



HAL
open science

Contribution théorique et numérique à la résolution du problème du Voyageur de Commerce

Emmanuel Wild

► **To cite this version:**

Emmanuel Wild. Contribution théorique et numérique à la résolution du problème du Voyageur de Commerce. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 2003. Français. NNT: . tel-00005167

HAL Id: tel-00005167

<https://theses.hal.science/tel-00005167>

Submitted on 29 Feb 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Recherche Opérationnelle, Combinatoire et Optimisation

préparée au

Laboratoire Informatique et Distribution (UMR 5132)

dans le cadre de l'Ecole Doctorale

Mathématiques, Sciences et Technologies de l'Information, Informatique.

présentée et soutenue publiquement par

Emmanuel Wild

le 26 septembre 2003

**Contribution théorique et numérique
à la résolution du problème du Voyageur de Commerce**

Directeur de thèse : M. DENIS NADDEF

Composition du jury

M. :	Denis	TRYSTRAM	Président
MM. :	Jean	FONLUPT	Rapporteurs
	Thomas	LIEBLING	
M. :	Denis	NADDEF	Directeur de thèse
M. :	Denis	TRYSTRAM	Examineur

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué à l'aboutissement du travail de recherche que j'ai mené au sein du Laboratoire Informatique et Distribution durant les trois dernières années, à commencer par mon directeur de thèse, Denis Naddef. Mes pensées vont aussi à Pierre-François Dutot et Eiad Sulaiman, mes collègues de bureau, pour leur disponibilité et leur aide pour résoudre les petits problèmes quotidiens que j'ai pu rencontrer.

Mes remerciements vont aussi à ceux qui ont accepté de lire ce travail et d'y apporter leurs remarques et suggestions d'amélioration. J'espère en avoir tenu compte et proposé une thèse de meilleure qualité grâce à eux.

Je souhaite aussi remercier les personnes qui m'ont aidé à résoudre les problèmes techniques liés à l'utilisation de bibliothèques provenant d'origines diverses sur du matériel dont la configuration variait régulièrement, en particulier Joëlle Prévost, Grégory Mounié et Mauricio Pillon.

Plus généralement je remercie tous les membres du laboratoire au sein duquel j'ai effectué mes travaux pour l'aide et le soutien qu'ils ont pu m'apporter lorsque j'en avais besoin.

Enfin, ce travail n'aurait évidemment pu aboutir sans le soutien constant de ma femme, qui a vécu à mes côtés les bons et moins bons moments liés à mes années d'études doctorales.

Table des matières

1	Le problème du Voyageur de Commerce	7
1.1	Présentation du problème et de sa modélisation	7
1.1.1	Le problème du Voyageur de Commerce	7
1.1.2	Modélisation	7
1.1.3	Applications	9
1.2	Optimisation polyédrale	9
1.2.1	Rappels d’algèbre linéaire	9
1.2.2	Motivation	11
1.3	Méthode “Branch & Cut”	12
1.3.1	Principe	12
1.3.2	Implémentation utilisée	14
1.4	Travail de recherche	16
1.4.1	Séparation	16
1.4.2	Branchement	17
2	Cadre pour l’étude polyédrale du problème	19
2.1	Principales propriétés polyédrales du problème du Voyageur de Commerce	19
2.1.1	Notations	19
2.1.2	Propriétés	19
2.2	Les relaxations du problème	21
2.2.1	Relaxation monotone	21
2.2.2	Relaxation en chaînes hamiltoniennes	21
2.2.3	Relaxation graphique	22
2.3	Relaxation graphique et propriétés polyédrales	23
2.3.1	Notations et propriétés générales	23
2.3.2	D’une facette à une autre...	25
2.3.3	Ajout de sommets	27
3	Méthodes avancées de séparation	29
3.1	Principales classes de contraintes et leurs propriétés polyédrales	29
3.1.1	Contraintes d’élimination de sous-tours	29
3.1.2	Contraintes de peigne	30
3.1.3	Généralisations des contraintes de peigne	32

3.2	Une classe particulière de contraintes : les inéquations de dominos . . .	34
3.2.1	Définition et validité	34
3.2.2	Propriétés des cycles hamiltoniens serrés relativement à une inéquation de dominos	36
3.2.3	Méthode de démonstration utilisée	40
3.2.4	Ajout de deux dents minimales ou d'une dent fondamentale paire dans un demi-domino	41
3.2.5	Remplacement d'une dent minimale par une dent impaire fondamentale	45
3.2.6	Ajout de sommets	50
3.2.7	Le théorème résultat	54
3.2.8	Séparation des contraintes de dominos	54
3.3	Génération de facettes à partir du partitionnement du graphe en coupes minimum	55
3.3.1	Motivation et objectifs	55
3.3.2	Recherche de contraintes	56
3.3.3	Réduction de la taille du problème	56
3.3.4	Génération de facettes à l'aide des coupes minimum du graphe support	65
3.3.5	Transformation en contrainte valide pour le problème initial	72
3.3.6	Mise en œuvre informatique	73
3.3.7	Résultats	74
4	Méthodes avancées de branchement	79
4.1	Principe des méthodes de branchement	79
4.2	Branchement sur une variable	79
4.3	Branchement sur un ensemble de variables	81
4.4	Branchement sur des contraintes	82
4.5	Choix des paramètres de branchement	82
4.6	Résultats	83
5	Bilan et perspectives	87
5.1	Bilan	87
5.1.1	De nouveaux outils pour la phase de séparation	87
5.1.2	Une nouvelle méthode de branchement	88
5.2	Perspectives	89
5.2.1	Extension des résultats polyédraux pour les contraintes de dominos	89
5.2.2	Améliorations du branchement	90
5.2.3	Parallélisation de la résolution	90

Chapitre 1

Le problème du Voyageur de Commerce

1.1 Présentation du problème et de sa modélisation

1.1.1 Le problème du Voyageur de Commerce

Etant données n villes et les distances entre tout couple de villes, le problème du Voyageur de Commerce consiste à trouver un trajet de longueur minimum qui passe une et une seule fois par chaque ville et revient à son point de départ. On appelle *cycle hamiltonien* un cycle passant une et une seule fois par chaque ville. En d'autres termes, le problème consiste à trouver un cycle hamiltonien de longueur minimum.

Ainsi présenté, la description du problème sous-entend une localisation de chaque ville de façon géographique. On considère une définition plus globale du problème dans laquelle la distance entre deux villes est un entier positif quelconque. Les propriétés géographiques ne sont donc pas forcément respectées.

Il existe plusieurs variantes du problème. On choisit de travailler sur le problème du Voyageur de Commerce Symétrique, dans lequel la distance entre deux villes est indépendante du sens de parcours. Une solution réalisable du problème est un cycle hamiltonien.

Les instances étudiées dans le cadre de ma thèse sont issues de la bibliothèque TSPLIB [29] qui contient un certain nombre d'instances du problème du Voyageur de Commerce comprenant de 14 à 85900 villes. Certaines de ces instances sont définies dans le plan et chaque ville est alors repérée par ses coordonnées, d'autres instances sont définies de façon explicite par une matrice des distances inter-villes.

1.1.2 Modélisation

1.1.2.1 Notations

Ce problème peut être représenté par un graphe non orienté $G = (V, E)$ où V est l'ensemble des sommets et E l'ensemble des arêtes. Un sommet sera noté v et

une arête $e = (v_1, v_2)$. Chaque sommet représente une ville. Chaque arête (v_1, v_2) est caractérisée par sa longueur, distance entre les villes v_1 et v_2 . On peut supposer que G est un graphe complet, car il est possible de se ramener à ce cas en ajoutant entre deux sommets u et v non reliés une arête e ayant pour longueur la valeur d'un plus court chemin entre u et v .

Avant de traiter de la modélisation de ce problème, on définit quelques notations qui seront utilisées par la suite.

Etant donné un ensemble de sommets $S \subset V$, l'ensemble noté $\delta(S)$ est constitué des arêtes ayant une et une seule extrémité dans S , c'est-à-dire $\delta(S) = \{(u, v) \in E : u \in S, v \notin S\}$. On appelle l'ensemble $\delta(S)$ *cocycle* de S .

L'ensemble noté $\gamma(S)$ est constitué des arêtes ayant les deux extrémités dans S , c'est-à-dire $\gamma(S) = \{(u, v) \in E : u \in S, v \in S\}$.

Si S et T sont deux ensembles de sommets tels que $S \subset V$ et $T \subset V \setminus S$, on note $(S : T)$ ou $(T : S)$ l'ensemble d'arêtes ayant une extrémité dans S et l'autre dans T . Ainsi $(S : T) = \{(u, v) \in E : u \in S, v \in T\}$.

On note \mathbb{R}^E l'ensemble des vecteurs indexés par E . Les composantes d'un vecteur de \mathbb{R}^E sont en correspondance avec les éléments de E .

Si F est un sous-ensemble de E et $x \in \mathbb{R}^E$ un vecteur, on note $x(F)$ la somme $\sum_{e \in F} x_e$. En particulier, si S est un sous-ensemble de sommets et x le vecteur des capacités des arêtes, alors $x(\delta(S))$ peut être vu comme la valeur de la coupe définie par S .

1.1.2.2 Programme linéaire en nombres entiers

Il y a plusieurs façons de modéliser le problème par un programme linéaire en nombres entiers. On choisit d'associer à chaque arête $e = (u, v)$ une variable x_e (ou x_{uv}) qui prend la valeur 1 si e fait partie du cycle hamiltonien optimal et 0 sinon. Cette modélisation est appelée *modélisation à double indice* car chaque variable est décrite par les deux extrémités de l'arête qui lui est associée.

Le programme linéaire en nombre entiers s'écrit alors :

$$\min \sum_{e \in E} c_e x_e$$

$$\text{sous les contraintes : } \sum_{e \in \delta(v)} x_e \geq 2 \quad \text{pour } v \in V \quad (1)$$

$$\sum_{e \in \delta(S)} x_e \geq 2 \quad \text{pour } 2 \leq |S| \leq \frac{|V|}{2} \quad (2)$$

$$0 \leq x_e \leq 1 \quad \text{pour } e \in E \quad (3)$$

$$x_e \text{ entier} \quad \text{pour } e \in E \quad (4)$$

Les équations (1) sont appelées contraintes de degré. Elles traduisent le fait que chaque sommet n'est visité qu'une et une seule fois. Les inéquations (2) sont appelées

contraintes d'élimination de sous-tours (par abus de langage elles sont aussi appelées contraintes de sous-tours). Elles imposent que tout cycle hamiltonien entre et sort de tout sous-ensemble de sommets du graphe.

La formulation de ce problème est très simple et se prête bien à la résolution par un logiciel spécialisé dans la résolution de programmes linéaires en nombres entiers (Ilog CPLEX par exemple), appelé aussi solveur. Cependant l'utilisation d'un tel logiciel montre ses limites pour des instances de très faible taille (une dizaine de villes). La plupart des instances de référence ne peuvent donc pas être résolues de cette façon.

1.1.3 Applications

Ce problème a de nombreuses applications pratiques et sert de référence pour d'autres problèmes.

Un exemple d'application industrielle du problème du voyageur de commerce est le placement de composants électroniques sur des plaques. La détermination de l'ordre optimal de placement des composants avec retour au point de départ est nécessaire pour traiter consécutivement les plaques du même modèle.

Une autre application utilisée actuellement concerne la production de cartes génétiques. Chabrier et al. [6] utilisent les techniques du voyageur de commerce pour l'ordonnancement de marqueurs qui permettent de produire des cartes génétiques vraisemblables.

Le problème du voyageur de commerce peut aussi être considéré comme un sous-problème du problème de tournées de véhicules. Les méthodes de résolution développées dans ce chapitre peuvent être appliquées à ce dernier problème. Cependant les problèmes de tournées de véhicules comportent deux aspects : un aspect *parcours* avec des tournées à déterminer, et un aspect *capacité*, avec les contraintes portant sur les capacités des véhicules. Ce dernier aspect n'est pas pris en compte par les méthodes présentées ici dans le cadre du problème du Voyageur de Commerce, d'autres méthodes sont donc développées pour combiner les deux aspects. En pratique les difficultés de résolution des problèmes de Tournées de Véhicules sont plus importantes, il existe encore des instances du problème à 76 villes qui ne sont pas résolues de façon optimale, alors que des instances du problème du Voyageur de Commerce comportant de l'ordre du millier de villes sont désormais couramment résolues.

1.2 Optimisation polyédrale

1.2.1 Rappels d'algèbre linéaire

L'objet de cette section est de donner les éléments d'algèbre linéaire nécessaires à la compréhension des bases théoriques de la méthode de résolution appelée "Branch & Cut". Pour obtenir des démonstrations ou avoir plus de précisions, consulter [25].

Définition 1 Soit S un sous-ensemble de \mathbb{R}^n de cardinal fini. On appelle **combinaison convexe** de points de S tout vecteur $x \in \mathbb{R}^n$ tel qu'il existe un ensemble fini de points $\{x^i : i = 1 \dots t\}$ dans S et t réels $\lambda_i > 0$ tels que $\sum_{i=1}^t \lambda_i = 1$ et $x = \sum_{i=1}^t \lambda_i x^i$.

Définition 2 L'enveloppe convexe de S , notée aussi $\text{conv}(S)$ est alors définie par $\text{conv}(S) = \{x : x \text{ est combinaison convexe de points de } S\}$.

Théorème 3 (Minkowski-Weyl) Toute enveloppe convexe peut être décrite par un ensemble fini d'équations et d'inéquations linéaires.

Définition 4 Un ensemble $S \subseteq \mathbb{R}^n$ est **convexe** si pour tout couple $(x, y) \in S^2$ et tout réel $\lambda \in [0, 1]$, le point $\lambda x + (1 - \lambda)y$ appartient à S .

Définition 5 Un ensemble de points $\{x^i : i = 1 \dots k\}$ est **linéairement indépendant** si l'unique solution de l'équation $\sum_{i=1}^k \lambda_i x^i = 0$ d'inconnues $\lambda_i, i = 1 \dots k$, est $\lambda_i = 0, i = 1 \dots k$. L'ensemble de points est **affinement indépendant** si l'unique solution du système d'équations $\sum_{i=1}^k \lambda_i x^i = 0$ et $\sum_{i=1}^k \lambda_i = 0$ est $\lambda_i = 0, i = 1 \dots k$.

Définition 6 Un **polyèdre** $P \subseteq \mathbb{R}^n$ est l'ensemble des points qui satisfont un ensemble fini d'inéquations linéaires. En d'autres termes, étant donné une matrice A à n lignes et m colonnes et un vecteur $b \in \mathbb{R}^n$, l'ensemble $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ est un polyèdre.

Définition 7 Un polyèdre est **borné** s'il existe un réel $\omega > 0$ tel que $P \subseteq \{x \in \mathbb{R}^n : -\omega \leq x_j \leq \omega, j = 1 \dots n\}$. On appelle **polytope** un polyèdre borné.

Définition 8 Un polyèdre P est de **dimension** k si le nombre maximum de points affinement indépendants dans P est $k + 1$. La dimension d'un polyèdre P est notée $\text{dim}(P)$.

Définition 9 On appelle **inéquation valide** (respectivement **équation valide**) d'un polyèdre P toute inéquation linéaire de la forme $cx \leq c_0$ vérifiant $P \subseteq \{x \in \mathbb{R}^n : cx \leq c_0\}$ (respectivement $P \subseteq \{x \in \mathbb{R}^n : cx = c_0\}$).

Définition 10 Si $cx \leq c_0$ est une inéquation valide d'un polyèdre P , l'ensemble F défini par $F = \{x \in P : cx = c_0\}$ est appelé **face** de P et l'on dit que l'inéquation $cx \leq c_0$ **représente** F . Une face F est **propre** si $F \neq \emptyset$ et $F \neq P$. Si F est une face de P non vide, on appelle **support** de P l'inéquation $cx \leq c_0$.

Définition 11 Une face F d'un polyèdre est appelée **facette** si $\text{dim}(F) = \text{dim}(P) - 1$.

Proposition 12 Pour chaque facette F d'un polyèdre P , une des inéquations représentant F est nécessaire dans la description de P . Toute inéquation représentant une face F telle que $\text{dim}(F) < \text{dim}(P) - 1$ est inutile dans la description de P .

Théorème 13 *Un polyèdre P de dimension $n-k$, avec $k > 0$, est représenté par un ensemble de k équations qui forment un ensemble maximal d'équations linéairement indépendantes de P et d'une inéquation par facette.*

Il existe cependant une infinité de représentations pour une même facette. Le théorème 14 décrit la famille d'inéquations linéaires nécessaires à la représentation d'une facette.

Théorème 14 *Soient $(A^=, b^=)$ un système de p équations linéairement indépendantes décrivant un polyèdre $P \subseteq \mathbb{R}^n$ et soit $F = \{x \in P : cx = c_0\}$ une face propre de P . Les deux assertions suivantes sont équivalentes :*

1. F est une facette de P
2. si $fx = f_0$ pour tout x vérifiant $cx = c_0$, alors il existe $\alpha \in \mathbb{R}$ et $\lambda \in \mathbb{R}^p$ tels que $(f, f_0) = (\alpha c + \lambda A^=, \alpha c_0 + \lambda b^=)$

1.2.2 Motivation

Comme on l'a vu précédemment, le programme linéaire en nombres entiers permettant de modéliser le problème du Voyageur de Commerce ne peut en pratique être résolu directement par un solveur. Si on note x^H le vecteur d'incidence d'un cycle hamiltonien H , l'objectif du problème est d'effectuer la minimisation suivante :

$$\min\{cx^H : H \text{ cycle hamiltonien}\}$$

Or cette minimisation est équivalente à :

$$\min\{cx : x \in \text{conv}\{x^H : H \text{ cycle hamiltonien}\}\}$$

Or l'ensemble $\text{conv}\{x^H : H \text{ cycle hamiltonien}\}$ est un polytope, aussi appelé **polytope associé au problème du Voyageur de Commerce**. On le note $STSP(n)$ pour une instance à n villes. D'après la section précédente, ce polyèdre peut être décrit par un ensemble d'équations et d'inéquations linéaires. Il est donc alors possible d'appliquer les techniques de programmation linéaire pour résoudre ce problème. La résolution de ce problème se fera donc sur le polyèdre associé et on parle d'*optimisation polyédrale*.

Le polytope $STSP(n)$, bien que décrit par un ensemble fini d'équations et d'inéquations linéaires, est très grand (le nombre de classes de facettes est exponentiel) et n'est pas entièrement connu. La modélisation du problème par le programme linéaire en nombres entiers présenté précédemment permet de déduire des inéquations linéaires définissant des faces du polytope : il s'agit des contraintes de degrés, de sous-tours et de bornes sur les variables. Cependant, du fait des contraintes imposant une valeur entière aux variables, la description du polytope par un ensemble d'équations et d'inéquations linéaires est bien plus complexe. Cet ensemble, bien que fini, a une taille exponentielle et n'est pas entièrement connu.

L'objectif du travail théorique effectué sur le polyèdre $STSP(n)$ consiste donc à trouver de nouvelles classes de contraintes définissant des facettes de ce polytope, de façon à obtenir une description plus complète de $STSP(n)$. Cependant le nombre de contraintes définissant le polytope étant très important, il est inenvisageable en pratique de résoudre un programme linéaire comportant toutes les contraintes connues. On utilisera donc comme point de départ une relaxation du programme linéaire en nombres entiers. Cette relaxation est obtenue en ôtant du programme linéaire en nombres entiers les contraintes imposant une valeur entière aux variables (contraintes (4)) et les contraintes d'élimination de sous-tours (contraintes (2)). Une solution optimale x^* de cette relaxation peut être solution réalisable du problème, auquel cas l'optimisation prend fin, ou peut être située en-dehors du polyèdre $STSP(n)$. Dans ce cas il est nécessaire d'ajouter à la relaxation des contraintes définissant des facettes de $STSP(n)$ non satisfaites par x^* . La recherche de telles contraintes constitue un objectif pratique pour la résolution du problème.

1.3 Méthode “Branch & Cut”

1.3.1 Principe

La méthode de résolution appelée “Branch & Cut” est une méthode exacte. L'idée de base est d'utiliser une relaxation du programme linéaire en nombres entiers étudié dans la section précédente. Pour cela, on considère le programme linéaire sans les contraintes (2) et (4) imposant aux variables une valeur entière. Ainsi ce programme linéaire peut se résoudre par une méthode utilisant le simplexe, en un temps très faible en pratique. La solution d'un tel programme relaxé ne sera cependant pas forcément un vecteur d'entiers, mais un vecteur de fractions. La valeur de la solution est inférieure ou égale à la valeur optimale du programme linéaire en nombres entiers.

La solution du programme linéaire relaxé peut être entière. Dans ce cas il s'agit de la solution optimale du problème. Si en revanche la solution est un vecteur de fractions non entier, elle est située à l'extérieur du polytope P défini par l'enveloppe convexe des vecteurs représentant des cycles hamiltoniens. Il existe alors une inéquation définissant une facette de P qui est violée par cette solution fractionnaire. Le but de la méthode “Branch & Cut” est d'ajouter au programme linéaire relaxé courant de telles contraintes, aussi appelées **coupes**.

En pratique la génération de coupes ne suffit pas toujours à résoudre le problème. Plusieurs raisons expliquent cette difficulté. D'une part les familles de contraintes définissant des facettes du polytope associé au problème du Voyageur de Commerce ne sont pas toutes connues. On ne dispose donc pas d'une connaissance complète du polytope, et alors on ne connaît pas de contrainte pouvant séparer une solution fractionnaire donnée du polytope associé au problème. D'autre part, et il s'agit du problème principal, les algorithmes dont on dispose pour trouver une contrainte violée appartenant à une famille connue ne sont pas exacts. Ils sont basés sur des

heuristiques, donc il arrive fréquemment qu’il existe une contrainte violée appartenant à une famille connue mais non trouvée par les algorithmes dont on dispose.

Dans le cas où la génération de coupes est insuffisante, un branchement est effectué. Le problème est séparé en deux sous-problèmes sur lesquels la même méthode est appliquée. Les deux sous-problèmes correspondent à deux jeux de solutions complémentaires. Il existe plusieurs méthodes de branchement, mais l’idée de base consiste en l’ajout d’une inéquation linéaire dans chaque sous-problème. Ainsi par exemple un branchement sur une variable e consistera en l’ajout de la contrainte $x_e \leq 0$ dans un sous-problème et $x_e \geq 1$ dans l’autre. Plus généralement un sous-problème respectera une contrainte $cx \leq c_0$, où c est un vecteur d’entiers et c_0 un entier, et l’autre sous-problème respectera la contrainte $cx \geq c_0 + 1$.

Ainsi le fonctionnement général de la méthode “Branch & Cut” peut se décrire de la façon suivante :

1. Résolution du programme linéaire relaxé. Si la solution est entière, arrêt, sinon passer au point 2.
2. Génération de coupes : si des coupes sont trouvées, les ajouter au programme linéaire relaxé et revenir au point 1, sinon passer au point 3.
3. Branchement : séparer le problème en deux sous-problèmes par l’ajout d’une contrainte au programme linéaire relaxé. Pour chaque sous-problème, repartir du point 1.

La résolution se base donc sur une exploration arborescente. On appelle arbre de résolution l’arbre des sous-problèmes générés. En pratique l’exploration arborescente peut s’effectuer de différentes manières. Lorsqu’un branchement est effectué, la valeur de la fonction objectif du dernier programme linéaire obtenu est une borne inférieure de la valeur du sous-problème correspondant. L’exploration par “meilleur d’abord” utilise cette valeur pour résoudre d’abord les sous-problème de la branche ayant la borne inférieure la plus faible. L’exploration d’une branche prend fin dans deux cas :

- la solution du programme linéaire courant est un vecteur d’entiers définissant un cycle hamiltonien réalisable.
- la différence entre la valeur du meilleur cycle hamiltonien trouvé jusqu’à présent et la valeur de la fonction objectif du programme linéaire est strictement inférieure à 1

Afin de rendre cette méthode efficace lorsque l’on traite un grand nombre de variables, comme dans le cas du problème du voyageur de commerce où une variable est associée à chaque arête du graphe complet, on utilise une extension de cette méthode appelée “Branch & Cut & Price”. Au lieu de considérer dans le programme linéaire relaxé toutes les variables, on considère un sous-ensemble de variables. Ces variables sont initialement choisies selon un critère de proximité, c’est-à-dire que pour chaque sommet on choisit les variables correspondant aux k plus proches voisins, avec $3 \leq k \leq 10$. Afin d’être sûr d’obtenir un programme linéaire réalisable, on ajoute à cet ensemble les variables correspondant aux arêtes d’un cycle hamiltonien. Ce sous-ensemble de variables est modifié régulièrement afin d’intégrer les

variables qui permettent une diminution de la fonction objectif. Toutes les variables du problèmes sont alors analysées (phase de *pricing*) et celles qui permettent une diminution de la fonction objectif (par le calcul du coût réduit) sont ajoutées au programme linéaire courant. Si on désigne par a_{ie} le coefficient de la variable e dans la contrainte i , si c_e désigne la longueur de l'arête e et si y_i désigne la variable duale correspondant à la contrainte i , une arête e^* non présente dans la programme linéaire courant est ajoutée si et seulement si $c_j - \sum_i a_{ie}y_i < 0$. De la même façon le calcul du coût réduit peut permettre de savoir si certaines arêtes sont présentes dans la solution finale ou non, et ainsi de fixer la valeur de la variable associée, puis, par implication logique (tous les sommets ont un degré égal à 2), d'autres variables peuvent être fixées.

1.3.2 Implémentation utilisée

1.3.2.1 Bibliothèque ABACUS

Du point de vue pratique, la partie “Branch & Cut” du problème du Voyageur de Commerce est implémentée dans ABACUS, une bibliothèque de fonctions écrites en C++ implémentant les bases d'une résolution par la méthode Branch & Cut. Cette bibliothèque a été initialement réalisée par Stefan Thienel [32] et en est actuellement à sa version 2.3. Une introduction à ABACUS peut être trouvée dans [34].

Le principe d'ABACUS est d'implémenter la boucle de résolution commune à tout algorithme de type Branch & Cut et de définir les spécifications des fonctions que l'utilisateur adapte à son propre problème. Ainsi il existe une classe définissant les variables et une classe définissant les contraintes. L'utilisateur dérive ces classes pour créer ses propres variables et ses propres contraintes. Ainsi une variable est définie par les deux extrémités de l'arête correspondante. Chaque type de contrainte est défini par une classe. La procédure de séparation est aussi définie par l'utilisateur. Ainsi ABACUS permet de définir des procédures correspondant à chaque étape de la méthode Branch & Cut sans avoir à programmer le déroulement de la méthode en tant que tel. Un exemple de résolution du problème de Voyageur de Commerce à l'aide d'ABACUS peut être trouvé dans [33].

ABACUS permet l'ajout de classes de contraintes de manière assez aisée. En effet il suffit de créer une classe de contraintes dérivée de la classe-mère ABACUS, dans laquelle le membre de droite est spécifié ainsi que le sens de la contrainte. Les coefficients de la contrainte sont définies par une méthode, de telle sorte qu'il n'est pas nécessaire de stocker tous les coefficients entre tous les couples de sommets (cela utilise trop de mémoire). A chaque classe de contraintes correspond une classe C+++. Ainsi les contraintes de degré, d'élimination de sous-tours, de peigne, d'arbres de cliques ainsi que de nombreuses autres sont définies.

La librairie ABACUS doit être couplée à un solveur de programmes linéaires pour pouvoir fonctionner. Plusieurs solveurs sont gérés par ABACUS afin de permettre une utilisation transparente, sans avoir à connaître le mode de fonctionnement de chaque solveur.

1.3.2.2 Variables globales

La bibliothèque ABACUS gère un certain nombre de variables globales qui permettent de gérer le processus itératif et de branchement de la méthode Branch & Cut. On distingue deux catégories de variables globales : les variables constitutives de la méthode et les variables d’ajustement de calcul, non nécessaires théoriquement mais permettant un meilleur contrôle du processus de résolution.

On détaille à présent les variables constitutives de la méthode Branch & Cut.

- Borne supérieure globale (GUB) : contient la valeur du meilleur cycle hamiltonien trouvé jusqu’à présent
- Borne inférieure globale (GLB) : contient le minimum des valeurs de la objectif obtenues sur l’ensemble des nœuds actifs de l’arbre de résolution. Dès que $GLB > GUB - 1$, la valeur GUB est la solution optimale du problème.

Les principales variables permettant un contrôle du processus itératif sont les suivantes :

- Fréquence de “pricing” : comme le “pricing” nécessite l’analyse de toutes les variables non présentes dans le programme linéaire courant, il est coûteux. On réalise donc cette opération à une fréquence définie au départ et non après chaque résolution de programme linéaire.
- Fréquence d’amélioration de GUB : la borne supérieure du problème correspond au meilleur cycle hamiltonien trouvé jusqu’à présent. Un cycle est recherché avant de résoudre le premier programme linéaire afin d’obtenir une première borne. Cette borne est améliorée à partir de la solution courante du programme linéaire à une certaine fréquence.
- Fréquence et valeur de “tailing off”. Le “tailing off” consiste à forcer l’étape de branchement même si la séparation permet encore de trouver des contraintes. En effet il arrive que l’étape de séparation aboutisse à la génération de contraintes qui ne font que très peu progresser la valeur de la fonction objectif. Dans ce cas, un branchement est effectué. La fréquence du contrôle de progression de la valeur de la fonction objectif peut être définie par l’utilisateur de même que le pourcentage de progression minimum entre deux vérifications.

1.3.2.3 Solveur de programmes linéaires

Le solveur de programmes linéaires utilisé dans le code développé au sein du laboratoire ID est CPLEX, dans sa version 7.5. Il existe deux raisons majeures à l’utilisation de CPLEX plutôt que d’un autre solveur. D’une part l’interface entre ABACUS et CPLEX est fournie d’origine, de telle sorte que l’utilisation de ce solveur se fait de manière transparente pour l’utilisateur. D’autre part CPLEX est éprouvé et est devenu une référence dans le domaine de la résolution de programmes linéaires, de telle sorte qu’en pratique aucun problème lié au solveur ne se présente.

1.3.2.4 Méthodes de séparation

L'étape de séparation se fait à l'aide de la solution fractionnaire courante, stockée sous forme d'un tableau, et du graphe, défini dans une classe particulière. Ces deux données permettent d'appliquer les algorithmes de séparation, qui génèrent les contraintes sous un format adapté, afin de pouvoir les ajouter à l'ensemble des contraintes existantes. Une contrainte non active dans un certain nombre de programmes linéaires relaxés successifs (dont le membre de droite n'est pas atteint) est supprimée.

1.3.2.5 Méthodes de branchement

La méthode de branchement implémentée par défaut est basée sur la valeur des variables dans le dernier programme linéaire relaxé. L'utilisateur définit un nombre de variables k qu'il souhaite tester pour réaliser le branchement. L'algorithme implémenté dans ABACUS choisit k variables dont la valeur est proche de 0.5, puis teste la valeur de la fonction objectif des deux sous-problèmes générés par un branchement sur cette variable, en résolvant le problème par l'ajout de la contrainte de branchement. La variable e^* la plus prometteuse selon les critères définis par l'utilisateur, est alors sélectionnée pour générer deux sous-problèmes, un dans lequel elle est fixée à 0, l'autre dans lequel elle est fixée à 1.

Cette méthode de branchement a un inconvénient majeur : le problème est déséquilibré du côté $x_{e^*} = 0$, car fixer une variable à 1 fait bien plus avancer la résolution du problème que de la fixer à 0. En effet toute solution réalisable du problème est un vecteur comportant n composantes ayant une valeur de 1 et $O(n^2)$ composantes ayant une valeur à 0. C'est pourquoi d'autres méthodes de branchement ont été imaginées, comme choisir un ensemble de sommets dont le cocycle est proche d'un nombre impair ou choisir un ensemble de variables de valeurs proches de 0. ABACUS permet l'ajout de méthodes de branchement de manière très simple, grâce à la redéfinition de la méthode utilisée par défaut.

1.4 Travail de recherche

Le contexte de résolution d'un problème de type voyageur de commerce étant cerné, on peut préciser à présent dans les grandes lignes en quoi a consisté mon travail de recherche. Ce travail avait pour but d'étudier et de proposer des nouvelles méthodes de séparation et de branchement afin de pouvoir résoudre plus rapidement le problème. On précise en quoi de telles méthodes peuvent permettre d'espérer une amélioration de la résolution du problème.

1.4.1 Séparation

Comme on l'a vu précédemment, lorsque la solution optimale du programme linéaire n'est pas une solution entière, elle se situe en-dehors du polyèdre associé

au problème. La séparation consiste à ajouter des contraintes valides pour toute solution réalisable du problème mais non satisfaites par la solution courante du programme linéaire relaxé. En d'autres termes, la séparation a pour but de renforcer la relaxation.

La principale méthode de séparation utilisée actuellement est la séparation par paradigme. L'étude polyédrale du problème du Voyageur de Commerce a permis (et permet encore) de trouver des familles de contraintes définissant des facettes du polytope associé. Pour chaque famille, on dispose d'algorithmes, basés sur une recherche heuristique, permettant de trouver des contraintes appartenant à cette famille et non satisfaites par la solution du programme linéaire relaxé. Ainsi la séparation par paradigme a pour but de rechercher des contraintes qui appartiennent à des familles de facettes connues et qui permettent de renforcer la relaxation. La découverte de nouvelles familles de contraintes, ainsi que d'algorithmes de séparation pour ces familles, permet alors de faire progresser l'étape de séparation.

Il est cependant aussi possible de renforcer la relaxation à l'aide de contraintes qui n'appartiennent pas à des familles connues de facettes. Mes travaux de recherche ont permis de développer une méthode de recherche de telles contraintes définissant des facettes du polytope associé au problème, mais dont on ne connaît pas de famille, dans la continuité des travaux menés par Applegate et al. [1]. Ces contraintes ne sont pas issues de l'étude polyédrale du problème et ne peuvent pas être rattachées à des familles, mais permettent de renforcer la relaxation.

Un renforcement du programme linéaire relaxé par l'ajout de contraintes définissant des facettes du polytope associé permet de faire progresser la valeur de la fonction objectif, donc on peut espérer que l'étape de branchement sera plus courte. En effet le branchement implique la résolution de deux sous-problèmes de dimension inférieure au problème initial seulement d'une unité.

1.4.2 Branchement

Le branchement a souvent l'inconvénient d'être déséquilibré. L'arbre binaire associé à la résolution a alors souvent l'inconvénient d'être profond et touffu d'un seul côté, signe qu'une des contraintes permettant d'obtenir un nouveau sous-problème n'est pas forte et n'apporte pas grand chose à la résolution. Si l'arbre de résolution était plus équilibré on pourrait espérer un nombre de branches plus faible et une résolution plus efficace. Le but du travail effectué est donc de trouver de nouvelles méthodes de branchement permettant une résolution plus efficace du problème.

Chapitre 2

Cadre pour l'étude polyédrale du problème

Cette section présente de nombreux outils utiles pour l'étude polyédrale du problème du Voyageur de Commerce. Les principales relaxations du problème sont présentées, dont en particulier les liens entre la relaxation que nous avons utilisée et l'étude du polytope associé au problème.

2.1 Principales propriétés polyédrales du problème du Voyageur de Commerce

2.1.1 Notations

On note $K_n = (V, E)$ le graphe complet à n sommets, où V désigne l'ensemble des sommets et E l'ensemble des arêtes de ce graphe.

On note $STSP(n)$ le polytope associé au problème du Voyageur de Commerce Symétrique à n villes.

On note x^T le vecteur d'incidence associé à un cycle hamiltonien T .

On dit qu'un cycle hamiltonien est serré relativement à une inéquation linéaire $fx \geq f_0$ si $fx^T = f_0$.

On note \mathcal{H}_n l'ensemble des cycles hamiltoniens de K_n . Etant donné une inéquation $fx \geq f_0$, on note \mathcal{H}_f^- l'ensemble des cycles hamiltoniens serrés relativement à f .

2.1.2 Propriétés

On donne ici quelques propriétés du polytope $STSP(n)$. On cherche à présent à établir la dimension de ce polytope. Pour cela, on utilise le lemme suivant.

Lemme 15 *On considère un graphe complet à n sommets K_n .*

- *Si $n = 2k + 1$, alors on peut partitionner l'ensemble E des arêtes de K_n en k cycles hamiltoniens disjoints (au sens des arêtes).*

- Si $n = 2k$, alors on peut partitionner l'ensemble E des arêtes de K_n en $k - 1$ cycles hamiltoniens disjoints (au sens des arêtes) et un couplage parfait.

Démonstration Supposons $n = 2k + 1$. On exhibe alors k cycles hamiltoniens disjoints $\Gamma_1, \dots, \Gamma_k$. On note $m[n]$ le reste de la division entière de m par n . Le cycle Γ_i est construit de la façon suivante : $\Gamma_i : 1, 1 + i, 1 + 2i[n], \dots, 1 + (n - 1)i[n], 1$. Les arêtes utilisées sont toutes disjointes.

Dans le cas où $n = 2k$, on considère le graphe K_n . Le sous-graphe complet K'_{n-1} constitué des sommets 1 à $n - 1$ admet un partitionnement en $k - 1$ cycles hamiltoniens disjoints que l'on vient d'exhiber précédemment. En remplaçant dans chacun de ces cycles l'arête $(1 + i, 1 + 2i[n - 1])$ par les deux arêtes $(1 + i, n)$ et $(1 + 2i[n - 1], n)$, on obtient $k - 1$ cycles hamiltoniens sur K_n disjoints au sens des arêtes. Les arêtes restantes forment un couplage de K_n . En effet chaque sommet est incident à $n - 1$ arêtes de K_n , dont $2(k - 1) = n - 2$ appartiennent à des cycles hamiltoniens. Par conséquent chaque sommet de K_n est incident à une et une seule arête, donc ces arêtes forment un couplage parfait de K_n . \square

Théorème 16 *La dimension du polytope $STSP(n)$ est $\frac{n(n-3)}{2}$.*

Démonstration Un graphe complet à 3 sommets ne contient qu'un seul cycle hamiltonien. Par conséquent $\dim(STSP(3)) = 0$. La proposition est donc vraie pour $n = 3$.

Supposons que la proposition soit vraie à l'ordre n . Montrons qu'elle est vraie à l'ordre $n + 1$.

On développe ici le cas où $n + 1 = 2k + 2$, donc $n + 1$ est pair.

D'après le lemme 15, il existe k cycles hamiltoniens disjoints au sens des arêtes $\Gamma_1, \dots, \Gamma_k$ sur le sous-graphe complet à $2k + 1$ sommets K_{2k+1} . Pour chaque cycle Γ_i , on crée $2k + 1$ cycles de K_{2k+2} en enlevant tour à tour une arête $(u, v) \in \Gamma_i$ pour la remplacer par deux arêtes $(u, 2k + 2)$ et $(v, 2k + 2)$. On crée ainsi $\frac{n^2 - n}{2}$ cycles hamiltoniens. Montrons que ces cycles sont linéairement indépendants. Si on considère la matrice d'incidence des cycles construits en regroupant les arêtes par le cycle hamiltonien auquel elles appartiennent et que les arêtes incidentes au sommet $n + 1$ sont regroupées dans les dernières colonnes, la sous-matrice constituée des colonnes correspondant aux arêtes non incidentes au sommet $n + 1$ est carrée, à $\frac{n(n-1)}{2}$ lignes et colonnes, et de plus carrée par bloc. Chaque bloc est triangularisable et possède un déterminant non nul, donc la matrice est de rang $\frac{n(n-1)}{2}$. Par conséquent la dimension du polytope est au moins égale à $\frac{n(n-1)}{2} - 1 = \frac{(n+1)(n-2)}{2}$. Or comme il existe $n + 1$ contraintes d'égalité reliant les arêtes entre elles, la dimension du polytope est au maximum égale à $\frac{n(n+1)}{2} - (n + 1) = \frac{(n+1)(n-2)}{2}$. D'où le résultat.

Dans le cas où $n + 1 = 2k + 1$, on reprend le même raisonnement en considérant $k + 1$ cycles hamiltoniens disjoints de K_{2k} . Ils sont constitués de k cycles hamiltoniens obtenus en appliquant le lemme 15 et d'un cycle hamiltonien obtenu en complétant

le couplage parfait issu de ce même lemme. 15. \square

De ce qui précède on peut déduire la proposition suivante :

Corollaire 17 *Les seules équations satisfaites par tous les cycles hamiltoniens sur un graphe complet sont les combinaisons linéaires d'équations de degré.*

Une facette du polytope associé au problème du Voyageur de Commerce a donc une dimension égale à $\frac{n(n-1)}{2} - n - 1$ où n est le nombre de sommets. Afin qu'une inéquation définisse une facette de $STSP(n)$, il est nécessaire et suffisant qu'il existe $\frac{n(n-1)}{2} - n$ cycles hamiltoniens affinement indépendants vérifiant cette contrainte à l'égalité.

2.2 Les relaxations du problème

Comme l'étude de polyèdres de pleine dimension se révèle plus simple que celle de polyèdres de dimensions inférieures, l'idée des relaxations du problème est d'étudier un polyèdre issu d'une relaxation du problème, de pleine dimension ou presque, dont $STSP(n)$ est une face. Dans le cadre de notre étude, la relaxation graphique sera utilisée, mais les relaxations monotones et en chaînes hamiltoniennes sont brièvement évoquées ici, afin de montrer la diversité des méthodes que l'on peut utiliser pour étudier le problème.

2.2.1 Relaxation monotone

On considère le polyèdre constitué de l'enveloppe convexe des sous-ensembles d'arêtes de cycles hamiltoniens (tous les sous-ensembles d'arêtes pouvant se compléter en cycle hamiltonien). On note ce polytope $MTSP(n)$. Le polytope $STSP(n)$ est bien une face de $MTSP(n)$, car $\sum_{e \in E} x_e \leq n$ est une inéquation valide pour la relaxation. Par conséquent, $STSP(n)$ est la face définie par $\sum_{e \in E} x_e = n$.

Théorème 18 *Le polytope $MTSP(n)$ est de pleine dimension.*

Démonstration Pour démontrer cette propriété, il faut et il suffit d'exhiber $m+1$ vecteurs d'arêtes affinement indépendants appartenant à $MTSP(n)$. On considère le vecteur nul et les m éléments $\Gamma_i \in MTSP(n)$ définis par $\Gamma_i = \{e_i\}$, où e_i sont les différentes arêtes du graphe. Ces vecteurs sont affinement indépendants donc le polytope $MTSP(n)$ est de pleine dimension. \square

2.2.2 Relaxation en chaînes hamiltoniennes

Cette relaxation part du constat qu'il existe une bijection entre les cycles hamiltoniens de K_n et les chaînes hamiltoniennes de K_{n+1} . On appelle chaîne hamiltonienne tout chemin pouvant se compléter en cycle hamiltonien par l'ajout d'une

arête. On considère le polytope $CTSP(n)$ défini par l'enveloppe convexe des chaînes hamiltoniennes de K_n .

Théorème 19 *Le polytope $CTSP(n)$ est de dimension $\frac{n(n-1)}{2} - 1$.*

Démonstration Ce polytope n'est pas de pleine dimension, mais presque. Pour montrer la propriété énoncée, il faut et il suffit de montrer que la description de ce polytope en équations et inéquations linéaires ne comporte qu'une seule équation. Cette équation est $\sum_{e \in E} x_e = n - 1$. Supposons qu'il existe une autre équation satisfaite par tous les éléments du polytope $CTSP(n)$. Cette équation s'écrit $\sum_{e \in E} f_e x_e = f_0$. On peut fixer f_0 , car un coefficient multiplicatif permet de lui donner n'importe quelle valeur. On fixe donc $f_0 = n - 1$. Soient $e = (u, v)$ et $e' = (u', v')$ deux arêtes du graphe. On considère une chaîne hamiltonienne Γ ayant pour extrémités u et v et comprenant l'arête e . Comme le graphe est complet, une telle chaîne existe. On considère la chaîne hamiltonienne $\Gamma' = \Gamma \setminus \{e\} \cup \{e'\}$. Ces deux chaînes vérifient l'équation à égalité. On en déduit donc $f_e = f_{e'}$. Par conséquent, il existe une constante c telle que, pour tout $e \in E$, $f_e = c$. Comme toute chaîne hamiltonienne comprend exactement $n - 1$ arêtes, on en déduit que $c = 1$. Ainsi la seule équation satisfaite par tous les éléments du polytope $CTSP(n)$ est $\sum_{e \in E} x_e = n - 1$. Par conséquent la description de $CTSP(n)$ par des équations et inéquations linéaires ne comprend qu'une équation, et ce polytope est de dimension $\frac{n(n-1)}{2} - 1$. \square

2.2.3 Relaxation graphique

La relaxation graphique est devenue la relaxation la plus fréquemment utilisée pour l'étude polyédrale du problème du Voyageur de Commerce. En effet il s'agit de la relaxation permettant des démonstrations aisées et ayant l'avantage de mener à un polytope de pleine dimension. Cette relaxation a été introduite par Cornuéjols et al. [8]. Des résultats plus complets que ceux mentionnés dans ce chapitre peuvent être trouvés dans [19].

On appelle **tour** (ou parcours fermé) d'un graphe G toute famille W d'arêtes tel que le graphe induit par W soit connexe et chaque sommet soit incident à un nombre pair strictement positif d'arêtes. A chaque tour W est associé un vecteur x^W indexé par l'ensemble des arêtes tel que x_e^W soit égal au nombre de fois où l'arête e est dans la famille W . Ainsi avec cette définition tout cycle hamiltonien est un tour particulier. On note $GTSP(n)$ l'enveloppe convexe des vecteurs représentatifs des tours sur K_n . Par la suite l'ensemble des tours sur K_n sera noté \mathcal{W}_n et l'ensemble des tours serrés relativement à une inéquation $fx \geq f_0$ sera noté \mathcal{W}_f^- .

Théorème 20 *Le polyèdre $GTSP(n)$ est de pleine dimension.*

Démonstration Afin de démontrer ce résultat, il est nécessaire et suffisant d'exhiber $|E| + 1$ tours affinement indépendants appartenant à $GTSP(n)$. Soit W un

tour quelconque de $GTSP(n)$. Pour chaque arête $e \in E$, on considère le tour $W_e = W \cup 2\{e\}$. Les $|E| + 1$ tours que l'on vient de définir sont affinement indépendants car les vecteurs associés à $W \cup 2\{e\} \setminus W$ sont linéairement indépendants. D'où le résultat. \square

Par la suite la relaxation graphique sera utilisée pour démontrer certaines propriétés polyédrales du polytope $STSP(n)$. Afin de pouvoir démontrer ces propriétés, il est nécessaire d'établir un lien plus concret entre les facettes de $GTSP(n)$ et les facettes de $STSP(n)$.

2.3 Relaxation graphique et propriétés polyédrales

Dans cette section on explicite les liens entre les polyèdres $GTSP(n)$ et $STSP(n)$. En effet sous certaines conditions une inéquation définissant une facette de $GTSP(n)$ peut aussi définir une facette de $STSP(n)$. Nous verrons par la suite qu'en pratique il est plus facile de montrer qu'une inéquation définit une facette de $GTSP(n)$ qu'une facette de $STSP(n)$. A l'aide des outils définis dans cette section, nous disposerons de moyens pour passer d'une facette à une autre. Les résultats présentés ici peuvent être trouvés sous forme plus détaillées dans [19].

2.3.1 Notations et propriétés générales

Tout d'abord, nous pouvons remarquer que toute inéquation valide de $GTSP(n)$ comporte uniquement des coefficients positifs. En effet, soit $fx \geq f_0$ une inéquation valide sur $GTSP(n)$. Supposons qu'il existe $e \in E$ tel que $f_e < 0$. Soit Γ un tour contenant l'arête e et vérifiant $fx^\Gamma = f_0$. Alors $\Gamma' = \Gamma + 2e$ est aussi un tour qui vérifie $fx^{\Gamma'} < f_0$, ce qui est en contradiction avec la validité de l'inéquation. Si $f_0 < 0$, il est possible d'ajouter un nombre suffisant d'équations de degrés pour obtenir une inéquation équivalente qui satisfait $f_0 > 0$.

Définition 21 *Un cycle hamiltonien Γ est serré relativement à une inéquation $cx \geq c_0$ si son vecteur d'incidence vérifie cette inéquation à égalité : $cx^\Gamma = c_0$*

Définition 22 *Un ensemble S est serré relativement à un cycle hamiltonien Γ si $|\Gamma \cap \delta(S)| = 2$.*

Ces notions d'ensembles ou de cycles hamiltoniens serrés ont leur importance dans le cadre d'une démonstration de propriété polyédrale, car une identification d'une facette de $STSP(n)$ repose sur le fait que $|E| - |V|$ cycles hamiltoniens linéairement indépendants sont serrés relativement à l'inéquation la définissant.

Définition 23 *Pour toute inéquation $fx \geq f_0$ définie sur \mathbb{R}^E et pour tout sommet $u \in V$, on définit le sous-ensemble d'arêtes $\Delta_f(u)$ de la façon suivante :*

$$\Delta_f(u) = \{(v, w) \in E \mid v \neq u, w \neq u, f_{vw} = f_{uv} + f_{uw}\}$$

La définition de cet ensemble permet d'aborder la notion d'inéquation triangulaire serrée.

Définition 24 Une inéquation $fx \geq f_0$ définie sur \mathbb{R}^E est appelée triangulaire serrée lorsque les conditions suivantes sont satisfaites :

- Les coefficients f_e satisfont l'inégalité triangulaire, c'est-à-dire $f_{uv} \leq f_{uw} + f_{wv}$ pour tout triplet u, v, w de sommets de V
- L'ensemble $\Delta_f(u)$ est non vide pour tout sommet $u \in V$.

La remarque 25 est une conséquence de la définition 24.

Remarque 25 On considère une inéquation $fx \geq f_0$ triangulaire serrée définissant une facette de $GTSP(n)$. Il existe alors une unique partition $\{V_1, \dots, V_k\}$ de l'ensemble V des sommets du graphe vérifiant :

- $f_e > 0$ et constant pour tout $e \in E(V_i, V_j), i \neq j$,
- $f_e = 0$ si et seulement si $e \in \gamma(V_j)$ pour un j quelconque, $1 \leq j \leq k$.

Démonstration Soient V_1, \dots, V_k les composantes connexes du graphe $G^0 = (V, E^0)$ où $E^0 = \{e \in E : f_e = 0\}$. Soient $e \in \gamma(V_j)$ et $W_e \in \mathcal{W}_f^-$ tels que $e \in W_e$. Supposons que $f_e > 0$, il existe un chemin dans V_j reliant les extrémités de e et pour lequel toutes les arêtes ont un coefficient nul. Le tour W' obtenu à partir de W_e en retirant e et en ajoutant les arêtes de ce chemin vérifie $fx^{W'} < f_0$, ce qui est en contradiction avec la validité de l'inéquation $fx \geq f_0$. Soient à présent e et e' deux arêtes de $(V_i : V_j)$ et supposons que $f_{e'} < f_e$. Soit $W_e \in \mathcal{W}_f^-$ tel que $e \in W_e$. Le tour W' , obtenu à partir de W_e en retirant l'arête e et en ajoutant les arêtes d'un chemin dans V_i allant de l'extrémité de e à l'extrémité de e' qu'il contient et en retirant l'arête e' et en ajoutant les arêtes d'un chemin dans V_j allant de l'extrémité de e' à l'extrémité de e qu'il contient, ne satisfait pas l'inéquation $fx \geq f_0$. \square

Définition 26 Une inéquation $fx \geq f_0$ triangulaire serrée est **simple** si pour tout $e \in E, f_e > 0$.

On peut transformer une inéquation triangulaire serrée quelconque en inéquation simple par la contraction de chaque ensemble V_i de la remarque 25 en un sommet unique.

Théorème 27 (Naddef et Rinaldi [19]) Une inéquation définissant une facette de $GTSP(n)$ appartient à l'une des trois catégories suivantes :

- les inéquations triviales $x_e \geq 0$ pour $e \in E$
- les inéquations de degré $x(\delta(u)) \geq 2$ pour tout $v \in V$
- les inéquations triangulaires serrées

2.3.2 D'une facette à une autre...

On s'intéresse à présent aux conditions suffisantes permettant à une inéquation définissant une facette de $GTSP(n)$ de définir aussi une facette de $STSP(n)$.

Définition 28 *Pour tout triplet ordonné (u, v, w) de sommets distincts de V , on appelle **raccourci** sur (u, v, w) le vecteur $s_{uvw} \in \mathbb{R}^E$ défini par :*

$$s_{uvw}(e) = \begin{cases} 1 & \text{si } e = (v, w) \\ -1 & \text{si } e \in \{(u, v), (u, w)\} \\ 0 & \text{sinon} \end{cases}$$

Le nom de raccourci est justifié par le lemme suivant :

Lemme 29 (Naddef et Rinaldi [19]) *Soient $fx \geq f_0$ une inéquation triangulaire serrée valide sur $GTSP(n)$ et W un tour serré relativement à cette inéquation comportant $t > n$ arêtes et qui contient une arête donnée e^* . Pour tout sommet u ayant un degré $k \geq 4$ dans W , il existe un raccourci s_{uvw} tel que $x^W + s_{uvw}$ est un tour serré qui contient e^* et qui a $t - 1$ arêtes.*

Définition 30 *Une base d'une inéquation $fx \geq f_0$ définissant une facette de $GTSP(n)$ est un ensemble \mathcal{B}_f de tours serrés dont les vecteurs d'incidence sont linéairement indépendants.*

Définition 31 *Un tour est **presque hamiltonien en u** si le sommet u a un degré égal à 4 et tous les autres sommets ont un degré égal à 2 dans celui-ci.*

Définition 32 *Une base \mathcal{B}_f d'une inéquation $fx \geq f_0$ définissant une facette de $GTSP(n)$ est **canonique** lorsqu'elle contient $|E| - n$ cycles hamiltoniens et n tours presque hamiltoniens, un pour chaque sommet $u \in V$.*

La remarque qui suit limite les bases auxquelles on se référera pour démontrer les théorèmes qui vont suivre, afin d'obtenir des conditions et des résultats simples.

Remarque 33 *Soit $fx \geq f_0$ une inéquation triangulaire serrée valide sur $GTSP(n)$. On considère une de ses bases \mathcal{B}_f . Soit $\{W_u : u \in V\}$ un ensemble de n cycles presque hamiltoniens de \mathcal{W}_f^- , dans lequel W_u est presque hamiltonien en u . Si chaque tour de \mathcal{B}_f peut se réduire en un cycle hamiltonien de \mathcal{H}_f^- en utilisant uniquement des raccourcis obtenus par combinaison linéaire d'éléments de $\mathcal{S} = \mathcal{H}_f^- \cup \{W_u : u \in V\}$, alors \mathcal{S} contient une base canonique de $fx \geq f_0$. Donc tout vecteur d'incidence des tours W de \mathcal{B}_f peut être exprimé comme une combinaison linéaire de vecteurs d'incidence d'éléments de \mathcal{S} . En général il est suffisant que chaque raccourci soit obtenu à l'aide d'une combinaison linéaire de vecteurs d'incidence d'éléments de \mathcal{S} avec des coefficients réels. Pour obtenir des conditions suffisantes simples, on se restreindra aux coefficients appartenant à $\{-1, 0, 1\}$.*

Théorème 34 (Naddef et Rinaldi [19]) *Une inéquation triangulaire serrée $fx \geq f_0$ définissant une facette de $STSP(n)$ définit aussi une facette de $GTSP(n)$.*

On définit à présent deux notions qui permettront d'obtenir des résultats sur la description polyédrale de $STSP(n)$.

Définition 35 *Deux arêtes distinctes $e_1, e_2 \in E$ sont **f-adjacentes** s'il existe un cycle hamiltonien serré relativement à f contenant à la fois e_1 et e_2 .*

Définition 36 *Soient $e = (u, v)$ et $e' = (u', v')$ deux arêtes distinctes de E et w un sommet de V . Les arêtes e et e' sont **f-adjacentes en w** si :*

1. e et e' appartiennent à l'ensemble $\Delta_f(w)$
2. Il existe un tour Γ_w presque hamiltonien en w , vérifiant $fx^{\Gamma_w} = f_0$, qui contient les arêtes (w, u) , (w, v) , (w, u') et (w, v')
3. Les tours $\Gamma_w - (w, u) - (w, v) + e$ et $\Gamma_w - (w, u') - (w, v') + e'$ sont des cycles hamiltoniens

Définition 37 *Un ensemble d'arêtes $F \subseteq E$ est **f-connecté** si pour toute paire d'arêtes distinctes $e, e' \in F$, il existe une séquence de k arêtes e'_1, \dots, e'_k de F vérifiant $e'_1 \equiv e, e'_k \equiv e'$ et pour tout $i \in \{1, \dots, k-1\}$, e'_i est f-adjacente à e'_{i+1} .*

On propose à présent deux lemmes donnant des conditions suffisantes pour qu'une inéquation définissant une facette de $GTSP(n)$ définisse aussi une facette de $STSP(n)$.

Lemme 38 (Naddef et Rinaldi [19]) *Soit $fx \geq f_0$ une inéquation triangulaire serrée définissant une facette de $GTSP(n)$. Si $\Delta_f(u)$ est f-connecté en u pour tout $u \in V$, alors $fx \geq f_0$ possède une base canonique et définit une facette de $STSP(n)$.*

Le lemme qui suit sera utile par la suite pour démontrer des propriétés polyédrales de certaines classes d'inéquations valides et définissant des facettes de $GTSP(n)$.

Lemme 39 (Naddef et Rinaldi [19]) *Soit $fx \geq f_0$ une inéquation triangulaire serrée définissant une facette de $GTSP(n)$. Si l'inéquation admet une base \mathcal{B}_f et que pour tout $u \in V$ il existe un ensemble non vide d'arêtes $J_u \subseteq \Delta_f(u)$, f-connecté en u , tel que tout tour fermé $W \in \mathcal{B}_f$ puisse être réduit en cycle hamiltonien serré uniquement en utilisant des raccourcis de l'ensemble $\{s_{uvw} : (v, w) \in J_u, u \in V\}$, alors $fx \geq f_0$ possède une base canonique et définit donc une facette de $STSP(n)$.*

2.3.3 Ajout de sommets

Cette méthode est appelée en anglais *node lifting*. La remarque 25 a permis d'établir la structure des inéquations triangulaires serrées définissant des facettes de $GTSP(n)$. L'idée de l'ajout de sommets est de n'étudier que des inéquations **simples**, c'est-à-dire dans lesquelles tous les coefficients sont non nuls, pour ensuite remplacer un sommet u par un ensemble de sommets V_u vérifiant $f_e = 0$ pour tout $e \in \gamma(V_u)$, ou bien ajouter une clique en dehors de tout sommet (ou ensemble) u (ou V_u). Dans un cas plus général, l'opération d'ajout de sommets consiste à étendre des inéquations définissant des facettes sur $STSP(n)$ (respectivement $GTSP(n)$) à des inéquations définissant des facettes sur $STSP(n^*)$ (respectivement $GTSP(n^*)$) où $n^* > n$. On définit de façon plus formelle cette notion dans la définition suivante.

Définition 40 Soient $K_n = (V_n, E_n)$ et $K_{n^*} = (V_{n^*}, E_{n^*})$ deux graphes complets à n et n^* sommets, où $n^* > n$. On note $\{u_1, \dots, u_n\}$ les sommets de K_n et $V_n \cup \{u_{n+1}, \dots, u_{n^*}\}$ les sommets de K_{n^*} . L'inéquation $f^*x \geq f_0^*$ définie sur $\mathbb{R}^{E_{n^*}}$ est obtenue par **ajout de sommets** de l'inéquation $fx \geq f_0$ définie sur \mathbb{R}^{E_n} si

$$f_{u_i, u_j}^* = f_{u_i, u_j} \quad \text{pour tout } 1 \leq i < j \leq n$$

On s'intéresse plus particulièrement à deux sortes d'ajouts de sommets : la duplication et la création, que l'on définit à présent, en reprenant les notations introduite dans la définition 40.

Définition 41 Un ajout de sommets est appelé **duplication du sommet u** (*0-node lifting du sommet u*) lorsque $u \in V_n$ et $f_{u,v}^* = 0$ pour tout $v \in \{u_{n+1}, \dots, u_{n^*}\}$. Une duplication permet de remplacer un sommet dans une configuration simple par un ensemble de sommets.

Définition 42 Un ajout de sommets est appelé **création** lorsqu'il n'est pas une duplication et que $n^* = n + 1$. Une opération de création consiste à ajouter un seul sommet dans la configuration

On fournit à présent un théorème qui permet de conclure sur la nature d'une inéquation obtenue par duplication de sommets lorsque celle-ci définit une facette de $GTSP(n)$.

Théorème 43 (Naddef et Rinaldi [19]) Soit $fx \geq f_0$ une inéquation définissant une facette de $GTSP(n)$. Toute inéquation obtenue à partir de $fx \geq f_0$ en remplaçant un sommet par une clique est facette de $GTSP(n^*)$, où n^* est le nombre de sommets du graphe résultat.

On fournit deux théorèmes permettant d'effectuer des opérations de duplication et de création sur des facettes du polytope $STSP(n)$. Les démonstrations de ces théorèmes sont techniques et ne sont pas détaillées dans cette section. Une démonstration complète est disponible dans [19].

Théorème 44 (Naddef et Rinaldi [19]) *Soit $fx \geq f_0$ une inéquation triangulaire serrée définissant une facette de $STSP(n)$. Une inéquation $f^*x \geq f_0^*$ obtenue par une opération de création à partir de $fx \geq f_0$ définit une facette de $STSP(n+1)$ si elle est triangulaire serrée et s'il existe un ensemble d'arêtes $F \subseteq \Delta_{f^*}(u_{n+1})$ et une base canonique \mathcal{B}_f de $fx \geq f_0$ vérifiant :*

1. $F \cap \Gamma \neq \emptyset$ pour tout $\Gamma \in \mathcal{B}_f$
2. pour tout $e \in F$ il existe $e' \neq e, e' \in \Delta_{f^*}(u_{n+1})$ et un cycle hamiltonien Γ serré relativement à f tels que e et e' appartiennent à Γ
3. toute composante connexe du graphe (V_n, F) contient au moins un cycle impair
4. F est f^* -connecté en u_{n+1}

Théorème 45 (Naddef et Rinaldi [19]) *Soit $f^*x^* \geq f_0$ une inéquation triangulaire serrée définie sur $\mathbb{R}^{E_{n^*}}$ et obtenue par la duplication du sommet $u \in V_n$ à partir d'une inéquation triangulaire serrée non triviale définissant une facette de $STSP(n)$. Si l'ensemble d'arêtes $\delta(u) \subseteq E_n$ est f -connecté alors $f^*x^* \geq f_0$ définit une facette de $STSP(n^*)$.*

Chapitre 3

Méthodes avancées de séparation

Dans ce chapitre on développe les méthodes utilisées pour effectuer l'étape de séparation. Cette étape consiste à renforcer la relaxation du programme linéaire en nombres entiers modélisant le problème, par l'ajout de contraintes définissant des facettes du polytope associé. L'ajout de contraintes issues de familles connues de facettes nécessite la connaissance de ces familles et des algorithmes suffisamment efficaces et performants pour trouver de telles contraintes. En fonction des familles de contraintes, on dispose d'algorithmes exacts permettant de trouver toutes les contraintes d'une famille violées par une solution d'une relaxation, ou le plus souvent d'heuristiques qui peuvent trouver des contraintes appartenant à une famille donnée, mais ne garantissent pas qu'il n'en existe pas si elles n'en trouvent pas. L'ajout de contraintes issues de familles non connues de facettes est aussi développé dans ce chapitre.

3.1 Principales classes de contraintes et leurs propriétés polyédrales

Dans cette section on décrit les principales classes de contraintes définissant des facettes pour le polyèdre associé au problème du Voyageur de Commerce, puis on s'attarde principalement sur les inéquations de dominos, dont nous avons montré des propriétés faciales au cours de mon travail de thèse.

3.1.1 Contraintes d'élimination de sous-tours

Les contraintes d'élimination de sous-tours sont utilisées pour la description du problème par un programme linéaire en nombres entiers. Elles traduisent le fait qu'au moins deux arêtes appartenant au tour optimal doivent sortir de tout sous-ensemble de sommets E vérifiant $1 \leq |E| \leq n - 1$. On élimine ainsi la possibilité d'avoir une solution réalisable constituée de deux composantes connexes, d'où le nom donné à ces contraintes.

Théorème 46 *Les contraintes d'élimination de sous-tours définissent des facettes pour le polytope $STSP(n)$.*

3.1.2 Contraintes de peigne

Les contraintes de peigne ont été découvertes par Chvátal [7] sous une forme restreinte, puis généralisée sous la forme actuelle par Grötschel et Padberg [14]. Ce sont eux qui ont démontré que ces inéquations induisaient des facettes de $STSP(n)$.

Définition 47 *Une configuration de peigne est constituée d'un ensemble $H \subset V$, appelé manche, et d'un nombre impair $t \geq 3$ de sous-ensembles de sommets appelés dents, tels que :*

$$\begin{aligned} H \cap T_i &\neq \emptyset && \text{pour } i \in \{1, \dots, t\} \\ T_i \setminus H &\neq \emptyset && \text{pour } i \in \{1, \dots, t\} \\ T_i \cap T_j &= \emptyset && \text{pour } 1 \leq i < j \leq t \end{aligned}$$

Etant donnée une configuration de peigne, on définit l'**inéquation de peigne** associée par :

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq 3t + 1$$

L'objet du théorème suivant est de montrer la validité des contraintes de peigne.

Théorème 48 *Les inéquations de peigne sont valides pour $GTSP(n)$ (et par conséquent pour $STSP(n)$).*

Démonstration Montrons que l'inégalité suivante est satisfaite :

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq \frac{1}{2} \left[\sum_{j=1}^t x(\delta(H \cap T_j)) + \sum_{j=1}^t x(\delta(T_j \setminus H)) + \sum_{j=1}^t x(\delta(T_j)) \right]$$

Pour cela pour chaque arête e on considère son coefficient c_e dans le membre de gauche de l'inégalité et son coefficient c'_e dans le membre de droite. On distingue les cas suivants :

- Il existe $i, j, i \neq j$ tels que $e \in \delta(T_i) \cap \delta(T_j) \cap \delta(H)$. Alors $c_e = 3$ et $c'_e = 2$.
- Il existe $i, j, i \neq j$ tels que $e \in (\delta(T_i) \cap \delta(T_j)) \setminus \delta(H)$. Alors $c_e = 2$ et $c'_e = 2$.
- Il existe i tel que $e \in \delta(T_i) \cap \delta(H), e \notin \cup_{j \neq i} \delta(T_j)$. Alors $c_e = 2$ et $c'_e = 1$.
- Il existe i tel que $e \in \delta(T_i), e \notin \delta(H)$ et $e \notin \cup_{j \neq i} \delta(T_j)$. Alors $c_e = 1$ et $c'_e = 1$.
- Lorsque $e \in \delta(H), e \notin \cup_{i=1}^t \delta(T_i)$, on obtient : $c_e = 1$ et $c'_e = 0$.

- Si on note $e = (u, v)$, la dernière possibilité est $\{u, v\} \cap (H \cup_{i=1}^t T_i) = \emptyset$. Dans ce cas $c_e = c'_e = 0$.

On a donc constaté que pour tout $e \in E$, $c_e \geq c'_e$. Donc l'inégalité est vérifiée.

D'autre part, pour tout $j \in \{1, \dots, t\}$, les inégalités suivantes sont vérifiées (propriétés d'un cocycle) : $x(\delta(H \cap T_j)) \geq 2$, $x(\delta(T_j \setminus H)) \geq 2$, et $x(\delta(T_j)) \geq 2$. On en déduit donc :

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq 3t$$

Comme tout cycle traverse les frontières d'un ensemble de sommets un nombre pair de fois, la somme $x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j))$ est somme de termes pairs, donc paire. Par conséquent on peut remplacer $3t$ par $3t + 1$ dans l'inégalité, d'où le résultat. \square

Théorème 49 *Les contraintes de peigne définissent des facettes de $GTSP(n)$.*

Théorème 50 *Les contraintes de peigne définissent des facettes de $STSP(n)$.*

Démonstration Plusieurs méthodes permettent de démontrer ce résultat. La démonstration originale, due à Grötschel et Padberg [14] [15] est longue et technique. Une deuxième méthode repose sur le fait que les inéquations de peigne définissent des facettes de $GTSP(n)$ (théorème 49), puis d'utiliser le lemme 39 pour montrer que ces inéquations définissent aussi des facettes de $STSP(n)$ et enfin d'utiliser les résultats associés à l'ajout de sommets.

On montre ici que la contrainte de peigne à 3 dents dans un graphe à 6 sommets définit une facette de $STSP(6)$. On verra dans la section 3.2 que ce résultat est suffisant, car les contraintes de peigne sont un cas particulier d'une autre famille de contraintes.

Une facette de $STSP(6)$ est de dimension 8, donc une contrainte définissant une telle facette est satisfaite à égalité par neuf cycles hamiltoniens affinement indépendants. Les neuf cycles hamiltoniens des figures 3.1, 3.2, et 3.3 sont affinement indépendants. De plus tous ces cycles satisfont l'inéquation de peigne à égalité, par conséquent cette inéquation définit une facette de $STSP(6)$. \square

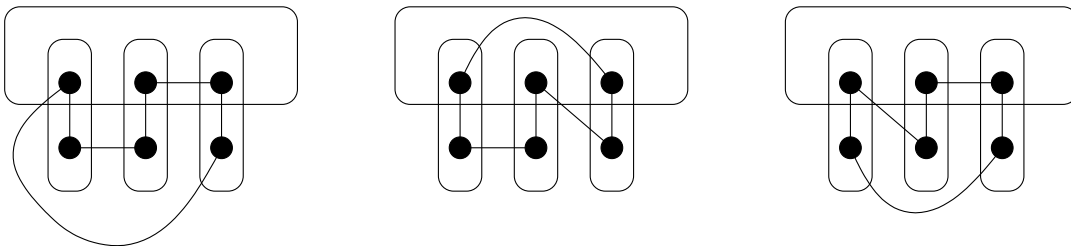


FIG. 3.1 – Cycles hamiltoniens vérifiant l'inéquation de peigne à égalité

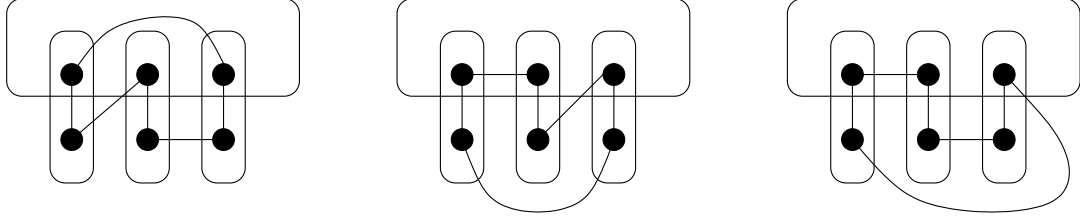


FIG. 3.2 – Cycles hamiltoniens vérifiant l'inéquation de peigne à égalité

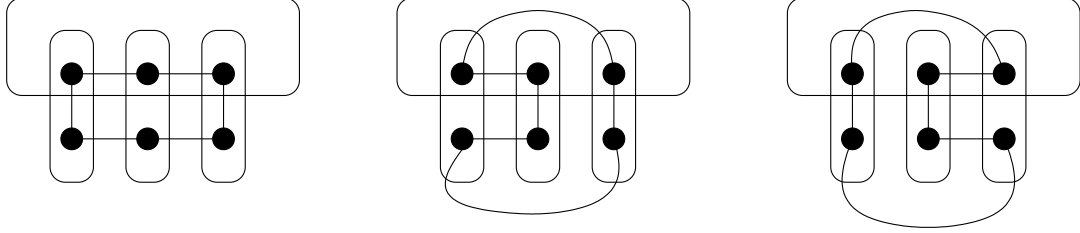


FIG. 3.3 – Cycles hamiltoniens vérifiant l'inéquation de peigne à égalité

3.1.3 Généralisations des contraintes de peigne

D'autres classes de contraintes définissant des facettes du polyèdre $GTSP(n)$ ou $STSP(n)$ ont été proposées. On présente ici deux de ces classes de contraintes, dont les peignes constituent un sous-ensemble. Il existe aussi des contraintes valides dont les contraintes de peigne ne sont pas un sous-ensemble, comme par exemple les contraintes d'échelle, dont Boyd et al. [3] ont montré qu'elles définissent des facettes de $STSP(n)$.

3.1.3.1 Inéquations d'étoile et chemin

Les inéquations d'étoiles ont été proposées pour la première fois par Fleischman [12]. Elles sont définies par deux ensembles de sous-ensembles de sommets : $\mathcal{K} = \{H_1, \dots, H_h\}$ est l'ensemble des **manches**, et $\mathcal{T} = \{T_1, \dots, T_t\}$ est l'ensemble des **dents**, avec $t \geq 3$ et t impair. On associe à chaque manche H_i un entier non nul α_i et à chaque dent un entier non nul β_j , de telle sorte que :

$$\begin{aligned}
 H_1 &\subset H_2 \subset \dots \subset H_h \\
 T_i \cap T_j &= \emptyset && \text{pour } 1 \leq i < j \leq t \\
 H_1 \cap T_j &\neq \emptyset && \text{pour } 1 \leq j \leq t \\
 T_j \setminus H_h &\neq \emptyset && \text{pour } 1 \leq j \leq t \\
 (H_{i+1} \setminus H_i) \setminus \cup_{j=1}^t T_j &= \emptyset && \text{pour } 1 \leq i \leq h-1
 \end{aligned}$$

Une condition supplémentaire porte sur la relations entre les coefficients α_i et β_j : la **propriété d'intervalle**. On appelle intervalle relatif à une dent T_j un ensemble

d'indices successifs de manches qui ont la même intersection avec T_j . Un intervalle est noté $I = l, l + 1, \dots, l + r$, on appelle poids de I l'entier $\sum_{i=l}^{l+r} \alpha_i$. La propriété d'intervalle est la suivante : pour chaque dent T_j , la valeur de β_j est supérieure ou égale au poids maximal d'un intervalle relatif à T_j .

L'inéquation de chemin est le cas particulier où tous les intervalles relatifs à une même dent T_k ont le même poids et le coefficient β_k de cette dent est exactement la valeur de ce poids.

L'inéquation associée à ces ensembles s'écrit :

$$\sum_{i=1}^h \alpha_i x(\delta(H_i)) + \sum_{j=1}^t \beta_j x(\delta(T_j)) \geq (t+1) \sum_{i=1}^h \alpha_i + 2 \sum_{j=1}^t \beta_j$$

On constate que l'inéquation de peigne est exactement l'inéquation de chemin avec un seul manche. Les propriétés connues actuellement de ces inéquations sont données par les théorèmes suivants.

Théorème 51 (Fleischmann [12]) *Les inéquations d'étoiles sont valides sur le polyèdre $GTSP(n)$ et par conséquent sur $STSP(n)$.*

Théorème 52 (Cornuéjols et al. [8]) *Les inéquations de chemin définissent des facettes de $GTSP(n)$*

Théorème 53 (Queyranne et Wang, Naddef et Rinaldi) *Les inéquations de chemin définissent des facettes de $STSP(n)$*

Une démonstration de ce théorème peut être trouvée dans l'article [20].

On ne dispose pas encore de conditions suffisantes pour que les inéquations d'étoiles définissent des facettes de $GTSP(n)$ ou $STSP(n)$.

3.1.3.2 Inéquations d'arbres de cliques

Les inéquations d'arbres de cliques ont été définies par Grötschel et Pulleyblank [16]. Un arbre de cliques est défini par des ensembles de sous-ensembles des sommets d'un graphe : $\mathcal{K} = \{H_1, \dots, H_h\}$ est l'ensemble des manches et $\mathcal{T} = \{T_1, \dots, T_t\}$ est l'ensemble des dents, de telle sorte que :

$$\begin{aligned} H_i \cap H_j &= \emptyset && \text{pour } 1 \leq i < j \leq h \\ T_i \cap T_j &= \emptyset && \text{pour } 1 \leq i < j \leq t \\ T_j \setminus \cup_{i=1}^n H_i &\neq \emptyset && \text{pour } 1 \leq j \leq t \\ t_j &= |\{j : T_j \cap H_i \neq \emptyset\}| \geq 1 && \text{pour } 1 \leq j \leq t \\ h_i &= |\{j : H_i \cap T_j \neq \emptyset\}| \geq 3 \text{ et impair} && \text{pour } 1 \leq i \leq h \\ &\text{si } T_j \cap H_i \neq \emptyset, \text{ c'est un ensemble} && \\ &\text{déconnectant de l'hypergraphe } (V, \mathcal{K} \cup \mathcal{T}). && \end{aligned}$$

Là encore, un arbre de cliques à un seul manche est un peigne. L'inéquation correspondant à une telle configuration s'écrit :

$$\sum_{i=1}^h x(\delta(H_i)) + \sum_{j=1}^t x(\delta(T_j)) \geq \sum_{i=1}^h (h_i + 1) + 2t$$

Les inéquations d'arbres de cliques sont aussi une généralisation des contraintes d'élimination de sous-tours. En effet, si l'on considère une configuration d'arbres de cliques sans manche et ne contenant qu'une seule dent, on obtient une configuration de sous-tour.

Théorème 54 (Grötschel et Pulleyblank [16]) *Les inéquations d'arbres de cliques définissent des facettes de $STSP(n)$.*

3.2 Une classe particulière de contraintes : les inéquations de dominos

On étudie à présent en détail une classe de contraintes valides dont nous avons démontré au cours du travail de thèse qu'elles définissaient des facettes de $STSP(n)$ sous certaines conditions. Ces contraintes constituent une généralisation des contraintes de peigne différentes des deux vues dans la section précédente.

3.2.1 Définition et validité

Tout d'abord nous définissons quelques notions qui vont nous être utiles pour la description des contraintes. On considère un ensemble S de sous-ensembles de V : $S = \{S_1, \dots, S_n\}, S_i \subset V$. La définition des inéquations de dominos repose sur la notion d'*ensembles emboîtés* que nous définissons ici.

Définition 55 *L'ensemble S est appelé emboîté si pour tout i, j , soit $S_i \cap S_j = \emptyset$, soit $S_i \subseteq S_j$, soit $S_j \subseteq S_i$.*

Par la suite, les éléments de S ne contenant aucun autre élément de S seront appelés **minimaux**, les éléments de S qui ne sont compris dans aucun autre élément de S sont appelés **maximaux**. Un élément $S_i \in S$ est dit **impair** s'il contient un nombre impair d'éléments de S (y compris lui-même), il est pair sinon. En particulier un élément minimal de S est impair.

Définition 56 *Un ensemble emboîté S de sous-ensembles de V est **couvert par des dominos** si chaque ensemble non minimal S_i est partitionné en deux ensembles non vides A_i et B_i . On appelle alors S_i un **domino**, A_i et B_i en sont ses deux **demi-dominos**. L'ensemble S définit une **couverture propre de dominos** si pour chaque ensemble non minimal S_i , pour tout sous-ensemble $S_j \subset S_i$, S_j est contenu dans un demi-domino de S_i et chaque demi-domino de S_i contient au moins un élément de S .*

Cette définition permet d'introduire les configurations de dominos desquelles on déduit les inéquations de dominos.

Définition 57 Une **configuration de dominos** est définie par un ensemble $H \subset V$, appelé manche, et une couverture propre de dominos T , dont les ensembles sont appelés dents : $T = \{T_1, \dots, T_t\}$, avec $t \geq 3$ et t impair. Lorsqu'une dent T_j n'est pas minimale elle peut s'écrire sous la forme $T_j = A_j \cup B_j$, avec $A_j \neq \emptyset, B_j \neq \emptyset, A_j \cap B_j = \emptyset$. On note alors $\alpha_j = |\{i : T_i \subset A_j, T_i \text{ maximale relativement à } A_i \text{ et impaire}\}|$ et $\beta_j = |\{i : T_i \subset B_j, T_i \text{ maximale relativement à } B_j \text{ et impaire}\}|$. Une configuration de dominos satisfait les propriétés suivantes :

- $H \cap T_j \neq \emptyset$ pour $j \in \{1, \dots, t\}$
- $T_j \setminus H \neq \emptyset$ pour $j \in \{1, \dots, t\}$
- $\alpha_j \geq 2, \beta_j \geq 2$ pour T_j non minimale, $j \in \{1, \dots, t\}$
- le nombre de dents maximales impaires est au moins de 3

On peut remarquer que les configurations de peigne sont des configurations de dominos particulières dans lesquelles toutes les dents sont minimales. D'autre part chaque demi-domino d'une dent non minimale contient au moins deux dents impaires.

Soit C l'ensemble des arêtes ayant une extrémité dans un demi-domino et l'autre extrémité dans l'autre demi-domino d'une même dent et qui ne traversent pas le manche : $C = (\cup_{j=1}^t (A_j : B_j)) \setminus \delta(H)$. L'**inéquation de dominos** s'écrit alors :

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) + 2x(C) \geq 3t + 1$$

Théorème 58 L'inéquation de dominos est valide pour le polytope $STSP(n)$.

Démonstration On utilise le même argument que pour la démonstration des inéquations de peigne. Ainsi si on appelle PM la premier membre de l'inéquation de dominos, il est facile de vérifier que :

$$PM \geq \frac{1}{2} \sum_{i=1}^t x(\delta(A_i)) + \frac{1}{2} \sum_{i=1}^t x(\delta(B_i)) + \frac{1}{2} \sum_{i=1}^t x(\delta(T_i)) \geq 3t$$

Comme PM est un nombre pair, on peut remplacer $3t$ par $3t + 1$. □

Définition 59 Une configuration de dominos est **minimale** si chaque sommet $v \in V$ appartient à une dent minimale.

On s'intéresse à présent aux différentes configurations de dominos ayant la même inéquation associée.

Définition 60 Deux configurations de dominos \mathcal{D} et \mathcal{D}' sont **équivalentes** si leurs inéquations associées sont les mêmes.

Le théorème suivant donne une méthode permettant de transformer une configuration de dominos en une équivalente.

Théorème 61 (Boyd et Cockburn [4]) Soit \mathcal{D} une configuration de dominos de manche H possédant une dent non minimale T dont les deux demi-dominos sont A et B . Alors la configuration de dominos dans laquelle T est remplacée par la dent $T' = ((V \setminus T) \cup B)$ ayant pour demi-dominos $V \setminus T$ et B , et dans laquelle le manche H est remplacé par le manche $H\Delta A$, différence symétrique de H et A , définit la même inéquation que \mathcal{D} .

3.2.2 Propriétés des cycles hamiltoniens serrés relativement à une inéquation de dominos

On s'intéresse à présent aux cycles hamiltoniens serrés relativement à une inéquation de dominos. Les propriétés énoncées et démontrées dans cette section permettront de déduire que les inéquations de dominos induisent des facettes de *STSP*.

Définition 62 Une arête e est **pénalisante** pour une configuration de dominos \mathcal{D} si elle appartient à l'ensemble $C = (\cup_{j=1}^t (A_j : B_j))$. En d'autres termes e est une arête qui n'intersecte pas le manche H et dont les extrémités appartiennent chacune à un demi-domino distinct d'une même dent.

Théorème 63 Etant données une inéquation de dominos et sa configuration associée, les cycles hamiltoniens Γ serrés relativement à cette configuration appartiennent à l'une des catégories suivantes :

- Toutes les dents sont serrées et Γ ne contient aucune arête pénalisante. Alors $|\delta(H) \cap \Gamma| = t+1$. On note e^* une des arêtes appartenant à $\delta(H) \cap \Gamma$ qui vérifie $|\Gamma \cap (A_j : B_j) \cap (\delta(H) \setminus \{e^*\})| = 1$ pour tout $j \in \{1, \dots, t\}$ et on l'appelle **arête joker**.
- Toutes les dents sont serrées et Γ contient une arête pénalisante $e \in (A_j : B_j)$. Alors $|\delta(H) \cap \Gamma| = t-1$ et $|\Gamma \cap (A_i : B_i) \cap \delta(H)| = 1$ pour tout $i \in \{1, \dots, t\}, i \neq j$, et par conséquent Γ ne contient pas d'arête joker.
- Toutes les dents, sauf une que l'on note T_j , sont serrées. Alors Γ ne contient pas d'arête pénalisante, $|\delta(H) \cap \Gamma| = t-1$ et $|\Gamma \cap (A_i : B_i) \cap \delta(H)| = 1$ pour tout $i \in \{1, \dots, t\}, i \neq j$, et par conséquent Γ ne contient pas d'arête joker non plus.

Démonstration Les résultats de ce théorème sont une conséquence directe de la démonstration de la validité des contraintes de dominos. \square

Définition 64 *Etant donnée une dent T^* , on appelle **visite propre** de T^* une chaîne qui passe par tous les sommets de T^* sans utiliser d'arête pénalisante ni d'arête joker et telle que toutes les dents comprises dans T^* (y compris T^*) sont serrées.*

Comme dans une visite propre, les sommets appartenant à une même classe sont visités successivement, on peut supposer pour les lemmes suivants que la configuration de dominos considérée est simple.

Lemme 65 *Soient T_j une dent d'une configuration de dominos et $a \in T_j$ un de ses sommets. Il existe une visite propre de T_j dont une extrémité est a .*

Démonstration La propriété est triviale si T_j est une dent minimale. Supposons à présent qu'il existe une ou plusieurs dents non minimales de la configuration de dominos pour lesquelles la propriété n'est pas vérifiée. Parmi toutes ces dents, on considère celle qui est minimale et on la note $T = A \cup B$. Les dents qui sont dans T vérifient donc la propriété. Supposons que $a \in A \cap H$ et que la configuration soit minimale. On construit une visite de la façon suivante : on commence par le sommet a , puis on visite proprement la dent maximale relativement à T qui contient a . On continue sans franchir le manche en allant vers une dent maximale non visitée jusqu'à présent dans le demi-domino A que l'on visite proprement, puis on répète le processus de visite des dents non visitées sans traverser le manche pour passer d'une dent à l'autre jusqu'à ce que toutes les dents de A soient visitées proprement. On se rend ensuite dans une dent maximale de l'autre demi-domino en utilisant une arête de $\delta(H)$, on la visite proprement et on répète le processus jusqu'à ce que toutes les dents maximales de B soient visitées.

Si la configuration n'est pas minimale, il se peut qu'il y ait des sommets, dont a , qui ne soient contenus dans aucune dent maximale de T . Si a est un tel sommet, on commence la visite par a puis on entre dans une dent maximale de A tout en ne traversant pas le manche, et on procède comme précédemment. On note Γ la chaîne obtenue. Il se peut qu'il y ait encore des sommets non visités. Si tel est le cas, ces sommets ne sont contenus dans aucune dent maximale de T . Ils sont alors un sous-ensemble de $\{a' \in A \cap H, \bar{a}' \in A \setminus H, b' \in B \cap H, \bar{b}' \in B \setminus H\}$. Montrons comment modifier Γ pour visiter un sous-ensemble quelconque de ces sommets non visités. On suit la chaîne Γ en commençant en a . Lorsque la visite de la première dent impaire maximale de A est effectuée, la chaîne est en-dehors de H , on insère dans Γ le sommet \bar{a}' s'il existe. Lorsque la seconde dent impaire maximale de A est visitée, on se situe à nouveau dans H et on peut donc insérer dans Γ l'éventuel sommet a' . Après avoir traversé le manche pour se rendre dans le demi-domino B on peut de la même façon insérer les éventuels sommets manquants dans Γ .

Une visite propre de la dent T a été construite, ce qui contredit l'hypothèse initiale. □

Lemme 66 *Soit T_j une dent non minimale d'une configuration de dominos dont les deux demi-dominos sont A_j et B_j . Soient $a \in A_j$ et $b \in B_j$ deux sommets. Alors il existe un cycle hamiltonien Γ serré relativement à la configuration de dominos tel que T_j et toutes les dents qui y sont contenues soient serrées et que Γ entre et sort de T_j en a et b .*

Démonstration La visite de T_j construite n'est pas forcément propre, cela dépend de la parité de la dent et de la position des sommets a et b relativement au manche. Comme le manche peut être remplacé par son complément dans une inéquation de dominos, supposons que $a \in H$. On garde les mêmes conventions de notation que dans le lemme 65.

On construit le cycle hamiltonien Γ en commençant par le sommet a . Si a n'est pas dans une dent maximale de A_j , on continue par la visite propre d'une dent maximale de A_j , sinon on visite proprement la dent maximale (relativement à T_j) contenant a , ce qui est possible d'après le lemme 65. Le parcours continue en visitant proprement les autres dents maximales du demi-domino A_j , en n'empruntant pas d'arête de $\delta(H)$ pour passer d'une dent à l'autre. Après la visite de la première dent impaire, passer par le sommet \bar{a}'_j , s'il existe, et après la visite de la seconde dent impaire, passer par l'éventuel sommet a'_j . La visite du demi-domino A_j se termine en un sommet u . Continuer le parcours en entrant dans B_j par l'utilisation d'une arête e de $\delta(H)$ telle que s'il existe un sommet n'appartenant à aucune dent maximale de B_j (sauf b), alors ce sommet soit une extrémité de e . Sinon choisir un sommet appartenant à une dent maximale de B_j ne contenant pas b . A partir de là, visiter proprement toutes les dents maximales de T_j , sauf celle qui contient b , en empruntant le sommet éventuel n'appartenant à aucune dent maximale de B_j après la visite propre de la première dent impaire rencontrée. Le parcours se termine par un sommet que l'on appelle c . Si b est dans une dent de T_j , visiter proprement la dent contenant b en commençant par b . La visite propre se termine en d . Relier alors c et d par une arête (cette arête peut éventuellement être l'arête joker si $(c, d) \in \delta(H)$). Tous les sommets de la dent T_j sont visités. Continuer la visite en visitant proprement les éventuels sommets n'appartenant à aucune dent et du même côté du manche que b , puis visiter proprement les autres dents maximales de la configuration de dominos en passant de l'une à l'autre sans emprunter d'arête de $\delta(H)$. Après la visite de la première dent maximale impaire, visiter éventuellement les sommets n'appartenant à aucune dent et non encore visités. Une fois que tous les sommets sont visités, relier a à l'autre extrémité du parcours. Si l'arête joker n'a pas encore été utilisée, l'arête servant à la liaison aura ce rôle. \square

Lemme 67 *On considère une configuration de dominos simple \mathcal{D} vérifiant $H \setminus \cup_{i=1}^t T_i = \emptyset$. Soient $e = (u, v)$ et $v = (u', v')$ deux arêtes de $\gamma(H)$ telles que les sommets u et u' appartiennent à deux dents maximales différentes T et T' et les sommets v et v' appartiennent à deux demi-dominos différents d'une dent maximale T^* différente de T et T' . Alors il existe un cycle hamiltonien Γ serré relativement à*

\mathcal{D} tel que $e \in \Gamma$ et $e' \in \Gamma$.

Démonstration Comme pour les démonstrations des lemmes précédents, on montre de quelle façon construire un cycle hamiltonien vérifiant les propriétés requises. Tout d'abord on construit une chaîne parcourant tous les sommets de T^* qui relie v et v' , à l'aide de la démonstration du lemme 66. Cette chaîne peut éventuellement contenir une arête joker suivant la parité de la dent T^* . On complète cette chaîne par les arêtes e et e' . A présent on dispose d'une chaîne ayant pour extrémités u et u' . On la complète par les visites propres de T et T' réalisées selon le lemme 65. On obtient alors une chaîne parcourant tous les sommets de T^*, T, T' ayant pour extrémités deux sommets que l'on appelle w et w' . S'il n'y a pas d'autre dent maximale dans la configuration considérée, les sommets w et w' ne sont pas dans le manche, car T et T' sont impaires. Dans ce cas on complète la chaîne par l'arête (w, w') s'il n'existe pas de sommet en-dehors de la configuration, sinon on relie w et w' en passant par les sommets hors-configuration. Si en revanche il existe d'autres dents maximales dans la configuration, on complète la chaîne à partir de w pour aller vers une dent maximale non visitée sans traverser le manche, puis on visite proprement cette dent maximale. On procède de même pour les autres dents maximales jusqu'à ce qu'elles soient toutes visitées. La première fois que la visite propre d'une dent maximale se termine en-dehors du manche, on ne va pas directement à la dent maximale suivante, mais on passe par les sommets hors-configuration s'il en existe. La chaîne est ensuite fermée par l'arête ayant pour extrémités le dernier sommet visité et w' . Si l'arête joker n'a pas été utilisée, cette dernière arête sera automatiquement une arête de $\delta(H)$, sinon elle ne le sera pas. On a donc construit un cycle hamiltonien serré pour l'inéquation de dominos considérée puisqu'une seule arête joker est utilisée et que toutes les dents sont serrées. \square

Lemme 68 Soit \mathcal{D} une configuration de dominos simple minimale. Soient $e = (u, v)$ et $e' = (u', v')$ deux arêtes de $\gamma(H)$, dont les sommets u et u' appartiennent à deux dents maximales distinctes T et T' , et les sommets v et v' à deux dents différentes distinctes de T et T' , et à la fois minimales et maximales. Alors il existe un cycle hamiltonien Γ serré relativement à \mathcal{D} tel que $e \in \Gamma$ et $e' \in \Gamma$.

Démonstration Soient $\{v, w\}$ et $\{v', w'\}$ les deux dents à la fois minimales et maximales. On considère la chaîne $\{e, (v, w), (w, w'), (w', v'), e'\}$ ayant u et u' pour extrémités. A partir de cette chaîne il est possible de construire un cycle hamiltonien serré relativement à \mathcal{D} de la même façon que dans la démonstration du lemme 67. \square

Lemme 69 Soit \mathcal{D} une configuration de dominos telle que $H \setminus \cup_{i=1}^t T_i = \emptyset$. Soit $T = \{a, b\}$ une dent minimale et maximale, et soient T' et T'' deux dents maximales distinctes différentes de T . Soient $u' \in T' \cap H$ et $u'' \in T'' \cap H$. Alors il existe un cycle hamiltonien serré relativement à \mathcal{D} contenant les arêtes (a, u') et (a, u'') .

Démonstration Soient $\Gamma_{u'}$ (respectivement $\Gamma_{u''}$) une visite propre de T' (resp. T'') dont une des extrémités est u' (resp. u''). Soit v' (resp. v'') l'autre extrémité de chacune des visites. On considère la chaîne définie par $\Gamma_{u'} + \{(a, u'), (a, u'')\} + \Gamma_{u''}$.

On considère le cas où T' et T'' sont deux dents paires, alors v et v' sont dans le manche. On complète la chaîne en partant de v'' et en se rendant à la prochaine dent maximale non visitée sans traverser le manche, en la visitant proprement et ainsi de suite jusqu'à ce que toutes les dents de \mathcal{D} soient visitées. Après la première visite d'une dent maximale impaire, on passe éventuellement par le sommet hors-configuration s'il existe, puis par le sommet b avant de compléter la chaîne par une visite propre d'une dent maximale non encore visitée. La chaîne constituée emprunte ainsi tous les sommets du graphe, ses deux extrémités sont situées dans le manche. On relie les deux extrémités pour obtenir un cycle hamiltonien qui se révèle être serré relativement à \mathcal{D} .

Dans le cas où au moins une des dents T' ou T'' est impaire, supposons T' , on procède de la même façon que précédemment pour construire la chaîne, sans passer par le sommet hors-configuration ni par le sommet b . Une des extrémité de la chaîne n'est pas dans le manche, on l'appelle w . On complète la chaîne par l'ajout des arêtes (v', \bar{c}) , (b, \bar{c}) et (b, w) où \bar{c} est l'éventuel sommet hors-configuration. Le cycle hamiltonien construit est ainsi serré relativement à \mathcal{D} . \square

Lemme 70 *Soit \mathcal{D} une configuration de dominos possédant une dent à la fois minimale et maximale T telle que $T \cap H = \{a\}$. Soient $u \in V \setminus T$ et $v \in T \setminus H$. Il existe un cycle hamiltonien serré relativement à \mathcal{D} qui contient les arêtes (u, a) et (a, v) .*

Démonstration Comme dans les démonstrations des lemmes précédents, on construit progressivement un cycle hamiltonien serré relativement à \mathcal{D} . On commence par construire une chaîne hamiltonienne de $T \setminus H$ qui relie v à un sommet $v' \in T \setminus H$. On complète cette chaîne par les arêtes (u, a) et (a, v) , puis par une visite des dents non visitées jusqu'à présent de la même manière que pour les démonstrations des lemmes précédents. Enfin on relie les deux extrémités de la chaîne précédente par une arête qui est l'arête joker si $(u, a) \in \gamma(H)$ et qui est une arête de $\gamma(H)$ si $(u, a) \in \delta(H)$. \square

3.2.3 Méthode de démonstration utilisée

Le théorème 61 permet de générer des configurations de dominos équivalentes. Il sera régulièrement utilisé pour montrer les résultats associés à l'étude polyédrale des contraintes de dominos.

Définition 71 *Etant donnée une configuration de dominos, on appelle **dent fondamentale** une dent T vérifiant une des trois conditions suivantes :*

- T est une dent paire dont un demi-domino contient uniquement deux dents minimales et l'autre demi-domino contient uniquement trois dents minimales
- T est une dent impaire dont chaque demi-domino contient uniquement 2 dents minimales
- T est une dent impaire dont chaque demi-domino contient uniquement 3 dents minimales

On s'intéresse à présent aux conditions sous lesquelles une contrainte de dominos définit une facette de $STSP(n)$. Pour cela on se base sur le fait que la contrainte de peigne simple à 6 sommets définit une facette de $STSP(6)$ et sur le fait que toute contrainte de dominos peut être obtenue en appliquant un nombre suffisant de fois les opérations suivantes à une contrainte de peigne à 3 dents de K_6 :

1. Ajout de deux dents minimales dans un même demi-domino.
2. Ajout de deux dents minimales en-dehors de toute dent existante.
3. Ajout d'une dent paire fondamentale dans un demi-domino.
4. Ajout d'une dent paire fondamentale en-dehors de toute dent.
5. Remplacement d'une dent minimale par une dent impaire fondamentale.

Puis après ces opérations, il est encore nécessaire d'appliquer des transformations de type création de sommets puis des transformations de type duplication de sommets.

Le but de la démonstration est de montrer que toutes ces opérations préservent les propriétés faciales des contraintes de dominos. On présente ici la méthode utilisée pour montrer ce résultat. Comme on a montré que l'inéquation de dominos $dx \geq d_0$ est valide sur $STSP(n)$ (théorème 58), elle définit une face de $STSP(n)$. Pour montrer que cette inéquation définit une facette de $STSP(n)$, on considère une inéquation $gx \geq g_0$ définissant une facette de $STSP(n)$ qui contient la face définie par $dx \geq d_0$. Comme il existe une infinité d'inéquations linéaires décrivant une facette donnée, une façon de contourner le problème est de fixer la valeur de g_e à d_e pour tout $e \in F^*$, où F^* est un ensemble d'arêtes couvrant V ne possédant qu'un unique cycle, qui est de longueur impaire. Ainsi si $dx \geq d_0$ définit une facette de $STSP(n)$, on aboutit à $d_e = g_e$ pour tout $e \in E$ et $d_0 = g_0$. Dans le cadre de la présente démonstration, l'ensemble F^* sera toujours constitué d'arêtes d'un arbre couvrant et d'une arête supplémentaire formant un cycle impair. Les détails de cette méthode due à Grötschel et Pulleyblank peuvent être consultés dans [16].

3.2.4 Ajout de deux dents minimales ou d'une dent fondamentale paire dans un demi-domino

Comme on applique les transformations décrites dans l'ordre indiqué, on suppose ici n'avoir à ajouter que des dents simples, c'est-à-dire contenant un nombre minimal de sommets dans et en-dehors du manche. Les sommets qui manquent éventuellement seront rajoutés grâce à des opérations de type duplication ou création.

Dans cette section on s'intéresse particulièrement à l'ajout d'une dent fondamentale paire ou de deux dents minimales dans un demi-domino d'une configuration simple définissant une facette de $STSP(n)$. L'ajout est considéré comme un ajout de dents maximales relativement au demi-domino A_i considéré. En utilisant de façon récursive le théorème 61, il existe une configuration de dominos équivalente dont l'ajout est effectué en-dehors de toute dent. Par conséquent on peut supposer que l'ajout de deux dents minimales ou d'une dent fondamentale paire s'effectue en-dehors de toute dent.

Théorème 72 *Soit \mathcal{D} une configuration de dominos simple définissant une facette. Si A représente un demi-domino d'une dent T ou si $A = V \setminus \cup_{i=1}^t T_i$, alors l'ajout de deux dents minimales dans A mène à une configuration de dominos \mathcal{D}^* définissant une facette.*

Démonstration D'après la remarque précédant ce théorème, on peut supposer que $A = V \setminus \cup_{i=1}^t T_i$. On ajoute deux dents $T_{t+1} = \{a, b\}$ et $T_{t+2} = \{c, d\}, \{a, c\} \subset H$.

Supposons que l'inéquation de dominos $d^*x \geq d_0^*$ correspondant à \mathcal{D}^* ne définisse pas une facette. La face qu'elle induit est alors contenue dans une facette définie par $gx \geq g_0$. Supposons que $g_e = d_e^*$ pour tout $e \in F^*$, où F^* est l'ensemble défini de telle façon que pour un sommet donné $w^* \in (H \setminus (T_{t+1} \cup T_{t+2}))$ et pour un sommet $\bar{w}^* \notin (H \cup T_{t+1} \cup T_{t+2})$, $\{(w^*, a), (w^*, c), (\bar{w}^*, b), (\bar{w}^*, d)\} \subset F^*$, et qu'en-dehors de ces quatre arêtes, F^* est un sous-graphe couvrant de $V \setminus (T_{t+1} \cup T_{t+2})$ ne possédant qu'un cycle, de longueur impaire. Soit F l'ensemble d'arêtes défini par $F = F^* \setminus \{(w^*, a), (w^*, c), (\bar{w}^*, b), (\bar{w}^*, d)\}$.

On montre tout d'abord que $g_{wa} = g_{wc}$ pour tout sommet $w \in H \setminus \{a, c\}$. Soit w un sommet qui ne se trouve pas dans la même dent maximale que w^* . D'après le lemme 68, il existe un cycle hamiltonien Γ serré relativement à d^* qui contient les arêtes (w^*, a) et (w, c) . Ce cycle est donc aussi serré relativement à l'inéquation $gx \geq g_0$. Le cycle hamiltonien Γ' défini par $\Gamma' = \Gamma \setminus \{(w^*, a), (w, c)\} + \{(w^*, c), (w, a)\}$ est aussi serré relativement à d^* et doit donc aussi être serré relativement à g , par conséquent on obtient l'égalité : $g_{w^*a} + g_{wc} = g_{w^*c} + g_{wa}$, mais comme $g_{w^*a} = d_{w^*a}^* = d_{w^*c}^* = g_{w^*c}$, on en déduit que $g_{wa} = g_{wc}$ pour tout w dans le manche et pas dans la même dent maximale que w^* . Soit v^* un sommet du manche qui ne se trouve pas dans la même dent maximale que w^* . On sait déjà que $g_{v^*a} = g_{v^*c}$, donc si on remplace w^* par v^* dans ce qui précède, on en déduit que $g_{wa} = g_{wc}$ pour tous les sommets w du manche. Comme de plus le manche et son complémentaire mènent à une même configuration de dominos, on obtient de la même manière le résultat $g_{\bar{w}b} = g_{\bar{w}d}$ pour tout $\bar{w} \notin H \cup T_{t+1} \cup T_{t+2}$.

A présent on considère un sommet $\bar{w} \notin H \cup T_{t+1} \cup T_{t+2}$. La chaîne constituée des arêtes $\{(\bar{w}, a), (a, b), (b, d), (d, c)\}$ peut être complétée en cycle hamiltonien serré relativement à d^* . En effet les deux extrémités de cette chaîne sont opposées par rapport au manche et il reste un nombre impair de dents maximales impaires à visiter (le début et la fin de la visite propre d'une dent impaire sont situés à des côtés opposés du manche, et l'arête joker est utilisée dans le chemin considéré). Soit

w un sommet tel que $w \neq d$ et $(w, c) \in \Gamma$. On sait que $w \in H$ puisque l'arête joker a été utilisée. Le cycle Γ' défini par $\Gamma' = \Gamma \setminus \{(\bar{w}, a), (w, c)\} + \{(\bar{w}, c), (w, a)\}$ est serré relativement à d^* et par conséquent aussi relativement à g , donc comme $g_{wa} = g_{wc}$ on en déduit $g_{\bar{w}a} = g_{\bar{w}c}$ pour tout $\bar{w} \notin H \cup T_{t+1} \cup T_{t+2}$. De même, en inversant les rôles de H et $V \setminus H$, on obtient $g_{wb} = g_{wd}$ pour tout $w \in H \setminus (T_{t+1} \cup T_{t+2})$.

On définit à présent la configuration de dominos \mathcal{D}^+ obtenue à partir de \mathcal{D} par l'ajout d'un sommet h dans le manche en-dehors de toute dent. D'après le théorème 75, \mathcal{D}^+ définit une facette, donc l'inéquation correspondant à cette configuration à laquelle on ajoute l'équation de degré $x(\delta(h)) = 2$ définit aussi une facette. On considère l'ensemble F^+ défini par $F^+ = F + \{(w^*, h)\}$. Soit \mathcal{B} un ensemble de cycles hamiltoniens permettant de montrer que cette inéquation définit une facette, en utilisant F^+ comme ensemble d'arêtes à valeur fixe.

De la même manière on définit la configuration $\bar{\mathcal{D}}^+$ obtenue à partir de \mathcal{D} par l'ajout d'un sommet \bar{h} en-dehors du manche et de toute dent. D'après le théorème 75, $\bar{\mathcal{D}}^+$ définit une facette, de même que l'inéquation obtenue en y ajoutant l'équation de degré $x(\delta(\bar{h})) = 2$. Soit $\bar{F}^+ = F + \{(\bar{w}^*, \bar{h})\}$. On définit $\bar{\mathcal{B}}$ comme étant un ensemble de cycles hamiltoniens permettant de montrer que cette dernière inéquation définit une facette, en utilisant \bar{F}^+ .

Soit $\Gamma \in \mathcal{B}$, on le complète en un cycle hamiltonien serré Γ^* de \mathcal{D}^* de la façon suivante : soient h^+ et h^- les sommets adjacents à h dans Γ , alors $\Gamma^* = \Gamma - \{(h^+, h), (h, h^-)\} + \{(h^+, a), (a, b), (b, d), (d, c), (c, h^-)\}$. Soit \mathcal{B}^* l'ensemble de cycles hamiltoniens obtenus lorsque $\Gamma \in \mathcal{B}$. De même on définit $\bar{\mathcal{B}}^*$ en modifiant les cycles de $\bar{\mathcal{B}}$ en cycles de \mathcal{D}^* .

Les cycles de \mathcal{B}^* et l'ensemble d'arêtes F^+ permettent de montrer que $g_e = d_e$ pour tout $e \in E \setminus (\gamma(T_{t+1} \cup T_{t+2}) \cup (\{b, d\} : (V \setminus \{a, b, c, d\})))$, c'est-à-dire pour toutes les arêtes sauf celles qui relient les sommets des dents ajoutées ou qui relient un des sommets b et d à un sommet en-dehors des dents ajoutées. Les cycles de $\bar{\mathcal{B}}^*$ et l'ensemble d'arêtes \bar{F}^+ permettent de montrer le même résultat en remplaçant b et d par a et c . Par conséquent les seules arêtes dont on ne connaisse pas le coefficient g_e sont celles appartenant à $\gamma(\{a, b, c, d\})$.

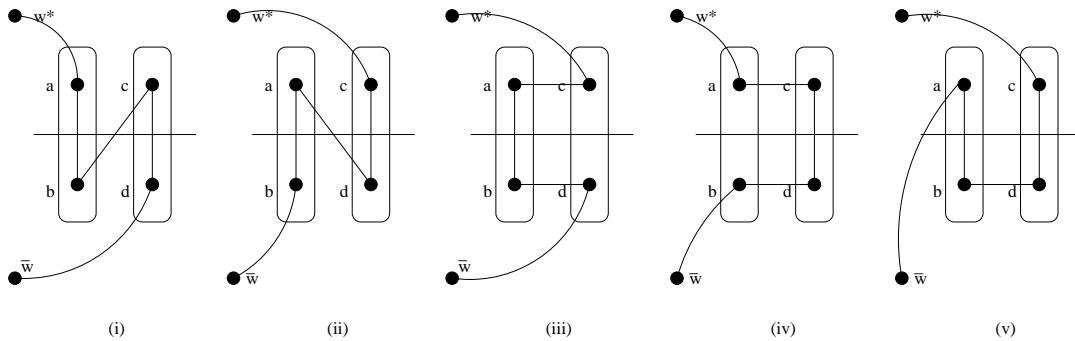


FIG. 3.4 – Intersection de cycles hamiltoniens serrés avec deux dents minimales

On considère à présent les cinq chaînes suivantes qui sont aussi visibles sur la

figure 3.4 :

- Chaîne (i) : $\{(w^*, a), (a, b), (b, c), (c, d), (d, \bar{w})\}$
- Chaîne (ii) : $\{(w^*, c), (c, d), (d, a), (a, b), (b, \bar{w})\}$
- Chaîne (iii) : $\{(w^*, c), (c, a), (a, b), (b, d), (b, \bar{w})\}$
- Chaîne (iv) : $\{(w^*, a), (a, c), (c, d), (d, b), (b, \bar{w})\}$
- Chaîne (v) : $\{(w^*, c), (c, d), (d, b), (b, a), (a, \bar{w})\}$

On complète ces chaînes en cycles hamiltoniens serrés relativement à d^* par le même ensemble d'arêtes pour chaque cycle. On en déduit alors :

- $g_{ad} = g_{bc}$ en comparant (i) et (ii)
- $g_{ab} = g_{cd}$ en comparant (iii) et (iv)
- $g_{ad} = g_{bd} + 1$ en comparant (ii) et (v), car $g_{\bar{w}a} = d_{\bar{w}a} = d_{\bar{w}a} = d_{\bar{w}b} + 1 = g_{\bar{w}b} + 1$ et $g_{w^*a} = g_{w^*c}$.
- $g_{ac} = g_{ab} + 1$ en comparant (iv) et (v), car $g_{\bar{w}a} = d_{\bar{w}a} = d_{\bar{w}a} = d_{\bar{w}b} + 1 = g_{\bar{w}b} + 1$ et $g_{w^*a} = g_{w^*c}$.

Par conséquent toutes les valeurs peuvent être déduites de la valeur de $g_{ab} = \alpha$.

On considère le cycle hamiltonien serré Γ_1 construit de la façon suivante. On commence le cycle par les sommets b, a, d, c (dans cet ordre), puis de c on visite proprement une première dent impaire maximale, en quittant cette dent par un sommet u . On continue par la visite de la dent impaire suivante en entrant par un sommet v , et ainsi de suite par des visites propres des dents impaires maximales. On termine la visite par les dents paires maximales (s'il en existe dans la configuration) et on termine le cycle par un retour au sommet b . Soit $\Gamma_2 = \Gamma_1 \setminus \{(a, d), (u, v)\} + \{(a, u), (d, v)\}$. Les cycles Γ_1 et Γ_2 sont serrés relativement à d^* , car toutes les dents sont visitées proprement et aucune arête pénalisante n'est utilisée tandis qu'une seule arête joker est utilisée. Par définition des coefficients des arêtes non pénalisantes d'une inéquation de dominos qui sont égaux au nombre de frontières qu'elles traversent, on a $g_{uv} = d_{uv} = d_{vd} + d_{ua} - 3$, donc on en déduit $g_{ad} = 3$, d'où $\alpha = 1$.

On a exhibé tous les coefficients g_e . Ils sont égaux aux coefficients d_e^* , par conséquent l'ajout de deux dents minimales à une configuration de dominos mène à une configuration de dominos définissant une facette. \square

Théorème 73 *Soit \mathcal{D} une configuration de dominos simple définissant une facette. Si A représente un demi-dominos d'une dent T ou si $A = V \setminus \cup_{i=1}^t T_i$, alors l'ajout d'une dent fondamentale paire dans A mène à une configuration de dominos \mathcal{D}^* définissant une facette.*

Démonstration La preuve est similaire à la précédente, le nombre de cas à considérer est plus important que précédemment, mais les grandes lignes sont les mêmes. Pour une preuve complète, consulter [23]. \square

3.2.5 Remplacement d'une dent minimale par une dent impaire fondamentale

Dans cette section on étudie le cas où une dent minimale de la configuration considérée est remplacée par une dent impaire fondamentale. Il existe deux types de dents impaires fondamentales : celles possédant 2 dents minimales dans chaque demi-domino, que l'on appellera dents (2, 2) et celles possédant 3 dents minimales dans chaque demi-domino que l'on appellera (3, 3).

Théorème 74 *Soit \mathcal{D} une configuration de dominos simple définissant une facette. Soit T_i une dent minimale de \mathcal{D} . Le remplacement de T_i par une dent fondamentale impaire mène à une configuration de dominos simple définissant une facette.*

Démonstration On considère une configuration de dominos \mathcal{D} minimale simple définissant une facette et son inéquation associée $dx \geq d_0$. Soit $T = \{a, b\}$ une de ses dents minimales. D'après le théorème 61 appliqué récursivement, on peut supposer que la dent T est aussi maximale, c'est-à-dire qu'elle n'est contenue dans aucune autre dent. Soit F un ensemble de $n = |V|$ arêtes constitué d'un arbre couvrant de $V \setminus \{a, b\}$ et des trois arêtes (w^*, a) , où $w^* \in H$, (a, b) et (\bar{w}^*, b) où $\bar{w}^* \notin H$, tel que l'unique cycle de F (auquel appartient l'arête (a, b)) soit de longueur impaire. Il est trivial qu'un tel ensemble existe quelle que soit la configuration de dominos choisie.

On appelle $d'x \geq d'_0$ l'inéquation obtenue à partir de $dx \geq d_0$ par l'ajout des équations de degré $x(\delta(a)) = 2$ et $x(\delta(b)) = 2$. Cette inéquation définit aussi une facette. Soit \mathcal{B} l'ensemble de cycles hamiltoniens permettant de montrer que $d'x \geq d'_0$ définit une facette en utilisant F comme ensemble d'arêtes à valeurs fixées.

Soit \mathcal{D}^* la configuration de dominos obtenue à partir de \mathcal{D} en remplaçant une dent minimale et maximale (on la note T_t) par une dent fondamentale impaire T^* . On suppose que la dent maximale T^* a le numéro de la dent remplacée, c'est-à-dire t . On définit un entier l qui prend pour valeur 4 si la dent ajoutée est (2, 2) et pour valeur 6 si la dent ajoutée est (3, 3). Les sommets des dents minimales de cette nouvelle dent fondamentale sont appelés a_i et b_i , pour i variant de 1 à l , avec $a_i \in H$ et $b_i \notin H$.

- On définit à présent l'ensemble F^* obtenu à partir de F de la façon suivante :
- si $l = 4$: $F^* = (F \setminus \{(a, b), (w^*, a), (\bar{w}^*, b)\}) \cup_{i=1}^4 \{(w^*, a_i)\} \cup_{i=1}^4 \{(\bar{w}^*, b_i)\} \cup \{(a_2, b_3)\}$
 - si $l = 6$: $F^* = (F \setminus \{(a, b), (w^*, a), (\bar{w}^*, b)\}) \cup_{i=1}^6 \{(w^*, a_i)\} \cup_{i=1}^4 \{(\bar{w}^*, b_i)\} \cup \{(b_3, a_4)\}$

La différence entre les deux cas vient du fait que l'arête allant d'un demi-domino à l'autre de la dent T^* est légèrement différente. La figure 3.5 permet de visualiser ces modifications. Sur les figures représentant les configurations de dominos, la frontière entre deux demi-dominos d'une même dent est représentée par des pointillés.

On note $d^*x \geq d_0^*$ l'inéquation de dominos associée à \mathcal{D}^* . Si elle ne définit pas une facette alors elle définit une face qui est contenue dans une facette que l'on note $gx \geq g_0$. On fixe les valeurs $g_e = d_e^*$ pour $e \in F^*$.

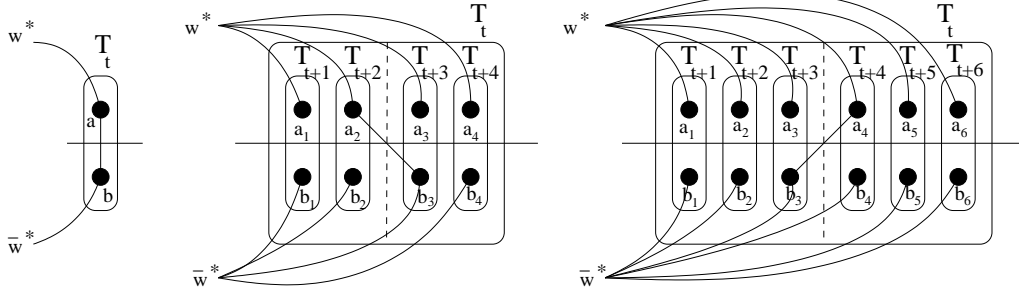


FIG. 3.5 – Les modifications de l'ensemble F

On montre dans un premier temps que pour tout $w \in H$, $g_{w,a_i} = g_{w,a_j}$ pour tout $i, j \in \{1, \dots, l\}$. Soit $w \in H$ un sommet ne se trouvant pas dans la même dent maximale que w^* . Soit $\Gamma_{i,j}$ un cycle hamiltonien serré contenant les arêtes (w^*, a_i) et (w, a_j) , où a_i et a_j ne se trouvent pas dans le même demi-domino. Ce cycle existe d'après le lemme 67. Soit $\Gamma'_{i,j}$ le cycle hamiltonien obtenu à partir de $\Gamma_{i,j}$ de la façon suivante : $\Gamma'_{i,j} = \Gamma_{i,j} \setminus \{(w^*, a_i), (w, a_j)\} \cup \{(w^*, a_j), (w, a_i)\}$. Comme $g_{w^*, a_i} = d_{w^*, a_i}^* = d_{w^*, a_j}^* = g_{w^*, a_j}$, on en déduit que $g_{w, a_i} = g_{w, a_j}$. En changeant le choix de i et j on peut prouver que pour un sommet donné $w \in H$ qui ne se trouve pas dans la même dent maximale que w^* , la valeur g_{w, a_i} ne dépend que de w et pas de i . Si on répète ce raisonnement en remplaçant w^* par un sommet ne se trouvant pas dans la même dent que w^* , on aboutit au même résultat pour tous les sommets du manche se trouvant dans la même dent maximale que w^* . Par conséquent pour tout $w^* \in H \setminus T^*$, $g_{w, a_i} = \beta_w$ pour tout i , valeur qui ne dépend que du sommet w .

En reprenant ce raisonnement et en remplaçant H par $V \setminus H$ et réciproquement, et en utilisant \bar{w}^* à la place de w^* , on obtient le même résultat pour $\bar{w} \notin (H \cup T^*)$, c'est-à-dire que $g_{\bar{w}, b_i} = \beta_w$ pour tout i , valeur qui ne dépend que de \bar{w} .

Les cycles hamiltoniens appartenant à \mathcal{B} sont serrés relativement à la dent T_t ou alors l'intersectent en 4 arêtes que l'on note $(a, w_1), (a, w_2), (b, \bar{w}_1), (b, \bar{w}_2)$. Dans ce dernier cas, aucune des 4 arêtes ne peut appartenir à $\delta(H)$, car aucune arête joker n'est autorisée. Par conséquent les sommets w_1 et w_2 se situent dans le manche et les sommets \bar{w}_1 et \bar{w}_2 en-dehors du manche. Soit $\Gamma \in \mathcal{B}$ un cycle hamiltonien serré relativement à T_t . On construit un cycle hamiltonien Γ^* serré relativement à d^* à partir de Γ de la façon suivante :

- si Γ est serré relativement à T_t : Γ^* est obtenu à partir de Γ en remplaçant les arêtes $(w, a), (a, b)$ et (b, \bar{w}) par une visite propre de la dent T_t (dent fondamentale impaire) comportant les arêtes (w, a_1) et (\bar{w}, a_l) .
- si Γ n'est pas serré relativement à T_t : Γ^* est obtenu à partir de Γ en remplaçant les arêtes $(w_1, a), (a, w_2), (\bar{w}_1, b)$ et (b, \bar{w}_2) par une visite de la dent T_t n'empruntant pas d'arête de $A_t : B_t$ telle que les dents minimales de T_t soient visitées proprement et que le cocycle de la dent T_t soit égal à 4.

Tous les cycles Γ^* obtenus de cette façon à partir de cycles $\Gamma \in \mathcal{B}$ utilisent les mêmes arêtes de la dent fondamentale impaire T_t , sauf pour l'arête qui va d'un demi-domino

à l'autre. Mais cette dernière arête appartient à l'ensemble F^* , donc on connaît son coefficient.

Les cycles Γ^* obtenus à partir des cycles $\Gamma \in \mathcal{B}$ permettent par conséquent de calculer les coefficients g_e pour :

- $e \in \gamma(V \setminus T_t)$
- $e = (w, a_1)$ pour tout $w \in V \setminus T_t$ (w peut ne pas appartenir à H)
- $e = (w, b_1)$ pour tout $w \in V \setminus T_t$ (w peut ne pas appartenir à H)

On peut en déduire les coefficients g_e pour $e = (w, a_i)$ pour tout i et tout $w \in H \setminus T_t$ ainsi que les coefficients g_e pour $e = (\bar{w}, b_i)$ pour tout i et tout $\bar{w} \in V \setminus (H \cup T_t)$.

La différence principale avec le cas de l'ajout d'une dent paire est qu'on ne peut pas utiliser plus de symétrie entre H et $V \setminus H$, car on connaît tous les coefficients g_{w,a_1} pour tout $w \notin T_t$, mais on ne connaît pas les coefficients g_{w,b_1} pour $w \in H$. Il en est de même pour g_{w,b_l} qui est connu pour $w \notin T_t$ alors que ce n'est pas le cas de g_{w,a_l} .

À présent les cas (2, 2) et (3, 3) sont traités séparément. On commence par traiter le cas où la dent T_t est remplacée par une dent (2, 2).

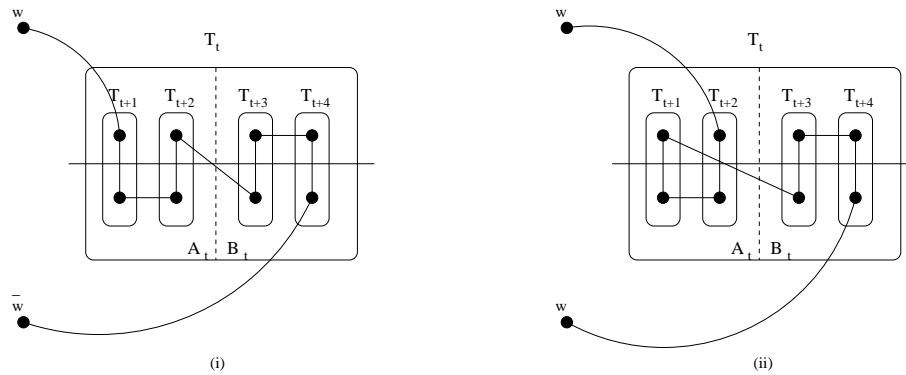


FIG. 3.6 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

Les figures 3.6(i) et 3.6(ii) montrent deux ensembles d'arêtes qui peuvent être complétés en cycles hamiltoniens serrés Γ_1 et Γ_2 par le même ensemble d'arêtes. En comparant ces deux cycles et en utilisant le fait que $g_{w,a_1} = g_{w,a_2}$, on obtient $g_{a_1,b_3} = g_{a_2,b_3} = 3$. On montre de la même manière que $g_{a_i,b_j} = g_{a_2,b_3} = 3$ pour $i \in \{1, 2\}$ et $j \in \{3, 4\}$.

Un cycle hamiltonien serré pour lequel la dent T_{t+2} n'est pas serrée a une intersection avec la dent (2, 2) qui peut être vue sur la figure 3.7(i). Un cycle dérivé de celui-ci pour lequel la dent T_{t+1} n'est pas serrée est visible en figure 3.7(ii). En comparant ces deux cycles on obtient : $g_{a_1,b_1} = g_{a_2,b_2}$ car on a déjà établi que $g_{a_2,b_3} = g_{a_1,b_3}$. En répétant cette opération sur le demi-dominos B_t , on obtient de même $g_{a_3,b_3} = g_{a_4,b_4}$.

On considère à présent un cycle hamiltonien serré Γ qui contient l'arête (a_3, b_4) . Un tel cycle est représenté à la figure 3.8(i). Les cycles présentés en figures 3.8(ii) et 3.8(iii) diffèrent de Γ par seulement quelques arêtes qui sont toutes montrées. En comparant les deux premiers cycles et en utilisant le fait que $g_{u,a_4} = g_{u,a_3}$, on obtient :

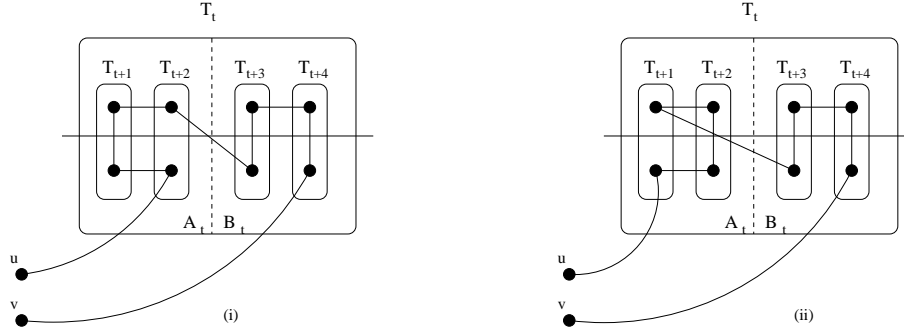


FIG. 3.7 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

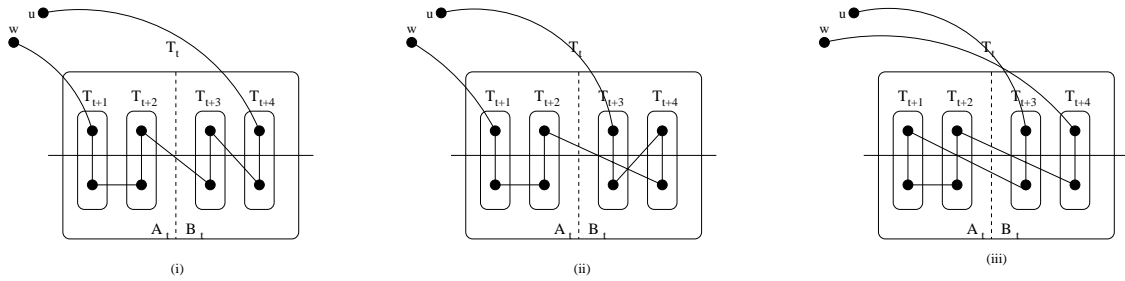


FIG. 3.8 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

$g_{a_4, b_3} = g_{a_3, b_4}$. De la même façon on arrive à montrer que $g_{a_1, b_2} = g_{a_2, b_1}$. En comparant Γ avec le cycle de la figure 3.8(iii), on obtient : $g_{a_4, b_3} = g_{a_3, b_4} = g_{a_1, b_2} = g_{a_2, b_1} = 3$.

Les ensembles d'arêtes représentés à la figure 3.9 peuvent être complétés par le même ensemble d'arêtes en des cycles hamiltoniens serrés. On déduit de ces deux cycles que $g_{a_4, b_1} = g_{a_4, b_2}$ et de la même façon, en utilisant des cycles hamiltoniens serrés pour lesquels la dent T_{t+3} n'est pas serrée, on montre que $g_{a_3, b_1} = g_{a_3, b_2}$.

Les ensembles de la figure 3.10 permettent de montrer que $g_{b_2, a_3} = g_{b_2, a_4}$, et par conséquent que les coefficients g_{a_i, b_j} , pour $i \in \{3, 4\}$ et $j \in \{1, 2\}$ sont tous identiques. En utilisant les ensembles d'arêtes symétriques par rapport au manche des ensembles représentés en figures 3.8(ii) et 3.8(iii) et en utilisant le fait que $g_{a_3, b_4} = 3$, on peut conclure que la valeur commune de ces coefficients est 3.

En comparant les cycles hamiltoniens des figures 3.11(i) et 3.11(ii), en utilisant de plus le fait que la valeur de g_{u, a_1} est connue pour $u \notin H \cup T_t$ et que $g_{u, a_1} = g_{u, b_1} + 1$, on obtient : $g_{a_1, b_2} = g_{b_1, b_2} + 1$, et par conséquent $g_{b_1, b_2} = 2$. On montre de la même façon que $g_{a_3, a_4} = 2$.

A présent on compare le cycle hamiltonien représenté sur la figure 3.8(iii) avec celui de la figure 3.12(i). Comme les seuls coefficients dont on ne connaît pas encore la valeur sont ceux des arêtes (a_1, a_2) et (b_3, b_4) , on obtient : $g_{a_1, a_2} = g_{b_3, b_4}$. Si on choisit un cycle hamiltonien serré qui traverse la dent T_t comme dans la figure 3.5 et qu'on inverse le rôle de chaque demi-domino, on obtient : $g_{a_1, a_2} + g_{b_3, b_4} = g_{a_3, a_4} + g_{b_1, b_2} = 4$, et par conséquent $g_{a_1, a_2} = g_{b_3, b_4} = 2$.

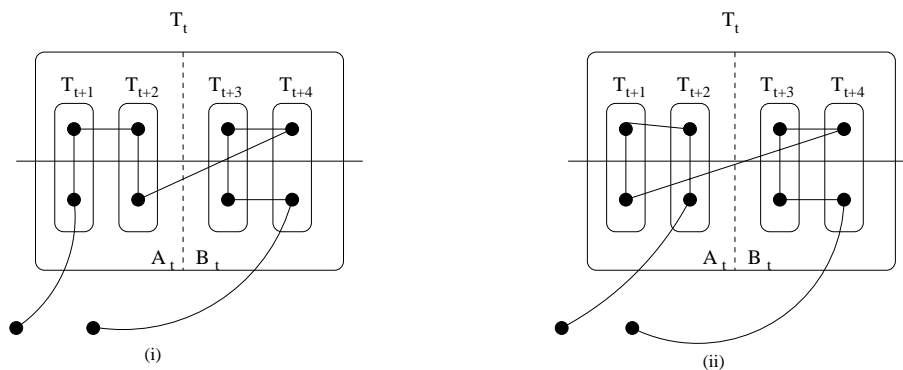


FIG. 3.9 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

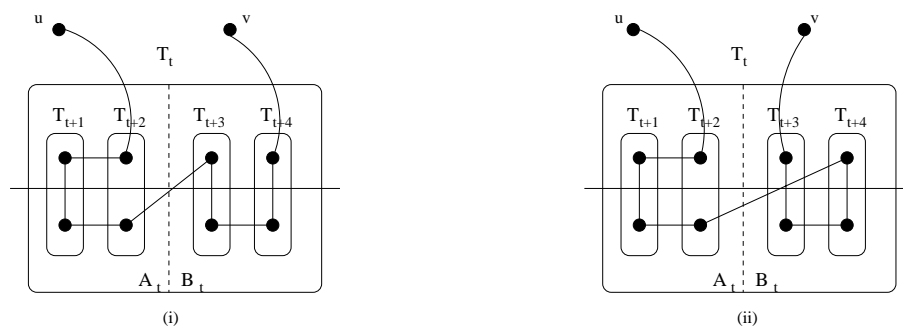


FIG. 3.10 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

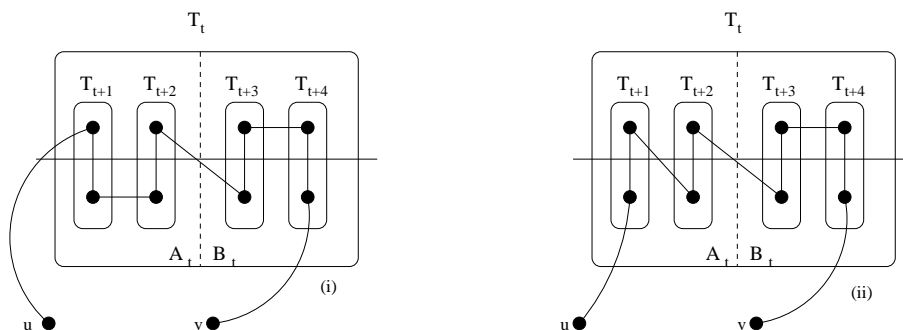


FIG. 3.11 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

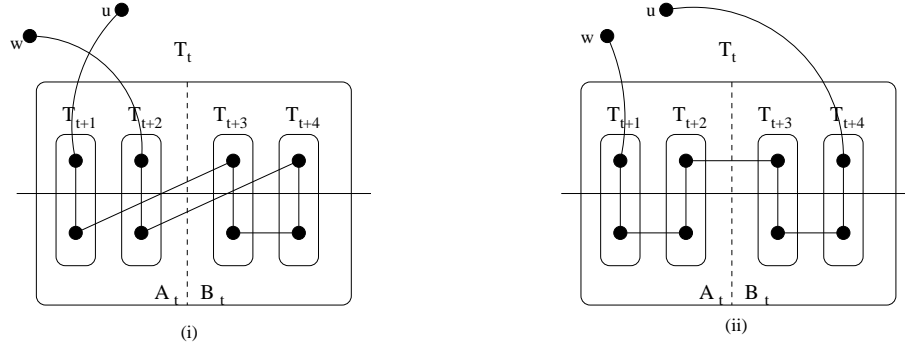


FIG. 3.12 – Intersection de cycles hamiltoniens serrés avec une dent (2,2)

La comparaison du cycle représenté par la figure 3.7(i) avec celui représenté par la figure 3.9(i) permet d'aboutir à $g_{a_4, b_4} = 1$. De la même façon on obtient $g_{a_i, b_i} = 1$ pour tout $i \in \{1, 2, 3, 4\}$.

En comparant le cycle de la figure 3.8(i) avec celui de la figure 3.12(ii) et en utilisant les valeurs déjà calculées, on obtient $g_{a_2, a_3} = 4$. On peut en déduire de la même façon la valeur des coefficients $g_{a_i, a_j} = g_{b_i, b_j} = 4$ pour tout $i \in \{1, 2\}$ et $j \in \{3, 4\}$. A présent on connaît tous les coefficients des arêtes intérieures à la dent T_t et il est facile de voir que $g_{w, a_i} = g_{w, a_1}$ pour tout $w \in H \setminus T_t$. On a donc démontré le résultat dans le cas d'une dent (2, 2).

Le cas d'une dent (3, 3) est analysé à l'aide des mêmes techniques, pour en avoir une démonstration complète consulter [23]. \square

3.2.6 Ajout de sommets

Lors de l'ajout de sommets, on suppose que les créations de sommets sont effectuées avant les duplications. Par conséquent le théorème suivant ne concerne que les configurations simples, étant donné qu'aucune opération de duplication n'a encore été effectuée.

Théorème 75 *Soit $fx \geq f_0$ une inéquation de dominos simple définissant une facette de $STSP(n)$. L'inéquation $f^*x^* \geq f_0^*$ obtenue par une création d'un sommet à partir de l'inéquation $fx \geq f_0$ définit une facette de $STSP(n+1)$.*

Démonstration Comme il y a symétrie entre H et $V \setminus H$, on peut supposer que le sommet u_{n+1} ajouté à la configuration de dominos par l'opération de duplication est situé dans le manche H . On peut aussi supposer que u_{n+1} est ajouté en-dehors de toute dent en utilisant de façon récursive le théorème 61.

Pour montrer que les opérations de duplication préservent la propriété polyédrale, on utilise le théorème 44. On définit l'ensemble F utilisé dans ce théorème de la façon

suivante :

$$F = \{e = (u, v) \in \gamma(H) : u \in T_i, v \in T_j, T_i \text{ et } T_j \text{ dents maximales, } i \neq j\}$$

Soit $e = (u, v)$ une arête de F , le coefficient de cette arête dans une configuration de dominos est égal au nombre de frontières de dents et manche traversés par celle-ci, auquel on ajoute 2 s'il s'agit d'une arête pénalisante. Comme u et v sont dans le manche, il n'est pas traversé. Comme les dents maximales auxquelles u et v appartiennent sont distinctes, le nombre de frontières de dents traversées par e est égal à la somme du nombre de frontières traversées par (u_{n+1}, u) et (u_{n+1}, v) . Par conséquent $e \in \Delta_{f^*}(u_{n+1})$, donc $F \subset \Delta_{f^*}(u_{n+1})$.

On vérifie à présent que les conditions du théorèmes 44 sont vérifiées par l'ensemble F que l'on vient d'exhiber. Comme la condition 1 de ce théorème nécessite la connaissance d'une base, on va montrer plus que cette condition, à savoir que tous les tours serrés intersectent F et non seulement les tours appartenant à une base. Soit Γ un tour serré. La configuration de dominos contient au moins trois dents maximales impaires. Au plus une dent maximale impaire n'est pas visitée proprement. Soient T_1 et T_2 deux dents maximales impaires visitées proprement par Γ , et v_1 et v_2 les extrémités dans H des chemins induits par Γ sur ces deux dents. Soient $(v_i, w_i) \in \delta(T_i) \cap \Gamma$, pour $i \in \{1, 2\}$ les arêtes de Γ qui quittent la dent T_i et sont incidentes à v_i . Ces deux arêtes peuvent ne pas être différentes, dans ce cas il s'agit de l'arête $(v_1, v_2) \in F$. Au plus une de ces deux arêtes appartient à $\delta(H)$, dans ce cas elle est l'unique arête joker possible, par conséquent l'autre appartient à F , donc $\Gamma \cap F \neq \emptyset$. La première condition du théorème 44 est satisfaite.

Soit $e = (u, v) \in F$. Soit Γ un cycle hamiltonien tel que $e \in \Gamma$ et toutes les dents de la configuration de dominos sont visitées proprement. Ce cycle est donc serré relativement à f . Alors il existe une arête joker e' allant d'une dent maximale à une autre qui appartient à Γ . De par la définition du coefficient d'une telle arête, il est facile de voir que $e' \in \Delta_{f^*}(u_{n+1})$, donc la seconde condition du théorème 44 est satisfaite.

Soient à présent $e = (u, v)$ et $e' = (u', v')$ deux arêtes distinctes de F . On souhaite montrer que l'ensemble F est f^* -connecté en u_{n+1} . Pour cela, on distingue les cas suivants :

1. Les sommets u et u' appartiennent à la même dent maximale et à des demi-dominos distincts, v et v' appartiennent à deux dents maximales différentes distinctes de la dent contenant u et u' .
2. Les sommets u et u' appartiennent au même demi-domino d'une dent maximale T , $v \in T_1$ et $v' \in T_2$; T, T_1 et T_2 sont distinctes deux à deux et toutes maximales.
3. Les sommets u, u', v, v' appartiennent à quatre dents maximales différentes.
4. Les sommets u et u' appartiennent à une même dent maximale T , les sommets v et v' à une même dent maximale $T' \neq T$.

Dans un premier temps, on considère le cas où les sommets u et u' appartiennent à la même dent maximale $T = A \cup B$, $u \in A$, $u' \in B$, $v \in T^*$, $v' \in T^{**}$, avec T, T^*, T^{**}

distinctes deux à deux. On se trouve dans les conditions du lemme 68, donc d'après ce lemme il existe un cycle hamiltonien serré contenant e et e' , ce qui prouve que ces deux arêtes sont f^* -connectées en u_{n+1} .

On analyse à présent la situation dans laquelle les sommets u et u' appartiennent au même demi-domino A de la dent maximale $T = A \cup B$, v et v' appartenant à deux dents maximales différentes, $T_1 \neq T$ et $T_2 \neq T$ respectivement. Si T est minimale, $H \cap T = \{u\} = \{u'\}$. Dans ce cas il résulte du lemme 69 qu'il existe un cycle hamiltonien serré Γ contenant e et e' . Si T n'est pas une dent minimale, soient $v \in T_1$ et $v' \in T_2$, où T_1 et T_2 sont des dents maximales. Si T_1 (respectivement T_2) est minimale, soit w_1 le sommet v (respectivement w_2 le sommet v'), sinon soit w_1 un sommet de $T_1 \cap H$ (respectivement $T_2 \cap H$) tel que w_1 (respectivement w_2) n'appartiennent pas au même demi-domino que v (respectivement v'). D'après le lemme 69 ou le lemme 68 (suivant que $v = w_1$ ou non), il existe un cycle hamiltonien serré contenant les arêtes $e = (u, v)$ et (w_1, w_2) . D'après les mêmes lemmes, il existe un cycle hamiltonien serré contenant les arêtes (w_1, w_2) et $e' = (u', v')$. Par conséquent e et e' sont f^* -connectées en u_{n+1} .

Si les sommets u, u', v, v' appartiennent à 4 dents maximales distinctes, d'après les cas précédents on sait que les arêtes $e = (u, v)$ et (v, v') sont f^* -connectées en u_{n+1} ainsi que les arêtes (v, v') et $(u', v') = e'$. Par conséquent e et e' sont f^* -connectées en u_{n+1} .

Enfin on considère le cas où les sommets u et u' appartiennent à la même dent maximale T_1 et les sommets v et v' appartiennent à la même dent maximale T_2 . Comme il existe au moins trois dents maximales dans la configuration, soit T_3 une dent maximale différente de T_1 et T_2 et soit w un sommet de T_3 . On sait que les deux arêtes $e = (u, v)$ et (v, w) sont f^* -connectées en u_{n+1} ainsi que les arêtes (v, w) et $(u', v') = e'$. Par conséquent e et e' sont f^* -connectées en u_{n+1} .

On en déduit donc que l'inéquation obtenue à l'issue d'une opération de création définit une facette si l'inéquation initiale en définit une. \square

On s'intéresse à présent au problème du remplacement d'un sommet d'une configuration simple par un ensemble de sommets. Il s'agit d'une opération de duplication. Cette opération peut être appliquée à tout sommet, même s'il a déjà été obtenu à partir d'une telle opération. Cependant, par souci de simplification, on suppose dans le théorème suivant que chaque sommet de la configuration simple est tout de suite remplacé par le nombre désiré de sommets.

Théorème 76 *Soit u_n un sommet d'un graphe à n sommets $G = (V_n, E_n)$. On considère une configuration de dominos \mathcal{D} sur G définissant une facette. On suppose que u_n est l'unique élément de sa classe. Alors la configuration de dominos \mathcal{D}^* obtenue par le remplacement de u_n par une clique de taille $k + 1$ définit une facette de $STSP(n + k)$.*

Démonstration On note $fx \geq f_0$ l'inéquation de dominos correspondant à la configuration \mathcal{D} et $f^*x^* \geq f_0^*$ celle correspondant à \mathcal{D}^* .

Le remplacement de u_n par une clique de taille $k + 1$ consiste à ajouter les sommets u_{n+1}, \dots, u_{n+k} dans la configuration de dominos \mathcal{D} exactement dans les mêmes ensembles que ceux auxquels appartient u_n . Par conséquent les arêtes (u_n, u_{n+j}) ne traversent aucune frontière des ensembles définissant la configuration \mathcal{D} , donc $f_{u_n, u_{n+j}}^* = 0$. De plus pour tout sommet $v \in V_n \setminus \{u_n\}$, les arêtes (u_n, v) et (u_{n+j}, v) traversent exactement les mêmes frontières, donc $f_{u_n, v}^* = f_{u_{n+j}, v}^*$ pour $1 \leq j \leq k$, et cela correspond exactement à la définition de l'opération de duplication.

Comme H et $V \setminus H$ jouent le même rôle, on suppose que le sommet u_n est dans le manche H . De plus en utilisant récursivement le théorème 61, on peut supposer que u_n appartient soit à une dent à la fois minimale et maximale, soit n'appartient à aucune dent.

Pour prouver que l'inéquation de dominos $f^*x^* \geq f_0$ définit une facette, on utilise le théorème 45. Afin de pouvoir conclure, il est suffisant de montrer que l'ensemble d'arêtes $\delta(u_n)$ est f -connecté. Soient $e = (u_n, v)$ et $e' = (u_n, v')$ deux arêtes distinctes de $\delta(u_n)$. On distingue les cas suivants, seul le dernier traite le cas où u_n appartient à une dent minimale, les autres traitent le cas où u_n n'appartient à aucune dent :

1. Les sommets v et v' appartiennent à deux dents maximales distinctes et au plus l'un d'entre eux n'appartient pas à H .
2. Les sommets v et v' appartiennent à la même dent maximale.
3. $v \notin H$ et $v' \notin H$.
4. Le sommet u_n appartient à une dent minimale et maximale.

Dans un premier temps, supposons que $u_n \in H \setminus (\cup_{i=1}^t T_i)$.

On considère tout d'abord le cas 1. On suppose que $v \in T, v' \in T'$ et $T \neq T'$, avec T et T' maximales. Soit \mathcal{D}' la configuration de dominos obtenue à partir des mêmes ensembles que \mathcal{D} mais sur le graphe $G \setminus \{u_n\}$. En utilisant les techniques de construction de cycles hamiltoniens des théorèmes précédents, on peut construire un cycle hamiltonien serré Γ contenant l'arête (v, v') . Comme on suppose de plus qu'un des sommets v ou v' n'est pas dans le manche, on en déduit que l'arête (v, v') est l'arête joker. Le cycle $\Gamma - (v, v') + (u_n, v') + (u_n, v)$ est serré relativement à f car $(v, v') \in \Delta_f(u_n)$. Par conséquent il existe un cycle hamiltonien serré contenant les arêtes e et e' .

Supposons à présent que les sommets v et v' appartiennent à la même dent maximale T (cas 2). Il existe une dent maximale T' différente de T . Soit $w \in T' \cap H$. D'après le cas précédent, $e = (u_n, v)$ et (u_n, w) sont f -connectées, de même que (u_n, w) et $(u_n, v' = e')$, par conséquent e et e' sont f -connectées.

Enfin on suppose que les sommets v et v' ne sont pas dans le manche (cas 3). Soit T une dent maximale. Soit $w \in T \cap H$. D'après les deux cas précédents, les arêtes $e = (u_n, v)$ et (u_n, w) sont f -connectées, de même que les arêtes (u_n, w) et $(u_n, v') = e'$. Par conséquent e et e' sont f -connectées.

Enfin on considère le cas où $\{u_n\} = T \cap H$ pour une dent minimale T (cas 4). Soient $e = (u_n, v)$ et $e' = (u_n, v')$ deux arêtes de $\delta(u_n)$. Si $e \in \gamma(T)$ et $e' \in \delta(T)$, ou bien $e \in \gamma(H)$ et $e' \in \gamma(H)$, alors les lemmes 69 et 70 permettent de conclure qu'il existe un cycle hamiltonien serré relativement à f qui contient les arêtes e et e' . On

considère le cas où $v \notin T \cup H$ et $v' \notin T \cup H$. Soit $w \in T \setminus H$. On se retrouve alors dans les conditions précédemment évoquées, c'est-à-dire qu'il existe un cycle hamiltonien serré relativement à f contenant les arêtes e et (u_n, w) , et un autre contenant les arêtes (u_n, w) et e' . Par conséquent e et e' sont f -connectées et la démonstration prend fin. \square

3.2.7 Le théorème résultat

A présent nous pouvons regrouper les différents résultats énoncés dans les sections précédentes en un théorème résultat. Ce théorème constitue une extension du résultat obtenu par Boyd et Cockburn [4]. Ces derniers ont en effet démontré que les inéquations de dominos définissent des facettes de $STSP(n)$ lorsqu'une seule dent est non minimale et que cette dent ne contient que des dents minimales.

Théorème 77 *Les inéquations de dominos définissent des facettes de $STSP(n)$, $n \geq 6$.*

Démonstration Comme chaque configuration de dominos peut être construite à partir d'une configuration de peigne minimale à 6 sommets grâce aux opérations décrites dans les sections précédentes, et comme toutes ces opérations préservent la propriété polyédrale de définition de facette, on déduit le résultat du fait que la configuration de peigne minimale à 6 sommets définit une facette de $STSP(6)$. \square

3.2.8 Séparation des contraintes de dominos

Définition 78 *Le graphe support d'une solution fractionnaire x^* d'un programme linéaire relaxé, utilisé au cours de la méthode "Branch & Cut" pour résoudre le problème du Voyageur de Commerce, est le graphe constitué des variables e telles que $x_e^* > 0$.*

Comme on sait que les contraintes de dominos définissent des facettes de $STSP(n)$, leur séparation présente un intérêt. En effet ces contraintes aident à la description du polytope $STSP(n)$ et peuvent ainsi faire progresser le processus de séparation, et repousser la limite de branchement lors de l'utilisation d'une méthode de type "Branch & Cut". L'intégration d'une nouvelle classe de contraintes définissant des facettes de $STSP(n)$ peut alors se révéler utile. Cependant deux conditions sont nécessaires pour que les contraintes de dominos permettent effectivement d'améliorer la résolution du problème du Voyageur de Commerce :

- l'existence d'un algorithme permettant de séparer ces contraintes d'une solution d'un programme linéaire relaxé
- la progression effective de la valeur de la fonction objectif du programme linéaire relaxé

Cette dernière condition ne peut être vérifiée que par l'expérience, mais elle est indispensable et il se pourrait que la valeur de la fonction objectif du programme linéaire relaxé ne progresse que très peu grâce à l'intégration de ces contraintes. Le polytope $STSP(n)$ étant constitué d'un nombre exponentiel de facettes (par exemple $STSP(10)$ est constitué de plus de 51 milliards de facettes!), le renforcement de la relaxation par certaines coupes peut n'avoir qu'un intérêt mineur.

Le résultat que l'on vient d'obtenir n'est donc qu'une première étape qui ne fait qu'indiquer que la recherche de contraintes de dominos violées au cours du processus de séparation peut avoir un intérêt dans le cadre d'une résolution par la méthode "Branch & Cut". Il est ensuite nécessaire de disposer d'algorithmes de séparation pour ces contraintes, algorithmes qui doivent être performants et efficaces. Letchford [18] a développé un tel algorithme de séparation utilisable lorsque le graphe support d'une solution du programme linéaire relaxé (graphe des arêtes à valeur strictement positive) est planaire. Cet algorithme a été amélioré par Boyd, Cockburn et Vella [5] afin de prendre en compte tous les graphes support. Ils ont testé cet algorithme sur le logiciel de résolution du problème du Voyageur de Commerce CONCORDE développé par Applegate, Bixby, Chvátal et Cook. Leur étude visait à savoir s'il était pertinent d'intégrer une recherche de contraintes de dominos dans le cadre de la séparation. Elle ne s'intéresse pas au temps de résolution du problème avec et sans ces méthodes, mais elle est prometteuse de nombreux tests exhaustifs sur les instances de la TSPLIB ont permis de montrer que le renforcement de la relaxation par ces contraintes permet d'obtenir une progression significative de la fonction objectif du programme linéaire.

3.3 Génération de facettes à partir du partitionnement du graphe en coupes minimum

3.3.1 Motivation et objectifs

On rappelle qu'étant donné le programme linéaire relaxé LP_i et sa solution fractionnaire $x^{(i)}$, la phase de séparation consiste à ajouter des contraintes valides, de préférence définissant des facettes de $STSP(n)$, qui ne sont pas satisfaites par $x^{(i)}$. Jusqu'à présent cette phase était réalisée de la manière suivante :

1. Recherche de contraintes d'élimination de sous-tours par une méthode exacte.
2. Si aucune contrainte d'élimination de sous-tours n'est trouvée, recherche de contraintes de peigne, de chemins, d'arbres de cliques, d'échelles, à l'aide d'heuristiques.

Les heuristiques utilisées dans le cadre de la séparation exploitent des techniques éprouvées développées par Naddef et Thienel dans [21] et [22]. On a vu que le premier membre des contraintes de peigne, d'arbres de cliques et d'échelles pouvait s'exprimer comme la somme des cocycles de différents ensembles. Les techniques utilisées dans les méthodes heuristiques, comme par exemple le "max-back", visent à

chercher de tels ensembles en partant d'un sommet, puis en augmentant cet ensemble de telle sorte que son cocycle soit le plus faible possible, en utilisant le graphe support de la solution fractionnaire courante. Cependant ces techniques de recherche d'ensembles sont locales : la recherche des ensembles nécessaires à la description de ces contraintes se fait sur un nombre restreint de sommets du graphe support.

Une idée d'amélioration de ces techniques est de rechercher ces contraintes de façon moins locale, en cherchant de plus grands ensembles de sommets de telle sorte que les contraintes utilisent une plus grande partie des sommets du graphe. C'est une de ces techniques qui a été développée au cours de mon travail de thèse et dont les grandes lignes sont exposées dans les sections suivantes. L'objectif est ainsi de générer des contraintes qui permettent de modifier de façon plus globale la solution fractionnaire et ainsi de générer de nouvelles contraintes permettant une résolution plus poussée lors de la phase de séparation.

3.3.2 Recherche de contraintes

La méthode développée au cours de mon travail de thèse se situe dans la continuité des travaux entrepris par Applegate et al. [1]. Elle consiste en une recherche de contraintes violées "à l'aveugle" dans le sens où aucune portion de graphe n'est privilégiée. La méthode utilisée commence par une réduction de la taille du problème étudié afin de pouvoir y appliquer une méthode exacte de recherche de contrainte. La coupe obtenue sur le graphe réduit est ensuite transformée en une coupe du graphe initial.

Plusieurs problèmes se posent pour mener à bien cette recherche de coupes :

- Réduction de la taille du problème : plusieurs possibilités sont offertes, les avantages et les inconvénients de chacune d'entre elles sont à prendre en compte.
- Méthode exacte de recherche de coupe sur le problème à taille réduite : la recherche doit être suffisamment rapide et doit de préférence mener à des coupes définissant des facettes sur le polytope associé au problème considéré.
- Transformation de la coupe trouvée à l'étape précédente en coupe sur le problème initial : il existe de nombreux outils qui permettent de réaliser cette étape sans trop de difficultés.

Nous développons ces différentes étapes dans les sections suivantes de façon à décrire le plus complètement possible la méthode implémentée, ses avantages et ses inconvénients.

3.3.3 Réduction de la taille du problème

Afin de simplifier la recherche de contraintes, la première étape consiste à réduire la taille du problème. Dans notre cas, il s'agit de diminuer le nombre de sommets du graphe support de la solution d'un programme linéaire relaxé obtenu au cours de la résolution du problème du Voyageur de Commerce par la méthode Branch & Cut. Pour cela on se sert des valeurs de chaque arête de ce graphe qui correspondent aux valeurs des variables associées. Dans [28], Padberg et Rinaldi montrent que deux

opérations de réduction du nombre de sommets du graphe peuvent s'effectuer sans perdre de coupes. Il s'agit de la réduction de chaînes de valeur 1 en une arête et de la réduction "triangulaire". Ces deux formes de réduction sont présentées en figure 3.13. On constate que le nombre de sommets du graphe support est réduit par ces opérations et ainsi le nombre d'arêtes, donc de variables.

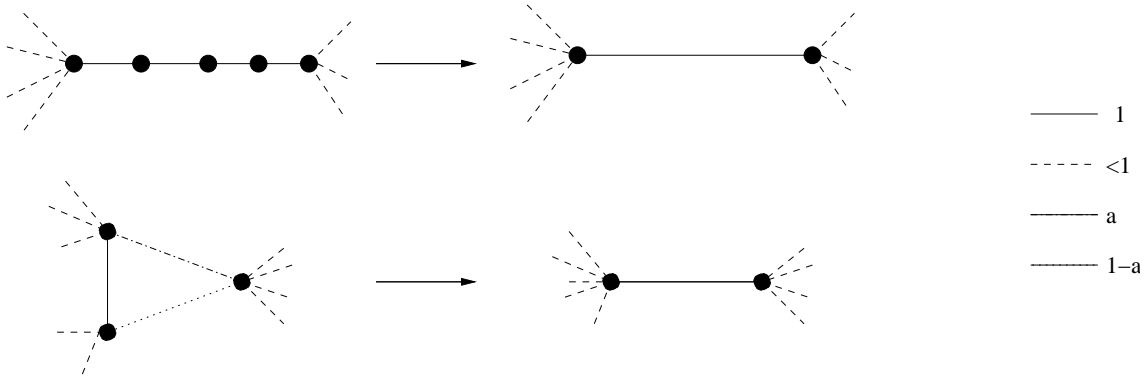


FIG. 3.13 – Réductions du graphe support d'une solution

Ces réductions permettent en général de réduire considérablement le nombre de sommets du graphe analysé (d'un facteur 2 à 10), mais il reste dépendant de la taille initiale du graphe et de la solution représentée par celui-ci. Il est plus intéressant de disposer d'une méthode qui réduit le graphe initial en un graphe d'une taille définie à l'avance, d'autant plus que nous verrons que les méthodes utilisées, étant exactes sur le graphe réduit, nécessitent des graphes de petite taille (moins de 35 sommets).

Applegate et al. proposent dans [1] de contracter des ensembles de sommets ayant un cocycle égal à 2. Pour cela ils procèdent localement en partant d'un sommet puis en ajoutant des sommets jusqu'à obtenir un ensemble V_i de cocycle égal à 2. A l'issue de l'opération de réduction, chaque ensemble V_i est contracté en un sommet de degré 2 et $V \setminus \cup_i V_i$ est contracté en un sommet particulier de degré supérieur à 2. Le graphe réduit compte environ 30 sommets. Une coupe est ensuite recherchée sur ce graphe à l'aide de tours particuliers.

Nous avons choisi de réduire la taille du problème en utilisant les partitions du graphe support en ensembles de cocycle de valeur 2. Cette réduction est assez intuitive. Soit V_n l'ensemble de sommets du graphe support. Si on dispose d'une partition des sommets de V_n en k ensembles E_1, \dots, E_k et que la valeur du cocycle de chaque ensemble $E_i, i = 1, \dots, k$ vaut 2, alors la réduction de ce graphe est un graphe G' à k sommets et dont la valeur $x'_{i,j}$ de chaque arête (i, j) est calculée de la façon suivante : $x'_{i,j} = \sum_{e \in (E_i: E_j)} x_e^*$ où x^* est le vecteur des coefficients des arêtes du graphe support initial.

Il reste à décrire de quelle façon l'ensemble V est partitionné en k sous-ensembles dont le cocycle a pour valeur 2. Pour cela on se sert de la représentation des coupes minimum d'un graphe en cactus de Dinic et al. [10] (voir aussi Fleischer [11] et De Vitis [35]), ainsi que du calcul effectif de ces coupes développé en C par Wenger [36].

On aborde sommairement les principaux résultats de cette étude dans le cadre du graphe support d'un problème du Voyageur de Commerce, des compléments peuvent être trouvées dans les références associées aux auteurs qui viennent d'être cités.

Définition 79 Si V l'ensemble de sommets d'un graphe et X un sous-ensemble de V , on note \bar{X} l'ensemble de sommets complémentaire de X par rapport à V : $\bar{X} = V \setminus X$.

Définition 80 Soit G un graphe pondéré. Deux coupes $\delta(X)$ et $\delta(Y)$ de G sont **croisées** si aucun des ensembles $X \cap Y, \bar{X} \cap Y, \bar{X} \cap \bar{Y}, X \cap \bar{Y}$ n'est vide.

Dans la suite de cette section, $c(X : Y)$ désigne la somme des poids des arêtes de $(X : Y)$.

Lemme 81 Soient $\delta(X)$ et $\delta(Y)$ deux coupes minimum croisées d'un graphe pondéré dont la valeur d'une coupe minimum est λ , et $V_1 = X \cap Y, V_2 = \bar{X} \cap Y, V_3 = \bar{X} \cap \bar{Y}$ et $V_4 = X \cap \bar{Y}$. Alors :

$$c(V_1 : V_3) = c(V_2 : V_4) = 0$$

et

$$c(V_1 : V_2) = c(V_2 : V_3) = c(V_3 : V_4) = c(V_4 : V_1) = \lambda/2$$

Un corollaire immédiat de ce lemme est que si $\delta(X)$ et $\delta(Y)$ sont des coupes croisées, alors $\delta(X \cup Y)$ et $\delta(X \cap Y)$ le sont aussi.

On introduit à présent des familles de sous-ensembles de V généralisant les propriétés des sous-ensembles V_1, V_2, V_3 et V_4 du lemme 81.

Définition 82 On appelle **partition circulaire** de G toute partition de V en $k \geq 3$ sous-ensembles non vides V_1, \dots, V_k pouvant être ordonnés de la façon suivante :

- $c(V_i, V_j) = 1$ si $|i - j| = 1$ ou $|i - j| = k - 1$, $c(V_i, V_j) = 0$ sinon
- si A est de la forme $A = \cup_{h=i}^j V_h$ ou $A = (\cup_{h=1}^i V_h) \cup (\cup_{h=j}^k V_h)$ avec $1 \leq i < j \leq k$, alors $\delta(A)$ est une coupe minimum de G et pour tout $\delta(B)$ qui n'est pas de cette forme il existe $i \in \{1, \dots, k\}$ tel que $B \subseteq V_i$ ou $\bar{B} \subseteq V_i$

On introduit à présent une notion de représentation assez générale dont les propriétés vont permettre d'aboutir à une représentation assez simple des coupes minimum d'un graphe.

Définition 83 Soit G un graphe pondéré. On dit que le graphe pondéré $H(G)$ **représente** G s'il existe deux applications $f : V \rightarrow V(H(G))$ et $g : V(H(G)) \rightarrow 2^V$ vérifiant les propriétés suivantes :

- $g(u) = \{i \in V : f(i) = u\} = f^{-1}(u)$ pour $u \in V(H(G))$
- si $\delta(X)$ est une coupe minimum de $H(G)$, alors $\delta(g(X))$ est une coupe minimum de G

- pour toute coupe minimum $\delta(S)$ de G , il existe $X \subseteq V(H(G))$, vérifiant $g(X) = S$ tel que $\delta(X)$ est une coupe minimum de $H(G)$.

Cette définition indique seulement que l'application f n'a pas besoin d'être surjective, c'est-à-dire qu'il peut exister des sommets $u \in H(G)$, appelés *sommets vides* tels que $g(u) = \emptyset$. Une conséquence de cette définition est qu'une coupe minimum S du graphe G peut correspondre à plusieurs coupes du graphe $H(G)$ dans lequel certains sommets vides peuvent appartenir à plusieurs ensembles définissant des coupes. L'application peut aussi ne pas être injective, il est donc possible d'avoir $|g(u)| > 1$ pour un sommet $u \in V(H(G))$.

Les coupes minimum d'un graphe pondéré G peuvent être représentées par un graphe $H(G)$ ayant une structure assez simple, que l'on appelle **cactus associé** à G .

Définition 84 *Un cactus associé à un graphe G est un graphe pondéré représentant G dans lequel toute arête appartient à au plus un cycle. Il existe donc deux types d'arêtes dans un cactus : les arêtes n'appartenant à aucun cycle, appelées arêtes d'arbre, et les arêtes appartenant à exactement un cycle, appelées arêtes de cycle. Dans un cactus, les arêtes d'arbre ont toutes le même poids, double de celui des arêtes de cycle.*

Théorème 85 (Naor et Vazinari [24]) *Les coupes minimum de tout graphe support G d'un problème de Voyageur de Commerce peuvent être représentées par un cactus C .*

On déduit directement de la définition que les coupes minimum d'un cactus peuvent être obtenues par la suppression d'une arête d'arbre ou de deux arêtes de cycle appartenant au même cycle.

Les sommets d'un cactus seront désormais appelés nœuds, de telle sorte que lorsqu'on utilise le terme *sommet*, il s'agit d'un sommet du graphe G , et lorsqu'on utilise le terme *nœud*, il s'agit d'un sommet du cactus qui représente les coupes de G . On dit d'un nœud u qu'il est *vide* si $g(u) = \emptyset$.

On peut à présent préciser un lien qui existe entre les partitions circulaires d'un graphe pondéré et sa représentation en cactus. Si le cactus C associé à un graphe pondéré G contient un cycle u_1, \dots, u_k , le fait d'enlever les arêtes de ce cycle déconnecte C en k composantes connexes $Z_i (i = 1, \dots, k)$ et aucune de ces composantes ne peut être constituée uniquement de nœuds vides (sinon elle peut être supprimée). Si on note $V_i = g(Z_i)$ pour $i = 1, \dots, k$, on peut constater que V_1, \dots, V_k est une partition circulaire de G . En outre le longueur du cycle est égale au nombre d'éléments de la partition circulaire. Par conséquent si G ne possède pas de partition circulaire, tout cactus représentant G est un arbre. D'autre part deux cycles distincts d'un cactus ne peuvent pas représenter la même partition circulaire.

On peut facilement vérifier que la représentation en cactus d'un graphe G n'est pas unique. Les cactus peuvent être simplifiés de deux façons. La première consiste à supprimer un nœud vide incident à deux arêtes d'arbre et remplacer ces deux

arêtes par une seule arête d'arbre. La seconde simplification consiste à supprimer un nœud vide v_0 incident à deux arêtes de cycle (v_0, v_1) et (v_0, v_2) et à une arête d'arbre (v_0, v_3) , et à remplacer ces arêtes par deux arêtes de cycle (v_1, v_3) et (v_2, v_3) . Ces simplifications sont illustrées par la figure 3.14. On appelle **cactus simple** un cactus dans lequel aucune de ces deux opérations ne peut être réalisée. Ces réductions ne sont possibles que si aucune autre arête que celles présentées n'est adjacente aux nœuds vides à supprimer.

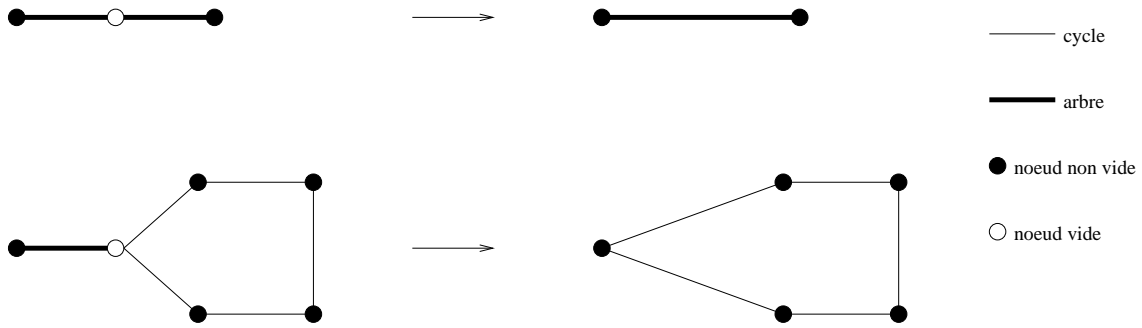


FIG. 3.14 – Simplification d'un cactus

Dans le cas des cactus simples, on dispose de résultats qui permettent de limiter la redondance de la représentation des coupes. C'est l'objet du théorème suivant (voir aussi [35] et [11]).

Théorème 86 *Soient G un graphe pondéré et C un cactus simple le représentant. Alors toute coupe minimum $\delta(S)$ de G correspond à au plus deux coupes minimum de C .*

Corollaire 87 *Si une coupe $\delta(S)$ d'un graphe G est représentée deux fois dans un cactus simple représentant G , alors il existe deux partitions circulaires V_1, \dots, V_k et V'_1, \dots, V'_k et deux entiers i et j tels que $S = V_i$ et $\bar{S} = V'_j$.*

On déduit de ce corollaire que le seul cas où une coupe est représentée deux fois dans un cactus simple est lorsqu'il existe un nœud vide de degré 4 commun à exactement deux cycles. On sait donc qu'un cycle dans un cactus correspond à une partition circulaire, le théorème suivant explicite la représentation des partitions circulaires dans un cactus.

Théorème 88 *Toute partition circulaire d'un graphe pondéré G possédant au moins 4 éléments correspond à un cycle dans tout cactus représentant G .*

Les seules partitions circulaires qui peuvent ne pas être représentées par un cycle dans un cactus sont les partitions à 3 ensembles. Cependant il existe des transformations qui permettent de les représenter en un cycle. On peut donc définir la représentation canonique d'un cactus.

Définition 89 (De Vitis [35]) *On appelle **représentation en cactus canonique** d'un graphe G le cactus simple C dans lequel toute partition circulaire de G est représentée par un cycle. Cette représentation est unique.*

Dans le cadre du problème du Voyageur de Commerce, la valeur d'une coupe minimum d'un graphe support d'une solution satisfaisant les contraintes de degré et de sous-tours est connue : elle vaut 2. Par conséquent le cactus canonique représentant un tel graphe aura des arêtes ayant pour valeur 1 s'il s'agit d'arêtes de cycle et 2 s'il s'agit d'arêtes d'arbre. Comme chaque sommet du graphe support est une coupe minimum et que toute coupe minimum est représentée par un cactus canonique, on en déduit que les nœuds du cactus associé à un tel graphe correspondent à au plus un sommet du graphe initial. Un exemple de représentation en cactus d'un graphe support d'une solution à un programme linéaire utilisé au cours de la méthode Branch & Cut est représenté par les figures 3.15 (graphe support) et 3.16 (cactus associé). Les arêtes de valeur 2 sont représentées par une ligne continue dans la figure 3.16.

De nombreux algorithmes ont été développés pour réaliser le cactus représentant les coupes minimum d'un graphe (voir [11] et [36]). Dans le cadre de ma thèse, j'ai utilisé les procédures développées en C par Klaus Wenger [36]. Ces procédures permettent la construction et la manipulation du cactus canonique associé à un graphe. En particulier ces procédures permettent d'obtenir, pour un entier k donné, l'ensemble des partitions des sommets du graphes en k coupes minimales. Entre temps Wenger a développé de nouvelles procédures plus rapides de construction du cactus associé à un graphe support d'une solution obtenue dans le cadre de la méthode Branch & Cut exploitant les propriétés des cactus associés à de tels graphes. Les procédures de Wenger ont l'avantage d'être présentées sous forme de librairie de telle sorte qu'elles puissent être utilisées facilement dans tout cadre de développement et elles permettent de savoir facilement si une coupe est représentée deux fois dans un cactus. Ainsi les réductions du graphe support en utilisant les partitions en coupes minimum de ce graphe peuvent s'opérer sans aboutir deux fois sur la même réduction.

Afin d'illustrer la méthode utilisée, on développe parallèlement à l'aspect théorique de la méthode utilisée un exemple issu de l'instance **pr439** à 439 sommets. Le graphe de la figure 3.17 est une partie du graphe support de la solution obtenue par l'application de la méthode Branch & Cut jusqu'à ce que les méthodes traditionnelles ne permettent plus de trouver de contraintes. Il deviendrait alors nécessaire d'effectuer un branchement. Le partitionnement du graphe à l'aide de sa représentation en cactus permet de trouver une partition en 13 coupes minimum qui sont représentées sur la figure.

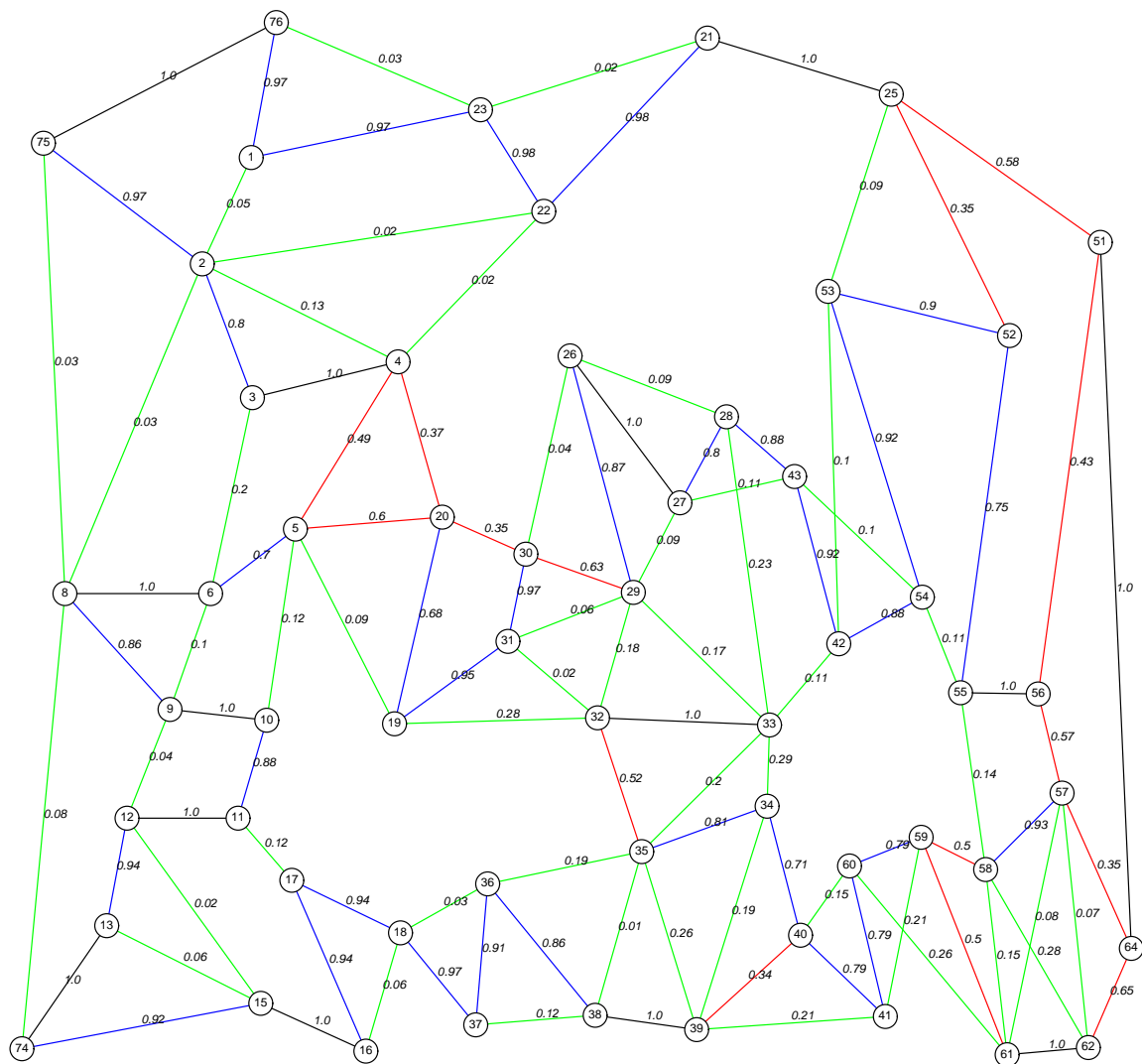


FIG. 3.15 – Graphe support (contracté) d'une solution de l'instance pr76

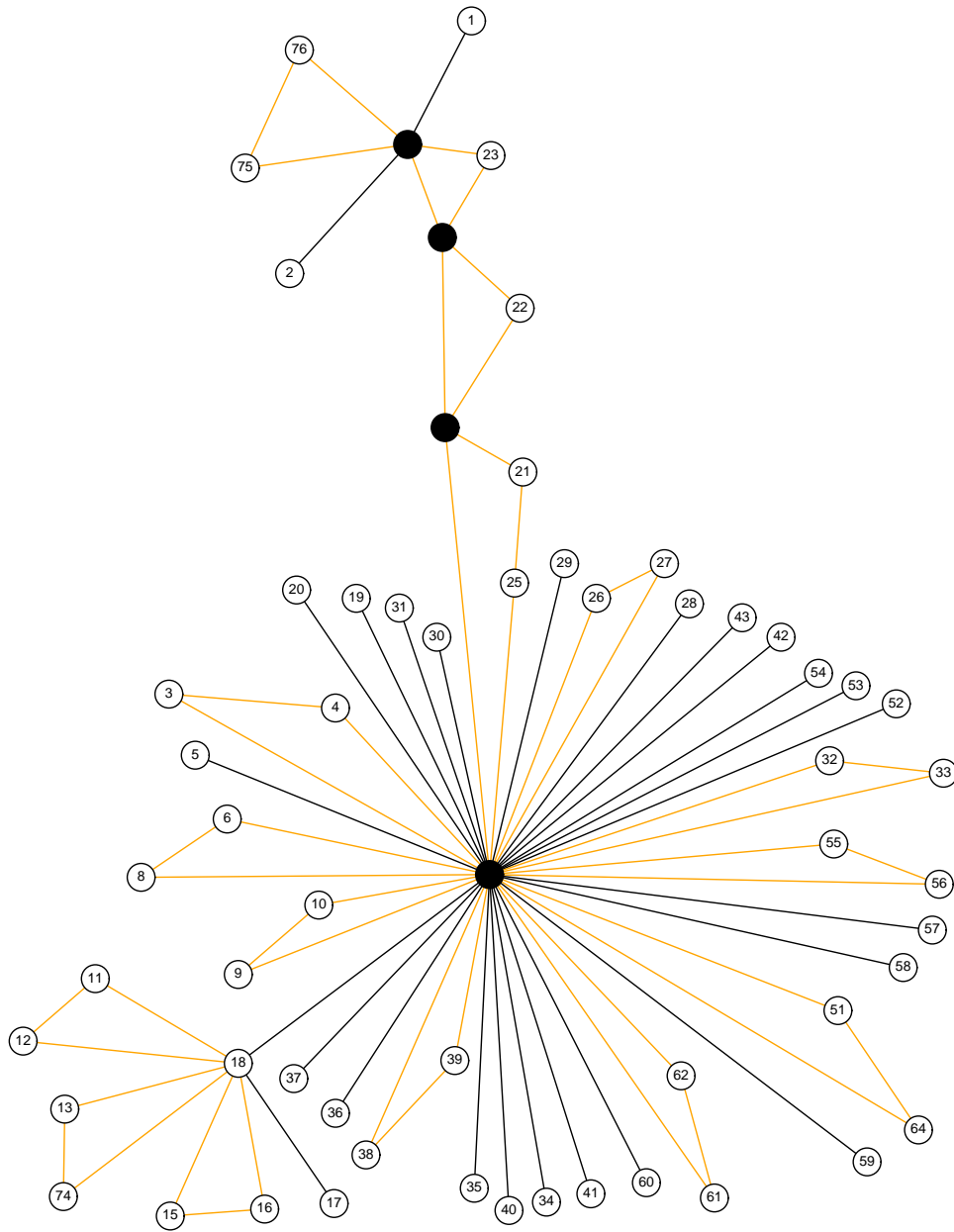


FIG. 3.16 – Cactus associé au graphe support présenté en figure 3.15

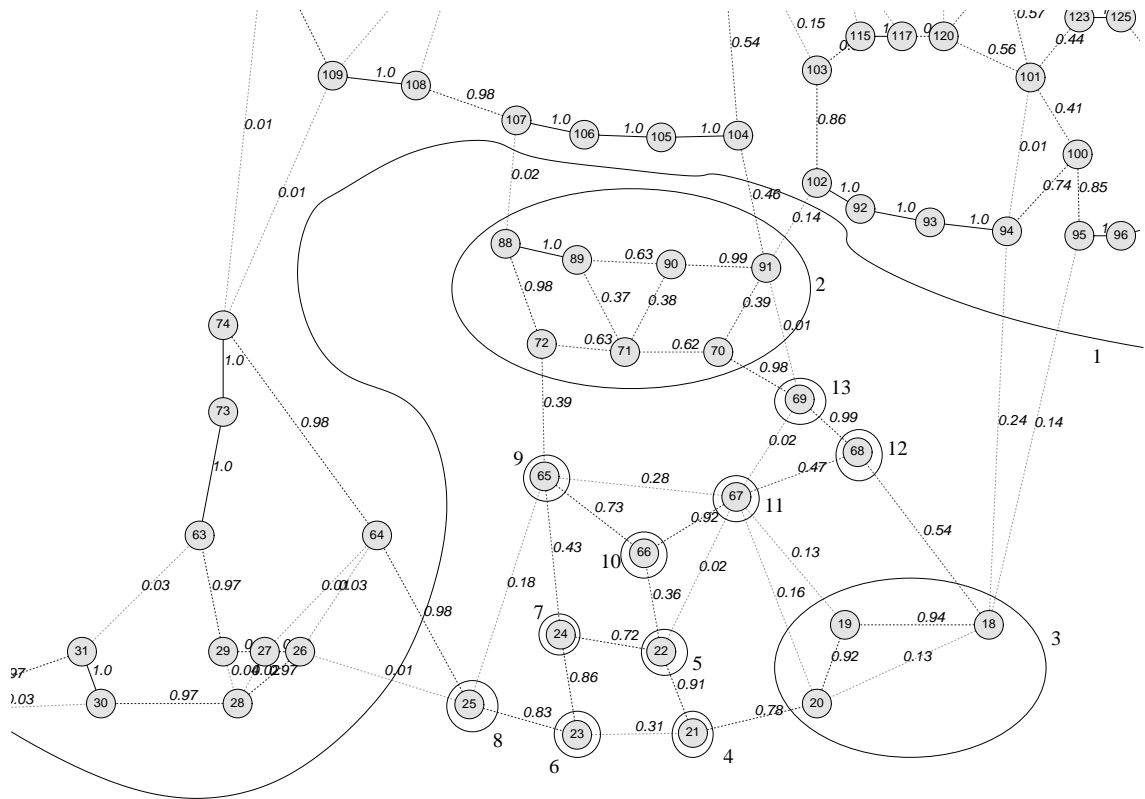


FIG. 3.17 – Partition du graphe support d'une solution en 13 coupes minimum

3.3.4 Génération de facettes à l'aide des coupes minimum du graphe support

3.3.4.1 Recherche d'une contrainte violée sur le graphe réduit

On considère le graphe G^* support d'une solution x^* d'un programme linéaire utilisé au cours de la méthode Branch & Cut qui satisfait les contraintes de degré et de sous-tours. La valeur d'une coupe minimum de G^* est 2. On utilise la représentation des coupes minimum de G^* en cactus afin de déterminer une partition de ce graphe en k sous-ensembles de sommets E_1, \dots, E_k ayant chacun un cocycle de valeur 2. Le graphe G' obtenu en réduisant chacun des sous-ensembles E_i en un sommet i et en déterminant la valeur d'une arête (i, j) comme étant la somme des valeurs des arêtes de $(E_i : E_j)$ est un graphe support d'une solution d'un programme linéaire relaxé. On note x' le vecteur solution associé à ce graphe. Ce graphe n'est en général pas complet.

La figure 3.18 représente le graphe obtenu en contractant chaque coupe minimum du graphe support de l'exemple étudié en un sommet. Le calcul de la valeur de l'arête (i, j) est bien la somme des valeurs des arêtes de (E_i, E_j) .

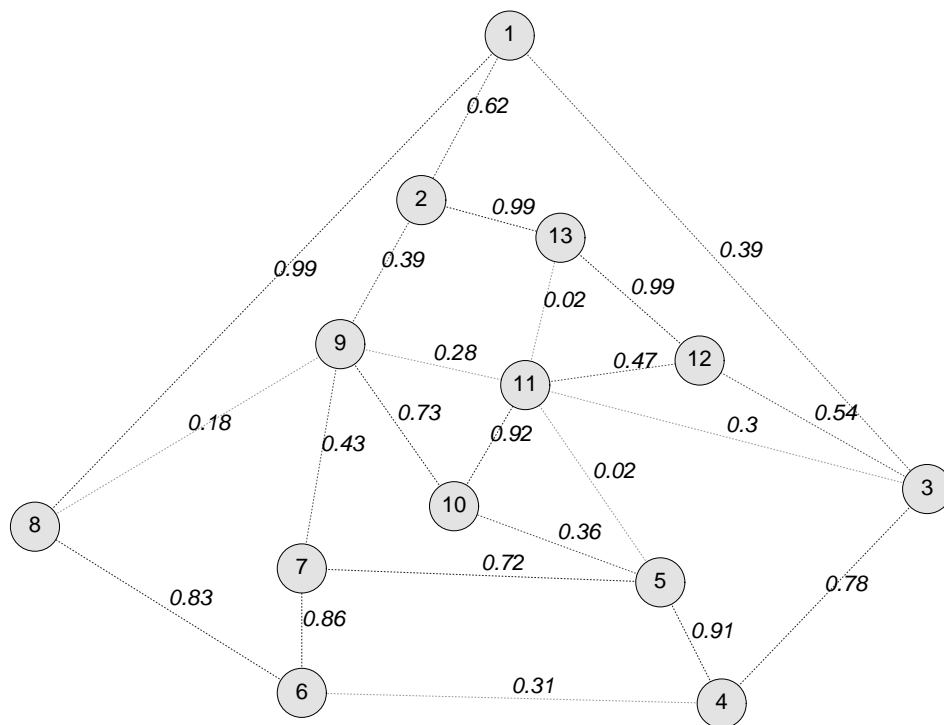


FIG. 3.18 – Graphe résultat obtenu suite à la contraction des coupes minimum

On cherche dans un premier temps à déterminer une contrainte définissant une facette séparant x' de $STSP(k)$, s'il en existe une. S'il n'existe pas de telle contrainte, alors x' est combinaison convexe de cycles hamiltoniens, donc la réduction n'aboutit

pas. Il est important de noter que la réduction du graphe initial se fait avec une perte d'information, c'est-à-dire qu'une contrainte violée pour le graphe support initial ne correspond pas forcément à une contrainte violée sur le graphe réduit. Pour s'en convaincre, il suffit par exemple de s'imaginer une contrainte dont le support est situé dans un des ensembles E_i : elle ne correspond à aucune contrainte violée par x' .

On note $vx' \geq v_0$ l'inéquation recherchée définissant une coupe séparant $STSP(k)$ de x' , v étant un vecteur d'entiers indexé par les arêtes de G' et v_0 un entier. Le vecteur v et l'entier v_0 sont les inconnues. Il paraît irréaliste de générer l'ensemble des cycles hamiltoniens de G' et de chercher une inéquation satisfaite par cet ensemble et violée par x' , même si cette méthode peut fonctionner pour de petits graphes (en général $k \leq 12$ car G' n'est pas complet). C'est pourquoi on utilise un sous-ensemble de cycles hamiltoniens $\mathcal{T} = \{\Gamma_1, \dots, \Gamma_s\}$. On ne recherche pas une contrainte satisfaite par tous les cycles hamiltoniens du graphe, mais uniquement par les cycles hamiltoniens de \mathcal{T} . S'il existe un cycle hamiltonien qui ne satisfait pas cette contrainte, il est ajouté à \mathcal{T} et une nouvelle contrainte est calculée. Ce processus continue jusqu'à ce que la contrainte soit valide. L'ensemble \mathcal{T} ne comporte qu'un cycle hamiltonien au départ, choisi de manière arbitraire, puis est complété au fur et à mesure jusqu'à obtenir une contrainte valide et violée, ou la certitude que le graphe est combinaison convexe de cycles hamiltoniens.

Pour trouver une inéquation satisfaite par les cycles hamiltoniens de \mathcal{T} et violée par x' , on résout un programme linéaire, dont les variables sont les composantes de l'inéquation. On ajoute des variables d'écart d_1, \dots, d_s pour chaque cycle hamiltonien de \mathcal{T} . On note x^{T_1}, \dots, x^{T_s} les vecteurs de $\mathbb{R}^{E'}$ associés aux cycles hamiltoniens de \mathcal{T} . Pour trouver les coefficients de l'inéquation linéaire satisfaite par tous les tours de \mathcal{T} , on résout alors le programme linéaire suivant :

$$\begin{array}{ll}
 \text{minimiser} & \sum_e c_e \\
 \text{sous les contraintes} & -cx^* + c_0 \geq 1 \\
 & cx^{T_1} - c_0 - d_1 = 0 \\
 & \vdots \\
 & cx^{T_s} - c_0 - d_s = 0 \\
 & c_e \geq 0, e \in E' \\
 & c_0 \geq 0 \\
 & d_i \geq 0, 1 \leq i \leq s
 \end{array}$$

Ce programme linéaire permet d'obtenir une inéquation $cx' \geq c_0$ valide sur l'ensemble \mathcal{T} et violée par x^* . Si une telle inéquation n'existe pas, le vecteur x^* est combinaison convexe des vecteurs x^{T_1}, \dots, x^{T_s} , donc il n'existe pas de coupe sur G' et l'algorithme s'arrête là. Si une telle inéquation existe, elle est éventuellement convertie en inéquation à coefficients entiers. Si cette inéquation n'est pas valide pour l'ensemble des cycles hamiltoniens du graphe G' , il existe un cycle hamiltonien $T_{s+1} \notin \mathcal{T}$ tel que $cx^{T_{s+1}} < c_0$. Un tel cycle peut être trouvé en résolvant un problème de Voyageur de Commerce sur G' en donnant pour longueur à chaque arête e son

coefficient dans l'inéquation c_e . Ainsi ce cycle est ajouté à \mathcal{T} et on réitère le processus jusqu'à ce que l'inéquation trouvée soit valide ou que G' soit une combinaison convexe de cycles hamiltoniens.

A l'issue de cette étape on dispose donc d'une inéquation valide définissant une coupe ou bien de la certitude que x^* est combinaison convexe de vecteurs associés à des cycles hamiltoniens.

On détaille à présent cette étape sur l'exemple que l'on étudie depuis le début de cette section. On rappelle qu'on cherche une inéquation valide sur le graphe support G' de la figure 3.18 mais qui ne soit pas satisfaite par la solution x^* représentée sur ce graphe. On procède donc en suivant les étapes suivantes :

- Le cycle hamiltonien [1-3-4-6-8-9-7-5-10-11-12-13-2-1] peut être obtenu en utilisant les arêtes de G' . L'ensemble \mathcal{T} est initialisé avec ce cycle hamiltonien. L'inéquation $x_{8,9} \geq 1$ est satisfaite par tous les cycles de \mathcal{T} mais est violée par x^* .
- Le cycle hamiltonien [1-2-13-11-12-3-4-5-10-9-7-6-8-1] est obtenu en utilisant les arêtes de G' et ne satisfait pas l'inéquation $x_{8,9} \geq 1$. Par conséquent on ajoute ce cycle dans \mathcal{T} et on cherche une inéquation satisfaite par les deux cycles de cet ensemble : on trouve $x_{5,10} \geq 1$, inéquation violée par x^* .
- Le cycle hamiltonien [1-2-13-12-11-10-9-8-6-7-5-4-3-1] ne satisfait pas l'inéquation $x_{5,10} \geq 1$ alors qu'il ne comprend que des arêtes de G' . On l'ajoute à \mathcal{T} et on calcule une inéquation satisfaite par les trois cycles de \mathcal{T} et violée par x^* : on obtient $x_{11,12} \geq 1$.
- Le cycle hamiltonien [1-2-9-10-11-13-12-3-4-5-7-6-8-1] est ensuite ajouté à \mathcal{T} car il ne satisfait pas l'inéquation précédente. Une inéquation satisfaite par les cycles de \mathcal{T} et violée par x^* est alors $x_{8,9} + x_{11,13} \geq 1$.
- Le cycle hamiltonien [1-3-4-5-10-11-12-13-2-9-7-6-8-1] est ajouté à \mathcal{T} . L'inéquation trouvée est alors $x_{1,3} + x_{11,13} \geq 1$.
- Le cycle hamiltonien [1-2-13-12-3-4-6-7-5-10-11-9-8-1] est ajouté à \mathcal{T} . L'inéquation trouvée est alors $x_{3,4} \geq 1$.
- L'inéquation précédente n'est pas satisfaite par le cycle hamiltonien [1-2-13-12-3-11-10-5-4-6-7-9-8-1], ce cycle est donc ajouté à \mathcal{T} . Une inéquation satisfaite par tous les cycles de \mathcal{T} et violée par x^* est alors $x_{1,3} + x_{2,9} + x_{5,10} + 2x_{8,9} + 2x_{11,13} \geq 3$.
- Le cycle hamiltonien [1-2-13-12-3-11-10-9-7-5-4-6-8-1] est ajouté à \mathcal{T} car il ne satisfait pas l'inéquation précédente. Une nouvelle inéquation satisfaite par tous les cycles de \mathcal{T} et violée par x^* est calculée : il s'agit de $x_{7,9} + x_{8,9} + x_{11,13} \geq 1$.

L'inéquation $x_{7,9} + x_{8,9} + x_{11,13} \geq 1$ est satisfaite par tous les cycles hamiltoniens ayant pour support le graphe G' . En effet la résolution du problème du voyageur de commerce sur G' avec les arêtes (7, 9), (8, 9) et (11, 13) ayant une longueur égale à 1 et les autres arêtes de G' ayant une longueur égale à 0 donne pour résultat un cycle hamiltonien de longueur 1. On a donc exhibé une inéquation valide sur G' violée par x^* . La violation est par ailleurs assez importante puisqu'elle vaut 0.37. Cependant la dimension de la face de $STSP(G')$ induite par cette inéquation n'est

pas connue à ce stade, elle nécessiterait des investigations supplémentaires. Cinq cycles hamiltoniens de \mathcal{T} satisfont cette inéquation à égalité et on vérifie qu'ils sont affinement indépendants. La face définie par l'inéquation valide est alors au moins de dimension 4. Or une facette d'un graphe non complet a pour dimension au plus $|E'| - |V'| - 1$, c'est-à-dire 9 dans le cas qui nous concerne. Il se peut que cette valeur soit plus faible, il faudrait pour cela calculer le nombre maximum de cycles hamiltoniens indépendants que possède un tel graphe. Par conséquent il existe une incertitude à ce stade de l'algorithme. On ne peut qu'affirmer que l'inéquation est valide, donc elle définit une face de $STSP(G')$. Cependant la marge d'incertitude concernant la dimension de cette face va être réduite par les étapes suivantes de l'algorithme.

3.3.4.2 Extension au graphe complet

Dans le cas où une coupe a été trouvée, il est nécessaire de l'étendre au graphe complet K_k . Pour cela on utilise la technique de l'ajout séquentiel. Cette technique doit son origine aux travaux de Gomory [13] et a été principalement élaborée par Balas [2], Hammer, Johnson et Peled [17], Padberg [26] [27] et Wolsey [37] [38].

L'ajout séquentiel consiste à transformer le graphe G' en un graphe complet en y ajoutant les arêtes manquantes. Ces arêtes sont ajoutées séquentiellement, une après l'autre, dans un ordre arbitraire, et pour chaque graphe intermédiaire, une inéquation valide pour ce graphe est générée. On procède de la façon suivante. On choisit une arête e'' n'appartenant pas au graphe G' . On l'ajoute à G' pour obtenir un graphe G'' . L'inéquation $cx' \geq c_0$ valide sur G' ne l'est pas sur G'' car le coefficient $c_{e''}$ n'est pas défini. Pour trouver la valeur de ce coefficient, on résout à nouveau un problème de type Voyageur de Commerce sur le graphe G'' , en utilisant les longueurs c_e sur les arêtes de G' et la longueur 0 sur l'arête e'' et en forçant l'utilisation de e'' . En d'autres termes on cherche le cycle hamiltonien de longueur minimum contenant e'' . Ce cycle a pour longueur l . On en déduit alors le coefficient $c_{e''}$ qui vaut $c_0 - l$. On vient donc d'exhiber une inéquation valide sur G'' , car tout cycle hamiltonien Γ de G'' vérifie $cx^\Gamma \geq c_0$. D'autre part la dimension de la coupe générée par cette inéquation a aussi été augmentée de 1, car Γ est un cycle hamiltonien indépendant de \mathcal{T} . En effet l'arête e'' appartient à Γ et à aucun autre élément de \mathcal{T} . Pour obtenir une inéquation valide sur le graphe complet ayant pour sommets V' , on réitère le processus en partant de G' et en remplaçant \mathcal{T} par $\Gamma \cup \mathcal{T}$.

A l'issue de cette phase une inéquation valide est exhibée pour le graphe complet ayant V' pour ensemble de sommets.

Dans le cas de notre exemple, l'inéquation $cx \geq c_0$ valide pour le graphe support de la figure 3.18 est transformée en une inéquation valide pour le graphe complet correspondant. On commence par ajouter l'arête (1, 4) au graphe G' . Le cycle hamiltonien de longueur minimum utilisant les arêtes de G' et contenant l'arête (1, 4) est [1-4-3-12-13-2-9-10-11-5-7-6-8-1]. Comme ce cycle n'utilise pas les arêtes (7, 9), (8, 9) ou (11, 13), il est de longueur nulle. Par conséquent le coefficient de l'arête (1, 4) dans l'inéquation est $1 - 0 = 1$. On procède de même pour les autres

arêtes manquantes du graphe et on obtient l'inéquation suivante :

$$x_{1,4} + x_{1,5} + x_{1,7} + x_{1,9} + x_{1,10} + x_{1,11} + x_{1,12} + x_{1,13} + x_{2,4} + x_{2,5} + x_{2,7} + x_{2,8} + x_{3,5} + x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} + x_{3,13} + x_{4,5} + x_{4,7} + x_{4,8} + x_{4,9} + x_{4,10} + x_{4,11} + x_{4,12} + x_{4,13} + x_{5,7} + x_{5,8} + x_{5,9} + x_{5,10} + x_{5,11} + x_{5,12} + x_{5,13} + x_{7,8} + x_{7,9} + x_{7,10} + x_{7,11} + x_{7,12} + x_{7,13} + x_{8,9} + x_{8,10} + x_{8,11} + x_{8,12} + x_{8,13} + x_{9,12} + x_{9,13} + x_{10,12} + x_{10,13} + x_{11,13} \geq 3.$$

Lors du processus d'ajout séquentiel d'arêtes, la recherche d'un cycle hamiltonien de longueur minimum utilisant l'arête (5, 6) a donné un cycle de longueur 2. Afin d'éviter d'obtenir une inéquation comportant des coefficients négatifs, on ajoute une contrainte de degré à l'inéquation, de telle sorte que le membre de droite ne vaut plus 1, mais 3. Afin d'éviter d'ajouter de trop nombreuses inéquations de degré, si une arête nécessite un tel ajout, les autres arêtes restant à ajouter sont analysées, puis les arêtes posant problème. En effet les coefficients de l'inéquation obtenue par la méthode de l'ajout séquentiel dépendent de l'ordre dans lequel sont ajoutées les arêtes.

Comme le calcul de chaque coefficient se fait en cherchant un cycle hamiltonien indépendant des précédents, l'ensemble \mathcal{T} contient à présent 55 cycles hamiltoniens indépendants supplémentaires, donc un total de 60 cycles hamiltoniens indépendants. La dimension de la face induite par l'inéquation générée vaut donc au moins 59. Sachant que la dimension d'une facette du polytope associé au graphe complet vaut 64, deux possibilités s'offrent à nous :

- l'inéquation que l'on vient d'exhiber définit une facette de $STSP(G')$: pour en avoir la confirmation, il est nécessaire d'ajouter cinq cycles hamiltoniens dans \mathcal{T} de telle sorte que \mathcal{T} soit un ensemble de cycles indépendants et que les cycles de \mathcal{T} vérifient l'inéquation à égalité.
- L'inéquation que l'on vient d'exhiber ne définit pas une facette de $STSP(G')$: elle est alors contenue dans une facette de $STSP(G')$ et il est nécessaire de définir une telle facette.

A présent on cherche à obtenir une inéquation définissant une facette de $STSP(k)$.

3.3.4.3 Transformation en facette

L'inéquation trouvée précédemment ne définit pas forcément une facette de K_k . En effet une inéquation définissant une facette de K_k est satisfaite à égalité par $\frac{k(k-1)}{2} - k$ cycles hamiltoniens affinement indépendants.

On dispose déjà d'un certain nombre de cycles hamiltoniens affinement indépendants dans l'ensemble \mathcal{T} . On ne garde dans \mathcal{T} qu'un nombre maximal de tels cycles hamiltoniens affinement indépendants. Ainsi $\dim \mathcal{T} = |\mathcal{T}| - 1$. Dans toute cette partie on note $\mathcal{T} = \{\Gamma_1, \dots, \Gamma_{|\mathcal{T}|}\}$ et les vecteurs d'incidence des cycles de \mathcal{T} sont notés $x^1, \dots, x^{|\mathcal{T}|}$. Pour compléter l'ensemble \mathcal{T} , on utilisera la procédure TILT donnée ci-dessous. Cette procédure permet, à partir d'une face de dimension d , d'un cycle hamiltonien Γ_0 n'appartenant pas à cette face et d'une équation non triviale (ni équation de degré ni arête égale à 1) satisfaite par $\mathcal{T} \cup \Gamma_0$, de trouver une inéquation valide définissant une face de dimension $d + 1$ serrée pour les cycles hamiltoniens de \mathcal{T} . On donne à présent l'algorithme utilisé pour la procédure TILT, avant de

détailler les paramètres d'appel.

TILT (a : un vecteur d'entiers,
 a_0 : un entier,
 v : un vecteur d'entiers,
 v_0 : un entier,
 x^0 : le vecteur d'incidence d'un cycle hamiltonien Γ_0)

Préconditions :

- l'inéquation $ax \geq a_0$ est valide sur $STS(k)$
- l'inéquation $ax \geq a_0$ est serrée relativement aux cycles hamiltoniens de \mathcal{T}
- Les cycles hamiltoniens de \mathcal{T} vérifient $vx^{T_i} = v_0$
- le cycle hamiltonien Γ_0 vérifie $vx^0 = v_0$ et $ax^0 > a_0$.

\bar{x} : le vecteur d'incidence d'un cycle hamiltonien qui minimise vx

si $v\bar{x} < v_0$
alors **si** $a\bar{x} > a_0$
 alors $\lambda = v_0 - v\bar{x}, \mu = a\bar{x} - a_0;$
 $v \leftarrow \lambda a + \mu v, v_0 \leftarrow \lambda a_0 + \mu v_0;$
 $d \leftarrow \text{PGCD}(\{v_i, i = 1, \dots, |E'|\});$
 $v \leftarrow \frac{v}{d}, v_0 \leftarrow \frac{v_0}{d};$
 $\text{TILT}(a, a_0, v, v_0, \bar{x});$
 sinon $v \leftarrow a, v_0 \leftarrow a_0, x^0 \leftarrow \bar{x};$

Fin

Les paramètres v, v_0, x^0 de cette procédure sont modifiés et donnent en sortie une inéquation valide sur $STS(k)$, et serrée pour les cycles hamiltoniens de $\mathcal{T} \cup \{\Gamma'\}$ où Γ' est le cycle hamiltonien dont le vecteur d'incidence est x^0 . Cette méthode consiste à effectuer une rotation afin d'augmenter la dimension de la face considérée. La réduction de cette rotation en transformations successives du vecteur v et de l'entier v_0 est appelée **méthode de Dinkelbach**. Un aperçu plus détaillé peut être trouvé dans [9] et [31].

Afin de transformer une contrainte valide en facette, il ne reste plus qu'à trouver les paramètres pour l'utilisation de la procédure TILT. Les paramètres a et b sont donnés par l'inéquation valide qu'on a déjà trouvé. Il reste à trouver un cycle hamiltonien indépendant de \mathcal{T} et une équation non triviale satisfaite à égalité par ce cycle. La recherche d'un cycle hamiltonien indépendant de \mathcal{T} se fait en construisant de manière complètement aléatoire des cycles hamiltoniens jusqu'à ce que l'un d'eux soit indépendant de \mathcal{T} . Cette étape n'est pas difficile et se réalise très rapidement. On note Γ_0 le cycle ainsi obtenu et x^0 son vecteur d'incidence. Pour trouver une équation vérifiée à égalité par ce cycle hamiltonien, on résout le programme linéaire suivant, dont les variables sont un vecteur de réels v , indexé par les arêtes de G' , et

un réel v_0 :

$$\begin{array}{ll}
 \text{minimiser} & \sum_{e \in E'} v_e \\
 \text{sous les contraintes} & v_0 + \sum_{e \in E'} v_e \geq 2 \\
 & \sum_{e \in E'} v_e \geq 0 \\
 & -v_0 + \sum_{e \in E'} v_e x_e^0 = 0 \\
 & -v_0 + \sum_{e \in E'} v_e x_e^1 = 0 \\
 & \vdots \\
 & -v_0 + \sum_{e \in E'} v_e x_e^{|\mathcal{T}|} = 0
 \end{array}$$

Les deux premières contraintes ont pour but de normaliser l'inéquation, afin que la minimisation puisse effectivement s'opérer. Les autres contraintes traduisent le fait que l'équation résultat doit être serrée pour les cycles de \mathcal{T} et pour Γ_0 . Il existe toujours une telle équation, car les équations de degré satisfont tous les cycles hamiltoniens à égalité. L'équation obtenue par la résolution de ce programme linéaire peut être triviale (équation de degré ou combinaison d'équations de degré), cependant il s'avère en pratique que c'est un cas très rare. Cependant dans le cas où l'équation est triviale il est possible de rechercher à nouveau une équation soit en modifiant Γ_0 soit en ajoutant une contrainte plus restrictive sur les coefficients, sachant que la somme des coefficients de l'équation correspondant à l'égalité de degré vaut $\frac{n'-1}{2}$.

Une fois que l'on dispose d'une telle équation, il est possible de faire appel à la procédure TILT qui trouve une inéquation valide et définissant une face de dimension incrémentée d'une unité par rapport à la face dont on disposait jusqu'à présent. On répète ensuite ce processus jusqu'à obtenir une facette de $STSP(k)$.

Pour l'exemple étudié tout au long de cette section, la procédure TILT permet d'ajouter cinq cycles hamiltoniens à \mathcal{T} permettant d'obtenir une inéquation valide satisfaite à égalité par 65 cycles hamiltoniens indépendants. On illustre le fonctionnement de la procédure TILT sur son premier appel : un cycle indépendant de \mathcal{T} est le cycle hamiltonien Γ suivant : [1-2-3-5-4-6-7-8-9-10-11-12-13-1]. L'équation satisfaite par tous les cycles de \mathcal{T} et par Γ est :

$$\begin{aligned}
 & 80x_{1,2} + 80x_{1,3} + 24x_{1,5} + 40x_{1,6} + 8x_{1,7} + 179x_{1,8} + 24x_{1,9} + 24x_{1,10} + 24x_{1,11} + \\
 & 24x_{1,12} + 24x_{1,13} + 88x_{2,3} + 32x_{2,4} + 48x_{2,5} + 139x_{2,6} + 56x_{2,8} + 56x_{2,9} + 56x_{2,10} + 56x_{2,11} + \\
 & 56x_{2,12} + 56x_{2,13} + 147x_{3,4} + 64x_{3,5} + 155x_{3,6} + 56x_{3,8} + 32x_{3,9} + 32x_{3,10} + 56x_{3,11} + \\
 & 56x_{3,12} + 32x_{3,13} + 123x_{4,5} + 214x_{4,6} + 182x_{4,7} + 131x_{4,8} + 91x_{4,9} + 91x_{4,10} + 91x_{4,11} + \\
 & 107x_{4,12} + 91x_{4,13} + 107x_{5,6} + 91x_{5,7} + 147x_{5,8} + 16x_{5,9} + 123x_{5,10} + 147x_{5,11} + 147x_{5,12} + \\
 & 198x_{6,7} + 238x_{6,8} + 198x_{6,9} + 107x_{6,10} + 123x_{6,11} + 91x_{6,13} + 222x_{7,8} + 75x_{7,9} + 182x_{7,10} + \\
 & 91x_{7,11} + 91x_{7,12} + 182x_{7,13} + 131x_{8,9} + 238x_{8,10} + 131x_{8,11} + 24x_{8,13} + 131x_{9,10} + 24x_{9,11} + \\
 & 24x_{10,11} + 24x_{11,12} + 24x_{12,13} = 1347.
 \end{aligned}$$

Cette équation est obtenue après l'ajout de contraintes de degré afin de rendre tous les coefficients positifs. Ces données permettent d'appeler la fonction avec comme paramètres l'inéquation valide, l'équation qui vient d'être trouvée, et Γ . Le cycle hamiltonien $\bar{\Gamma}$ suivant : [1-4-2-7-3-10-11-9-5-13-8-12-6-1] minimise l'équation vx . On a ainsi $vx^{\bar{\Gamma}} = 192$. Comme de plus $ax^{\bar{\Gamma}} = 9$, on en déduit la valeur de λ et μ . On obtient ainsi $\lambda = 1155$ et $\mu = 6$. La procédure TILT est alors appelée

récurivement jusqu'à l'obtention d'une valeur de sortie.

Finalement l'inéquation trouvée par ce moyen est la même que celle déjà exhibée à l'issue du lifting séquentiel. On dispose donc d'une preuve qu'elle définit une facette de $STSP(k)$.

3.3.4.4 Mise sous forme triangulaire serrée

On normalise enfin cette inéquation en la mettant sous forme triangulaire serrée. Pour cela on utilise le lemme suivant :

Lemme 90 (Naddef et Rinaldi [19]) *Soit $cx \geq c_0$ une inéquation définissant une facette de $STSP(n)$. Une inéquation $fx \geq f_0$ équivalente à $cx \geq c_0$ est triangulaire serrée si et seulement si $f = \lambda A_n + \pi c$ et $f_0 = \lambda U_2 + \pi c_0$, où A_n est la matrice d'incidence sommet arête du graphe K_n et U_2 est le vecteur dont toutes les composantes sont égales à 2. En outre $\pi > 0$ et $\lambda \in \mathbb{R}^{V_n}$ satisfont :*

$$\lambda_u = \frac{1}{2}\pi \max\{c_{v,w} - c_{u,v} - c_{u,w} \mid u, v, w \in V, u \neq v, v \neq w\}$$

On dispose donc d'une méthode permettant de transformer une inéquation quelconque définissant une facette de $STSP(k)$ en une inéquation triangulaire serrée équivalente. Cette opération permet d'obtenir une forme canonique de la contrainte qui est unique. En pratique on cherche à calculer les valeurs des coefficients λ_u qui s'obtient en calculant $\max\{v_{v,w} - v_{u,v} - v_{u,w} \mid i, v, w \in V', u \neq v, v \neq w\}$. La valeur de π est obtenue de telle sorte que $\pi = 1$ si λ_u est pair pour tout $u \in V'$ et $\pi = 2$ sinon, afin que les coefficients de la contrainte triangulaire serrée générée soient tous entiers.

A l'issue de cette étape, on dispose d'une inéquation triangulaire serrée définissant une facette de $STSP(k)$.

L'exemple détaillé permet de trouver l'inéquation suivante sur le graphe réduit à 13 sommets :

$$\begin{aligned} & 2x_{1,2} + 2x_{1,3} + 3x_{1,4} + 2x_{1,5} + 2x_{1,6} + 2x_{1,7} + x_{1,8} + 3x_{1,9} + 3x_{1,10} + 4x_{1,11} + 4x_{1,12} + \\ & 3x_{1,13} + 2x_{2,3} + 3x_{2,4} + 2x_{2,5} + 2x_{2,6} + 2x_{2,7} + 3x_{2,8} + x_{2,9} + x_{2,10} + 2x_{2,11} + 2x_{2,12} + \\ & x_{2,13} + x_{3,4} + 2x_{3,5} + 2x_{3,6} + 2x_{3,7} + 3x_{3,8} + 3x_{3,9} + 3x_{3,10} + 2x_{3,11} + 2x_{3,12} + 3x_{3,13} + \\ & x_{4,5} + x_{4,6} + x_{4,7} + 2x_{4,8} + 2x_{4,9} + 2x_{4,10} + 3x_{4,11} + 3x_{4,12} + 2x_{4,13} + x_{5,8} + x_{5,9} + x_{5,10} + \\ & 2x_{5,11} + 2x_{5,12} + x_{5,13} + x_{6,8} + x_{6,9} + x_{6,10} + 2x_{6,11} + 2x_{6,12} + x_{6,13} + x_{7,8} + x_{7,9} + x_{7,10} + \\ & 2x_{7,11} + 2x_{7,12} + x_{7,13} + 2x_{8,9} + 2x_{8,10} + 3x_{8,11} + 3x_{8,12} + 2x_{8,13} + x_{9,11} + 3x_{9,12} + 2x_{9,13} + \\ & x_{10,11} + 3x_{10,12} + 2x_{10,13} + 2x_{11,12} + 3x_{11,13} + x_{12,13} \geq 14. \end{aligned}$$

Cette inéquation est équivalente à l'inéquation précédente mais elle est sous forme triangulaire serrée. Sa violation par la solution fractionnaire associée est de 0.75.

3.3.5 Transformation en contrainte valide pour le problème initial

Il reste à effectuer la transformation de la contrainte précédemment trouvée en une facette pour le graphe support initial. En fait après avoir recherché une

contrainte sur le graphe réduit, il est à nouveau nécessaire de la transformer en une contrainte valide $fx \geq f_0$ définissant une facette de $STSP(n)$, et à ce titre exploitable.

L'idée intuitive qui se présente à nous est de remplacer dans le graphe réduit chaque sommet par l'ensemble qu'il représente, et de modifier les coefficients des arêtes de telle sorte que pour tout $i \in E_i$ et tout $j \in E_j$, $i \neq j$, $f_{i,j} = v_{i,j}$ et pour tout $i_1, i_2 \in E_i$, $f_{i_1, i_2} = 0$. Cette opération peut s'effectuer par l'ajout de sommets dans le graphe réduit grâce à une méthode de duplication. On obtient alors une contrainte valide, triangulaire serrée. Comme la contrainte sur le graphe réduit est facette de $STSP(k)$, le théorème 45 permet d'affirmer qu'elle est facette de $STSP(n)$ si pour tout sommet u , l'ensemble $\delta(u)$ est f -connecté. Cette condition n'est pas forcément vérifiée. Cependant la contrainte sur le graphe réduit est facette de $GTSP(k)$, donc de $GTSP(n)$ d'après le théorème 43. Il s'agit d'un résultat assez fort qui permet de conserver une utilité théorique à ces contraintes. En effet aucune classe de contrainte facette de $GTSP(n)$ et non facette de $STSP(n)$ n'a à ce jour été exhibée. Cela ne signifie absolument pas qu'il n'en existe pas, mais il semblerait que ces contraintes soient des cas rares. Par ailleurs le théorème 45 ne donne qu'une condition suffisante, donc il se peut que cette condition soit trop forte pour la démonstration des propriétés polyédrales d'une contrainte facette de $GTSP(n)$. Il existe alors une marge d'incertitude liée aux outils incomplets dont nous disposons, ce qui est dû à la complexité de la description de l'enveloppe convexe des tours (ou cycles hamiltoniens) du problème du Voyageur de commerce.

3.3.6 Mise en œuvre informatique

On précise ici quelques points de la mise en œuvre informatique de la recherche de telles contraintes. Afin de gagner en vitesse les contraintes retournées par l'algorithme développé n'étaient pas systématiquement facettes de $STSP(n)$, en revanche elles étaient toutes valides, de telle sorte qu'elles étaient exploitables. L'implémentation en mémoire de telles contraintes est relativement lourde, car les ensembles E_1, \dots, E_k doivent pouvoir être restitués. Afin de limiter la représentation en mémoire de ces ensembles, l'ensemble E_i de cardinal le plus grand parmi les ensembles E_1, \dots, E_k n'est pas stocké. Le gain moyen par cette astuce avoisine un facteur de 2, car dans de nombreux cas, un ensemble a un cardinal nettement plus élevé que les autres. Concernant la contrainte associée au graphe réduit, ses coefficients sont tous stockés dans un tableau, mais cela ne représente que $\frac{k(k-1)}{2} + 1$ coefficients (un coefficient par arête et le membre de droite). Une autre possibilité aurait été de mettre ces contraintes sous forme d'hypergraphe. l'avantage aurait été d'avoir un nombre d'ensembles stockés plus réduit, mais des sommets se seraient trouvés plusieurs fois dans les mêmes ensembles et une étape supplémentaire aurait été nécessaire.

Le calcul du cactus associé à un graphe utilise une place en mémoire relativement importante, de l'ordre de $O(n^2)$ où n est le nombre de sommets du graphe. Les nouvelles procédures développées par Wenger [36] peuvent cependant réduire cet espace mémoire à $O(m)$ où m est le nombre d'arêtes du graphe support. Le temps

utilisé par la construction du cactus est aussi fortement diminué par ces nouvelles procédures, mais il ne s'agit pas d'un facteur critique.

Au niveau du temps utilisé par la recherche de contraintes, il est principalement utilisé dans la phase d'ajout séquentiel de sommets, car à chaque ajout correspond la résolution d'un problème de type Voyageur de Commerce à k sommets. Cette résolution peut être longue et avoir de nombreux branchements, du fait que les instances résolues ne sont pas géométriques, donc il est impossible d'y appliquer des heuristiques de recherche de cycles hamiltoniens basées sur les propriétés géométriques du plan ou de l'espace. De plus très souvent beaucoup d'arêtes ont la même valeur, rendant la résolution plus difficile.

On a choisi de résoudre ces problèmes en un temps limité, car une borne inférieure sur la longueur du cycle hamiltonien optimal permet encore d'obtenir une inéquation valide dans la phase d'ajout séquentiel. Ainsi on n'obtient pas systématiquement une contrainte définissant une facette de $STSP(n)$, mais on essaie de s'en rapprocher le plus possible !

3.3.7 Résultats

La recherche de telles contraintes a été effectuée sur les principales instances de la TSPLIB ayant moins de 3000 sommets. Les contraintes trouvées sont majoritairement des facettes n'appartenant pas à des classes connues, car on dispose d'algorithmes de séparation pour ces classes connues qui sont efficaces, et utilisent aussi une partition du graphe à l'aide de cactus.

Le tableau 3.1 regroupe les résultats obtenus avec et sans l'utilisation des contraintes obtenues grâce aux coupes de poids minimum. Les problèmes sont résolus avec un branchement sur variable (une seule variable est utilisée pour le branchement). Comme les contraintes utilisant les coupes minimum sont ajoutées après avoir recherché les contraintes "classiques", les instances pour lesquelles on donne les résultats sont celles nécessitant un branchement dans le cas où seules les contraintes "classiques" sont utilisées. Les valeurs données dans ce tableau sont :

- la valeur du résultat du dernier programme linéaire obtenu avant branchement, notée **valeur racine**
- le nombre de branches de l'arbre d'exécution, noté **taille arbre**
- le nombre de contraintes trouvées à l'aide de la partition du graphe support en coupes minimum, noté **nombre contraintes partitions**

Ces valeurs sont complétées par **sans partition** ou **avec partition** suivant que l'on applique ou pas la recherche de contraintes en utilisant les coupes minimum du graphe support. Pour rappel, le nombre de sommets d'une instance est donné dans son nom (exemple : l'instance `ts225` a 225 sommets).

L'application de la recherche de contraintes à l'aide de coupes se fait durant un temps qui varie dynamiquement en fonction des contraintes déjà trouvées, mais le temps initial utilisé est le même pour toutes les instances. C'est pourquoi sur des instances de petite taille, le temps d'exécution total du problème sera fortement augmenté tandis que sur des instances de taille plus grande, le temps d'exécution

sera proche, voire diminué, par rapport au temps de référence. Par conséquent les temps d'exécution ne sont pas précisés dans cette section, le but étant de valider la pertinence de ces contraintes.

La puissance de la machine dont nous disposons ne permettait pas d'effectuer une résolution complète du problème pour les instances marquées '-', cependant la valeur au nœud racine est indiquée afin d'avoir une idée de la progression induite par ces contraintes dans la résolution.

On constate que la valeur du programme linéaire au nœud racine progresse lorsque les contraintes utilisant les coupes minimum sont utilisées, ce qui est tout à fait logique puisque la séparation est plus poussée. L'importance de cette progression dépend beaucoup des instances, mais elle est réelle pour de nombreuses instances et permet souvent alors une résolution évitant des branchements. Comme les contraintes utilisant les coupes minimum sont recherchées dans chaque branche de l'arbre de résolution du problème, la progression peut se poursuivre lors de la résolution. Ainsi certaines instances pour lesquelles la valeur au nœud racine ne progresse pas beaucoup peuvent être résolues en un nombre plus faible de branches grâce à l'exhibition de contraintes après le nœud racine, comme par exemple l'instance `pa561`.

Quelques instances voient aussi le nombre de branches de l'arbre de résolution augmenter, comme par exemple l'instance `u724`. Ces instances permettent de mettre en évidence l'influence du branchement sur la résolution du problème. Le choix des variables de branchement s'effectue différemment selon le graphe support de la solution au nœud racine, donc une meilleure solution au nœud racine peut amener à une résolution plus longue à cause d'un choix peu judicieux d'une variable de branchement.

Quatre instances ne permettent pas de trouver des contraintes violées en utilisant les coupes minimum du graphe support. Deux raisons principales peuvent expliquer cette absence :

- Tous les graphes réduits ne sont pas étudiés lors de la recherche de contraintes à l'aide des coupes minimum. En effet cette étude serait trop longue et ne permettrait pas d'obtenir un temps de résolution compétitif. Il se peut que les graphes réduits choisis n'aient pas été suffisamment pertinents et soient tous des combinaisons convexes de cycles hamiltoniens. Comme l'intérêt de cette méthode est de chercher des contraintes appartenant à des classes inconnues (recherche "à l'aveugle"), la notion de pertinence d'un graphe réduit n'a pas vraiment de sens.
- Il n'existe pas de partition du graphe en 8 à 30 sommets à l'aide des coupes minimum. Cela se produit pour l'instance `ts225` par exemple. Dans ce cas il n'est pas possible d'effectuer la recherche de contraintes, et il n'y a pas de progression dans la résolution du problème.

Malgré ces instances où le résultat de cette recherche n'est pas probant, la plupart des instances étudiées permettent de montrer l'intérêt d'une telle recherche de contraintes sur le nombre de branchements à effectuer. Comme le branchement est gourmand en mémoire, la taille en mémoire du problème se trouve diminuée,

Instance	valeur racine sans partition	valeur racine avec partition	taille arbre sans partition	taille arbre avec partition	nombre contraintes partition
pr76	<i>108009.4</i>	108009.4	<i>5</i>	5	0
pr124	<i>59002.1</i>	59030	<i>3</i>	1	32
pr136	<i>96647.5</i>	96672	<i>3</i>	1	16
gr137	<i>69836.1</i>	69853	<i>3</i>	1	7
d198	<i>15778.3</i>	15780	<i>3</i>	1	72
ts225	<i>124986.7</i>	124986.7	<i>293</i>	293	0
gr229	<i>134536.3</i>	134584.0	<i>3</i>	3	6
pr299	<i>48185.4</i>	48185.4	<i>3</i>	3	0
rd400	<i>15263.7</i>	15263.7	<i>15</i>	15	0
gr431	<i>171367.7</i>	171369.1	<i>11</i>	7	16
pr439	<i>107126.3</i>	107137.0	<i>9</i>	7	21
pcb442	<i>50759.8</i>	50759.8	<i>11</i>	7	49
att532	<i>27681.2</i>	27681.1	<i>5</i>	3	9
pa561	<i>2760.3</i>	2760.7	<i>9</i>	7	41
rat575	<i>6769.6</i>	6769.6	<i>9</i>	9	11
d657	<i>48896.6</i>	48896.9	<i>7</i>	7	42
u724	<i>41898.5</i>	41900.5	<i>5</i>	13	17
dsj1000	<i>18659407.9</i>	18659654.8	<i>3</i>	3	60
pr1002	<i>259027.2</i>	259045	<i>3</i>	1	44
u1060	<i>224027.8</i>	224027.9	<i>11</i>	11	8
vm1084	<i>239182.9</i>	239184.3	<i>11</i>	9	30
pcb1173	<i>56876.3</i>	56876.3	<i>13</i>	11	63
rl1323	<i>270048.4</i>	270048.4	<i>25</i>	29	22
nrv1379	<i>56630.9</i>	56631.5	<i>7</i>	5	45
fl1400	<i>20116.3</i>	20116.4	<i>7</i>	7	170
d1655	<i>62125.4</i>	62128	<i>5</i>	1	17
vm1748	<i>336455.0</i>	<i>336455.0</i>	11	13	12
pcb3038	<i>137644.2</i>	137645.0	-	-	30
fnl4461	<i>182534.5</i>	182535.8	-	-	52

TAB. 3.1 – Principaux résultats de la séparation à l'aide des coupes minimum

ainsi que le temps de résolution pour les grandes instances. Il est aussi intéressant de remarquer que toutes les partitions du graphe à l'aide des coupes minimum ne permettent pas de trouver des contraintes violées. En effet seule une minorité de partitions mènent à une contrainte violée. En moyenne 3% des graphes réduits permettent de trouver une contrainte violée, ce qui signifie que pour 3 contraintes trouvées, 100 graphes ont été analysés. Une méthode simple permettant d'augmenter le pourcentage de réussite permettrait de réduire considérablement le temps passé à la recherche de telles contraintes violées.

Chapitre 4

Méthodes avancées de branchement

Dans ce chapitre on aborde les méthodes de branchement utilisées au cours de la résolution du problème du Voyageur de Commerce par la méthode “Branch & Cut”. Ces méthodes ont une grande importance pour la résolution des problèmes de grande taille, car le nombre de branches de l’arbre de résolution influe directement sur la durée de résolution. Il est donc nécessaire de rendre ce nombre minimum, mais on ne dispose que de peu d’informations sur le comportement de la résolution lorsqu’on réalise un tel branchement.

4.1 Principe des méthodes de branchement

Le principe de base d’une méthode de branchement est de générer deux sous-problèmes du problème initial, puis de les résoudre séparément et de garder la meilleure solution réalisable obtenue. L’union des solutions entières réalisables de chacun des problèmes doit évidemment contenir toutes les solutions réalisables du problème initial. On appelle *fil*s les sous-problèmes et *père* le problème initial. Chaque fil d’un problème est généré en ajoutant une contrainte dans le programme linéaire relaxé initial. L’union des solutions des programmes linéaires obtenus par l’ajout de ces contraintes doit couvrir l’ensemble des solutions du problème, mais ne contient pas la solution fractionnaire relative au père. On choisit des contraintes disjointes, de telle sorte que les deux fils générés n’aient pas de solutions communes.

On décrit à présent les différents types de branchement qui peuvent être effectués ainsi que leurs avantages et inconvénients.

4.2 Branchement sur une variable

Une première possibilité de branchement est de choisir une variable e dont la valeur x_e^* dans le programme linéaire père est fractionnaire et de générer les fils en ajoutant respectivement les contraintes $x_e = 0$ et $x_e = 1$. Ainsi on progresse

dans la résolution du problème en éliminant au moins une variable fractionnaire. Historiquement cette méthode est la première à avoir été mise en pratique car elle est facile à mettre en œuvre, étant donné que la valeur des arêtes est connue.

En pratique, on constate que cette méthode est bien adaptée aux problèmes de moins de 1000 sommets pour lesquels il n'existe pas de "nuages" de points fortement éloignés les uns des autres. La figure 4.1 illustre ce problème. Cette figure représente une portion du graphe support d'une solution au nœud racine de l'instance *fl1577*. On constate que la valeur de l'arête $e = (1291, 776)$ est de 0.45. Une possibilité est d'effectuer un branchement sur cette variable. On aboutit alors à deux sous-problèmes fils : on ajoute au premier fils la contrainte $x_{1291,776} = 1$ et au deuxième fils la contrainte $x_{1291,776} = 0$. On analyse à présent les deux fils. Le premier fils aboutit à une solution fortement modifiée, car on impose le passage entre les deux nuages de points de la figure. Par conséquent le parcours de chacun des nuages est fortement conditionné par ce passage. On aboutit donc rapidement à une solution. En revanche pour le deuxième fils le problème est plus aigu. En effet le fait d'imposer que l'arête $e = (1291, 776)$ a un coefficient égal à 0 ne permet pas de progresser. Les arêtes $(1291, 958)$, $(1291, 957)$, $(1394, 776)$ et $(1353, 776)$, etc... ont une longueur très proche de l'arête e , donc le fait d'imposer $x_e = 0$ ne modifiera presque pas la solution et créera le même problème jusqu'à ce que de nombreuses arêtes soient fixées. Or fixer de nombreuses arêtes équivaut à obtenir un arbre de résolution profond, donc une résolution qui peut être lente...

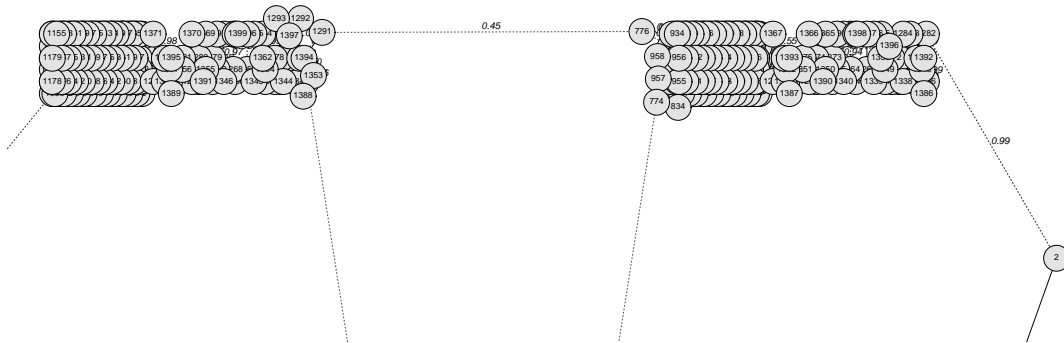


FIG. 4.1 – Instance *fl1577* : extrait de la solution avant branchement

Un autre problème est abordé par l'exemple précédent : le déséquilibre de l'arbre de résolution. En effet la contrainte $x_e = 1$ est nettement plus forte que la contrainte $x_e = 0$, le nombre d'arêtes ayant une valeur de 1 étant égal au nombre de sommets du graphe. Ce problème est commun à toutes les instances du Voyageur de Commerce, cependant l'arbre de résolution est acceptable pour des problèmes où la répartition des villes est relativement homogène. En revanche sur des instances du type *fl1577*, l'arbre de résolution obtenu par l'exécution de notre algorithme a une taille im-

portante. Dans le cas de l'instance considérée, il comporte 275 nœuds lorsqu'une résolution avec branchement sur les variables est effectuée.

Le choix des variables sur lesquelles effectuer les différents branchement se fait en fonction de leur valeur dans la solution relaxée. Les variables ayant une valeur proches de 0.5 sont choisies, et on préfère les variables correspondant à des arêtes de grande longueur dans l'instance du problème, car la mise à 1 d'une telle variable modifie de façon importante la valeur de la fonction objectif. Plusieurs variables candidates peuvent être sélectionnées pour effectuer un branchement, la section 4.5 précise quelle variable parmi cet ensemble de candidats est choisie.

4.3 Branchement sur un ensemble de variables

Dans le cas représenté par la figure 4.1, on se rend compte qu'un branchement exploitant les nuages de sommets pourrait être plus efficace que le branchement sur une variable. Un branchement sur un ensemble de variables est dans ce cas plus efficace. Ce branchement consiste à choisir un ensemble d'arêtes F , puis à générer les fils du problème en ajoutant respectivement les contraintes $x(F) = 0$ et $x(F) \geq 1$. La principale difficulté consiste à choisir l'ensemble F . Le choix s'est porté sur un ensemble de type $(V_1 : V_2)$, c'est-à-dire toutes les arêtes ayant une extrémité dans un ensemble de sommets V_1 et l'autre extrémité dans un ensemble de sommets V_2 . Si on note V_1 et V_2 les nuages de sommets de la figure, on peut générer deux fils en ajoutant respectivement les contraintes $x(V_1 : V_2) = 0$ et $x(V_1 : V_2) \geq 1$. La contrainte $x(V_1 : V_2) = 0$ permet d'imposer qu'aucune arête de $(V_1 : V_2)$ ne se trouve dans la solution obtenue dans cette branche, donc celle-ci sera radicalement modifiée au niveaux des ensembles V_1 et V_2 . La deuxième contrainte quant à elle impose un passage entre V_1 et V_2 . Cette contrainte est moins forte qu'un branchement sur une arête seule, car il se peut que l'on obtienne une solution fractionnaire.

En pratique on se rend compte que sur les instances où la répartition des villes n'est pas homogène, l'arbre de résolution du problème est plus équilibré que si un branchement sur une variable est effectué, car la contrainte $x(F) = 0$ est plus forte qu'une contrainte de type $x_e = 0$ tandis que la contrainte $x(F) \geq 1$ est moins forte qu'une contrainte de type $x_e = 1$. Ce ré-équilibrage de l'arbre de résolution se fait au profit de son nombre de nœuds, car on constate une diminution qui peut aller jusqu'à 30%. Cependant cette méthode doit être combinée avec des méthodes de branchement sur variable, car en fin de branchement (près des feuilles de l'arbre), les ensembles n'ont plus une structure qui convient à une telle résolution.

Le choix des deux ensembles de sommets V_1 et V_2 est primordial, car il conditionne la progression de la valeur de la solution du programme linéaire relaxé. Ces ensembles ne doivent pas avoir un cardinal trop grand ni trop petit, sinon on crée à nouveau un déséquilibre de l'arbre de résolution, synonyme de perte de temps de résolution. Il s'agit là du principal point faible de la méthode développée, car on ne dispose pas de critères suffisamment pertinents pour choisir un ensemble plutôt qu'un autre. Les ensembles candidats sont donc des ensembles pour lesquels

$0.2 \leq x^*(V_1 : V_2) \leq 0.7$. La sélection parmi plusieurs ensembles candidats est expliquée dans la section 4.5.

4.4 Branchement sur des contraintes

Le branchement sur contrainte est aussi une méthode de branchement permettant de progresser dans la résolution du problème. Une classe de contrainte souvent utilisée est la classe des contraintes de sous-tours. En effet si on exhibe un ensemble U de sommets dans le graphe support d'une solution x^* du programme linéaire relaxé ayant un cocycle proche d'une valeur impaire $2k + 1$, on peut générer deux fils en ajoutant respectivement les contraintes $x(\delta(U)) \leq 2k$ et $x(\delta(U)) \geq 2k + 2$ au programme linéaire relaxé. En effet tout cycle hamiltonien a une intersection avec tout cocycle en un nombre pair d'arêtes. Dans le cas de notre recherche, les ensembles ayant un cocycle proches de 3 sont les ensembles pour lesquels k est le plus faible, car les contraintes de sous-tours sont toutes satisfaites. D'autres contraintes peuvent aussi être utilisées dans le cadre des méthodes de branchement, telles les contraintes de peigne.

L'utilisation de telles contraintes a aussi comme avantage, tout comme pour les ensembles d'arêtes, de limiter le déséquilibre de l'arbre de résolution du problème du Voyageur de Commerce. En effet, contrairement aux branchements sur une variable, le branchement sur une contrainte de sous-tour dont la valeur est proche d'un petit nombre impair (typiquement 3) permet d'obtenir un arbre de résolution relativement équilibré. Là aussi la recherche d'un tel ensemble ne peut se faire que par heuristiques, mais l'essentiel est d'en trouver suffisamment puisque le choix d'un tel ensemble s'effectue ensuite (voir section 4.5)

4.5 Choix des paramètres de branchement

Lorsqu'on a décidé d'une méthode de branchement pour un problème donné, il est possible d'exhiber plusieurs contraintes pour une solution x^* donnée. Le logiciel ABACUS permet d'effectuer un test permettant de choisir quelle contrainte on souhaite effectivement ajouter au programme linéaire relaxé. Ce test consiste à calculer la valeur de la fonction objectif du programme linéaire en ajoutant la règle de branchement et en résolvant le programme un certain nombre d'itérations en tenant compte de la règle de branchement. Ainsi on peut avoir une idée de la performance de la règle de branchement et choisir celle qui semble la plus prometteuse.

Pour chaque contrainte de branchement, le test mené par ABACUS donne une borne inférieure de la fonction objectif pour chaque fils. On choisit de tester N couples de contraintes différents, avec $10 \leq N \leq 50$. Pour un couple i de contraintes, on obtient deux bornes inférieures a_{i_1} et a_{i_2} . Le choix du couple de contraintes intégrées au problème afin de générer deux fils peut se faire de différentes manières que l'on détaille à présent afin de percevoir leurs avantages et inconvénients pratiques.

Historiquement le couple de valeurs choisi (a_{k_1}, a_{k_2}) était celui pour lequel $\max\{a_{k_1}, a_{k_2}\} = \max\{a_{i_1}, a_{i_2}; i = 1, \dots, N\}$ où N est le nombre de couples candidats. Cette méthode permet d'effectuer un branchement sur la contrainte qui fait progresser la valeur de la fonction objectif du programme linéaire relaxé de la façon la plus importante. Cependant toutes les contraintes sont en couples, de telle sorte qu'il arrive que la deuxième contrainte du couple ne fasse que peu progresser la valeur de la fonction objectif. Si tel est le cas, l'utilité d'un tel branchement est négligeable, puisqu'une des branches est un sous-problème très proche du problème père. Peu à peu cette méthode de choix parmi les candidats au branchement a été abandonnée au profit d'autres méthodes jugées plus efficaces.

La deuxième possibilité exploitée est une méthode "prudente" obtenue en appliquant une stratégie maximin. Elle consiste à choisir le couple de contraintes maximisant $\min(a_{i_1}, a_{i_2})$. Ainsi on garantit une progression minimale dans la résolution du problème. C'est cette méthode que nous utilisons dans notre logiciel de résolution du problème du Voyageur de Commerce. Elle a l'avantage de limiter le déséquilibre de l'arbre de résolution du problème tout en réalisant une progression honorable.

Les deux méthodes qui viennent d'être brièvement exposées utilisent comme unique critère d'amélioration de la solution la valeur donnée par la borne inférieure. Or cette borne, même si elle donne une idée du potentiel de progression, ne permet pas de savoir si une contrainte fera plus progresser la valeur de la solution du programme linéaire relaxé qu'une autre. Il s'agit donc d'un critère forcément incomplet, mais actuellement on ne dispose pas de meilleur critère.

D'autre part, les méthodes exposées ne proposent qu'une amélioration gloutonne. Un branchement qui paraît mauvais au moment du choix parmi les candidats pourrait en effet très bien donner de bons résultats par la suite et mener à une résolution rapide. Cependant on ne connaît pas de critère permettant d'évaluer la pertinence d'un branchement par rapport à un autre en fonction du comportement global de la résolution. Le seul moyen serait d'effectuer une résolution complète pour chaque candidat, donc de résoudre complètement le problème plusieurs fois.

4.6 Résultats

Les résultats avec un branchement sur variable sont fournis dans les tableaux du chapitre précédent. En effet les résolutions des instances de référence pour tester les contraintes obtenues par la séparation à l'aide des coupes minimum du graphe support ont été obtenues en effectuant un branchement par variable. On rappelle ces résultats dans le tableau 4.1 afin que la comparaison des méthodes soit plus facile.

Pour comparer d'autres méthodes de branchement à celle par variable, les instances dont l'arbre de résolution comporte au moins cinq nœuds avec le branchement par variable sont les plus intéressantes, car ce sont celles qui permettent d'espérer une amélioration. En effet dès qu'un branchement est nécessaire, l'arbre de résolution comporte au moins trois nœuds, donc pour les instances résolues en trois nœuds avec le branchement par variable, aucune amélioration n'est possible. On peut seulement

Instance	Taille arbre branchement sur variables	Taille arbre branchement sur ensembles
pr76	5	5
ts225	293	215
gr229	3	3
pr299	3	3
rd400	15	11
gr431	7	7
pr439	7	7
pcb442	7	11
att532	3	3
pa561	7	11
rat575	9	9
d657	7	7
u724	13	11
dsj1000	3	3
u1060	11	7
vm1084	9	9
pcb1173	11	11
rl1323	29	39
nrv1379	5	5
fl1400	3	15

TAB. 4.1 – Principaux résultats de la séparation à l'aide des coupes minimum

vérifier qu'il n'y a pas de détérioration de la qualité de la résolution.

Les résultats montrent que le branchement par variables peut être amélioré dans certains cas par le branchement sur des ensembles. Comme on ne dispose pas de critère suffisamment pertinent pour sélectionner les ensembles candidats, le branchement sur des ensembles se révèle dans de nombreux cas moins efficace que le branchement par variables, pour lequel on dispose d'un critère de choix qui a

fait ses preuves. Cependant le branchement sur des ensembles semble prometteur, la sélection des ensembles devant encore être affinée.

Chapitre 5

Bilan et perspectives

5.1 Bilan

5.1.1 De nouveaux outils pour la phase de séparation

Le travail effectué au cours de ces trois années de thèse a permis d'aboutir au développement d'outils nouveaux permettant d'améliorer le processus de séparation lorsque les outils précédemment utilisés montraient leurs limites. On a vu que la séparation à l'aide des coupes minimum du graphe support permet d'améliorer la borne inférieure obtenue lors de la résolution des programmes linéaires relaxés, de telle sorte que souvent le nombre de branchements nécessaires pour obtenir une solution optimale du problème diminue. Les contraintes ajoutées par ce moyen sont issues d'une recherche "à l'aveugle". Elles ne dépendent que du graphe support et ne permettent pas une meilleure connaissance de $STSP(n)$ dans le cas général. Ainsi ces contraintes ne sont valables que pour un problème donné, et on ne peut pas, dans l'état actuel de nos connaissances, les réutiliser facilement dans d'autres situations. Mais cet inconvénient est aussi un avantage, car il permet d'ajouter au programme linéaire relaxé des contraintes définissant des facettes de $STSP(n)$ appartenant à des classes inconnues. Comme le nombre de classes de contraintes définissant des facettes de $STSP(n)$ augmente exponentiellement avec n , il n'est pas possible de toutes les décrire, donc une telle méthode permet de pallier de façon très partielle cet inconvénient. On a vu qu'une telle méthode permet dans la plupart des cas de réduire le nombre de branches de l'arbre de résolution du problème, donc son avantage est certain. Cependant la génération de contraintes est plus lente que pour la génération par heuristiques, à cause de la résolution de nombreux problèmes de Voyageur de Commerce de petite taille (moins de 30 villes). Comme la résolution du problème est appelée de façon récursive, toute amélioration de la séparation ou du branchement profite à cette méthode!

Le travail théorique effectué sur les contraintes de dominos a permis quant à lui d'apporter un résultat sur les propriétés polyédrales des faces définies par ces inéquations. Ces contraintes permettent ainsi de dégager une nouvelle classe de contraintes définissant des facettes de $STSP(n)$ afin de mieux connaître ce polyèdre.

Dans le cadre de la séparation de ces contraintes, on a vu qu'il existe déjà des algorithmes de séparation qui ont montré leur efficacité. Le fait d'exhiber des conditions nécessaires pour qu'une contrainte de dominos définisse une facette de $STSP(n)$ permet de développer des algorithmes de séparation ne recherchant que de telles contraintes, celles ne définissant pas de facettes étant contenues dans d'autres facettes.

5.1.2 Une nouvelle méthode de branchement

Depuis l'utilisation de la méthode "Branch & Cut" pour la résolution du problème du Voyageur de Commerce, l'essentiel des efforts a été porté sur la séparation. Une séparation poussée permet de retarder la phase de branchement, qui est en général plus coûteuse en temps. Cependant dans le cas de la plupart des problèmes à plusieurs centaines de villes, le branchement s'avère nécessaire, malgré tous les efforts déployés pour effectuer une séparation efficace. Il devenait donc important de concentrer les efforts d'amélioration sur cette phase du processus de résolution. Le résultat est pour l'instant mitigé mais prometteur. Il s'avère qu'un branchement mixte en cherchant des candidats parmi les variables et parmi les ensembles permet d'obtenir de meilleurs résultats qu'un branchement avec une seule des méthodes prise séparément. Cependant un problème majeur subsiste : le choix parmi les différents candidats au branchement ne peut pas se faire avec un critère précis. En effet ce choix ne peut pas s'effectuer sur un critère de minimisation du nombre de branches de l'arbre de résolution du problème, mais sur une borne inférieure de la valeur de la fonction objectif du programme linéaire relaxé. Cette borne est peu représentative du comportement de la résolution, de telle sorte qu'il est difficile d'avoir la certitude de choisir un bon candidat. Il n'est malheureusement pas possible d'obtenir une meilleure borne sans résoudre le problème, et le test de toutes les résolutions du problème est inutile, donc inenvisageable. Par conséquent l'absence d'un critère précis de choix des candidats au branchement force à ne proposer que des "bons" candidats, c'est-à-dire des candidats qui permettront une progression de la résolution, même si ce n'est pas la meilleure progression possible.

La méthode proposée permet en pratique d'obtenir un arbre de résolution plus équilibré sur un certain nombre de problèmes, mais n'est pour l'instant véritablement efficace que sur les problèmes où les branchements par variable ont réellement montré leur limite, c'est-à-dire des problèmes dans lesquels se trouvent des regroupements de sommets éloignés les uns des autres, ou des problèmes très réguliers, comme l'emplacement de composants sur des plaques électroniques. Mais elle ne constitue pas une panacée, dans la mesure où il ne s'agit pas d'une méthode exacte et que son ajustement pratique nécessite de nombreux tests. Elle possède cependant un avantage théorique certain sur le branchement par variable.

5.2 Perspectives

5.2.1 Extension des résultats polyédraux pour les contraintes de dominos

Au cours du travail effectué sur les contraintes de dominos, on a exhibé des conditions suffisantes pour qu'elles définissent des facettes de $STSP(n)$. Quelques conditions nécessaires ont aussi été exhibées, mais il reste une marge d'incertitude. Le prolongement naturel de ces résultats serait d'exhiber des conditions nécessaires et suffisantes pour que les contraintes de dominos définissent des facettes de $STSP(n)$. La figure 5.1 représente une configuration de dominos ne possédant qu'une dent maximale impaire, et deux dents maximales paires. A l'heure actuelle on ne sait pas si cette configuration définit une facette de $STSP(28)$. Si on arrive à montrer qu'elle ne définit pas une facette de $STSP(28)$, il est alors nécessaire que toute configuration de dominos comporte au moins trois dents impaires maximales pour qu'elle induise une facette de $STSP(n)$. On disposerait ainsi de conditions nécessaires et suffisantes pour que les contraintes de dominos définissent des facettes de $STSP(n)$.

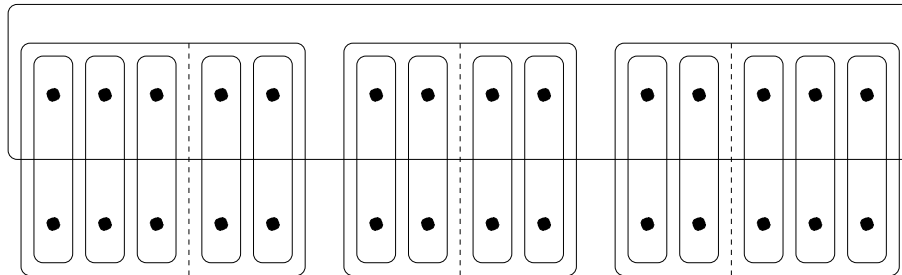


FIG. 5.1 – Configuration de dominos aux propriétés polyédrales non connues

Pour montrer que la configuration précédemment citée n'induit pas de facette de $STSP(28)$ deux méthodes sont possibles :

- montrer que l'inéquation définie par cette configuration est dominée par une inéquation définissant une facette connue de $STSP(28)$: nous n'avons pas trouvé de telle configuration jusqu'à présent
- montrer que le nombre maximum de cycles hamiltoniens affinement indépendants serrés relativement à cette inéquation est strictement inférieur à 350, nombre nécessaire pour qu'une face soit facette de $STSP(28)$. Cependant comment exhiber ces cycles hamiltoniens ? La génération exhaustive des cycles hamiltoniens serrés relativement à cette inéquation pourrait être faisable, mais ne donnerait pas la facette dans laquelle cette face est contenue.

Il s'agit là du seul résultat à obtenir pour avoir une condition nécessaire et suffisante sur les propriétés polyédrales des inéquations de dominos.

5.2.2 Améliorations du branchement

Comme on l'a vu précédemment, le branchement dans sa forme actuelle peut encore être amélioré. Pour cela un travail d'ajustement des paramètres de recherche d'ensembles est nécessaire dans un premier temps. Ensuite un calcul plus précis de la borne inférieure de la valeur de la fonction objectif pour chaque candidat devrait être proposé. Un tel calcul ne doit cependant pas être beaucoup plus coûteux en temps. Pour l'instant le logiciel ABACUS se charge de ce calcul. D'autres méthodes de branchement pourraient encore être proposées, mais une exploitation judicieuse des méthodes actuelles semble déjà assez prometteuse. Cependant tant que la valeur de la borne inférieure du problème est peu fiable, il est très difficile d'obtenir de façon quasi-certaine un branchement efficace.

Pour obtenir une borne plus fiable, la seule méthode actuellement exploitable est de pousser plus loin les tests des candidats. Cependant un test effectuant une partie de la séparation peut s'avérer coûteux en temps, donc le gain risque d'être faible s'il est mal géré. Il s'agit là d'un problème important, dont la résolution semble difficile mais permettrait d'améliorer considérablement le temps d'exécution pour les instances à grand nombre de villes.

5.2.3 Parallélisation de la résolution

A l'heure actuelle un simple ordinateur de bureau est encore trop peu puissant pour résoudre de façon séquentielle le problème du Voyageur de Commerce pour des instances à plusieurs milliers de villes. Une parallélisation de la résolution semble donc appropriée. Il existe déjà une version parallèle d'ABACUS et de CONCORDE, qui effectuent une résolution parallèle du problème à un bas niveau, en tenant compte de la plate-forme utilisée. Par ailleurs ces versions utilisent une parallélisation du branchement et non du reste du problème. Un des buts du projet ACI DOC-G (Défis en Optimisation Combinatoire sur Grappes et Grilles de Grappes), dont le laboratoire Informatique et Distribution est partenaire, est de résoudre à terme ce problème sur des grappes de calcul. Pour cela le développement se réalise indépendamment de la plate-forme utilisée, grâce à une interface de programmation d'applications parallèles : Athapascan [30]. Afin de pouvoir résoudre efficacement le problème du Voyageur de Commerce sur une architecture multi-processeur, il est nécessaire de le décomposer afin d'obtenir un "grain" de résolution plus fin. Une idée innovante dans ce cadre serait de paralléliser la recherche de contraintes lors de la phase de séparation, car les contraintes trouvées sont valides tout au long de la résolution. Cela permettrait ainsi d'avoir une séparation plus rapide et de tirer parti du parallélisme sans attendre la phase de branchement. Plusieurs problèmes se posent néanmoins, dont celui des transmissions de variables globales aux différents processeurs, car le programme linéaire, et surtout sa solution fractionnaire, sont des données qui occupent beaucoup de place mémoire et dont on a besoin pour réaliser la séparation. Par ailleurs le développement de façon indépendante de l'architecture utilisée à l'aide d'Athapascan impose certaines contraintes, dont celle de ne pas avoir de processeur

dédié à des tâches particulières. Ceci est particulièrement difficile pour la résolution des programmes linéaires. Ainsi de nombreux problèmes se posent encore pour effectuer une telle résolution, mais le challenge est très intéressant car il permettrait, s'il aboutit, une mise en oeuvre aisée et l'espoir de résolution de nombreux problèmes à grand nombre de villes en un temps record et pour un coût financier assez faible, puisque ne nécessitant que des ordinateurs de bureau.

Bibliographie

- [1] David Applegate, Robert Bixby, Vasek Chvátal, and William Cook. TSP Cuts Which Do Not Conform to the Template Paradigm. In Michael Jünger Denis Naddef, editor, *Computational Combinatorial Optimization*, volume 2241 of *LNCS*, pages 261–304. Springer, 2001.
- [2] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8 :146–164, 1975.
- [3] S.C. Boyd, W.H. Cunningham, M. Queyranne, and Y. Wang. Ladders for the traveling salesmen. *SIAM Journal on Optimization*, 5 :408–420, 1993.
- [4] Sylvia Boyd and Sally Cockburn. A family of facet-inducing domino-parity inequalities for the STSP. Technical report, University of Ottawa, August 2001.
- [5] Sylvia Boyd, Sally Cockburn, and Danielle Vella. On the domino-parity inequalities for the STSP. Technical report, University of Ottawa, October 2001.
- [6] Patrick Chabrier, Christine Gaspin, Simon de Givry, and Thomas Schiex. Application des techniques du voyageur de commerce à la production de cartes génétiques. In *5e congrès de la société Française de Recherche Opérationnelle et d’Aide à la Décision, ROADEF 2003*, pages 74–76, February 2003.
- [7] Vasek Chvátal. Edmonds polytopes and weakly hamiltonian graphs. *Mathematical Programming*, 5 :29–40, 1973.
- [8] Gérard Cornuéjols, Jean Fonlupt, and Denis Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33 :1–27, 1985.
- [9] B. D. Craven. *Fractional Programming*. Heldermann, Berlin, 1988.
- [10] E.A. Dinic, A.V. Karzanov, and M.L. Lomonosov. *On the structure of the family of minimal weighted cuts in a graph*, pages 290–306. Publications Nauka, Moscou, 1976.
- [11] Lisa Fleischer. Building chain and cactus representations of all minimum cuts from hao-orlin in the same asymptotic run time. *Journal of Algorithms*, 33 :51–72, October 1999.
- [12] B. Fleischmann. A new class of cutting planes of the symmetric traveling salesman problem. *Mathematical Programming*, 40 :225–246, 1988.
- [13] R. E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and Its Applications*, 2 :451–558, 1969.

- [14] Martin Grötschel and Manfred W. Padberg. On the symmetric traveling salesman problem i : inequalities. *Mathematical Programming*, 16 :265–280, 1979.
- [15] Martin Grötschel and Manfred W. Padberg. On the symmetric traveling salesman problem ii : lifting theorems and facets. *Mathematical Programming*, 16 :281–302, 1979.
- [16] Martin Grötschel and William Pulleyblank. Clique tree inequalities an the symmetric traveling salesman problem. *Mathematics of Operations Research*, 11 :537–569, 1986.
- [17] P. L. Hammer, E. L. Johnson, and U. N. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8 :179–206, 1975.
- [18] Adam N. Letchford. Separating a superclass of combs inequalities in planar graphs. *Mathematics of Operations Research*, 25(3) :443–454, 2000.
- [19] Denis Naddef and Giovanni Rinaldi. The graphical relaxation : A new framework for the symmetric traveling salesman polytope. *Mathematical Programming*, pages 53–88, 1992.
- [20] Denis Naddef and Giovanni Rinaldi. The symmetric traveling salesman polytope : new facets from the graphical relaxation. Technical report, Laboratoire Informatique et Distribution - IMAG, 2000.
- [21] Denis Naddef and Stefan Thienel. Efficient separation routines for the Symmetric Traveling Salesman Problem I : general tools and comb separation. *Mathematical Programming*, 92(2) :237–255, April 2002.
- [22] Denis Naddef and Stefan Thienel. Efficient separation routines for the Symmetric Traveling Salesman Problem II : separating multi handle inequalities. *Mathematical Programming*, 92(2) :257–283, April 2002.
- [23] Denis Naddef and Emmanuel Wild. The domino inequalities : facets for the symmetric traveling salesman polytope. *Mathematical Programming*, Series B 98 :223–251, 2003.
- [24] D. Naor and V. V. Vazinari. Representing and enumerating edge connectivity cuts in RNC. In *Lecture Notes in Computer Science*, volume 519, pages 273–285. Springer Verlag, 1991.
- [25] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, chapter 1.4. Wiley, 1988.
- [26] M. W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5 :199–215, 1973.
- [27] M. W. Padberg. A note on zero-one programming. *Operations Research*, 23 :833–837, 1975.
- [28] Manfred W. Padberg and Giovanni Rinaldi. An efficient algorithm for the minimum capacity cut problem. *Mathematical Programming*, 47 :19–36, 1990.
- [29] Gerhard Reinelt. TSPLIB - a traveling salesman library. *ORSA Journal on Computing*, 3 :376–384, 1991.

- [30] J.-L. Roch, F. Galilée, M. Doreille, G. Cavalheiro, N. Maillard, R. Revire, and A. Defrenne. *Athapascan : API for Asynchronour Parallel Programming*, 2002. http://www-id.imag.fr/Logiciels/ath1/manual/A1_Documentation/athapascan1/.
- [31] I. M. Stancu-Minasian. *Fractional Programming*. Kluwer, Dordrecht, 1997.
- [32] Stefan Thienel. *ABACUS - A Branch-And-CUt System*. PhD thesis, Universität zu Köln, 1995.
- [33] Stefan Thienel. *A Simple TSP-Solver : An ABACUS Tutorial*, 1996.
- [34] Stefan Thienel. Introduction to abacus - a branch-and-cut system. Technical Report 97.263, Universität zu Köln, 1997.
- [35] Andrea De Vitis. The cactus representation of all minimum cuts in a weighted graph. Technical Report 454, IASI, Viale Manzoni 30, 00185 Roma, Italy, May 1997.
- [36] Klaus M. Wenger. A New Approach to Cactus Construction Applied to TSP Support Graphs. In William J. Cook and Andreas S. Schulz, editors, *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, Proceedings*, volume 2337 of *LNCS*, pages 109–126. Springer, May 2002.
- [37] L. A. Wolsey. Faces for a linear inequality in 0-1 variables. *Mathematical Programming*, 8 :165–178, 1975.
- [38] L. A. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24 :367–372, 1975.

Résumé

Ce travail de thèse comporte deux composantes, l'une théorique sur l'enveloppe convexe des cycles hamiltoniens, aussi appelée polytope du Voyageur de Commerce, et une autre plus numérique sur l'amélioration de la résolution exacte par la méthode "Branch & Cut" du problème du Voyageur de Commerce.

L'apport théorique consiste en la démonstration qu'une classe d'inéquations, les contraintes de domino, induisent des facettes du polytope du Voyageur de Commerce. L'aspect numérique aborde la séparation hors paradigme de classe en proposant la génération de coupes à partir de la contraction d'un grand graphe en un plus petit à l'aide de la représentation en cactus des coupes minimum. Enfin diverses pistes ont été étudiées pour rendre l'étape de branchement plus robuste.

Mots-clé : voyageur de commerce, optimisation polyédrale, branchement, séparation, programmation linéaire.

Abstract

This dissertation is divided into two part, a theoretical one deals with the traveling salesman polytope, the convex hull of hamiltonian cycles, and a more numerical one in which we try to improve the "Branch and Cut" method for traveling salesman problem.

The theoretical contribution consists in proving that the class of domino inequalities is facet inducing for the traveling salesman polytope. The numerical one deals with the separation outside the template paradigm by studying cut generation from small graphs obtained from a bigger one by shrinking minimum cuts. Finally branching is studied in order to make more robust this part of "Branch and Cut"

Keywords : traveling salesman problem, polyhedral optimization, branch, cut, linear programming.