step of the procedure, let S_l be a subset of N of cardinality smaller than M . Initially, the family contains one subset, S_1 ,

having a single element (arbitrarily chosen). We denote by *free node* a node that is not included in any subset and by \bar{N} the set of all free nodes. To achieve our objective, we identify the free node n such that

$$n = \operatorname{argmax}_{j \in \bar{N}} \{\bar{y}_{ij}, \forall i \in S_l\}.$$

If n exists, then we add it to S_l and move to the next step : continue with the construction of S_l , if $|S_l| < M$, or otherwise, proceed to the construction of S_{l+1} (by selecting arbitrarily some free node and then repeating the process). If, however, no free node is connected by an arc to at least one node in S_l , we choose n arbitrarily among the free nodes. The procedure stops when there are no more free nodes. The outline of the procedure is summarized next.

Procedure CreateMultiSet(M) :

- 1- Set $\bar{N} \leftarrow N, L \leftarrow 1$.
- 2- If $\bar{N} = \emptyset$, STOP.
Otherwise, select (arbitrarily) a node $m \in \bar{N}$.
- 3- Add m to S_L and update $\bar{N} \leftarrow \bar{N} \setminus \{m\}$.
- 4- If $|S_L| \geq M$, $L \leftarrow L + 1$ and go to 2.
- 5- Find $n \in \bar{N}$ such that $n = \operatorname{argmax}_{j \in \bar{N}} \{\bar{y}_{ij}, \forall i \in S_L\}$.
- 6- If n exists, $m \leftarrow n$ and go to 3.
Otherwise, go to 2.

Note that, the procedure attempts to include first in S_l a free node that is connected by an arc to at least one node in S_l to avoid generating VI that are aggregations of previously generated VI. Figure 3 illustrates such a situation, where we assume that all possible violated VI using cutsets induced by subsets of cardinality 1 have already been generated. Currently, the procedure is constructing subsets of cardinality 2. If we chose to include the free node 2 into the set $S_l = \{1\}$ to create the new set $S_l = \{1, 2\}$, we would identify a new cutset, but the resulting cutset inequality would just be the aggregation of the previously generated cutset inequalities induced by $\{1\}$ and $\{2\}$. Thus, this new cutset would not identify any violated VI.

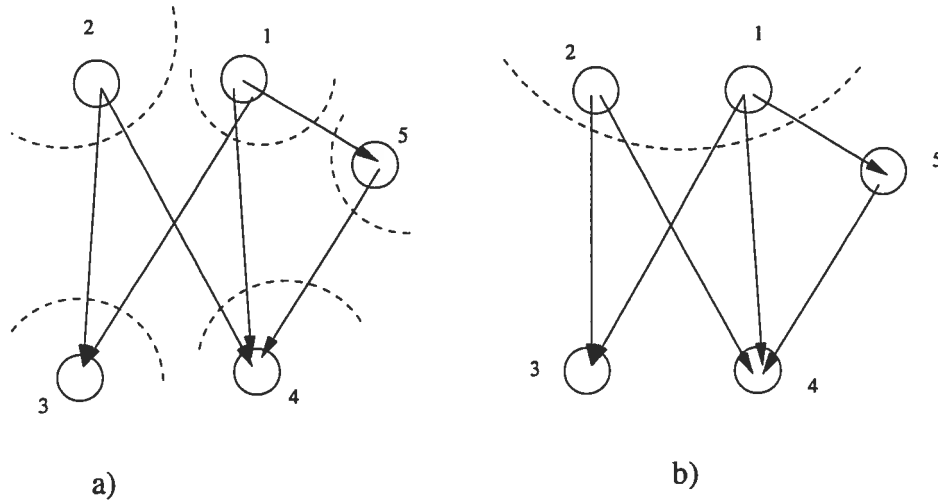


FIG. 3 – Add-drop example

Once a family of cutsets is determined by Procedure **CreateMultiSet(M)**, all possible cuts are generated for this family. Then, new families of cutsets will be obtained by performing exchanges of nodes (similar to add-drop moves) among subsets of the current family. The basic idea behind these exchanges is to obtain a new subset $S_{l'}$ from a subset S_l by moving a node n from some set S_k , $S_k \subset \bar{S}_l$, to S_l . These exchanges are performed by Procedure **MultiAddDrop**, which considers at each step a set S_l and aims to identify and move to S_l the node n such that

$$n = \operatorname{argmin}_{m \in S_k, S_k \subset \bar{S}_l} \frac{\sum_{(i,j) \in (S_l \cup \{m\}, \bar{S}_l \cup \{m\})} u_{ij} \bar{y}_{ij}}{d_{(S_l \cup \{m\}, \bar{S}_l \cup \{m\})}}.$$

Note that, n is chosen among the set of nodes connected by an arc to at least one node in S_l . This strategy attempts to avoid generating VI that are aggregations of previously generated VI. Once n is identified, we move it from S_k to S_l . Then, the procedure repeats the process by considering subset S_k at the next iteration. Starting with S_1 , the procedure stops whenever it finds a couple of subsets (S_l, S_k) involved in an exchange, such that $k < l$. This strategy attempts to modify as less as possible the cardinality of the subsets. It is based on the observation that, to obtain a new family of cutsets, it is not necessary to perform exchanges for each subset $S_l, l = 1, \dots, L$. In fact, only a small number of exchanges allows to significantly modify the initial family. Note that, in some cases, the procedure identifies a cycle on which the nodes

are moved around; when this happens, just the first and last subsets in this cycle have modified cardinalities. This is illustrated in Figure 4, where the procedure stops after successively moving nodes from S_2 to S_1 , from S_3 to S_2 , from S_4 to S_3 , and from S_2 to S_4 . By doing so, subsets S_3 and S_4 have the same cardinality as before the exchanges, while S_1 and S_2 have, respectively, one additional element and one element less. The outline of this procedure is presented next.

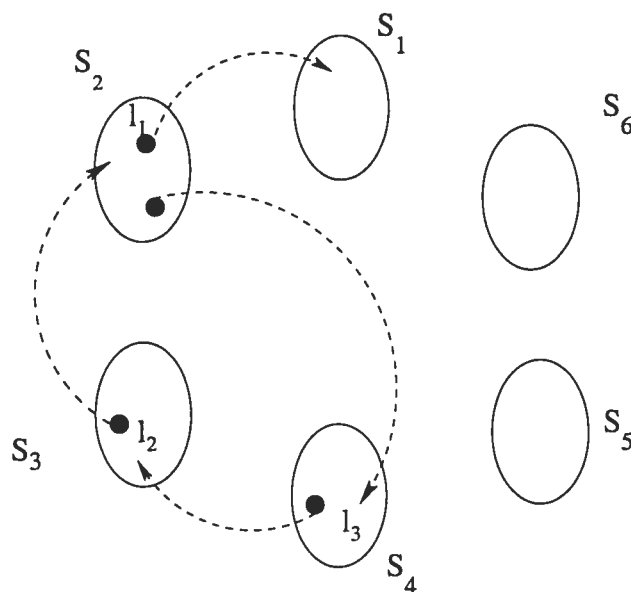


FIG. 4 – Add-drop example

Procedure MultiAddDrop :

- 1- Start with a family of cutsets $(S_l, \bar{S}_l)_{l=1, \dots, L}$, and let $l = 1$.
- 2- Find $S_k \subset \bar{S}_l$, such that $n \in S_k$ and

$$n = \operatorname{argmin}_{m \in S_k, S_k \subset \bar{S}_l} \frac{\sum_{(i,j) \in (S_l \cup \{m\}, \overline{S_l \cup \{m\}})} u_{ij} \bar{y}_{ij}}{d_{(S_l \cup \{m\}, \overline{S_l \cup \{m\}})}}.$$

- 3- Move n from S_k to S_l .
- 4- If $k < l$, STOP.
Otherwise, $l \leftarrow k$ and return to 2.

We show next how to use these two procedures within the cutting-plane algorithm.

6.2 Cutting-Plane Algorithm

The cutting-plane algorithm is divided into an initialization step and two improvement phases. The initialization step solves the LP relaxation of the problem. In the first phase, the cutting-plane procedure performs the generation of cuts based on the initial family of cutsets obtained by partitioning N into $|N|$ subsets, each containing a single element (i.e., $S_l = \{i\}, \forall i \in N, l = 1, \dots, |N|$). The generation of cuts involves the addition of the corresponding VI, if any, to the formulation, and the resolution of the resulting LP relaxation. The first phase of the algorithm iterates on the initial family of cutsets until no further significant improvement is observed.

To obtain additional improvements, the second phase of the procedure is launched. In this phase, new families of cutsets are identified by performing procedure **CreateMultiSet(M)**, where M varies from 2 to M_{max} . For each family of cutsets, the algorithm performs the generation of cuts and identifies families of cutsets by applying procedure **MultiAddDrop** for a given number of iterations, I_{max} . If the solution obtained at the end of this second phase improves upon that of the first phase, then the first phase is launched all over again. Thus, the proposed approach iterates over the first and second phases until no further improvement is obtained. In addition, to limit the computational effort, the second phase is invoked no more than T_{max} times.

The cutting-plane algorithm is summarized next. Here, Z and (\bar{x}, \bar{y}) represent the current LP optimal value and optimal solution, respectively, while Z_{last} is the LP optimal value at the last iteration during the first phase, and during the second phase, it is the LP optimal value at the end of the first phase. It is used to verify the stopping condition $Z - Z_{last} \leq \epsilon$, used in any of the two phases (ϵ is

1. *Initialization step* :

- $T \leftarrow 0, Z_{last} \leftarrow 0$.
- Solve the LP relaxation; let Z be the optimal value and (\bar{x}, \bar{y}) be the corresponding optimal solution.
- If (\bar{x}, \bar{y}) is feasible, go to 4.

2. *First phase :*

- Initialize the family of cutsets by partitioning N into $|N|$ subsets (i.e., $S_l = \{i\}, \forall i \in N, l = 1, \dots, |N| = L$).
- While $Z - Z_{last} > \epsilon$
 - Add cuts; if no cuts were found, go to 3.
 - $Z_{last} \leftarrow Z$.
 - Solve the LP relaxation; let Z be the optimal value and (\bar{x}, \bar{y}) be the corresponding optimal solution.
 - If (\bar{x}, \bar{y}) is feasible, go to 4.

3. *Second phase :*

- If $T \geq T_{max}$, go to 4.
- $T \leftarrow T + 1$ and $Z_{last} \leftarrow Z$.
- For $M = 2$ to M_{max}
 - Identify a family of cutsets by performing Procedure **CreateMultiSet**(M).
 - Add cuts.
 - If at least one cut has been added
 - Solve the LP relaxation; let Z be the optimal value and (\bar{x}, \bar{y}) be the corresponding optimal solution.
 - If (\bar{x}, \bar{y}) is feasible, go to 4.
- For $I = 1$ to I_{max}
 - Identify a family of cutsets by performing Procedure **MultiAddDrop**.
 - Add cuts.
 - If at least one cut has been added
 - Solve the LP relaxation; let Z be the optimal value and (\bar{x}, \bar{y}) be the corresponding optimal solution.
 - If (\bar{x}, \bar{y}) is feasible, go to 4.
- If $Z - Z_{last} > \epsilon$, go to 2.

4. Return Z and (\bar{x}, \bar{y}) .

The main objective of this study being to improve the quality of the lower bounds using a reasonable computational effort, it is interesting to evaluate the improvement that one might obtain by performing the overall algorithm compared to the first phase alone, and for how much additional computational effort. To this end, we identify two different approaches, the *simple cutting-plane algorithm (SCP)*, which is the given algorithm stopped at the end of the first phase, and the *advanced cutting-plane algorithm (ACP)*, which is the overall algorithm.

Adding the strong inequalities (SI) to the formulation of the MCND improves the quality of the lower bounds. Yet, for large-scale instances, solving the LP relaxation including all these inequalities is time consuming. Moreover, not all such inequalities are necessary. Hence, during the initialization step, the cutting-plane algorithm solves the weak LP relaxation, and subsequently, it adds only SI that are violated by the current LP solution. The outline of the *Add cuts* part in the algorithm becomes :

- Add violated SI.
- For each cutset $(S_l, \bar{S}_l), l = 1, \dots, L$:
 - Add violated LCI.
 - Add violated LMCI.
 - Add violated NCI or GNCI.

Using GNCI instead of NCI can increase significantly the size of the problem since for each violated GNCI, we add one continuous variable and the inequalities that define it. We might thus rightfully ask what is the improvement one can obtain by using GNCI instead of NCI and for how much additional computational effort. To answer these issues, we treat these two families, NCI and GNCI, separately in the analysis of the computational results.

To sum up, in our experiments, we identify and compare the performance of four different approaches :

- SCP-NCI : the simple cutting-plane algorithm using NCI ;
- ACP-NCI : the advanced cutting-plane algorithm using NCI ;

- SCP-GNCI : the simple cutting-plane algorithm using GNCI;
- ACP-GNCI : the advanced cutting-plane algorithm using GNCI.

7 Computational Results

To test the performance of the cutting-plane approaches, we report and analyze computational results on a large set of test problems. The implementations are performed in C++, using the option *dualopt* of CPLEX (version 7.1, 2002) to solve to optimality the LP relaxation of the MCND. All experiments are performed on a Sun Enterprise 10000 with 64 Gigabytes of RAM and operating under Solaris 2.7.

The values of the parameters in our implementation are $T_{max} = 5$, $I_{max} = 20$, $M_{max} = \lceil \frac{N}{3} \rceil$ and $\epsilon = 0.1$. In addition, in Procedure **OpenCloseArcs**, we set $\epsilon_0 = 0.01$ and $\epsilon_1 = 0.7$. These values have been selected following a preliminary computational study.

Our test sets for the experimental study are described in Crainic, Frangioni, and Gendron (2001) and Gendron and Crainic (1994, 1996). The same test sets are used in several papers on the MCND such as Crainic, Gendreau, and Farvolden (1998), Ghamlouche, Crainic, and Gendreau (2001, 2002), and Crainic, Gendron, and Hernu (2002). These problem tests consist of two series of general transshipment networks with one commodity per origin-destination and no parallel arcs. Positive transportation and fixed costs are associated with each arc. Note that, the transportation costs on each arc are the same for all commodities.

The first set of instances, denoted **C**, consists of 43 problem instances characterized by the number of nodes, the number of arcs, and the number of commodities denoted $|N|$, $|A|$, and $|K|$ respectively. Two additional letters are used to characterize the fixed cost level, “F” for high and “V” for low, relatively to the transportation cost, and the capacity level, “T” for tight and “L” for loose, compared to the total demand.

The second set of instances, denoted **R**, consists of nine sets of nine problem instances each. Each set is characterized by the same number of nodes, arcs, and com-

modities for the nine instances but with various levels of fixed cost and capacity ratios. Thus, $F = 0.01$ (F01), $F = 0.05$ (F05), and $F = 0.10$ (F10), are used to qualify the importance of the fixed costs compared to the transportation costs where the fixed cost ratio is computed as $F = |K| \sum_{(i,j) \in A} f_{ij} / \sum_{k \in K} d^k \sum_{(i,j) \in A} c_{ij}^k$. Similarly, $C = 1$ (C1), $C = 2$ (C2), and $C = 8$ (C8), are used to qualify the tightness of the total capacity compared to the total demand, where the capacity ratio is computed as $C = |A| \sum_{k \in K} d^k / \sum_{(i,j) \in A} u_{ij}$.

Our results, bounds, and computational times are compared to those obtained by computing the strong LP bounds using CPLEX (version 7.1, 2002) to solve the strong LP relaxation. The gap relative to the optimum and the CPU time needed to achieve the lower bounds are our primary measures of performance. In order to assess the solution quality relative to the optimal solution, we tried to solve all instances using the branch-and-cut (B&C) of CPLEX (version 7.1, 2002) with default settings for all parameters (Chouman, Gendron, and Crainic 2001). A limit of 10 hours of computation was imposed for each instance. The results show that CPLEX is quite efficient at solving small size instances within the time allowed. However, when the problem size is increasing (especially the number of commodities), CPLEX is unable to find the optimal solution (not even a feasible solution for problems with 400 commodities) within the available computational time. For these instances, the best feasible solution found in Ghamlouché, Crainic, and Gendreau (2002) is used. It should be mentioned that these results also show that CPLEX is unable to detect the knapsack structure hidden in our problems class, thus it was unable to generate any CI on a large set of instances.

To avoid overloading the paper, we report detailed results for only 13 selected problems in class **C**, with various dimensions (detailed results for all instances are presented in Appendix 2). These problems are difficult because they have high fixed costs (F) and tight capacities (T), except for P12. We decided to include P12, which has high fixed costs (F), but loose capacities (L), since 10 hours of CPU time was not sufficient to solve its LP relaxation using CPLEX. Table 1 shows results for the selected problems. For each problem, we report the description (DESCR) by identifying the

number of nodes, arcs, and commodities respectively, the optimal (indicated by *) or the best feasible solution (SOL), and the gap of the strong LP bound (GAP-LP) computed with respect to the best known solution SOL. The last four columns indicate the gap of the lower bound and the number of VI for each family generated in SCP-GNCI and ACP-GNCI, respectively. The CPU time elapsed in seconds is given under each solution. The letter (t) indicates that the solution is feasible but its optimality could not be verified within 10 hours with the B&C of CPLEX.

The results show that the bounds derived from SCP-GNCI and ACP-GNCI improve upon the strong LP bound. The average of gap improvement for problems P1 to P11 is 1% while it can reach up to 6% for some instances (P7 for example). It should be pointed out that, even if the improvement is small it is still quite important : we have noted that a large number of nodes must be explored in the enumeration tree to get such improvement on some instances using the B&C of CPLEX. The improvement in the CPU time needed to find the lower bound is the most significant when compared to that needed by CPLEX to compute the strong LP bound, especially when the problem dimension is increasing. For example, for P12, CPLEX does not manage to solve the strong LP relaxation in 10 hours while it requires about 80 minutes to compute the lower bound using SCP-GNCI and 90 minutes using ACP-GNCI.

Table 2 summarizes the gap and the improvement distribution for all C problems according to the number of commodities and the different levels of fixed cost and capacity ratios. The gap and the improvement over the gap of the strong LP bound are given for the four approaches : SCP-NCI, ACP-NCI, SCP-GNCI, and ACP-GNCI. These results show that the bounds are improved over the strong LP bounds for the four approaches. Note also that the advanced cutting-plane algorithms (ACP-NCI and ACP-GNCI) outperform the simple cutting-plane versions especially for small size problems. Moreover, the bounds derived from the two advanced cutting-plane procedures (ACP-NCI and ACP-GNCI) are competitive. Thus, using GNCI instead of NCI is better for some instances, but not always. This is not surprising since we have used a heuristic to solve the separation problem for the GNCI while the separation problem for the NCI is solved exactly. Finally, it is clear that for all approaches the gap improvement

	DESCR N , A , K	SOL	GAP-LP	SCP-GNCI		ACP-GNCI	
				GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)
P1	25, 100, 10 FT	49899* (40.6)	9.76 (0.3)	8.21 (0.7)	(60, 12, 12, 51)	7.6 (5.4)	(60, 23, 42, 51)
P2	25, 100, 30 FT	85530* (534.2)	3.63 (1.6)	3.56 (1.2)	(105, 20, 20, 33)	3.53 (3.3)	(105, 20, 20, 33)
P3	20, 230, 40 FT	643036* (671.7)	1.48 (0.6)	1.41 (2.7)	(92, 20, 17, 17)	1.35 (15.2)	(92, 21, 18, 31)
P4	20, 300, 40 FT	604198* (59.8)	1.21 (0.6)	1.14 (3.2)	(95, 22, 25, 5)	1.14 (11.1)	(95, 22, 23, 5)
P5	20, 230, 200 FT	137271 (t)	4.17 (2431.1)	4.16 (306.8)	(1286, 22, 18, 56)	4.16 (348.4)	(1286, 22, 18, 56)
P6	20, 300, 200 FT	108252 (t)	4.26 (2605.1)	4.12 (282.2)	(841, 14, 13, 43)	4.12 (343.6)	(841, 14, 13, 41)
P7	100, 400, 10 FT	66364 (t)	23.02 (13.0)	18.95 (20.7)	(187, 20, 18, 325)	17.05 (343.4)	(202, 35, 28, 1236)
P8	100, 400, 30 FT	141278 (t)	15.44 (110.2)	13.68 (40.6)	(273, 45, 49, 230)	13.33 (279.6)	(297, 54, 55, 742)
P9	30, 520, 100 FT	98357 (t)	4.41 (863.1)	4.24 (157.9)	(672, 32, 42, 42)	4.24 (356.2)	(672, 33, 43, 44)
P10	30, 700, 100 FT	55082 (t)	2.58 (356.0)	2.48 (114.6)	(517, 33, 32, 47)	2.48 (238.6)	(517, 33, 32, 48)
P11	30, 520, 400 FT	163352 (59820.1)	8.32 (11706.8)	8.29 (2794.4)	(1618, 10, 6, 39)	8.29 (3374.2)	(1618, 10, 6, 42)
P12	30, 700, 400 FL	144615 (57574.8)	X (t)	9.60 (4747.0)	(2042, 15, 8, 39)	9.60 (5314.1)	(2042, 15, 8, 39)
P13	30, 700, 400 FT	138777 (48693.7)	8.07 (34057.0)	8.07 (4113.1)	(1539, 7, 4, 28)	8.07 (4534.6)	(1539, 7, 4, 28)

TAB. 1 – Detailed results for 13 selected instances

decreases as the number of commodities increases. Also, for large-scale problems, with 200 and 400 commodities, the gap improvement is almost the same for all approaches. Table 3 displays the CPU reduction (CPU) and the gap improvement (IMP) compared to CPLEX solving the strong LP relaxation, for large size problems. The average reduction in CPU time ranges from 65% up to 87% for the different approaches. Thus, the proposed approaches perform well and find better bounds in much shorter time (reduction of about 80% on average) when compared to the strong LP bounds computed by CPLEX.

Our remarks and conclusions are confirmed by the results obtained on the **R** problems. Table 4 displays the aggregated results (9 problems in each category). Problems are grouped according to their difficulty from the easiest (F01,C1) to the most difficult (F10,C8). The results indicate that the bounds derived from the different cutting-plane approaches are better than the strong LP bounds. Moreover, the improvement increases with increasing problem difficulty. The gap improvement is more significant for advanced cutting-plane versions than for simple cutting-plane versions. Note that, the average improvement is 0.25%, 0.30%, 0.27% and 0.30% for SCP-NCI, ACP-NCI,

K		SCP-NCI			ACP-NCI		SCP-GNCI		ACP-GNCI	
		GAP-LP	GAP	IMP	GAP	IMP	GAP	IMP	GAP	IMP
10	VL	2.03%	1.52%	0.51%	1.34%	0.69%	1.64%	0.39%	1.49%	0.54%
	FL	15.35%	12.32%	3.03%	10.41%	4.94%	12.04%	3.31%	10.03%	5.32%
	FT	16.39%	13.50%	2.89%	11.98%	4.41%	13.58%	2.81%	12.34%	4.05%
30	FL	9.18%	7.49%	1.69%	7.31%	1.86%	7.75%	1.43%	6.99%	2.19%
	VT	0.72%	0.66%	0.06%	0.64%	0.08%	0.63%	0.09%	0.61%	0.11%
	FT	9.53%	8.59%	0.94%	8.45%	1.08%	8.62%	0.07%	8.43%	1.10%
40	VL	0.28%	0.26%	0.02%	0.23%	0.05%	0.26%	0.02%	0.26%	0.02%
	FL	1.84%	1.83%	0.01%	1.83%	0.01%	1.82%	0.02%	1.82%	0.02%
	VT	0.74%	0.73%	0.01%	0.72%	0.02%	0.72%	0.02%	0.72%	0.02%
	FT	1.34%	1.28%	0.06%	1.25%	0.09%	1.27%	0.07%	1.24%	0.10%
100	VL	1.17%	1.11%	0.06%	1.11%	0.06%	1.11%	0.06%	1.11%	0.06%
	FL	4.60%	4.55%	0.05%	4.55%	0.05%	4.55%	0.05%	4.54%	0.06%
	VT	1.66%	1.52%	0.14%	1.51%	0.15%	1.52%	0.14%	1.52%	0.14%
	FT	3.49%	3.36%	0.13%	3.35%	0.14%	3.36%	0.13%	3.36%	0.13%
200	VL	2.86%	2.81%	0.05%	2.81%	0.05%	2.81%	0.05%	2.81%	0.05%
	FL	6.13%	6.02%	0.11%	6.02%	0.11%	6.02%	0.11%	6.02%	0.11%
	VT	1.80%	1.77%	0.03%	1.77%	0.03%	1.77%	0.03%	1.77%	0.03%
	FT	4.22%	4.18%	0.04%	4.17%	0.05%	4.14%	0.08%	4.14%	0.08%
400	VL	3.71%	3.70%	0.01%	3.70%	0.01%	3.70%	0.01%	3.70%	0.01%
	FL	7.75%	7.72%	0.03%	7.72%	0.03%	7.71%	0.04%	7.71%	0.04%
	VT	5.17%	5.17%	0.00%	5.17%	0.00%	5.17%	0.00%	5.17%	0.00%
	FT	8.20%	8.18%	0.02%	8.18%	0.02%	8.18%	0.02%	8.18%	0.02%

TAB. 2 – Gap and improvement distribution according to problem characteristics, C problems

N , A , K	SCP-NCI		ACP-NCI		SCP-GNCI		ACP-GNCI	
	CPU	IMP	CPU	IMP	CPU	IMP	CPU	IMP
20, 230, 200	82%	0.06%	78%	0.06%	80%	0.07%	77%	0.07%
20, 300, 200	87%	0.05%	81%	0.05%	85%	0.06%	82%	0.06%
30, 520, 400	72%	0.02%	65%	0.02%	67%	0.02%	65%	0.02%
30, 700, 400	85%	0.00%	82%	0.00%	84%	0.00%	84%	0.00%

TAB. 3 – CPU reduction and gap improvement according to problem dimensions, large C problems

SCP-GNCI, and ACP-GNCI, respectively.

		SCP-NCI			ACP-NCI		SCP-GNCI		ACP-GNCI	
		GAP-LP	GAP	IMP	GAP	IMP	GAP	IMP	GAP	IMP
F01	C1	0.61%	0.56%	0.05%	0.56%	0.05%	0.56%	0.05%	0.56%	0.05%
	C2	1.19%	1.14%	0.05%	1.14%	0.05%	1.13%	0.06%	1.13%	0.06%
	C8	2.45%	2.28%	0.17%	2.22%	0.23%	2.25%	0.20%	2.21%	0.24%
F05	C1	4.28%	4.05%	0.23%	4.05%	0.23%	4.04%	0.24%	4.04%	0.24%
	C2	3.99%	3.74%	0.25%	3.69%	0.30%	3.68%	0.31%	3.65%	0.34%
	C8	4.33%	3.98%	0.35%	3.90%	0.42%	3.95%	0.38%	3.88%	0.45%
F10	C1	6.61%	6.29%	0.32%	6.29%	0.32%	6.28%	0.33%	6.28%	0.33%
	C2	6.10%	5.68%	0.42%	5.46%	0.64%	5.66%	0.44%	5.65%	0.45%
	C8	5.17%	4.79%	0.38%	4.69%	0.48%	4.72%	0.45%	4.61%	0.56%

TAB. 4 – Gap distribution according to fixed cost and capacity level, **R** problems

Table 5 shows the CPU reduction and the gap improvement for the **R** problems according to problem dimensions. The results indicate that the gap improvement generally decreases as the problem size increases. However, the CPU reduction is more significant on larger instances.

		SCP-NCI		ACP-NCI		SCP-GNCI		ACP-GNCI	
$ N , K ,$	$ A $	CPU	IMP	CPU	IMP	CPU	IMP	CPU	IMP
20, 100,	100	40%	0.45%	16%	0.53%	38%	0.46%	15%	0.50%
	200	57%	0.06%	43%	0.06%	50%	0.08%	35%	0.08%
	300	43%	0.11%	30%	0.11%	28%	0.11%	18%	0.11%
20, 200,	100	54%	0.15%	53%	0.15%	51%	0.16%	37%	0.16%
	200	74%	0.07%	69%	0.08%	74%	0.09%	66%	0.10%
	300	81%	0.08%	75%	0.08%	79%	0.09%	74%	0.09%

TAB. 5 – CPU reduction and gap improvement according to problem dimensions, **R** problems

To evaluate more precisely the performance of the proposed cutting-plane algorithms, we have run the B&B and the B&C of CPLEX on the formulation obtained from ACP-GNCI. A limit of 10 hours of CPU time is imposed for each instance. The results are compared to those of the B&C of CPLEX on the strong formulation. We are interested in the category of problems for which CPLEX fails to find the optimal solution within the CPU time allowed. This category is composed of 68 problems (28 **C** problems and 40 **R** problems). The results show that the B&B on the new formulation found a better solution for 27 problems out of 68, compared to 19 better solutions found by the B&C on the new formulation, and 22 by the B&C on the strong formulation. Thus, within a limit of 10 hours of CPU time, the B&B of CPLEX performs better than

PSet	P	OPT	<-2%	(-2% - -1%)	(-1% - 0%)	(0% - 1%)	(1% - 2%)	>2%
C	28	3	1	1	6	11	-	6
R	40	1	2	6	9	17	2	3

TAB. 6 – Distribution of relative improvement for the 68 unsolved problems

its B&C when applied to the new formulation. For the 68 unsolved problems, Table 6 gives the distribution of the best feasible solutions found by the B&B on the new formulation when compared to those found by the B&C on the strong formulation. Among the 28 **C** problems, 3 are solved to optimality, a better solution is obtained in 17 cases and a worst solution in 8 cases. The improvement in the solution quality is greater than 2% for 6 problems. Among the 40 **R** problems, one problem is solved to optimality, 22 solutions are better and 17 solutions are worst. The gap for the worst solutions is more than 2% for only 2 problems.

These results emphasize the performance of the cutting-plane method to improve the quality of the bounds, and hence the formulation of the problem. This suggests implementing a B&C algorithm, where the cutting-plane procedure is used at each node in the enumeration tree.

8 Conclusion

In this paper, we have presented a cutting-plane algorithm based on cutset inequalities for the multicommodity capacitated fixed charge network design problem. We have described three families of valid inequalities : the cover inequalities, the minimum cardinality inequalities, and the network cutset inequalities. The latter constitutes a new family of VI derived from the cutset structure. We have developed computationally efficient separation heuristics to be used in conjunction with a cutset generation algorithm. We have also developed new lifting strategies, adapted to cover and minimum cardinality inequalities. Finally, we have presented computational results conducted on a large set of instances.

In the light of our computational study, we can conclude that the proposed cutting-plane algorithm achieves better lower bounds with competitive computational effort

when compared to the state-of-the-art MIP software package CPLEX. This conclusion points out to interesting research avenues. The cutting-plane procedure can either be included in a branch-and-cut algorithm or a cut-and-branch algorithm, since the cuts used are valid at all nodes in the enumeration tree. An additional important advantage of our cutting-plane method is that it uses VI derived from any cutset structure, which constitutes the most common structure found in almost all capacitated network design problems. Thus, it is interesting to investigate the usefulness of the proposed method to improve the formulations of other capacitated network design formulations.

Bibliographie

- [1] Aardal K. 1998, *Capacitated Facility Location : Separation Algorithms and Computational Experience*, Mathematical Programming 81, 149-175.
- [2] Aardal K., Pochet Y. and Wolsey L.A. 1995, *Capacitated Facility Location : Valid Inequalities and Facets*, Mathematics of Operations Research 20, 562-582.
- [3] Balas E. 1975, *Facets of the Knapsack Polytope*, Mathematical Programming 8, 146-164.
- [4] Barahona F. 1996, *Network Design Using Cut Inequalities*, SIAM Journal of Optimization 6, 823-837.
- [5] Bienstock D., Chopra S., Günlük O. and Tsai C.Y. 1998, *Minimum Cost Capacity Installation for Multicommodity Network Flows*, Mathematical Programming 81, 177-199.
- [6] Bienstock D. and Günlük O. 1996, *Capacitated Network Design-Polyhedral Structure and Computation*, INFORMS Journal on Computing 8, 243-259.
- [7] Chouman M., Crainic T.G. and Gendron B. 2001, *Revue des inégalités valides pertinentes aux problèmes de conception de réseaux*, INFOR, forthcoming.
- [8] CPLEX Optimization, Inc. 2002, *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, version 7.1.
- [9] Crainic, T.G. 1999, *Network Design in Freight Transportation*, European Journal of Operational Research, 122, (2), 272-288.
- [10] Crainic T.G., Frangioni A. and Gendron B. 2002, *Bundle-Based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems*, Discrete Applied Mathematics, 112, 73-99.

- [11] Crainic T.G. and Gendreau M. 2002, *Cooperative Parallel Tabu Search for Capacitated Network Design*, Journal of Heuristics 8, 601-627.
- [12] Crainic T.G., Gendreau M. and Farvolden J.M. 2000, *A Simplex-Based Tabu Search Method for Capacitated Network Design*, INFORMS Journal on Computing 12, 223-236.
- [13] Crainic T.G., Gendron B. and Hernu G. 2002 *A Slope Scaling/Lagrangian Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design*, Publication CRT, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [14] Gendron B. and Crainic T.G. 1994, *Relaxations for Multicommodity Capacitated Network Design Problems*, Publication CRT-945, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [15] Gendron B. and Crainic T.G. 1996, *Bounding Procedures for Multicommodity Capacitated Fixed Charge Network Design Problems*, Publication CRT-96-06, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [16] Gendron B., Crainic T.G. and Frangioni A. 1998, *Multicommodity Capacitated Network Design*, *Telecommunications Network Planning*, Sansó B. and Soriano P. (eds.), Kluwer Academic Publishers, Norwell MA., 1-19.
- [17] Ghamlouche I., Crainic T.G. et Gendreau M. 2001, *Cycle-Based Neighborhoods for Fixed-Charge Capacitated Multicommodity Network Design*, Operations Research, forthcoming.
- [18] Ghamlouche I., Crainic T.G. et Gendreau M. 2002, *Path Relinking, Cycle-Based Neighborhoods and Capacitated Multicommodity Network Design*, Publication CRT, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [19] Gu Z., Nemhauser G.L. and Savelsbergh M.W.P. 1998, *Lifted Cover Inequalities for 0-1 Integer Programs : Computation*, INFORMS Journal on Computing 10, 427-437.
- [20] Gu Z., Nemhauser G.L. and Savelsbergh M.W.P. 1999, *Lifted Cover Inequalities for 0-1 Integer Programs : Complexity*, INFORMS Journal on Computing 11, 117-123.

- [21] Günlük O. 1999, *A Branch-and-Cut Algorithm for Capacitated Network Design Problems*, *Mathematical Programming* 86, 17-39.
- [22] Holmberg K. and Yuan D. 2000, *A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem*, *Operations Research* 48, 461-481.
- [23] Leung J.M.Y. and Magnanti T.L. 1989, *Valid Inequalities and Facets of the Capacitated Plant Location Problems*, *Mathematical Programming* 44, 271-291.
- [24] Magnanti T.L., Mirchandani P.B. and Vachani R. 1993, *The Convex Hull of Two Core Capacitated Network Design Problems*, *Mathematical Programming* 60, 233-250.
- [25] Magnanti T.L., Mirchandani P.B. and Vachani R. 1995, *Modeling and Solving the Two-Facility Capacitated Network Loading Problem*, *Operations Research* 43, 142-157.
- [26] Magnanti T.L. and Wong R.T. 1984, *Network Design and Transportation Planning : Models and Algorithms*, *Transportation Science* 18, 1-55.
- [27] Martello S. and Toth P. 1990, *Knapsack Problems, Algorithms and Computer Implementations*, Wiley-Interscience.
- [28] Martello S. and Toth P. 1997, *Upper Bounds and Algorithms for Hard 0-1 Knapsack Problems*, *Operations Research* 45, 768-778.
- [29] Minoux M. 1989, *Network Synthesis and Optimum Network Design Problems : Models, Solution Methods and Applications*, *Networks* 19, 313-360.
- [30] Nemhauser G.L., Savelsbergh M.W.P. and Sigismondi G. 1994, *MINTO, a Mixed Integer Optimizer*, *Operations Research Letters* 15, 47-58.
- [31] Nemhauser G.L. and Wolsey L.A. 1988, *Integer and Combinatorial Optimization*, Wiley-Interscience.
- [32] Sellmann M., Kliewer G. and Koberstein A. 2002, *Capacitated Network Design, Cardinality Cuts and Coupled Variable Fixing Algorithms based on Lagrangian Relaxations*, Technical report tr-ri-02-234, Department of Computer Science, University of Paderborn.

- [33] Van Roy T.J. and Wolsey L.A. 1987, *Solving Mixed Integer Programming Problems Using Automatic Reformulation*, Operations Research 35, 45-57.
- [34] Wolsey L.A. 1975, *Faces of Linear Inequalities in 0-1 variables*, Mathematical Programming 8, 165-178.
- [35] Wolsey L.A. 1998, *Integer Programming*, Wiley-Interscience.
- [36] Xpress-MP 2002, *Xpress-MP Integrated into MIMI*, Modeling and Optimization Software, Dash Optimization Inc., New York, NY.

Appendix 1

Here we present variants of the NCI and the GNCI presented in Section 5. These variants are, theoretically, stronger than the original versions. In fact, the basic idea behind these variants, is to try to identify an appropriate decomposition of the set of commodities K on each arc into two subsets such that the corresponding inequality includes only interested flows at the left hand side. By doing so, we minimize also the value of the right hand side, thus the resulting inequality is stronger. Let $L \subseteq K$, $x_{ij}^L = \sum_{k \in L} x_{ij}^k$ and $b_{ij}^L = \min\{u_{ij}, \sum_{k \in L} d^k\}$. Further assume that for any arc $(i, j) \in A$, K is partitioned into two subsets K_{ij}^1 and K_{ij}^0 . Considering these assumptions, we introduce the *single-arc network cutset inequality (SNCI)*, defined by Proposition 4.

Proposition 4 : Let $(i, j) \in (\bar{S}, S)$, $C_1 \subseteq (\bar{S}, S) \setminus \{(i, j)\}$ and $C_2 \subseteq (S, \bar{S})$. Then,

$$\sum_{(r,t) \in C_1} x_{rt}^{K_{rt}^1} + x_{ij}^K \leq (\sum_{(r,t) \in C_2} b_{rt}^{K_{rt}^1} + d_{(S,\bar{S})})y_{ij} + \sum_{(r,t) \in C_2} x_{rt}^{K_{rt}^0} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}^K + (1 - y_{ij}) \sum_{(r,t) \in C_1} b_{rt}^{K_{rt}^1} \quad (16)$$

is a VI for the MCND.

Proof : Let (x, y) be a feasible solution of the MCND. Then,

- if $y_{ij} = 0$, then $x_{ij}^K = 0$ and (16) is valid since $\sum_{(r,t) \in C_1} x_{rt}^{K_{rt}^1} \leq \sum_{(r,t) \in C_1} b_{rt}^{K_{rt}^1}$;
- if $y_{ij} = 1$,

$$\begin{aligned} \sum_{(r,t) \in C_1} x_{rt}^{K_{rt}^1} + x_{ij}^K &\leq \sum_{(r,t) \in C_2} x_{rt}^K + d_{(S,\bar{S})} + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}^K, \\ &\quad \text{(flow conservation)} \\ &\leq \sum_{(r,t) \in C_2} b_{rt}^{K_{rt}^1} + d_{(S,\bar{S})} + \sum_{(r,t) \in C_2} x_{rt}^{K_{rt}^0} \\ &\quad + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}^K, \quad \text{(capacity constraint)} \\ &\leq (\sum_{(r,t) \in C_2} b_{rt}^{K_{rt}^1} + d_{(S,\bar{S})})y_{ij} + \sum_{(r,t) \in C_2} x_{rt}^{K_{rt}^0} \\ &\quad + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt}^K. \quad \text{(since } y_{ij} = 1) \end{aligned}$$

Note that, the SNCI (16) is stronger than inequality (14) either when $\sum_{(r,t) \in C_1} b_{rt}^{K_{rt}^1} <$

$\sum_{(r,t) \in C_1} u_{rt}$ or when $\sum_{(r,t) \in C_2} b_{rt}^{K_{rt}^1} < \sum_{(r,t) \in C_2} u_{rt}$.

Similarly, based on the same idea, we define the *multiple arc network cutset inequality (MNCI)*, derived from the GNCI and introduced in the following proposition.

Proposition 5 : Let $C \subseteq (\bar{S}, S), C_1 \subseteq (\bar{S}, S) (C \cap C_1 = \emptyset), C_2 \subseteq (S, \bar{S})$ and $Y_C = \max_{(r,t) \in C} y_{rt}$. Then,

$$\sum_{(r,t) \in C_1} x_{rt}^{K^1} + \sum_{(r,t) \in C} x_{rt}^K \leq (\sum_{(r,t) \in C_2} b_{rt}^{K^1} + d_{(S, \bar{S})}) Y_C + \sum_{(r,t) \in C_2} x_{rt}^{K^0} + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}^K + (1 - Y_C) \sum_{(r,t) \in C_1} b_{rt}^{K^1}, \quad (17)$$

is a VI for the MCND.

Proof : Let (x, y) be a feasible solution of the MCND. Then,

- if $y_{rt} = 0, \forall (r, t) \in C$, then $Y_C = 0$ and the MNCI is valid since

$$\sum_{(r,t) \in C_1} x_{rt}^{K^1} \leq \sum_{(r,t) \in C_1} b_{rt}^{K^1};$$

- if there exists $(i, j) \in C$ such that $y_{ij} = 1$, then $Y_C = 1$ and

$$\begin{aligned} \sum_{(r,t) \in C_1} x_{rt}^{K^1} + \sum_{(r,t) \in C} x_{rt}^K &\leq \sum_{(r,t) \in C_2} x_{rt}^K + d_{(S, \bar{S})} + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}^K, \\ &\quad \text{(flow conservation)} \\ &\leq \sum_{(r,t) \in C_2} b_{rt}^{K^1} + d_{(S, \bar{S})} + \sum_{(r,t) \in C_2} x_{rt}^{K^0} \\ &\quad + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}^K, \quad \text{(capacity constraint)} \\ &\leq (\sum_{(r,t) \in C_2} b_{rt}^{K^1} + d_{(S, \bar{S})}) Y_C + \sum_{(r,t) \in C_2} x_{rt}^{K^0} \\ &\quad + \sum_{(r,t) \in (S, \bar{S}) \setminus C_2} x_{rt}^K. \quad \text{(since } Y_C = 1) \end{aligned}$$

Separation :

Given a set $S \subset N$ and an arc $(i, j) \in (\bar{S}, S)$, the separation problem for the SNCI consists in finding the sets $K_{rt}^1 \subseteq K$, for each arc $(r, t), C_1 \subseteq (\bar{S}, S)$ and $C_2 \subseteq (S, \bar{S})$ such that the violation of the VI with respect to the current LP solution, (\bar{x}, \bar{y}) , is maximal. Clearly, it is optimal to set

$$K_{rt}^1 = \{k \in K : \bar{x}_{rt}^k > 0\}, \quad \forall (r, t) \in (\bar{S}, S) \cup (S, \bar{S}),$$

$$C_1 = \{(r, t) \in (\bar{S}, S) \setminus \{(i, j)\} : \bar{x}_{rt}^{K^1} > (1 - \bar{y}_{ij}) b_{rt}^{K^1}\},$$

$$C_2 = \{(r, t) \in (S, \bar{S}) : b_{rt}^{K^1} \bar{y}_{ij} < \bar{x}_{rt}^{K^1}\}.$$

Note that, with this choice of C_1 , the SNCI (16) always dominates inequality (14).

The separation problem for the MNCI is somewhat similar to the GNCI. Given the above definition of K_{rt}^1 for each arc (r, t) , it consists in finding three sets, $C \subseteq (\bar{S}, S)$, $C_1 \subseteq (\bar{S}, S)$ ($C \cap C_1 = \emptyset$), and $C_2 \subseteq (S, \bar{S})$, such that the violation with respect to the current LP solution, (\bar{x}, \bar{y}) , is maximal. Note that, similarly to the GNCI, once we have determined C , we can efficiently construct the sets C_1 and C_2 as in the generation of the SNCI. To identify the set C , we consider the following MIP problem :

$$\begin{aligned} \max \quad & \sum_{(i,j) \in (\bar{S}, S)} \bar{x}_{ij}^K Z_{ij} - d_{(S, \bar{S})} Y, \\ & Y \geq \bar{y}_{ij} Z_{ij}, \quad \forall (i, j) \in (\bar{S}, S), \\ & Z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in (\bar{S}, S), \\ & 0 \leq Y \leq 1, \end{aligned}$$

where Z is the characteristic vector of C . To solve this problem, we have devised a heuristic procedure, **Gen-MNCI**, based on the fact that, if Y is fixed, the problem becomes easy to solve. Next, we summarize the outline of this procedure, which accepts as input the values of two user-supplied parameters, \bar{Y} and δ .

Procedure Gen-MNCI :

- 1- Fix Y_C at some small value, $Y_C \leftarrow \bar{Y}$, $0 < \bar{Y} < 1$.
- 2- Let $C \leftarrow \{(i, j) \in (\bar{S}, S) \mid \bar{y}_{ij} \leq Y_C\}$ and readjust Y_C by setting $Y_C \leftarrow \max_{(i,j) \in C} \bar{y}_{ij}$.
- 3- Let $K_{ij}^1 = \{k \in K : \bar{x}_{ij}^k > 0\}$, $\forall (i, j) \in (\bar{S}, S) \cup (S, \bar{S})$; construct C_1 and C_2 as follows :

$$- C_1 \leftarrow \{(i, j) \in (\bar{S}, S) \mid b_{ij}^{K^1} (1 - Y_C) < \bar{x}_{ij}^{K^1}\},$$

$$- C_2 \leftarrow \{(i, j) \in (S, \bar{S}) \mid b_{ij}^{K^1} Y_C < \bar{x}_{ij}^{K^1}\}.$$

- 4- If the MNCI is violated, then add it to the formulation, along with the inequalities

- $Y_C \geq y_{ij}, \forall (i, j) \in C,$
- $Y_C \leq \sum_{(i,j) \in C} y_{ij},$
- $0 \leq Y_C \leq 1.$

5- Otherwise, give Y_C a smaller value, $Y_C \leftarrow Y_C - \delta;$

- if $Y_C > 0,$ return to 2,
- else STOP.

Note that, if a violated MNCI is generated at step 4, one has to add the violated inequality, but also a new continuous variable Y_C and a set of linear inequalities equivalent to $Y_C = \max_{(i,j) \in C} y_{ij}.$

The investigation and the study of the usefulness of these inequalities is under developments.

Appendix 2

Results for each problem instance is given in the following tables. Tables 7 to 17 show results for simple and advanced cutting-plane using the NCI. While Tables 18 to 28 show results for simple and advanced cutting-plane using the GNCI. For each problem instance, the following information is displays :

- **DESCR** : The description of the problem given by the number of nodes, arcs, and commodities, respectively. Additional letters are used to characterize the fixed cost level relatively to the transportation cost and the capacity level compared to the total demand.
- **SOL** : The optimal (indicated by *) or the best feasible solution obtained by using the branch-and-cut (B&C) of CPLEX (version 7.1, 2002). A limit of 10 hours of computation was imposed for each instance. For the problems where CPLEX was unable to find feasible solution within the time allowed, we report the best feasible solution found in Ghamlouche, Crainic, and Gendreau (2002). The CPU time elapsed in seconds is given under each solution. The letter (t) indicates that the solution is feasible but its optimality could not be verified within 10 hours with the B&C of CPLEX.
- **GAP-LP** : Displays the gap of the strong LP bound computed with respect to the best solution (SOL).
- The last four columns indicate the gap of the lower bound and the number of valid inequalities for each family generated in the simple and advanced cutting-plane using NCI or GNCI. The CPU time elapsed in second is given under each GAP.

All experiments are performed on a Sun Enterprise 10000 with 64 Gigabytes of RAM and operating under Solaris 2.7.

DESCR N , A , K	SOL		SCP-NCI (SI,LCI,LMCI,NCI)		ACP-NCI (SI,LCI,LMCI,NCI)	
	SOL	GAP-LP	GAP		GAP	
25-100-10	14712*	0.69	0.69	(33, 14, 16, 22)	0.67	(33, 15, 17, 22)
V-L	(0.36)	(0.03)	(0.3)		(0.9)	
25-100-10	14941*	12.87	9.56	(129, 17, 29, 51)	9.31	(136, 20, 31, 54)
F-L	(53.6)	(0.5)	(1.4)		(5.1)	
25-100-10	49899*	9.76	7.22	(61, 12, 12, 41)	5.08	(61, 28, 15, 45)
F-T	(40.6)	(0.3)	(0.6)		(5.2)	
25-100-30	365272*	0.24	0.24	(75, 20, 20, 27)	0.24	(75, 20, 20, 27)
V-T	(16.6)	(0.6)	(0.9)		(1.9)	
25-100-30	37055*	8.26	7.08	(178, 27, 41, 36)	7.07	(178, 27, 41, 37)
F-L	(1727.9)	(4.7)	(3.6)		(6.7)	
25-100-30	85530*	3.63	3.43	(100, 20, 20, 27)	3.42	(102, 22, 21, 28)
F-T	(534.2)	(1.6)	(1.5)		(5.7)	
20-230-40	423848*	0.23	0.20	(114, 27, 47, 26)	0.19	(114, 28, 48, 26)
V-L	(10.4)	(0.3)	(2.9)		(7.2)	
20-230-40	371475*	0.71	0.69	(82, 13, 11, 4)	0.69	(82, 13, 12, 4)
V-T	(52.4)	(0.6)	(2.2)		(6.9)	
20-230-40	643036*	1.48	1.43	(94, 20, 17, 3)	1.37	(95, 22, 19, 3)
F-T	(671.7)	(0.6)	(2.4)		(15.9)	
20-300-40	429398*	0.33	0.33	(100, 29, 48, 11)	0.33	(100, 29, 48, 11)
V-L	(3.76)	(0.3)	(3.0)		(7.3)	
20-300-40	586077*	1.84	1.83	(148, 21, 30, 21)	1.83	(148, 21, 30, 21)
F-L	(145.5)	(1.0)	(4.4)		(11.6)	
20-300-40	464509*	0.77	0.77	(71, 14, 19, 6)	0.76	(71, 15, 20, 6)
V-T	(83.8)	(0.7)	(3.2)		(11.8)	
20-300-40	604198*	1.21	1.14	(95, 22, 23, 3)	1.14	(95, 22, 23, 3)
F-T	(59.8)	(0.6)	(3.0)		(9.4)	
20-230-200	94386	3.26	3.20	(1334, 20, 21, 2)	3.20	(1334, 20, 21, 2)
V-L	(t)	(1083.5)	(193.0)		(225.5)	
20-230-200	141737.4	6.84	6.66	(1499, 17, 23, 0)	6.66	(1499, 17, 23, 0)
F-L	(t)	(2235.1)	(378.3)		(405.8)	
20-230-200	97914	2.29	2.29	(1087, 22, 25, 0)	2.29	(1087, 22, 25, 0)
V-T	(t)	(594.3)	(166.8)		(208.7)	
20-230-200	137271	4.17	4.17	(1257, 22, 17, 0)	4.17	(1257, 22, 18, 0)
F-T	(t)	(2431.1)	(258.3)		(302.1)	
20-300-200	74972.4	2.46	2.46	(943, 9, 10, 0)	2.46	(943, 9, 10, 0)
V-L	(t)	(1332.1)	(197.9)		(277.6)	
20-300-200	117306	5.43	5.39	(1124, 13, 15, 0)	5.39	(1124, 13, 15, 0)
F-L	(t)	(3290.3)	(310.7)		(393.5)	
20-300-200	74991	1.31	1.26	(755, 11, 11, 0)	1.26	(755, 11, 11, 0)
V-T	(t)	(672.3)	(124.3)		(201.9)	
20-300-200	108252	4.26	4.17	(835, 14, 13, 0)	4.17	(835, 14, 13, 0)
F-T	(t)	(2605.1)	(252.7)		(324.7)	

TAB. 7 – Computational Results Simple and Advanced Cutting-Plane-NCI, C problems

DESCR N , A , K	SOL		SCP-NCI		ACP-NCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,NCI)	GAP	(SI,LCI,LMCI,NCI)
100-400-10	28423*	3.37	2.35	(47, 15, 17, 56)	2.12	(51, 34, 39, 58)
V-L	(84.8)	(0.3)	(1.0)		(34.3)	
100-400-10	24436	17.82	15.08	(365, 17, 14, 193)	10.79	(517, 56, 57, 221)
F-L	(t)	(12.8)	(29.5)		(185.8)	
100-400-10	66364	23.02	19.78	(202, 21, 17, 176)	19.17	(232, 30, 22, 209)
F-T	(t)	(13.0)	(17.1)		(102.6)	
100-400-30	385544	1.21	1.08	(146, 35, 39, 109)	1.06	(146, 37, 41, 109)
V-T	(t)	(6.1)	(4.0)		(44.8)	
100-400-30	50496	10.11	7.91	(922, 50, 76, 289)	7.54	(1007, 56, 78, 299)
F-L	(t)	(116.3)	(148.8)		(355.4)	
100-400-30	141278	15.44	13.76	(298, 45, 50, 178)	13.52	(312, 52, 55, 183)
F-T	(t)	(110.2)	(42.2)		(158.9)	
30-520-100	53966	1.74	1.61	(470, 27, 27, 11)	1.61	(470, 27, 27, 11)
V-L	(t)	(65.9)	(44.8)		(140.2)	
30-520-100	95294	5.37	5.29	(954, 39, 47, 15)	5.29	(954, 39, 47, 15)
F-L	(t)	(1106.5)	(224.4)		(337.4)	
30-520-100	52085	1.45	1.34	(341, 26, 29, 7)	1.34	(341, 26, 29, 7)
V-T	(t)	(59.9)	(38.9)		(131.1)	
30-520-100	98357	4.41	4.25	(662, 31, 42, 9)	4.24	(663, 32, 43, 9)
F-T	(t)	(863.1)	(156.0)		(520.8)	
30-700-100	47603*	0.61	0.61	(424, 30, 32, 10)	0.60	(430, 31, 42, 10)
V-L	(1736.1)	(57.5)	(54.6)		(270.9)	
30-700-100	60525	3.83	3.81	(801, 45, 61, 16)	3.81	(801, 45, 61, 16)
F-L	(t)	(758.0)	(149.6)		(360.2)	
30-700-100	45944.5	1.88	1.70	(418, 22, 25, 4)	1.70	(418, 22, 25, 4)
V-T	(t)	(166.3)	(59.5)		(280.8)	
30-700-100	55082	2.58	2.48	(507, 33, 32, 7)	2.48	(510, 33, 32, 7)
F-T	(t)	(356.0)	(102.2)		(313.2)	
30-520-400	112997.5	1.09	1.08	(1426, 8, 7, 0)	1.08	(1426, 8, 7, 0)
V-L	(t)	(5356.8)	(1804.3)		(2236.2)	
30-520-400	159007	7.75	7.72	(1831, 16, 18, 0)	7.72	(1831, 16, 18, 0)
F-L	(33641.2)	(10649.4)	(2822.6)		(3406.1)	
30-520-400	119744	4.74	4.74	(1106, 5, 2, 0)	4.74	(1106, 5, 2, 0)
V-T	(20461.9)	(4910.5)	(1536.3)		(1890.3)	
30-520-400	163352	8.32	8.29	(1615, 10, 6, 0)	8.29	(1615, 10, 6, 0)
F-T	(59820.1)	(11706.8)	(2749.9)		(3356.9)	
30-700-400	103138	6.33	6.33	(1859, 18, 18, 0)	6.33	(1859, 18, 18, 0)
V-L	(17881.3)	(19428.7)	(2740.7)		(3564.6)	
30-700-400	144615	X	9.60	(2021, 14, 8, 0)	9.60	(2021, 14, 8, 0)
F-L	(57574.8)	(t)	(4275.7)		(5196.5)	
30-700-400	99606	5.61	5.61	(1349, 8, 7, 0)	5.61	(1349, 8, 7, 0)
V-T	(31405.7)	(10923.9)	(2285.8)		(2877.1)	
30-700-400	138777	8.07	8.07	(1527, 7, 4, 0)	8.07	(1527, 7, 4, 0)
F-T	(48693.7)	(34057.0)	(4077.2)		(4953.2)	

TAB. 8 - Computational Results Simple and Advanced Cutting-Plane-NCI, C problems

DESCR 20,100,40	SOL			SCP-NCI		ACP-NCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,NCI)	GAP	(SI,LCI,LMCI,NCI)	
R10-F01-C1	200087* (10.7)	0.58 (0.2)	0.24 (1.1)	(92, 25, 39, 17)	0.24 (1.9)	(92, 25, 39, 17)	
R10-F05-C1	346813.5* (788.2)	3.84 (2.1)	3.22 (4.8)	(209, 30, 48, 49)	3.22 (7.1)	(209, 30, 48, 49)	
R10-F10-C1	488015* (563.3)	4.27 (4.9)	3.33 (6.3)	(313, 30, 53, 61)	3.33 (8.6)	(313, 30, 53, 61)	
R10-F01-C2	229196* (184.3)	2.32 (0.9)	2.09 (1.6)	(97, 13, 16, 14)	2.09 (3.1)	(97, 13, 16, 14)	
R10-F05-C2	411664* (5123.2)	6.18 (3.4)	5.59 (3.9)	(176, 25, 36, 39)	5.59 (6.6)	(176, 25, 36, 39)	
R10-F10-C2	609104* (27260.5)	7.18 (7.3)	6.41 (6.5)	(218, 27, 45, 54)	5.60 (23.7)	(241, 30, 47, 56)	
R10-F01-C8	486895* (391.3)	4.78 (5.8)	4.16 (2.4)	(85, 13, 15, 21)	3.73 (8.4)	(80, 15, 15, 21)	
R10-F05-C8	951056* (555.5)	7.98 (7.7)	6.24 (3.4)	(117, 16, 22, 19)	5.58 (11.9)	(118, 20, 22, 19)	
R10-F10-C8	1421740* (608.4)	8.90 (9.4)	6.65 (4.1)	(137, 17, 24, 20)	5.73 (12.2)	(137, 23, 26, 20)	

TAB. 9 – Computational Results Simple and Advanced Cutting-Plane-NCI, R10

DESCR 20,100,100	SOL			SCP-NCI		ACP-NCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,NCI)	GAP	(SI,LCI,LMCI,NCI)	
R11-F01-C1	714431* (392.5)	0.51 (8.0)	0.45 (8.5)	(274, 19, 22, 30)	0.45 (11.5)	(274, 19, 22, 30)	
R11-F05-C1	1265985 (t)	3.53 (52.4)	2.78 (21.7)	(559, 26, 42, 52)	2.78 (61.41)	(559, 26, 42, 52)	
R11-F10-C1	1846295 (t)	5.50 (130.5)	4.82 (37.6)	(731, 29, 39, 43)	4.82 (44.8)	(731, 29, 39, 43)	
R11-F01-C2	870451* (2960.1)	1.12 (19.8)	1.10 (8.9)	(218, 13, 17, 24)	1.10 (12.4)	(218, 13, 17, 24)	
R11-F05-C2	1623640* (21739.5)	3.42 (85.7)	2.57 (22.9)	(341, 23, 29, 26)	2.09 (62.7)	(358, 28, 31, 26)	
R11-F10-C2	2414060 (t)	4.68 (104.9)	3.38 (27.8)	(428, 25, 38, 34)	3.03 (47.1)	(439, 27, 38, 34)	
R11-F01-C8	2294912* (31.5)	0.26 (15.8)	0.21 (16.1)	(368, 25, 25, 21)	0.21 (17.5)	(368, 25, 25, 21)	
R11-F05-C8	3507100* (79.8)	0.56 (27.6)	0.27 (22.3)	(427, 27, 30, 27)	0.27 (23.3)	(427, 27, 30, 27)	
R11-F10-C8	4579353* (48.5)	0.27 (21.8)	0.23 (19.2)	(469, 30, 36, 35)	0.23 (20.6)	(469, 30, 36, 35)	

TAB. 10 – Computational Results Simple and Advanced Cutting-Plane-NCI, R11

DESCR 20,100,200	SOL	GAP-LP	GAP	SCP-NCI (SI,LCI,LMCI,NCI)	GAP	ACP-NCI (SI,LCI,LMCI,NCI)
R12-F01-C1	1639443* (4436.7)	0.83 (54.3)	0.82 (27.5)	(368, 11, 13, 24)	0.82 (33.4)	(368, 11, 13, 24)
R12-F05-C1	3403074.2 (t)	3.99 (400.8)	3.87 (69.6)	(710, 23, 29, 29)	3.87 (82.9)	(710, 23, 29, 29)
R12-F10-C1	5276171 (t)	6.47 (969.9)	6.06 (85.0)	(826, 25, 41, 28)	6.06 (97.9)	(826, 25, 41, 28)
R12-F01-C2	2303557* (1739.8)	0.61 (122.9)	0.55 (35.6)	(324, 16, 18, 20)	0.55 (40.1)	(324, 16, 18, 20)
R12-F05-C2	4669799* (3145.6)	0.82 (243.0)	0.57 (45.8)	(408, 24, 30, 18)	0.55 (62.7)	(409, 26, 30, 18)
R12-F10-C2	7100019* (1683.8)	0.69 (624.2)	0.24 (52.8)	(521, 27, 42, 19)	0.23 (62.9)	(521, 28, 42, 19)
R12-F01-C8	7635270* (78.1)	0.02 (68.1)	0.02 (58.4)	(1221, 40, 38, 22)	0.02 (58.8)	(1221, 40, 38, 22)
R12-F05-C8	10067742* (61.5)	0.01 (54.0)	0.01 (53.5)	(1352, 40, 36, 30)	0.01 (52.5)	(1352, 40, 36, 30)
R12-F10-C8	11967768* (50.3)	0.00 (48.9)	0.00 (48.6)	(1519, 40, 40, 31)	0.00 (47.6)	(1519, 40, 40, 31)

TAB. 11 – Computational Results Simple and Advanced Cutting-Plane-NCI, R12

DESCR 20,200,40	SOL	GAP-LP	GAP	SCP-NCI (SI,LCI,LMCI,NCI)	GAP	ACP-NCI (SI,LCI,LMCI,NCI)
R13-F01-C1	142947* (6.4)	0.31 (0.5)	0.31 (3.6)	(179, 33, 60, 24)	0.31 (5.9)	(179, 33, 60, 24)
R13-F05-C1	263800* (547.9)	4.82 (7.4)	4.73 (12.4)	(527, 33, 60, 23)	4.73 (20.4)	(527, 33, 60, 23)
R13-F10-C1	365800* (116.6)	5.63 (13.7)	5.62 (22.1)	(706, 37, 68, 33)	5.62 (30.5)	(706, 37, 68, 33)
R13-F01-C2	150977* (90.1)	1.76 (0.7)	1.68 (4.2)	(171, 25, 38, 13)	1.68 (9.8)	(171, 25, 38, 13)
R13-F05-C2	282682* (3551.8)	4.87 (11.1)	4.69 (11.2)	(359, 32, 50, 31)	4.69 (20.7)	(359, 32, 50, 31)
R13-F10-C2	406790* (29695.6)	6.81 (21.2)	6.44 (19.7)	(463, 32, 55, 55)	6.40 (39.4)	(468, 32, 56, 55)
R13-F01-C8	208088 (t)	3.53 (7.2)	2.95 (4.1)	(64, 14, 13, 14)	2.90 (21.0)	(64, 16, 14, 14)
R13-F05-C8	446997 (t)	5.47 (31.3)	4.92 (11.7)	(121, 28, 35, 33)	4.92 (21.1)	(121, 28, 35, 33)
R13-F10-C8	698280 (t)	5.31 (54.6)	4.73 (14.6)	(176, 26, 32, 43)	4.73 (26.4)	(176, 26, 32, 43)

TAB. 12 – Computational Results Simple and Advanced Cutting-Plane-NCI, R13

DESCR 20,200,100	SOL	GAP-LP	GAP	SCP-NCI (SI,LCI,LMCI,NCI)	GAP	ACP-NCI (SI,LCI,LMCI,NCI)
R14-F01-C1	403414* (495.7)	0.53 (12.9)	0.53 (15.5)	(442, 26, 43, 14)	0.53 (27.8)	(442, 26, 43, 14)
R14-F05-C1	750091 (t)	5.04 (123.4)	5.03 (62.8)	(996, 34, 57, 39)	5.03 (86.0)	(996, 34, 57, 39)
R14-F10-C1	1078595 (t)	7.30 (319.2)	7.28 (131.7)	(1319, 37, 60, 70)	7.28 (155.8)	(1319, 37, 60, 70)
R14-F01-C2	437607* (1327.3)	0.91 (18.3)	0.91 (15.8)	(326, 19, 22, 9)	0.91 (25.8)	(326, 19, 22, 9)
R14-F05-C2	852330 (t)	4.35 (223.1)	4.29 (59.8)	(622, 26, 30, 48)	4.29 (83.3)	(622, 26, 30, 48)
R14-F10-C2	1230749.7 (t)	5.19 (515.8)	4.96 (90.3)	(778, 37, 37, 79)	4.96 (111.0)	(778, 37, 37, 79)
R14-F01-C8	670064 (t)	2.50 (121.4)	2.46 (21.1)	(143, 13, 13, 30)	2.46 (30.8)	(143, 13, 13, 30)
R14-F05-C8	1643376 (t)	6.09 (321.1)	6.05 (49.4)	(280, 24, 35, 29)	6.05 (71.8)	(280, 24, 36, 29)
R14-F10-C8	2630332 (t)	6.38 (591.3)	6.22 (57.1)	(357, 27, 49, 26)	6.21 (98.6)	(357, 27, 49, 28)

TAB. 13 – Computational Results Simple and Advanced Cutting-Plane-NCI, R14

DESCR 20,200,200	SOL	GAP-LP	GAP	SCP-NCI (SI,LCI,LMCI,NCI)	GAP	ACP-NCI (SI,LCI,LMCI,NCI)
R15-F01-C1	1000787* (5384.7)	0.66 (95.5)	0.66 (72.8)	(712, 18, 16, 22)	0.66 (94.1)	(712, 18, 16, 22)
R15-F05-C1	1979413 (t)	4.44 (1923.4)	4.25 (360.5)	(1551, 29, 29, 81)	4.25 (396.1)	(1551, 29, 29, 81)
R15-F10-C1	2949264 (t)	7.47 (4390.3)	7.18 (632.3)	(2015, 33, 40, 79)	7.18 (668.0)	(2015, 33, 40, 79)
R15-F01-C2	1148604 (t)	0.91 (207.7)	0.91 (90.5)	(550, 13, 9, 38)	0.91 (111.2)	(550, 13, 9, 38)
R15-F05-C2	2488753 (t)	3.75 (2478.0)	3.75 (302.7)	(992, 26, 28, 71)	3.75 (340.2)	(992, 26, 28, 71)
R15-F10-C2	3972667 (t)	7.36 (9459.6)	7.33 (492.8)	(1296, 29, 37, 69)	7.33 (525.2)	(1296, 29, 37, 69)
R15-F01-C8	2301326.6 (t)	1.10 (525.9)	0.97 (97.7)	(272, 17, 23, 20)	0.97 (114.8)	(272, 17, 23, 20)
R15-F05-C8	5573412.8* (16107.9)	0.50 (967.7)	0.48 (154.7)	(439, 30, 44, 20)	0.44 (261.0)	(443, 37, 44, 20)
R15-F10-C8	8696932* (18930.9)	0.21 (1224.4)	0.21 (281.6)	(570, 31, 51, 21)	0.21 (283.3)	(570, 31, 51, 21)

TAB. 14 – Computational Results Simple and Advanced Cutting-Plane-NCI, R15

DESCR 20,300,40	SOL			SCP-NCI (SI,LCI,LMCI,NCI)		ACP-NCI (SI,LCI,LMCI,NCI)	
	SOL	GAP-LP	GAP	GAP	GAP	GAP	GAP
R16-F01-C1	136161* (1.1)	0.00 (0.5)	0.00 (7.1)	(260, 34, 65, 44)	0.00 (8.3)	(260, 34, 65, 44)	
R16-F05-C1	239500* (1641.3)	3.95 (12.8)	3.95 (30.8)	(807, 38, 70, 40)	3.95 (45.6)	(807, 38, 70, 40)	
R16-F10-C1	325671* (4345.9)	4.50 (22.5)	4.50 (45.0)	(1041, 37, 70, 43)	4.50 (63.2)	(1041, 37, 70, 43)	
R16-F01-C2	138532* (27.6)	0.32 (1.1)	0.32 (9.9)	(244, 31, 51, 28)	0.32 (12.4)	(244, 31, 51, 28)	
R16-F05-C2	241801* (851.9)	2.44 (16.6)	2.37 (28.2)	(642, 40, 61, 42)	2.37 (43.8)	(642, 40, 61, 42)	
R16-F10-C2	337762* (8173.5)	4.47 (37.4)	4.10 (42.2)	(745, 39, 67, 52)	4.10 (60.5)	(745, 39, 67, 52)	
R16-F01-C8	169502 (t)	4.67 (3.9)	4.53 (6.9)	(113, 16, 18, 8)	4.46 (45.9)	(113, 20, 22, 8)	
R16-F05-C8	352976 (t)	7.37 (63.3)	6.95 (23.9)	(244, 29, 37, 43)	6.95 (38.8)	(244, 29, 37, 43)	
R16-F10-C8	541626 (t)	8.92 (115.8)	8.72 (35.6)	(323, 30, 44, 65)	8.72 (57.1)	(323, 30, 44, 65)	

TAB. 15 – Computational Results Simple and Advanced Cutting-Plane-NCI, R16

DESCR 20,300,100	SOL			SCP-NCI (SI,LCI,LMCI,NCI)		ACP-NCI (SI,LCI,LMCI,NCI)	
	SOL	GAP-LP	GAP	GAP	GAP	GAP	GAP
R17-F01-C1	354138* (332.1)	0.47 (17.7)	0.46 (37.9)	(685, 37, 65, 16)	0.46 (60.5)	(685, 37, 65, 16)	
R17-F05-C1	651025 (t)	5.59 (182.5)	5.24 (163.1)	(1565, 41, 74, 14)	5.24 (191.2)	(1565, 41, 74, 14)	
R17-F10-C1	917956 (t)	8.28 (432.9)	7.78 (230.0)	(1947, 41, 78, 21)	7.78 (264.0)	(1947, 41, 78, 21)	
R17-F01-C2	370590* (2258.8)	0.79 (38.7)	0.78 (37.9)	(583, 29, 36, 9)	0.78 (59.9)	(583, 29, 36, 9)	
R17-F05-C2	708698 (t)	4.24 (331.0)	4.21 (135.8)	(1089, 40, 51, 51)	4.21 (171.5)	(1089, 40, 51, 51)	
R17-F10-C2	1029242 (t)	7.46 (841.1)	7.41 (216.1)	(1235, 37, 52, 100)	7.41 (258.8)	(1235, 37, 52, 100)	
R17-F01-C8	505310 (t)	2.94 (141.4)	2.93 (28.9)	(243, 10, 9, 12)	2.93 (61.9)	(243, 10, 9, 12)	
R17-F05-C8	1112076 (t)	4.13 (634.5)	4.09 (133.2)	(394, 22, 29, 64)	4.09 (171.5)	(394, 22, 29, 64)	
R17-F10-C8	1847334 (t)	7.92 (1551.1)	7.90 (152.0)	(489, 30, 34, 65)	7.90 (191.0)	(489, 30, 34, 65)	

TAB. 16 – Computational Results Simple and Advanced Cutting-Plane-NCI, R17

DESCR 20,300,200	SOL	GAP-LP	GAP	SCP-NCI (SI,LCI,LMCI,NCI)	GAP	ACP-NCI (SI,LCI,LMCI,NCI)
R18-F01-C1	828559 (t)	1.63 (241.6)	1.63 (161.5)	(1208, 22, 22, 7)	1.63 (236.8)	(1208, 22, 22, 7)
R18-F05-C1	1542197 (t)	3.38 (7144.3)	3.38 (846.8)	(2546, 37, 50, 89)	3.38 (999.1)	(2546, 37, 50, 89)
R18-F10-C1	2292684 (t)	10.07 (17349.7)	10.03 (1268.4)	(2887, 41, 56, 107)	10.03 (1298.3)	(2887, 41, 56, 107)
R18-F01-C2	921762 (t)	2.02 (423.7)	2.01 (164.9)	(894, 12, 11, 14)	2.01 (253.5)	(894, 12, 11, 14)
R18-F05-C2	1866215 (t)	5.92 (9144.4)	5.68 (691.6)	(1506, 19, 17, 116)	5.68 (773.9)	(1506, 29, 17, 116)
R18-F10-C2	2895982 (t)	11.10 (13964.3)	10.90 (861.7)	(1578, 23, 17, 101)	10.90 (907.5)	(1578, 23, 17, 101)
R18-F01-C8	1485690.8 (t)	2.31 (1408.0)	2.30 (197.4)	(343, 9, 9, 45)	2.30 (269.7)	(343, 9, 9, 45)
R18-F05-C8	4010736.7 (t)	6.86 (4536.6)	6.84 (483.8)	(499, 24, 34, 23)	6.84 (463.0)	(499, 24, 24, 23)
R18-F10-C8	6663506.4 (t)	8.63 (9860.1)	8.50 (392.2)	(584, 29, 46, 24)	8.50 (485.0)	(584, 29, 46, 24)

TAB. 17 – Computational Results Simple and Advanced Cutting-Plane-NCI, R18

DESCR N , A , K				SCP-GNCI	ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)
25-100-10	14712*	0.69	0.69	(33, 14, 16, 16)	0.67	(33, 15, 17, 16)
V-L	(0.36)	(0.03)	(0.1)		(1.5)	
25-100-10	14941*	12.87	9.75	(123, 17, 29, 65)	9.21	(126, 20, 32, 70)
F-L	(53.6)	(0.5)	(1.2)		(4.2)	
25-100-10	49899*	9.76	8.21	(60, 12, 12, 51)	7.64	(60, 23, 412, 51)
F-T	(40.6)	(0.3)	(0.6)		(5.4)	
25-100-30	365272*	0.24	0.21	(75, 20, 20, 26)	0.21	(75, 20, 20, 26)
V-T	(16.6)	(0.6)	(0.5)		(2.9)	
25-100-30	37055*	8.26	7.05	(170, 27, 41, 45)	6.40	(185, 30, 44, 54)
F-L	(1727.9)	(4.7)	(3.4)		(9.7)	
25-100-30	85530*	3.63	3.56	(105, 20, 20, 33)	3.56	(105, 20, 20, 33)
F-T	(534.2)	(1.6)	(1.2)		(3.3)	
20-230-40	423848*	0.23	0.20	(114, 27, 47, 24)	0.20	(114, 28, 48, 24)
V-L	(10.4)	(0.3)	(1.1)		(13.8)	
20-230-40	371475*	0.71	0.68	(81, 13, 12, 11)	0.68	(81, 13, 12, 22)
V-T	(52.4)	(0.6)	(0.7)		(10.9)	
20-230-40	643036*	1.48	1.41	(92, 20, 17, 17)	1.35	(92, 21, 18, 31)
F-T	(671.7)	(0.6)	(1.0)		(15.2)	
20-300-40	429398*	0.33	0.33	(98, 29, 48, 20)	0.33	(98, 29, 48, 20)
V-L	(3.76)	(0.3)	(1.2)		(11.1)	
20-300-40	586077*	1.84	1.82	(146, 21, 30, 24)	1.82	(146, 21, 30, 24)
F-L	(145.5)	(1.0)	(1.9)		(12.4)	
20-300-40	464509*	0.77	0.77	(71, 14, 19, 6)	0.77	(71, 15, 20, 6)
V-T	(83.8)	(0.7)	(1.1)		(17.2)	
20-300-40	604198*	1.21	1.14	(95, 22, 25, 5)	1.14	(95, 22, 23, 5)
F-T	(59.8)	(0.6)	(1.1)		(11.1)	
20-230-200	94386	3.26	3.17	(1333, 20, 21, 48)	3.17	(1333, 20, 21, 51)
V-L	(t)	(1083.5)	(244.9)		(291.9)	
20-230-200	141737.4	6.84	6.66	(1497, 17, 23, 47)	6.66	(1497, 17, 23, 49)
F-L	(t)	(2235.1)	(353.8)		(411.2)	
20-230-200	97914	2.29	2.29	(1101, 22, 25, 38)	2.29	(1101, 22, 25, 38)
V-T	(t)	(594.3)	(156.7)		(210.7)	
20-230-200	137271	4.17	4.16	(1286, 22, 18, 56)	4.16	(1286, 22, 18, 56)
F-T	(t)	(2431.1)	(306.8)		(348.4)	
20-300-200	74972.4	2.46	2.46	(929, 9, 10, 15)	2.46	(944, 9, 10, 15)
V-L	(t)	(1332.1)	(200.1)		(259.6)	
20-300-200	117306	5.43	5.39	(1114, 13, 15, 34)	5.39	(1114, 13, 15, 35)
F-L	(t)	(3290.3)	(336.0)		(412.51)	
20-300-200	74991	1.31	1.26	(748, 11, 11, 17)	1.26	(748, 11, 11, 17)
V-T	(t)	(672.3)	(117.3)		(198.5)	
20-300-200	108252	4.26	4.12	(841, 14, 13, 41)	4.12	(841, 14, 13, 41)
F-T	(t)	(2605.1)	(282.2)		(343.6)	

TAB. 18 – Computational Results Simple and Advanced Cutting-Plane-GNCI, C problems

DESCR N , A , K	SCP-GNCI			ACP-GNCI		
	SOL	GAP-LP	GAP	(SI, LCI, LMCI, GNCI)	GAP	(SI, LCI, LMCI, GNCI)
100-400-10	28423*	3.37	2.60	(49, 15, 17, 53)	2.32	(50, 57, 46, 72)
V-L	(84.8)	(0.3)	(0.7)		(27.1)	
100-400-10	24436	17.82	14.32	(357, 17, 24, 229)	11.40	(410, 42, 44, 957)
F-L	(t)	(12.8)	(30.0)		(398.8)	
100-400-10	66364	23.02	18.95	(187, 20, 18, 325)	17.05	(202, 35, 28, 1236)
F-T	(t)	(13.0)	(20.7)		(343.4)	
100-400-30	385544	1.21	1.05	(145, 35, 39, 121)	1.02	(145, 37, 41, 511)
V-T	(t)	(6.1)	(2.8)		(16.58)	
100-400-30	50496	10.11	8.46	(881, 50, 76, 276)	7.59	(936, 56, 81, 433)
F-L	(t)	(116.3)	(157.8)		(305.8)	
100-400-30	141278	15.44	13.68	(273, 45, 49, 230)	13.33	(297, 54, 55, 742)
F-T	(t)	(110.2)	(40.6)		(279.6)	
30-520-100	53966	1.74	1.61	(473, 27, 27, 24)	1.61	(473, 27, 27, 24)
V-L	(t)	(65.9)	(34.2)		(103.9)	
30-520-100	95294	5.37	5.28	(929, 40, 47, 85)	5.28	(929, 40, 47, 101)
F-L	(t)	(1106.5)	(248.3)		(326.4)	
30-520-100	52085	1.45	1.34	(346, 26, 29, 22)	1.34	(346, 26, 29, 22)
V-T	(t)	(59.9)	(23.2)		(109.1)	
30-520-100	98357	4.41	4.24	(672, 32, 42, 42)	4.24	(672, 33, 43, 44)
F-T	(t)	(863.1)	(157.9)		(356.2)	
30-700-100	47603*	0.61	0.61	(419, 30, 42, 20)	0.61	(425, 31, 43, 21)
V-L	(1736.1)	(57.5)	(42.2)		(282.9)	
30-700-100	60525	3.83	3.81	(782, 45, 60, 73)	3.81	(782, 45, 60, 79)
F-L	(t)	(758.0)	(179.8)		(319.1)	
30-700-100	45944.5	1.88	1.71	(417, 22, 25, 19)	1.71	(417, 22, 25, 19)
V-T	(t)	(166.3)	(42.9)		(186.7)	
30-700-100	55082	2.58	2.48	(517, 33, 32, 47)	2.48	(517, 33, 32, 48)
F-T	(t)	(356.0)	(114.6)		(238.6)	
30-520-400	112997.5	1.09	1.08	(1426, 8, 7, 39)	1.08	(1426, 8, 7, 39)
V-L	(t)	(5356.8)	(1902.1)		(2259.9)	
30-520-400	159007	7.75	7.71	(1840, 16, 18, 44)	7.71	(1840, 16, 18, 44)
F-L	(33641.2)	(10649.4)	(3026.3)		(3341.8)	
30-520-400	119744	4.74	4.73	(1115, 5, 2, 25)	4.73	(1115, 5, 2, 25)
V-T	(20461.9)	(4910.5)	(1707.8)		(1949.9)	
30-520-400	163352	8.32	8.29	(1618, 10, 6, 39)	8.29	(1618, 10, 6, 42)
F-T	(59820.1)	(11706.8)	(2974.4)		(3374.2)	
30-700-400	103138	6.33	6.33	(1866, 18, 18, 29)	6.33	(1866, 18, 18, 29)
V-L	(17881.3)	(19428.7)	(2772.8)		(3167.9)	
30-700-400	144615	X	9.60	(2042, 15, 8, 39)	9.60	(2042, 15, 8, 39)
F-L	(57574.8)	(t)	(4747.0)		(5314.1)	
30-700-400	99606	5.61	5.61	(1359, 8, 7, 19)	5.61	(1359, 8, 7, 19)
V-T	(31405.7)	(10923.9)	(2418.0)		(2787.7)	
30-700-400	138777	8.07	8.07	(1539, 7, 4, 28)	8.07	(1539, 7, 4, 28)
F-T	(48693.7)	(34057.0)	(4113.1)		(4534.6)	

TAB. 19 – Computational Results Simple and Advanced Cutting-Plane-GNCI, C problems

DESCR 20,100,40	SOL		SCP-GNCI			ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)	
R10-F01-C1	200087* (10.7)	0.58 (0.2)	0.24 (0.5)	(92, 25, 16, 23)	0.24 (3.7)	(92, 25, 39, 14)	
R10-F05-C1	346813.5* (788.2)	3.84 (2.1)	3.34 (3.5)	(218, 30, 48, 46)	3.34 (6.1)	(218, 30, 48, 46)	
R10-F10-C1	488015* (563.3)	4.27 (4.9)	3.33 (5.2)	(303, 30, 53, 60)	3.33 (8.7)	(303, 30, 63, 60)	
R10-F01-C2	229196* (184.3)	2.32 (0.9)	2.09 (0.9)	(97, 13, 16, 18)	2.09 (4.0)	(97, 13, 16, 18)	
R10-F05-C2	411664* (5123.2)	6.18 (3.4)	5.34 (3.3)	(174, 26, 38, 52)	5.34 (7.7)	(174, 26, 38, 53)	
R10-F10-C2	609104* (27260.5)	7.18 (7.3)	6.41 (6.0)	(219, 27, 45, 59)	6.41 (10.2)	(219, 27, 45, 59)	
R10-F01-C8	486895* (391.3)	4.78 (5.8)	3.67 (2.1)	(79, 13, 15, 26)	3.43 (11.5)	(82, 14, 15, 26)	
R10-F05-C8	951056* (555.5)	7.98 (7.7)	6.19 (3.4)	(117, 16, 22, 26)	5.51 (13.3)	(119, 19, 23, 26)	
R10-F10-C8	1421740* (608.4)	8.90 (9.4)	6.30 (4.2)	(140, 17, 24, 31)	5.34 (13.5)	(140, 26, 25, 31)	

TAB. 20 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R10

DESCR 20,100,100	SOL		SCP-GNCI			ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)	
R11-F01-C1	714431* (392.5)	0.51 (8.0)	0.45 (5.9)	(272, 19, 23, 36)	0.45 (14.0)	(272, 19, 22, 36)	
R11-F05-C1	1265985 (t)	3.53 (52.4)	2.77 (19.9)	(549, 26, 42, 74)	2.77 (28.4)	(549, 26, 42, 74)	
R11-F10-C1	1846295 (t)	5.50 (130.5)	4.81 (33.9)	(704, 29, 39, 58)	4.81 (43.4)	(704, 29, 39, 58)	
R11-F01-C2	870451* (2960.1)	1.12 (19.8)	1.09 (6.7)	(218, 13, 17, 27)	1.09 (15.2)	(218, 13, 17, 27)	
R11-F05-C2	1623640* (21739.5)	3.42 (85.7)	2.54 (21.0)	(357, 23, 30, 44)	2.25 (56.3)	(362, 26, 30, 48)	
R11-F10-C2	2414060 (t)	4.68 (104.9)	3.30 (26.9)	(429, 25, 38, 59)	3.30 (37.9)	(429, 25, 38, 61)	
R11-F01-C8	2294912* (31.5)	0.26 (15.8)	0.21 (12.8)	(368, 25, 25, 24)	0.21 (19.6)	(368, 25, 25, 24)	
R11-F05-C8	3507100* (79.8)	0.56 (27.6)	0.27 (18.9)	(427, 27, 30, 25)	0.27 (25.6)	(427, 27, 30, 25)	
R11-F10-C8	4579353* (48.5)	0.27 (21.8)	0.23 (18.6)	(469, 30, 36, 11)	0.23 (25.7)	(469, 30, 36, 11)	

TAB. 21 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R11

DESCR 20,100,200	SOL		GAP-LP		SCP-GNCI (SI,LCI,LMCI,GNCI)		ACP-GNCI (SI,LCI,LMCI,GNCI)	
	SOL	GAP-LP	GAP	GAP	GAP	GAP	GAP	GAP
R12-F01-C1	1639443* (4436.7)	0.83 (54.3)	0.77 (24.6)	0.77 (24.6)	(375, 11, 13, 29)	0.77 (39.0)	0.77 (39.0)	(375, 11, 13, 29)
R12-F05-C1	3403074.2 (t)	3.99 (400.8)	3.81 (71.5)	3.81 (71.5)	(718, 22, 28, 36)	3.81 (93.5)	3.81 (93.5)	(718, 22, 28, 36)
R12-F10-C1	5276171 (t)	6.47 (969.9)	6.05 (85.7)	6.05 (85.7)	(826, 25, 41, 33)	6.05 (107.6)	6.05 (107.6)	(826, 25, 41, 33)
R12-F01-C2	2303557* (1739.8)	0.61 (122.9)	0.53 (30.9)	0.53 (30.9)	(327, 16, 18, 28)	0.53 (53.4)	0.53 (53.4)	(327, 16, 18, 28)
R12-F05-C2	4669799* (3145.6)	0.82 (243.0)	0.57 (39.8)	0.57 (39.8)	(408, 24, 30, 23)	0.55 (84.0)	0.55 (84.0)	(409, 26, 30, 23)
R12-F10-C2	7100019* (1683.8)	0.69 (624.2)	0.24 (52.0)	0.24 (52.0)	(525, 27, 42, 23)	0.23 (85.5)	0.23 (85.5)	(525, 28, 42, 23)
R12-F01-C8	7635270* (78.1)	0.02 (68.1)	0.02 (48.4)	0.02 (48.4)	(1221, 40, 38, 0)	0.02 (69.5)	0.02 (69.5)	(1221, 40, 38, 0)
R12-F05-C8	10067742* (61.5)	0.01 (54.0)	0.01 (56.1)	0.01 (56.1)	(1352, 40, 36, 0)	0.01 (72.4)	0.01 (72.4)	(1352, 40, 36, 0)
R12-F10-C8	11967768* (50.3)	0.00 (48.9)	0.00 (48.8)	0.00 (48.8)	(1519, 40, 40, 0)	0.00 (64.6)	0.00 (64.6)	(1559, 40, 40, 0)

TAB. 22 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R12

DESCR 20,200,40	SOL		GAP-LP		SCP-GNCI (SI,LCI,LMCI,GNCI)		ACP-GNCI (SI,LCI,LMCI,GNCI)	
	SOL	GAP-LP	GAP	GAP	GAP	GAP	GAP	GAP
R13-F01-C1	142947* (6.4)	0.31 (0.5)	0.31 (1.9)	0.31 (1.9)	(176, 33, 60, 35)	0.31 (7.3)	0.31 (7.3)	(176, 33, 60, 35)
R13-F05-C1	263800* (547.9)	4.82 (7.4)	4.73 (12.7)	4.73 (12.7)	(532, 32, 60, 51)	4.73 (21.4)	4.73 (21.4)	(532, 32, 60, 51)
R13-F10-C1	365800* (116.6)	5.63 (13.7)	5.62 (27.2)	5.62 (27.2)	(730, 36, 68, 68)	5.62 (35.2)	5.62 (35.2)	(730, 36, 68, 68)
R13-F01-C2	150977* (90.1)	1.76 (0.7)	1.55 (2.2)	1.55 (2.2)	(174, 25, 38, 24)	1.55 (13.8)	1.55 (13.8)	(174, 25, 38, 24)
R13-F05-C2	282682* (3551.8)	4.87 (11.1)	4.58 (11.2)	4.58 (11.2)	(350, 33, 50, 58)	4.58 (20.6)	4.58 (20.6)	(350, 33, 50, 59)
R13-F10-C2	406790* (29695.6)	6.81 (21.2)	6.45 (18.8)	6.45 (18.8)	(456, 31, 52, 64)	6.33 (41.1)	6.33 (41.1)	(464, 32, 53, 76)
R13-F01-C8	208088 (t)	3.53 (7.2)	3.18 (2.1)	3.18 (2.1)	(64, 13, 13, 25)	3.18 (11.1)	3.18 (11.1)	(64, 13, 13, 26)
R13-F05-C8	446997 (t)	5.47 (31.3)	4.88 (9.2)	4.88 (9.2)	(129, 26, 34, 51)	4.88 (21.5)	4.88 (21.5)	(129, 26, 34, 53)
R13-F10-C8	698280 (t)	5.31 (54.6)	4.60 (14.3)	4.60 (14.3)	(186, 30, 34, 88)	4.60 (26.3)	4.60 (26.3)	(186, 30, 34, 90)

TAB. 23 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R13

DESCR 20,200,100	SOL		SCP-GNCI		ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)
R14-F01-C1	403414* (495.7)	0.53 (12.9)	0.53 (11.1)	(439, 26, 43, 34)	0.53 (28.5)	(439, 26, 43, 34)
R14-F05-C1	750091 (t)	5.04 (123.4)	4.99 (73.5)	(990, 34, 57, 50)	4.99 (97.2)	(990, 34, 57, 50)
R14-F10-C1	1078595 (t)	7.30 (319.2)	7.28 (174.8)	(1304, 37, 60, 112)	7.28 (208.4)	(1309, 37, 60, 141)
R14-F01-C2	437607* (1327.3)	0.91 (18.3)	0.91 (10.2)	(326, 19, 22, 19)	0.91 (29.4)	(326, 19, 22, 19)
R14-F05-C2	852330 (t)	4.35 (223.1)	4.29 (62.1)	(628, 26, 30, 47)	4.29 (85.3)	(628, 26, 30, 47)
R14-F10-C2	1230749.7 (t)	5.19 (515.8)	4.94 (80.8)	(772, 34, 38, 91)	4.94 (103.5)	(772, 34, 38, 92)
R14-F01-C8	670064 (t)	2.50 (121.4)	2.46 (16.6)	(143, 13, 13, 43)	2.46 (37.6)	(143, 13, 13, 43)
R14-F05-C8	1643376 (t)	6.09 (321.1)	6.05 (52.1)	(281, 24, 36, 52)	6.05 (78.8)	(281, 24, 36, 54)
R14-F10-C8	2630332 (t)	6.38 (591.3)	6.12 (54.5)	(351, 26, 49, 49)	6.12 (140.2)	(351, 26, 49, 242)

TAB. 24 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R14

DESCR 20,200,200	SOL		SCP-GNCI		ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)
R15-F01-C1	1000787* (5384.7)	0.66 (95.5)	0.66 (70.2)	(713, 18, 16, 32)	0.66 (105.4)	(713, 18, 16, 32)
R15-F05-C1	1979413 (t)	4.44 (1923.4)	4.25 (411.8)	(1576, 31, 29, 60)	4.25 (453.9)	(1576, 31, 29, 60)
R15-F10-C1	2949264 (t)	7.47 (4390.3)	7.18 (600.1)	(2028, 33, 40, 101)	7.18 (670.8)	(2028, 33, 40, 102)
R15-F01-C2	1148604 (t)	0.91 (207.7)	0.90 (70.1)	(555, 14, 10, 42)	0.90 (105.6)	(555, 14, 10, 42)
R15-F05-C2	2488753 (t)	3.75 (2478.0)	3.69 (278.7)	(1022, 28, 28, 96)	3.69 (320.5)	(1022, 28, 28, 97)
R15-F10-C2	3972667 (t)	7.36 (9459.6)	7.25 (447.8)	(1319, 29, 37, 116)	7.25 (504.4)	(1329, 29, 37, 134)
R15-F01-C8	2301326.6 (t)	1.10 (525.9)	0.97 (73.8)	(272, 17, 23, 25)	0.96 (160.4)	(272, 17, 23, 27)
R15-F05-C8	5573412.8* (16107.9)	0.50 (967.7)	0.48 (142.2)	(436, 30, 44, 26)	0.44 (344.2)	(440, 37, 44, 26)
R15-F10-C8	8696932* (18930.9)	0.21 (1224.4)	0.21 (238.7)	(570, 31, 51, 28)	0.21 (268.1)	(570, 31, 51, 28)

TAB. 25 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R15

DESCR 20,300,40	SOL			SCP-GNCI		ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)	
R16-F01-C1	136161* (1.1)	0.00 (0.5)	0.00 (4.3)	(260, 35, 65, 57)	0.00 (8.4)	(260, 35, 65, 57)	
R16-F05-C1	239500* (1641.3)	3.95 (12.8)	3.92 (35.0)	(803, 37, 70, 105)	3.92 (61.3)	(803, 37, 70, 177)	
R16-F10-C1	325671* (4345.9)	4.50 (22.5)	4.47 (70.4)	(1050, 36, 69, 143)	4.47 (95.9)	(1050, 36, 69, 190)	
R16-F01-C2	138532* (27.6)	0.32 (1.1)	0.32 (4.4)	(245, 31, 51, 35)	0.32 (8.1)	(245, 31, 51, 35)	
R16-F05-C2	241801* (851.9)	2.44 (16.6)	2.30 (32.8)	(627, 40, 63, 96)	2.30 (50.5)	(627, 40, 63, 140)	
R16-F10-C2	337762* (8173.5)	4.47 (37.4)	4.09 (49.1)	(735, 39, 67, 150)	4.09 (135.2)	(735, 39, 67, 303)	
R16-F01-C8	169502 (t)	4.67 (3.9)	4.58 (2.8)	(107, 15, 18, 17)	4.44 (89.6)	(118, 24, 24, 118)	
R16-F05-C8	352976 (t)	7.37 (63.3)	6.88 (21.6)	(238, 27, 38, 61)	6.88 (51.5)	(241, 28, 38, 63)	
R16-F10-C8	541626 (t)	8.92 (115.8)	8.71 (39.8)	(326, 29, 44, 101)	8.71 (84.7)	(326, 29, 44, 231)	

TAB. 26 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R16

DESCR 20,300,100	SOL			SCP-GNCI		ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)	
R17-F01-C1	354138* (332.1)	0.47 (17.7)	0.46 (39.6)	(681, 37, 65, 49)	0.46 (71.4)	(681, 37, 65, 52)	
R17-F05-C1	651025 (t)	5.59 (182.5)	5.24 (227.1)	(1558, 40, 74, 65)	5.24 (3254.0)	(1558, 40, 74, 65)	
R17-F10-C1	917956 (t)	8.28 (432.9)	7.78 (330.5)	(1951, 41, 78, 89)	7.78 (362.2)	(1951, 41, 78, 90)	
R17-F01-C2	370590* (2258.8)	0.79 (38.7)	0.78 (34.6)	(585, 29, 36, 36)	0.78 (65.3)	(585, 29, 36, 36)	
R17-F05-C2	708698 (t)	4.24 (331.0)	4.21 (136.0)	(1064, 39, 49, 58)	4.21 (167.3)	(1064, 39, 49, 58)	
R17-F10-C2	1029242 (t)	7.46 (841.1)	7.41 (201.7)	(1230, 37, 52, 71)	7.41 (239.8)	(1230, 37, 52, 71)	
R17-F01-C8	505310 (t)	2.94 (141.4)	2.93 (21.0)	(244, 10, 9, 18)	2.93 (52.8)	(244, 10, 9, 18)	
R17-F05-C8	1112076 (t)	4.13 (634.5)	4.09 (113.4)	(398, 23, 30, 72)	4.09 (147.7)	(398, 23, 30, 72)	
R17-F10-C8	1847334 (t)	7.92 (1551.1)	7.88 (144.7)	(504, 29, 36, 102)	7.88 (193.5)	(504, 29, 36, 122)	

TAB. 27 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R17

DESCR 20,300,200	SOL		SCP-GNCI		ACP-GNCI	
	SOL	GAP-LP	GAP	(SI,LCI,LMCI,GNCI)	GAP	(SI,LCI,LMCI,GNCI)
R18-F01-C1	828559 (t)	1.63 (241.6)	1.63 (152.4)	(1202, 22, 22, 36)	1.63 (216.3)	(1202, 22, 22, 38)
R18-F05-C1	1542197 (t)	3.38 (7144.3)	3.36 (920.3)	(2575, 37, 50, 82)	3.36 (1075.7)	(2575, 37, 50, 146)
R18-F10-C1	2292684 (t)	10.07 (17349.7)	10.03 (1229.7)	(2884, 41, 55, 105)	10.03 (1318.6)	(2884, 41, 55, 111)
R18-F01-C2	921762 (t)	2.02 (423.7)	2.01 (181.1)	(894, 12, 11, 30)	2.01 (239.2)	(894, 12, 11, 30)
R18-F05-C2	1866215 (t)	5.92 (9144.4)	5.68 (674.0)	(1467, 19, 18, 93)	5.68 (758.4)	(1467, 19, 18, 93)
R18-F10-C2	2895982 (t)	11.10 (13964.3)	10.90 (827.4)	(1579, 25, 18, 113)	10.90 (895.0)	(1579, 25, 18, 116)
R18-F01-C8	1485690.8 (t)	2.31 (1408.0)	2.30 (205.7)	(349, 9, 9, 66)	2.30 (278.4)	(349, 9, 9, 67)
R18-F05-C8	4010736.7 (t)	6.86 (4536.6)	6.81 (363.3)	(508, 27, 34, 42)	6.81 (432.9)	(508, 27, 34, 42)
R18-F10-C8	6663506.4 (t)	8.63 (9860.1)	8.49 (358.1)	(585, 29, 46, 30)	8.49 (462.0)	(585, 29, 46, 30)

TAB. 28 – Computational Results Simple and Advanced Cutting-Plane-GNCI, R18

Chapitre 4

**A Branch-and-Cut algorithm for
multicommodity capacitated fixed
charge network design**

A Branch-and-Cut algorithm for multicommodity capacitated fixed charge network design

Mervat Chouman, Teodor Gabriel Crainic
and Bernard Gendron

Abstract

In this paper, we present a Branch-and-Cut algorithm to solve the multicommodity capacitated fixed charge network design problem. We investigate the use of a cutting-plane procedure which incorporates valid inequalities based on cutsets of the network. We propose a new branching rule based on a combination of two variables and a heuristic approach to compute feasible solutions. Accumulated information through the resolution process is used to guide the search to explore first promising spaces that possibly contain feasible solutions of good quality. We present computational results conducted on a large set of instances with various characteristics. Our results show a promising behavior of the algorithm. For large scale instances with many commodities, our algorithm offers feasible solutions of good quality.

Key words : multicommodity capacitated fixed charge network design, Branch-and-Cut, cutting-plane, valid inequalities.

Résumé

Dans ce papier, nous présentons un algorithme de “Branch-and-Cut” pour résoudre le problème de conception de réseaux multiproduits, avec coût fixe et capacité. Nous étudions l’utilisation d’une méthode de coupes qui intègre des inégalités valides basées sur des coupes du réseau. Nous proposons une nouvelle stratégie de branchement basée sur une combinaison de deux variables ainsi qu’une approche heuristique pour identifier des solutions réalisables. Des informations accumulées tout au long du processus de résolution sont utilisées afin de diriger la recherche à explorer en premier des espaces prometteuses pouvant contenir de solutions de bonne qualité. Nous

présentons des résultats expérimentaux obtenus sur un grand nombre de problèmes ayant différentes caractéristiques. Nos résultats montrent un comportement prometteur de l'algorithme. Pour les instances de grande taille ayant plusieurs commodités, notre algorithme offre des solutions réalisables de bonne qualité.

Mots-clés : problème de conception de réseaux multiproduits avec coût fixe et capacité, Branch-and-Cut, méthode de coupes, inégalités valides.

1 Introduction

In this paper, we present a Branch-and-Cut algorithm (B&C) to solve the multi-commodity capacitated fixed charge network design problem (MCND). This problem is a particular case of a broad class of problems known as network design problems. The MCND arises in a wide variety of problem contexts including transportation, telecommunications, production scheduling, etc (for a survey of network design see Magnanti and Wong 1984, Minoux 1989, Crainic 1999). Briefly, given a capacitated network and a set of traffic demands between origin-destination nodes, the MCND aims to design a network, by specifying the arcs to be used, and to satisfy the demand requirements, by determining the amounts of flows on each arc, at minimum cost. In this problem, in addition to the usual per unit routing cost, a fixed cost is paid as soon as a link is used.

The MCND is modeled usually as a mixed integer programming problem where continuous variables represent the flows on arcs and 0-1 integer variables represent decision concerning the arcs to be used. Considerable algorithmic challenges are imposed when trying to solve this combinatorial problem. In addition to the large size of real applications, these challenges are due to the tradeoff between routing and fixed costs and to the competition among commodities when sharing links with limited capacity.

Several bounding procedures based on classical relaxation approaches, in particular Lagrangian relaxations, have been analysed in Gendron and Crainic (1994, 1996). Based on these studies, exact and heuristic approaches, namely Lagrangian heuristics within Branch-and-Bound framework, have been proposed to solve the MCND (Gendron, Crainic, and Frangioni 1998, Crainic, Frangioni, and Gendron 2002, Holmberg and Yuan 2000, and Sellmann, Kliwer, and Koberstein 2002). Heuristic approaches have also been proposed for computing feasible solutions (Crainic, Gendreau, and Farvolden 2000, Ghamlouche, Crainic, and Gendreau 2001, 2002, Crainic, Gendron, and Hernu 2002, Crainic and Gendreau 2002). To assess the quality of the solutions produced by these heuristics, optimal solutions must be known. However, to this date, even tight lower bounds cannot be computed efficiently in a reasonable amount

of time for large set of instances.

Recently, Chouman, Crainic, and Gendron (2003) proposed to improve the formulation of the problem using polyhedral approaches. In fact, they devised an efficient cutting-plane procedure that incorporates useful valid inequalities (VI) based on cutsets in the network. Authors have identified, studied, and adapted known VI as well as new families of VI. These inequalities are of cutset type inequalities in the sense that, in any feasible solution, the capacity of the arcs of any cutset that cuts off some origins from their destinations must be sufficient to flow the demand across the cutset. They also developed efficient heuristics to identify different cutsets in the network. Their computational results show that the proposed cutting-plane procedure is effective to improve the formulation of the problem and, hence, the quality of bounds, in a reasonable amount of time.

Branch-and-Cut (B&C), introduced by Padberg and Rinaldi (1987, 1991), constitutes a significant development in addressing combinatorial optimization problem. The philosophy of this approach is to strengthen the formulation of the problem using an automatic cut generation not only prior to starting the Branch-and-Bound but also at each node of the enumeration tree. One important point for the success of this approach, from a practical point of view, is that the added cuts at any node must be valid at all other nodes. This point does not constitute, however, a drawback of the approach since it has led to a number of successful papers when treating special or general combinatorial problems (Balas, Ceria, and Cornuéjols 1996, Padberg and Rinaldi 1987, 1991, Günlük 1999) and to a standard MIP softwares such as MPSARX (Van Roy and Wolsey 1987), MINTO (Nemhauser, Savelsbergh, and Sigismondi 1994), Xpress-MP (2002) and CPLEX (2002). However, Chouman, Crainic, and Gendron (2001) have shown that the B&C of CPLEX (2002) cannot solve large size instances of the MCND in a reasonable amount of time.

The goal of this paper is to present a B&C algorithm that incorporates an adaptation of the cutting-plane procedure proposed in Chouman, Crainic, and Gendron (2003) to solve the MCND. The proposed B&C algorithm is motivated essentially

by two facts. First, the cuts used in the cutting-plane are valid at any node in the enumeration tree. This fact allows the exploitation of strengthened formulations not only at a specific node but also at all other nodes. Second, when compared to the state-of-the-art LP software CPLEX (2002), the cutting-plane shows a remarkable improvement in the time needed to compute the lower bounds, especially for large size instances with many commodities. In this context, it is interesting to investigate the usefulness of this cutting-plane within a B&C algorithm.

The contribution of this paper is the development of a B&C algorithm which solves at each node of the enumeration tree strengthened LP formulations of the MCND problem. These formulations are obtained by adding violated VI found using an adaptation of the cutting-plane procedure proposed in Chouman, Crainic, and Gendron (2003). We also propose a new branching rule based on the identification of two branching variables and we develop a heuristic approach to compute feasible solutions. Accumulated information through the B&C algorithm is used in order to choose a good combination of two variables for the branching rule and to guide the search to explore first promising spaces which possibly contain feasible solutions of good quality. Our computational results show that the B&C of CPLEX (2002) is superior to our B&C for small size instances. However, when the problem size is getting larger, especially when the number of commodities is greater than 100, our B&C approximates closely CPLEX, in terms of solution quality, and outperforms it when the number of commodities is large. In fact, for problems with 400 commodities, for which CPLEX was unable to find any feasible solution within 10 hours of computation, our B&C offers good quality feasible solutions which outperform those obtained by one of the best approximated methods available in the literature, the *Path-Relinking* heuristic proposed in Ghamlouche, Crainic, and Gendreau (2002). Although the results show a promising behavior of the algorithm, its performance may be improved by performing several refinement steps which justify additional efforts.

This paper is organized as follows. In the next section, we present a MIP formulation of the MCND and the main steps in a general Branch-and-Cut algorithm. In Section 3, we briefly describe the families of valid inequalities and the cutting-plane

procedure. The new branching rule and the upper bound heuristic to compute feasible solutions are discussed in Sections 4 and 5. Computational results are presented in Section 6. Finally, conclusions and some forthcoming developments are drawn in Section 7.

2 Problem Formulation and Algorithmic Framework

The multicommodity, capacitated, fixed charge network design problem (MCND) consists on designing a network which will be able to satisfy demand requirements at minimum cost. Consider the basic network structure as a graph $G(N, A)$, where N is the set of nodes, A is the set of directed arcs, and let K be the set of commodities. Without loss of generality, we consider that each commodity k has only one origin, $O(k)$, and one destination, $D(k)$. Three characteristics are associated with each arc : the per unit routing cost c_{ij}^k for each commodity, the fixed cost f_{ij} which will be payed as soon as the arc is used, and the global capacity u_{ij} on the total amount of flow that moves on the arc. Two types of variables are used to formulate the MCND problem :

- the continuous flow variable x_{ij}^k which represents the flow of commodity k that moves on arc (i, j) ,
- the binary design variable y_{ij} which will be 1 when (i, j) is used (opened) and 0 otherwise.

Given these definitions, we present the mathematical formulation of the MCND as follows.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij}, \quad (1)$$

$$\sum_{j \in N_i^+} x_{ij}^k - \sum_{j \in N_i^-} x_{ji}^k = \begin{cases} d^k, & \text{if } i = O(k), \\ -d^k, & \text{if } i = D(k), \\ 0, & \text{otherwise,} \end{cases} \quad \forall i \in N, \forall k \in K, \quad (2)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i, j) \in A, \quad (3)$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K, \quad (4)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A, \quad (5)$$

where $N_i^+ = \{j \in N | (i, j) \in A\}$ and $N_i^- = \{j \in N | (j, i) \in A\}$.

Constraints set (2) represent the flow conservation constraint for each node and each commodity. Constraints set (3) represent the forcing constraints which ensure that, first the flow cannot exceed the total capacity of the related arc, and second, no flow is allowed until the fixed cost of the arc is payed.

The LP relaxation of the problem is obtained by dropping the integrality constraints (5) and adding

$$0 \leq y_{ij} \leq 1, \quad \forall (i, j) \in A.$$

By doing so, we obtain thus the so-called weak LP relaxation of the MCND problem.

Over the last twenty years, Branch-and-Cut (B&C) algorithm, introduced originally by Padberg and Rinaldi (1987, 1991) in the context of the traveling salesman problem, constitutes a significant development in addressing integer and combinatorial optimization problems. The basic idea of this approach is to combine Branch-and-Bound and polyhedral approaches in an algorithmic fashion. Thus, the idea is to recursively partition the solution space and to solve strengthened continuous LP relaxation obtained by adding violated valid inequalities (VI) to the formulation.

The general B&C procedure is summarized next. Here, we denote by *active node* a node that is not evaluated yet and by *NodeList* the list of all active nodes of the enumeration tree. Z^* and (x^*, y^*) represent the best objective value and feasible solution found, respectively, while \bar{Z} and (\bar{x}, \bar{y}) represent the optimal value and optimal solution of the LP relaxation at the current node.

Branch-and-Cut Procedure

1. Initialization :

- $NodeList \leftarrow$ root node represented by the original formulation.
- $Z^* \leftarrow \infty, (x^*, y^*) \leftarrow (0, 0)$.

2. *Node selection* :

- If *NodeList* is empty, go to *Stop*.
- Choose and remove the first node i at the top of the *NodeList*; let LP_i be the LP relaxation of the formulation at the current node i .

3. *LP relaxation* :

- Solve the LP_i relaxation; let \bar{Z} and (\bar{x}, \bar{y}) be the optimal value and the associated optimal solution.

4. *Pruning criteria* :

- If LP_i is infeasible, go to *Node selection*.
- If \bar{y} is integer and $\bar{Z} < Z^*$
 - Update $Z^* : Z^* \leftarrow \bar{Z}$.
 - Update $(x^*, y^*) : x^* \leftarrow \bar{x}, y^* \leftarrow \bar{y}$.
 - Go to *Node selection*.
- If $\bar{Z} \geq Z^*$, go to *Node selection*.

5. *Upper bound heuristic* :

- Find a feasible solution; let Z be the optimal value and (x, y) be the corresponding optimal solution.
- If $Z < Z^*$
 - Update $Z^* : Z^* \leftarrow Z$.
 - Update $(x^*, y^*) : x^* \leftarrow x, y^* \leftarrow y$.
 - If $\bar{Z} \geq Z^*$, go to *Node selection*.

6. *Cut generation* :

- Perform the cutting-plane procedure; let Z be the optimal value and (x, y) be the corresponding optimal solution returned from the cutting-plane.
- If $Z \neq \bar{Z}$
 - Update $\bar{Z} : \bar{Z} \leftarrow Z$.
 - Update $(\bar{x}, \bar{y}) : \bar{x} \leftarrow x, \bar{y} \leftarrow y$.

- If \bar{y} is integer or $\bar{Z} \geq Z^*$, go to *Pruning criteria*.

7. *Branching* :

- Create new nodes and add them to *NodeList*.
- Go to *Node selection*.

8. *Stop* : Return Z^* and (x^*, y^*) .

When the algorithm stops, if $Z^* < \infty$ then (x^*, y^*) is an optimal solution of the problem, otherwise the problem is infeasible.

The basic components of the B&C procedure, *Cut generation*, *Branching*, and *Upper bound heuristic* are described in the next sections. Furthermore, we denote by $\bar{A} = \bar{A}_1 \cup \bar{A}_0$ the set of all fixed variables at the current node in the enumeration tree where $\bar{A}_1 = \{(i, j) \in A, y_{ij} = 1\}$ and $\bar{A}_0 = \{(i, j) \in A, y_{ij} = 0\}$.

3 Cut Generation

Here, we describe briefly the families of valid inequalities (VI) and the cutting-plane procedure used in our B&C algorithm. Details are presented in Chouman, Crainic, and Gendron (2003). We note that, in order to enhance the performance of the proposed B&C algorithm, we do not use in our cutting-plane all families of VI studied in this paper. In fact, the generation of some of these inequalities requires much computational effort and they do not lead to a significant improvement in the formulation. Thus, based on a measure of cut quality compared to the computational effort needed for the generation, we decide to incorporate only three families of VI in our cutting-plane. Thus, Section 3.1 describes the *strong inequalities* while Section 3.2 describes families of cutset type inequalities, *lifted cover inequalities* and *network cutset inequalities*. These inequalities are based on cutset of the network in the sense that in any feasible solution, the capacity of the arcs of any cutset that cuts off some origins from their destinations must be sufficient to flow the demand across the cutset.

3.1 Strong Inequalities

Gendron, Crainic, and Frangioni (1998) show that the lower bounds derived from the LP relaxation of the MCND are weak. To improve these bounds, the following VI, called *strong inequalities*, can be added to obtain the *strong LP relaxation* :

$$x_{ij}^k \leq d^k y_{ij}, \quad \forall (i, j) \in A, k \in K.$$

It is well-known that adding these inequalities to the formulation improves significantly the quality of LP lower bounds. Yet, solving the LP relaxation including all these inequalities is time consuming and not all such inequalities are necessary. Hence, in our cutting-plane, we solve the weak LP relaxation and generate and add only SI that are violated by the current LP solution.

3.2 Cutset Type Inequalities

let $S \subset N$ be any non empty subset of N and its complement $\bar{S} = N \setminus S$, and let $K(S, \bar{S}) \subseteq K$ be the set of commodities having their origin in S and their destination in \bar{S} . Aggregating the capacity constraints (3) and the flow conservation equations (2) over all nodes in S gives the well-known *cutset inequality* :

$$\sum_{(i,j) \in (S, \bar{S})} u_{ij} y_{ij} \geq d_{(S, \bar{S})},$$

where $d_{(S, \bar{S})} = \sum_{k \in K(S, \bar{S})} d^k > 0$ and (S, \bar{S}) is the set of arcs that connect a node in S to a node in \bar{S} (a *cutset*). Note that, the cutset inequality is redundant for the MCND and for its LP relaxation. However, some useful families of VI derived from the cutset structure might improve the LP relaxation. These families are studied in detail in Chouman, Crainic, and Gendron (2003). For the sake of completeness, we briefly describe these families in the next two subsections.

Lifted Cover Inequalities

Consider the set P_S induced by the given cutset (S, \bar{S}) ,

$$P_S = \{y \in \{0, 1\}^m : \sum_{(i,j) \in (S, \bar{S})} u_{ij} y_{ij} \geq d_{(S, \bar{S})}\},$$

where $m = |(S, \bar{S})|$. A set $C \subseteq (S, \bar{S})$ is called a minimal cover if

- the total capacity of arcs in $(S, \bar{S}) \setminus C$ does not cover the demand : $\sum_{(i,j) \in (S, \bar{S}) \setminus C} u_{ij} < d_{(S, \bar{S})}$, and,
- it is sufficient to open any arc in C to cover the demand : $\sum_{(i,j) \in (S, \bar{S}) \setminus C} u_{ij} + u_{pq} \geq d_{(S, \bar{S})}, \forall (p, q) \in C$.

For any cover C , the *cover inequality* (CI), developed independently by Balas (1975) and Wolsey (1975), defined as

$$\sum_{(i,j) \in C} y_{ij} \geq 1$$

is valid for P_S , for any subset S . The cover inequality is valid for the MCND, but it is not “strong”, in general. Lifting procedure can be applied to get a stronger one, however.

Consider two disjoint subset of arcs C_1 (the open arcs) and C_0 (the closed arcs) in (S, \bar{S}) that satisfy the condition

$$\sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} u_{ij} \geq d_{(S, \bar{S})} - \sum_{(i,j) \in C_1} u_{ij} > 0.$$

We then define the restricted set $P_{S_{(C_1, C_0)}}$ induced by C_1 and C_0 as

$$P_{S_{(C_1, C_0)}} = \{y \in \{0, 1\}^m : \sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} u_{ij} y_{ij} \geq d_{(S, \bar{S})} - \sum_{(i,j) \in C_1} u_{ij}\}.$$

We call *restricted CI* a cover inequality derived from this restricted set, with its corresponding cover denoted C . Since this inequality is restricted to open arcs in C_1 and closed arcs in C_0 , lifting (down for the variables in C_1 and up for the variables in C_0) is necessary to ensure global validity. In addition, by lifting down the variables in $(S, \bar{S}) \setminus (C_1 \cup C_0 \cup C)$, we obtain the *lifted cover inequality* (LCI) :

$$\sum_{(i,j) \in (S, \bar{S}) \setminus C} \gamma_{ij} y_{ij} + \sum_{(i,j) \in C} y_{ij} \geq 1 + \sum_{(i,j) \in (S, \bar{S}) \setminus (C \cup C_0)} \gamma_{ij}.$$

The choice of the sets C_1 and C_0 is of crucial importance for the performance of this lifting strategy. Implementation issues for determining C_1 and C_0 and computing lifting coefficients γ_{ij} are discussed in Chouman, Crainic, and Gendron (2003). Here we give the general idea of how to separate a violated LCI.

Separation : Given a fractional optimal solution \bar{y} for some LP relaxation, the separation problem for a given family of VI consists in finding an inequality that is violated by the current solution \bar{y} or showing that such inequality does not exist. The separation problem for the LCI is solved within three steps.

- First, determine, *a priori*, two subsets of arcs $C_1 \subset (S, \bar{S})$, $C_0 \subset (S, \bar{S})$, and $C_1 \cap C_0 = \emptyset$.
- Second, generate the most violated restricted cover C by solving the following 0-1 knapsack problem :

$$LCI_{sep} \quad \min \quad \begin{aligned} & \sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} Z_{ij} \bar{y}_{ij}, \\ & \sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} Z_{ij} u_{ij} > \sum_{(i,j) \in (S, \bar{S}) \setminus (C_1 \cup C_0)} u_{ij} - d_{(S, \bar{S})} \\ & \quad \quad \quad + \sum_{(i,j) \in C_1} u_{ij}, \\ & Z_{ij} \in \{0, 1\}, \forall (i, j) \in (S, \bar{S}) \setminus (C_1 \cup C_0). \end{aligned}$$

- Third, lift down all variables in $(S, \bar{S}) \setminus (C \cup C_0)$ and then lift up all variables in C_0 .

In our implementation, we use an adaptation of the classical greedy heuristic to solve the 0-1 knapsack problem LCI_{sep} , while we use an exact dynamic programming algorithm to solve the 0-1 knapsack problems emerged when computing lifting coefficients. Also, the quality of the resulting inequality depends on the lifting order and lift down the variables in $(S, \bar{S}) \setminus (C \cup C_0)$ contribute to the violation. So, we lift down first those variables, in non-decreasing order of their current values, and then we lift up variables in C_0 , in non-increasing order of their current values.

Network Cutset Inequalities

Let assume that $x_{ij} = \sum_{k \in K} x_{ij}^k$, and let $(i, j) \in (\bar{S}, S)$, $C_1 \subseteq (\bar{S}, S) \setminus \{(i, j)\}$ and $C_2 \subseteq (S, \bar{S})$.

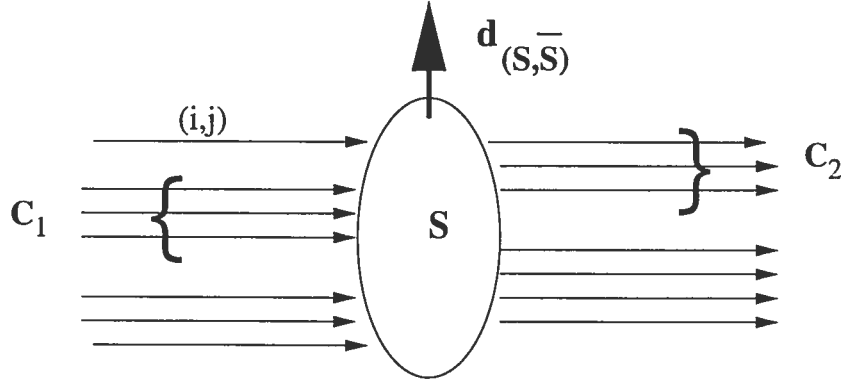


FIG. 1 – Network cutset structure

Considering these assumptions, any feasible solution for the MCND problem must satisfy the *network cutset inequality (NCI)*, defined as

$$\sum_{(r,t) \in C_1} x_{rt} + x_{ij} \leq \left(\sum_{(r,t) \in C_2} u_{rt} + d_{(S,\bar{S})} y_{ij} \right) + \sum_{(r,t) \in (S,\bar{S}) \setminus C_2} x_{rt} + (1 - y_{ij}) \sum_{(r,t) \in C_1} u_{rt}. \quad (6)$$

The NCI is a VI for the MCND problem. In fact, if $y_{ij} = 0$, then this inequality is implied by an aggregation of the capacity constraints over all arcs in C_1 . However, if $y_{ij} = 1$, then the inequality is implied by a combination and an aggregation of capacity constraints and flow conservation constraints over all arcs in C_1 and (S, \bar{S}) . In the following, we show how to separate a violated NCI.

Separation : Given a set $S \subset N$ and an arc $(i, j) \in (\bar{S}, S)$, the separation problem for the NCI consists in finding two sets $C_1 \subseteq (\bar{S}, S)$ and $C_2 \subseteq (S, \bar{S})$ such that the violation with respect to the current LP solution, (\bar{x}, \bar{y}) , is maximal. It is clear that taking

$$C_1 = \{(r, t) \in (\bar{S}, S) \setminus \{(i, j)\} : \bar{x}_{rt} > (1 - \bar{y}_{ij}) u_{rt}\},$$

$$C_2 = \{(r, t) \in (S, \bar{S}) : u_{rt} \bar{y}_{ij} < \bar{x}_{rt}\},$$

gives the optimal choice. Chouman, Crainic, and Gendron (2003) have implemented and tested this inequality on a large set of instances. The results show the usefulness of this inequality, combined with the LCI, to improve the lower bounds of the MCND in a reasonable computational time.

3.3 Cutting-Plane Algorithm

Given that each cutset (S, \bar{S}) might allow to generate LCI and NCI, we propose to identify and maintain simultaneously multiple cutsets, called a *family*. Such a family of cutsets is obtained by partitioning the set of nodes N into L subsets $S_l, l = 1, \dots, L$, such that $S_l \cap S_k = \emptyset$, for all $l \neq k$, and $\cup_{l=1, \dots, L} S_l = N$. Then, each cutset (S_l, \bar{S}_l) induced by subset $S_l, l = 1, \dots, L$, determines a family of cutsets $(S_l, \bar{S}_l)_{l=1, \dots, L}$ available for the generation of violated VI.

The cutting-plane algorithm is divided into two improvement phases. In the first phase, the cutting-plane procedure performs the generation of cuts on the initial family of cutsets obtained by partitioning the set of nodes N into $|N|$ subsets, each containing a single node (i.e., $S_l = \{i\}, \forall i \in N, l = 1, \dots, |N|$). The generation of cuts involves the addition of SI, LCI, and NCI, if any, to the formulation, and the resolution of the resulting LP relaxation. The first phase of the algorithm iterates on the initial family of cutsets until no further significant improvement is observed.

In the second phase of the algorithm, new families of cutsets are identified by partitioning N into L subsets S_l each with a maximum cardinality of M , where M varies from 2 to a predetermined number M_{max} . In each partitioning, we attempt to construct the sets S_l with the objective of minimizing $\sum_{(i,j) \in (S_l, \bar{S}_l)} \bar{y}_{ij}$ in order to increment the chance to find violated cuts. For each family of cutsets, the algorithm performs the generation of cuts and identifies, for a given number of iterations I_{max} , a new family of cutsets by performing exchanges of nodes among subsets of the current family. Each identified family is then used for the generation of cuts. If the solution obtained at the end of the second phase is improved upon that of the first phase, then the first phase is launched all over again. Thus the proposed cutting-plane iterates over the first and second phases until no further improvement is obtained. In addition, to limit the computational effort, the second phase is invoked no more than T_{max} times.

The cutting-plane procedure is summarized next. Here Z and (x, y) represent the current LP optimal value at the last iteration during the first phase, and during the second phase, it is the LP optimal value at the end of the first phase. They are ini-

tialized in the *Initialization step* to \bar{Z} and (\bar{x}, \bar{y}) . Z is used to verify the stopping condition $Z - Z_{last} \leq \epsilon$, used in any of the two phases (ϵ is a parameter). Z^* and (x^*, y^*) represent the best objective value and feasible solution found in the B&C algorithm. It is used to stop the cutting-plane procedure when the LP optimal value is greater than Z^* .

Cutting-plane procedure

1. *Initialization step* :

- $T \leftarrow 0, Z_{last} \leftarrow 0$.
- $Z \leftarrow \bar{Z}, (x, y) \leftarrow (\bar{x}, \bar{y})$.

2. *First phase* :

- Initialize the family of cutsets by partitioning N into $|N|$ subsets (i.e., $S_l = \{i\}, \forall i \in N, l = 1, \dots, |N| = L$).
- While $Z - Z_{last} > \epsilon$
 - Add cuts; if no cuts were found, go to 3.
 - $Z_{last} \leftarrow Z$.
 - Solve the LP relaxation; let Z be the optimal value and (x, y) be the corresponding optimal solution.
 - If (x, y) is feasible or $Z \geq Z^*$, go to 4.

3. *Second phase* :

- If $T \geq T_{max}$, go to 4.
- $T \leftarrow T + 1$ and $Z_{last} \leftarrow Z$.
- For $M = 2$ to M_{max}
 - Identify a family of cutsets by partitioning the set of nodes N into L subsets each with maximal cardinality of M .
 - Add cuts.
 - If at least one cut has been added
 - Solve the LP relaxation; let Z be the optimal value and (x, y) be the corresponding optimal solution.

- If (x, y) is feasible or $Z \geq Z^*$, go to 4.
 - For $I = 1$ to I_{max}
 - Identify a family of cutsets by performing exchanges of nodes among subsets of the current family.
 - Add cuts.
 - If at least one cut has been added
 - Solve the LP relaxation; let Z be the optimal value and (x, y) be the corresponding optimal solution.
 - If (x, y) is feasible or $Z \geq Z^*$, go to 4.
 - If $Z - Z_{last} > \epsilon$, go to 2.
4. Return Z and (x, y) .

Note that, the identification of new families of cutsets, either by identifying a new partitioning of nodes or by performing exchanges of nodes among subsets of the current family, is described in details in Chouman, Crainic, and Gendron (2003).

Obviously, using the overall cutting-plane algorithm, composed from the two phases, is computationally more expensive than using only the first phase. Moreover, based on an extensive study of this cutting-plane algorithm (Chouman, Crainic, and Gendron 2003), the authors observed that the behavior of the first phase, when compared to the overall algorithm, is very satisfactory in the sense that it improves the formulation of the problem in a reasonable amount of time. Thus we propose to use the overall algorithm at the root node of the enumeration tree, prior to starting B&C, and then to use only the first phase of the algorithm at each subsequent node. By doing so, we improve as much as possible the original formulation of the problem, possibly by allowing more effort, while we try to reach a good balance between improving the formulation and the computational effort allowed at subsequent nodes in the enumeration tree. Also, we limit unconsciously the number of added cuts and thus, the size of the formulation is controlled.

4 Branching

As mentioned in the B&C algorithm in Section 2, the approach starts with a root node representing the original formulation of the problem and constructs iteratively an enumeration tree by partitioning the solution space associated to the current node into subspaces. Thus, each active node in the enumeration tree is evaluated. If this node can not be discarded, offspring nodes are then generated by choosing variables for branching.

It is well-known that the choice of the branching variables can influence considerably the size of the enumeration tree. In fact, there is a branching on certain variables that does not lead to any improvement and, hence, it is computationally very expensive. In addition, from prior experiments, we found that using only one variable for partitioning is not an appropriate branching strategy for our test sets since we deal with difficult large problems. Thus, if the current solution can not be discarded, we propose to select two branching binary variables y_{e_1} and y_{e_2} with fractional values, and to create four offspring nodes obtained by simply fixing the chosen variables either to 1 or to 0 when solving the LP relaxation of the subproblem associated with the parent node. The issue here is how to choose such variables.

In order to concentrate on promising parts of the solution space, we propose a branching rule motivated by reaching a good balance between the fixed cost of the arc and the amount of flow moving through. Basically, for a given arc with fractional value at the current LP solution (\bar{x}, \bar{y}) , if the carried flow is important when compared to the fixed cost required to open the arc, then it is promising to open the arc (i.e. set the design variable to 1) and to explore this branch first. Similar branching strategy has been used in Crainic, Delorme, and Dejax (1993) in their Branch-and-Bound algorithm for multicommodity location problem where they succeed to find rapidly good feasible solutions.

Based on this motivation and on the fact that in any feasible solution, we have to open enough arcs at origin-destination nodes to cover the demand, we try to identify the two branching variables, one after the other, by considering incident arcs at origin

or destination nodes first. To do so, one computes

$$e_1 = \operatorname{argmin}_{(i,j) \in (\tilde{A}_O \cup \tilde{A}_D)} \frac{f_{ij}}{\sum_{k \in K} x_{ij}^k},$$

where

- \tilde{A}_O is the set of all outgoing arcs for any origin node where the available capacity at that node does not cover the total supply : $\tilde{A}_O = \{(i, j) \in A \setminus \bar{A} \mid i \in O, \sum_{(i,j) \in \bar{A}_1 \cap N_i^+} u_{ij} < \sum_{k \in K} d_i^k, j \in N_i^+, \sum_{k \in K} x_{ij} > 0\}$,
- similarly, \tilde{A}_D is the set of all ingoing arcs for any destination node where the available capacity at that node does not cover the total demand : $\tilde{A}_D = \{(j, i) \in A \setminus \bar{A} \mid i \in D, \sum_{(j,i) \in \bar{A}_1 \cap N_i^-} u_{ji} < \sum_{k \in K} d_i^k, j \in N_i^-, \sum_{k \in K} x_{ji} > 0\}$, and
- $O = \{i \in N \mid \exists k \in K, i = O(k)\}$ and $D = \{i \in N \mid \exists k \in K, i = D(k)\}$.

Once the first arc e_1 is determined, we select the second arc e_2 using the same criterion, excluding, of course, the arc e_1 from the set of free nodes $A \setminus \bar{A}$. However, if e_1 does not exist, meaning that we have enough capacity on all origin-destination nodes, using the same criterion, we select any two free arcs from $A \setminus \bar{A}$ that realize the minimum.

So, if we succeed to select such two arcs, four nodes are then created :

- node 1, obtained by setting $e_1 = 0$ and $e_2 = 0$,
- node 2, obtained by setting $e_1 = 0$ and $e_2 = 1$,
- node 3, obtained by setting $e_1 = 1$ and $e_2 = 0$, and
- node 4, obtained by setting $e_1 = 1$ and $e_2 = 1$.

These nodes are then added into the list node by pushing at the top of the list node 1, node 2, node 3, and node 4, respectively. The first node at the top of the list is then selected to be evaluated (node 4, in this case). By doing so, we explore first the most promising branch of the enumeration tree, and thus we increment the chance to find good feasible solutions.

If, however, we have succeed to identify just one arc e_1 , meaning that either there is only one origin or destination node with only one fractional arc, or there is only

one arc with fractional value at the current LP solution, two offspring nodes are then created and added to the node list, in the same way as before.

Considering the way that the node list is updated, by pushing at the top, and the node is selected, by extracting the first node at the top of the list, it is clear that our searching strategy is the depth-first. Thus, we keep moving deeper in the enumeration tree until either an infeasible problem occurs, an integer solution is found, or the optimal LP solution exceeds the best upper bound. This choice of searching strategy is based on the motivation that the LP relaxation of any node and its offspring is almost the same, except for some added cuts, if any, and the bound values for the two selected branching variables. Therefore, the evaluation of the node is accelerated considerably by reoptimization which is done by considering the previous optimal bases in the simplex method.

5 Upper Bound Heuristic

Note that, assigning 0 or 1 to each design variable determines a design vector, \tilde{y} , and its associated network structure defined by $G = (N, \tilde{A})$, where $\tilde{A} \subseteq A$ and $\tilde{A} = \{(i, j) \in A, \tilde{y}_{ij} = 1\}$. So, considering the design vector \tilde{y} and its associated subset of arcs \tilde{A} , the MCND becomes the following continuous programming problem known as the multicommodity capacitated network flow problem (MCNF) :

$$\min Z(x(\tilde{y})) = \sum_{k \in K} \sum_{(i,j) \in \tilde{A}} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in \tilde{A}} f_{ij},$$

$$\sum_{j \in \tilde{N}_i^+} x_{ij}^k - \sum_{j \in \tilde{N}_i^-} x_{ji}^k = \begin{cases} d^k, & \text{if } i = O(k), \\ -d^k, & \text{if } i = D(k), \quad \forall i \in N, \forall k \in K, \\ 0, & \text{otherwise,} \end{cases}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij}, \quad \forall (i, j) \in \tilde{A},$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in \tilde{A}, \forall k \in K,$$

where $\tilde{N}_i^+ = \{j \in N | (i, j) \in \tilde{A}\}$ and $\tilde{N}_i^- = \{j \in N | (j, i) \in \tilde{A}\}$.

To obtain feasible solutions for the MCND which will be used to update the best feasible solution and to cut branches in the B&C algorithm, we propose to determine a fixed design vector \tilde{y} and its associated subset of arcs \tilde{A} , and to solve the underlying MCNF problem. If the MCNF is not feasible, then the network represented by \tilde{A} does not contain enough arcs to satisfy the demand. If, however, the subset \tilde{A} contains too many arcs, then some of them will not be used to carry any flow in the MCNF optimal solution. In this case, those arcs are eliminated from the solution simply by assigning 0 to their design variables and by removing their corresponding fixed cost from the optimal objective value $Z(\tilde{x}(\tilde{y}))$, where \tilde{x} is the MCNF optimal solution. Thus, the feasible solution to the MCND is determined as

$$\begin{aligned}\bar{x}_{ij}^k &= \begin{cases} \tilde{x}_{ij}^k & , \text{if } (i, j) \in \tilde{A}, \\ 0 & , \text{otherwise,} \end{cases} & \forall k \in K, \forall (i, j) \in A, \\ \bar{y}_{ij} &= \begin{cases} 0 & , \text{if } \sum_{k \in K} \bar{x}_{ij}^k = 0, \\ 1 & , \text{otherwise,} \end{cases} & \forall (i, j) \in A,\end{aligned}$$

and the upper bound on the MCND optimal solution is computed as

$$\bar{Z} = \sum_{k \in K} \sum_{(i, j) \in A} c_{ij}^k \bar{x}_{ij}^k + \sum_{(i, j) \in A} f_{ij} \bar{y}_{ij}.$$

But, the issue to address here is how to determine \tilde{A} . Indeed, the choice of \tilde{A} influences the performance of the B&C algorithm since finding rapidly good feasible solutions, i.e. tight upper bounds, avoids unnecessary computational effort through fathoming. Recall that, our branching rule is based on exploring the most promising parts of the space solution first, then we believe that those space parts contain high quality feasible solutions. Thus, we propose a heuristic approach, denoted by “ α -fixing”, to determine a promising set \tilde{A} using the design variables already fixed at the current node, \bar{A} , as well as other sources of informations accumulated through the B&C algorithm.

More precisely, starting with $\tilde{A} = \bar{A}_1$, the heuristic α -fixing tries to augment this subset by adding recursively some arcs to it. Those arcs are selected using simultaneously two strategies.

The first strategy aims to select arcs using the informations provided by the frequency, on average, that arcs appears in previous LP solutions. The idea is based on the remark that if some arcs are repeatedly used in previous optimal LP solutions, then those arcs are most likely to be used in good feasible solutions, or even in the optimal solution. Thus, it is promising to include them into \tilde{A} . Ideally, we should include in \tilde{A} arcs with frequency average greater than some predetermined threshold, denoted α . To achieve this, let $F \in R^{|A|}$ be the vector of frequency of all design variables and N_{LP} be the number of LP solved during the B&C algorithm. The vector F is updated at each LP solution (\bar{x}, \bar{y}) by setting

$$F_{ij} = F_{ij} + 1, \text{ if } \bar{y}_{ij} > \beta,$$

where β is used as a threshold to determine the importance of an arc (i, j) in the current LP solution. An arc (i, j) is added to \tilde{A} and, thus, fixed to 1, if

$$F_{ij} \geq \alpha N_{LP}.$$

Once the selection of arcs using the first strategy is performed, we use the second strategy to complete the construction of the set \tilde{A} . The main idea in the second strategy is based on the fact that any node who has at least one outgoing arc opened, must have at least one ingoing arc opened, and inversely. We believe that this idea is useful since any transfer node that has outgoing arcs (or ingoing arcs) without any ingoing arcs (outgoing arcs) is not useful for any solution. But, the issue here is which arc to select in this case.

Note that, we cannot only use the frequency vector F to chose such arcs since the available arcs have not been chosen by the first strategy, meaning that they do not show an important appearance in previous LP solutions. Therefore, we must use other informations in order to determine suitable arcs. To this aim, two sources of informations may be used : the reduced cost of design variables and the frequency of arcs that have been included in violated LCI, generated during the cutting-plane procedure. A small accumulated reduced cost value for a given arc (i, j) indicates that this arc

is an interesting one to be used to get a feasible solution. Also, if an arc has been used repeatedly in the generation of minimal cover in a violated LCI, then we would believe that such arc is a good candidate to be open, and thus to be used in feasible solutions. Define $R \in R^{|A|}$ and $L \in R^{|A|}$ as the accumulated reduced cost vector and the LCI-frequency vector of arcs, respectively. The B&C algorithm starts with $R = 0^{|A|}$, the null vector, and $L = 1^{|A|}$, the identity vector. During the resolution approach, R is updated using the reduced cost, denoted \bar{R} , associated with each LP solution as follows

$$R_{ij} = R_{ij} + \bar{R}_{ij}, \forall (i, j) \in A.$$

Similarly, the LCI-frequency vector L is updated during the cutting-plane procedure each time we succeed to find violated LCI as follows

$$L_{ij} = L_{ij} + 1, \forall (i, j) \in C,$$

where C is the minimal cover obtained in the LCI. So, to complete the selection of variables using the second strategy, for each node $i \in N$ which has at least one outgoing arc, $\sum_{j \in \tilde{N}_i^+} y_{ij} \geq 1$, and no ingoing arc, $\sum_{j \in \tilde{N}_i^-} y_{ji} = 0$, we select and add to \tilde{A} the arc which realizes

$$\operatorname{argmax}_{(j,i), j \in N_i^- \setminus \tilde{N}_i^-} F_{ji} L_{ji} (-R_{ji}).$$

Similarly, for each node $i \in N$ which has at least one ingoing arc, $\sum_{j \in \tilde{N}_i^-} y_{ji} \geq 1$, and no outgoing arc, $\sum_{j \in \tilde{N}_i^+} y_{ij} = 0$, we select and add to \tilde{A} the arc which realizes

$$\operatorname{argmax}_{(i,j), j \in N_i^+ \setminus \tilde{N}_i^+} F_{ij} L_{ij} (-R_{ij}).$$

To sum up, once the set \tilde{A} is determined using the α -fixing heuristic, composed from the first and the second strategies, the underlying MCNF problem is solved. If the problem is feasible, its optimal solution is then trimmed in order to identify feasible solution for the MCND problem.

According to our computational results, this heuristic has proved effective to identify rapidly, in most cases after exploring only few nodes in the enumeration tree, feasible solutions of good quality.

6 Computational Results

In this section we report on our computational experience with the B&C algorithm. To evaluate the performance of the code, results are compared with those of the B&C of CPLEX (version 7.1, 2002) obtained on the *strong formulation*, with default options for all parameters. A limit of 10 hours of computation was imposed for each instance in the two codes. Since the B&C of CPLEX was unable to identify any feasible solution for large size problems with many commodities, we compare also our results with those obtained by the meta-heuristic *Path-Relinking* approach, denoted PR, proposed in Ghamlouche, Crainic, and Gendreau (2002). Indeed, to make a valid comparison between the B&C and the PR approaches, we use the solution found in our code within the same time used by the PR approach. However, when the PR time solution takes more than 10 hours of CPU time, we use the solution found within the time allowed to our code. The implementation is done in C++. All experimentations, for the three codes, are performed on a Sun Enterprise 10000 with 64 Gigabytes of RAM, operating under solaris 2.7.

The values of the parameters in our implementation are $\alpha = 0.45$ and $\beta = 0.3$. In addition, in the cutting-plane procedure we set $T_{max} = 5$, $I_{max} = 20$, and $M_{max} = \lceil \frac{N}{3} \rceil$. We use the option *dualopt* of CPLEX (version 7.1, 2002) to solve to optimality the LP relaxation and the multicommodity capacitated network flow problem arises in the α -fixing heuristic.

To test our algorithm, we use two data sets of general transshipment networks with one commodity per origin-destination and no parallel arcs. One positive fixed cost and one positive transportation cost, which is the same for all commodities, are associated with each arc. The same test sets have been used in several papers on the MCND such as Crainic, Gendreau, and Farvolden (1998), Ghamlouche, Crainic, and Gendreau (2001, 2002), and Crainic, Gendron, and Hernu (2002).

The first data set, denoted **C**, consists of 43 problem instances defined by the number of nodes, the number of arcs, and the number of commodities denoted $|N|$, $|A|$, and

$|K|$ respectively. Each instance is characterized also by two additional letters which give the fixed cost level compared to the transportation cost, “F” for high and “V” for low, and the capacity level compared to the total demand, “T” for tight and “L” for loose.

The second set of instances, denoted **R**, consists of 81 problem instances : nine sets of nine problem instances each. Each set is characterized by the same number of nodes, arcs, and commodities and were generated by combining three levels of fixed cost ratios and three levels of capacity ratios. The fixed cost ratio is computed as $F = |K| \sum_{(i,j) \in A} f_{ij} / \sum_{k \in K} d^k \sum_{(i,j) \in A} c_{ij}^k$, where three values are considered, $F = 0.01$ (F01), $F = 0.05$ (F05), and $F = 0.10$ (F10), to measure the importance of the fixed cost compared to the transportation cost. Similarly, the capacity ratio is computed as $C = |A| \sum_{k \in K} d^k / \sum_{(i,j) \in A} u_{ij}$, where three values are considered, $C = 1$ (C1), $C = 2$ (C2), and $C = 8$ (C8), to measure the tightness of the capacity relatively to the total demand.

Table 1 and 2 display results for the set **C**. For each problem, we report the description (PROB) by identifying the number of nodes, arcs, and commodities as well as two additional letters summarizing the fixed cost and capacity information. The optimal solution (indicated by *) or the best feasible solution found by the B&C of CPLEX and the number of explored nodes are given in columns 2 and 3, denoted CPLEX and CPLEX NODES, respectively. The letter **X** in the CPLEX column indicates that the B&C of CPLEX was unable to identify any feasible solution within the time allowed. The column PR shows the solution found by the path-relinking approach. Columns UB, BC, and BC NODES give respectively the solution found in our B&C using the same time as PR, without exceeding the time limit, the best feasible solution found by our B&C and the number of explored nodes in the enumeration tree. The CPU time elapsed in seconds is given under each solution. The letter **t** indicates that the solution is feasible but its optimality could not be verified within the time allowed. The last three columns give respectively the gap of UB with respect to PR, the gap of BC with respect to PR and the gap of BC with respect to CPLEX.

Considering the results shown in Tables 1 and 2, we conclude the following. CPLEX is efficient to solve small size instances. But when the problem size is getting larger, CPLEX was unable to verify the optimality of the solution, or even was unable to find any feasible solution, within the time allowed. The solutions found in our B&C algorithm approximate closely those found by CPLEX for small to medium size instances. Indeed, for problems with high fixed cost and 100 or 200 commodities, our solution is very close to the one found by CPLEX and outperforms it for 4 problems out of 8 where the maximal improvement reaches up to 1.99% (problem 20-230-200-F-L) and the average improvement is 0.04%. Compared to the PR solution and using the same time, our UB feasible solution, is somewhat competitive. For large size instances with 400 commodities, CPLEX does not manage to find any feasible solution for 6 instances out of 8. For these problems, the solutions found in our B&C outperform those found by the PR approach (except for only one instance) in much shorter time. We note that, for problems where optimality could not be verified within the time of computation allowed, we have been able to explore much more nodes than CPLEX. This fact is due especially to the very large size of the strong formulation, the formulation presented in Section 2 including all strong inequalities. Although the strong formulation contains much larger number of constraints, $|A| \times |K|$ additional constraints, it provides much better solution when using the B&C of CPLEX (Chouman, Crainic, and Gendron 2001).

To make more deeply comparison, Table 3 summarizes the average gap distribution according to the number of commodities and the different level of fixed cost and capacity ratios. The gap of the UB with respect to PR, the gap of the BC with respect to PR and the gap of the BC with respect to CPLEX are given in UB_{PR} , BC_{PR} and BC_{CPLEX} , respectively. These results indicate that our upper bound UB is competitive with the PR solution and outperforms it, in average, for the class of problems with high fixed cost and large capacity (FL). Our BC solution is slightly better than the one found by CPLEX for this class of problems where the number of commodities is 100 or 200. We note that, the maximal average gap of our BC solution is less than 1.51% when compared to CPLEX. It is clear that our B&C gives the better performance in terms of quality of solution and computation time when compared to the PR approach

PROB	CPLEX	CPLEX NODES	PR	UB	BC	BC NODES	UBPR	GAP BCPR	BCCPLEX
25-100-10	14712*	5	14712	14712	14712	97	0%	0%	0%
V-L	(0.36)		(12.9)		(12.1)				
25-100-10	14941*	432	14941	15358	14941	5957	2.79%	0%	0%
F-L	(53.6)		(15.2)		(4147.8)				
25-100-10	49899*	419	49899	51364	49899	37971	2.93%	0%	0%
F-T	(40.6)		(22.1)		(10278.2)				
25-100-30	365272*	42	365322	365282	365272	849	-0.01%	-0.01%	0%
V-T	(16.6)		(91.9)		(380.7)				
25-100-30	37055*	2449	37928	40535	37325.5	16789	6.87%	-1.59%	0.73%
F-L	(1727.9)		(72.6)		(t)				
25-100-30	85530*	1350	86215	86481	85530	31659	0.31%	-0.79%	0%
F-T	(534.2)		(97.8)		(t)				
20-230-40	423848*	15	424385	424242	423848	717	-0.03%	-0.13%	0%
V-L	(10.4)		(129.5)		(421.0)				
20-230-40	371475*	102	371475	371681	371475	22767	0.05%	0%	0%
V-T	(52.4)		(141.4)		(26116.3)				
20-230-40	643036*	1618	644464	645025	643036	44897	0.08%	-0.22%	0%
F-T	(671.7)		(146.1)		(t)				
20-300-40	429398*	3	429398	429398	429398	3701	0%	0%	0%
V-L	(3.76)		(212.7)		(2296.8)				
20-300-40	586077*	214	593304	589925	588229	22413	-0.57%	-0.86%	0.37%
F-L	(145.5)		(223.9)		(t)				
20-300-40	464509*	191	464509	465311	464509	20007	0.17%	0%	0%
V-T	(83.8)		(213.7)		(14820.4)				
20-300-40	604198*	92	604828	605764	604198	8079	0.15%	-0.10%	0%
F-T	(59.8)		(137.3)		(8500.0)				
20-230-200	94386	42	100205	100530	94955	409	0.32%	-5.23%	0.60%
V-L	(t)		(2017.3)		(t)				
20-230-200	141737.4	8	148044	151128	138912	227	2.08%	-6.16%	-1.99%
F-L	(t)		(2493.5)		(t)				
20-230-200	97914	63	104128	101270	99930	477	-2.74%	-4.03%	2.06%
V-T	(t)		(1923.2)		(t)				
20-230-200	137271	11	144857	144331	136929	263	-0.36%	-5.47%	-0.25%
F-T	(t)		(2926.9)		(t)				
20-300-200	74972.4	31	79128.5	82412	75262.6	375	4.15%	-4.89%	0.39%
V-L	(t)		(3487.2)		(t)				
20-300-200	117306	5	123179	122666	119294	153	-0.42%	-3.15%	1.69%
F-L	(t)		(3765.3)		(t)				
20-300-200	74991	67	77839.5	77612	75713	489	-0.29%	-2.73%	0.96%
V-T	(t)		(3691.2)		(t)				
20-300-200	108252	10	111590	115195	108319	259	3.23%	-2.93%	0.06%
F-T	(t)		(3546.6)		(t)				

TAB. 1 – Computational Results, C problems

PROB	Cplex	Cplex Nodes	PR	UB	BC	BC Nodes	UBPR	GAP BCPR	BCcplex
100-400-10 V-L	28423* (84.8)	429	28485 (96.0)	28423	28423 (t)	15341	-0.21%	-0.21%	0 %
100-400-10 F-L	24436 (t)	3989	24244 (109.6)	27389	26139 (t)	3013	12.97%	7.81%	6.97%
100-400-10 F-T	66364 (t)	1253	64527 (193.4)	75597	68278 (t)	2607	17.15%	5.81%	2.88%
100-400-30 V-T	385544 (t)	8931	384879 (424.1)	396951	384917 (t)	9241	3.13%	0.01%	-0.16%
100-400-30 F-L	50496 (t)	141	51187 (297.1)	54983	50637 (t)	565	7.41%	-1.07%	0.28%
100-400-30 F-T	141278 (t)	104	141548 (529.0)	142974	142304 (t)	681	1.01%	0.53%	0.72%
30-520-100 V-L	53966 (t)	1115	54282 (1250.6)	55625	54112 (t)	4679	2.47%	-0.31%	0.27%
30-520-100 F-L	95294 (t)	25	102016 (1722.4)	98026	95270 (t)	505	-3.91%	-6.61%	-0.02%
30-520-100 V-T	52085 (t)	711	52510 (1381.4)	53155	52299 (t)	4271	1.22%	-0.40%	0.41%
30-520-100 F-T	98357 (t)	20	104188 (1336.9)	104429	98811 (t)	331	0.23%	-5.16%	0.46%
30-700-100 V-L	47603* (1736.1)	96	47924 (1730.2)	48259	47603 (t)	2585	0.70%	-0.67%	0%
30-700-100 F-L	60525 (t)	51	62349 (2084.1)	62843	60484 (t)	709	0.79%	-2.99%	-0.07%
30-700-100 V-T	45944.5 (t)	237	46565 (1633.9)	46507	46221 (t)	1101	-0.12%	-0.74%	0.60%
30-700-100 F-T	55082 (t)	118	56246 (1534.6)	56466	55333 (t)	683	0.39%	-1.62%	0.45%
30-520-400 V-L	112997.5 (t)	0	118873 (29650.7)	115216	115216 (t)	49	-3.07%	-3.07%	1.96%
30-520-400 F-L	X (t)	0	159007 (33641.2)	154493	154493 (t)	21	-2.84%	-2.84%	-
30-520-400 V-T	114949.1 (t)	1	119744 (20461.9)	115330	115330 (t)	49	-3.69%	-3.69%	0.33%
30-520-400 F-T	X (t)	0	163352 (59820.1)	157305	157305 (t)	25	-3.70%	-3.70%	-
30-700-400 V-L	X (t)	0	103138 (17881.3)	101007	101007 (t)	25	-2.07%	-2.07%	-
30-700-400 F-L	X (t)	0	144615 (57574.8)	145901	145901 (t)	17	0.89%	0.89%	-
30-700-400 V-T	X (t)	0	99606 (31405.7)	98269	98269 (t)	33	-1.34%	-1.34%	-
30-700-400 F-T	X (t)	0	138777 (48693.7)	137929	137929 (t)	17	-0.61%	-0.61%	-

TAB. 2 – Computational Results, C problems

K	VL			FL		
	UBPR	BCPR	BCCPLEX	UBPR	BCPR	BCCPLEX
40	-0.02%	-0.06%	0%	-0.57%	-0.86%	0.37%
100	1.58%	-0.49%	0.14%	-1.56%	-4.80%	-0.05%
200	2.24%	-5.06%	0.49%	0.83%	-5.66%	-0.15%
400	-2.57%	-2.57%	X	-1.87%	-1.87%	X
K	VT			FT		
	UBPR	BCPR	BCCPLEX	UBPR	BCPR	BCCPLEX
40	0%	0%	0%	0.12%	-0.16%	0%
100	0.55%	-0.57%	0.51%	0.31%	-3.39%	0.46%
200	-1.52%	-3.38%	1.51%	1.44%	-4.20%	-0.10%
400	-2.52%	-2.52%	X	-2.16%	-2.16%	X

TAB. 3 – Gap distribution according to problem characteristics, **C** problems

and to the software CPLEX for problems with 400 commodities.

	C1			C2			C8		
	UBPR	BCPR	BCCPLEX	UBPR	BCPR	BCCPLEX	UBPR	BCPR	BCCPLEX
F01	-0.75%	-1.42%	0.04%	-0.05%	-0.71%	0.06%	-0.63%	-1.38%	0.47%
F05	2.05%	-3.71%	0.41%	0.97%	-2.87%	1.51%	-1.55%	-3.10%	1.50%
F10	1.74%	-4.25%	0.99%	0.90%	-3.71%	1.28%	-2.56%	-5.30%	0.01%

TAB. 4 – Gap distribution according to fixed cost and capacity level, **R** problems

To avoid overloading the paper, we report on aggregated results only for the **R** problems. Detailed results for all instances are presented in the Appendix. Table 4 displays the aggregated gap for 9 group of problems each containing 9 instances. Problems are grouped according to their different level of fixed cost and capacity ratios from the easiest (F01, C1) to the most difficult (F10, C8). Results indicate that, for the same level of fixed cost, the improvement of the UB compared to PR augment as the capacity is more tight. Similarly, for problems with tight capacity C8, the improvement augment also as the fixed cost ratios is more important. Moreover, for the most difficult instances, (F10, C8), our UB solution outperforms the PR solution and reaches up to 2.56% on average over 9 difficult instances. Results support that the solutions found in our B&C algorithm approximate closely those found by CPLEX. For the most difficult problems, the maximum of gap reaches 0.01% on average for the 9 difficult instances.

Table 5 shows the average gap distribution for the **R** problems according to problem dimension. These results indicate that the improvement of the UB upon that of the PR

A	100			200			300		
	N , K	UBPR	BCPR	BCCPLEX	UBPR	BCPR	BCCPLEX	UBPR	BCPR
20, 40	2.16%	-1.10%	0.24%	2.14%	-2.29%	1.16%	1.22%	-1.82%	0.20%
20, 100	0.36%	-1.67%	0.31%	0.72%	-4.37%	0.28%	-0.43%	-4.10%	0.36%
20, 200	-1.94%	-3.96%	0.23%	-1.90%	-3.50%	1.72%	-3.14%	-4.69%	1.97%

TAB. 5 – Gap distribution according to problem dimension, **R** problems

increased as the problem size increased, especially when the number of commodities is getting larger. The behavior of our B&C is satisfactory when compared to CPLEX where the maximal average gap is 1.97% for the largest instances.

To sum up, over the 124 instances (**C** and **R** problems), we find that our UB solution is better than the PR solution in 61 cases, worse in 58 cases and equivalent in 5 cases. While, our BC solution is better than the one found by CPLEX in 18 cases, worse in 63 cases and equivalent in 43 cases.

The distribution of the quality of the UB and the BC solutions compared to PR

	PSet	<-3%	(-3%,-2%]	(-2%,-1%]	(-1%,0%]	(0%,1%]	(1%,2%]	(2%,3%]	>3%
UBPR	C	4	3	1	11	11	2	4	7
	R	10	5	12	21	12	5	4	12
BCCPLEX	C	6	-	1	17	14	2	2	1
	R	-	-	2	35	23	13	4	4

TAB. 6 – Distribution of relative improvement

and to CPLEX solutions are given in Table 6 for the two sets **C** and **R** separately. Table 6 shows that the performance, in terms of quality, of the PR approach and our B&C, using the same time of computation, are somewhat comparable. However, overall the instances, we found that our UB outperforms the PR and obtains an average improvement of 0.29% and 0.05% for the sets **C** and **R**, respectively, while the maximal improvement is 3.91% and 10.56%, respectively. Comparing the BC solutions and CPLEX solutions, Table 6 shows that the gap ranges essentially between -1% and 2% meaning that the BC solutions approximate closely those of CPLEX for almost instances. Indeed, overall the instances, we obtain an average gap of 0.33% and 0.70% for the sets **C** and **R**, respectively, while the maximal improvement is 1.99% and 1.87%, respectively.

Finally, the results support that the MCND problem is not yet solved and additional efforts have to be done. The proposed B&C algorithm constitutes a considerable step towards an efficient resolution of large-scale MCND problems, however. Indeed, for difficult large scale instances, it offers the best performance among the state-of-the-art software CPLEX and one of the best approximated methods proposed in the literature.

7 Conclusion

In this paper, we have proposed a Branch-and-Cut algorithm to solve the multicommodity capacitated fixed charge network design problem. The proposed B&C algorithm investigated the use of a cut generation procedure at any node in the enumeration tree. The cuts used in this procedure are derived from the cutset structure and are valid at all nodes. We have proposed to use the informations accumulated during the cutting-plane in the development of a new branching rule, as well as in computing feasible solutions of good quality. Computational results using 124 instances of various characteristics are also presented.

The results show that the proposed B&C algorithm is inferior than the state-of-the-art MIP software package CPLEX for small size instances. However, the two algorithms become somewhat competitive, in terms of solution quality, when the problem size is getting larger. Especially, for difficult and large scale instances with many commodities, the proposed B&C method gives the best performance in terms of solution quality and computational effort among the MIP software CPLEX and one of the best current approximated solution methods available in the literature. Although, the B&C show a relatively promising behavior, its performance must be improved, however, which justify additional efforts. In the following we point out to interesting futures.

- The new branching rule performs quite well, but the number of explored nodes may be reduced considerably by combining the depth-first and the best-bound strategies.

- The overall performance of the method may be improved by performing periodically some refinement steps such as variable fixing and preprocessing.
- An appropriate control of the formulation size might improve the computational effort at each node. This is done either by limiting the number of added cuts at each node or by considering the effectiveness of added cuts and periodically delete some of them.
- To improve the quality of the upper bounds, we may combine the α -fixing heuristic to a more sophisticated heuristic methods such as Tabu search method.
- The cutting-plane procedure is useful to improve the formulation of the problem. However, it is not sufficient to compute tight lower bounds. To improve these bounds, additional polyhedral study of the problem structure is required.
- The evaluation of a node may be accelerated by using a Lagrangian approach. In this case, the violated cuts may be relaxed and added to the objective function in order to preserve the particular structure of the subproblems.
- Many of the approaches and techniques used through this paper can be useful in studying other combinatorial problems with the same flavor, especially network design problems. Thus it is interesting to investigate the usefulness of these approaches to solve other related problems.

We intend to report on the investigation of some of these subjects in forthcoming developments.

Bibliographie

- [1] Balas E., Ceria S., and Cornuéjols G. 1996, *Mixed 0-1 Programming by Lift-and-Project in a Branch-and-Cut Framework*, Management Science vol. 42, 9, 1229-1246.
- [2] Chouman M., Crainic T.G. and Gendron B. 2001, *Revue des inégalités valides pertinentes aux problèmes de conception de réseaux*, INFOR, forthcoming.
- [3] Chouman M., Crainic T.G. and Gendron B. 2003, *A Cutting-Plane Algorithm Based on Cutset Inequalities for Multicommodity Capacitated Fixed Charge Network Design*, Publication CRT-03, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [4] CPLEX Optimization, Inc. 2002, *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, version 7.1.
- [5] Crainic, T.G. 1999, *Network Design in Freight Transportation*, European Journal of Operational Research, 122, (2), 272-288.
- [6] Crainic T.G., Delorme L. and Dejax P.J. 1993, *A Branch-and-Bound Method for Multicommodity Location with Balancing Requirements*, European Journal of Operational Research 65, 3, 368-382.
- [7] Crainic T.G., Frangioni A. and Gendron B. 2002, *Bundle-Based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems*, Discrete Applied Mathematics, 112, 73-99.
- [8] Crainic T.G. and Gendreau M. 2002, *Cooperative Parallel Tabu Search for Capacitated Network Design*, Journal of Heuristics, 8, (6), 601-627.

- [9] Crainic T.G., Gendreau M. and Farvolden J.M. 2000, *A Simplex-Based Tabu Search Method for Capacitated Network Design*, *INFORMS Journal on Computing* 12, (3), 223-236.
- [10] Crainic T.G., Gendron B. and Hernu G. 2002, *A Slope Scaling/Lagrangian Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design*, Publication CRT, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [11] Gendron B. and Crainic T.G. 1994, *Relaxations for Multicommodity Capacitated Network Design Problems*, Publication CRT-96-05, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [12] Gendron B. and Crainic T.G. 1996, *Bounding Procedures for Multicommodity Capacitated Fixed Charge Network Design Problems*, Publication CRT-96-06, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [13] Gendron B., Crainic T.G. and Frangioni A. 1998, *Multicommodity Capacitated Network Design, Telecommunications Network Planning*, Sansó B., Soriano P. (éds.), Kluwer Academic Publishers, Norwell MA., 1-19.
- [14] Ghamlouche I., Crainic T.G. and Gendreau M. 2001, *Cycle-Based Neighborhoods for Fixed-Charge Capacitated Multicommodity Network Design*, *Operations Research* forthcoming.
- [15] Ghamlouche I., Crainic T.G. and Gendreau M. 2002, *Path Relinking, Cycle-Based Neighborhoods and Capacitated Multicommodity Network Design*, Publication CRT, Centre de recherche sur les transports, Université de Montréal, Montréal.
- [16] Günlük O. 1999, *A Branch-and-Cut Algorithm for Capacitated Network Design Problems*, *Mathematical Programming* 86, 17-39.
- [17] Holmberg K. and Yuan D. 2000, *A Lagrangian Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem*, *Operations Research* 48, 461-481.
- [18] Magnanti T.L. and Wong R.T. 1984, *Network Design and Transportation Planning : Models and Algorithms*, *Transportation Science* 18, 1-55.

- [19] Minoux M. 1989, *Network Synthesis and Optimum Network Design Problems : Models, Solution Methods and Applications*, Networks 19, 313-360.
- [20] Nemhauser G.L., Savelsbergh M.W.P. and Sigismondi G. 1994, *MINTO, a Mixed Integer Optimizer*, Operations Research Letters 15, 47-58.
- [21] Padberg M. and Rinaldi G. 1987, *Optimization of a 532-City Traveling Salesman Problem by Branch-and-Cut*, Operation Research Letters 6, 1-7.
- [22] Padberg M. and Rinaldi G. 1991, *A Branch-and-Cut Algorithm for the Resolution of Large-scale Symmetric Traveling Salesman Problem*, SIAM Revue 33, 60-100.
- [23] Sellmann M., Kliewer G. and Koberstein A. 2002, *Capacitated Network Design, Cardinality Cuts and Coupled Variable Fixing Algorithms based on Lagrangian Relaxations*, Technical report tr-ri-02-234, University of Paderborn.
- [24] Van Roy T.J. and Wolsey L.A. 1987, *Solving Mixed Integer Programming Problems Using Automatic Reformulation*, Operations Research 35, 45-57.
- [25] Xpress-MP 2002, *Xpress-MP Integrated into MIMI*, Modeling and Optimization Software, Dash Optimization Inc., New York, NY.

Appendix

Results for each problem instance of the set \mathbf{R} are given in the next tables. Each set of problem contains nine instances with the same dimension identified by the number of nodes, the number of arcs, and the number of commodities. For each set, the following information is displayed.

- **PROB** : Two letters are used to characterize the fixed cost level relatively to the transportation cost and the capacity level compared to the total demand.
- **CPLEX** and **CPLEX NODES** : The optimal (indicated by *) or the best feasible solution obtained by the Branch-and-Cut (B&C) of CPLEX (version 7.1, 2002) and the number of explored nodes in the enumeration tree. The CPU time elapsed in seconds is given under each solution. The letter (t) indicates that the solution is feasible but its optimality could not be verified within 10 hours of CPU time.
- **PR** : The solution obtained by the path-relinking approach.
- **UB** : The solution found in our B&C within the same computation time as PR.
- **BC** and **BC NODES** : The best feasible solution found in our B&C within the limit of time allowed and the number of explored nodes.
- The last three columns indicate the gap of UB with respect to PR, the gap of BC with respect to PR and the gap of BC with respect to CPLEX, respectively.

All experiments, for the three approaches, are performed on a Sun Enterprise 10000 with 64 Gigabytes of RAM and operating under Solaris 2.7.

TAB. 7 – Computational Results Simple Cutting-Plane R10 : 20,100,40

PROB	CPLEX	CPLX NODES	PR	UB	BC	BC NODES	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	200087*	47	200147 (75.2)	200087	200087 (50.6)	125	-0.03%	-0.03%	0%
F05-C1	346813.5*	1208	347459 (788.2)	378155	346813 (16971.8)	5675	8.83%	-0.19%	0%
F10-C1	488015*	641	502724 (563.3)	525214	488015 (4468.6)	1081	4.47%	-2.93%	0%
F01-C2	229196*	431	229196 (184.3)	234851	229251 (t)	17265	2.46%	0.02%	0.02%
F05-C2	411664*	5027	425230 (5123.2)	424978	415683 (t)	9265	-0.06%	-2.24%	0.97%
F10-C2	609104*	13654	628353 (27260.5)	649912	615961 (t)	7819	3.43%	-1.97%	1.12%
F01-C8	486895*	542	487235 (391.3)	488890	486895 (t)	27889	0.34%	-0.07%	0%
F05-C8	951056*	732	963914 (555.5)	967425	951661 (t)	26985	0.36%	-1.27%	0.06%
F10-C8	1421740*	1091	1439320 (608.4)	1438540	1421750 (t)	25405	-0.05%	-1.22%	0%

TAB. 8 – Computational Results Simple Cutting-Plane R11 : 20,100,100

PROB	CPLEX	CPLX NODES	PR	UB	BC	BC NODES	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	714431*	153	720236 (392.5)	714431	714431 (4271.4)	1223	-0.81%	-0.81%	0%
F05-C1	1265985	1284	1296050 (t)	1326340	1279370 (t)	969	2.34%	-1.29%	1.06%
F10-C1	1846295	644	1940180 (t)	2018070	1873680 (t)	805	4.01%	-3.43%	1.48%
F01-C2	870451*	760	875908 (2960.1)	876192	871320 (t)	8607	0.03%	-0.52%	0.09%
F05-C2	1623640*	2574	1674340 (21739.5)	1651260	1627160 (t)	1349	-1.38%	-2.82%	0.22%
F10-C2	2414060	1647	2546730 (t)	2544520	2413990 (t)	961	-0.08%	-5.21%	-0.01%
F01-C8	2294912*	14	2294912 (31.5)	2294912	2294912 (2029.1)	855	0%	0%	0%
F05-C8	3507100*	51	3529530 (79.8)	3508520	3507100 (25792.6)	14673	-0.59%	-0.64%	0%
F10-C8	4579353*	11	4592820 (48.5)	4579353	4579353 (1378.4)	737	-0.29%	-0.29%	0%

TAB. 9 – Computational Results Simple Cutting-Plane R12 : 20,100,200

PROB	Cplex	Cplex Nodes	PR	UB	BC	BC Nodes	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	1639443* (4436.7)	419	1668370 (2434.6)	1642190	1639443 (t)	2561	-1.57%	-1.73%	0%
F05-C1	3403074.2 (t)	134	3717800 (2975.4)	3634910	3409220 (t)	311	-2.23%	-8.30%	0.18%
F10-C1	5276171 (t)	94	6037960 (3245.5)	6049730	5374350 (t)	465	0.19%	-10.99%	1.86%
F01-C2	2303557* (1739.8)	106	2326930 (5539.9)	2308760	2303560 (t)	1631	-1.01%	-1.00%	0%
F05-C2	4669799* (3145.6)	125	4954250 (5709.1)	4707990	4669799 (t)	1289	-4.97%	-5.74%	0%
F10-C2	7100019* (1683.8)	51	7638050 (4735.9)	7100019	7100019 (24326.5)	983	-7.04%	-7.04%	0%
F01-C8	7635270* (78.1)	2	7638050 (4735.9)	7635270	7635270 (1490.5)	257	-0.04%	-0.04%	0%
F05-C8	10067742* (61.5)	2	10085000 (3315.9)	10067742	10067742 (1805.9)	173	-0.17%	-0.17%	0%
F10-C8	11967768* (50.3)	0	12042800 (2276.5)	11967768	11967768 (61.6)	1	-0.62%	-0.62%	0%

TAB. 10 – Computational Results Simple Cutting-Plane R13 : 20,200,40

PROB	Cplex	Cplex Nodes	PR	UB	BC	BC Nodes	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	142947* (6.4)	10	142947 (130.4)	142947	142947 (212.7)	325	0%	0%	0%
F05-C1	263800* (547.9)	206	273853 (141.3)	279342	263800 (t)	8307	2.00%	-3.67%	0%
F10-C1	365800* (116.6)	187	378922 (145.8)	413221	365836 (t)	4555	9.03%	-3.45%	0.01%
F01-C2	150977* (90.1)	141	150977 (143.1)	151740	150977 (25304.0)	10255	0.51%	0%	0%
F05-C2	282682* (3551.8)	1885	289706 (142.8)	306933	286600 (t)	6537	5.94%	-1.07%	1.38%
F10-C2	406790* (29695.6)	6354	435724 (159.9)	450997	422921 (t)	2761	3.51%	-2.94%	3.96%
F01-C8	208088 (t)	7346	210898 (183.1)	212624	209970 (t)	6175	0.82%	-0.44%	0.90%
F05-C8	446997 (t)	1140	469035 (182.0)	470178	458063 (t)	2379	0.24%	-2.34%	2.48%
F10-C8	698280 (t)	942	761505 (232.1)	740309	710294 (t)	2497	-2.78%	-6.72%	1.72%

TAB. 11 – Computational Results Simple Cutting-Plane R14 : 20,200,100

PROB	CPLEX	CPLEX NODES	PR	UB	BC	BC NODES	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	403414*	221	410430	403897	403414	1171	-1.59%	-1.71%	0%
	(495.7)		(413.9)		(2852.4)				
F05-C1	750091	354	778492	833940	752929	1629	7.12%	-3.28%	0.38%
	(t)		(503.9)		(t)				
F10-C1	1078595	118	1127980	1115500	1088790	637	-1.11%	-3.74%	0.94%
	(t)		(555.5)		(t)				
F01-C2	437607*	467	444170	442631	437607	7879	-0.35%	-1.48%	0%
	(1327.3)		(463.3)		(t)				
F05-C2	852330	228	883241	923389	863178	951	4.54%	-2.27%	1.27%
	(t)		(580.6)		(t)				
F10-C2	1230749.7	72	1335170	1378910	1216100	569	3.28%	-8.92%	-1.19%
	(t)		(608.6)		(t)				
F01-C8	670064	266	689762	676832	672549	1515	-1.87%	-2.49%	0.37%
	(t)		(1294.3)		(t)				
F05-C8	1643376	110	1740600	1699850	1645250	695	-2.34%	-5.48%	0.11%
	(t)		(1810.3)		(t)				
F10-C8	2630332	115	2939990	2908150	2646800	549	-1.08%	-9.97%	0.62%
	(t)		(2555.5)		(t)				

TAB. 12 – Computational Results Simple Cutting-Plane R15 : 20,200,200

PROB	CPLEX	CPLEX NODES	PR	UB	BC	BC NODES	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	1000787*	338	1027930	1015380	1000790	2119	-1.22%	-2.64%	0%
	(5384.7)		(1361.7)		(t)				
F05-C1	1979413	15	2135640	2096380	1987930	147	-1.84%	-6.92%	0.43%
	(t)		(1871.1)		(t)				
F10-C1	2949264	2	3131640	3018470	2980480	61	-3.61%	-4.83%	1.06%
	(t)		(2106.4)		(t)				
F01-C2	1148604	312	1184780	1175240	1149770	679	-0.81%	-2.95%	0.10%
	(t)		(2049.6)		(t)				
F05-C2	2488753	9	2732310	2753960	2695310	123	0.79%	-1.35%	8.30%
	(t)		(5114.4)		(t)				
F10-C2	3972667	2	4264090	4231150	4170730	137	-0.77%	-2.19%	4.98%
	(t)		(4594.5)		(t)				
F01-C8	2301326.6	131	2350730	2324970	2309760	417	-1.09%	-1.74%	0.37%
	(t)		(14515.9)		(t)				
F05-C8	5573412.8*	357	5830430	5598080	5587420	417	-3.99%	-4.17%	0.25%
	(16107.9)		(23169.4)		(t)				
F10-C8	8696932*	20	9123020	8706750	8696932	499	-4.56%	-4.67%	0%
	(18930.9)		(19548.3)		(t)				

TAB. 13 – Computational Results Simple Cutting-Plane R16 : 20,300,40

PROB	CPLEX	CPLEX NODES	PR	UB	BC	BC NODES	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	136161*	0	136161	136161	136161	1	0%	0%	0%
	(1.1)		(194.4)		(9.4)				
F05-C1	239500*	340	244070	242465	239500	6891	-0.66%	-1.87%	0%
	(1641.3)		(200.4)		(t)				
F10-C1	325671*	531	330138	349507	325671	3057	5.87%	-1.35%	0%
	(4345.9)		(212.2)		(t)				
F01-C2	138532*	23	139131	138532	138532	337	-0.43%	-0.43%	0%
	(27.6)		(184.9)		(265.4)				
F05-C2	241801*	95	248717	251252	241801	1053	1.02%	-2.78%	0%
	(851.9)		(204.7)		(14359.0)				
F10-C2	337762*	961	343165	349809	340859	1993	1.94%	-0.67%	0.92%
	(8173.5)		(206.9)		(t)				
F01-C8	169502	4482	171651	173046	170300	6467	0.81%	-0.79%	0.47%
	(t)		(209.2)		(t)				
F05-C8	352976	486	371129	374607	360999	3709	0.94%	-2.73%	2.27%
	(t)		(244.1)		(t)				
F10-C8	541626	295	564151	572854	531486	1197	1.54%	-5.79%	-1.87%
	(t)		(229.9)		(t)				

TAB. 14 – Computational Results Simple Cutting-Plane R17 : 20,300,100

PROB	CPLEX	CPLEX NODES	PR	UB	BC	BC NODES	GAP		
							UBPR	BCPR	BCCPLEX
F01-C1	354138*	41	364325	354138	354138	537	-2.80%	-2.80%	0%
	(332.1)		(536.7)		1822.8)				
F05-C1	651025	255	669935	687479	647696	507	2.62%	-3.32%	-0.51%
	(t)		(710.9)		(t)				
F10-C1	917956	63	973486	968363	935451	325	-0.53%	-3.91%	1.90%
	(t)		(845.3)		(t)				
F01-C2	370590*	459	378905	374172	370590	4243	-1.25%	-2.19%	0%
	(2258.8)		(615.1)		(t)				
F05-C2	708698	126	743173	757171	708369	563	1.88%	-4.68%	-0.05%
	(t)		(713.2)		(t)				
F10-C2	1029242	35	1084280	1121610	1028850	353	3.44%	-5.11%	-0.04%
	(t)		(832.1)		(t)				
F01-C8	505310	222	521468	517190	505884	1393	-0.82%	-2.99%	0.11%
	(t)		(1026.3)		(t)				
F05-C8	1112076	37	1191580	1169820	1137460	403	-1.83%	-4.54%	2.28%
	(t)		(1486.9)		(t)				
F10-C8	1847334	17	1985220	1893970	1839690	261	-4.60%	-7.33%	-0.41%
	(t)		(2300.7)		(t)				

TAB. 15 – Computational Results Simple Cutting-Plane R18 : 20,300,200

PROB	CPLEX	CPLEX NODES	PR	UB	BC	BC NODES	UBPR	GAP BCPR	BCCPLEX
F01-C1	828559 (t)	280	858043 (1926.3)	868545	831476 (t)	1575	1.22%	-3.10%	0.35%
F05-C1	1542197 (t)	2	1649730 (2279.5)	1586310	1575350 (t)	57	-3.84%	-4.51%	2.15%
F10-C1	2292684 (t)	0	2418190 (2167.1)	2354620	2331070 (t)	29	-2.63%	-3.60%	1.67%
F01-C2	921762 (t)	79	964367 (2209.1)	968326	924582 (t)	741	0.41%	-4.13%	0.30%
F05-C2	1866215 (t)	1	1980170 (2810.9)	X	X (t)	-	-	-	-
F10-C2	2895982 (t)	0	2928930 (3575.3)	2948790	2948790 (t)	45	0.68%	0.68%	1.82%
F01-C8	1485690.8 (t)	28	1568050 (8151.7)	1508100	1508100 (t)	137	-3.82%	-3.82%	1.51%
F05-C8	4010736.7 (t)	6	4553560 (15068.2)	4254450	4254260 (t)	101	-6.57%	-6.57%	6.07%
F10-C8	6663506.4 (t)	2	7639160 (26296.6)	6832200	6790440 (t)	97	-10.56%	-11.11%	1.90%

Chapitre 5

Conclusion

Dans cette thèse, nous nous sommes consacrés à l'étude du problème de conception de réseaux multiproduits, avec coût fixe et capacité. Nous nous sommes intéressés particulièrement aux méthodes polyédrales qui, de nos jours, semblent figurer parmi les méthodes les plus appropriées pour résoudre des problèmes combinatoires de grande taille. Ces méthodes visent essentiellement à dériver de meilleures formulations du problème moyennant l'ajout des informations supplémentaires, représentées par des inégalités valides (ou IV). Notre thèse s'est orientée alors dans cette direction de recherche. Ainsi, nous nous sommes attardés, essentiellement sur deux problématiques majeures des méthodes polyédrales. La première, qui est d'ordre théorique, consiste à identifier des formulations mathématiques qui définissent des ensembles de contraintes linéaires, ou inégalités valides. La deuxième, qui est d'ordre algorithmique, consiste à concevoir des procédures de générations d'IV violées et de les intégrer dans une méthode de résolution exacte.

Dans ce qui suit, nous présentons un résumé des principales contributions de cette thèse et nous dégageons ensuite des avenues de recherche qu'il serait intéressant d'explorer.

1 Principales contributions

Dans le chapitre 2, nous avons présenté une revue de la littérature portant sur les IV pertinentes au PCMC et qui ont été étudiées pour des problèmes apparentés à ce dernier. Dans un premier temps, nous avons introduit le modèle de conception

de réseau multiproduits, avec coût fixe et capacité et passé en revue les principaux travaux qui lui ont été dédiés. Ensuite, nous avons présenté trois grandes classes d'IV proposées dans la littérature qui ont été étudiées pour des structures particulières que nous pouvons identifier à des relaxations du PCMC. Nous avons dégagé les principales contributions des chercheurs portant sur les études polyédrales des modèles apparentés au PCMC. Nous avons présentés des résultats numériques obtenues en utilisant des implantations générales des IV proposées dans la littérature dans un logiciel sophistiqué. Ces résultats ont montré que ces implantations générales ne sont pas suffisantes pour résoudre des exemplaires de grande taille du PCMC, d'où la nécessité d'entreprendre une étude polyédrale du PCMC. Finalement, nous avons soulevé quelques questions importantes donnant naissance à des voies de recherche. Ce chapitre, qui se voulait être essentiellement une revue des propriétés polyédrales des modèles apparentés au PCMC, constituait en fait un premier jalon vers l'élaboration d'une étude poussée des propriétés polyédrales du PCMC.

Dans le chapitre 3, nous avons proposé une procédure de génération de coupes qui intègre des IV basées essentiellement sur des coupes du réseau. En premier, nous avons adapté d'une façon adéquate au PCMC des IV connues dans la littérature. Plus précisément, nous avons développé des procédures innovatrices de séparation et d'élévation séquentielle qui se sont avérées efficaces à identifier des IV violées. Nous avons également identifié de formulations mathématiques qui définissent de nouvelles familles d'IV pour la structure induite par des coupes du réseau, en général, donc pour le PCMC, en particulier. Ensuite, nous avons proposé des heuristiques visant l'identification de nouvelles structures de coupes du réseau qui sont d'ailleurs nécessaires à la génération des IV violées. Finalement, nous avons démontré l'utilité des IV étudiées et utilisées dans le cadre d'une méthode de coupes à dériver de meilleures formulations du problème, et donc à améliorer la qualité des bornes obtenues de ses relaxations, en un temps raisonnable.

Dans le chapitre 4, nous avons proposé une méthode de résolution exacte, un algorithme de Branch-and-Cut, qui intègre une méthode de coupes, basée sur des familles d'IV étudiées dans le chapitre 3, à un algorithme de Branch-and-Bound. La procédure

de génération de coupes violées est utilisée à chaque noeud de l'arbre d'énumération afin d'obtenir de meilleures formulations. Notons que les IV générées à un noeud sont valides à n'importe quel autre noeud de l'arbre d'énumération. C'est un aspect avantageux de l'algorithme car, d'une part, il rend la tâche de gérer les IV dans la formulation moins ardue, d'autre part, il permet l'exploitation de ces meilleures formulations au niveau de chaque noeud. Nous avons proposé et exploré une nouvelle stratégie de branchement basée sur des décisions polytomiques. Nous avons également proposé une heuristique pour l'identification de bornes supérieures sur l'objectif. Nous avons montré une utilisation judicieuse de différentes sources d'informations accumulées tout au long du processus de résolution. Ces informations ont été utilisées dans le choix des variables de branchement ainsi que dans l'heuristique de calcul de bornes supérieures.

Bien que nous avons dégagé à la fin du chapitre 4 plusieurs avenues de recherche visant à améliorer la performance de l'algorithme de Branch-and-Cut proposé, il constitue, néanmoins, un considérable jalon vers l'élaboration des méthodes efficaces de résolution pour résoudre des exemplaires de grande taille du PCMC.

2 Nouvelles avenues de recherche

Plusieurs avenues de recherche s'ouvrent à la suite de cette recherche. Notre but principal étant d'élaborer une méthode efficace pour résoudre des exemplaires de taille réelle du PCMC. Dans ce contexte, plusieurs recherches futures pourront porter sur l'amélioration de la performance de l'algorithme de Branch-and-Cut proposé dans cette thèse. Mentionnons principalement les idées suivantes.

- Le développement d'une stratégie mixte de branchement, basée sur une combinaison judicieuse des deux stratégies "depth-first" et "best-bound", pourrait réduire considérablement la taille de l'arbre d'énumération.
- Une amélioration de l'heuristique de calcul de bornes supérieures en utilisant des techniques de recherche local sophistiquées, telle la recherche avec Tabou, pourrait être très bénéfique.

- Le développement des heuristiques sophistiquées de génération de coupes du réseau, dans la perspective de maximiser la violation des classes d'IV basées sur ces coupes, entraînera nécessairement des améliorations substantielles à la formulation.

D'autres avenues de recherche portant sur des concepts plus généralisés s'ouvrent à la suite de cette thèse. Dans ce qui suit, nous citons quelques-unes.

- Suite à l'identification de nouvelles familles d'IV dans le chapitre 3, il serait intéressant d'examiner la dimension de celles-ci et d'étudier la possibilité d'une éventuelle amélioration en appliquant des techniques d'élévation séquentielle.
- Les inégalités étudiées dans cette thèse ont prouvé leur utilité à dériver de meilleures reformulations du problème en un temps raisonnable. Toutefois, les améliorations apportées n'étaient pas suffisantes pour obtenir de bornes inférieures serrées sur l'objectif. Dans ce contexte, un effort supplémentaire visant une meilleure compréhension des propriétés polyédrales du problème serait nécessaire pour resserrer les bornes.
- Les IV présentées et étudiées dans cette recherche sont valides pour une structure particulière présente dans presque tous les modèles de conception de réseaux : la structure induite par des coupes du réseau. Il serait pertinent alors d'examiner l'utilité de ces IV à améliorer d'autres problèmes de conception de réseaux.
- L'utilisation des IV générées dans une nouvelle approche algorithmique qui combine une méthode de relaxation lagrangienne avec un algorithme de Branch-and-Bound paraît très judicieuse. Cette approche consiste à introduire les IV violées dans l'objectif avec des termes de pénalités. L'avantage principale de cette approche étant de préserver la structure particulière des sous-problèmes et ainsi rendre le calcul des bornes au niveau de chaque noeud de l'arbre d'énumération plus efficace.