N° d'Ordre : D.U. 1555
EDSPIC : 314

UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II

ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

THÈSE

présentée par

Pierre FOUILHOUX

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : INFORMATIQUE

Graphes k-partis et conception de circuits VLSI

Soutenue publiquement le 16 décembre 2004 devant le jury

MM	Alain QUILLIOT	Président
	Philippe CHRÉTIENNE	Rapporteur
	Jean FONLUPT	Rapporteur
	Philippe MAHEY	Éxaminateur
	Jean MAILFERT	Éxaminateur
	Ali Ridha MAHJOUB	Directeur de Thèse

Remerciements

Je tiens tout d'abord à remercier Monsieur Ali Ridha Mahjoub, Professeur à l'université Blaise Pascal à Clermont-Ferrand, d'avoir accepté d'être mon directeur de thèse et de m'avoir encadré durant ces années. Je veux lui exprimer toute ma gratitude pour m'avoir fait confiance pour toutes les tâches de recherche, d'enseignement ou d'animation que j'ai réalisées sous sa direction. Par dessus tout, je lui suis profondément reconnaissant d'avoir quotidiennement œuvré pour me transmettre ses connaissances, son expérience et sa passion pour la recherche et l'enseignement.

J'adresse mes plus sincères remerciements à Monsieur Philippe Chrétienne, Professeur à l'Université Pierre et Marie Curie à Paris, pour l'intérêt qu'il a bien voulu porter à ce travail, et pour m'avoir fait l'honneur d'accepter la charge de rapporteur.

Je suis très reconnaissant envers Monsieur Jean Fonlupt, Professeur à l'Université Pierre et Marie Curie à Paris, pour s'être intéressé à mon travail et avoir accepté de rapporter ma thèse. Je le remercie pour l'attention qu'il a porté à mon mémoire et ses remarques constructives sur mes travaux.

Je voudrais également remercier Monsieur Alain Quilliot, Professeur à l'Université Blaise Pascal à Clermont-Ferrand, d'avoir accepté de présider le jury de cette thèse. Je remercie aussi Monsieur Philippe Mahey, Professeur à l'Université Blaise Pascal à Clermont-Ferrand, d'avoir examiné mes travaux et participé au jury.

Enfin, je remercie Monsieur Jean Mailfert, Maître de Conférence à l'Université d'Auvergne à Clermont-Ferrand, d'avoir participé à l'examen de mon mémoire et au jury. Je voudrais d'une manière plus générale remercier Monsieur Jean Mailfert et Madame Fatiha Bendali-Mailfert, Maître de Conférence à l'Université Blaise Pascal à Clermont-Ferrand. Durant ces années de formation universitaire et plus particulièrement de thèse, ils ont, chacun à leur manière, fait montre d'attentions et de gentillesse dans tous les conseils et les encouragements qu'ils m'ont donnés.

Je tiens à témoigner ici de toute mon amitié pour Pierre Pesneau et Sylvie Borne qui ont partagé l'un après l'autre mon bureau et mes soucis pendant ces années de thèse. Si le travail et la passion pour la recherche et l'enseignement nous a naturellement réunis, je souhaite que notre amitié continue à être vive et tenace malgré les distances. Je veux particulièrement remercier Sylvie Borne pour m'avoir porté et supporté quotidiennement du début à la fin de cette étrange année 2004.

Je tiens plus globalement à remercier les membres titulaires et temporaires de l'équipe EPOC (Equipe Polyèdre et Optimisation Combinatoire) du laboratoire LIMOS de Clermont-Ferrand, au sein de laquelle j'ai passé des années très agréables. Je remercie ainsi à nouveau Ridha Mahjoub, Jean Mailfert, Fatiha Bendali-Mailfert, Pierre Pesneau et Sylvie Borne. Je veux aussi remercier Lise Slama, David Huygens, Ibrahima Diarrassouba et Bernard Fortz qui ont su être quotidiennement de bons compagnons pour parler de travail et de tant d'autres choses. Je tiens plus particulièrement à remercier Hervé Kérivin, Maître de Conférences à l'Université Blaise Pascal, qui a toujours pris le temps de prévenir mes soucis et me donner de bons conseils.

Je tiens encore plus globalement à remercier les membres du laboratoire LIMOS, du département mathématiques et informatique de l'Université Blaise Pascal ainsi que de l'IUT d'Auvergne, avec qui j'ai collaboré pour l'enseignement mais aussi que j'ai croisés dans les couloirs et les cafétérias. La liste serait bien longue et périlleuse mais je tiens à remercier très particulièrement Fabien de Marchi, Maître de Conférence à l'Université Claude Bernard Lyon 1, pour ses leçons de chose, sa confiance et son amitié.

Tout ceci serait bien incomplet si je ne remerciais pas ma famille, et en particulier mes parents, de toujours m'aimer, me redonner confiance en moi et surtout me supporter au quotidien, hier pendant la thèse, comme demain...

Résumé

Le problème de Via Minimization contraint concerne la dernière étape du processus de conception de circuits VLSI. Pour positionner les réseaux de pistes du circuit sur une carte de façon à ce que deux segments de réseaux différents ne soient pas connectés, il est parfois nécessaire de faire des percements dans la carte pour connecter des segments sur des couches différentes. De tels percements sont appelés des vias. Le problème de Via Minimization contraint consiste à déterminer une affectation des réseaux aux couches qui utilise un nombre minimum de vias. Dans cette thèse, nous étudions ce problème.

Nous montrons dans un premier temps que le problème sur k couches se ramène au problème du sous-graphe k-parti induit maximum.

Par la suite, nous considérons le problème de Via Minimization dans le cas de 2 couches. Dans ce cas, le problème est équivalent au problème du sous-graphe biparti induit. Nous étudions le polyèdre associé à ce problème. Nous décrivons plusieurs classes de contraintes valides et nous donnons des conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes du polytope. Nous discutons également d'algorithmes de séparation et nous montrons que certaines de ces classes peuvent être séparées en temps polynomial. Nous développons aussi un algorithme de coupes et branchements basé sur ces résultats pour le problème du sous-graphe biparti induit. Nous utilisons cet algorithme pour résoudre des instances du problème de Via Minimization contraint.

Nous étudions également le problème général du sous-graphe k-parti induit pour k quelconque. Nous proposons une formulation en nombres entiers pour laquelle nous développons un algorithme de génération de colonnes et branchements. Celui-ci est utilisé pour résoudre des instances du problème de Via Minimization contraint ayant plus de deux couches.

Enfin, nous nous intéressons au problème d'assemblage SNP d'haplotypes qui constitue une étape du séquençage du génome pour les organismes diploïdes. Pour le critère dit d'enlèvement minimum de fragments, il a été prouvé que ce problème se ramène au problème du sous-graphe biparti induit de cardinalité maximum. Considérant cette modélisation, nous utilisons notre algorithme de coupes et branchements pour résoudre aussi des instances de ce problème. Nous proposons également une modélisation du problème d'assemblage SNP d'haplotypes pour le critère dit du nombre minimum de corrections. Nous montrons que le problème avec ce critère se ramène aussi au problème du sous-graphe biparti induit.

Mots clefs: sous-graphe k-parti induit, Via Minimization contraint, polytope, facette, problème de séparation, génération de colonnes, problème d'assemblage SNP d'haplotypes.

Abstract

The constrained via minimization problem arises as the last step of VLSI circuit design process. In a layer assignment of the wire nets on a VLSI card so that two segments of distinct nets have no connection, it can be necessary to drill a hole through the card in order to connect segments assigned on different layers. Such a hole is called a via. The constrained via minimization problem consists in determining a layer assignment with a minimum number of vias. In this thesis, we study this problem.

We first show that the problem on k layers can be reduced to the maximum k-partite induced subgraph problem.

After that, we consider the via minimization problem in the two layer case. The problem, in this case, is equivalent to the bipartite induced subgraph problem. We study the polytope associated to this problem. We describe several classes of valid constraints and give necessary and sufficient conditions for these constraints to be facet defining. We also discuss separation algorithms and show that some of these classes can be separated in polynomial time. We develop a branch-and-cut algorithm based on these results for the bipartite induced subgraph problem. We use this algorithm to solve some instances of the constrained via minimization problem.

We also consider the general k-partite induced subgraph problem for any k. We discuss an integer programming formulation for which we develop a branch-and-price algorithm. This is used to solve some intances of the constrained via minimization problem with more than two layers.

Finally, we study the SNP haplotype assembly problem which is a step of the DNA sequencing process for diploid organisms. For the minimum fragment removal criterion, it has been shown that this problem can be reduced to the maximum cardinality bipartite induced subgraph problem. We use our branch-and-cut algorithm to solve some instances of that problem. We also give a model for the SNP haplotype assembly problem for the minimum correction number criterion. We prove that the problem for that criterion can also be reduced to the bipartite induced subgraph problem.

Key word: k-partite induced subgraph, constrained via minimization, polytope, facet, separation problem, column generation, SNP haplotype assembly problem.

Table des matières

Introduction

L'optimisation combinatoire est une branche de l'informatique et de la programmation mathématique. Elle combine des techniques de la combinatoire, de la programmation linéaire et de la théorie des algorithmes afin de résoudre des problèmes d'optimisation ayant des structures discrètes (généralement un graphe). Un problème d'optimisation combinatoire consiste à déterminer un élément (le meilleur) dans un ensemble fini. Les approches polyédrales constituent un des outils puissants pour résoudre ce type de problèmes. Initiées en 1965 par Jack Edmonds pour le problème du couplage, elles sont maintenant de plus en plus utilisées pour résoudre les problèmes d'optimisation combinatoire difficiles (comme par exemple le problème du voyageur de commerce, le problème de la coupe maximum,...).

Les problèmes traités par l'optimisation combinatoire et la recherche opérationnelle couvrent des thèmes aussi variés que les problèmes industriels (systèmes de production, planification,...), les problèmes de la cité (transports publics, administration de la santé, de l'enseignement, écologie,...) ou les problèmes liés aux nouvelles technologies (télécommunications, circuits VLSI, génomique,...). L'étude d'un problème consiste généralement à le ramener à un modèle mathématique ou informatique, puis à utiliser des méthodes (comme les méthodes polyédrales) afin de proposer des solutions.

De nombreux travaux d'optimisation combinatoire sont issus du problème de conception de circuits VLSI. Ce dernier consiste à optimiser la conception des cartes de circuits intégrés de grandes dimensions (Very Large Scale Integrated). Le processus de conception se divise habituellement en trois phases, le placement des composants, le routage des réseaux de pistes et l'affectation des réseaux aux couches.

Dans cette dernière étape du processus, les réseaux de pistes doivent être positionnés sur la carte de façon à ce que deux segments de réseaux différents ne soient pas connectés. Pour cela, on peut positionner deux segments d'un même réseau sur des couches différentes puis les relier d'une couche à l'autre en perçant l'épaisseur de la carte. De tels percements sont appelés des vias. Le problème de Via Minimization contraint consiste à déterminer une affectation des réseaux aux couches qui utilise un nombre minimum de vias.

Dans cette thèse, nous proposons un modèle exact pour le problème de Via Minimization contraint. Nous montrons que le problème de Via Minimization contraint sur k couches se ramène à la recherche d'un sous-graphe k-parti induit de poids maximum dans un graphe approprié.

Dans un premier temps, nous étudions le problème dans le cas 2 couches, ici le problème est équivalent au problème du sous-graphe biparti induit de poids maximum. Nous considérons le polyèdre associé à ce problème. Nous décrivons plusieurs classes de contraintes valides et nous donnons des conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes du polytope. Nous discutons également d'algorithmes de séparation et nous montrons que certaines de ces classes peuvent être séparées en temps polynomial. Nous développons aussi un algorithme de coupes et branchements basé sur ces résultats pour le problème du sous-graphe biparti induit. Nous utilisons également cet algorithme pour résoudre des instances du problème de Via Minimization contraint.

Par la suite, nous considérons le problème du sous-graphe k-parti induit pour un k quelconque. Nous proposons une formulation en nombres entiers pour laquelle nous développons un algorithme de génération de colonnes et branchements. Nous utilisons également cet algorithme pour résoudre des instances du problème de Via Minimization contraint ayant plus de deux couches.

Enfin, nous nous intéressons au problème d'assemblage SNP d'haplotypes qui constitue une étape du séquençage du génome pour les organismes diploïdes. Pour le critère dit d'enlèvement minimum de fragments, il a été prouvé que ce problème se ramène au problème du sous-graphe biparti induit de cardinalité maximum. Considérant cette modélisation, nous avons utilisé notre algorithme de coupes et branchements pour le problème du sous-graphe biparti induit afin de résoudre des instances de ce problème. Nous proposons également une modélisation du problème d'assemblage SNP d'haplotypes pour un autre critère, appelé critère du nombre minimum de corrections. Nous montrons que ce problème se ramène aussi au problème du sous-graphe biparti induit.

Dans le premier chapitre, nous introduisons quelques notions de base de l'approche polyédrale et nous donnons certaines définitions et notations. Dans le chapitre 2, nous présentons un état de l'art pour le problème de conception de circuits VLSI. Dans le chapitre 3, nous proposons notre modèle pour le problème de Via Minimization contraint. Les chapitres 4 et 5 présentent l'étude du polytope des sous-graphes bipartis induits. Dans le chapitre 6, nous développons un algorithme de coupes et branchements pour le problème du sous-graphe biparti induit et nous présentons des résultats expérimentaux. Le chapitre 7 discute du problème du sous-graphe k-parti induit pour k quelconque. Enfin, le chapitre 8 s'intéresse au problème d'assemblage SNP d'haplotypes.

Chapitre 1

Notions préliminaires et définitions

Dans ce chapitre, nous donnons quelques notions de base sur les polyèdres combinatoires et la méthode de coupes et branchements. Nous donnons également des définitions et des notations qui seront utilisées tout au long de ce mémoire.

1.1 Notions préliminaires

1.1.1 Eléments de la théorie des polyèdres

Dans cette section, nous allons introduire quelques définitions et propriétés de la théorie des polyèdres. Pour plus de détails, on peut se référer à Pulleyblank [92] et Schrijver [96].

Soit $x \in \mathbb{R}^n$. On dit que x est une *combinaison linéaire* des points $x_1, ..., x_k \in \mathbb{R}^n$ s'il existe k scalaires $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$ tels que $x = \sum_{i=1}^{k} \lambda_i x_i$. Si, de plus, $\sum_{i=1}^{k} \lambda_i = 1$, alors on dit que x est une *combinaison affine* de ces points. De même, si $\lambda_i \geq 0$ pour tout $i \in \{1, ..., k\}$ avec $\sum_{i=1}^{k} \lambda_i = 1$, x est alors dit *combinaison convexe* de ces points.

Des points $x_1, ..., x_k \in \mathbb{R}^n$ sont dits *linéairement indépendants* (resp. *affinement indépendants*) si le système

$$\sum_{i=1}^{k} \lambda_i x_i = 0 \qquad \left(\text{resp. } \sum_{i=1}^{k} \lambda_i x_i = 0 \text{ et } \sum_{i=1}^{k} \lambda_i = 0\right)$$

admet une solution unique, $\lambda_i = 0$ pour $i = 1, ..., k$.

Soit S un ensemble non vide de points de $I\!R^n$. L'*enveloppe convexe* des points de S, notée *conv(S)*, est l'ensemble de tous les points de $I\!R^n$ qui peuvent s'écrire comme combinaison convexe de points de S.

Un *polyèdre* P est un ensemble de points de $I\!R^n$ engendré par l'intersection d'un nombre fini de demi-espaces de $\mathbb{R}^n$. D'une manière équivalente, P est l'ensemble des solutions d'un système d'inégalités linéaires, c'est-à-dire $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, où A est une matrice à m lignes et n colonnes, et b un vecteur à m composantes. Nous dirons alors que le système $Ax \leq b$ définit (ou caractérise) le polyèdre P. Un *polytope* est un polyèdre borné. Par exemple, l'enveloppe convexe d'un ensemble fini de points est un polytope.

Un polyèdre P de $I\!R^n$ est de *dimension* d si le nombre maximum de points de P affinement indépendants est égal à $d+1$. On écrit alors $dim(P) = d$. Un polyèdre P est dit de *pleine dimension* si $dim(P) = n$. Un point $x \in I\!R^n$ est un *point extrême* d'un polyèdre P s'il ne peut pas être écrit comme combinaison convexe d'autres points de P.

Une inégalité (ou *contrainte*) $ax \leq \alpha$ est dite *valide* pour un polyèdre P si elle est vérifiée par tous les points de P. Soit $ax \leq \alpha$ une inégalité valide pour P. Le sous-ensemble $F = \{x \in P \mid ax = \alpha\}$ est appelé *face* de P. On dit aussi que F est la face définie par $ax \leq \alpha$. Si $F \neq \emptyset$ et $F \neq P$, on dit que la face F est *propre*. Si F est propre et $dim(F) = dim(P) - 1$, alors F est appelée *facette* de P.

Une inégalité est dite *essentielle* pour P si elle définit une facette du polyèdre P. En revanche, une inégalité est dite *redondante* dans un système $Ax \leq b$ définissant un polyèdre P, si le sous-système, obtenu à partir de $Ax \leq b$ en supprimant cette inégalité, définit le même polyèdre P. Soit $x^* \in \mathbb{R}^n$. Nous dirons que l'inégalité $ax \leq a_0$ est *serrée* (resp. *violée*) par x^* si $ax^* = a_0$ (resp. $ax^* < \alpha$).

1.1.2 Approche polyédrale, méthode de coupes et branchements

Soient $\mathcal{P}$ un problème d'optimisation combinatoire et $\mathcal{S}$ l'ensemble de ses solutions. Le problème $\mathcal{P}$ s'écrit alors

$$\max \{cx \mid x \in \mathcal{S}\},$$

où c est une fonction coût associée aux variables du problème. Considérons l'enveloppe convexe $conv(\mathcal{S})$ des solutions de $\mathcal{P}$. Le problème $\mathcal{P}$ est alors équivalent au programme linéaire

$$\max \{cx \mid x \in conv(\mathcal{S})\}.$$

Par conséquent, si nous caractérisons le polyèdre $conv(\mathcal{S})$ par un système d'inégalités linéaires, alors nous ramenons le problème $\mathcal{P}$ à la résolution d'un programme linéaire.

L'*approche polyédrale*, introduite par Edmonds [35] dans le cadre du problème du couplage, consiste à étudier le polytope $conv(\mathcal{S})$ afin de pouvoir résoudre $\mathcal{P}$ comme un programme linéaire. La caractérisation complète du polytope $conv(\mathcal{S})$ est généralement difficile à obtenir. Par ailleurs, une description complète du polyèdre peut comporter un nombre exponentiel d'inégalités. Cependant, un nombre réduit de ces inégalités peut être suffisant pour résoudre le problème à l'aide d'une *méthode de coupes*. Cette méthode permet de résoudre un problème $\mathcal{P}$ comme une séquence de programmes linéaires, chacun contenant un nombre raisonnable de contraintes. Pour pouvoir utiliser une telle méthode, nous devons déterminer des classes d'inégalités valides (et non redondantes) pour $conv(\mathcal{S})$, pour lesquelles le problème suivant, appelé *problème de séparation*, peut être résolu efficacement.

Problème Soit $\mathcal{C}$ une classe d'inégalités valides pour $conv(\mathcal{S})$. Etant donné un point $x \in I\!R^n$, le *problème de séparation* associé à $\mathcal{C}$ et x consiste à déterminer si x satisfait toutes les inégalités de $\mathcal{C}$ et sinon de trouver une inégalité de $\mathcal{C}$ violée par x.

Le problème de séparation est la clef de voûte d'une méthode de coupes. En effet, Grötschel, Lovász et Schrijver [74] ont montré qu'un problème d'optimisation combinatoire sur un ensemble de contraintes $\mathcal{C}$ peut être résolu en temps polynomial si et seulement si le problème de séparation associé à $\mathcal{C}$ peut être résolu en temps polynomial. Ainsi, une méthode de coupes permet de résoudre un problème d'optimisation combinatoire en temps polynomial, si on sait résoudre en temps polynomial le problème de séparation pour un système d'inégalités caractérisant le polyèdre associé.

Cependant, comme il a déjà été noté auparavant, il y a peu d'espoir de connaître la caractérisation complète du polyèdre des solutions du problème. D'autre part, le problème de séparation sur certaines classes d'inégalités valides peut être lui-même NP-difficile. Dans ce cas, on ne peut disposer que de techniques de séparation approchées. Ainsi, une méthode de coupes seule peut ne fournir que des solutions fractionnaires (non optimales). Dans ce cas, on exécute une étape de branchement qui peut consister à choisir une variable fractionnaire x_i dans la solution et à considérer deux sous-problèmes du problème courant en fixant x_i à 0 pour l'un et à 1 pour l'autre. On applique alors la méthode de coupes pour les deux sous-problèmes. La solution optimale du problème sera donc la meilleure entre les deux solutions entières obtenues pour les deux sous-problèmes. Cette phase de branchement est répétée de manière récursive jusqu'à l'obtention d'une solution entière optimale. Cette combinaison de la méthode de branchements et d'une méthode de coupes au niveau de chaque noeud de l'arbre de branchement est appelée *méthode de coupes et branchements* (*Branch and Cut method*). Cette méthode s'est révélée très efficace pour la résolution de problèmes d'optimisation combinatoire réputés pour être difficiles tels que le problème du voyageur de commerce [7] ou celui de la coupe maximum [12, 13].

1.1.3 Notations et définitions

Les graphes que nous considérons sont finis, non orientés et sans boucle ni arête multiple. Nous notons un graphe par $G = (V, E)$ où V est l'*ensemble des sommets* et E l'*ensemble des arêtes* de G. Si $e \in E$ est une arête d'extrémités u et v, l'arête e peut être également notée uv.

Soit $G = (V, E)$ un graphe. Un *sous-graphe* $H = (W, F)$ de G est un graphe tel que $W \subseteq V$ et $F \subseteq E$. On dit qu'un sous-graphe $H = (W, F)$ de G *couvre* les sommets de G si $W = V$. On appelle *sous-graphe partiel* de G un sous-graphe $G' = (V, E')$ avec $E' \subset E$. Si $W \subset V$, alors $E(W)$ désigne l'ensemble de toutes les arêtes ayant leurs deux extrémités dans W. Le graphe $H = (W, E(W))$ est le sous-graphe de G *induit* par W.

Si $F \subset E$, alors $V(F)$ désigne l'ensemble des sommets de V qui sont extrémités des arêtes de F. Si $W \subset V$, alors $\delta(W)$ est l'ensemble des arêtes avec une extrémité dans W et l'autre extrémité dans $V \setminus W$. L'ensemble $\delta(W)$ est appelé une *coupe*. On note $\delta(v)$ à la place de $\delta(\{v\})$ pour $v \in V$ et on appelle $\delta(v)$ l'*étoile* de v. Pour $v \in V$, on désigne par $N(v)$ l'ensemble des sommets adjacents à v. Si $W \subset V$, on pose $N(W) = \left(\bigcup_{v \in W} N(v)\right) \setminus W$, et on nomme $N(W)$ l'ensemble des voisins de W. Soit F un ensemble d'arêtes. On dénote par $G \setminus F$ le graphe obtenu à partir de G en supprimant les arêtes de F.

Une *chaîne* P dans $G = (V, E)$ est une séquence d'arêtes $e_1, e_2, ..., e_k$ telles que $e_1 = v_0v_1, e_2 = v_1v_2, ..., e_k = v_{k-1}v_k$. P est dite *chaîne élémentaire* si elle passe au plus une fois par le même sommet. Les sommets v_0 et v_k sont les extrémités de P et on dit que P relie v_0 à v_k. Le nombre k d'arêtes de P est appelé la *taille* de P. Si $P = e_1, e_2, ..., e_k$ est une chaîne (resp. chaîne élémentaire) reliant v_0 à v_k et $e_{k+1} = v_0v_k \in E$, alors la séquence $e_1, e_2, ..., e_k, e_{k+1}$ est dite un *cycle* (resp. *cycle élémentaire*) de taille $k + 1$. Lorsqu'il n'y a pas ambiguïté, on utilisera chaîne pour désigner une chaîne élémentaire. Un cycle (chaîne) est dit *impair* si la taille est impaire, autrement, il est dit *pair*. Si P est un cycle ou une chaîne et uv une arête de $E \setminus P$ avec $u, v \in V(P)$, alors uv est appelé une *corde* de P. Un *trou* de G est un cycle sans corde.

Un graphe $G = (V, E)$ est *connexe* si, pour toute paire de sommets u, v de V, il existe au moins une chaîne entre u et v. Une *composante connexe* d'un graphe est un sous-graphe connexe qui est maximal par rapport à cette propriété. Un graphe ne possédant pas de cycle est appelé une *forêt*. Un *arbre* est une forêt connexe.

Un graphe $G = (V, E)$ est dit *complet* si toute paire de sommets de V est jointe par une arête de E. On note par K_q le graphe complet sur q sommets. On appelle *clique* de G un sous-graphe complet de G. Un *stable* d'un graphe est un ensemble de sommets deux à deux non adjacents.

La *contraction* d'une arête $e = uv$ consiste à supprimer e, à identifier u et v en préservant les adjacences. Un graphe G est dit *contractible* à un graphe G' si G' peut être obtenu à partir de G par suppressions et/ou contractions d'arêtes. La subdivision d'une arête $e = uv$ consiste à supprimer e et à insérer un nouveau sommet de degré 2 lié à u et à v.

Un graphe est dit *planaire* si on peut le dessiner sur un plan de manière à ce qu'aucune arête n'en croise une autre. On appelle *représentation planaire* d'un graphe planaire une telle représentation. Soit $G = (V, E)$ un graphe planaire. On identifie alors une représentation planaire du graphe G avec l'ensemble de points P du plan composé des points associés aux sommets de V et aux segments associés aux arêtes de E. On nomme ses régions connexes les *faces* de G. Comme P est un ensemble borné (c'est-à-dire que P est contenu dans un disque D suffisamment large), exactement une face de G n'est pas bornée. On appelle cette face *face extérieure* et les autres faces de G *faces intérieures*. La frontière d'une face de G peut être vue comme un sous-graphe de G. On nomme ce sous-graphe *bordure* de la face ou encore *face* de G. Deux faces de G sont dites *adjacentes* si leurs bordures ont une arête commune. Par la suite, quand il n'y a pas d'ambiguïté, on utilisera le mot face pour désigner la bordure d'une face.

1.2 Graphes k-partis

1.2.1 Le problème du sous-graphe k-parti induit

Un graphe est dit *k-parti*, $k \geq 1$, si son ensemble de sommets peut être partitionné en k sous-ensembles de sommets disjoints V_i, $i = 1, ..., k$, tels que les sommets de V_i, pour $i = 1, ..., k$, sont deux à deux non adjacents. Une telle partition est appelée *k-partition*. Si $k = 1$, un graphe k-parti est un stable. En fait, un graphe est k-parti s'il existe k stables disjoints couvrant les sommets du graphe.

Soit $S \subseteq I\!N$. Une *coloration* des sommets d'un graphe $G = (V, E)$ est une fonction $r : V \rightarrow S$ telle que $r(u) \neq r(v)$ pour tout couple de sommets adjacents u, v. Les éléments de l'ensemble S sont appelés les *couleurs* disponibles. Une *k-coloration* est une coloration $c : V \rightarrow \{1, .., k\}$. Un graphe est dit *$k$-coloriable* s'il possède une k-coloration. Clairement, un graphe est k-parti si et seulement s'il est k-coloriable.

Tester si un graphe est k-parti est un problème NP-complet si $k \geq 3$, et il est polynomial si $k = 2$ [40].

Etant donnée une fonction $c : V \rightarrow I\!R$ qui associe à tout sommet $v \in V$ un poids $c(v)$, le *problème du sous-graphe k-parti induit* (k-PSPI) consiste à déterminer un sous-graphe k-parti induit $(W, E(W))$ de G tel que $c(W) = \sum_{v \in W} c(v)$ soit maximum. Le k-PSPI est NP-complet même dans les cas $k = 1$ (problème du stable) ou $k = 2$ (problème du sous-graphe biparti induit).

A notre connaissance, le k-PSPI n'a jamais été traité dans la littérature quand $k \geq 3$. En revanche, le problème de coloration a donné lieu à de nombreux travaux. On appelle *problème de coloration* le problème consistant à déterminer le nombre chromatique d'un graphe. Le *nombre chromatique* $\chi(G)$ d'un graphe G est la plus petite valeur k tel que G soit k-coloriable.

Dans le chapitre 7, nous proposons un algorithme de génération de colonnes et branchements pour le problème du sous-graphe k-parti induit quand $k \geq 3$.

1.2.2 Le problème du sous-graphe biparti induit

Un graphe est appelé *biparti* si son ensemble de sommets peut être partitionné en deux ensembles de sommets non vides V_1 and V_2 tels qu'aucun couple de sommets de V_1 ou de V_2 ne soit lié par une arête. Un graphe biparti est donc un graphe 2-parti. Une 2-partition est également appelée *bipartition*. Il est bien connu qu'un graphe est biparti si et seulement s'il ne contient pas de cycle impair. Tester si un graphe est biparti peut se réaliser en temps polynomial par un parcours de graphe particulier.

Etant donnée une fonction $c : V \rightarrow I\!R$ qui associe à tout sommet $v \in V$ un poids $c(v)$, le *problème du sous-graphe biparti induit* (PSBI) consiste à déterminer un sous-graphe biparti induit $(W, E(W))$ de G tel que $c(W) = \sum_{v \in W} c(v)$ soit maximum.

Le problème du stable consiste à déterminer un stable de poids maximum. Le PSBI est une généralisation du problème du stable. En fait, si $H = (W, F)$ est un graphe, alors le problème du stable dans H peut se ramener au PSBI dans le graphe $G = (V, E)$

obtenu à partir de H en ajoutant pour chaque arête uv de H, un sommet w de poids M, où M est une grande valeur positive, ainsi que les arête wu et wv. Ceci implique que le PSBI est NP-difficile. Le PSBI a été montré NP-difficile même dans les graphes de degré maximum trois et dans les graphes planaires quand le degré maximum est supérieur ou égal à 4 [28]. En revanche, le PSBI peut être résolu en temps polynomial dans les graphes série-parallèles [15] et dans les graphes planaires quand le degré maximum des sommets est limité à 3 [28]. (Un graphe série-parallèle s'il n'est pas contractible à K_4).

Si $W \subset V$, on pose $x^W \in I\!R^V$ tel que $x^W(u) = 1$ si $u \in W$ et $x^W(u) = 0$ sinon, x^W est appelé le *vecteur d'incidence* de W. L'enveloppe convexe $P_B(G)$ des vecteurs d'incidence de tous les ensembles de sommets induisant un sous-graphe biparti de G, i.e.

$$P(G) = conv\{x^W \in I\!R^V \mid W \subset V, (W, E(W)) \text{ est biparti}\},$$

est appelé le *polytope des sous-graphes bipartis induits de G.*

Dans [14] Barahona et Mahjoub étudient le polytope $P_B(G)$. Ils exhibent plusieurs classes de facettes définissant des inégalités et décrivent plusieurs méthodes de construction de facettes. Dans [15], ils étudient une technique de composition pour le polytope $P_B(G)$ dans les graphes qui sont décomposables par des 2-sommets d'articulation (deux sommets de G dont l'enlèvement augmente le nombre de composantes connexes dans G). Si G se décompose en G_1 et G_2, ils montrent que $P_B(G)$ peut être obtenu à partir de deux systèmes linéaires associés à G_1 et G_2. En appliquant ce résultat, ils donnent un algorithme polynomial pour résoudre le PSBI dans les graphes serie-parallèles.

Un *système d'indépendants* $(N, \mathcal{I})$ consiste en un ensemble fini $N = \{u_1, ..., u_n\}$ et une collection non vide $\mathcal{I}$ de sous-ensembles de N qui est fermée au sens de l'inclusion, i.e. $I \in \mathcal{I}$ et $J \subset I$ impliquent $J \in \mathcal{I}$. Les éléments de $\mathcal{I}$ sont dits *indépendants* et ceux de $2^N \setminus \mathcal{I}$ *dépendants*. Un *circuit* de $(N, \mathcal{I})$ est un sous-ensemble dépendant minimal de N. Pour un graphe $G = (V, E)$ donné, la paire $(V, \beta(G))$ est un système d'indépendants où $\beta(G)$ est la famille des ensembles de sommets de V induisant un sous-graphe biparti de G. Ainsi, les ensembles dépendants du système $(V, \beta(G))$ sont les cycles impairs de G et les circuits sont les trous impairs de G.

Dans [89], Nemhauser et Trotter introduisent le *polytope d'un système d'indépendants* $(N, \mathcal{I})$ comme étant l'enveloppe convexe des vecteurs d'incidence des sous-ensembles indépendants de $\mathcal{I}$, i.e. $P_{\mathcal{I}} = conv\{x^I \in \{0,1\}^n \mid I \in \mathcal{I}\}$. Ainsi le polytope $P_B(G)$ peut être vu comme le polytope d'un système d'indépendants.

Dans [93], Euler, Jünger et Reinelt présentent des inégalités valides pour $P_{\mathcal{I}}$ et discutent de conditions nécessaires et suffisantes pour que ces inégalités définissent des facettes de $P_{\mathcal{I}}$. A leur suite, Laurent [67] propose une généralisation de ces classes d'inégalités et donne des conditions nécessaires et suffisantes pour que cette nouvelle classe définisse des facettes de $P_{\mathcal{I}}$.

Soit $G = (V, E)$ un graphe. Etant donné un vecteur-poids c sur les sommets de G. Le *problème du sous-graphe biparti partiel* consiste à déterminer un ensemble d'arêtes $E' \subseteq E$ de poids maximum tel que le sous-graphe partiel $G' = (V', E')$ soit biparti. On appelle *polytope des sous-graphes bipartis partiels* l'enveloppe convexe des vecteurs d'incidence des ensembles de sommets E' tel que (V, E') soit biparti.

Dans [36] Barahona, Grötschel et Mahjoub décrivent plusieurs classes d'inégalités définissant des facettes du polytope des sous-graphes bipartis partiels. Un graphe est dit *faiblement biparti* si son polytope des sous-graphes bipartis partiels coïncide avec le polyèdre défini par les inégalités triviales et les inégalités dites de cycles impairs. En proposant un algorithme de séparation polynomial pour ces contraintes, Grötschel et Pulleyblank [48] montrent que le problème du sous-graphe biparti partiel peut être résolu en temps polynomial dans cette classe de graphes. Barahona [8] montre que les graphes planaires appartiennent à cette classe de graphes. Dans [75], Mahjoub étend ce résultat en montrant que les graphes non contractibles à K_5 sont faiblement bipartis. Guenin [49] a récemment donné une caractérisation pour cette classe de graphes. Dans [38], Fonlupt, Mahjoub et Uhry étudient des compositions dans le polytope des bipartis.

Chapitre 2

Le problème de conception de circuits VLSI

L'objectif de ce chapitre est de donner une vue d'ensemble et les définitions nécessaires à la compréhension du processus de conception de circuits électroniques de grandes dimensions, appelés aussi circuits VLSI.

La conception de circuits de grandes dimensions (VLSI design) est un processus complexe divisé traditionnellement en une succession de problèmes théoriques et pratiques qui mettent en jeu les concepts de fiabilité, de performance du circuit et d'optimisation de la production.

Les sections 2.1 et 2.2 donnent un aperçu de ces différentes étapes en insistant plus particulièrement sur l'étape de routage. Le but de ces deux sections est d'introduire le problème de Via Minimization contraint pour lequel nous donnons un état de l'art exhaustif dans la section 2.3. Dans le chapitre 3, nous étudions le problème de Via Minization contraint et nous montrons qu'il peut se ramener au problème du sous-graphe k-parti induit.

2.1 Vue d'ensemble et définitions

Dans cette section, nous donnons tout d'abord quelques termes techniques et définitions d'électronique. Ensuite nous indiquons les principales caractéristiques techniques et les étapes traditionnelles de la conception de circuits électroniques.

2.1.1 Définitions

On appelle *circuit logique* un dispositif électronique permettant la réalisation d'une fonction logique connue (stockage mémoire, calcul numérique, pilotage de robot,...). Un circuit logique est entièrement décrit par un schéma électronique donnant les liens logiques entre les différents éléments du circuit. Dans tout le document, nous considérons les schémas de circuits logiques (ou, plus simplement, circuits logiques) comme une donnée d'entrée. On ignore de même sa fonction et son utilisation. Cette simplification permet ainsi de donner une définition simple d'un circuit.

Un *circuit* est défini par un ensemble de *composants* et un ensemble de *réseaux*. Les composants seront considérés comme des boîtes noires pouvant schématiser les composants de base (transistors, condensateurs,...) ou des macro-composants (processeurs, sous-circuits,...). A chaque composant correspond un ensemble de *points terminaux*. Un point terminal est une zone ponctuelle du composant qui connecte le composant au reste du circuit.

Un *réseau* est un ensemble de points terminaux devant être électriquement connectés par des pistes conductrices. On appelle *liste des réseaux* (netlist, en anglais) l'ensemble de tous les réseaux d'un circuit. Par extension, on appelle également *réseau* la collection des pistes qui connectent électriquement les points terminaux d'un même réseau. Dans ce cas, un réseau est donc un ensemble de lignes droites ou courbes reliant plusieurs composants.

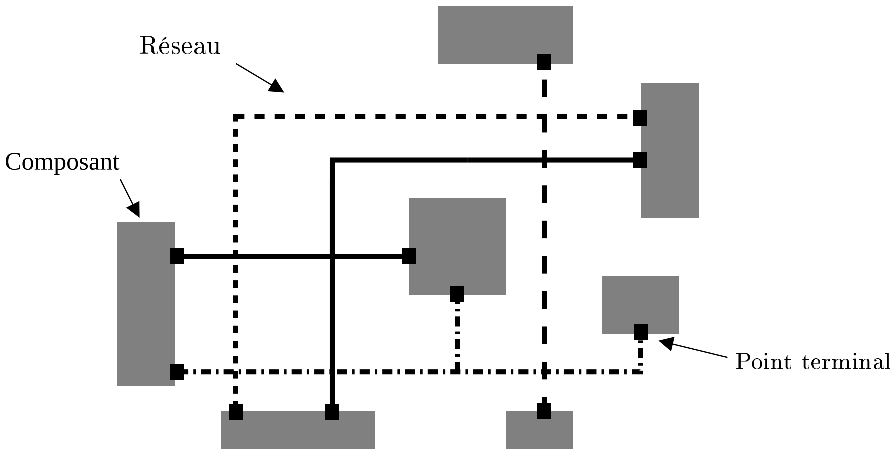


FIG. 2.1 – *Un exemple de circuit*

La figure 2.1 illustre les définitions précédentes. Les composants sont reliés par quatre réseaux différenciés par l'utilisation de types de tracés différents. Certains réseaux ne sont que des lignes droites mais certains se ramifient en plusieurs segments de droite.

Les définitions données jusqu'ici permettent de décrire globalement des circuits issus des différentes technologies existantes. Il existe principalement deux types de technologies: les cartes de circuits imprimés (Printed Circuit Board ou PCB) et les circuits intégrés.

Les cartes de circuits imprimés servent en général à rassembler les gros composants électroniques de manière à former par exemple les cartes mères des micro-ordinateurs, les systèmes de pilotage de robots ou les appareils de mesure. Cette technologie consiste à souder les composants sur la face supérieure et/ou inférieure d'une carte en epoxy dans laquelle circulent des pistes de cuivre correspondant aux réseaux. Parmi ces pistes de cuivre, certaines permettent de véhiculer l'alimentation des composants.

Les circuits intégrés nécessitent une technologie plus récente qui a permis la miniaturisation des processeurs, ainsi appelés *puce* (chip, en anglais). La plupart des circuits intégrés sont créés sur un support en silicium dans lequel sont gravés ou superposés des pistes et des composants. Cette techonologie permet de concevoir des circuits de tailles et de complexités si importantes que certains observateurs les présentent comme les objets les plus complexes créés par l'homme. Cela induit pour leur conception la nécessité d'utiliser des procédés entièrement automatisés et optimisés pour leur conception et leur fabrication.

Pour désigner ces circuits imprimés ou intégrés, on utilise en français comme en anglais le terme *circuits VLSI* (Very Large Scale Integrated, en anglais).

On appelle *problème de conception de circuit VLSI* (VLSI design problem, en anglais) l'ensemble du processus qui permet de déduire le circuit VLSI à partir d'un circuit logique, en prenant en compte toutes les caractéristiques de la technologie choisie concernant les composants et les réseaux.

2.1.2 Caractéristiques techniques

Suivant la technologie choisie, le problème de conception de circuit VLSI exige de prendre en compte des caractéristiques techniques et des solutions électroniques particulières. Cette section présente certaines de ces notions communes à plusieurs technologies.

Une fois positionné sur le support du circuit, un réseau forme un ensemble de lignes droites ou courbes reliant ses points terminaux. Le plus fréquemment, ces lignes sont des segments de droites positionnés en angle droit sur le support. Les points embranchement entre les différentes pistes composant un même réseau sont appelés des *jonctions*. Le nombre de segments de piste d'un même réseau qui sont incidents à la même jonction est appelé *degré de jonction*. Suivant les technologies, le degré de jonction peut être plus ou moins limité. Par exemple, pour un circuit imprimé, on essaye de limiter les jonctions au degré 2, dit *coude*, ou, au plus, au degré 3, dit *jonction en T*. Cette limitation augmente la fiabilité du circuit. En revanche, un circuit intégré peut posséder des jonctions de degrés quelconques. Notons toutefois que si les pistes ne sont positionnées que selon deux directions (horizontale-verticale), le degré de jonction est conséquemment limité.

Concernant le positionnement des réseaux sur le support, la principale difficulté est d'éviter de faire chevaucher deux pistes de réseaux différents. Chaque réseau véhicule une information qui lui est propre et ne doit pas être connecté à un autre réseau. Heureusement, le support d'un circuit VLSI peut posséder plusieurs *couches* (layer, en anglais). En fait, il est souvent impossible de faire circuler tous les réseaux d'un circuit sur une seule surface sans que ceux-ci ne se chevauchent.

Un circuit imprimé peut posséder de 1 à 12 couches (rarement plus). En fait on appelle face supérieure et face inférieure les deux couches extérieures car elles correspondent aux deux faces du support. Les couches intérieures sont quant-à-elles à l'intérieur de l'épaisseur du support en epoxy. Ainsi les composants sont positionnés uniquement sur une ou sur les deux faces du support. Comme certains composants ont besoin d'une alimentation électrique, une ou deux couches peuvent être réservées aux pistes d'alimentation.

Un circuit intégré peut posséder de 2 à 6 couches qui ont souvent chacune des caractéristiques très précises. Parfois même ces couches ne sont pas construites dans le même type de matériaux et ne respectent pas les mêmes caractéristiques techniques.

Afin de relier des points terminaux situés sur des couches différentes ou permettre une plus grande flexibilité des réseaux d'un circuit, on peut vouloir positionner un même réseau sur des couches différentes. Néanmoins un réseau doit rester connecté. Concrètement, un changement de couche est rendu possible par le placement d'un *via* qui permet de connecter électriquement les différents segments de piste d'un réseau entre des couches différentes. Pour un circuit imprimé, un via est un trou gainé de cuivre, percé au travers des couches du support en epoxy. En circuit intégré, un via est un point de contact entre les pistes de deux couches adjacentes. Ainsi, certaines technologies de circuits intégrés autorisent uniquement des vias entre deux couches alors que d'autres technologies obligent les vias à percer un support de part en part. Néanmoins, un via ne traverse pas forcément tout le support mais peut se limiter à quelques couches adjacentes. Les vias sont utiles et fréquemment nécessaires, mais ils dégradent la performance du circuit et peuvent être la cause de contraintes électrostatiques. De plus, les vias ont un coût de production non négligeable. Il est souvent préférable de minimiser le nombre de vias (si ce n'est pas au détriment d'autres critères).

La figure 2.2 a) présente un circuit logique comportant trois réseaux reliant chacun deux points terminaux. Ce circuit ne peut être dessiné sur une seule face sans que deux pistes ne se chevauchent. De même, sur deux couches, une solution est rendue possible par le percement d'un via. La vue en trois dimensions de la figure 2.2 b) donne une solution possible sur deux couches. Un des réseaux est découpé en deux sous-réseaux qui sont affectés sur deux couches différentes. Un via percé entre deux couches permet de relier les deux sous-réseaux.

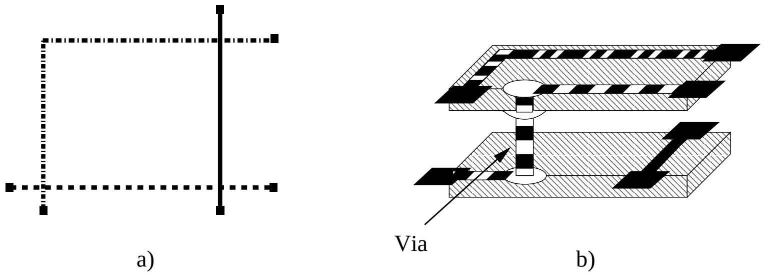


FIG. 2.2 – *Un exemple de circuit sur deux couches avec un via*

Comme nous l'avons précisé, deux réseaux différents ne doivent pas avoir de pistes connectées. Pour des raisons électrostatiques, ces connections interdites ont une définition qui dépend de la technologie employée. Par exemple, deux pistes, peuvent être dites connectées si elles sont proches de moins d'une distance fixée en fonction de la technologie choisie. Cette proximité peut être due à deux pistes positionnées l'une à côté de l'autre sur une même couche ou due à une superposition de deux pistes, sur deux couches adjacentes. Ainsi, pour un circuit imprimé, on positionne souvent sur une couche uniquement des pistes horizontales et sur les couches adjacentes uniquement des pistes verticales. En effet, cette disposition entraîne que deux pistes situées l'une sous l'autre sur deux couches adjacentes forment un angle droit. Cette position relative minimise les risques électrostatiques.

Afin de respecter ces normes d'écartement des pistes, le routage s'effectue fréquemment sur des *grilles*, c'est à dire des emplacements possibles pour les pistes, schématisés par des lignes droites verticales et horizontales suffisamment distantes pour éviter toute connection non voulue.

L'ensemble des caractéristiques techniques d'un circuit VLSI exige un positionnement complexe des composants et des pistes. Les différents critères à respecter sont souvent contradictoires: nombre de couches, nombre de vias, degré de jonction,... De plus, les supports ont souvent des surfaces réduites ou que l'on veut minimiser. A cela se rajoutent des caractéristiques plus particulières encore: longueur maximale des pistes d'un réseau, nombre de vias par réseau, synchronisation des signaux circulant sur les réseaux,... Toutes ces exigences techniques ont donné lieu à de nombreux travaux que l'on divise habituellement en plusieurs étapes de conception.

2.1.3 Etapes de conception

Le problème de conception de circuits VLSI se découpe traditionnellement en plusieurs étapes. Notons, au préalable, que ce principe d'enchaînement d'étapes interdit de certifier que le circuit produit puisse être le meilleur possible. Néanmoins, comme la conception de circuits VLSI est un problème à plusieurs critères souvent contradictoires, il est impossible de définir ce que serait un circuit "optimal". En revanche, ce découpage en étapes permet de contrebalancer la charge importante de ce travail de conception. De plus, ce découpage permet à tout moment une intervention humaine, ceci permet de corriger les imperfections éventuelles causées par l'automatisation et aussi d'orienter la conception vers une dimension humaine: par exemple on peut vouloir privilégier une division du circuit en zones prédéfinies, afin d'obtenir une meilleure visualisation du circuit (zones de calcul, zones de stockage,...). On distingue principalement trois étapes: le placement des composants, le routage des réseaux et l'affectation des réseaux aux couches.

L'étape de placement consiste à positionner les composants sur le support. Cette étape doit suivre énormément de contingences matérielles: orientation des composants, distance d'un composant à l'alimentation, logique générale du circuit, connexité réciproque des composants,... Cette étape est donc difficile à modéliser et a plutôt été traitée heuristiquement jusqu'ici. Citons néanmoins les travaux de Lengauer et Lügering [69] qui donnent des formulations entières pour les problèmes de placement et de routage. Aussi, dans [80], Möhring, Wagner et Wagner donnent une vision approfondie des problèmes combinatoires induits par ces étapes.

Une étape préalable peut précéder l'étape de placement. On appelle *étape de partitionnement* le fait de découper un grand circuit en plusieurs parties disjointes afin de traiter chaque partie indépendamment. Etant donné un circuit logique et sa liste des réseaux N, on appelle *partitionnement de la liste des réseaux* N le fait de partitionner les points terminaux du circuit de telle façon que le nombre total de liens logiques entre les partitions soit minimum. Il existe plusieurs variantes du problème selon que l'on considère les points terminaux déjà positionnés sur une surface ou non. Pour plus de renseignements sur l'étape de partitionnement, voir le survey de Alpert et Kahng [4].

Une fois que les composants et leurs points terminaux sont positionnés sur le support du circuit VLSI, on peut considérer une instance de circuits VLSI comme une liste T de points terminaux ayant chacun des coordonnées dans le plan et une liste de réseaux N où un réseau est une collection de points terminaux de T.

Etant donné une liste de points terminaux T et une liste de réseaux N, on appelle *étape de routage* le problème consistant à donner des emplacements pour des pistes reliant tous les points terminaux d'un même réseau de N sans que deux pistes de réseaux différents ne soient connectées. Cette étape est multicritère suivant que l'on veuille minimiser la surface totale du circuit, la longueur maximale des fils, le nombre de couches ou de vias. De plus, le routage diffère suivant que l'on dispose ou non d'une grille pour positionner les pistes. Nous nous intéressons plus longuement à cette étape dans la section 2.2.

A nouveau, les grandes dimensions des circuits VLSI rendent l'étape de routage difficile. Heureusement, comme le support d'un circuit VLSI est une surface planaire, on peut le diviser en plusieurs zones -par exemple des rectangles- qui peuvent être traités indépendamment. Ainsi, on peut effectuer le routage de chaque zone par des heuristiques efficaces, voire des méthodes exactes. On appelle *routage local*, ou aussi *routage détaillé*, le fait de réaliser un routage d'une de ces zones. Néanmoins, un routage local d'une zone donnée doit être compatible avec les routages locaux des zones adjacentes. Pour cela, on effectuer antérieurement une étape appelée *routage global*. Les travaux de Korte, Prömel et Steger [63] et de Grötschel, Martin et Weismantel [77, 47] donnent un aperçu des rapports entre le routage global et le routage local.

La dernière étape de conception de circuit VLSI est appelée *étape d'affectation des réseaux aux couches* (layer assignement, en anglais). Elle consiste à positionner les pistes des réseaux sur les différentes couches de manière à ce que deux pistes de réseaux différents ne soient pas connectées. Comme on place précisément les pistes sur le support, on parle aussi de *câblage* (wiring, en anglais). Cette étape possède deux critères principaux: le nombre de couches et le nombre de vias. En effet, plus le nombre de couches est grand, moins il y a besoin de vias. De même, il est possible que, compte-tenu des problèmes électrostatiques, un circuit ne puisse pas être positionné sur k couches alors qu'il le sera sur k' couches avec $k' > k$.

En pratique, le nombre de couches est fréquemment imposé par la technologie choisie, surtout pour un circuit intégré. Le fait d'avoir un nombre de couches fixé entraine alors un nombre de vias nécessaires. On peut illustrer la nécessité des vias par l'exemple de la figure 2.2. Ce circuit est composé de trois réseaux qui se croisent deux à deux sur la représentation planaire 2.2 a) du routage. Ces réseaux ne sont pas positionnables sur deux couches sans les découper et la figure 2.2 b) montre comment ce câblage est possible en utilisant un via. En revanche sur trois couches, ce circuit est *câblable* sans via en positionnant un réseau sur chaque couche.

Cette étape est parfois déjà incluse dans l'étape de routage, on parle alors de routage avec affectation des réseaux aux couches dont nous reparlons dans la section 2.2.3. Lorque l'on considère le routage avec pour critère unique le nombre de vias, on parle du *problème de Via Minimization topologique* ou, encore, *problème de Via Minimization non contraint* qui sera évoqué dans la section 2.2.1.

Néanmoins, comme nous le verrons par la suite, le routage est fréquemment réalisé sans prendre en compte l'affectation des réseaux aux couches. Ce problème devient alors un problème à part entière. Lorsque l'on considère cette étape avec pour unique critère le nombre de vias, on parle du *problème de Via Minimization contraint* qui sera longuement traité dans les sections 2.3 et 3. Ce dernier problème peut être vu comme une étape toujours valide. En effet, comme le problème de routage est un problème multicritère, on n'a aucune assurance d'avoir un nombre de vias minimum. Comme l'étape de Via Minimization contraint minimise le nombre de vias sans affecter le routage, cette étape de Via Minimization contraint peut toujours être effectuée dans le but de minimiser le nombre de vias.

Il existe d'autres étapes mineures. Citons par exemple l'optimisation de la fabrication du circuit: trouver le chemin minimal qu'un bras robot doit parcourir pour positionner les composants aux emplacements définis. On peut aussi citer les étapes permettant de synchroniser les informations circulant sur les pistes en jouant sur les paramètres de conception (largeur et longueur des pistes, éloignement des composants,...)

2.1.4 Conception de circuits VLSI en industrie

La conception de circuits VLSI est évidemment issue d'un besoin industriel. Elle dépend d'une technologie en constant développement qui rend rapidement caduque certains aspects de la conception au profit de nouveaux concepts. Les grands groupes producteurs de puces électroniques ne sont pas très nombreux et gardent pour eux la plupart de leurs secrets de fabrication. Ces groupes travaillent en collaboration avec des laboratoires et des équipes de recherche qui font vivre un domaine scientifique à part entière. Même si les connections avec les domaines de la recherche opérationnelle et de l'optimisation combinatoire sont très forts et fréquents, des chercheurs sont spécialistes du problème de conception de circuits VLSI sans être ni informaticien, ni mathématicien, ni électronicien.

Les entreprises et les laboratoires qui ont besoin de créer leurs propres circuits VLSI utilisent des suites logicielles capables de concevoir entièrement un circuit à partir d'un circuit logique. Citons par exemple l'environnenement Mentor Graphics de Mentor [57] pour les circuits imprimés et la suite logiciel Cadence VLSI Design Tools de Cadence [56] pour les circuits intégrés.

Afin de permettre aux différentes équipes de chercheurs de travailler sur les mêmes données à travers la communauté scientifique, des banques de jeux d'essais existent pour certaines étapes du processus de conception. Il existe par exemple “The ISPD98 Circuit Benchmark Suite” [54] et “ISPD02 IBM-MS Mixed-size Placement Benchmarks”

[55] qui proposent des circuits logiques de références pour tester les algorithmes de branchement et de routage.

2.2 Routage et affectation aux couches

Dans le processus de conception de circuits VLSI, l'étape de routage est un problème multicritère qui se décline en plusieurs problèmes suivant les critères envisagés. Si l'on désire considérer uniquement le nombre de vias, le problème de routage s'appelle alors problème de Via Minimization topologique. Ce dernier problème est surtout théorique car il est assez éloigné de la réalité des circuits électroniques. En pratique, l'étape de routage prend en compte le positionnement des fils et l'affectation des réseaux aux couches. On verra, entre autres, qu'une simplification fréquente de ce problème consiste à effectuer le routage sans fixer l'affectation des réseaux aux couches.

2.2.1 Le problème de Via Minimization topologique

Si l'on considère le problème de routage en omettant tous les critères sauf celui du nombre de vias, le problème de routage devient un problème topologique théorique que l'on nomme *problème de Via Minimization topologique* (ou parfois problème de Via Minimization non contraint).

Dans le cas le plus général, le problème de Via Minimization topologique peut se définir comme suit. Soit M une surface connexe délimitée par une courbure $\mathcal{C}$ et contenant des obstacles. Soit T un ensemble de points terminaux de M et $N_1, ..., N_n$ un ensemble de n réseaux, c'est-à-dire que $(N_1, ..., N_n)$ forme une partition des terminaux de telle façon que $|N_i| \geq 2$ pour $i \in \{1, ..., n\}$. On suppose que tous les terminaux sont accessibles sur toutes les couches. On appelle *réseau de courbes* une collection connexe de courbes. Le problème de Via Minimization topologique sur k couches (k-PVMT) consiste à donner n réseaux de courbes $S_1, ..., S_n$ qui joignent respectivement tous les terminaux du réseau N_i, $i = 1, ..., n$. De plus, ces réseaux de courbe peuvent circuler sur k couches et ne doivent pas se croiser sur une même couche.

La figure 2.3 schématise une instance du 2-PVMT. Il y a 8 points terminaux répartis en 3 réseaux, les points terminaux de même numérotation appartenant au même réseau. La solution proposée sur la figure comporte deux vias représentés par des ronds noirs. La différence de type de traits correspond aux deux couches.

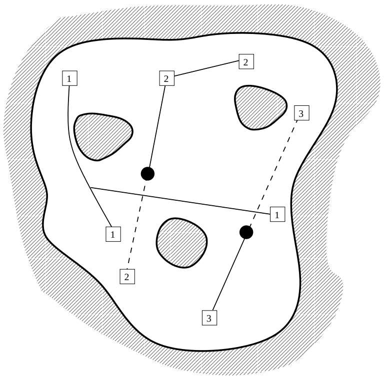


FIG. 2.3 – *Le problème de Via Minimization topologique*

Dans [76], Marek-Sadowska montre que le problème de Via Minimization topologique est NP-complet. Pour cela, il étudie un cas particulier du 2-PVMT où les points terminaux sont tous situés sur la courbure. Dans ce cas, le problème généralise le problème de planéarisation d'un graphe par suppression d'arêtes qui est connu pour être NP-complet. Dans [95], Sarrafzasdeh et Lee considèrent le 2-PVMT lorsque la surface est rectangulaire et que tous les points terminaux sont situés sur le bord avec des réseaux limités à deux points terminaux. Dans [95] et dans [6], on trouve une preuve du caractère NP-difficile de ce problème. En revanche, ce dernier problème peut être résolu en temps polynomial pour les instances où chaque réseau relie deux points terminaux situés sur des faces opposées du rectangle. Ce problème reste polynomial si le nombre de couches est fixé à k, $k \geq 2$ dans [5]. De plus, dans [5], il est aussi montré que le k-PVMT peut être résolu en temps polynomial si la surface est un disque.

Il existe de nombreux travaux heuristiques sur le k-PVMT. Citons deux approches intéressantes. Dans [2], Abdullah et Sastry essayent de déterminer le nombre minimum de vias nécessaires pour résoudre une instance du k-PVMT. Ils donnent une borne pour cette question en utilisant des algorithmes de coloration d'un graphe d'intersection. Dans [33], Cong et Liu abordent le problème de k-PVMT en mettant des poids sur les réseaux de manière à placer un ensemble de réseaux de poids maximum sur une même couche, dite alors couche préférée. Ils démontrent que ce problème est NP-difficile ou polynomial dans les mêmes cas que le k-PVMT.

2.2.2 Routage sur grilles

L'étape de routage est déduite de l'étape du placement des composants. Par conséquent, le nombre d'emplacements possibles pour le routage des réseaux est réduit. Ainsi, contrairement au problème de Via Minimization topologique, les réseaux sont routés sur des emplacements possibles prédéfinis que l'on représente sous forme d'une grille.

On appelle *grille de routage* un ensemble de lignes (droites) entrelacées formant les emplacements possibles sur lesquels peuvent être positionnés les réseaux. On suppose qu'une grille de routage est déduite des caractéristiques technologiques et des résultats de l'étape de placement des composants. Ainsi les pistes positionnées selon une telle grille respectent par exemple l'espacement nécessaire entre deux pistes et le contournement des composants. La figure 2.4 donne un exemple de grille de routage régulière, c'est-à-dire dont toutes les lignes sont espacées régulièrement. Cette grille laisse en blanc les emplacements de quatre composants où aucune piste ne peut être routée. Notons que si le circuit dispose de plusieurs couches, on considère le routage sur plusieurs grilles de routage superposées. Suivant les caractéristiques technologiques du circuit, ces grilles peuvent être identiques ou non.

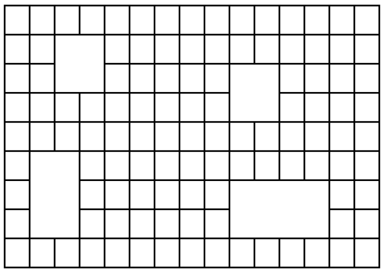


FIG. 2.4 – *Une grille de routage*

En fait, chaque technologie est associée à des types de grilles précis. La plupart des grilles ne comportent que des traits horizontaux et verticaux, mais il existe des grilles utilisant des lignes obliques.

On suppose que les points terminaux sont situés aux intersections des lignes de la grille correspondant aux contours de la grille et des composants. Mais ces points terminaux peuvent être limités uniquement aux côtés de la grille (routage switchbox) ou sur deux cotés opposés de la grille (routage en canal ou channel routing en anglais).

On appelle *problème de routage des réseaux sur la grille* le problème consistant à donner, pour chaque sous-ensemble S_i, $i \in \{1, ..., n\}$, un ensemble de pistes positionnées sur la grille, ne formant pas de cycle et reliant les points terminaux de S_i. Ce routage doit, de plus, être effectué selon trois critères d'optimisation: minimiser la surface totale de la grille utilisée, minimiser la longeur totale des pistes, minimiser le nombre total de vias.

On peut exiger qu'un réseau donné doive avoir au plus p vias, $p \geq 0$, une longueur au plus égale à L, $L \in I\!R^+$, ou que toutes ces jonctions aient un degré inférieur à d,

$d \geq 2$,...

Lors du routage, on peut interdire ou autoriser des positions particulières des réseaux les uns par rapport aux autres. Deux pistes qui se croisent en angle droit sont dites *en croisement*, voir figure 2.5 a). Le croisement est le chevauchement de pistes qui provoque le moins de problèmes électrostatiques. Comme une ligne d'une grille peut représenter plusieurs routages possibles pour des pistes, il peut y avoir un nombre maximum m de pistes pouvant être positionnées sur la même ligne. Deux pistes parallèles situées au même endroit sur une grille sont dites *en superposition* (overlap), figure 2.5 b). A l'inverse, si $m = 0$, on interdit que deux pistes de réseaux différents ne soient positionnées en chevauchement. Un cas très particulier de chevauchement est dit *en genou* (knock-knee). Il s'agit de deux réseaux partageant une jonction de degré 2, comme illustré sur la figure 2.5 c).

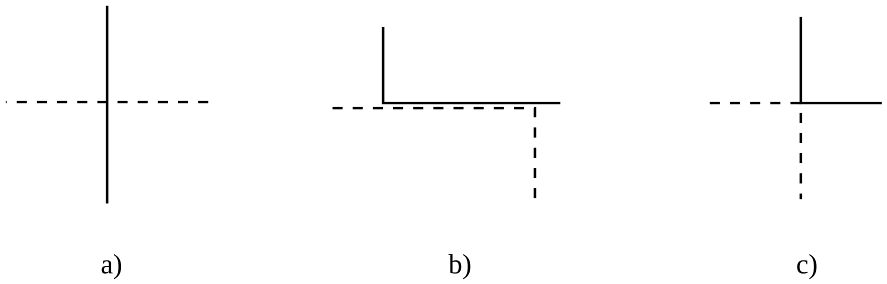


FIG. 2.5 – *Deux réseaux a) en croisement, b) en superposition, c) en chevauchement en genou*

On peut facilement définir un graphe à partir d'une grille de routage en associant des sommets aux intersections de la grille et en reliant par une arête deux sommets correspondant à des intersections voisines. On appelle les graphes ainsi obtenus des *grilles*. On appelle *grille multiple* un graphe composé de k copies superposées d'une même grille et dont on relie par une arête les copies d'un même sommet qui sont sur deux couches adjacentes. Les couches d'une grille vont bien entendu correspondre aux différentes couches du circuit. De même, les arêtes reliant deux couches correspondent aux emplacements possibles pour les vias. On identifie les points terminaux du circuit (qui sont un sous-ensemble des intersections de la grille) avec un ensemble T de sommets de G. Ainsi, les réseaux du circuit peuvent être alors vus comme une partition $(N_1, ..., N_n)$ de T.

La figure 2.6 montre une grille multiple à deux couches sur lesquelles sont positionnées trois réseaux. Pour des raisons de lisibilité, les arêtes reliant les deux couches ne sont pas représentées. Les cercles correspondent aux points terminaux de chacun des réseaux. La figure donne un routage possible des trois réseaux. Le réseau 2 porte un

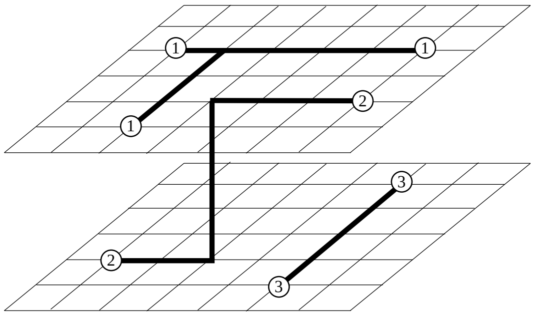


FIG. 2.6 – *Un routage de trois réseaux sur une grille multiple*

via qui est représenté par le changement de grille.

Ainsi, dans un tel graphe, les différentes variantes du problème de routage se ramènent au problème du pavage d'arbres Steiner, que l'on peut définir comme suit.

Soit $G = (V, E)$ un graphe muni d'un poids w associé aux arêtes. Soit $S \subset V$ un sous-ensemble de sommets appelés *sommets terminaux*. Le *problème de l'arbre Steiner* de poids minimum consiste à trouver un arbre couvrant S et de poids minimum. Les sommets non terminaux sont appelés *sommets Steiner*.

Soient $S_1, ..., S_n$ n ensembles disjoints de sommets terminaux de G et soit c un vecteur entier de capacités associées aux arêtes de G. Le *problème de pavage d'arbres Steiner* (the packing Steiner tree problem) consiste à trouver n arbres Steiner $T_1, ..., T_n$ dans G couvrant respectivement $S_1, ..., S_n$ tels que le nombre d'arbres parmi $T_1, ..., T_n$ utilisant une arête e de E soit inférieur ou égale à $c(e)$ et la somme $\sum_{i=1}^{n} w(E(T_i))$ soit maximum. On dit que deux arbres T et T' d'un graphe sont *arête-disjoints* (resp. *sommet-disjoints*) si T et T' n'ont aucune arête (resp. sommet) en commun. On appelle *problème de pavage d'arbres Steiner arête-disjoints* (resp. *sommet-disjoints*) le problème de pavage d'arbres Steiner tel que les arbres Steiner formant la solution soient arête-disjoints (resp. sommet-disjoints).

Le problème de pavage d'arbres Steiner est NP-difficile même lorsque $n = 1$. Dans le cas du pavage de deux arbres, il est prouvé dans [63] que le problème est NP-complet dans les graphes planaires et dans les graphes issus d'une grille à deux couches. Dans [103], Uchoa et Poggi de Aragão donnent une formulation du problème de pavage de deux arbres Steiner sommet-disjoints. Chopra [29] étudie diverses formulations en nombres entiers pour $n \geq 2$. Dans [77], Martin et Weismantel introduisent une approche polyédrale pour le problème de pavage d'arbres Steiner général. Grötschel, Martin et Weismantel poursuivent cette étude polyédrale dans [47, 45, 46, 43, 44]. Dans [58], Jorgensen et Meyling proposent une nouvelle formulation pour le problème et en déduisent un algorithme de génération de colonnes et branchements. Dans [61], Kaufmann et Mehlhorn discutent des rapports entre le problème de pavage de chemins arête-disjoints et le problème de ultiflot. Vues les grandes dimensions des instances de circuits VLSI, de nombreuses heuristiques ont été proposées pour résoudre le problème de pavage d'arbres Steiner, [63, 62, 32].

Le problème de routage demande de prendre en compte de nombreux critères: longueur total des réseaux, nombre de vias, surface totale utilisée, nombre de couches... Deux de ces critères peuvent être considérés comme principaux: minimiser la longueur totale des réseaux et minimiser le nombre de vias. Ces deux critères sont souvent en contradiction. En effet, si l'on minimise la longueur des pistes, on augmente le nombre de chevauchements des réseaux et conséquemment le nombre de vias nécessaires. En fait, si l'on définit une grille G à k couches associée à l'instance, le choix d'un poids w sur les arêtes permet d'obtenir une solution qui respecte l'un des deux critères ou bien un compromis entre les deux.

Néanmoins, le fait que ce problème soit multicritère incite à diviser le problème en deux parties: tout d'abord minimiser la longueur totale des réseaux puis minimiser le nombre de vias. En fait, c'est cette solution qui est le plus souvent choisie.

2.2.3 Routage sans affectation aux couches

Le critère principal de l'étape de routage est de minimiser la longueur totale des pistes. Si l'on considère uniquement ce critère pour le routage des réseaux, une étape spécifique d'affectation des réseaux aux couches doit être exécutée par la suite. Nous discutons d'une telle étape dans la section 2.3. D'autre part, cette limitation à un seul critère permet de réduire la taille du graphe dans lequel on résout le problème de routage, c'est-à-dire ici le problème de pavage d'arbres Steiner.

En effet, plutôt que résoudre le problème de pavage d'arbres Steiner sur une grille multiple, on peut se limiter à une grille simple. Une grille simple est une grille à une couche où chaque arête correspond à k emplacements superposés sur les k couches. A chaque arête $e \in E$, on place une capacité $c(e)$ qui correspond au nombre de réseaux qui peuvent être routés sur les emplacements correspondants. Ce modèle s'appelle *modèle knock-knee (ou en genou)* car il autorise que deux réseaux soient positionnés en genou. Si l'on interdit les chevauchements en genou, ce modèle s'appelle alors *modèle Manhattan*.

2.2.4 Le problème du câblage sur k couches

On dit qu'un circuit est *câblable* (wirable) sur k-couches, $k \geq 1$, s'il existe un routage possible des réseaux sur k couches en respectant éventuellement des caractéristiques techniques. Par exemple, il peut exister des zones où les vias sont interdits ou même simplement des zones où trop de réseaux doivent être routés. Le câblage a été évoqué principalement par Molitor [83, 85].

Une première question naturelle est donc de déterminer si un circuit est câblable sur un nombre donné de couches. Ce problème est en fait très proche du problème de Via Minimization contraint dont nous discutons dans la section 2.3. En effet, tout algorithme donnant une solution au problème de routage en affectant correctement les réseaux sur k couches, donne une solution au problème de câblage sur k couches. C'est le cas des algorithmes résolvant le problème de Via Minimization contraint. Pinter [91] a donné un algorithme polynomial pour le problème de Via Minimization contraint sur deux couches pour les réseaux avec un degré maximum de jonction limité à 3. Ainsi, le problème de câblage est polynomial pour $k = 2$ dans ce cas. En revanche, dans [72] il est montré que décider si un routage sur une grille carrée peut être réalisé sur trois couches est NP-complet. Par contre, Brady et Brown [18] ont montré que tout circuit peut être câblé sur 4 couches si l'on autorise les chevauchements en genou.

Une autre question consiste à déterminer, à partir d'une grille sur k couches où un circuit ne peut être câblé, comment augmenter la grille pour obtenir un routage possible. Brady et Sarrafzadeh ont montré dans [19] que pour une grille à trois couches où les chevauchement en genou sont autorisés, le problème est NP-complet.

Pour les circuits intégrés, le nombre de couches disponibles est fréquemment donné à l'avance pour des raisons technologiques. En effet, le rôle respectif des couches empêche de leur conférer le même rôle lors de l'affectation des réseaux. Ainsi, même pour un circuit intégré de 6 couches, le routage des réseaux peut se réaliser entièrement sur les deux couches supérieures.

En revanche, le problème du nombre de couches est crucial pour les cartes de circuits imprimés. En effet, s'il est possible d'avoir un nombre important de couches (12 ou plus), il est intéressant de minimiser le nombre de couches pour réduire les coûts de production. Néanmoins comme ce critère de nombre de couches est en contradiction avec la fiabilité du circuit (un nombre important de pistes sur des couches trop proches provoque des problèmes électrostatiques), il est souvent préférable de fixer à l'avance le nombre de couches en fonction de la densité des réseaux à router.

2.3 Le problème de Via Minimization contraint

Les sections précédentes ont présenté l'étape d'affection des circuits aux couches comme une étape facultative du processus de conception de circuits VLSI. En effet, l'affectation des circuits aux couches est parfois réalisée pendant l'étape de routage. Néanmoins, cette affectation est souvent ommise pendant le routage. Une telle ommission permet de privilégier les critères consistant à minimiser la longueur totale des réseaux ou la surface de routage. Elle permet aussi de simplifier les étapes de routage en utilisant par exemple le modèle Manhattan, voir section 2.2.3.

Le nombre minimal de vias est le critère principal de l'affectation des réseaux aux couches, on parle alors d'étape de Via Minimization contraint. Comme le processus de routage est multicritère, même si l'affectation des réseaux aux couches a été réalisée, le nombre de vias n'est pas forcément minimal. Ainsi, une étape finale de Via Minimization contraint est toujours utile.

Dans cette section, nous présentons un état de l'art du problème de Via Minimization contraint. Nous donnons ainsi une définition formelle et quelques extensions du problème.

2.3.1 Définitions

On considère un circuit VLSI pour lequel les étapes de placements des composants et de routage des réseaux ont été réalisées. Par conséquent, les réseaux sont positionnés sur le support du circuit, c'est-à-dire que des coordonnées dans le plan ont été attribuées aux points terminaux de chacun des réseaux ainsi qu'à chacune des pistes formant le réseau. On appelle *routage physique des réseaux* la donnée de ces coordonnées sur le support du circuit. On peut voir un routage physique comme le dessin de tous les réseaux du circuit sur un même plan, en oubliant la répartition des réseaux sur les couches. Par la suite, pour un routage physique donné, on appelera *réseau* la donnée des coordonnées des points terminaux et des pistes qui composent le réseau.

Un routage physique des réseaux peut donc être facilement représenté sur un plan par un ensemble de lignes qui peuvent se chevaucher. La donnée de l'emplacement des composants est alors gommée, comme le montre la figure 2.7. Sur cette figure, les quatre réseaux sont symbolisés par quatre types de lignes différents et les carrés représentent leurs points terminaux.

Rappelons, tout d'abord, deux définitions évoquées dans les sections et2.2. Une *jonction* est un point où au moins deux segments de piste d'un même réseau sont connectés. Le nombre de segments de piste qui sont incidents à la même jonction est appelé *degré*

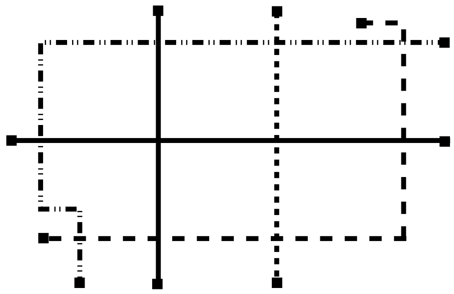


FIG. 2.7 – *Un routage physique des réseaux*

de jonction. Sur la figure 2.7, le réseau en traits noirs comporte une jonction de degré 4. On appelle *chevauchement* le fait que deux réseaux soient partiellement ou totalement superposés dans le plan du routage physique. Le plus simple des chevauchement est le *croisement*, c'est-à-dire un point où les segments de pistes de (deux) réseaux différents se rencontrent en formant un angle droit.

On appelle *affectation valide* le fait d'affecter, sur les couches du circuit, les segments de pistes de chacun des réseaux de façon à ce que deux segments de piste de réseaux différents ne se chevauchent pas sur une même couche. Concrètement, un changement de couche est rendu possible par le placement d'un via qui permet la connection électrique des différents segments de piste d'un réseau entre des couches différentes (voir section 2.3.1). Comme les vias dégradent la performance du circuit et ont un coût de production non négligeable, il est préférable de minimiser le nombre de vias (sans changer le routage). Le *problème de Via Minimization contraint sur k couches* (k-PVMC) se définit ainsi: étant donné un routage physique des réseaux, déterminer une affectation valide sur les k couches avec un nombre minimum de vias.

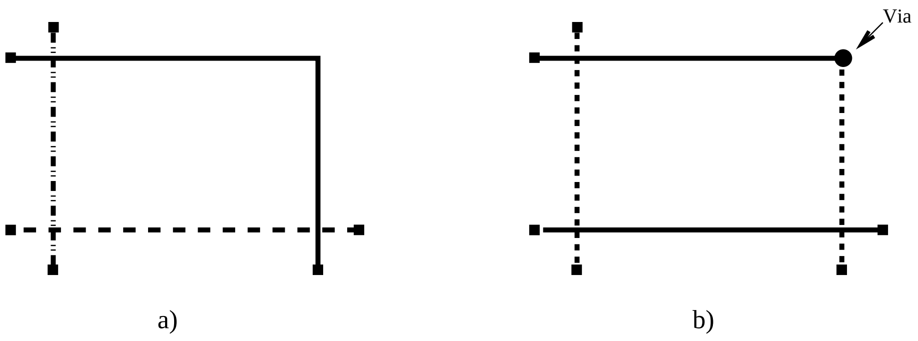


FIG. 2.8 – *Exemple de nécessité d'un via pour un circuit sur deux couches*

Observons l'utilité d'un via sur l'exemple 2.8. Le circuit de la figure 2.8 a) comporte trois réseaux qui se croisent deux à deux. On ne peut directement effectuer une affectation de ces réseaux sur deux couches: en effet, si l'on place l'un des réseaux sur une couche, les deux autres sont nécessairement sur l'autre couche et ainsi se croisent. En revanche, si l'on divise un des réseaux en deux segments, on peut affecter un segment à une couche et l'autre segment à l'autre couche. L'affectation sur deux couches est alors possible mais elle nécessite l'utilisation d'un via pour relier les deux segments. Une telle solution est proposée par la figure 2.8 b) où les deux types de traits indiquent sur quelle couche sont les réseaux ou les segments des réseaux.

La figure 2.9 illustre le fait que deux affectations valides peuvent avoir des nombres de vias très différents. En effet, les figure 2.9 a) et b) donne deux affectations valides des réseaux du circuit de la figure 2.7. La première possède 4 vias tandis que l'autre n'en nécessite que 2.

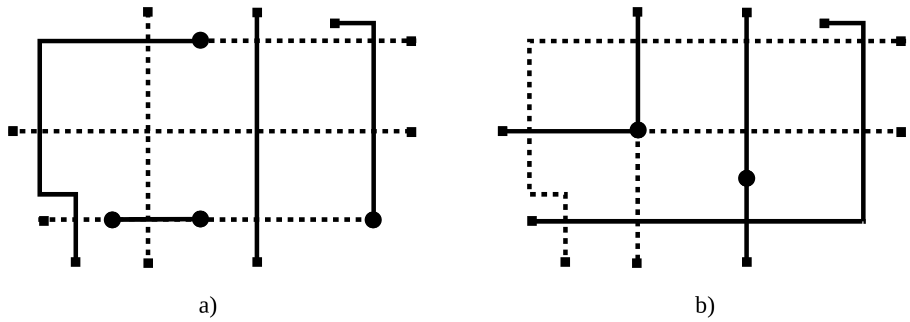


FIG. 2.9 – *Deux exemples d'affectations valides pour le circuit de la figure 2.7*

Le positionnement des pistes des réseaux les unes par rapport aux autres influe sur les positions des vias. Par exemple, on place plutôt les vias sur les jonctions en coude ou sur les jonctions de degré $d \leq 3$. De même, certains emplacements peuvent être interdits pour le placement d'un via. C'est par exemple le cas sur des pistes de réseaux différents qui sont trop proches ou se superposent.

Suivant la technologie du circuit, certaines caractéristiques physiques doivent être prises en compte. Par exemple, certaines technologies n'autorisent que des vias perçant deux couches.

Les caractéristiques des composants et du circuit peuvent aussi entraîner le fait que certains réseaux soient affectés a priori sur une couche donnée. On parle de réseau ou de segment de réseaux *pré-affectés.*

Pour des raisons de performance, on peut vouloir qu'une couche possède plus ou moins de réseaux que les autres couches. On désigne alors une *couche privilégiée* à laquelle doivent être affectés un nombre plus ou moins important de réseaux. Dans ce cas, il faut éventuellement augmenter le nombre de vias pour permettre que d'avantage de réseaux soient affectés sur la couche privilégiée.

2.3.2 Le cas 2 couches

Le problème de Via Minimization contraint a été particulièrement étudié pour le cas deux couches. Ceci est dû, entre autres raisons, au fait que les technologies permettant des circuits multicouches sont plus tardives. Le cas particuliers 2 couches est encore fréquemment utilisé pour les circuits imprimés. De plus, au sein d'un circuit intégré multicouche, les couches ont souvent des fonctions précises. Ainsi, la minimisation du nombre de via est uniquement possible qu'entre certaines couches, ce qui peut revenir à ne pouvoir traiter uniquement le cas 2 couches.

2.3.2.1 Etat des travaux

Jusqu'aux années soixante, seules les entreprises d'électronique étaient confrontées au problème de conception de circuits VLSI. A l'intérieur de ces entreprises, les concepteurs utilisaient des méthodes simples provenant de leur expérience d'électroniciens. La grande majorité des circuits respectait à l'époque la norme consistant à utiliser uniquement des réseaux de pistes horizontales-verticales et contenant uniquement des jonctions de degré 2. De plus, les cartes support ne possèdaient que deux couches, la face supérieure et la face inférieure.

La méthode la plus simple et la plus répandue pour répondre au problème d'affectation des réseaux aux couches consistait alors à affecter tous les segments de pistes verticaux sur la couche supérieure et tous les segments horizontaux sur l'autre couche. Ainsi aucune piste n'en chevauche une autre. En revanche, à chaque coude de chaque réseau, un via est nécessaire. C'est encore cette heuristique qui est développée dans [20] en 1972.

Notons néanmoins, que dans le cas d'une carte de circuits à 2 couches, cette heuristique donne facilement une solution valide pour une instance du 2-PVMC.

Hashimoto et Stevens [52] ont été les premiers à s'intéresser au 2-PVMC. En fait, ils ont été les premiers à véritablement isoler la notion de Via Minimization contraint au sein du processus de conception de circuits VLSI. Ils considèrent le problème du 2-PVMC restreint aux instances de degré maximum de jonction 2. De plus, ils limitent les emplacements possibles pour les vias aux coudes que forment les réseaux.

L'idée principale de leur modélisation est de construire un graphe planaire à partir d'une instance du 2-PVMC. Ils démontrent que résoudre une instance du 2-PVMC dans ce cas particulier se ramène à rechercher un sous-graphe biparti partiel de poids maximum. Ainsi, une bipartisation de l'ensemble des sommets du graphe correspond à une bipartition de l'ensemble des segments de réseaux de l'instance, c'est-à-dire une répartition des segments entre les deux couches.

Hadlock a développé dans [50] le premier algorithme polynomial pour le problème de la coupe maximum dans les graphes planaires. Ainsi, les travaux d'Hashimoto et Stevens ont montré que le 2-PVMC possède des cas particuliers polynomiaux.

Ciesielski et Kinnen [31] proposent une formulation du problème de 2-PVMC sous forme d'un programme linéaire en 0-1. Cette formulation permet de traiter toute instance du 2-PMVC dans le cadre très général des circuits VLSI avec degré de jonction quelconque. Néanmoins, la taille du programme linéaire résultant est exponentielle. De plus, cette formulation ne permet que d'utiliser des méthodes de branchements, ce qui rend cette formulation inutilisable en pratique. Elle est cependant la première méthode qui traite le cas général du 2-PVMC.

Le cas particulier du 2-PVMC est celui où d, le degré maximum des jonctions d'une instance de 2-PVMC, est limité à 3. Les travaux initiés par Chen, Kajitani et Chan [59, 24] d'une part et ceux de Pinter [91], d'autre part, ont pu montrer que le 2-PVMC avec $d \leq 3$ peut être réduit au problème de la coupe maximum dans un graphe planaire, et peut ainsi être résolu en temps polynomial.

Ces deux modélisations, ainsi que les travaux qui se sont appuyés dessus, sont présentés en détail dans les sections 2.3.2.2 et 2.3.2.3.

Dans [9, 11], Barahona donne une modélisation proche de celle de Kajitani. Pour les instances du 2-PVMC avec $d \leq 2$, il propose un algorithme polynomial en $O(n^{3/2}log(n))$ où n est le nombre de croisements de l'instance. Pour cela, il ramène le problème dans ce cas particulier au problème du postier chinois [10].

Dans [97], Shi propose un modèle pour le 2-PVMC quelque soit le degré de jonction des réseaux. Il ramène ce problème au problème d'équilibrage maximal dans un hypergraphe signé. Constatant qu'il n'existe pas d'algorithme pour résoudre ce problème, Shi, Vannelli et Vlach proposent dans [99] une heuristique qu'ils testent sur de petites instances de circuits VLSI. Dans [98], Shi et Brzozowski prouvent que ce dernier problème est NP-complet.

Des méthodes heuristiques ont été également développées pour le 2-PVMC, (voir Chen, Kajitani et Chan [24], de Naclerio, Masuda et Nakajima [87], de Chang et Du [23] et de Ahn et Sahni [3]). La plupart des heuristiques permettent de donner des solutions approchées pour des instances du 2-PVMC ayant un degré maximum de jonctions supérieur à 3.

Parmi les solutions optimales ayant un nombre de vias minimum, certaines solutions peuvent être préférées à d'autres pour des raisons électrostatiques. Dans [25], Chen et Koren proposent d'améliorer une solution obtenue pour une instance du 2-PVMC. Pour une instance donnée, ils définissent un indicateur probabiliste de pannes, basé, entre autres, sur la densité de vias sur une surface donnée. Leur heuristique déplace les vias le long des résaux de manière à diminuer la valeur de l'indicateur.

Enfin, plusieurs travaux traitent du problème de Via Minimization contraint hiérarchique sur deux couches. Un circuit peut être divisé en sous-circuits qui peuvent être, à leur tour, subdivisés. Par exemple, les *circuits systoliques* sont composés d'un quadrillage de sous-circuits identiques. Tous les sous-circuits identiques d'un tel circuit doivent avoir le même routage et la même affectation des réseaux aux couches. Néanmoins, ces sous-circuits sont reliés entre eux par des réseaux de pistes qu'il faut, eux-aussi, affecter aux couches. On parle dans ce cas de *problème de Via Minimization contraint hiérarchique* (2-PVMCH).

Dans [82], Molitor traite le cas des circuits systoliques. Il propose un algorithme polynomial, basé sur le modèle de Pinter, qui résout le 2-PVMCH pour ce cas particulier. Molitor généralise ensuite ses résultats dans [84] à d'autres circuits.

Molitor, Sparmann et Wagner [86] s'interrogent sur l'impact de la préaffectation des points terminaux sur des couches. Dans le cas où un circuit est divisé en plusieurs sous-circuits distincts, on considère déjà réalisée l'affectation aux couches des sous-circuits. Les auteurs définissent le problème suivant: déterminer une solution du 2-PVMC, pour le circuit tout entier, en conservant l'affectation aux couches des sous-circuits. Ils démontrent que ce problème est NP-complet. En revanche, ils définissent le cas particulier où les pistes conduisant l'alimentation électrique des composants du circuit ont déjà été affectées à une couche particulière. Ils démontrent alors que ce cas particulier est solvable en temps polynomial.

Un survey sur le 2-PVMC peut être trouvé dans [80] où Möhring, Wagner et Wagner proposent un survey général de la conception de circuits VLSI. Dans [85], Molitor présente un large survey sur les notions de câblage (voir section 2.2.4) et du problème de Via Minimization contraint.

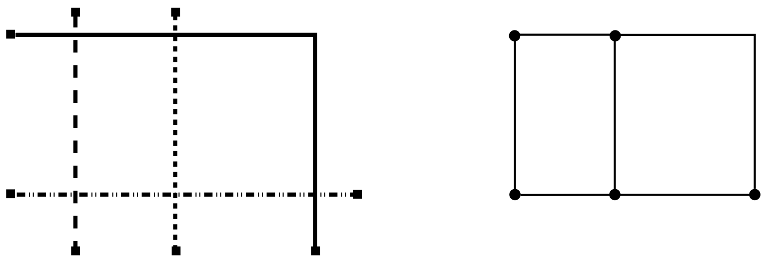
Dans les deux sections qui suivent, nous présentons les travaux issus des modèles de Kajitani [59] et de Pinter [91].

2.3.2.2 Le modèle de Kajitani

Kajitani présente dans [59] un modèle pour le 2-PVMC lorsque les réseaux ont un degré de jonction limité à 2. Il démontre que le 2-PVMC se ramène au problème de la coupe maximum dans un graphe planaire. Dans [24], Chen, Kajitani et Chan démontrent que le 2-PMVC se ramène au problème du sous-graphe biparti partiel. Dans [81], Molitor étend le modèle aux instances de 2-PVMC pour des instances de degré maximum de jonction 3.

Nous présentons ici ce modèle dans le cas des instances de degré maximum de jonction limité à 2.

Soit une instance du 2-PVMC de degré maximum de jonction 2 dont tous les points terminaux sont situés sur le bord. On suppose que deux segments de pistes de l'instance qui se chevauchent forment un croisement. Soit $G_K = (V, E)$ le graphe où les sommets de V correspondent aux croisements de l'instance et il existe une arête entre deux sommets v_1, v_2 si les deux croisements correspondant à v_1 et à v_2 sont reliés par un segment de piste d'un réseau, sans que ce segment ne porte d'autre croisement. Par construction, le graphe G_K est planaire. La figure 2.10 a) montre une instance du 2-PVMC et la figure 2.10 b) donne le graphe G_K associé.



a) b)

FIG. 2.10 – *a) Une instance du 2-PVMC et b) le graphe G_K correspondant*

En fait, lors de cette modélisation, un réseau est partitionné en segments. Ces segments sont les pistes (d'un même réseau) qui se situent entre deux croisements. Le problème du 2-PVMC pour cette instance revient à trouver une affectation valide de tous ces segments, c'est-à-dire à donner une bipartisation $V = V_1 \cap V_2$ du graphe G_K de façon à ce que les deux conditions suivantes soient respectées.

- Deux sommets adjacents correspondant à deux segments d'un même réseau sont tous deux dans V_1 ou dans V_2.
- Deux sommets adjacents correspondant à deux segments de réseaux différents sont l'un dans V_1 et l'autre dans V_2.

Ainsi le 2-PVMC, dans ce cas, se ramène à la recherche d'un sous-graphe biparti partiel de cardinalité maximum dans G_K. Comme G_K est planaire, il en résulte que le 2-PMVC peut être résolu en temps polynomial dans ce cas.

Dans [24], Chen, Kajitani et Chan indiquent comment le modèle peut être étendu aux cas où les chevauchements ne se limitent pas à un point. En revanche, dans [70], Letrouit considère les instances où les points terminaux peuvent être situés au centre du circuit. Il montre que, dans ce cas, le résultat de Chen, Kajitani et Chan ne peut être appliqué.

2.3.2.3 Le modèle de Pinter

Pinter [91] introduit une modélisation du 2-PVMC différente de celle de Kajitani. Celle-ci permet de traiter le cas des instances de degré maximum de jonctions inférieur ou égal à 3. Pinter démontre que, dans ce cas, le 2-PVMC se ramène au problème de la coupe maximale dans un graphe planaire. Dans ce qui suit, on présente ce modèle.

On considère une instance du 2-PVMC de degré maximum de jonction limité à 2. On suppose que, pour cette instance, deux segments de pistes qui se chevauchent, forment un croisement. Dans un premier temps, on énumère les segments de réseaux qui ne peuvent porter de vias. Par exemple, lorsque deux segments de réseaux disctints sont trop proches l'un de l'autre, ils ne peuvent ni l'un ni l'autre porter un via.
La figure 2.11 donne une instance du 2-PVMC. Les 7 réseaux sont tous de degré maximum de jonction 2 et sont différenciés par des types de traits distincts.

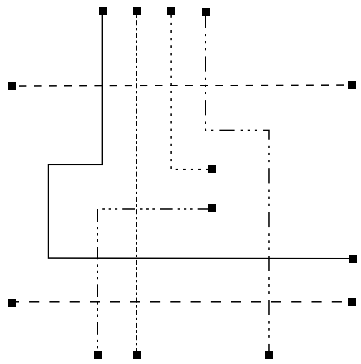


FIG. 2.11 – *Une instance du 2-PVMC*

On appelle *sections-câble* les sections croisant d'autres réseaux et ne pouvant pas porter de via et *sections libres* celles connectant deux sections-câble. Les réseaux peuvent donc être partionnés en sections-câbles et sections libres.

La figure 2.12 a) représente les différentes sections pour l'instance de la figure 2.11. Les sections-câble sont en trait épais et sont numérotées. Les sections libres sont en pointillés. Par exemple, les sections-câble 5 et 10 correspondent à deux segments de réseaux qui se croisent et qui ne peuvent pas porter de via. De même, les sections 6,7,11 et 13 sont quatre segments qui appartiennent à des réseaux distincts et qui sont trop proches les uns des autres pour pouvoir porter un via.

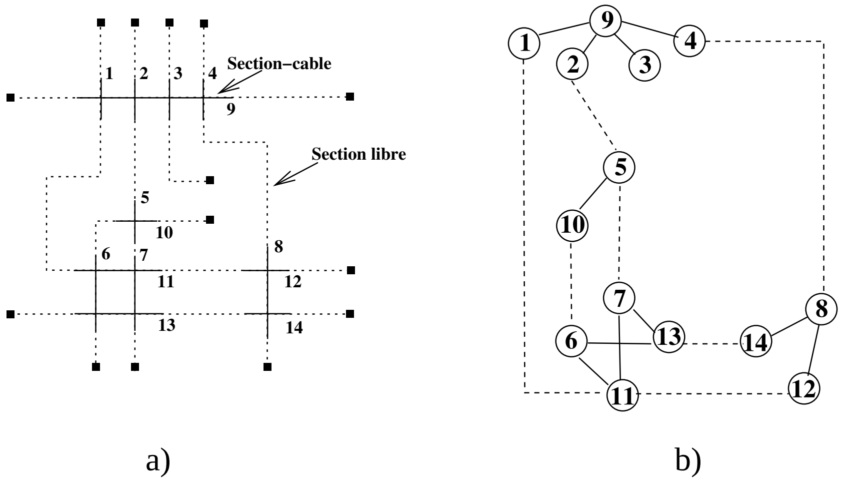


FIG. 2.12 – *a) Les sections de l'instance de la figure 2.11 et b) le graphe G_P associé*

Soit le graphe $G_P = (S, C \cup X)$ où les sommets correspondent aux sections-câble. L'ensemble des arêtes est formé de deux ensembles. L'ensemble C des arêtes dites *de continuité* et l'ensemble X des arêtes dites *de conflit*. Une arête de continuité $e \in C$ relie deux sommets correspondant à deux sections-câble connectées par une section libre. Une arête de conflit $e \in X$ relie deux sommets correspondant à des sections-câble qui se chevauchent. (On rappelle que deux segments de réseaux se chevauchent lorsqu'ils sont trop proches pour être affectés à la même couche, voir section 2.3.1). La figure 2.12 b) donne le graphe G_P associé à l'instance de la figure figure 2.12 a). Les arêtes en pointillés du graphe de la figure 2.12 b) représentent les arêtes de continuité et celles en traits pleins correspondent aux arêtes de conflit.

Considérons le graphe $G_X = (S, X)$. Noter que G_X n'est pas connexe et que chacune de ses composantes connexes correspond à un ensemble de sommets reliés par des arêtes de conflit. Si une des composantes connexes de G_X n'est pas bipartie, alors il n'existe pas d'affectation valide sur deux couches pour les sections-câble de la composante connexe. En effet, s'il existe une affectation valide des sections-câbles, alors on peut construire une bipartition des sommets de la composante de façon à ce que, pour toute arête (de conflit), ses extrémités soient dans des éléments différents de la partition.

On peut donc supposer que chacune des composantes connexes de G_X est bien bipartie. Ainsi, pour chaque composante connexe de G_X, si l'on fixe sur une couche la section-câble correspondant à l'un des sommets de la composante, alors l'affectation des autres sections-câble peut être directement déduite. Ainsi, il n'existe que deux affectations valides possibles pour chaque composante.

Parallèlement, considérons une arête de continuité v_1v_2 de C. Si les sections-câble correspondant à v_1 et v_2 sont sur deux couches différentes, la section libre correspondant à v_1v_2 doit porter un via.

Soit $G_{/X} = (V, E)$ le graphe obtenu à partir de G_P en contractant les composantes connexes de G_X et en remplaçant les arêtes parallèles qui en résultent par une arête unique. Noter que $G_{/X}$ est planaire. On associe à chaque sommet u de V un des sommets de la composante contractée. On note alors v_i le sommet de V correspondant à la section-câble i associée à ce sommet.

On associe à chaque arête $e = v_1v_2 \in E$ le poids $\sigma(e)$ égal au nombre de changements de couches nécessaires sur un réseau si les sections-câble correspondant à v_1 et v_2 sont sur une même couche. De même, on associe à $e = v_1v_2 \in E$ un deuxième poids $\delta(e)$ égal au nombre de changements de couches nécessaires sur un réseau si les sections-câble correspondant à v_1 et v_2 sont sur des couches différentes. La figure 2.13 a) donne le graphe $G_{/X}$ correspondant à l'instance de la figure 2.13 a). Sur la figure, Les arêtes $e \in E$ du graphe portent le couple de poids $(\sigma(e), \delta(e))$.

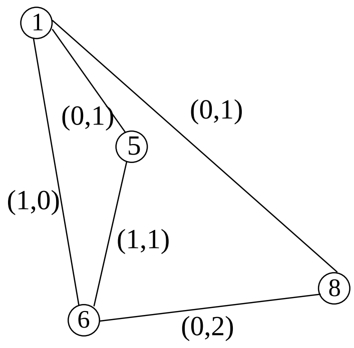


FIG. 2.13 – *Graphe $G_{/X}$ associé à l'instance de la figure 2.12*

Soit W un sous-ensemble de sommets de V. On définit alors le poids suivant associé à W.

$$z(W) = \sum_{e \in E(W)} \sigma(e) + \sum_{e \in E(\overline{W})} \sigma(e) + \sum_{e \in \delta(W)} \delta(e).$$

Posons $K = \sum_{e \in E} \sigma(e)$ et, pour tout $e \in E$, $c(e) = \delta(e) - \sigma(e)$. On peut alors réécrire $z(W)$ ainsi

$$z(W) - K = \sum_{e \in \delta(W)} c(e).$$

Pinter [91] montre que rechercher une affectation valide pour les réseaux de l'instance avec un nombre de vias minimum, se ramène à la détermination d'une coupe de poids maximum dans le graphe $G_{/X}$ muni du système de poids c.

Ce résultat peut s'étendre aux instances de degré maximum de jonction égal à 3. Pour cela, on doit modifier légèrement le graphe G_P. Si trois sections-câble se joignent en une jonction de degré 3, G_P contient trois arêtes reliant entre eux les sommets correspondant à ces sections-câble (voir la figure 2.14). On considère ces arêtes comme des arêtes de continuité et elles interviennent différemment dans le calcul des poids δ et σ.

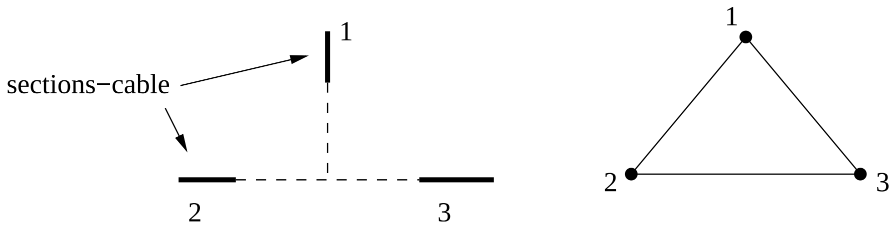


FIG. 2.14 – *Passage au degré 3 de jonction*

Ainsi, Pinter montre que le 2-PVMC dans le cas des instances de degré maximum de jonction limité à 3 se ramène au problème de la coupe maximum dans un graphe planaire. Comme, dans ce cas, le problème de la coupe maximum est polynomial, il en résulte que le 2-PVMC est polynomial dans ce cas.

Dans [64] Kuo, Chern et Shih considèrent le modèle de Pinter et donnent un algorithme de résolution en $O(n^{3/2}log(n))$ où n est le nombre de composantes connexes dans le graphe G_X.

Barahona, Grötschel, Jünger et Reinelt [12] proposent un algorithme de coupes pour le 2-PVMC basé sur le polyèdre des coupes. L'étude de ce polyèdre edt initiée par Barahona et Mahjoub dans [13].

Dans [12], les auteurs proposent une méthode pour prendre en compte la notion de couche préférée (voir section 2.3.1). Ils définissent un paramètre permettrant de fixer à l'avance le pourcentage de réseaux que l'on désire affecter sur la couche préférée. Pour une valeur donnée de ce paramètre, leur algorithme permet de donner la solution optimale du 2-PVMC.

En utilisant à nouveau le modèle de Pinter, Grötschel, Jünger et Reinelt [42] donnent une heuristique pour le 2-PVMC dans le cas des instances de degré maximum de jonction limité à 3.

2.3.3 Le cas multicouche

Lorsque le nombre de couches dépasse 2, le problème de Via Minimization contraint n'a été traité que par des méthodes heuristiques. Ces heuristiques s'appliquent à des instances de degré maximum de jonction quelconque.

Chang et Du [23] sont les premiers à proposer une heuristique pour le 3-PVMC. Elle consiste à améliorer itérativement une solution initiale en essayant d'éliminer un via par la réaffectation sur une autre couche des segments de réseaux incidents à ce via. Fang, Chang, Feng et Chen [37] améliorent cette idée en proposant deux niveaux, global et local, pour effectuer ces itérations. Dans [30], Chou et Lin proposent un recuit simulé pour des instances sur 3 couches. Leur heuristique s'appuie sur un graphe de modélisation dont ils essayent de partitionner l'ensemble des sommets en 3 sous-ensembles. Chang et Cong [22] donnent également une heuristique itérative proche d'un recuit simulé. Leur algorithme est basé sur une modélisation du problème sous forme d'un graphe dans lequel l'algorithme recherche un arbre induit.

2.3.4 Complexité

Nous avons vu dans la section précédente que le k-PVMC peut être résolu en temps polynomial si $k = 2$ et si le degré maximum de jonction est inférieur ou égal à 3 [59, 24, 91].

Choi, Nakajima et Rim [28] montrent que le 2-PVMC est NP-complet pour un degré maximum de jonction d égal à 4. Ils montrent que le problème du sous-graphe biparti induit dans un graphe planaire de degré maximum égal à 4 se ramène (en temps polynomial) au 2-PVMS avec $d = 4$. Comme le problème du sous-graphe biparti induit est NP-complet dans ce cas [28], le 2-PVMC avec $d = 4$ est également NP-complet.

Dans [83], Molitor montre que le k-PVMC pour $k \geq 3$ est NP-complet même lorsque $d \leq 3$. Pour cela, il réduit une instance du problème de 3-coloration des graphes en une instance du 3-PVMC.

2.4 Conclusion

La conception de circuits VLSI est un processus complexe qui donne lieu à plusieurs problèmes d'optimisation combinatoire. Progressivement, ce thème d'électronique est devenu un domaine de recherche en optimisation combinatoire.

Le problème de Via Minimization contraint a attiré l'attention de plusieurs chercheurs. Des dizaines de papiers ont été publiés sur différents aspects, en particulier la complexité du problème et des modèles exacts et approchés. Pour le premier type de modèle, seul le cas de 2 couches et de degré maximum de jonction de jonction ≤ 3 a été étudié. Dans le chapitre suivant, nous proposons un modèle pour ce problème quelque soit le nombre de couches et le degré maximum de jonction.

Chapitre 3

Modélisation du problème de Via Minimization contraint

Dans ce chapitre, nous donnons une modélisation pour le problème de Via Minimization contraint quelque soit le nombre k de couches et le degré maximum des jonctions. Nous montrons que ce problème se ramène à la recherche d'un sous-graphe k-parti induit dans un graphe approprié. Nous discutons également d'extensions de notre modèle pour prendre en compte certaines contraintes techniques.

3.1 Définitions

Comme on l'a vu dans le chapitre précédent, le problème de Via Minimization contraint sur k couches (k-PVMC) consiste à déterminer une affectation valide sur les k couches avec un nombre minimum de vias.

Considérons donc un routage physique des réseaux. Supposons ici que deux réseaux qui se chevauchent forment un croisement ou un chevauchement en genou. Par conséquent, ce chevauchement est limité à un point. On appelle *problème de Via Minimization contraint avec croisements* (k-PVMCC) le problème de Via Minimization contraint sur k couches lorsque les chevauchements sont limités à un point. D'autre part, nous supposons que les vias peuvent être percés en tout lieu du circuit. La figure 3.1 présente un routage physique de 4 réseaux.

Remarquons tout d'abord que tout réseau peut être décrit comme un arbre dont les sommets pendants sont les points terminaux et les autres sommets sont de degré ≥ 3.

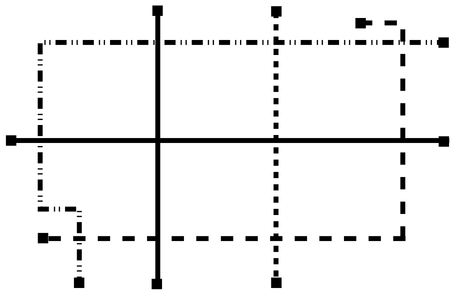


FIG. 3.1 – *Une instance du k-PVMCC*

Aussi tout sommet non terminal d'un tel arbre est une jonction de degré ≥ 3. De plus, tout point d'un segment de piste différent d'un sommet de cet arbre peut être considéré comme une jonction de degré 2.

Dans le soucis de simplifier la modélisation, nous introduisons quelques terminologies. On appelle *2-segment* toute collection de segments de piste d'un même réseau si ces segments sont adjacents et ne contiennent pas de jonction de degré ≥ 3 en tant que point intérieur. Notons qu'un 2-segment peut être vu comme une piste qui joint deux points du réseau et qui ne passent pas par une jonction de degré ≥ 3. A tout 2-segment qui ne contient pas de croisement et dont chaque extrémité est soit un point terminal, soit un croisement, on associe une jonction (de degré 2) qui correspond à un point (fixé) à l'intérieur du 2-segment. Une jonction sera appelée *jonction centrale* si elle est une jonction associée à un 2-segment ou si elle est une jonction de degré ≥ 3. Ces jonctions centrales, associées aux 2-segments, peuvent être vues comme de nouveaux points du réseau. On appelle *branche* tout 2-segment qui ne contient pas de jonction centrale et qui lie soit deux jonctions centrales, soit une jonction centrale et un terminal. On peut observer que, pour chaque jonction centrale j de degré d, il existe d branches qui ont j pour extrémité. On peut aussi remarquer que l'ensemble des branches constitue une partition du circuit. De plus, comme les réseaux ne se croisent pas, il y a exactement deux branches qui se rencontrent à chaque croisement, et, de même, qu'une branche ne peut croiser qu'au plus une autre branche.

La figure 3.2 a) montre un circuit de trois réseaux. On peut observer par exemple que le réseau en pointillés possède quatre branches, deux d'entre elles sont entourées sur la figure 3.2 b). Sur la figure 3.2 b), les jonctions centrales sont indiquées par des triangles. On peut aussi noter que le réseau en traits pleins contient une jonction de degré 4.

La première étape de résolution d'une instance du k-PVMCC est de prévoir les emplacements qui peuvent porter un via. Le lemme suivant identifie ces places pour

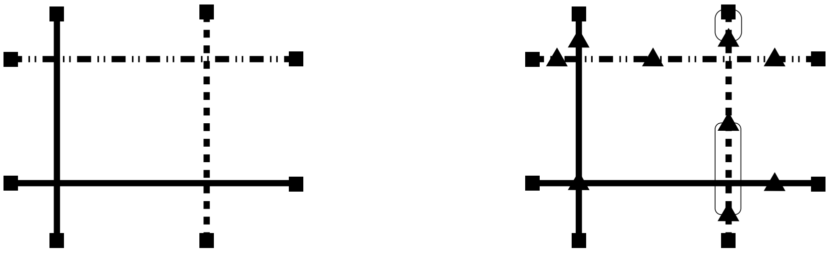


FIG. 3.2 – *Une instance du k-PVMCC et ses jonctions centrales*

notre modèle.

Lemme 3.1.1 *Etant donnée une instance du k-PVMCC, un ensemble optimal de vias peut être pris parmi les jonctions centrales.*

Preuve. Notons tout d'abord que, pour toute instance du k-PMVC, il existe une affectation valide, et, par conséquent, une affectation valide optimale. En fait, on peut d'abord affecter arbitrairement les branches de l'instance aux k-couches de manière à ce que deux branches qui se croisent soient sur des couches différentes. Les vias correspondant à cette affectation doivent alors être percés sur les jonctions centrales qui ont au moins deux de leur branches incidentes sur des couches diffférentes.
Considérons à présent une affectation valide optimale. Pour prouver le lemme, il suffit de montrer qu'une telle affectation peut être choisie sans via à l'intérieur d'une branche.
Considérons une branche b dont les extrémités sont p_1 et p_2. Soit k_i le nombre de segments de piste incidents à p_i pour $i = 1, 2$. Notons que si $k_i = 1$ (resp. $k_i > 1$), p_i est un point terminal (une jonction centrale). Pour $i = 1, 2$, nous disons que p_i est de type 1 si tous les segments de pistes incidents à p_i (et différents de b) sont affectés sur une même couche. Autrement, p_i est dit de type 2.
Supposons tout d'abord que b ne croise pas de segment de piste. Si les deux extrémités de b sont de type 2, alors deux vias sur p_1 et p_2 sont suffisants pour relier b et ses segments incidents. Comme l'affectation est optimale, aucun via supplémentaire n'est nécessaire sur b. Si une seule des extrémités, disons p_1, est de type 2, comme l'affectation est optimale, il doit y avoir exactement un via sur b. De plus, ce via peut être percé sur p_1. Si ses deux extrémités sont de type 1, au plus un via est nécessaire pour b, et il peut donc être percé sur p_1 ou sur p_2.
Maintenant supposons que b croise un segment de piste affecté à une couche, disons A. Notons par b_0 le sous-segment de b contenant le croisement. b_0 doit être affecté à une couche différente de A. Notons aussi par b_1 (resp. b_2) les segments entre p_1 (resp. p_2) et b_0. Si, pour $i \in \{1, 2\}$, p_i est soit de type 2, soit de type 1 avec tous ses

segments incidents sur la couche A, alors b_i doit porter un via qui peut être percé en p_i. Autrement, les deux extrémités p_1 et p_2 sont de type 1 avec tous leurs segments incidents sur des couches différentes de A. Ce dernier cas requiert au plus un via sur b qui peut être percé sur p_1 ou p_2. □

D'après le lemme 3.1.1, un ensemble optimal de vias peut être choisi parmi l'ensemble des jonctions centrales, ce qui revient à dire que, pour toute solution optimale, toute branche doit être entièrement affectée à l'une des couches. De plus, une jonction centrale porte un via si et seulement si au moins deux de ses branches incidentes sont affectées à des couches différentes. Par conséquent, le k-PVMCC se ramène à rechercher une affection valide des branches sur les k couches en mimisant le nombre de jonctions centrales ayant ses branches sur des couches différentes.

3.2 Modélisation du k-PVMCC

3.2.1 Le graphe de modélisation

Considérons une instance du k-PVMCC avec N réseaux, dont on note J l'ensemble des jonctions centrales et B celui des branches. On associe à l'instance le graphe $G = (V, E)$ défini comme suit: pour chaque jonction centrale $j \in J$ (resp. branche $b \in B$), on associe un sommet dans V appelé *sommet-jonction* (resp. *sommet-branche*). Une arête uv appartient à E si et seulement si soit u et v correspondent à des branches qui se croisent, soit un des sommets correspond à une jonction centrale j et l'autre à une branche incidente à j. On peut remarquer que ce graphe ne dépend pas du nombre de couches du circuit. Pour le circuit de la figure 3.1, le graphe de modélisation correspondant est montré dans la figure 3.3, où les sommets-jonction sont représentés par des triangles et les sommets-branche par des cercles.

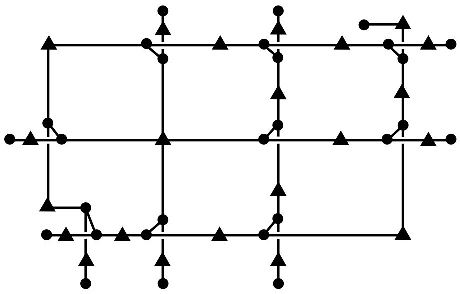


FIG. 3.3 – *Le graphe G associé à l'instance de la figure 3.1*

Dans ce qui suit, nous discutons des propriétés du graphe G.

Lemme 3.2.1

i) Un sommet-jonction est adjacent uniquement à des sommets-branche.
ii) Un sommet-branche est adjacent à au moins un sommet-jonction.
iii) Un sommet-branche est de degré au plus trois.
iv) Un sommet-branche est adjacent à au plus un sommet-branche.

Preuve. *i)*, *ii)* et *iii)* découlent directement de la construction de G.
iv) Supposons qu'il existe une chaîne consistuée de trois sommets-branche u_1, u_2, u_3. Soit b_1, b_2, b_3 les branches correspondant respectivement à u_1, u_2, u_3,. Par construction de G, cela signifie que b_2 croise à la fois b_1 et b_3. Cependant le 2-segment défini entre les croisements (b_2, b_1) et (b_2, b_3) doit porter une jonction centrale. Ceci contredit le fait que b_2 est une branche. $\square$

Considérons à présent le graphe $G^k = (V^k, E^k)$ obtenu à partir de G comme suit: pour chaque arête $uv \in E$ tel que u et v correspondent respectivement à une jonction $j \in J$ et une branche $b \in B$, on ôte uv et on ajoute un sous-graphe complet de $k-1$ sommets universels à u et à v. On note $\mathcal{K}$ l'ensemble des graphes complets de $k-1$ sommets ajoutés à G. On peut remarquer que ces graphes sont deux à deux sommets disjoints. Le graphe G^k sera appelé *graphe de modélisation*. Les nouveaux sommets de $V^k \setminus V$ seront appelés *sommets pseudo-branches*. La figure 3.4 montre le graphe de modélisation G^3 associé à l'instance de la figure 3.4.

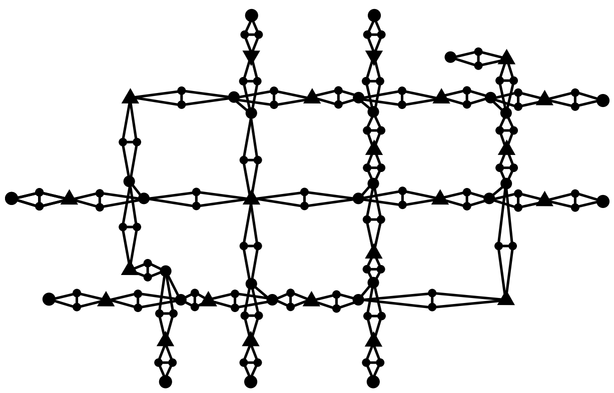


FIG. 3.4 – *Le graphe de modélisation G^3 de l'instance de la figure 3.1*

La figure 3.5 a) montre une instance du 3-PVMC et la figure 3.5 b) montre le graphe G^3 qui en est associé. On peut noter que cette instance de 3-PVMC est une des plus petites instances ne pouvant être câblées sur 3 couches sans via. En effet, une solution sans via oblige à affecter intégralement deux réseaux sur la même couche. Or, sur cette instance, les 4 réseaux se croisent les uns les autres, ce qui rend impossible une telle affectation.

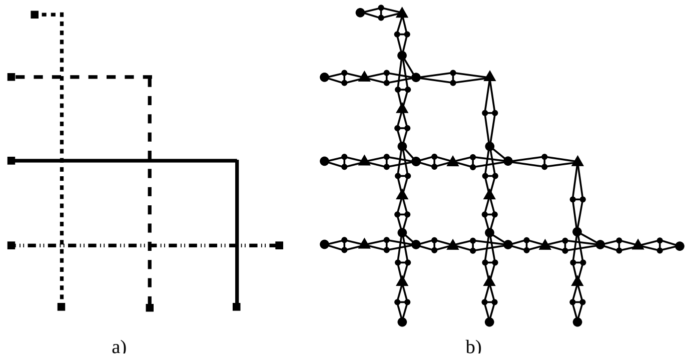


FIG. 3.5 – *a) Une instance du 3-PVMC et b) le graphe G^3 correspondant*

Comme nous allons le démontrer, le k-PVMCC est équivalent au problème du sous-graphe k-parti induit sur G^k selon un poids particulier associé à ces sommets. Afin de donner ce résultat, nous discutons d'abord de certaines propriétés du graphe de modélisation.

Lemme 3.2.2

i) Un sommet pseudo-branche v est de degré k, il est adjacent à $k-2$ autres sommets pseudo-branches, un sommet-branche et un sommet-jonction.

ii) Un sommet-jonction, de degré d dans G, est universel dans G^k à d graphes de $\mathcal{K}$.

iii) Un sommet-branche est universel à au plus deux graphes de $\mathcal{K}$.

iv) Un sommet-branche est adjacent à au plus un autre sommet-branche.

Preuve. Ces quatre propriétés peuvent être directement déduites de la construction de G et de G^k. □

Dans le reste de cette section, on note U_i l'ensemble des sommets de V qui sont associés au réseau i, $i \in \{1, ..., n\}$. On note également U_i^k le sous-ensemble de V^k formé par les sommets de U_i et les sommets pseudo-branches adjacents à U_i dans G^k, $i \in \{1, ..., n\}$.

Lemme 3.2.3

i) Toute arête ayant exactement une extrémité dans U_i^k a pour extrémité deux sommets-branche.

ii) $G^k(U_i^k)$ *est k-parti.*

iii) *Pour toute k-partition $(W_1, ..., W_k)$ de $G^k(U_i^k)$, il existe $j \in \{1, ..., k\}$, tel que les sommets-branche de U_i^k sont tous dans W_j.*

Preuve. Les propriétés i) et ii) peuvent être facilement déduites de la construction du graphe G^k et du fait que $G(U_i)$ est un arbre pour $i = 1, ..., N$.
iii) Il suffit de prouver que la propriété est vraie pour deux sommets-branche u_1, u_2 adjacents dans G au même sommet-jonction v. Remarquons tout d'abord que u_1 (resp. u_2) est universel à un graphe K^1 (resp. K^2) de $\mathcal{K}$. Maintenant, s.p.d.g., supposons que $u_1 \in W_1$. Alors $|W_i \cap V^k(K^1)| = 1$ pour $i = 2, ..., N$. Puisque v est universel à K^1, v doit appartenir à W_1. Et, puisque v est aussi universel à K^2, on a $|W_i \cap V^k(K^2)| = 1$, pour $i = 1, ..., N$. Ce qui implique que u_2 appartient à W_1. □

3.2.2 Equivalence avec le k-PSPI

Dans cette section, nous démontrons que le k-PVMCC est équivalent au k-PSPI sur le graphe de modélisation G^k.

On associe, aux sommets du graphe de modélisation G un vecteur poids $\bar{c} \in I\!R^V$ défini par $\bar{c}(u) = 1$ si u est un sommet-jonction et $\bar{c}(u) = 3$ si u est un sommet-branche ou un sommet pseudo-branche. On a alors le résultat suivant.

Lemme 3.2.4 *Tout sous-graphe k-parti induit de G^k de poids maximum selon c contient tous les sommets-branche et tous les sommets pseudo-branches.*

Preuve. Supposons, au contraire, qu'il existe un sous-graphe k-parti induit de poids maximum $\widetilde{G}^k = (\widetilde{V}^k, E^k(\widetilde{V}^k))$ et un sommet-branche ou sommet pseudo-branche v_0 en dehors de $\widetilde{V}^k$. Prenons $(\widetilde{V}_1^k, ..., \widetilde{V}_k^k)$ une k-partition de $\widetilde{V}^k$. On peut remarquer que v_0 doit être adjacent à chacun des $\widetilde{V}_i^k$, $i = 1, ..., k$. Dans le cas contraire, si, par exemple, $\widetilde{V}_i^k$ n'est pas adjacent à v_0 pour $i \in \{1, ..., k\}$, alors en considérant les ensembles $V_i' = \widetilde{V}_i^k \cup \{v_0\}$, $V_j' = \widetilde{V}_j^k$, pour $j \in \{1, ..., k\} \setminus \{i\}$, il est clair que le sous-graphe induit par $V' = V_1' \cup ... \cup V_k'$ est k-parti. De plus, comme $\bar{c}(v_0) > 0$, on obtient $c(V') > c(\widetilde{V}^k)$, une contradiction. Par conséquent, v_0 est adjacent à au moins k sommets de $\widetilde{V}^k$.
A présent, si v_0 est un sommet pseudo-branche, par le lemme 3.2.2 i) et la remarque précédente, un de ses voisins est un sommet-jonction, disons w_0, contenu dans $\widetilde{V}^k$. Ainsi, l'ensemble de sommets $(\widetilde{V}^k \setminus \{w_0\}) \cup \{v_0\}$ induit un sous-graphe k-parti de G^k. Comme $\bar{c}(v_0) > \bar{c}(w_0)$, ceci est impossible.

Maintenant, supposons que tous les sommets pseudo-branches sont dans $\widetilde{V}^k$. Alors v_0 est un sommet-branche. Comme v_0 est adjacent à au moins k sommets de $\widetilde{V}^k$, par le lemme 3.2.2 iii), v_0 est adjacent dans G^k à un ou deux graphes de $\mathcal{K}$. Aussi, par le lemme 3.2.2 iv) v_0 peut être adjacent à au plus un sommet-branche. Pour le reste de la preuve, nous supposons que v_0 est adjacent dans G^k à deux graphes K^1 et K^2 de $\mathcal{K}$ et à un sommet-branche b. (Les autres cas où v_0 est adjacent à exactement un graphe de $\mathcal{K}$ et adjacent ou non à un sommet-branche, peuvent être traités de la même manière). De plus, par le lemme 3.2.2 i) il existe deux sommets-jonction distincts v_1 et v_2 universels respectivement à K^1 et K^2. Posons $V'' = \widetilde{V}^k \setminus \{v_1, v_2\}$ et notons $V''_1, ..., V''_k$ les restrictions de $\widetilde{V}^k_1, ..., \widetilde{V}^k_k$ à V''. Clairement, $(V''_1, ..., V''_k)$ induit un sous-graphe k-parti de G^k. Supposons que b appartient à $\widetilde{V}^k$. S.p.d.g., on peut supposer que $b \in V''_1$. Comme les graphes K^1 et K^2 ne sont pas adjacents,, les sommets de $K^1 \cup K^2$ peuvent être réorganisés de façon à ce que chaque V''_i, $i = 1, ..., k-1$, contienne un sommet de K^1 et un sommet de K^2. Comme v_0 est seulement adjacent à K^1, K^2 et b, on obtient une partition $(V''_1, ..., V''_k)$ telle que v_0 n'est adjacent à aucun sommet de V''_k. Ainsi, les ensembles $V''_1, ..., V''_{k-1}, V''_k \cup \{v_0\}$ induisent un sous-graphe k-partis de G^k. Comme $\bar{c}(v_0) > \bar{c}(v_1) + \bar{c}(v_2)$, on obtient une contradiction.
Si b n'appartient pas à $\widetilde{V}_k$, on obtient de la même façon une contradiction. □

Rechercher une affectation valide des réseaux sur les k couches est à présent équivalent à rechercher une partition $(W_1, ..., W_k)$ qui induit un sous-graphe k-parti du graphe de modélisation, de façon à ce que toute arête, dont les extrémités sont des sommets-branche, soit entre deux éléments de la partition. De plus, si u est un sommet-jonction adjacent à k sommets pseudo-branches $w_1, ..., w_k$ avec $w_i \in W_i$ pour $i = 1, ..., k$, alors la jonction correspondant à u doit porter un via. Maintenant, on peut établir le résultat principal de cette section.

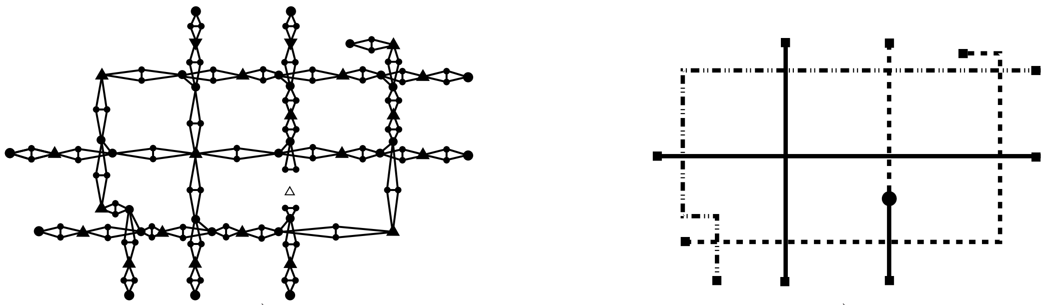
Théorème 3.2.5 *Le k-PVMCC est équivalent au k-PSPI sur le graphe $G^k = (V^k, E^k)$ avec le vecteur poids $\bar{c}$.*

Preuve. Nous montrons dans un premier temps qu'une affectation valide des réseaux sur k couches correspond à un sous-graphe k-parti induit $\widetilde{G}^k = (\widetilde{V}^k, E^k(\widetilde{V}^k))$ de G^k, où $V_0 = V^k \setminus \widetilde{V}^k$ est l'ensemble des sommets-jonction qui correspondent aux vias.
Considérons une affectation valide des réseaux aux k couches. Par le lemme 3.1.1, il existe un sous-ensemble J_0 de jonctions centrales qui correspondent aux vias. Soit V_0 l'ensemble de sommets-jonction correspondant à J_0. Il suffit de montrer que le sous-graphe $\widetilde{G}^k = G^k(V^k \setminus V_0)$ est k-parti. Pour cela, nous devons construire une k-partition $(V_1, ..., V_k)$ de $\widetilde{G}^k$. Dans ce but, nous affectons tout d'abord chacun des sommets-branche de $\widetilde{G}_k$ à V_i, si la branche correspondante est sur la couche i, $i \in \{1, ..., k\}$. On peut

noter que les sommets-branche provenant de la même couche sont deux à deux non-adjacents. Dans ce qui suit, nous donnons une procédure qui permet d'affecter les autres sommets aux différents $V_i's$ de telle manière que $(V_1, ..., V_k)$ soit une k-partition de $\widetilde{G}^k$. Soit $\widetilde{U}_i = U_i^k \setminus V_0$ pour $i = 1, ..., N$. Soit W_0 l'ensemble des sommets d'une composante connexe de $\widetilde{U}_i$, $i \in \{1, ..., N\}$. On affirme que les sommets-branche de W_0 appartiennent tous au même ensemble V_j, pour un $j \in \{1, ...k\}$. Pour montrer cela, on considère tout d'abord deux sommets-branche b_1 et b_2 de W_0. Supposons que b_1 et b_2 soient adjacents dans G au même sommet-jonction w_0. Comme toute chaîne joignant b_1 et b_2 dans G^k passe par w_0 et que $G^k(W_0)$ est connexe, alors w_0 appartient à W_0, et donc w_0 ne correspond pas à un via. Par conséquent, les branches correspondant à b_1 et b_2 sont sur la même couche, ce qui implique par construction de $V_1, ..., V_k$, que b_1 et b_2 sont dans le même ensemble V_j. A présent, comme toute paire de sommets-branche du même réseau est reliéé dans G par une chaîne consistant en une séquence alternée de sommets-branche et de sommets-jonction, par le principe décrit ci-dessus, on montre que tous les sommets-branche de W_0 sont dans le même ensemble V_j. Par le lemme 3.2.3 ii) et iii), il existe une k-partition $(W_1, ..., W_k)$ de $G^k(U_i^k)$ et $l \in \{1, ..., k\}$ tel que les sommets-branche de U_i^k sont tous dans W_l. S.p.d.g., on peut supposer que $l = j$. Maintenant, plaçons les sommets de $W_0 \cap W_t$ dans V_t pour $t = 1, .., k$. Observons que, à cette étape, les sommets d'un même ensemble V_t sont tous deux à deux non-adjacents. Par le lemme 3.2.3 i), les arêtes connectant les différents $\widetilde{U}_i$ sont toutes entre des sommets-branche. Comme notre affectation des réseaux est valide, ces sommets-branche doivent provenir de couches différentes. Ainsi, si on répète la procédure décrite ici pour chaque composante connexe de $\widetilde{U}_i$ et pour tout $i \in \{1, ..., N\}$, alors on obtient une partition $(V_1, ..., V_k)$ de $V^k \setminus V_0$ qui induit un graphe k-parti. Ceci implique que $\widetilde{G}^k$ est k-parti.
Réciproquement, posons $\widetilde{G}^k = (\widetilde{V}^k, E^k(\widetilde{V}^k))$ un sous-graphe k-parti induit de G^k. Soit $(\widetilde{V}_1^k, ..., \widetilde{V}_k^k)$ une k-partition de $\widetilde{V}^k$. Par le lemme 3.2.4, $\widetilde{G}^k$ contient tous les sommets-branche et tous les sommets pseudo-branches de V^k et, par conséquent, les sommets de $V^k \setminus \widetilde{V}^k$ sont tous des sommets-jonction. Prenons B_i l'ensemble des sommets-branche contenus dans $\widetilde{V}_i^k$, $i = 1, ..., k$. Alors, si on affecte les branches correspondantes aux sommets de B_i à la couche i, pour $i = 1, ..., k$, alors on obtient une affectation valide. Celle-ci s'obtient en perçant un via sur chaque jonction centrale correspondant à un sommet de $V^k \setminus \widetilde{V}^k$. On affirme que cette affectation utilise exactement $|V^k \setminus \widetilde{V}^k|$ vias. En fait, supposons qu'il existe une jonction centrale de degré d, correspondant à un sommet de $V^k \setminus \widetilde{V}^k$, disons w_0, qui ne correspond pas à un via dans cet affectation. Alors, on a d branches incidentes à cette jonction qui sont affectées à la même couche. De plus, les sommets-branche correspondants, disons $b_1, ..., b_d$, sont dans le même ensemble $\widetilde{V}_j^k$, pour un $j \in \{1, ..., k\}$. Par le lemme 3.2.2 i) et ii), pour $i \in \{1, ..., d\}$, il existe un graphe K_i de $\mathcal{K}$ universel à la fois à w_0 et b_i. Comme tous les sommets du sous-ensemble $W = \{w_0, b_1, ..., b_d, V(K^1), ..., V(K^d)\}$ proviennent du même ensemble,

par le lemme 3.2.3 ii) et iii) on obtient une partition $(W_1, ..., W_k)$ de W qui induit un sous-graphe k-parti de G^k et telle que tous les sommets-branche $b_1, ..., b_d$ sont dans le même ensemble W_l, pour un $l \in \{1, ..., k\}$. S.p.d.g., on peut supposer que $l = j$. Si on affecte le sommet pseudo-branche de W_t à $\widetilde{V}_t^k$, pour $t = \{1, ..., k\} \setminus \{j\}$, et w_0 à $\widetilde{V}_j^k$, la partition obtenue $(\widetilde{V}_1^k, ..., \widetilde{V}^k)$ induit encore un sous-graphe k-parti de G^k. Comme $c(w_0) > 0$, ceci contredit le fait que $\widetilde{V}^k$ est de poids maximum, ce qui termine la preuve de notre théorème. $\square$

Comme nous l'avons montré, le k-PVMCC se ramène à la recherche d'un sous-graphe k-parti induit $\widetilde{G}^k = (\widetilde{V}^k, E^k(\widetilde{V}^k))$ de G^k où les sommets de $V^k \setminus \widetilde{V}^k$ correspondent aux vias. De plus, si $\widetilde{V}^k = (\widetilde{V}_1^k, ..., \widetilde{V}_k^k)$ est une k-partition de $\widetilde{V}^k$, une solution optimale du k-PVMCC peut être obtenue en affectant les branches correspondant aux sommets-branche de $\widetilde{V}_i^k$ à la couche i, pour $i = 1, ..., k$.



a) b)

Fig. 3.6 – *a) Un sous-graphe 3-parti induit de poids maximum du graphe G^3 correspondant à l'instance de la figure 3.1 et b) la solution correspondante pour le 3-PVMC*

Afin d'illuster ceci, considérons l'instance du k-PVMCC de la figure 3.1. La figure 3.7 a) montre un sous-graphe 3-parti induit de poids maximum du graphe de modélisation G^3 de la figure 3.4 (ici on considère $k = 3$.). Notons que le sous-graphe est obtenu en ôtant un des sommets-jonction, ce sommet-jonction ôté est représenté par un triangle blanc. la figure 3.7 b) représente l'affectation valide optimale des réseaux sur 3 couches. Chaque type de lignes correspond à une des couches. Le via est représenté par un cercle plein et correspond au sommet ôté à G^3.

3.2.3 Le cas 2 couches

3.2.3.1 Les graphes G et G^2

Par le théorème 3.2.5, résoudre une instance du 2-PVMC revient à rechercher un sous-graphe biparti induit de poids maximum dans le graphe G^2 associé à l'instance. Nous montrons ici que résoudre une instance du 2-PVMC peut se ramener à la recherche d'un sous-graphe biparti induit dans le graphe G associé à l'instance. Cette preuve utilise le fait qu'un graphe biparti est un graphe sans cycle impair.

Soit une instance du 2-PVMC. On associe, aux sommets du graphe G correspondant à l'instance, un vecteur poids $\bar{c} \in I\!R^V$ défini par $\bar{c}(u) = 1$ si u est un sommet-jonction et $\bar{c}(u) = 3$ si u est un sommet-branche. On a alors le résultat suivant.

Lemme 3.2.6 *Tout sous-graphe biparti induit de G de poids maximum par rapport à $\bar{c}$ contient tous les sommets-branche.*

Preuve. Supposons, au contraire, qu'il existe un sous-graphe biparti induit de poids maximum $G' = (V', E(V'))$ et un sommet-branche u_0 qui ne soit pas dans V'. Alors u_0 doit avoir au moins deux voisins dans V'. Si ce n'est pas le cas, comme les poids sont strictement positifs, u_0 peut être ajouté à V', ce qui contredit le fait que G' est de poids maximum. De plus, par le lemme 3.2.1 iii), u_0 est de degré ≤ 3 dans G. Par conséquent, dans G', u_0 est de degré 2 ou 3. Supposons que u_0 soit de degré 3 dans G. Par le lemme 3.2.1 iv), u_0 possède deux voisins v_1, v_2 qui sont des sommets-jonction. Posons $V'' = (V' \setminus \{v_1, v_2\}) \cup \{u_0\}$. Comme V' induit aussi un sous-graphe biparti de G et $\bar{c}(u_0) > \bar{c}(v_1) + \bar{c}(v_2)$, on a une contradiction. Le cas où u_0 est de degré 2 peut être traité de manière similaire. □

Nous donnons ici une preuve alternative au théorème 3.2.5 dans le cas particulier où $k = 2$ et utilisant le graphe G.

Théorème 3.2.7 *Le 2-PVMC est équivalent au PSBI sur le graphe G muni du vecteur-poids $\bar{c}$.*

Preuve. On montre tout d'abord que toute affectation valide correspond à un sous-graphe biparti induit $G' = (V', E')$ de G, où $V_0 = V \setminus V'$ correspond à l'ensemble des jonctions qui portent les vias. Considérons une affectation valide. Par le lemme 3.1.1 il existe un sous-ensemble J_0 des jonctions centrales qui correspond aux vias. Posons

V_0 l'ensemble des sommets-jonction qui correspondent à J_0. Il suffit de montrer que le sous-graphe $G(V \setminus V_0)$ est biparti. Pour cela, supposons le contraire. Soit C un cycle impair de $G(V \setminus V_0)$. Par le lemme 3.2.1, C doit contenir au moins un sommet-jonction. De plus, tous les sommets-jonction de C sont libres de via. Aussi, par le lemme 3.2.1 i), les voisins de C doivent être affectés à la même couche. Associons le label A (resp. B) à chaque sommet-branche de C correspondant à une branche de la couche A (resp. B). Ainsi, les sommets de C adjacents au même sommet-jonction de C ont le même label. Remarquons que le label de chaque sommet-jonction est l'opposé de celui de ses voisins. Par conséquent, les sommets de chaque arête de C ont le même label. Comme C est impair, ceci est impossible.
Réciproquement, considérons un graphe $G' = (V', E')$ où $V' = V'_1 \cup V'_2$ induit un sous-graphe biparti de poids maximum du graphe G. Par le lemme 3.2.6, G' contient tous les sommets-branche et, par conséquent, les sommets de $V \setminus V'$ sont des sommets-jonction. Posons W_1 (resp. W_2) les ensembles de sommets-branche contenus dans V'_1 (resp. V'_2). Ainsi, si on affecte les branches correspondant aux sommets de W_1 sur la couche, disons A, et ceux qui correspondent aux sommets de W_2 sur la couche B, on obtient une affectation valide. Cette affectation implique le percement de via sur chacune des jonctions centrales correspondant aux sommets de $V \setminus V'$. Cette affectation utilise exactement $V \setminus V'$ vias. En effet, supposons que l'une des jonctions centrales correspondant à un sommet de $V \setminus V'$, que l'on note u_0, ne porte pas de via dans cette affectation. Alors, cela signifie que toutes les branches incidentes à cette jonction sont affectées à une même couche et ainsi que tous les sommets-branche adjacents à u_0 dans G sont soit dans W_1, soit dans W_2. Par le lemme 3.2.1 i), si on ajoute u_0 à V', le graphe induit obtenu est encore biparti. Comme $c(u_0) > 0$, on a une contradiction avec le fait que V' est de poids maximum. Ceci termine la preuve de notre théorème. □

3.2.3.2 2-PVMCC et bipartisation

Dans ce qui suit, nous précisons comment une solution d'une instance du 2-PVMCC peut être facilement déduite d'un sous-graphe biparti maximum du graphe G associé à l'instance.

Considérons une instance du k-PVMCC et son graphe de modélisation $G^k = (V^k, E^k)$. Le problème du k-PVMCC pour un $k \geq 3$ se ramène non seulement à rechercher un sous-graphe k-parti induit $\widetilde{G}^k = (\widetilde{V}^k, E^k(\widetilde{V}^k))$ de G^k mais aussi à trouver une k-partition $\widetilde{V}^k = (\widetilde{V}^k_1, ..., \widetilde{V}^k_k)$ de $\widetilde{V}^k$. Or la recherche d'une k-partition d'un graphe est elle-même un problème NP-complet lorsque $k \geq 3$.

En revanche, pour $k = 2$, rechercher une bipartition de G^2 (ou même de G) peut se faire efficacement en temps polynomial. Le 2-PVMC se ramène ainsi simplement à la recherche d'un sous-ensemble de sommets V' induisant un plus grand sous-graphe biparti de G. La recherche d'une bipartition (V'_1, V'_2) de V' peut être réalisée en temps polynomial et une solution optimale du 2-PVMC est obtenue en affectant les branches correspondant aux sommets-branche de V'_1 (resp. V'_2) sur la couche 1 (resp. 2).

Rechercher un sous-graphe biparti induit dans G revient en fait à briser tous les cycles impairs de G en ôtant des sommets. D'après le théorème 3.2.6, les sommets ôtés pour bipartiser optimalement le graphe G sont tous des sommets-jonction. On peut donc en déduire que tous les cycles impairs du graphe G contiennent au moins un sommet-jonction.

On peut aussi remarquer qu'un cycle impair, dans le graphe G, correspond à une zone de l'instance devant porter un via. Par exemple, prenons l'instance de la figure 3.1 et son graphe G correspondant donné dans la figure 3.3. Comme les arêtes du graphe G sont dessinées en reprenant les positions des réseaux de l'instance, on peut visuellement repérer de quel réseau provient un sommet de G. Ainsi, les sommets d'un cycle impair du graphe G proviennent en fait d'un ensemble de réseaux, disons $N_1, ..., N_p$, tels que N_i croise N_{i+1}, pour $i = 1, ..., p-1$ et N_p croise N_1.

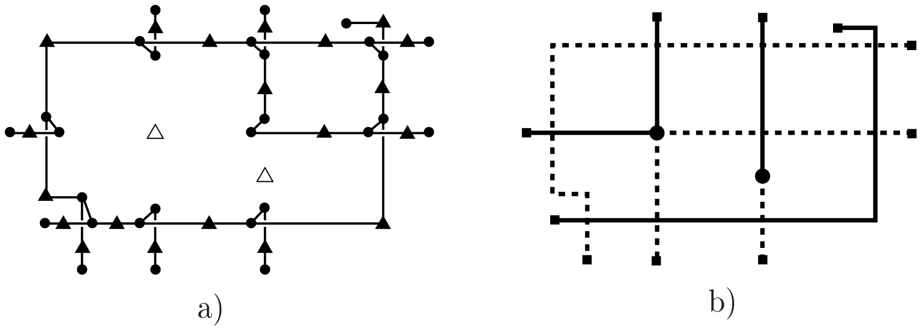


FIG. 3.7 – *a) Une solution du PSBI pour le graphe G correspondant à la figure 3.1 et b) la solution du 2-PVMC correspondante*

Par exemple pour le graphe de modélisation de la figure 3.3 provenant de l'instance de la figure 3.1, le sous-graphe biparti induit maximum est donné par la figure 3.7 a). On peut remarquer que ce sous-graphe est obtenu en ôtant deux sommets-jonction qui brisent tous les cycles impairs du graphe G. La figure 3.7 b) donne l'affectation optimale des réseaux correspondante où chaque type de lignes correspond à une couche. Les vias sont représentés par des cercles pleins noirs et correspondent en fait aux sommets ôtés du graphe de modélisation.

En comparant les solutions du 2-PVMC et du 3-PVMC pour cette instance, on peut remarquer que lorsqu'on dispose davantage de couches, le nombre de vias nécessaires diminue.

3.3 Extensions du k-PVMC

Dans cette section, nous discutions des extensions du k-PVMC dans le but de pouvoir prendre en considération des cas pratiques techniques.

3.3.1 Vias interdits

En pratique, certains segments de piste peuvent ne pas pouvoir porter de vias. On peut facilement prendre ce cas dans notre modèle en associant un poids suffisamment grand à tout sommet du graphe de modélisation qui correspond à un tel segment. Le choix de ce poids doit être fait de telle façon que le poids de tout sommet-branche et de tout sommet pseudo-branche soit au moins trois fois le poids d'un sommet-jonction.

On peut remarquer que dans ce cas, le k-PVMCC peut ne pas avoir de solution réalisable. Ceci correspond, dans notre modèle, au fait que le sous-graphe k-parti optimal ne contienne pas tous les sommets de grands poids. En fait, cela signifie que toute affectation valide doit percer un via sur un segment qui ne peut en porter un.

3.3.2 Préaffectation

Certains segments de piste sont parfois préaffectés à des couches. Bien entendu, par hypothèses, deux branches qui se chevauchent ne sont pas préaffectées sur la même couche. Dans le but de prendre en considération ce cas particulier, nous ajoutons un graphe complet H de k sommets à notre graphe G^k. On réalise ainsi une correspondance entre chaque couche i et un sommet h_i de ce sous-graphe, $i = 1, ..., k$. Pour chaque branche préaffectée sur une couche, on relie le sommet-branche correspondant à tous les sommets de H qui ne correspondent pas à cette couche. De plus, on associe un très grands poids aux sommets de H. On nomme toujours le graphe résultat par G^k, alors le théorème 3.2.5 peut être facilement étendu à ce cas.

3.3.3 Réseaux en superposition

Une situation qui est fréquente dans la pratique est lorsque des réseaux se superposent. Remarquons tout d'abord que si k réseaux se superposent les uns sur les autres, une affectation valide nécessite au moins k' couches avec $k' \geq k$.

Une zone commune à deux réseaux qui se superposent est appelée une *zone de superposition*. Remarquons que tout croisement est une zone de superposition qui est réduite à un seul point. On appelle *croisement étendu* tout point d'une zone de superposition qui est soit l'extrémité d'une zone de superposition, soit une jonction de degré ≥ 3 dont au moins un des segments incidents sort de la zone de superposition. On peut remarquer qu'un croisement est un croisement étendu.

Les définitions de 2-segment, de jonction centrale et de branche, données dans la section 3.1, restent valides quand croisement est remplacé par croisement étendu. Une jonction centrale sera dite *empilée* sur une branche b si elle est incidente à une branche qui se superpose à b. On peut noter qu'une jonction centrale qui est un croisement étendu est en même temps une jonction empilée. Par contre, une jonction empilée peut ne pas être un croisement étendu.

L'instance de la figure 3.8 a) montre trois réseaux avec deux zones de superposition. Les croisements étendus sont indiqués par des cercles. Dans la figure 3.8 b), les triangles correspondent à des jonctions centrales de l'instance. Notons que la jonction centrale j_1 est aussi un croisement étendu. De plus, la jonction centrale j_3 est empilée sur une branche entre les jonctions centrales j_1 et j_2.

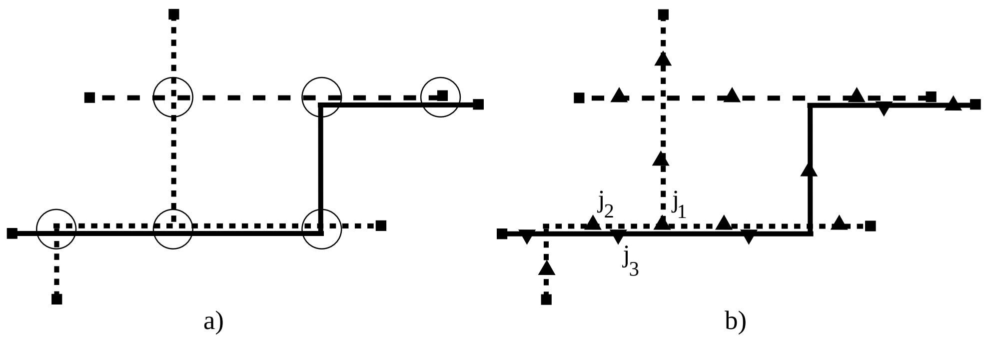


FIG. 3.8 – *Une instance du k-PVMC avec deux réseaux en superposition*

Observons que, par construction, une branche contient au plus un croisement étendu. On peut aussi noter qu'une jonction centrale empilée peut être utilisée pour porter un via. Un via percé sur une jonction centrale empilée sera nommé un *via empilé*. Ici, nous supposons que les vias empilés ne sont pas autorisés. Avec cette hypothèse, nous montrons, avec des idées similaires à celles du lemme 3.1.1, qu'un ensemble optimal de vias peut toujours être pris parmi les jonctions centrales.

On appelle *problème de Via Minimization contraint sur k couches avec superposition* (k-PVMCS) le problème de Via Minimization contraint sur k couches quand les réseaux en superposition sont autorisés.

On associe à une instance du k-PVMCS les graphes G et G^k introduits dans la section 3.2.1 (où, comme nous venons de le signaler, les croisements sont remplacés par croisements étendus). Nous associons également aux sommets de G^k le vecteur poids $\bar{c}$ comme il a été défini en section 3.2.1. Notons que les graphes G et G^k ont les mêmes propriétés que dans les sections précédentes à part le fait qu'ici un sommet-branche d'une zone de superposition peut avoir plus d'un sommet-branche comme voisin. En fait, un ensemble de l branches qui sont deux à deux en superposition correspond dans G à un sous-graphe complet de l sommets-branche.

Par conséquent, le k-PVMCS quand les vias empilés sont interdits peut être formulé comme la recherche d'un sous-graphe k-parti induit comme montré dans la section 3.2.2. On peut noter que les vias empilés n'existent pas dans le cas 2 couches et qu'ainsi, le 2-PVMC avec superposition reste équivalent au PSBI.

3.3.4 Vias empilés

On suppose à présent que les vias empilés sont autorisés. Cette hypothèse demande un traitement spécifique.

Considérons par exemple l'instance du 3-PVMCS donnée par la figure 3.9 a). Cette instance possède quatre réseaux dont deux sont en superposition. Une affectation valide sur trois couches pour cette instance nécessite au moins un via. La figure 3.9 b) donne une solution optimale avec exactement un via. On peut remarquer que ce via est un via empilé, percé sur la jonction j_0, au travers de seulement deux couches.

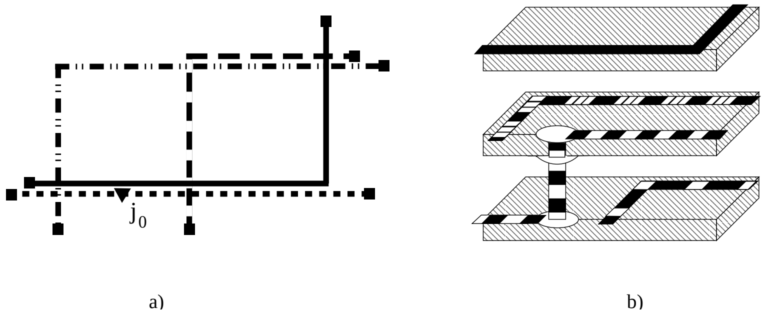


FIG. 3.9 – *a) Une instance et b) une solution avec un via empilé*

On peut remarquer que si une jonction centrale est empilée sur $k-1$ branches, alors cette jonction ne peut pas porter de via. Notons également qu'une instance du k-PVMCS peut avoir une affectation valide qui n'est pas câblable sur k couches, voir section 2.2.4. Par exemple, considérons l'instance de la figure 3.10 a) formée par deux réseaux qui se superposent. La figure 3.10 b) montre une affectation valide sur trois couches A, B, C qui n'est pas câblable. Dans cette solution, le réseau dessiné en pointillés a été affecté à la couche B, et le réseau donné en traits pleins est divisé en deux branches, affectées respectivement aux couches A et B. Comme indiqué sur la figure 3.10 b), le câblage de ces deux branches implique qu'un via est percé au travers des trois couches. Ce percement conduit à une connection interdite entre deux réseaux.

Quand les vias empilés sont autorisés, le k-PVMCS se ramène en fait à rechercher une affectation des réseaux aux couches qui soit optimale, valide et câblable. Comme nous allons le voir, ceci revient à résoudre le k-PSPI avec des contraintes additionnelles.

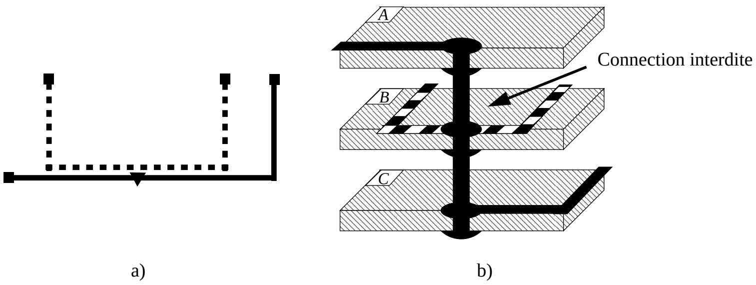


FIG. 3.10 – *Une affectation valide des réseaux mais non câblable*

Considérons une instance du k-PVMCS et nommons $\{1, ..., k\}$ ses couches. Dans ce qui suit, on note par $[l, l']$ l'ensemble $\{l, l+1, ..., l'-1, l'\}$ pour $1 \leq l \leq l' \leq k$.

Si une jonction centrale j portant un via atteint la couche s, pour tout $s \in [l, l']$, alors l'ensemble $[l, l']$ est appelé le *champ* de cette jonction centrale ou même, par extension, le champ du via. Si une jonction centrale ne porte pas de via, son champ est alors limité à une couche. Nous avons le résultat suivant.

Lemme 3.3.1 *Le k-PVMCS, quand les vias empilés sont autorisés, consiste à rechercher une affectation valide optimale telle que, pour toute jonction centrale j d'une zone de superposition, il existe un champ associé $[l_j, l'_j]$, $1 \leq l_j \leq l'_j \leq k$, tel que*

i) une branche b incidente à j est affectée à une couche $l \in [l_j, l'_j]$, et

ii) une branche b sur laquelle j est empilée doit être affectée à une couche $l \in \{1, ..., k\} \setminus [l_j, l'_j]$.

Les conditions i) et ii) assurent que l'affectation valide est câblable. Soit B_0 (resp. J_0) l'ensemble des sommets-branche (resp. sommets-jonction) correspondant aux branches (resp. jonctions centrales) contenues dans une zone de superposition. Notons que B_0 et J_0 sont à la fois des sous-ensembles de sommets de G et de G^k. Pour un sommet-jonction

$v \in J_0$, associé à une jonction centrale j, nous posons $B_v \subset B_0$ comme l'ensemble des sommets-branche correspondant aux branches sur lesquelles j est empilée. Le corollaire suivant est une conséquence du théorème 3.2.5 et du lemme 3.3.1.

Corollaire 3.3.2 *Le k-PVMCS, quand les vias empilés sont autorisés, se ramène à rechercher un sous-graphe k-parti induit $\widetilde{G}^k = (\widetilde{V}^k, E^k(\widetilde{V}^k))$ de G^k avec une k-partition $(\widetilde{V}_1, ..., \widetilde{V}_k)$ satisfaisant les propositions suivantes:*
pour tout sommet-jonction v de J_0, il existe l_v et l'_v avec $1 \leq l_v \leq l'_v \leq k$, tels que

i) si un sommet-branche $u \in B_0$ est adjacent à v dans G, alors il existe $i \in [l_j, l'_j]$ où $u \in \widetilde{V}_i$,

ii) si un sommet-branche u appartient à B_v, alors il existe $i \in \{1, ..., k\} \setminus [l_v, l'_v]$, où $u \notin \widetilde{V}_i$,

iii) si $v \in \widetilde{V}^k$, alors $l_v = l'_v$.

Dans le reste de cette section, nous proposons une formulation en nombres entiers pour le problème du k-PVMCS quand les vias empilés sont autorisés.

On considère une instance du k-PVMCS où les vias empilés sont autorisés. Soit le graphe de modélisation $G^k = (V^k, E^k)$ correspondant à cette instance et le vecteur poids c, comme définis dans la section précédente. On associe à chaque sommet de V^k un vecteur $x_u = (x_u^1, ..., x_u^k)$ de $\{0,1\}^k$. On associe, de plus, à tout sommet-jonction $v \in J_0$ un vecteur $y_v = (y_v^1, ..., y_v^k)$ de $\{0,1\}^k$ tel que $y_v^l = 1$ si et seulement si l appartient au champ de la jonction centrale correspondant à v. Considérons le programme en nombres entiers (Q_k) suivant.

$$\text{Max} \sum_{u \in V^k} \sum_{l=1}^{k} c(u) x_u^l$$

$$\sum_{l=1}^{k} x_u^l \leq 1, \qquad \text{pour tout } u \in V^k, \tag{3.1}$$

$$x_u^l + x_v^l \leq 1, \qquad \text{pour tout } e = uv \in E^k \text{ et } 1 \leq l \leq k, \tag{3.2}$$

$$y_v^{l'} + y_v^{l''} - y_v^l \leq 1, \qquad \text{pour tout } 1 \leq l' < l < l'' \leq k \text{ et pour tout } v \in J_0, \tag{3.3}$$

$$x_u^l \leq y_v^l,\ \forall\, 1 \leq l \leq k, \quad \text{pour tout } u \in B_0 \text{ adjacent à } v \text{ dans } G \text{ et pour tout } v \in J_0, \tag{3.4}$$

$$
\begin{aligned}
& x_u^l + y_v^l \leq 1,\ \forall\ u \in B_v, && \text{pour tout } 1 \leq l \leq k \text{ et pour tout } v \in J_0, && (3.5)\\
& \sum_{l=1}^{k} y_v^l \leq (k-1)(1-\sum_{l=1}^{k} x_v^l)+1, && \text{pour tout } v \in J_0, && (3.6)\\
& x_v^l \leq y_v^l, && \text{pour tout } 1 \leq l \leq k \text{ et pour tout } v \in J_0 && (3.7)\\
& y_v^l \in \{0,1\}, && \text{pour tout } 1 \leq l \leq k \text{ et pour tout } v \in J_0, && (3.8)\\
& x_u^l \in \{0,1\}, && \text{pour tout } 1 \leq l \leq k \text{ et } u \in V^k. && (3.9)
\end{aligned}
$$

Le théorème suivant donne une formulation en nombres entiers pour le k-PVMCS, quand les vias empilés sont autorisés.

Théorème 3.3.3 *Le k-PVMCS, quand les vias empilés sont autorisés, est équivalent au programme en nombres entiers (Q_k).*

Preuve: Considérons tout d'abord une solution (x, y) du programme (Q_k). Posons $W_l = \{u \in V^k \mid x_u^l = 1\}$, pour tout $l \in \{1, ..., k\}$. Soit le graphe $\widetilde{G}^k$ induit par les sommets de $\cup_{l=1}^k W_l$. Comme x vérifie les contraintes de type (7.1), un sommet u de $\widetilde{G}^k$ appartient à au plus un élement parmi $W_1, ..., W_k$. De plus, comme x vérifie les contraintes (7.2), deux sommets adjacents de $\widetilde{G}^k$ ne peuvent appartenir au même élément de $W_1, ..., W_k$. Ainsi $(W_1, ..., W_k)$ constitue une k-partition de $\widetilde{G}^k$. Pour prouver le résultat, il suffit donc de montrer que le sous-graphe k-parti $\widetilde{G}^k$ verifie bien les propriétés du théorème 3.3.2.
Soit v un sommet-jonction de J_0. Posons $\Lambda_v = [l_v, l'_v]$. Comme y vérifie les inégalités (3.3), il existe deux entiers l_v et l'_v, $1 \leq l_v \leq l'_v \leq k$, tels que $\Lambda_v = [l_v, l'_v]$. De plus, comme y respecte les inégalités (3.4), si un sommet-branche u adjacent à v est affecté à un couche i, $i \in \{1, ..., k\}$, alors $i \in \Lambda_v$. Ce qui correspond à la propriété i) du théorème 3.3.2. D'une manière similaire on prouve que, comme y vérifie les inégalités (3.5) (resp. (3.6) et (3.7)), alors la propriété ii) (resp. iii)) du théorème 3.3.2 est vérifiée pour v.
Comme x vérifie les inégalités (7.1) et maximise la fonction objectif de (Q_k), alors $\widetilde{G}^k$ est bien un sous-graphe k-parti induit de poids maximum de G^k qui vérifie les propriétés du théorème 3.3.2. Réciproquement, il est facile de constater que tout sous-graphe k-parti induit de G^k, satisfaisant ces propriétés, est une solution de (Q_k). □

3.4 Conclusion

Dans ce chapitre, nous avons proposé un modèle exact pour le problème de Via Minimization contraint quelque soit le nombre k de couches et le degré maximum des jonctions. Nous avons montré que le problème peut se ramener au problème du sous-graphe k-parti induit de poids maximum. A notre connaissance, il s'agit du premier modèle exact pour ce problème.

Afin de développer un outil de résolution exacte pour le problème de Via Minimization contraint, nous nous sommes intéressés au problème du sous-graphe k-parti induit. Ainsi nous présentons une étude polyédrale pour le problème du sous-graphe 2-parti (biparti) induit dans les deux chapitres suivants. Cette étude nous a permis de proposer, dans le chapitre 6, un algorithme de coupes et branchements efficace pour résoudre le problème du 2-PVMC. Pour $k \geq 3$, nous proposons, dans le chapitre 7, un algorithme de génération de colonnes et branchements pour le problème du sous-graphe k-parti induit et nous l'utiliserons pour résoudre des instances multicouches du k-PVMC.

Chapitre 4

Facettes du polytope des sous-graphes bipartis induits

Dans ce chapitre, nous étudions une approche polyédrale du problème du sous-graphe biparti induit. Nous présentons tout d'abord plusieures familles de contraintes valides pour le polytope des sous-graphes bipartis induits. Nous donnons des conditions nécessaires et suffisantes pour que certaines d'entre elles de ces contraintes définissent des facettes du polytope. Nous discutons également du problème de séparation pour les contraintes, et nous montrons que, pour certaines d'entre elles, le problème peut être résolu en temps polynomial. Enfin, nous donnons des méthodes permettant de construire "des facettes à partir de facettes".

4.1 Introduction

Soit $G = (V, E)$ un graphe. L'enveloppe convexe $P_B(G)$ des vecteurs d'incidence des ensembles de sommets des sous-graphes bipartis induits de G, i.e.

$$P_B(G) = conv\{x^W \in I\!R^V \mid W \subset V, (W, E(W)) \text{ est biparti}\},$$

est appelé le *polytope des sous-graphes bipartis induits de* G. Soit

$$\beta(G) = \{W \subset V, (W, E(W)) \text{ est biparti}\},$$

la famille des ensembles de sommets des sous-graphes bipartis de G.

Le polytope $P_B(G)$ est de pleine dimension. En effet, les ensembles de sommets $\emptyset$, $\{v\}, v \in V$ induisent $|V|+1$ sous-graphes bipartis de G.

Si $B \subset V$ est un sous-ensemble de sommets induisant un sous-graphe biparti de G, alors le vecteur d'incidence de B, x^B, vérifie les inégalités

$$\begin{aligned} 0 \leq x(v) \leq 1, &\quad \text{pour tout } v \in V, && (4.1)\\ x(W) \leq |W|-1, &\quad \text{pour tout } W \subset V \text{ tel que } (W, E(W)) && (4.2)\\ &\quad \text{est un cycle impair.} \end{aligned}$$

De plus, toute solution en $0-1$ des inégalités (4.1), (4.2) est un vecteur d'incidence d'un sous-graphe biparti induit de G. Les contraintes (4.1) sont dites *inégalités triviales* et les contraintes (4.2) sont appelées *inégalités de cycles impairs*. Dans [14], il est montré que les inégalités (4.1) définissent des facettes de $P_B(G)$. Dans la section suivante, nous discutons d'autres contraintes ainsi que de conditions nécessaires et suffisantes pour qu'elles définissent des facettes

Soit $x^* \in I\!R^V$ un vecteur-poids associé aux sommets de V. Dans ce qui suit, on note $\bar{x}$ le vecteur $1\!\!1 - x^*$, où $1\!\!1 = (1, ..., 1)^T$. Aussi on note par G_{x^*} le graphe support de x^*, c'est-à-dire le graphe induit par les sommets u tels que $x^*(u) \neq 0$. Les algorithmes de séparation qui seront considérés dans ce chapitre seront étudiés par rapport à x^*.

4.2 Contraintes de cycles impairs

Soient $G = (V, E)$ un graphe et $H = (W, F)$ un cycle impair induit de G. Considérons l'inégalité de cycle impair

$$x(W) \leq |W| - 1 \tag{4.3}$$

associée à H. Dans [14], des conditions nécessaires et suffisantes sont données pour que cette inégalité définisse une facette de $P_B(G)$.

Théorème 4.2.1 [14] *L'inégalité (4.3) définit une facette de $P_B(G)$ si et seulement si*

i) *(W, T) est sans corde,*

ii) *pour tout $v \in V \setminus W$, il existe un sommet $w \in W$ tel que le graphe induit par $W \setminus \{w\} \cup \{v\}$ ne contient pas de cycle impair.*

La séparation des inégalités de cycles impairs (4.2) dans un graphe $G = (V, E)$ peut être effectuée en $O(|V|^3)$ [74]. Ces inégalités définissent des facettes pour $P_B(G)$ seulement si les cycles sont des trous impairs [14]. En modifiant légérement l'algorithme cité dans [74], la séparation des inégalités de trous impairs peut être également réalisée en $O(|V|^3)$. Cet algorithme peut se décrire de la façon suivante.

L'idée principale de l'algorithme développé dans [74] consiste à considérer un nouveau graphe $G'_{x^*} = (V', E')$ avec deux sommets u' et u'' pour tout sommet u de G_{x^*}. Pour toute arête uv de G_{x^*}, on définit dans E' les arêtes $u'v''$ et $u''v'$ munies du poids $\frac{\bar{x}(u)+\bar{x}(v)}{2}$. Il est clair que le graphe G'_{x^*} est biparti.

Maintenant, remarquons que la chaîne élémentaire $P'_{u'v''}$ de G'_{x^*} allant d'un sommet u' à un sommet v'' correspond à une chaîne impaire P_{uv} dans G_{x^*} entre u et v. Pour tout sommet u de G_{x^*}, on calcule une plus courte chaîne $P'_{u'u''}$ dans G' entre u' et u''. La plus courte chaîne P_{uu} correspond à un cycle impair C_u de G contenant u.

Soit C_{u_0} tel que $\bar{x}(C_{u_0}) = min_{u \in V'} \bar{x}(C_u)$. Si $\bar{x}(C_{u_0}) \geq 1$, alors $x^*(C) \leq |C| - 1$ pour tout cycle impair, et ainsi les inégalités (4.2) sont toutes satisfaites par x^*. Autrement, la contrainte $x(C_{u_0}) \leq |C_{u_0}| - 1$ est violée par x^*.

De plus, si la chaîne $P'_{u'_0 u''_0}$ correspondant à C_{u_0} est choisie comme étant une chaîne de plus petite cardinalité parmi les plus courtes chaînes de G' entre u'_0 et u''_0, alors le cycle C_{u_0} est un trou impair.

La recherche d'une plus courte chaîne peut être réalisée par l'algorithme de Dijkstra. Dans le but d'accélérer cette recherche, on utilise l'astuce bien connue qui consiste à débuter la recherche de l'arbre des plus courtes chaînes à partir des deux extrémités. Grace aux symétries du graphe G'_{x^*}, les deux arbres des plus courtes chaînes de racine u' et u'' sont identiques et le calcul de l'un des deux suffit.

4.3 Contraintes de cliques

Soit $G = (V, E)$ un graphe. Un ensemble de sommets $W \subset V$ est appelé *clique* si toute paire de sommets de W est jointe par une arête. Une clique est appelé *maximale* si elle n'est pas strictement contenue dans une clique. Si $W \subset V$ est une clique alors un sous-graphe biparti induit de G ne peut contenir plus de deux sommets, ce qui implique que l'inégalité

$$x(W) \leq 2 \tag{4.4}$$

est valide pour $P_B(G)$. Dans [14], les auteurs donnent des conditions nécessaires pour que cette inégalité définissent des facettes de $P_B(G)$.

Théorème 4.3.1 [14] *L'inégalité (4.4) définit une facette de $P_B(G)$ si et seulement si W est une clique maximale et $|W| \geq 3$.*

Nous proposons une heuristique de séparation pour les contraintes de cliques dans le cadre d'un algorithme de coupes et branchements présenté au chapitre 6.

4.4 Contraintes d'antiwebs

4.4.1 Inégalités d'antiwebs

Dans [102], Trotter introduit les deux classes de graphes suivantes et montre qu'elles induisent des inégalités valides pour le polytope du stable.

Etant donnés deux entiers $n \geq 2$ et t, $1 \leq t \leq \frac{n}{2}$, on définit le graphe dont l'ensemble des sommets est $V_n = \{u_1, ..., u_n\}$ et l'ensemble d'arêtes

$$\{(u_i, u_j) \mid j = i + t, ..., i + n - t \text{ pour } i \in \{1, ..., n\}\}.$$

De tels graphes sont appelés (n, t)-*webs* et sont notés $\mathcal{W}(n, t)$.

On appelle *complémentaire d'un graphe* $G = (V, E)$ le graphe $\bar{G} = (V, \bar{E})$ où $\bar{E} = \{(u, v) \mid u, v \in V, u \neq v, (u, v) \notin E\}$. On appelle *antiweb* le complémentaire d'un web $\mathcal{W}(n, t)$, où $n \geq 2$ et $t \in \{1, ..., \frac{n}{2}\}$. Noter qu'un antiweb n'est pas nécessairement un web. Trotter [102] montre qu'un antiweb est un web si et seulement si le web dont il est le complémentaire est un trou impair où un anti-trou impair (c'est-à-dire le complémentaire d'un trou impair).

Dans cette section, on considère la classe des antiwebs introduites dans [102] élargie à la classe des cliques. On appelera également ces derniers graphes antiwebs. Etant donnés $n \geq 2$ et t, $1 \leq t \leq \frac{n}{2} + 1$, on appelle (n, t)-*antiweb* le graphe $\overline{\mathcal{W}}(n, t) = (V_n, E_n)$ où $V_n = \{u_1, ..., u_n\}$ et $E_n = \{(u_i, u_j) \mid j = i + 1, ..., i + t - 1 \text{ pour } i \in \{1, ..., n\}\}$.

La classe des (n, t)-antiwebs, pour un $t \geq 1$ donné, est nommée t-*antiwebs*. Par exemple, les 2-antiwebs sont les cycles. De plus, lorsque $t = \left\lfloor \frac{n}{2} \right\rfloor + 1$, un (n, t)-antiweb est la clique K_n sur n sommets. La figure 4.1 donne un (8,3)-antiweb.

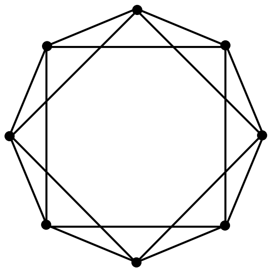


FIG. 4.1 – *Un (8,3)-antiweb*

Soit $G=(V,E)$ un graphe et soit $H=(W,F)$ un (n,t)-antiweb induit dans G. On associé à $H=(W,F)$ l'inégalité

$$\sum_{i=1}^{n} x(u_i) \leq \left\lfloor \frac{2n}{t} \right\rfloor \tag{4.5}$$

Les inégalités de type (4.5) sont nommées *inégalités d'antiwebs*. On a le résultat suivant.

Théorème 4.4.1 *L'inégalité (4.5) est valide pour $P_B(G)$.*

Preuve. Remarquons tout d'abord que l'ensemble de sommets $\{u_i, u_{i+1}, ..., u_{i+t-1}\}$ est une clique de taille t, pour tout $i \in \{1, ..., n\}$. Donc les inégalités de type (4.4)

$$\sum_{j=i}^{i+t-1} x(u_j) \leq 2$$

sont valides pour $P_B(G)$, pour tout $i = 1, ..., n$. Comme tout sommet u_i, $i \in \{1, ..., n\}$ est contenu dans t de ces cliques, en sommant les n inégalités associées et en arrondissant le second membre à l'entier inférieur, on obtient l'inégalité (4.5). □

Dans le cas où $t \neq n$ et t divise $2n$, alors l'inégalité (4.5) ne produit pas de facette pour $P_B(G)$. En revanche, lorsque $n = t$, l'inégalité (4.5) associée à un (n,t)-antiweb $\overline{\mathcal{W}}(n,t)$ est une inégalité de clique (4.4), par conséquent elle définit une facette de $P_B(\overline{\mathcal{W}}(n,t))$. Dans la suite, nous nous intéressons aux cas où t ne divise pas $2n$.

Un (n,t)-antiweb peut être vu comme un cas particulier de la structure appelée (n,t,q)-antiweb généralisé introduite par Laurent dans [68]. Laurent décrit le système d'indépendants suivant. Soient n, t, q trois entiers tels que $n \geq t \geq q \geq 2$. Soient

$U = \{u_1, ..., u_n\}$ et $T_i = \{u_i, u_{i+1}, ..., u_{i+t-1}\}$, pour tout $i \in \{1, ..., n\}$, (les indices sont pris modulo n). On note $\mathcal{AW}(n,t,q)$ la famille de sous-ensembles de U définie par

$$\mathcal{AW}(n,t,q) = \{C \subseteq U \mid |C| = q \text{ et } C \subseteq T_i \text{ pour un } i \in \{1, ..., n\}\}.$$

$(U, \mathcal{AW}(n,t,q))$ est un système d'indépendants appelé (n,t,q)-*antiweb généralisé.* Laurent [67] montre le résultat suivant.

Théorème 4.4.2 [68] *L'inégalité*

$$\sum_{u \in U} x(u) \geq \left\lfloor \frac{n(q-1)}{t} \right\rfloor \tag{4.6}$$

est valide pour le polytope $P_{\mathcal{I}}(U, \mathcal{AW}(n,t,q))$. De plus, elle définit une facette si et seulement si soit $n = t$, soit t ne divise pas $n(q-1)$.

Soit $\overline{\mathcal{W}}(n,t)$ un (n,t)-antiweb avec $t \geq 3$ et considérons l'ensemble $\mathcal{T}$ des triangles de $\overline{\mathcal{W}}(n,t)$. Si l'on identifie l'ensemble V des sommets de l'antiweb à l'ensemble U, les ensembles T_i, $i = 1, ..., n$, correspondent aux n cliques de taille t de l'antiweb $\overline{\mathcal{W}}(n,t)$. De même, l'ensemble $\mathcal{AW}(n,t,3)$ n'est rien d'autre que l'ensemble des triangles de $\overline{\mathcal{W}}(n,t)$. D'autre part, $(V, \mathcal{T})$ est un système d'indépendants. Ainsi les polytopes $P_B(\overline{\mathcal{W}}(n,t))$ et $P_{\mathcal{I}}((U, \mathcal{AW}(n,t,3))$ coïncident. Par conséquent, on peut déduire du théorème 4.4.2 le corollaire suivant.

Corollaire 4.4.3 *L'inégalité (4.5) définit une facette de $P_B(H)$ si et seulement si soit $n = t$, soit t ne divise pas $2n$.*

4.4.2 Séparation

Dans [27], Cheng et De Vries montrent que, pour t fixé, la classe des (n,t)-antiwebs peut être séparée en temps polynomial. Ce résultat peut être facilement étendu au polytope des sous-graphes bipartis induits. Pour ce faire, donnons d'abord le résultat suivant qui permet de construire de nouvelles facettes par identification de paires de sommets.

Théorème 4.4.4 (Identification de deux sommets non adjacents)
Soit $G = (V, E)$ un graphe et $G' = (V', E')$ le graphe obtenu à partir de G en identifiant

deux sommets non-adjacents v_1 et v_2 de G. Soit v_0 le sommet de G' résultant de cette identification. Si $ax \leq b$ est une inégalité valide pour $P_B(G)$, alors

$$(a(v_1) + a(v_2))x(v_0) + \sum_{u \in V' \setminus \{v_0\}} a(u)x(u) \leq b \tag{4.7}$$

est une inégalité valide pour $P_B(G')$.

Preuve. Soit $B' \subset V'$ un ensemble de sommets induisant un sous-graphe biparti H' de G'. Supposons d'abord que $v_0 \notin B'$. Alors B' induit dans G un sous-graphe biparti de G qui ne contient ni v_1 ni v_2. De plus, comme $(a(v_1) + a(v_2))x^{B'}(v_0) + \sum_{u \in V' \setminus \{v_0\}} a(u)x^{B'}(u) = \sum_{u \in V} a(u)x^{B'}(u) \leq b$, l'inégalité (4.7) est vérifiée. Supposons maintenant que $v_0 \in B'$ et posons $B = (B' \setminus \{v_0\}) \cup \{v_1, v_2\}$. L'ensemble B induit un sous-graphe biparti de G. En effet, supposons le contraire et considérons un cycle impair C de B. Si C contient l'un des sommets v_1 ou v_2, disons v_1, alors $(V(C) \setminus \{v_1\}) \cup \{v_0\}$ induit un cycle impair de H', une contradiction. Si C contient v_1 et v_2, comme v_1 et v_2 sont non-adjacents dans G, il existe dans H' deux cycles C' et C'' ayant en commun le sommet v_0. De plus, comme C est impair, un des cycles C' ou C'' est impair, ce qui est une contradiction. Par conséquent, B induit un sous-graphe biparti de G. Comme $\sum_{u \in V} a(u)x^{B}(u) = a(v_1) + a(v_2) + \sum_{u \in V' \setminus \{v_0\}} a(u)x^{B'}(u) \leq b$, l'inégalité (4.7) est vérifiée. $\square$

Un (n,t)-*antiweb généralisé* est un graphe obtenu à partir d'un (n,t)-antiweb par identifications de sommets non adjacents. Par le théorème 4.4.4, il en résulte qu'un (n,t)-antiweb généralisé induit une contraintes valide pour le polytope $P_B(G)$. Celle-ci peut être obtenue par application de la procédure du théorème 4.4.4 à partir de la contrainte d'antiweb correspondante. On appellera ces contraintes *inégalités d'antiwebs généralisés*. En adaptant la preuve de Cheng et De Vries au polytope des sous-graphes bipartis induits, on obtient le résultat suivant.

Théorème 4.4.5 *Soit $t \geq 2$ un entier fixé. Alors la classe des inégalités de (n,t)-antiwebs peut être séparée en temps polynomial.*

4.5 Contraintes de p-roues d'ordre q

4.5.1 Inégalités de p-roues d'ordre q

Un graphe $G = (V, E)$ est appelé *p-roue d'ordre q* où p et q sont des entiers positifs, si $V = V_h \cup V_c$ où V_h induit un trou de longueur p et V_c induit une clique sur q sommets universels à V_h (i.e. tout sommet de la clique est adjacent à tout sommet du trou) (voir figure 4.2 pour une 5-roue d'ordre 3).

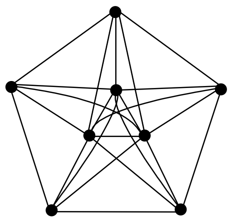


FIG. 4.2 – *Une 5-roue d'ordre 3*

Théorème 4.5.1 *Soit $G = (V, E)$ un graphe et $H = (W_h \cup W_c, F)$ une p-roue d'ordre q induite dans G. Alors l'inégalité*

$$x(W_h) + \left\lfloor \frac{p}{2} \right\rfloor x(W_c) \leq 2 \left\lfloor \frac{p}{2} \right\rfloor \tag{4.8}$$

est valide pour $P_B(G)$. De plus, (4.8) définit une facette de $P_B(G)$ si et seulement si

i) p est impair,

ii) pour tout sommet u de $V \setminus (W_h \cup W_c)$ où $\{u\} \cup W_c$ est une clique, au moins une des conditions suivantes est vérifiée:

a) Il existe un ensemble $S \subset W_h$ de $\lfloor \frac{p}{2} \rfloor$ sommets tels que $S \cup \{u\}$ est un stable dans G.

b) Il existe un sommet v de W_h tel que le graphe induit par $(W_h \setminus \{v\}) \cup \{u\}$ est biparti.

Preuve. Nous montrons tout d'abord que l'inégalité (4.8) est valide pour $P_B(G)$. Pour cela, considérons un ensemble $B \in \beta(G)$. Comme W_c est une clique, B contient au plus deux sommets de W_c.
- Si $|B \cap W_c| = 2$, alors $B \cap W_h = \emptyset$ et donc x^B satisfait (4.8).
- Si $|B \cap W_c| = 1$, alors $B \cap W_h$ est un stable de W_h, et ainsi $|B \cap W_h| \leq \lfloor \frac{p}{2} \rfloor$. Par conséquent x^B satisfait (4.8).
- Si $B \cap W_c = \emptyset$, alors $B \cap W \subset W_h$. Si p est impair alors, par l'inégalité (4.2), on obtient $|B \cap W_h| \leq p - 1$. Et si p est pair, alors, $|B \cap W_h| \leq p$.
Dans les deux cas, on a$|B \cap W_h| \leq 2 \lfloor \frac{p}{2} \rfloor$. Par conséquent, (4.8) est valide pour $P_B(G)$.
Supposons que $p = 2k$ pour un $k \geq 2$. Soit $W_h = \{u_1, ..., u_{2k}\}$ où $u_i u_{i+1} \in E$ pour $i = 1, ..., p$ (les indices sont pris modulo p). Considérons les inegalités

$$x(W_c) + x_{u_i} + x_{u_{i+1}} \leq 2, \text{ pour } i = 1, 3, ..., 2k - 1,$$

qui sont de type (4.4), et donc valide pour $P_B(G)$. En sommant ces inégalités, on obtient la contrainte (4.8). Ce qui implique que (4.8) ne peut définir une facette.
Pour la suite, supposons p impair. Notons que, dans ce cas, l'inégalité (4.8) peut être écrite sous la forme

$$x(W_h) + \frac{p-1}{2}x(W_c) \leq p - 1.$$

Soit $u \in V \setminus (W_h \cup W_c)$ tel que $W_c \cup \{u\}$ est une clique. Supposons que ni a) ni b) du ii) ne sont vérifiées. Soit $B \in \beta(G)$ tel que x^B satisfait (4.8) à l'égalité. On affirme que $u \notin B$. En fait, si ce n'est pas le cas, alors $|B \cap W_c| \leq 1$, autrement dit le graphe induit par B contiendrait un cycle impair.
- Si $B \cap W_c = \emptyset$, alors B doit contenir $p - 1$ sommets de W_h. Cependant, comme par hypothèse, la condition ii) b) n'est pas vérifiée, ceci implique que le graphe induit par B n'est pas biparti, une contradiction.
- Si $|B \cap W_c| = 1$, alors $B \cap W_h$ est un stable de $\frac{p-1}{2}$ sommets. Comme ii) a) n'est pas vérifié, il existe un sommet $v \in B \cap W_h$, qui est adjacent à u. Alors $\{u, v, w\}$ induit un cycle impair, où w est le sommet de $B \cap W_c$. Comme $u, v, w \in B$, c'est à nouveau une contradiction.
Ainsi u n'appartient à aucun ensemble $B \in \beta(G)$ dont le vecteur d'incidence satisfait (4.8) à l'égalité. Mais ceci implique que l'inégalité (4.8) est équivalente à la contrainte $x(u) \geq 0$, une contradiction.

Maintenant, supposons que i) et ii) sont vérifiées et soit $p = 2k + 1$, $k \geq 1$. Soit $S \subset W_h$ un stable de k sommets et soit $v_0 \in W_c$. Posons $V_1 \subset V \setminus (W_h \cup W_c)$ l'ensemble des sommets u tels que $\{u\} \cup W_c$ soit une clique. Soit $V_2 = V \setminus (W_h \cup W_c \cup V_1)$. Soit $V_1' \subset V_1$ un sous-ensemble de sommets de V_1' tel que pour tout $u \in V_1'$ il existe un ensemble S_u de W_h de taille k où $S_u \cup \{u\}$ est un stable de G. Par ii) b), pour tout $u \in V_1 \setminus V_1'$, il existe un sommet, disons w_u, de W_h tel que le graphe induit par $(W_h \setminus \{w_u\}) \cup \{u\}$ soit biparti. Aussi si $u \in V_2$ alors il existe un sommet, disons v_u, de W_c qui n'est pas adjacent à u. Posons

$$B_u = \begin{cases} W_h \setminus \{u\} & \text{si } u \in W_h, \\ S \cup \{u\} & \text{si } u \in W_c, \\ S_u \cup \{u, v_0\} & \text{si } u \in V_1', \\ (W_h \setminus \{w_u\}) \cup \{u\} & \text{si } u \in V_1 \setminus V_1', \\ S \cup \{u, v_u\} & \text{si } u \in V_2. \end{cases}$$

Pour tout $u \in V$, $B_u \in \beta(G)$ et x^{B_u} satisfait (4.8) à l'égalité. De plus, il n'est pas difficile de voir que les vecteurs x^{B_u}, $u \in V$, sont linéairement indépendants. □

Les inégalités (4.8) sont appelées *inégalités de p-roues d'ordre q*.

4.5.2 Séparation

Dans ce qui suit, nous allons montrer que les contraintes de p-roues d'ordre q où q est fixé, peuvent être séparées en temps polynomial.

Théorème 4.5.2 *Si q est fixé, les inégalités de p-roues d'ordre q, avec p impair, peuvent être séparées en temps polynomial.*

Preuve. Puisque q est fixé, on peut tester en temps polynomial les inégalités de la forme
$$x(K) + x_u + x_v \leq 2 \tag{4.9}$$
où K est un graphe complet sur q sommets et $u, v \in V$ deux sommets adjacents et universels à K. On peut donc supposer que les inégalités (4.9) sont satisfaites par x^*. Soit $\bar{G} = (\bar{V}, \bar{E})$ le graphe où $\bar{V} = \{u \in V \mid u \text{ est universel à } K\}$ et $\bar{E} = \{uv \in E \mid u, v \in \bar{V}\}$. Soit $\lambda = x^*(K)$. A chaque arête $uv \in \bar{E}$, on associe le poids $w^*_{uv} = (2-\lambda) - (x^*_u + x^*_v)$. Notons que, puisque les contraintes (4.9) sont supposées satisfaites par x^*, les poids sur les arêtes dans $\bar{G}$ sont tous non négatifs. Par conséquent, il existe une inégalité du type (4.8), correspondant à K, qui est violée par x^* si et seulement si le poids minimum d'un cycle impair dans $\bar{G}$, par rapport au vecteur poids x^*, est strictement inférieur à $(2-\lambda)$. Un cycle impair C dans $\bar{G}$, avec $|C| = 2k+1$, a pour poids
$$w^*(C) = (2k+1)(2-\lambda) - 2x^*(C) = 2(2k - k\lambda - x^*(C)) + (2-\lambda).$$
Ainsi, $w^*(C) < 2 - \lambda$ si et seulement si $x^*(C) + k\lambda < 2k$, c'est-à-dire si l'inégalité (4.8) est violée. Comme rechercher un cycle impair de poids minimum dans un graphe avec des poids non négatifs peut être fait en temps polynomial [74], le théorème est démontré. □

4.6 Contraintes de p-roues liftées

4.6.1 Inégalités de p-roues liftées

Une *p-roue liftée* $H = (W, F)$ est une subdivision d'une p-roue d'ordre 1 H' telle que les cycles obtenus à partir des triangles de H' par la subdivision soient tous impairs. La p-roue liftée H est dite *paire* (resp. *impaire*) si p est pair (resp. impair). Le sommet

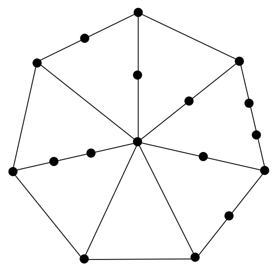


FIG. 4.3 – *Une 7-roue liftée*

unique du graphe complet de H' est appelée le *centre* de H. La figure 4.3 montre une 7-roue liftée.

Si $H = (W, F)$ est une p-roue liftée, on note par C^* la face extérieure de H qui sera appelée *jante* de H. Soient u_0 le centre de H et $u_1, ..., u_p$ les sommets de degré 3 de C^* pris dans le sens de la montre. Pour $i \in \{1, ..., p\}$, la sous-chaîne $P_{i,i+1}$ de C^* liant u_i à u_{i+1} et dont les sommets intérieurs sont tous de degré 2, est appelée une *chaîne-jante* de H (les indices sont pris modulo p). Enfin, on note par $P_{0,i}$, $i \in \{1, ..., p\}$, la chaîne de H entre u_0 et u_i dont les sommets intérieurs sont tous de degré 2. $P_{0,i}$ est appelé le *rayon* de H.

Soit $G = (V, E)$ un graphe contenant une p-roue liftée $H = (W, F)$, avec p impair. On associe à H l'inégalité

$$\sum_{u \in W \setminus \{u_0\}} x(u) + \frac{p-1}{2} x(u_0) \leq |W| - 2. \tag{4.10}$$

Une inégalité du type (4.10) est appelée *inégalité de p-roue liftée*. Dans la suite, nous montrons que ces inégalités définissent des facettes de $P_B(H)$. Mais d'abord nous établissons la validité.

Théorème 4.6.1 *L'inégalité (4.10) est valide pour* $P_B(G)$.

Preuve: Soit $B \in \beta(G)$. Remarquons tout d'abord que, comme $W \setminus \{u_0\}$ est un cycle impair (la jante de H) avec des sommets pendants, alors $|B \cap (W \setminus \{u_0\})| \leq |W \setminus \{u_0\}| - 1$. Par conséquent, si $u_0 \notin B$, x^B satisfait (4.10). Autrement, supposons que $u_0 \in B$. Notons par $C_1, ..., C_p$ les p faces intérieures de H. Remarquons que tout sommet de $W \setminus \{u_0\}$ est contenu dans au plus deux cycles distincts parmi $C_1, ..., C_p$. Comme $|V(C_i) \setminus B| \geq 1$ pour $i = 1, ..., p$, au moins $\frac{p+1}{2}$ sommets sont nécessaires pour casser les cycles impairs $C_1, ..., C_p$. Ainsi $x^B(W \setminus \{u_0\}) \leq |W \setminus \{u_0\}| - \frac{p+1}{2}$, c'est-à-dire, l'inégalité

(4.10) est satisfaite par x^B. □

Théorème 4.6.2 *L'inégalité (4.10) définit une facette de $P_B(H)$.*

Preuve: Notons l'inégalité (4.10) par $ax \leq \alpha$ et supposons qu'il existe une inégalité $bx \leq \beta$ définissant une facette de $P_B(H)$ telle que $\{x \in P_B(H) \mid ax = \alpha\} \subseteq \{x \in P_B(H) \mid bx = \beta\}$. Comme, l'inégalité (4.10) est valide pour $P_B(G)$, pour montrer qu'elle définit aussi une facette, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.
Pour cela, montrons tout d'abord qu'il existe $\rho \in I\!R$ tel que $b(u) = \rho$ pour tout sommet u de C^*. Prenons un sommet arbitraire u de $V(P_{i,i+1})$, pour $i \in \{1, ..., p\}$ (les indices sont pris modulo p). Considérons les ensembles de sommets

$W_1 = W \setminus \{u_i, u_{i+2}, ..., u_{i+p-1}\}$,

$W_2 = (B_1 \setminus \{u\}) \cup \{u_i\}$.

Comme les graphes induits par W_1 et W_2 sont des arbres, W_1 et W_2 induisent des graphes bipartis. De plus, $ax^{W_1} = ax^{W_2} = |W|-2 = \alpha$. Il s'ensuit que $0 = bx^{W_1} - bx^{B_2} = b(u) - b(u_i)$. En posant $\rho = b(u_1)$, on obtient $b(u) = \rho$ pour tout sommet $u \in P_{1,2}$. Par conséquent, comme $u_i \in V(P_{i-1,i}) \cap V(P_{i,i+1})$, pour $i \in \{1, ..., k\}$, alors $b(u) = \rho$ pour tout sommet $u \in V(C^*)$. D'une manière similaire, on peut montrer que $b(u) = b(u_i)$, pour $u \in P_{0,i}$, $i \in \{1, ..., k\}$. On a alors $b(u) = \rho$ pour tout sommet de $W \setminus \{u_0\}$.
Considérons l'ensemble $W_3 = W \setminus \{u_0, u_1\}$. Il est clair que $W_3 \in \beta(H)$ et $ax^{W_3} = |W|-2$. Comme $bx^{W_3} = bx^{B_2}$, on obtient alors $b(u_0) = \sum_{i=3,5,...,p} b(u_i) = \frac{p-1}{2}\rho = \rho a(u_0)$. En conséquence, $b = \rho a$.
De plus, comme $ax \leq \alpha$ n'est pas une contrainte triviale, $bx \leq \gamma$ définit une facette non triviale de $P_B(H)$, et alors $b \geq 0$. Par conséquent $\rho > 0$ ce qui achève la preuve de ce théorème. □

4.6.2 Séparation

La classe des p-roues liftées peut être vue comme une généralisation des p-roues d'ordre 1 qui, d'après le théorème 4.5.2, sont séparables en temps polynomial. Nous discutons ici du problème de séparation pour la classe des p-roues liftées.

Dans [26], Cheng et Cunningham introduisent une large classe d'inégalités valides pour le polytope du stable, appelée inégalités de roues généralisées. Ils proposent également un algorithme de séparation polynomial pour ces inégalités. Nous utilisons dans ce paragraphe des idées similaires à celles de Chang et Cunningham pour montrer que les inégalités de p-roues liftées peuvent être séparées en temps polynomial.

Comme pour la séparation des inégalités d'antiweb, on définit une classe d'inégalités valides plus larges et contenant les inégalités de p-roues liftées. On appelle *p-roue liftée généralisée* le graphe obtenu à partir d'une p-roue liftée d'ordre p par identifications de sommets non adjacents. Une inégalité correspondant à une p-roue liftée généralisée, obtenue par application du théorème 4.4.4 sera appelée *inégalité de p-roue liftée généralisée*. Par le théorème 4.4.4, il en résulte que les inégalités de p-roues liftées généralisées sont valides pour $P_B(G)$. Le reste de cette section décrit un algorithme polynomial de séparation pour cette classe de contraintes. Cet algorithme permet de séparer en même temps la classe des p-roues liftées.

L'algorithme décrit ici effectue des recherches de plus courtes chaînes paires et impaires non élémentaires pour déterminer les sous-chaînes élémentaires d'une p-roue liftée. la recherche d'une plus courte chaîne paire ou impaire peut être réalisée en temps polynomial [74]. Alors que la recherche d'une plus courte chaîne élémentaire de parité donnée est un problème NP-complet [40].

Soit $G = (V, E)$ un graphe. Soit $\omega(e) = \frac{\bar{x}(u)+\bar{x}(v)}{2}$, pour toute arête $e = uv \in E$ (Rappelons que $\bar{x} = \mathbb{1} - x^*$). L'algorithme commence par calculer pour tout couple de sommets distincts $u, v \in V$, le poids minimum par rapport à ω, d'une chaîne paire (resp. impaire) entre u et v dans G. Soit $w_e(u, v)$ (resp. $w_o(u, v)$) ce minimum. Ce calcul peut être réalisé en $O(|V|^3)$ en utilisant l'algorithme de Dijkstra dans un graphe biparti approrié [74].

Soit u_0 un sommet donné de G. On construit un graphe complet $G' = (V' = V'_e \cup V'_o, E')$ où V'_e et V'_o sont deux copies de $V \setminus \{u_0\}$. On définit alors le poids $w'(u'v')$ sur G', pour toute arête $e = u'v' \in E'$, comme suit

$$\begin{cases} \frac{1}{2}w_e(u_0, u) + \frac{1}{2}w_e(u_0, v) + w_o(u, v) - \frac{1}{4}\bar{x}(u) - \frac{1}{4}\bar{x}(v) - \frac{1}{2} & \text{si } u \in V'_e \text{ et } v \in V'_e, \\ \frac{1}{2}w_o(u_0, u) + \frac{1}{2}w_o(u_0, v) + w_o(u, v) - \frac{1}{4}\bar{x}(u) - \frac{1}{4}\bar{x}(v) - \frac{1}{2} & \text{si } u \in V'_o \text{ et } v \in V'_o, \\ \frac{1}{2}w_e(u_0, u) + \frac{1}{2}w_o(u_0, v) + w_e(u, v) - \frac{1}{4}\bar{x}(u) - \frac{1}{4}\bar{x}(v) - \frac{1}{2} & \text{si } u \in V'_e \text{ et } v \in V'_o, \end{cases}$$

où u (resp. v) est un sommet de G dont u' (resp. v') est une copie.

En fait, une arête $u'v'$ de G' représente une chaîne-jante d'une roue de centre u_0. De plus, le vecteur-poids w' associé aux arêtes de G' est choisi de façon à ce que la somme des poids des arêtes d'un cycle C dans G' corresponde au poids d'une $|C|$-roue liftée centrée en u_0. D'autre part si x^* satisfait les inégalités triviales et les inégalités de cycles impairs, on peut alors montrer que $w'(e)$ est non négatif pour toute arête $e \in E'$. Par conséquent, la recherche d'une chaîne fermée impaire $\mathcal{W}$ de poids minimum dans G' peut être effectuée en $O(|V|^3)$ [74]. En utilisant des idées similaires à celles développées dans [26], on montre qu'il existe une inégalité de p-roue liftée généralisée dont le graphe support est centré en u_0 et qui soit violée par x^*, si et seulement si la somme des poids des arêtes de $\mathcal{W}$ est strictement inférieure à $1 - \frac{1}{2}x^*(u_0)$.

Si l'on répète cette recherche pour tout sommet $u \in V$, on détecte ainsi s'il existe une inégalité de p-roue liftée généralisée violée par x^* s'il en reste une.

L'algorithme présenté ici consiste à effectuer, pour tout sommet de G, la recherche d'un plus petit cycle dans le graphe G'. Comme la recherche d'un cycle minimum dans G' peut être réalisé en $O(|V|^3)$, on peut établir le résultat suivant.

Théorème 4.6.3 *Le problème de séparation des inégalités de p-roues liftées généralisées peut être résolu en $O(|V|^4)$.*

4.7 Inégalités de chemin-cycles

Un graphe $H = (W, F)$ est appelé un *chemin-cycle d'ordre* p où p est un entier positif, si H consiste en p trous impairs $C_1, ..., C_p$ et une chaîne P^* sans corde tel que

i) $C_i \cap C_{i+1} = P_i$ où P_i est une chaîne sans corde consistant en au moins une arête, pour $i = 1, ..., p-1$,

ii) les chaînes P^*, P_1, ..., P_{p-1} sont deux à deux sommets disjointes,

iii) pour tout $i, j \in \{1, ..., p\}$ tel que $i + 1 < j$, C_i et C_j sont sommet-disjoints,

iv) la chaîne P^* joint un sommet de C_1 , disons w_0, à un sommet de C_p, disons w_p, tels que $w_0 \in V(C_1) \setminus V(P_1)$, $w_p \in V(C_p) \setminus V(P_{p-1})$ et $V(P^*) \setminus \{w_0, w_p\} \cap (\bigcup_{i=1,...,p} V(C_i)) = \emptyset$.

On peut remarquer qu'une chemin-cycle d'ordre p est un graphe planaire avec $p+2$ faces (dont la face extérieure) et un degré maximal des sommets égal à trois. Un chemin-cycle d'ordre 4 est donné par la figure 5.5.

Etant donné un chemin-cycle $H = (W, F)$ d'ordre p, soit $H' = (W', F')$ le sous-graphe de H obtenu en ôtant toutes les arêtes internes des chaînes $P_1, ..., P_{p-1}$. Notons que le graphe H' consiste en trois chaînes arête-disjointes P^*, P', P'' entre w_0 et w_p. On note par C' et C'' les trous de H' formés par les chaînes P' et P^* d'une part, et P'' et P^*, d'autre part.

Soient $w_1, ..., w_{p-1}$ et $z_1, ..., z_{p-1}$ les sommets de P' et P'' qui appartiennent respectivement aux chaînes $P_1, ..., P_{p-1}$, de telle façon que P_i joigne le sommets w_i au sommet z_i. Pour simplifier les notations, on posera $z_0 = w_0$ et $z_p = w_p$ et on notera P_0 et P_p respectivement, les chaînes de w_0 à z_0 et de w_p à z_p. (Ainsi, P_0 et P_p sont vides.)

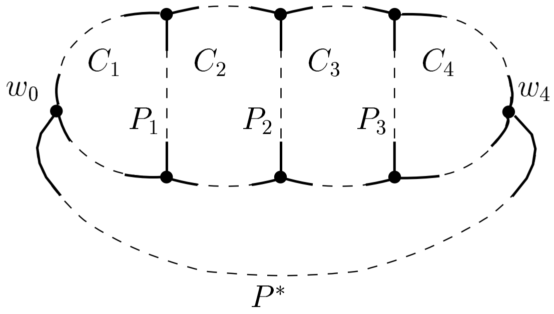


FIG. 4.4 – *Un chemin-cycle d'ordre 4.*

Si u et v sont deux sommets de P' (resp. P'') alors $u \sim v$ représente la sous-chaine de P' (resp. P'') entre u et v. Etant donnés deux entiers j, k, $1 \leq j \leq k \leq p$, $H_{j,k}$ désigne le sous-graphe H induit par $\bigcup_{i=j,\dots,k} V(C_i)$.

Avant de pouvoir montrer que les chemin-cycles induisent des facettes, nous donnons un lemme technique que nous utilisons fréquemment par la suite.

Lemme 4.7.1 *Soit $H = (W, T)$ un chemin-cycle d'ordre p. Soient j, k, $1 \leq j < k \leq p$, deux entiers tels que $k - j + 1$ soit pair. Soit $u_i \in \{w_i, z_i\}$ un sommet donné de P_i pour $i = j, j+2, \dots, k-1$. Soient v_1 et v_2 respectivement deux sommets de $V(C_j) \setminus \{u_j\}$ et $V(C_k) \setminus \{u_{k-1}\}$, qui appartiennent tout deux soit à $V(P')$ soit à $V(P'')$. Alors le sous-graphe $\bar{H}_{j,k}$ de $H_{j,k}$ induit par $\bigcup_{i=j,\dots,k} V(C_i) \setminus \{u_j, u_{j+2}, \dots, u_{k-1}\}$ est un arbre. De plus, la chaîne de $\bar{H}_{j,k}$ entre v_1 et v_2 a la même parité que $v_1 \sim v_2$.*

Preuve. Posons $U = \{u_j, u_{j+2}, \dots, u_{k-1}\}$. Il est facile de voir que $\bar{H}_{j,k}$ est un arbre et qu'il contient exactement une chaîne joignant v_1 et v_2. On nomme R cette chaîne (voir la figure 4.5).

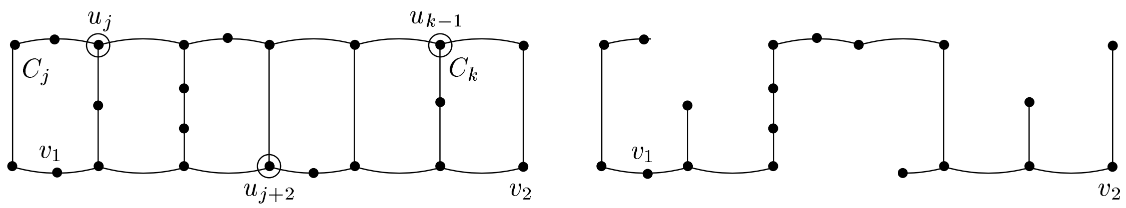


FIG. 4.5 –

Remarquons tout d'abord que $\{w_{j-1}, w_{j+1}, w_{j+3}, \dots, w_k\} \subset (W \setminus U)$. Soit R_i l'unique chaîne de $\bar{H}_{j,k}$ entre w_i et w_{i+2} pour $i = j-1, \dots, k-2$. Soit $w_i \in \{w_{j-1}, w_{j+1}, w_{j+3}, \dots,$

$w_{k-2}\}$. Si $w_{i+1} \in V(R)$, alors R_i est précisemment la sous-chaîne $w_i \sim w_{i+2}$. Si ce n'est pas le cas, alors R_i est la chaîne $P_i \cup (z_i \sim z_{i+2}) \cup P_{i+2}$. Et, par conséquent, $R_i \cup (w_i \sim w_{i+2})$ est le trou $H_{i,i+2} \setminus P_{i+1}$. Comme $H_{i,i+2}$ consiste en deux cycles impairs avec une chaîne en commun, il s'ensuit que $H_{i,i+2} \setminus P_{i+1}$ est pair et, qu'ainsi, R_i a la même parité que $(w_i \sim w_{i+2})$. Ceci implique que la chaîne, disons R', dans $\bar{H}_{j,k}$ entre w_{j-1} et w_k a la même parité que $w_{j-1} \sim w_k$.
Supposons que v_1 et v_2 appartiennent tous deux à $V(P')$. Le cas où v_1, v_2 appartienent à $V(P'')$ se déduit par symétrie. On remarque que $w_{j-1} \sim w_k = (v_1 \sim v_2) \cup (w_{j-1} \sim v_1) \cup (v_2 \sim w_k)$. Alors R' a la même parité que la chaîne $(v_1 \sim v_2) \cup (w_{j-1} \sim v_1) \cup (v_2 \sim w_k)$. Supposons que $w_j \in U$ et $w_{k-1} \notin U$. (Les autres cas: $w_j, w_{k-1} \in U$, $w_j, w_{k-1} \notin U$, $w_j \notin U$ et $w_{k-1} \in U$, peuvent être traités de la même façon.) Ainsi $R' = (R \setminus (w_{j-1} \sim v_1)) \cup (v_2 \sim w_k)$. Et, de même, les chaînes $(v_1 \sim v_2) \cup (w_{j-1} \sim v_1) \cup (v_2 \sim w_k)$ et $(R \setminus (w_{j-1} \sim v_1)) \cup (v_2 \sim w_k)$ ont même parité. On en déduit donc que R possède la même parité que $v_1 \sim v_2$. $\square$

Remarque 4.7.2 *p est pair si et seulement si P' et P'' ont même parité.*

Preuve. Soit $q = \sum_{i=1}^{p} |E(C_i)|$. Il est clair que p est pair si et seulement si q est pair. Comme chaque arête de P_i, $i = 1, ..., p-1$, est comptée deux fois dans la somme définissant q, alors q est pair si et seulement si P' et P'' ont même parité.$\square$

Maintenant, considérons un graphe $G = (V, E)$ et soit $H = (W, F)$ un sous-graphe induit de G qui est un chemin-cycle d'ordre p avec $p \geq 1$ tel que C' et C'' sont tous deux impairs. Par la remarque ci-dessus, il s'ensuit que p est pair. On associe à $H = (W, F)$ l'inégalité

$$x(W) \leq |W| - (\frac{p}{2} + 1). \tag{4.11}$$

Les inégalités du type (4.11) sont appelées *inégalités de chemin-cycles*. On a le résultat suivant.

Théorème 4.7.3 *L'inégalité (4.11) est valide pour $P_B(G)$.*

Preuve. Soit $B \in \beta(G)$, et supposons que $|B \cap W|$ est maximum. (Si l'inégalité (4.11) est vérifiée par x^B alors elle est vérifiée par le vecteur d'incidence de tout ensemble de sommets induisant un sous-graphe biparti de G.) Posons $\bar{B} = W \setminus B$ et $\overline{W} = W \setminus \bigcup_{i=1,\ldots,p-1} V(P_i)$. On distingue deux cas.

Cas 1. $\overline{W} \cap \bar{B} \neq \emptyset$. Soit $u \in \overline{W} \cap \bar{B}$. Si $u \in V(P^*) \setminus \{w_0, w_p\}$, alors comme $V(C_i) \cap \bar{B} \neq \emptyset$ pour $i = 1, 3, ..., p-1$, on obtient $|\bar{B}| \geq \frac{p}{2} + 1$, et ainsi (4.11) est satisfaite par x^B. Si ce n'est pas le ca, alors $u \in V(C_j)$ pour un $j \in \{1, 2, ..., p\}$. On peut supposer que j est pair. (Les autres cas où j est impair suivent par symétrie). Ainsi, comme $V(C_i) \cap \bar{B} \neq \emptyset$ pour $i = 1, 3, ..., j-1, j+1, j+3, ..., p-1$, il s'ensuit que $|\bar{B}| \geq \frac{p}{2} + 1$, impliquant que (4.11) est satisfaite par x^B.

Cas 2. $\overline{W} \cap \bar{B} = \emptyset$. Comme $V(C_i) \cap \bar{B} \neq \emptyset$, pour $i = 1, 3, ..., p-1$, on obtient $|\bar{B}| \geq \frac{p}{2}$. Supposons que $|\bar{B}| = \frac{p}{2}$. Il s'ensuit que $|\bar{B} \cap V(P_i)| = 1$ pour $i = 1, 3, ..., p-1$. Soit u_i le sommet de $\bar{B} \cap V(P_i)$ pour $i = 1, 3, ..., p-1$. Comme $|B \cap W|$ est maximum, on peut supposer que $u_1, u_3, ..., u_{p-1}$ sont tous dans $\{w_1, ..., w_{p-1}, z_1, ..., z_{p-1}\}$. D'après le lemme 4.7.1, il en résulte que $\bar{H} = H \setminus \{u_1, u_3, ..., u_{p-1}\}$ contient une chaîne, disons R, entre w_0 et w_p différent de P^* et ayant la même parité que $P_p = w_0 \sim w_p$. Comme C' est impair, le trou qui est formé des chaînes R et P^* est alors impair. Donc, $\bar{H}$ n'est pas biparti, une contradiction. Par conséquent, $|\bar{B}| \geq \frac{p}{2} + 1$ et x^B satisfait (4.11). □

Théorème 4.7.4 *L'inégalité (4.11) définit une facette de $P_B(H)$.*

Preuve. Notons l'inégalité (4.11) par $ax \leq \alpha$ et supposons qu'il existe une inégalité $bx \leq \beta$ qui définit une facette de $P_B(H)$ telle que $\{x \in P_B(H) \mid ax = \alpha\} \subseteq \{x \in P_B(H) \mid bx = \beta\}$. Nous montrons qu'il existe $\rho > 0$ tel que $b = \rho a$.
Pour cela, nous montrons d'abord qu'il existe $\rho \in \mathbb{R}$ tel que $b(u) = \rho$ pour tout sommet $u \in V(C') \cup V(C'')$. Soit $j \in \{1, 3, ..., p-1\}$ tel que j est impair. Soit $u \in V(C')$. Considérons les ensembles de sommets

$B_1 = W \setminus \{z_1, z_3, ..., z_j, w_j, z_{j+2}, z_{j+4}, ..., z_{p-1}\}$,

$B_2 = (B_1 \setminus \{u\}) \cup \{w_j\}$.

D'après le lemme 4.7.1, en prenant $v_1 = w_0$ et $v_2 = w_p$, il s'ensuit que le graphe obtenu à partir de $H_{0,p}$ en ôtant les sommets $z_1, z_3, ..., z_{p-1}$ est un arbre. Ainsi le graphe induit par $W \setminus \{z_1, z_3, ..., z_{p-1}\}$ contient un unique cycle (impair). Par conséquent, les graphes induits par B_1 et B_2 sont bipartis. De plus, $ax^{B_1} = ax^{B_2} = |W| - (\frac{p}{2} + 1) = \alpha$. Et alors $0 = bx^{B_1} - bx^{B_2} = b(u) - b(w_j)$. En posant $\rho_1 = b(w_j)$, on obtient que $b(u) = \rho_1$ pour tout sommet $u \in V(C')$. De la même manière, on peut montrer qu'il existe $\rho_2 \in \mathbb{R}$ tel que $b(v) = \rho_2$ pour tout sommet $v \in V(C'')$. Comme $w_0 \in V(C') \cap V(C'')$, il s'ensuit que $\rho_1 = \rho_2 = \rho$, et par conséquent

$$b(u) = \rho, \text{ pour tout } u \in V(C') \cup V(C''). \tag{4.12}$$

Nous montrons ensuite que $b(w) = \rho$, pour tout $w \in V(P_i) \setminus \{w_i, z_i\}$, et pour $i = 1, 2, ..., p-1$. Pour cela, considérons une chaîne P_i, pour $i = 1, 2, ..., p-1$, dont l'ensemble des sommets intérieurs est non vide et soit $w \in V(P_i) \setminus \{w_i, z_i\}$. Si i est pair (resp. impair), considérons les ensembles de sommets suivants

$$\begin{aligned} & B_3 = W \setminus \{z_1, z_3, ..., z_{i-1}, z_{i+1}, w_{i+2}, z_{i+3}, z_{i+5}, ..., z_{p-1}\}, \\ & B_4 = (B_3 \setminus \{w\}) \cup \{z_{i+1}\}. \\ \text{(resp.}\quad & B'_3 = W \setminus \{z_1, z_3, ..., z_i, w_{i+1}, z_{i+2}, z_{i+4}, ..., z_{p-1}\}, \\ & B'_4 = (B'_3 \setminus \{w\}) \cup \{z_i\}.) \end{aligned}$$

Par le lemme 4.7.1, il s'ensuit que B_3 et B_4 (resp. B'_3 et B'_4) induisent des sous-graphes bipartis de H. De plus, on a $ax^{B_3} = ax^{B_4} = \alpha$ (resp. $ax^{B'_3} = ax^{B'_4} = \alpha$). Ainsi, $0 = bx^{B_3} - bx^{B_4}$ (resp. $0 = bx^{B'_3} - bx^{B'_4}$). Ceci implique que $b(w) = b(z_{i+1})$ (resp. $b(w) = b(z_i)$). Par (4.12), on a donc $b(w) = \rho$.
Ainsi on obtient $b = \rho a$. De plus, comme $bx \leq \gamma$ définit une facette non triviale de $P_B(H)$, et, comme $b \geq 0$, on a $\rho > 0$, ce qui achève la preuve du théorème. □

Théorème 4.7.5 *Soit H^* le sous-graphe de G induit par $W \cup N(W)$. Supposons que H^* est de degré maximum trois et que les voisins dans H^* de chaque sommet $u \in N(W)$ appartiennent tous à la même face de H. Soit H_u, où $u \in N(W)$, le sous-graphe induit par $W \cup \{u\}$. Alors l'inégalité (4.11) définit une facette de $P_B(G)$ si et seulement si pour chaque sommet $u \in N(W)$ de degré trois dans H^*, au moins une des faces de H_u est paire.*

Une preuve du théorème 4.7.5 est donnée en annexe. Les inégalités de chemins-cycle seront également discutées comme une classe particulière d'une famille plus générale de contraintes valides appelées inégalités cellulaires qui seront présentées et étudiées dans le chapitre 5.

4.8 Construction de facettes pour $P_B(G)$

Dans cette section, nous décrivons certaines procédures de construction de facettes. Dans [14], Barahona et Mahjoub décrivent des méthodes de construction de facettes à partir de facettes pour le polytope $P_B(G)$. Celles-ci sont basées sur des transformations de facettes du polytope $P_B(G)$ et du polytope des stables.

Nous donnons tout d'abord un lemme technique qui sera utile pour les preuves de cette section.

Lemme 4.8.1 *Si $ax \leq \alpha$ définit une facette de $P_B(G)$, différente d'une inégalité triviale, alors $a(u) \geq 0$ pour tout $u \in V$ et $\alpha > 0$.*

Preuve. Supposons qu'il existe un sommet v avec $a_v < 0$. Comme $ax \leq \alpha$ est différente d'une inéqualité induite par $x(v) \geq 0$, il doit exister un ensemble $B \in \beta(G)$ tel que $u \in B$ et $ax^B = \alpha$. Soit $B' = B \setminus \{v\}$. On a $B' \in \beta(G)$. Cependant, comme $a_v < 0$, on obtient $ax^{B'} > \alpha$, ce qui est impossible.
D'un autre côté, comme $ax \geq \alpha$ définit une facette, au moins un des coefficients de a, disons a_u, est positif. Comme $\{u\} \in \beta(G)$, il s'ensuit que $0 < a_u \leq \alpha$. □

4.8.1 Ajout d'un sommet universel

Notre première méthode consiste à ajouter un sommet universel.

Théorème 4.8.2 *Soient $G = (V, E)$ un graphe et $ax \leq \alpha$ une inégalité non triviale définissant une facette de $P_B(G)$. Posons $\delta = max\{ax^S \mid S$ un stable de $G\}$. Soit $G' = (V', E')$ le graphe obtenu à partir de G en ajoutant un sommet universel v_0. Soit $\bar{a}x \leq \bar{\alpha}$ tel que*

$\bar{a}_u = a_u$ *si* $u \in V$,

$\bar{a}_{v_0} = \alpha - \delta,$

$\bar{\alpha} = \alpha.$

Alors $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(G')$.

Preuve. Comme $ax \leq \alpha$ n'est pas une inégalité triviale, alors, par le lemme 4.8.1, $\alpha > 0$. Montrons tout d'abord que $\bar{a}x \leq \bar{\alpha}$ est valide pour $P_B(G')$. Soit $B' \in \beta(G')$. Si $v_0 \notin B'$, alors $B' \in \beta(G)$ et donc $\bar{a}x^{B'} = ax^{B'} \leq \alpha = \bar{\alpha}$. Si $v_0 \in B'$, alors $B = B' \setminus \{v_0\}$ est un stable de G, et ainsi $ax^B \leq \delta$. Donc, $\bar{a}x^{B'} = ax^B + (\alpha - \delta) \leq \delta + \alpha - \delta = \bar{\alpha}$.
Ensuite, nous allons exhiber $|V'|$ ensembles de sommets induisant des sous-graphes bipartis induits dont les vecteurs d'incidence satisfont $\bar{a}x \leq \bar{\alpha}$ à l'égalité et sont linéairement indépendants, ce qui montre que $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(G')$. En effet, comme $ax \leq \alpha$ définit une facette de $P_B(G)$ avec $\alpha > 0$, il existe $n = |V|$ ensemble de sommets $B_1, ..., B_n \in \beta(G)$ tels que $ax^{B_i} = \alpha$, pour $i = 1, ..., n$ et $x^{B_1}, ..., x^{B_n}$ sont linéairement indépendants. Maintenant, considérons les ensembles

$B'_i = B_i$, pour $i = 1, ..., n$,

$B'_{n+1} = S^* \cup \{v_0\}$,

où S^* est un stable de G tel que $ax^{S^*} = \delta$. Il est clair que $B'_1, ..., B'_{n+1} \in \beta(G')$. De plus, leurs vecteurs d'incidence satisfont $\bar{a}x \leq \bar{\alpha}$ à l'égalité et sont linéairement indépendants.
□

Le théorème suivant est une conséquence directe du théorème 4.8.2.

Corollaire 4.8.3 (Ajout d'un graphe complet universel)
Soient $G = (V, E)$ un graphe et $ax \leq \alpha$ une inégalité non triviale définissant une facette de $P_B(G)$. Posons $\delta = max\{a^t x^S \mid S \text{ stable de } G\}$. Soit $G' = (V', E')$ le graphe obtenu à partir de G en ajoutant un graphe complet $H = (W, F)$ et en liant chaque sommet de H à tout sommet de G. Soit $\bar{a}x \leq \bar{\alpha}$ tel que

$\bar{a}_u = a_u$ *si* $u \in V$,

$\bar{a}_v = \alpha - \delta$ *si* $v \in W$,

$\bar{\alpha} = \alpha.$

Alors $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(G')$.

4.8.2 Eclatement et contraction

Notre seconde procédure de contruction de facettes consiste à éclater un sommet. Le théorème suivant permet de décrire cette procédure et montre que l'opération inverse est possible.

Théorème 4.8.4 (a) (Eclatement d'un sommet)
Soient $G = (V, E)$ un graphe et $ax \leq \alpha$ une inégalité définissant une facette non triviale de $P_B(G)$. Prenons $v \in V$ tel que $a_v > 0$ et F un ensemble d'arêtes de $\delta(v)$. Soit $G' = (V', E')$ un graphe obtenu à partir de G en éclatant le sommet v en deux sommets v_1, v_2 tels que v_1 soit incident à toutes les arêtes de F et v_2 soit incident à toutes les arêtes de $\delta(v) \setminus F$. Soit $a'_u = a_u$ pour tout $u \in V \setminus \{v\}$ et $a'_{v_1} = a'_{v_2} = a_v$. Supposons que tout sous-graphe biparti induit maximal de G' intersecte $\{v_1, v_2\}$ et que $\alpha + 2a_v = max\{a'x^B \mid B \in \beta(G')\}$. Soit $\bar{G} = (\bar{V}, \bar{E})$ le graphe obtenu à partir de G' en ajoutant un nouveau sommet v_3 et les arêtes v_1v_3 et v_3v_2 (Voir la figure 4.6). Posons

$\bar{a}_u = a_u$ *si* $u \in V \setminus \{v\}$,

$\bar{a}_{v_1} = \bar{a}_{v_2} = \bar{a}_{v_3} = a_v,$

$\bar{\alpha} = \alpha + 2a_v.$

Alors $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(\bar{G})$.

(b) (Contraction d'une chaîne de longueur deux)
Soit $G = (V, E)$ un graphe et $ax \leq \alpha$ une inégalité non triviale définissant une facette de $P_B(G)$. Soient v_1, v_2, v_3 trois sommets de G tels que (v_1, v_2, v_3) soit une chaîne, v_3 soit de degré deux et $a_{v_1} = a_{v_2} = a_{v_3} > 0$. Supposons que tout sous-graphe biparti induit maximal de G intersecte $\{v_1, v_2\}$. Soit $G' = (V', E')$ le graphe obtenu à partir de G en contractant v_1v_3 et v_3v_2. Notons par v le sommet créé par la contraction, et posons

$\bar{a}_u = a_u$ *si* $u \in V \setminus \{v\}$,

$\bar{a}_v = a_{v_1}$,

$\bar{\alpha} = \alpha - 2a_{v_1}$.

Alors $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(G')$.

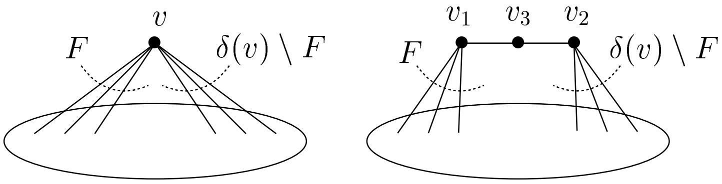


FIG. 4.6 –

Preuve. (a) Soit $\bar{B} \in \beta(\bar{G})$. Si $|\{v_1, v_2, v_3\} \cap \bar{B}| \leq 2$, comme $\bar{B} \setminus \{v_1, v_2, v_3\} \in \beta(G)$, il s'ensuit que $\bar{a}x^{\bar{B}} \leq \alpha + 2a_v$. Si $\{v_1, v_2, v_3\} \subset \bar{B}$, alors on pose $B = (\bar{B} \setminus \{v_1, v_2, v_3\}) \cup \{v\}$. Il est clair que $B \in \beta(G)$ et ainsi $\bar{a}x^{\bar{B}} = ax^B - a_v + 3a_v \leq \alpha + 2a_v$. Par conséqent $\bar{a}x \leq \bar{\alpha}$ est valide pour $P_B(\bar{G})$.
Maintenant nous montrons que $\bar{a}x \leq \bar{\alpha}$ définit une facette. Soit $bx \leq \gamma$ une inégalité définissant une facette de $P_B(\bar{G})$ telle que $\{x \in P_B(\bar{G}) \mid \bar{a}x = \bar{\alpha}\} \subseteq \{x \in P_B(\bar{G}) \mid bx = \gamma\}$. Nous allons montrer que $b = \rho\bar{a}$ pour un $\rho > 0$.
Soit $B^* \in \beta(G')$ tel que $a'x^{B^*} = \alpha + 2a_v$. Alors on a $v_1, v_2 \in B^*$. En fait, supposons, par exemple, que $v_1 \notin B^*$. Soit $B = B^* \setminus \{v_2\}$ si $v_2 \in B^*$ et $B = B^*$ sinon. Ainsi $B \in \beta(G)$ et $ax^B > \alpha$, une contradiction. Maintenant considérons les ensembles $B_1^* = (B^* \setminus \{v_1\}) \cup \{v_3\}$ et $B_2^* = (B^* \setminus \{v_2\}) \cup \{v_3\}$. Alors $B^*, B_1^*, B_2^* \in \beta(\bar{G})$ avec $\bar{a}x^{B^*} = \bar{a}x^{B_1^*} = \bar{a}x^{B_2^*} = \bar{\alpha}$. Ceci implique que $bx^{B^*} = bx^{B_1^*} = bx^{B_2^*} = \gamma$, et donc $b_{v_1} = b_{v_2} = b_{v_3}$.
Comme $ax \leq \alpha$ définit une facette de $P_B(G)$, il existe $n = |V|$ ensembles de sommets $B_1, ..., B_n \in \beta(G)$ tels que $x^{B_1}, ..., x^{B_n}$ vérifient $ax \leq \alpha$ à l'égalité et sont linéairement indépendants. Comme $ax \leq \alpha$ est différent d'une inégalité triviale, il existe au moins un ensemble parmi $B_1, ..., B_n$ qui contient (qui ne contient pas) v. Donc supposons que v appartient à k, $k < n$, de ces ensembles. S.p.d.g., on peut supposer que $v \in B_1, ..., B_k$. Comme tout ensemble de sommets induisant un sous-graphe biparti maximal de G' intersecte $\{v_1, v_2\}$, il s'ensuit que pour chaque ensemble B_i, $i = k+1, ..., n$, il existe $v_i \in \{v_1, v_2\}$ tels que $B_i' = B_i \cup \{v_i\} \in \beta(G')$. Considérons les ensembles de sommets suivants

$\bar{B}_i = B_i \setminus \{v\} \cup \{v_1, v_2, v_3\}$, pour $i = 1, ..., k$,

$\bar{B}_i = B_i' \cup \{v_3\}$, pour $i = k+1, ...n$,

qui appartiennent à $\beta(\bar{G})$ et vérifient $\bar{a}x^{\bar{B}_i} = \bar{\alpha}$, pour $i = 1, ..., n$. On sait que $bx^{\bar{B}_i} = \gamma$, pour $i = 1, .., n$. Soit $b' \in \mathbb{R}^V$ tel que $b'_u = b_u$, pour $u \in V \setminus \{v\}$ et $b'_v = b_{v_1}$. Les vecteurs d'incidence $x^{B_1}, ..., x^{B_n}$ satisfont les inégalités $b'x \leq \gamma - 2b_{v_1}$ à l'égalité et sont linéairement indépendants. Par conséquent, $b'x \leq \gamma - 2b_{v_1}$ définit une facette pour $P_B(G)$.

On peut remarquer que cette facette contient celle définie par $ax \leq \alpha$. Ainsi, $b_{v_1} = \rho a_v$ et $b_u = \rho a_u$, pour tout $u \in V \setminus \{v\}$, pour un $\rho \in I\!R$. Ceci implique $b_{v_2} = b_{v_3} = \rho a_v$, et, finalement, $b = \rho \bar{a}$.
De plus, comme $ax \leq \alpha$ n'est pas une inégalité triviale, $bx \leq \gamma$ est aussi non triviale et alors $\rho > 0$. Ainsi $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(\bar{G})$.

(b) La validité de la nouvelle inégalité peut être facilement démontrée par des constructions élémentaires comme il a été fait en (a). Nous allons montrer que $\bar{a}x \leq \bar{\alpha}$ définit une facette.
Comme $ax \leq \alpha$ définit une facette de $P_B(G)$, il existe $n = |V|$ ensembles de sommets $B_1, ..., B_n \in \beta(G)$ dont les vecteurs d'incidence vérifient $ax^{B_i} = \alpha$, pour $i = 1, ..., n$ et sont linéairement indépendants. Posons

$$B'_i = \begin{cases} B_i \setminus \{v_1, v_2, v_3\} \cup \{v\}, & \text{si } \{v_1, v_2, v_3\} \subset B_i, \\ B_i \setminus \{v_1, v_2, v_3\}, & \text{sinon}, \end{cases}$$

pour $i = 1, .., n$. Il est clair que $B'_1, ..., B'_n \in \beta(G')$. De plus, on sait que $\bar{a}x^{B'_i} = \bar{\alpha}$ pour $i = 1, ..., n$. Posons M (resp. M') la matrice dont les colonnes sont les vecteurs $x^{B_1}, ..., x^{B_n}$ (resp. $x^{B'_1}, ..., x^{B'_n}$). Pour prouver notre résultat, il suffit de montrer que M' est de rang $n-2$. Pour cela notons d'abord que pour $i = 1, ..., n$, $|B_i \cap \{v_1, v_2, v_3\}| \geq 2$. En effet, si ce n'est pas le cas, alors par hypothèse, on déduit qu'il existe $B_i, i \in \{1, ..., n\}$, tels que $v_3 \notin B_i$ et B_i contient exactement un des sommets v_1, v_2, disons v_1. Cependant comme $B_i \cup \{v_3\}$ induit encore un sous-graphe biparti et $a_{v_3} > 0$, il s'ensuit que $ax^{B_i \cup \{v_3\}} > \alpha$, une contradiction. De plus, il doit exister au moins un ensemble B_i qui contient exactement un des sommets parmi $\{v_1, v_2, v_3\}$ et au moins un ensemble B_i qui contient $\{v_1, v_2, v_3\}$. Autrement, la facette définie par $ax \leq \alpha$ devrait être contenue soit dans la face $\{x \in P_B(G) \mid x(v_1) + x(v_2) + x(v_3) = 2\}$ soit dans la face $\{x \in P_B(G) \mid x(v_1) + x(v_2) + x(v_3) = 3\}$. Comme $x(v_1) + x(v_2) + x(v_3) \leq 2$ n'est pas valide (on peut remarquer que $\{v_1, v_2, v_3\} \in \beta(G)$), ceci contredit le premier cas. Si le deuxième cas se produit alors il s'ensuit que la facette est incluse dans la facette triviale $\{x \in P_B(G) \mid x(v_1) = 1\}$. Mais ceci constitue à nouveau une contradiction. Par conséquent, on peut supposer, s.p.d.g, que les ensembles $B_1, ..., B_n$ sont ordonnés de telle façon que $B_1, ..., B_k$, $1 \leq k < n$, contiennent $\{v_1, v_2, v_3\}$ et $B_{k+1}, ..., B_n$ contiennent exactement deux sommets parmi $\{v_1, v_2, v_3\}$. Par conséquent M' a la forme suivante

$$M' = \left[\begin{array}{ccccccc} & & & A & & & \\ \hline 1 & \ldots & 1 & & 0 & \ldots & 0 \end{array} \right]$$

où la dernière ligne correspond à v et A est la sous-matrice de M donnée par les lignes associées aux sommets de $V \setminus \{v_1, v_2, v_3\}$.
Notons par $L = (2, ..., 2)$ le vecteur-ligne qui contient uniquement des 2. On remarque que $L = \frac{2}{\alpha} a^t.M$ (on rapelle que $\alpha > 0$). Si l'on soustrait L à la somme des lignes de M correspondant à v_1, v_2, v_3, on obtient la dernière ligne de M'. Ce qui implique que les lignes de M' ne sont rien d'autres que des combinaisons linéaires des lignes de M. Ce qui montre que M' est de rang $n - 2$. $\square$

4.8.3 Subdivision d'une étoile

Notre dernière opération consiste à subdiviser une étoile.

Théorème 4.8.5 *Soit $G = (V, E)$ un graphe et $ax \leq \alpha$ une inégalité non triviale définissant une facette de $P_B(G)$. Soit $v \in V$ tel que $a_v > 0$. Supposons que, pour chaque arête $e \in E$ incidente à v, il existe un ensemble B qui induit un sous-graphe biparti de $G \setminus e$ et tel que $ax^B = \alpha + a_v$. Posons $G' = (V', E')$ le graphe obtenu à partir de G en subdivisant les arêtes incidentes à v (voir la figure 4.7). Posons*

$\bar{a}_u = a_u$, *si* $u \in V$,

$\bar{a}_u = a_v$, *si* $u \in V' \setminus V$,

$\bar{\alpha} = \alpha + |N(v)|a_v$.

Alors $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(G')$.

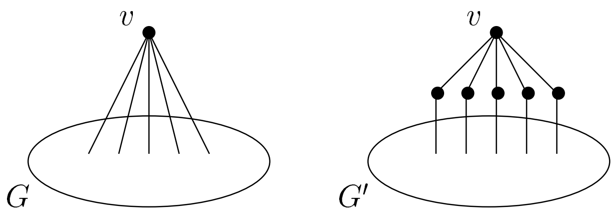


FIG. 4.7 –

Preuve. Soit $N(v) = \{u_1, ..., u_k\}$, $k \geq 1$. Pour $j = 1, ..., k$, on pose v_j le sommet ajouté sur l'arête vu_j.
Montrons tout d'abord que $\bar{a}x \leq \bar{\alpha}$ est valide pour $P_B(G')$. Soit $B' \in \beta(G')$. Si $v \notin B'$, alors $B' \setminus \{v_1, ..., v_k\} \in \beta(G)$ et ainsi $\bar{a}x^{B'} \leq \alpha + ka_v$. Supposons donc que $v \in B'$. On considère deux cas
Cas 1. $|\{v_1, ..v_k\} \cap B'| < k$. Comme $B' \setminus \{v, v_1, ..., v_k\} \in \beta(G)$, il s'ensuit que $\bar{a}x^{B'} \leq \alpha + ka_v$.

Cas 2. $\{v_1, .., v_k\} \subset B'$. Soit $B = B' \setminus \{v_1, ..v_k\}$. On a $B \in \beta(G)$. En effet, supposons que ce ne soit pas le cas, et considérons un cycle impair C de $(B, E(B))$. Il est clair que $v \in V(C)$, autrement C serait un cycle impair dans le graphe obtenu en enlevant v, ce qui est impossible. Soient u_1, u_2 les sommets de $V(C)$ adjacents à v et soit P la chaîne de C joignant u_1 et u_2 et ne contenant pas v, c'est-à-dire $P = C \setminus \{u_1v, vu_2\}$. Soit C' le cycle obtenu à partir de P en ajoutant les arêtes $\{u_1v_1, v_1v, vv_2, v_2u_2\}$. Noter que C' est impair. Comme C' est un cycle du sous-graphe de G' induit par B', on a une contradiction. Donc $B \in \beta(G)$ et, par conséquent, $\bar{a}x^{B'} = ax^B + ka_v \leq \alpha + ka_v \leq \bar{\alpha}$. Par conséquent, $\bar{a}x \leq \bar{\alpha}$ est valide pour $P_B(G')$.
Comme $ax \leq \alpha$ définit une facette de $P_B(G)$, il existe $n = |V|$ ensembles de sommets $B_1, ..., B_n \in \beta(G)$ tels que $ax^{B_i} = \alpha$ pour $i = 1, ..., n$ et $x^{B_1}, ..., x^{B_n}$ sont linéairement indépendants. Pour $u_j \in N(v)$, on pose D_j l'ensemble de $\beta(G \setminus vu_j)$ tel que $ax^{D_j} = \alpha + a_v$. Considérons les ensembles

$$\begin{array}{ll} B'_i = B_i \cup \{v_1, ...v_k\}, & \text{pour } i = 1, .., n, \\ B'_{n+j} = D_j \cup \{v_1, ..., v_k\} \setminus \{v_j\}, & \text{pour } j = 1, ..., k, \end{array}$$

qui induisent des sous-graphes bipartis de G'. De plus, $\bar{a}x^{B'_i} = \bar{\alpha}$ pour $i = 1, ..., n+k$. Soit M (resp. M') la matrice dont les colonnes sont les vecteurs x^{B_i}, $i = 1, ..., n$ (resp. $x^{B'_i}$, $i = 1, ..., n+k$). Alors M' a la forme suivante:

$$M' = \left[\begin{array}{cccccccc} & & & & & & & \\ & M & & & & A & & \\ & & & & & & & \\ 1 & .. & 1 & 0 & 1 & . & . & 1 \\ & & & 1 & . & & & . \\ : & & : & . & & . & & . \\ & & & . & & & . & 1 \\ 1 & .. & 1 & 1 & . & . & 1 & 0 \end{array}\right],$$

où les k dernières lignes de M' correspondent aux sommets $v_1, ..., v_k$ et A est une matrice en 0-1. Remarquons que la ligne n de M' correspond au sommet v. Aussi remarquons que $a_v > 0$. Maintenant, effectuons les opérations suivantes sur M': remplacer la ligne n par $\sum_{i=1}^{i=n} a_i M'_i$, où M'_i est la ligne i de M', et multiplier chaque ligne i par α, pour $i = n+1, ..., n+k$. Par la suite, soustraire la ligne n à chaque ligne i, pour $i = n+1, ..., n+k$. Ceci permet de construire une matrice de la forme

$$\left[\begin{array}{cc} \bar{M} & \bar{A} \\ 0 & -D \end{array}\right]$$

où $\bar{M}$ est la matrice obtenue à partir de M par des combinaisons linéaires et D est une matrice qui peut être décrite comme

$$\begin{bmatrix} \alpha + a_v & a_v & . & . & a_v \\ a_v & \alpha + a_v & & & . \\ . & & . & & . \\ . & & & . & a_v \\ a_v & . & . & a_v & \alpha + a_v \end{bmatrix}.$$

Par conséquent $\bar{M}$ est non-singulière. De plus, comme $\alpha, a_v > 0$, il est facile de voir que D est aussi non-singulière. En conséquence, M' est non-singulière et donc $x^{B'_1}, ..., x^{B'_{n+k}}$ sont linéairement indépendants, ce qui implique que $\bar{a}x \leq \bar{\alpha}$ définit une facette de $P_B(G')$. □

4.8.4 Applications et exemples

Tout d'abord on peut remarquer que la validité de certaines des inégalités introduites dans la section précédente peut également être montré en utilisant les opérations de constructions de facettes. En fait, une inégalité de clique (4.4) peut être obtenue à partir d'un triangle en ajoutant des sommets universels (théorème 4.8.2). De même, une inégalité de p-roue d'ordre q peut être obtenue à partir d'un cycle de longueur p en ajoutant un graphe complet sur q sommets (corollaire 4.8.3).

D'autres inégalités valides et définissant des facettes peuvent êtres également générées en utilisant les procédures de constructions de facettes. Considérons par exemple un trou $H_1 = (W_1, E_1)$ sur trois sommets, voir la figure 4.8 (a). Dans [14], il est montré que l'inégalité de clique $x(W_1) \leq 2$ définit une facette de $P_B(H_1)$. En appliquant deux fois le théorème 4.8.2, on obtient le graphe complet $H_2 = (W_2, E_2)$ de la figure 4.8 (b) et l'inégalité $x(W_2) \leq 2$ qui définit une facette de $P_B(H_2)$. Maintenant, choisissons un sommet v, prenons un ensemble F de deux arêtes incidentes à v et appliquons le théorème 4.8.4 (a) pour v et F. Ceci conduit au graphe $H_3 = (W_3, F_3)$ de la figure 4.8 (c) et l'inégalité associée $x(W_3) \leq 4$ qui définit une facette. En appliquant plusieurs fois le théorème 4.8.4 (a) pour les sommets restants de W_2, on obtient le graphe $H_4 = (W_4, E_4)$ de la figure 4.8 (d). L'inégalité définissant une facette obtenue est alors $x(W_4) \leq 12$. Maintenant, si on répète le théorème 4.8.5 pour les sommets du cycle extérieur de H_4, on obtient le graphe $H_5 = (W_5, E_5)$ de la figure 4.8 (e), et l'inégalité associée $x(W_5) \leq 27$ qui définit une facette. Finalement, en utilisant le théorème 4.8.4 (b), on obtient le graphe $H_6 = (W_6, E_6)$ de la figure 4.8 (f) et l'inégalité définissant une facette $x(W_6) \leq 7$. On peut remarquer que le graphe H_5 obtenu est le graphe de Petersen.

Il est facile de voir qu'une inégalité de p-roue liftée (4.10) peut être obtenue à partir

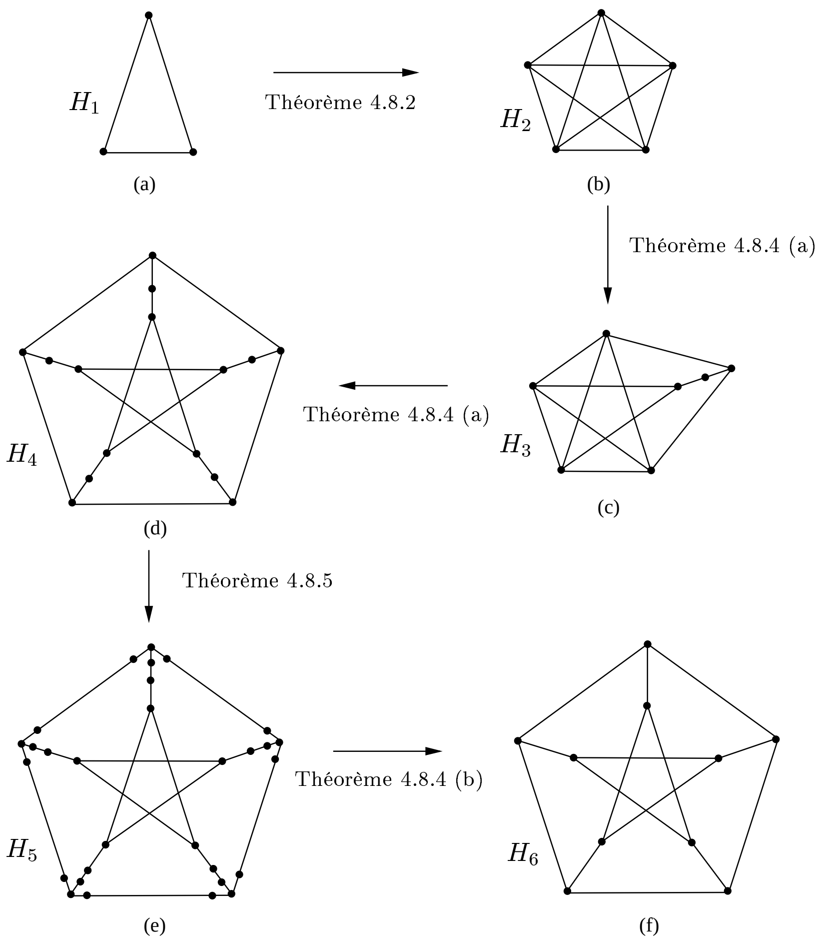


FIG. 4.8 – *Construction de facettes*

d'une inégalité de p-roue d'ordre 1 en appliquant plusieurs fois les opérations: contraction d'une chaîne et subdivision d'une étoile. Ceci implique que les inégalités (4.10) sont valides pour $P_B(G)$. De plus, ces inégalité peuvent définir des facette. La figure 4.9 montre la construction d'une 5-roue liftée (Figure 4.9(c)) à partir d'une 5-roue d'ordre 1 (figure 4.9(a)). Le graphe de la figure 4.9(b) est obtenu par les opérations de subdivision des étoiles $\delta(v_i), i = 1, 2, 4, 5$. On peut remarquer que $\delta(v_1)$ et $\delta(v_2)$ sont subdivisées deux fois. Le graphe de la figure 4.9(c) est obtenu par contraction de chaînes de trois sommets.

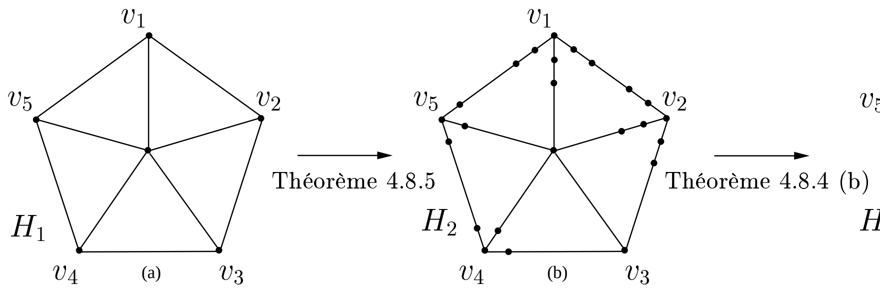


FIG. 4.9 – *Construction d'une 5-roue liftée*

4.9 conclusion

Dans ce chapitre, nous avons présenté des inégalités valides pour le problème du sous-graphe biparti induit. Nous avons également donné des conditions nécessaires et suffisantes pour que certaines de ces contraintes définissent des facettes pour $P_B(G)$. Pour plusieurs classes de contraintes, nous avons montré que le problème de séparation associé peut être résolu en temps polynomial. Enfin, nous avons proposé des procédures de constructions de facettes.

Dans le chapitre suivant, nous poursuivrons l'étude faciale de $P_B(G)$ en étudiant une classe d'inégalités valides appelées inégalités cellulaires, qui généralisent les inégalités de cycles impairs et de chemins-cycle présentées dans ce chapitre.

Chapitre 5

Inégalités cellulaires

Dans ce chapitre, nous présentons une autre classe de contraintes valides pour le polytope des sous-graphes bipartis induits. Nous discutons d'abord certaines propriétés concernant leurs graphes induits appelés graphes cellulaires. Nous donnons ensuite des conditions nécessaires et suffisantes pour que ces inégalités définissent des facettes de $P_B(G)$. Enfin, nous étudions deux cas particuliers de ces contraintes et nous montrons que le problème de séparation associé peut être résolu en temps polynomial.

5.1 Graphes cellulaires

Etant donné un graphe planaire G, il est clair que tout graphe obtenu de G en reliant, à l'aide d'une chaîne, deux sommets quelconques d'une même face est aussi planaire.

Une graphe $G = (V, E)$ sera dit *cellulaire* s'il peut être construit à partir d'un trou en appliquant l'opération suivante.

θ : Choisir deux sommets de degré deux d'une même face et les relier par une chaîne élémentaire.

En conséquence, les graphes cellulaires sont les graphes planaires, 2-connexes et de degré maximum 3. La figure 5.1 montre un graphe cellulaire avec 8 faces (y compris la face extérieure).

Lemme 5.1.1 *Un graphe cellulaire contient un nombre pair de faces impaires.*

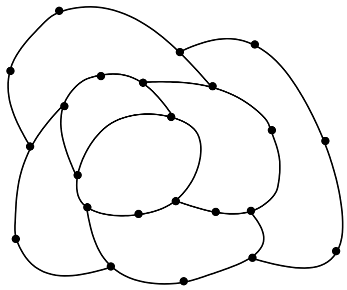


FIG. 5.1 – *Un graphe cellulaire à 8 faces*

Preuve. Soit $G = (V, E)$ un graphe cellulaire contenant p faces et soit $G_D(V_D, E_D)$ le graphe dual de G. Ainsi, un sommet u de G_D correspondant à une face de G_D a un degré égal au nombre d'arêtes de cette face. Comme un graphe possède un nombre pair de sommet de degré impair, le graphe G contient un nombre pair de faces impaires. □

On peut alors déduire le résultat suivant à propose des représentations planaires d'un graphe cellulaire.

Corollaire 5.1.2 *Soit $G = (V, E)$ un graphe celullaire. Sur une représentation planaire de G, la face extérieure est impaire si et seulement G contient un nombre impair de faces intérieures impaires.*

Si $G = (V, E)$ est un graphe cellulaire, on dit que deux faces de G sont *adjacentes* si elles ont au moins une arête en commun. On donne à présent un lemme technique qui sera utile pour montrer que les graphes cellulaires produisent des facettes de $P(G)$.

Lemme 5.1.3 *Soit $G = (V, E)$ un graphe cellulaire contenant p faces impaires, $p \geq 2$. Il existe un ensemble W de sommets de V et un ensemble F d'arêtes de E tels que le graphe $G' = (V \setminus W, E \setminus F)$ est un graphe cellulaire contenant p faces impaires et dont une de ces faces impaires est uniquement adjacente à des faces impaires.*

Preuve. Soit C_0 une face impaire de G. Les ensembles W et F peuvent être construits en suivant l'algorithme suivant:

> tant que C_0 est adjacente à une face paire, ôter les arêtes et les sommets intérieurs des chaînes communes à C_0 et à ses faces paires adjacentes.

A chaque itération de cet algorithme, G perd une face paire et conserve toutes ses faces impaires. Par conséquent, le graphe G obtenu à la fin de cet algorithme contient p faces impaires et sa face C_0 est adjacente uniquement à des faces impaires. □

5.2 Inégalités cellulaires

5.2.1 Définition

Soient $G = (V, E)$ un graphe et $H = (W, F)$ un sous-graphe cellulaire induit de G contenant p faces impaires. Par le lemme 5.1.1, p est pair. Considérons l'inégalité suivante associée au sous-graphe H

$$x(W) \leq |W| - \frac{p}{2}. \tag{5.1}$$

Les inégalités du type (5.1) seront appelées *inégalités cellulaires*. On a le résultat suivant.

Théorème 5.2.1 *L'inégalité (5.1) est valide pour $P_B(G)$.*

Preuve. La preuve est par récurrence sur p.
Si $p = 0$, comme H est biparti, $x(V(W)) \leq |W|$. Si $p = 2$, comme H contient un cycle impair, il est clair que $x(V(W)) \leq |W| - 1$. Maintenant, supposons que l'inégalité (5.1) est valide pour tous les graphes cellulaires contenant p' faces impaires avec $2 \leq p' \leq p - 2$.
Considérons d'abord le cas où il existe une face impaire C_0 de H qui est adjacente uniquement à des faces impaires. Supposons que la contrainte (5.1) n'est pas valide pour $P_B(G)$. Par conséquent, il existe $B \in \beta(G)$ tel que $|W \setminus B| \leq \frac{p-2}{2}$. Nous allons montrer que $|B \cap V(C_0)| = \emptyset$, ce qui est une contradiction. Soient $u_1, ..., u_q$ les sommets de C_0 de degré 3 pris dans le sens de la montre. Soient P_i, $i = 1, ..., q$, la sous-chaîne de C_0 liant u_i à u_{i+1} et qui ne contient aucun autre sommet parmi $u_1, ..., u_q$ (les indices sont pris modulo q). Soit $i_0 \in \{1, ..., q\}$. Soit $H' = (W', E')$ le graphe obtenu en supprimant les sommets intérieurs des chaînes P_{i_0} et P_{i_0+1}. Alors H' est un graphe cellulaire auquel s'ajoute une chaîne pendante. De plus, H' possède $p - 2$ faces impaires. D'où, par l'hypothèse de récurrence, on a $|W' \setminus B| \geq \frac{p-2}{2}$. Ainsi, $W \setminus B$ ne contient aucun des sommets intérieurs de la chaîne $(P_{i_0} \cup P_{i_0+1})$. Par conséquent, B contient tous les sommets de la face C_0, ce qui est une contradiction. Ainsi, l'inégalité (5.1) est valide dans ce cas.
Considérons maintenant le cas où toutes les faces impaires de H sont adjacentes à des faces paires. D'après le lemme 5.1.3, il existe $W' \subset W$ et $F' \subset F$ tels que le graphe $H'' = (W \setminus W', F \setminus F')$ est un graphe cellulaire contenant p faces impaires dont une est uniquement adjacente à des faces impaires. D'après l'hypothèse de récurrence et le cas précédent, l'inégalité

$$x(W \setminus W') \leq |W \setminus W'| - \frac{p}{2}. \tag{5.2}$$

est valide pour H''. Ainsi, l'inégalité (5.1) peut être obtenue en sommant l'inégalité (5.2) avec les inégalités $x(u) \leq 1$, $u \in W'$, ce qui termine la preuve. □

5.2.2 Conditions nécessaires

Dans cette section, nous donnons des conditions nécessaires pour que l'inégalité (5.1) définisse une facette pour $P_B(G)$.

Théorème 5.2.2 *L'inégalité (5.1) définit une facette de $P_B(G)$ seulement si*
i) $p \geq 2$,
ii) H ne contient pas de face paire possédant un sommet de degré 2,
iii) H ne contient pas une chaîne P dont les sommets intérieurs sont de degré 2 et telle que $H \setminus P$ soit non connexe.

Preuve. i) Si $p = 0$, l'inégalité (5.1) ne définit clairement pas de dacettes. ii) Supposons que G contienne un sommet u de degré 2 appartenant à une face paire. Alors le graphe induit par $W \setminus \{u\}$ est un graphe cellulaire auquel s'ajoute des chaînes pendantes. De plus ce graphe contient encore p faces impaires et donc l'inégalité $x(W \setminus \{u\}) \leq |W \setminus \{u\}| - \frac{p}{2}$ est valide pour $P(G)$. En sommant cette dernière inégalité et $x(u) \leq 1$, on obtient l'inégalité (5.1) qui ne peut donc définir de facette pour $P(G)$.
iii) Supposons que H possède une chaîne P dont les sommets intérieurs sont de degré 2 et telle que $H \setminus P$ soit non connexe. Par conséquent, cette chaîne est commune à deux faces de H, disons C_1 et C_2, de manière à ce que H puisse être représenté planairement comme sur la figure 5.2.

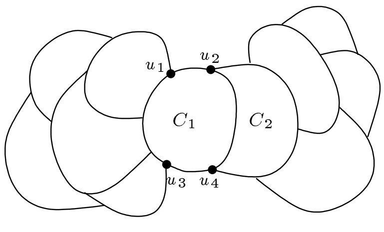


FIG. 5.2 –

Soit H' le sous-graphe obtenu à partir de H en ôtant les sommets intérieurs des chaînes $(C_1 \cap C^*) \cup (C_2 \cap C^*) \cup (C_1 \cap C_2)$. Noter que H' n'est pas connexe et que toutes ses composantes connexes sont des graphes cellulaires. S.p.d.g., on peut supposer qu'il y a seulement deux composantes connexes. Soient $H'_1 = (W'_1, F'_1)$ (resp. $H'_2 = (W'_2, F'_2)$) la composante connexe de H' contenant des sommets de la face C_1 (resp. C_2). Considérons le graphe H_1 (resp. H_2) induit par les sommets de $W'_1 \cup V(C_1)$ (resp. $W'_2 \cup V(C_2)$). Notons p_1 (resp. p_2, p'_1, p'_2) le nombre de faces impaires du graphe H_1 (resp. H_2, H'_1, H'_2). On peut montrer facilement que soit $p = p'_1 + p_2$, soit $p = p_1 + p'_2$. S.p.d.g., supposons que $p = p'_1 + p_2$. Dans ce cas, l'inégalité (5.1) peut être obtenue en sommant les inégalités cellulaires associées à H'_1 et H_2

$$x(W'_1) \leq |W'_1| - \tfrac{p_1}{2} \quad \text{et} \quad x(W'_2 \cup V(C_2)) \leq |W'_2 \cup V(C_2)| - \tfrac{p_2}{2}.$$

et les inégalités triviales $x(u) \leq 1$ pour tout $u \in ((V(C_1) \setminus W'_1) \setminus V(C_2))$, ce qui implique que l'inégalité (5.1) ne peut définir de facette pour $P_B(G)$. $\square$

On peut noter que lorsqu'un graphe cellulaire contient une face impaire possédant uniquement des sommets de degré 3, alors l'inégalité (5.1) associée peut définir une facette.

5.2.3 Graphes p-cellulaires

Dans cette section, nous introduisons une sous-classe des inégalités cellulaires pour laquelle nous décrivons des conditions nécessaires et suffisantes pour qu'elles définissent des facettes de $P_B(G)$.

On dit qu'un graphe cellulaire H' est *p-cellulaire* s'il contient exactement p faces qui sont toutes impaires et qu'il ne contient pas une chaîne P dont les sommets intérieurs sont de degré 2 et telle que $H' \setminus P$ soit non connexe. Un graphe p-cellulaire $H = (W, F)$ est dit *fortement p-cellulaire* si, pour toute face C de H, il existe un ensemble de sommets $B_C \subset W$ tels que $|W \setminus B_C| = \frac{p-2}{2}$, $V(C) \setminus B_C = \emptyset$ et $(B_C \setminus \{u\}) \in \beta(H)$, pour tout $u \in V(C)$.

On a le résultat suivant.

Théorème 5.2.3 *Soit $H = (W, F)$ un graphe fortement p-cellulaire. Alors l'inégalité cellulaire (5.1) associée à H définit une facette de $P_B(H)$.*

Preuve. Notons par $ax \leq \alpha$ l'inégalité (5.1) et supposons qu'il existe une inégalité $bx \leq \beta$ qui définit une facette de $P_B(H)$ telle que $\{x \in P_B(H) \mid ax = \alpha\} \subseteq \{x \in P_B(H) \mid bx = \beta\}$. Afin de prouver que (5.1) définit une facette de $P_B(H)$, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Soient $C_1, ..., C_p$ les faces de H. Soit $i_0 \in \{1, ..., p\}$. Montrons d'abord qu'il existe $\rho_{i_0} \in \mathbb{R}$ tel que $b(v) = \rho_{i_0}$ pour tout sommet $v \in V(C_{i_0})$. Comme H est fortement p-cellulaire, il existe un ensemble de sommets B_0 tel que $|W \backslash B_0| = \frac{p-2}{2}$ et $V(C_{i_0}) \backslash B_0 = \emptyset$. De plus, pour tout sommet $v \in V(C_{i_0})$, $B_0 \setminus \{v\}$ induit un sous-graphe biparti de H. Soient u_1 et u_2 deux sommets distincts de C_{i_0}. Par conséquent, $ax^{B_0 \backslash \{u_1\}} = ax^{B_0 \backslash \{u_2\}} = |W| - (\frac{p}{2}) = \alpha$ et alors $0 = bx^{B_0 \backslash \{u_1\}} - bx^{B_0 \backslash \{u_2\}} = b(u_1) - b(v_2)$. En posant $\rho_{i_0} = b(u_1)$, on obtient alors $b(v) = \rho_{i_0}$ pour tout sommet $v \in V(C_{i_0})$.
Maintenant, posons $\rho = \rho_1$. Comme toute face de H est adjacente à une autre face de H, on peut déduire que $\rho_i = \rho$, pour tout $i \in \{1, ..., p\}$. Il s'ensuit que $b(v) = \rho$, pour tout v dans W.
Donc, $b = \rho a$. De plus, comme $ax \leq \alpha$ n'est pas triviale, $bx \leq \gamma$ définit une facette non triviale de $P_B(H)$, et ainsi $b \geq 0$. Par conséquent $\rho > 0$, ce qui achève la preuve. □

Nous donnons maintenant une condition nécessaire pour qu'une inégalité cellulaire définisse une facette de $P_B(G)$.

Soient $G = (V, E)$ un graphe et $H = (W, F)$ un sous-graphe p-cellulaire de G. Soit H^* le sous-graphe de G induit par $W \cup N(W)$. Supposons que H^* est de degré maximum trois et que les voisins dans H^* de chaque sommet $u \in N(W)$ appartiennent tous à la même face de H. Soit H_u, pour $u \in N(W)$, le sous-graphe induit par $W \cup \{u\}$. On peut remarquer tout d'abord que H_u est planaire et tout sommet de H_u est de degré ≤ 3.

Théorème 5.2.4 *Si pour tout sommet $u \in N(W)$ de degré trois dans H^*, toutes les faces de H_u sont impaires, alors l'inégalité (5.1) ne définit pas une facette pour $P_B(G)$.*

Preuve. Soit $u \in V \setminus W$ un sommet de $N(W)$ de degré trois tel que toutes les faces de H_u soient impairs. On peut remarquer que H_u est un graphe cellulaire avec exactement $p + 2$ faces qui sont toutes impaires. Alors l'inégalité suivante

$$x(W \cup \{u\}) \leq |W| + 1 - \frac{p+2}{2} = |W| - \frac{p}{2} \tag{5.3}$$

est valide pour $P_B(G)$. Ceci signifie que l'inégalité (5.1) peut être obtenue en sommant l'inégalité (5.3) et l'inégalité triviale $-x(u) \leq 0$. Par conséquent l'inégalité (5.1) ne définit pas de facette pour $P_B(G)$. □

Le théorème suivant donne une condition nécessaire et suffisante pour qu'une inégalité (5.1) associée à un graphe fortement p-cellulaire définisse une facette de $P_B(G)$.

Théorème 5.2.5 *Si H est fortement p-cellulaire, alors l'inégalité (5.1) définit une facette de $P_B(G)$ si et seulement si pour chaque sommet $u \in N(W)$ de degré trois dans H^*, au moins une des faces de H_u est paire.*

Preuve.
($\Rightarrow$) Si pour tout sommet $u \in N(W)$ de degré trois dans H^*, toutes les faces de H_u sont impaires, alors, par le théorème 5.2.4, l'inégalité (5.1) ne définit pas une facette de $P_B(G)$.
($\Leftarrow$) Par le théorème 5.2.3, l'inégalité (5.1) définit une facette de $P_B(H)$, afin de montrer qu'elle définit aussi une facette de $P_B(G)$, il suffit de prouver que, pour tout sommet $u \in V \setminus W$, il existe un ensemble $B \in \beta(G)$, avec $B \cap (V \setminus W) = \{u\}$, qui contient u et dont le vecteur d'incidence satisfait (5.1) à l'égalité. Pour cela, prenons arbitrairement un sommet u de $V \setminus W$. Par hypothèse, tous les voisins de u appartiennent à la même face C de H et, par conséquent, u est contenu dans C. Comme C est impaire, il s'ensuit qu'exactement une des faces de H_u contenant u est impaire. Soit u_1 un voisin de u qui appartient à cette face. Comme H est fortement p-cellulaire, il existe un ensemble de sommets $B_C \subset W$ tels que $|W \setminus B_C| = \frac{p-2}{2}$, $V(C) \setminus B_C = \emptyset$ et $(B_C \setminus \{u_1\}) \in \beta(H)$. Considérons l'ensemble de sommets $B_u = (B_C \setminus \{u_1\}) \cup \{u\}$. Il est clair que B_u induit un sous-graphe biparti de G et son vecteur d'incidence vérifie (5.1) à l'égalité. De plus $B_C \cap (V \setminus W) = \{u\}$. □

En section 5.2.4.3, nous montrons que les chemin-cycles (introduits dans la section 4.7) sont fortement p-cellulaire, le théorème 5.2.5 permet donc de prouver le théorème 4.7.5.

Une question naturelle qui peut se poser est: existe-t-il des graphes p-cellulaires qui ne sont pas fortement p-cellulaires. Aussi, on peut remarquer que, par définition, les seuls graphes 3-cellulaires sont les 3-roues liftées. En fait, nous n'avons pas été capable de produire un graphe p-cellulaire qui ne soit pas fortement p-cellulaire. Ceci nous conduit à la conjecture suivante.

Conjecture 5.2.1 *Un graphe p-cellulaire $H = (W, F)$ est fortement p-cellulaire et, ainsi l'inégalité (5.1) associée à H définit une facette de $P_B(H)$ si et seulement si p est impair.*

Nous définissons $\mathcal{S}$ comme l'ensemble de toutes les subdivisions d'une étoile de degré trois. Un graphe cellulaire est dit *étoile-cellulaire* s'il peut être construit à partir d'un trou impair en appliquant l'opération θ' suivante.

θ': Prendre trois sommets u_1, u_2, u_3 de degré deux sur la face extérieure et un graphe

S de $\mathcal{S}$. Identifier les sommets de degré un de S avec u_1, u_2 et u_3 de façon à ce que le graphe obtenu soit un graphe plan et les deux nouvelles faces intérieures soient impaires.

La classe des graphes étoile-cellulaires possède la propriété suivante.

Lemme 5.2.6 *Un graphe étoile-cellulaire, construit à partir d'un trou impair en répétant $\frac{p-2}{2}$ l'opération θ' est un graphe fortement p-cellulaire.*

Preuve. Soit $H = (W, F)$ un graphe étoile-cellulaire obtenu à partir d'un trou impair en répétant q fois l'opération θ'. Noter que H est $(2q+2)$-cellulaire. Soient $v_1, ..., v_q$ les q sommets de W qui sont utilisés comme le sommet u_2 durant l'une des itérations de l'opération θ'. Alors $\{v_1, ..., v_q\} \cap V(C^*) = \emptyset$ et $W \setminus \{u, v_1, ..., v_q\} \in \beta(H)$ pour tout $v \in V(C^*)$. $\square$

La conjecture 5.2.1 est liée à la conjecture suivante.

Conjecture 5.2.2 *Un graphe p-cellulaire est un graphe étoile-cellulaire.*

En effet, nous avons ce qui suit.

Lemme 5.2.7 *La conjecture 5.2.2 implique la conjecture 5.2.1.*

Preuve. Soit C^* la face exterieure d'un graphe étoile-cellulaire. Par le lemme 5.2.6, on peut remarquer qu'il existe un ensemble de sommets $B_{C^*} \subset W$ tel que $|W \setminus B_{C*}| = \frac{p-2}{2}$, $(V(C^*) \setminus B_{C^*}) = \emptyset$ et $(B_{C^*} \cup \{u\}) \in \beta(H)$, pour tout $u \in V(C^*)$.
Soit H un graphe p-cellulaire. Si la conjecture 5.2.2 est vraie, alors H est étoile-cellulaire, et par la remarque ci-dessus, H est fortement p-cellulaire. Ce qui implique que la conjecture 5.2.1 est aussi vraie. $\square$

5.2.4 Cas particuliers d'inégalités cellulaires

Dans cette section, nous présentons des cas particuliers d'inégalités cellulaires.

5.2.4.1 Inégalités de 1-murs

Un graphe $H = (W, F)$ est appelé un *1-mur d'ordre* q, où q est un entier positif, $q \geq 1$, si H consiste en q trous impairs $C_1, ..., C_q$ tels que

i) $(C_i \cap C_{i+1}) = P_i$ où P_i est une chaîne d'au moins une arête, pour $i = 1, ..., q-1$,
ii) les chaînes $P_1, ..., P_{q-1}$ sont deux à deux sommet-disjointes,
iii) pour $i, j \in \{1, ..., q\}$ tels que $i + 1 < j$, C_i et C_j sont sommet-disjoints.

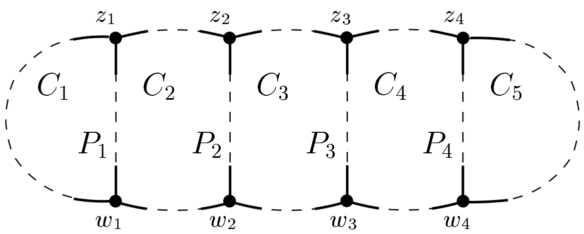


FIG. 5.3 – *Un 1-mur d'ordre 5*

Etant donné un 1-mur $H = (W, F)$ d'ordre q, soit H' le graphe obtenu de H en enlevant tous les sommets intérieurs des chaînes $P_1, ..., P_{q-1}$. Notons que le graphe H' est un trou. On note w_i, z_i les extrémités de la chaîne P_i pour tout $i \in \{1, ..., q-1\}$ de façon à ce que $w_1, ..., w_{q-1}, z_{q-1}, z_{q-2}, , ..., z_1$ soient pris dans cet ordre sur le cycle H'. Si u et v sont deux sommets de H' alors $u \sim v$ représente la sous-chaîne H' entre u et v qui ne contient pas à la fois un sommet de $\{w_1, ..., w_{q-1}\}$ et un sommet de $\{z_1, ..., z_{q-1}\}$.

On peut remarquer qu'un 1-mur d'ordre 1 est un cycle impair. De plus, un cycle impair est le seul 1-mur qui soit p-cellulaire, ce qui permet d'énoncer le résultat suivant

Lemme 5.2.8 *Si H est un 1-mur d'ordre q alors l'inégalité (5.1) est valide pour $P_B(G)$ et elle définit une facette de $P_B(H)$ si et seulement si $q = 1$.*

5.2.4.2 Inégalités de fleurs

Un graphe $H = (W, F)$ est appelé une *fleur* d'ordre q, où q est un entier positif, $q \geq 2$, si H consiste en un 1-mur d'ordre q tel que $P_q = ((C_1 \setminus P_1) \cap (C_q \setminus P_{q-1}))$ soit une chaîne d'au moins une arête. On note alors w_q et z_q les deux extrémités de P_q telles qu'il existe un sous-chaîne de la face extérieure entre z_q et z_1 dont tous les sommets intérieurs sont de degré 2. La figure 5.4 montre une fleur d'ordre 6.

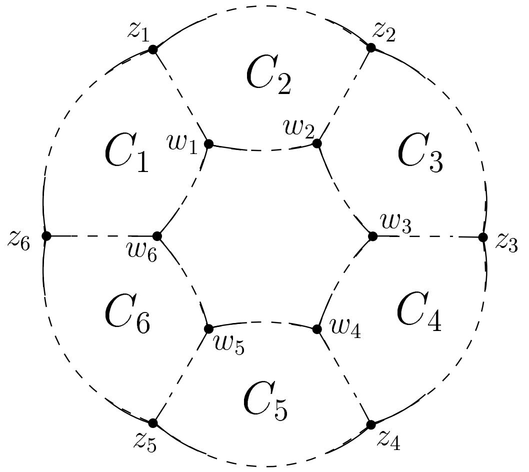


FIG. 5.4 – *Une fleur d'ordre 6*

On peut remarquer qu'une fleur impaire H d'ordre q est un graphe cellulaire contenant $q+2$ faces. Posons C_0 la face intérieure de H distincte de $C_1, ..., C_q$. On dit qu'une fleur est *paire* (*impaire*) si C_0 est pair (*impair*).

Proposition 5.2.9 Une fleur impaire d'ordre q, $q \geq 3$ avec q pair, est fortement q-cellulaire.

Preuve. Soit $H = (W, F)$ une fleur impaire d'ordre q avec q pair. Si $q \geq 3$, alors H est $q+2$-cellulaire. Pour prouver que H est fortement $q+2$-cellulaire, on doit exhiber un sous-ensemble de sommets $B_C \subset W$ de W, pour toute face C de H, tel que $|W \backslash B_C| = \frac{q}{2}$, $V(C) \setminus B_C = \emptyset$ et $(B_C \setminus \{u\}) \in \beta(H)$, pour tout $u \in V(C)$.
Soit C^* la face extérieure de H. Soit $B_{C^*} = W \setminus \{w_1, w_3, ..., w_{q-1}\}$ et $B_{C_0} = W \setminus \{z_1, z_3, ..., z_{q-1}\}$. Pour $i \in \{1, ..., q\}$, on pose également

$$B_{C_i} = \begin{cases} W \setminus \{w_1, w_3, ..., w_{i-3}, z_{i-2}, w_{i+1}, w_{i+3}, w_{i+5}, ..., w_{q-1}\}, \text{si } i \text{ est pair,} \\ W \setminus \{w_1, w_3, ..., w_{i-2}, z_{i+1}, w_{i+2}, w_{i+4}, w_{i+6}, ..., w_{q-1}\}, \text{si } i \text{ est impair.} \end{cases}$$

Pour toute face C de H, $|W \setminus B_C| = \frac{q}{2}$ et $B_C \cup V(C) = \emptyset$. De plus, B_C induit un sous-graphe qui consiste en un cycle avec des chaînes pendantes. Par conséquent, H est fortement q-cellulaire. □

D'après le lemme précédent et le théorème 5.2.3, on a le résultat suivant.

Corollaire 5.2.10 *Soit $H = (W, F)$ une fleur d'ordre q, $q \geq 3$. L'inégalité associée (5.1)*

$$\sum_{u \in W} x(u) \leq |W| - \left\lceil \frac{q+1}{2} \right\rceil \tag{5.4}$$

définit une facette de $P_B(H)$ si et seulement si H est impaire et q est pair.

Preuve. Supposons d'abord que H est paire et q pair, alors l'inégalité (5.4) peut être obtenue en sommant les inégalités de cycles impairs correspondant aux faces $C_1, C_3, ..., C_{q-1}$ et les inégalités triviales des sommets de W privé des sommets des faces $C_1, C_3, ..., C_{q-1}$.
Considérons le cas où H est paire avec q impair. En fait ce cas est aussi celui où H est impaire avec q impair. Dans ce cas, H contient nécessairement un sommet de degré 2, par le lemme 5.2.2 ii), (5.4) ne définit pas de facette de $P(H)$.
Supposons donc que H est impaire avec q impair. Dans ce cas, d'après le théorème 5.2.9, H est fortement $(q+1)$-cellulaire. Ainsi, d'après le théorème 5.2.3, l'inégalité (5.4) définit une facette de $P_B(H)$. □

Comme il a été remarqué, un graphe cellulaire est planaire. Une fleur où l'on échange la définition des sommets w_q et z_q, n'est pas un graphe planaire. Cependant, ce graphe peut être représenté sur *le ruban de Moebius* de façon à ce que deux arêtes ne se croisent pas. Le ruban de Moebius est une surface non-orientable à un seul côté que l'on obtient à partir d'un rectangle en attachant à l'envers deux bords opposés du rectangle. Une fleur, dont on a échangé les définitions des sommets w_q et z_q, peut être représentée sur cette surface de manière à ce que deux arêtes adjacentes ne se croisent pas. Néanmoins si cette surface n'a qu'une seule face, elle n'est pourtant pas planaire.

5.2.4.3 Inégalités de chemin-cycles

Dans la section 4.7, nous avons introduit les inégalités de chemin-cycles qui sont valides pour $P_B(G)$. Nous avons également proposé des conditions nécessaires et suffisantes pour que ces inégalités définissent des facettes pour $P_B(G)$. Dans cette section, nous donnons des preuves alternatives pour ces résultats.
On peut remarquer qu'un chemin-cycle $H = (W, F)$ d'ordre q où q est un entier avec $q \geq 2$ peut être redéfini de la manière suivante: H consiste en un 1-mur H' d'ordre q et une chaîne sans corde P^* joignant un sommet de C_1, disons w_0, à un sommet de C_q, disons w_q, et tel que $w_0 \in V(C_1) \setminus V(P_1)$, $w_q \in V(C_q) \setminus V(P_{q-1})$ et $V(P^*) \setminus \{w_0, w_q\} \cap V(H') = \emptyset$.

Proposition 5.2.11 Un chemin-cycle d'ordre q, $q \geq 2$, avec q pair, est fortement $q+2$-cellulaire.

Preuve. Soit H un chemin-cycle d'ordre q. Si $q \geq 3$, alors H est un q-cellulaire. Pour prouver que H est fortement q-cellulaire, il suffit d'exhiber un ensemble de sommets

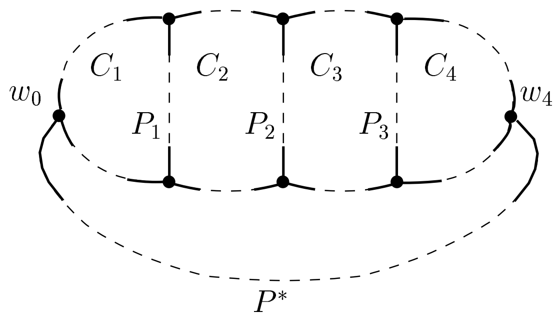


FIG. 5.5 – *Un chemin-cycle d'ordre 4*

$B_C \subset W$, pour toute face C de H, tel que $|W \setminus B_C| = \frac{p-1}{2}$, $V(C) \setminus B_C = \emptyset$ et $(B_C \setminus \{u\}) \in \beta(H)$, pour tout $u \in V(C)$.
Soit C^* la face extérieure de H et C_0 la face intérieure de H distinctes des faces $C_1, ..., C_q$. On pose $B_{C^*} = W \setminus \{w_1, w_3, ..., w_{q-1}\}$ et $B_{C_0} = W \setminus \{z_1, z_3, ..., z_{q-1}\}$. De plus, pour $i \in \{1, ..., q\}$ on note également

$$B_{C_i} = \begin{cases} W \setminus \{w_1, w_3, ..., w_{i-3}, z_{i-2}, w_{i+1}, w_{i+3}, ..., w_{q-1}\}, \text{si } i \text{ est pair,} \\ W \setminus \{w_1, w_3, ..., w_{i-2}, z_{i+1}, w_{i+2}, w_{i+4}, ..., w_{q-1}\}, \text{si } i \text{ est impair.} \end{cases}$$

Pour toute face C de H, $|W \setminus B_C| = \frac{q}{2}$ et $B_C \cup V(C) = \emptyset$. De plus, B_C induit un sous-graphe qui consiste en un cycle avec des chaînes pendantes. Par conséquent, H est fortement q-cellulaire. $\square$

Soient $G = (V, E)$ un graphe et $H = (W, F)$ un chemin-cycle d'ordre q induit de G. Alors l'inégalité cellulaire associée à H n'est rien d'autre que la contrainte (4.11). Comme par le théorème 5.2.11 un chemin-cycle est fortement p-cellulaire, les théorèmes 4.7.3, 4.7.4 et 4.7.5 peuvent être aussi vus comme des conséquences des théorèmes 5.2.3 et 5.2.5.

5.2.4.4 Inégalités de 2-murs

Un graphe $H = (W, F)$ est appelé *2-mur d'ordre* q, où q est un entier , $q \geq 2$, si H contient deux sous-graphes H_1 et H_2 qui sont des 1-murs respectivement d'ordre q et $q-1$ et qui vérifient les propriétés suivantes

i) w_i^2 est un sommet intérieur de $z_i^1 \sim z_{i+1}^1$ pour $i = 0, ..., q-1$,
w_0^2 est un sommet intérieur de $V(C_1^1 \setminus P_1^1) \cap (V(C_1^2 \setminus P_1^2)$,
w_{q-1}^2 est un sommet intérieur de $V(C_q^1 \setminus P_{q-1}^1) \cap V(C_{q-1}^2 \setminus P_{q-2}^2)$,
ii) $(C_i^1 \cap C_i^2)$ est la chaîne $w_{i-1}^2 \sim z_i^1$ de H_1, et $(C_i^2 \cap C_{i+1}^1)$ est la chaîne $z_i^1 \sim w_i^2$ de H_1, pour $i = 1, ..., q$,

iii) et toutes les autres paires de cycles parmi $C_1^1, ..., C_q^1, C_1^2, ..., C_{q-1}^2$ sont sommet-disjoints,

ici C_i^j, P_i^j, w_i^j et z_i^j sont respectivement les faces intérieures C_i, les chaînes P_i et les sommets w_i et z_i du 1-mur H_j, pour $j = 1, 2$ (voir la figure 5.6 qui montre un 2-mur d'ordre 5).

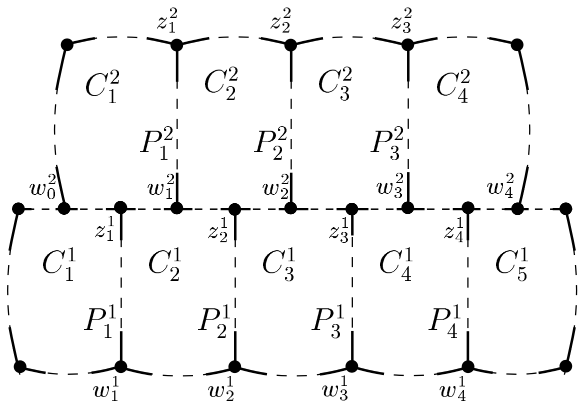


FIG. 5.6 – *Un 2-mur d'ordre 5*

Proposition 5.2.12 Un 2-mur d'ordre q, $q \geq 2$, est fortement $2q$-cellulaire.

Preuve. Soit H un 2-mur d'ordre q. Pour $q \geq 2$, H est $(2q-1)$-cellulaire. Soit C^* la face extérieure de H. Soit $B_{C^*} = W \setminus \{z_1^1, ..., z_{q-1}^1\}$. Posons
$B_{C_i^1} = W \setminus \{w_0^2, ..., w_{i-1}^2, w_{i+1}^2, ..., w_q^2\}$, pour $i \in \{1, ..., q\}$, et,
$B_{C_i^2} = W \setminus \{z_1^1, z_2^1, ..., z_{i-1}^1, w_i^1, z_{i+1}^1, w_{i+2}^1, ..., w_{q-1}^1$, pour $i \in \{1, ..., q-1\}$.
Pour toute face C de H, $|W \setminus B_C| = q - 1$ et $B_C \cup V(C) = \emptyset$. De plus, B_C induit un sous-graphe qui consiste en un cycle avec des chaines pendantes. Par conséquent, H est fortement $(2q-1)$-cellulaire. □

Corollaire 5.2.13 *Soit $H = (W, F)$ un 2-mur d'ordre q, $q \geq 2$. Alors l'inégalité associée (5.1)*

$$\sum_{u \in W} x(u) \leq |W| - (q+1) \tag{5.5}$$

définit une facette de $P_B(H)$.

La procédure de construction des 2-murs peut naturellement être généralisée pour définir des structures appelées p-murs qui consisteraient en un empilement de p 1-murs.

Dans le cadre de l'algorithme de coupes et branchements présenté dans le chapitre 6, nous proposons des heuristiques de séparation pour les 2-murs ainsi que pour les chemins-cycle.

5.3 Séparation d'inégalités cellulaires

Dans cette section, nous démontrons que le problème de séparation peut être résolu en temps polynomial pour deux classes particulières d'inégalités cellulaires.

5.3.1 Séparation des inégalités de fleurs

Dans ce paragraphe, nous montrons que la séparation des inégalités de fleurs peut être réalisée en temps polynomial. La technique de preuve utilisée ici et dans le paragraphe suivant utilise des idées similaires à celles développées par Cheng et Cunningham [26] pour la séparation des inégalités de roues liftées pour le polytope du stable.

On appelle *inégalités de fleurs impaires d'ordre* q les inégalités de fleurs (5.4) dont le graphe induit est une fleur impaire d'ordre q, $q \geq 2$. On appelle *fleur impaire généralisée d'ordre* q un graphe obtenu à partir d'une fleur impaire d'ordre q par identifications de sommets non adjacents. D'après le théorème 4.4.4, une fleur impaire généralisée induit une contrainte valide pour $P_B(G)$. On appelle ces contraintes *inégalités de fleurs impaires généralisées d'ordre* q.

Par commodité d'écriture, on utilise dans cette section la notation suivante: o pour impair, e pour pair et r pour un objet qui peut exister pour les deux parités. On se permet ainsi d'écrire $r \in \{o, e\}$, et d'utiliser les égalités binaires $o + o = e$, $o + e = o$, $e + e = e$ et l'opérateur $opp(r)$ qui donne la parité opposée à r, pour $r \in \{o, e\}$.

Soit $G = (V, E)$ un graphe. Soient $x^* \in I\!R^V$, $\bar{x} = 1\!\!1 - x^*$ et $\omega(e) = \frac{\bar{x}(u)+\bar{x}(v)}{2}$, pour toute arête $e = uv \in E$.

Lemme 5.3.1 *Soit $H = (W, F)$ une fleur impaire (généralisée) d'ordre q avec $q \geq 2$. Alors l'inégalité de fleur (5.2.9) associée à H peut être écrite sous la forme*

$$\sum_{i=1}^{q} \omega(P_i) + \sum_{i=1}^{q} \left(\omega(w_i \sim w_{i+1}) + \omega(z_i \sim z_{i+1})\right) - \frac{1}{2}\left(\sum_{i=1}^{q} \bar{x}(w_i) + \bar{x}(z_i)\right) \geq \frac{q}{2} + 1. \quad (5.6)$$

Preuve. L'inégalité de fleur (5.2.9) peut être écrite sous la forme $\sum_{u \in W} \bar{x}(u) \geq \frac{q}{2} + 1$. D'où (5.6) $\square$

La remarque suivante permet de décrire une fleur d'ordre q à partir des chaînes qui la composent.

Remarque 5.3.2 Soient $z_1, ..., z_q, w_1, ..., w_q$, $2q$ sommets distincts de G. Soient des chaînes élémentaires $P(w_i, z_i)$ (resp. $P(w_i, w_{i+1})$ et $P(z_i, z_{i+1})$) reliant w_i et z_i (resp. w_i et w_{i+1}, et z_i et z_{i+1}), $i \in \{1, ..., q\}$, choisis de telle façon que chacun des cycles formés par les arêtes de $P(w_i, z_i)$, $P(z_i, z_{i+1})$, $P(w_{i+1}, z_{i+1})$ et $P(w_{i+1}, w_i)$ soit impair pour tout $i \in \{1, ..., q\}$ (les indices sont pris modulo q). Alors l'ensemble des chaînes $\{P(w_i, z_i),\ P(w_i, w_{i+1}),\ P(z_i, z_{i+1}), i \in \{1, ..., q\}\}$ forme une fleur d'ordre q.

Pour tout couple de sommets u, v de G, on note par L^e_{uv} (resp. L^o_{uv}) une plus courte chaîne paire (resp. impaire) de G par rapport à ω reliant u et v. Pour simplifier les notations, si, pour un couple de sommets u, v donné dans G, la chaîne L^e_{uv} (resp. L^o_{uv}) n'existe pas dans G, alors L^e_{uv} (resp. L^o_{uv}) désigne une chaîne de taille infinie. On note alors par $\mathcal{P}$ l'ensemble des chaînes L^e_{uv} et L^o_{uv} pour tout couple de sommets distincts u, v de G.

On définit maintenant un graphe auxilliaire $G_P = (V^1 \cup V^2, E_P)$ construit à partir de G de telle manière qu'à chaque élément de $\mathcal{P}$ correspondent un sommet de V^1 et un sommet de V^2. L'ensemble des arêtes E_P est tel que $E_P = E_e \cup E_o$ où E_e et E_o sont définis de la manière suivante. Les arêtes de E_e relient tout couple de sommets de V^1 ou de V^2. Les arêtes de E_o relient un sommet de V^1 et un sommet de V^2 si ces deux sommets ne correspondent pas au même élément de $\mathcal{P}$.

En fait, dans le graphe G_P, une arête de E_P correspondra à un cycle impair de G. Cette correspondance peut être décrite de la manière suivante. Soient deux sommets $t \in V^i$ et $s \in V^{i''}$, correspondant aux éléments de $\mathcal{P}$, respectivement L^r_{uv} et $L^{r'}_{u'v'}$. Alors si $i = i'$, il existe une arête de E_e reliant t et s qui correspondra au cycle composé des chaînes $L^e_{uu'}$, $L^{r'}_{u'v'}$, $L^{opp(r+r')}_{vv'}$ et L^r_{uv}. De la même façon, si $i \neq i'$, il existe une arête de E_o reliant t et s qui correspondra au cycle composé des chaînes $L^o_{uu'}$, $L^{r'}_{u',v'}$, $L^{r+r'}_{vv'}$ et L^r_{uv}. Ces cycles sont impairs et correspondront aux faces intérieures impaires d'une fleur.

Si u, v sont deux sommets de G, on note par $w_o(u, v)$ (resp. $w_e(u, v)$) le poids d'une chaîne impaire (resp. paire) entre u et v de G par rapport au vecteur-poids ω (si une telle chaîne n'existe pas, $w_o(u, v)$ (resp. $w_e(u, v)$) a une valeur infinie). Pour $t \in V^i$ et $s \in V^{i'}$ deux sommets de V_P, $i, i' \in \{1, 2\}$, correspondant respectivement à L^r_{uv} et $L^{r'}_{u'v'}$, soit

$$f(t,s) = \begin{cases} \frac{1}{2}\left(w_r(u,v) + w_{r'}(u',v')\right) + w_e(u,u') + w_{opp(r+r')}(v,v') \\ \qquad -\frac{1}{4}\left(\bar{x}(u) + \bar{x}(v) + \bar{x}(u') + \bar{x}(v')\right) - \frac{1}{2}, \qquad \text{si } i = i', \\ \frac{1}{2}\left(w_r(u,v) + w_{r'}(u',v')\right) + w_o(u,u') + w_{r+r'}(v,v') \\ \qquad -\frac{1}{4}\left(\bar{x}(u) + \bar{x}(v) + \bar{x}(u') + \bar{x}(v')\right) - \frac{1}{2}, \qquad \text{si } i \neq i'. \end{cases}$$

Lemme 5.3.3 *Si x^* satisfait toutes les inégalités triviales et les inégalités de cycles impairs, alors le vecteur-poids f est non négatif.*

Preuve. Soit ts une arête de E_P entre deux sommets $t \in V^i$ et $s \in V^{i'}$ de V_P, $i, i' \in \{1,2\}$. supposons que $i \neq i'$ (le cas où $i = i'$ se prouve de façon similaire). On peut remarquer que si l'une des chaînes $L^r_{uv}, L^{r'}_{vv'}, L^o_{v'u'}$ ou $L^{r+r'}_{u'u}$ n'existe pas, alors un des poids $w_r(u,v)$, $w_{r'}(u',v')$, $w_o(u,u')$, $w_{r+r'}(v,v')$ est l'infini et donc $f(t,s)$ est positif. Supposons donc que les quatres chaînes existent et soit $C = (L^r_{uv}, L^{r'}_{vv'}, L^o_{v'u'}, L^{r+r'}_{u'u})$ le cycle de G formé par ces chaînes. Noter que C est impair. Comme x^* satisfait les inégalités de cycles impairs, on a $\omega(C) \geq 1$, i.e.

$$\omega(L^r_{uv}) + \omega(L^{r'}_{vv'}) + \omega(L^o_{v'u'}) + \omega(L^{r+r'}_{u'u}) \geq 1.$$

D'où

$$\frac{1}{2}\left(w_r(u,v) + w_{r'}(u',v') + w_o(u,u') + w_{r+r'}(v,v')\right) - \frac{1}{2} \geq 0.$$

D'autre part, par définition du poids ω, pour toute chaîne P de E entre deux sommets u_1 et u_2 de G, $\omega(P) - \frac{1}{2}(\bar{x}(u_1) + \bar{x}(u_2)) \geq 0$. Ainsi, on a $\frac{1}{2}\left(w_o(u,u') + w_{r+r'}(v,v')\right) - \frac{1}{4}\left(\bar{x}(u) + \bar{x}(v) + \bar{x}(u') + \bar{x}(v')\right) \geq 0$. Par conséquent, $f(t,s) \geq 0$. □

Lemme 5.3.4 *Soit $H = (W,F)$ une fleur (généralisée) d'ordre q de G avec $q \geq 2$. Pour $j \in \{1,...,q\}$, soit t_j un des deux sommets de V_P correspondant à l'élément $L^{r_j}_{w_j,z_j}$ de $\mathcal{P}$, où r_j est la parité de la chaîne P_j reliant w_j et z_j. Alors l'inégalité de fleur (5.2.9) associée à H peut s'écrire sous la forme*

$$\sum_{j=1}^{q} \Omega(t_j, t_{j+1}) \geq 1, \tag{5.7}$$

où,

$$\Omega(t_j,t_{j+1}) = \frac{1}{2}\left(\omega(P_j) + \omega(P_{j+1})\right) + \omega(w_j \sim w_{j+1}) + \omega(z_j \sim z_{j+1}) - \frac{1}{4}\left(\bar{x}(z_j) + \bar{x}(w_j) + \bar{x}(z_{j+1}) + \bar{x}(w_{j+1})\right) - \frac{1}{2}.$$

De plus, $\displaystyle\sum_{j=1}^{q} \Omega(t_j,t_{j+1}) \geq \sum_{j=1}^{q} f(t_j,t_{j+1})$.

Preuve. L'inégalité (5.7) s'obtient directement du lemme 5.3.1. De plus, par définition, $\omega(P_j) \geq w_{r_j}(w_j, z_j)$. Soit $j_0 \in \{1, ..., q\}$ et supposons que $w_{j_0} \sim w_{j_0+1}$ est pair (le cas où $w_{j_0} \sim w_{j_0+1}$ est impair peut être traité de façon similaire). On a $\omega(w_{j_0} \sim w_{j_0+1}) \geq w_e(w_{j_0}, w_{j_0+1})$ et $\omega(z_{j_0} \sim z_{j_0+1}) \geq w_{opp(r_{j_0}+r_{j_0+1})}(z_{j_0}, v_{z_0+1})$. Par conséquent, $\Omega(t_{j_0}, t_{j_0+1}) \geq f(t_{j_0}, t_{j_0+1})$ □

Nous avons montré par le corollaire 5.2.10, que les inégalités de fleurs impaires d'ordre q définissent des facettes seulement si q est pair. Le théorème suivant s'intéresse à la séparation des inégalités de fleurs impaires d'ordre pair.

Théorème 5.3.5 *La classe des inégalités de fleurs impaires d'ordre pair peut être séparée en $O(|V|^6)$.*

Preuve. Considérons le graphe $G_P = (V^1 \cup V^2, E_P)$ correspondant à G. Pour tout élément L^r_{uv} de $\mathcal{P}$, on calcule dans G_P une chaîne paire de poids minimum, par rapport au vecteur poids f, entre les deux sommets de V_P qui correspondent à L^r_{uv}. Comme par le lemme 5.3.3, f est non négatif, de telles chaînes peuvent être déterminées en temps polynomial. Soit $R^* = (t^*_1 t^*_2, ..., t^*_{q^*} t^*_{q^*+1})$, avec q^* pair, la chaîne minimum parmi toutes ces chaînes. Soit $L^{r_j}_{w^*_j, z^*_j}$ l'élément de $\mathcal{P}$ qui correspond au sommet t^*_j, pour $j \in \{1, ..., q^*\}$. D'après la remarque 5.3.2, la chaîne R^* correspond à une fleur généralisée H^* d'ordre q^*. Si t^*_j et t^*_{j+1}, $j \in \{1, ..., q^*\}$, sont deux sommets de V^1 ou deux sommets de V^2 (resp. un sommet de V^1 et un sommet de V^2), on pose $p_j = e$ (resp. $p_j = o$). Ainsi p_j désigne la parité de la chaîne $w^* \sim w^*_{j+1} = L^{p_j}_{w^*_j w^*_{j+1}}$ de H^*. De plus, H^* est impaire. En effet, par construction du graphe G_P, une chaîne reliant un sommet de V^1 et un sommet de V^2 contient un nombre impair d'arêtes entre des sommets de V^1 et de V^2. Par conséquent, le cycle de G formé par les chaînes $L^{p_j}_{w^*_j w^*_{j+1}}$, pour $j = 1, ..., q^*$, est impair, ce qui implique que H^* est impaire.
L'inégalité de fleur associée à H^* peut être décrite de la façon suivante. Soit $q_j = opp(r_j + r_{j+1})$ (resp. $q_j = r_j + r_{j+1}$), si p_j est paire (resp. impaire) pour $j = \{1, ..., q^*\}$. Alors, par le lemme 5.3.1, l'inégalité

$$\sum_{j=1}^{q^*} \omega(L^{r_j}_{w^*_j z^*_j}) + \sum_{j=1}^{q^*} \left(\omega(L^{p_j}_{w^*_j w^*_{j+1}}) + \omega(L^{q_j}_{z^*_j z^*_{j+1}}) \right) - \frac{1}{2} \left(\sum_{j=1}^{q^*} \bar{x}(w^*_j) + \bar{x}(z^*_j) \right) \geq \frac{q^*}{2} + 1. \quad (5.8)$$

est l'inégalité de fleur généralisée associée à H^*. Si le poids de la chaîne R^* par rapport à f est strictement inférieur à 1, i.e. $\sum_{j=1}^{q^*} f(t^*_j, t^*_{j+1}) < 1$, alors l'inégalité (5.8) est violée par x^*. Dans le cas contraire, toutes les inégalités de fleurs impairs d'ordre pair de G sont satisfaites par x^*. En effet, supposons qu'il existe une telle inégalité qui ne soit pas violée par x^*. Notons par q l'ordre de la fleur induite par l'inégalité et r_j, $j \in \{1, ..., q\}$, la parité de sa chaîne P_j d'extrémités w_j et z_j. Soit t_j, $j \in \{1, ..., q\}$, un des deux sommets de V_P correspondant à l'élément $L^{r_j}_{w_j,z_j}$ de $\mathcal{P}$. Alors par le lemme 5.3.4, on obtient $1 > \sum_{j=1}^{q} \Omega(t_j, t_{j+1}) \geq \sum_{j=1}^{q} f(t_j, t_{j+1})$. Or, par définition, $\sum_{j=1}^{q} f(t_j, t_{j+1}) \geq \sum_{j=1}^{q^*} f(t^*_j, t^*_{j+1})$, ce qui est une contradiction.

L'algorithme présenté ci-dessus consiste à rechercher pour les deux parités et pour tout couple de sommets de G, une chaîne paire dans le graphe G_P. La recherche d'une telle chaîne paire peut être réalisée par l'algorithme de Dijkstra dans un graphe biparti composé de deux copies de G_P. Comme le graphe G_P contient $4|V|^2$ sommets, une telle recherche se fait en $O(|V|^4)$. Par conséquent, l'algorithme est en $O(|V|^6)$. $\square$

5.3.2 Séparation des inégalités de chemin-cycle

On appelle *chemin-cycle généralisé* le graphe obtenu à partir d'un chemin-cycle par des identifications de sommets non adjacents. D'après le lemme 4.4.4, un chemin-cycle généralisé induit une contrainte valide pour le polytope $P_B(G)$. On appelle ces contraintes *inégalités de chemin-cycles généralisés*. Dans cette section, on montre que la classe des inégalités de chemin-cycles généralisés peut être séparée en temps polynomial.

On donne tout d'abord le lemme technique suivant permettant de mettre sous forme appropriée les inégalités de chemin-cycles généralisés.

Lemme 5.3.6 *Soit $H = (W, F)$ un chemin-cycle (généralisé) d'ordre q de G avec $q \geq 2$. L'inégalité (5.2.1) associée à H peut être écrite sous la forme*

$$\sum_{i=1}^{q-1} \omega(P_i) + \sum_{i=0}^{q-1} (\omega(w_i \sim w_{i+1}) + \omega(z_i \sim z_{i+1})) - \frac{1}{2} \sum_{i=1}^{q-1} (\bar{x}(w_i) + \bar{x}(z_i))$$
$$+\omega(P^*) - \frac{1}{2}(\bar{x}(w_0) + \bar{x}(w_q)) \geq \frac{q}{2} + 1. \tag{5.9}$$

Preuve. L'inégalité de chemins-cycle (5.2.9) peut être écrite sous la forme $\sum_{u \in W} \bar{x}(u) + \omega(P^*) - \frac{1}{2}(\bar{x}(w_0) + \bar{x}(w_q)) \geq \frac{q}{2} + 1$. D'où (5.10) $\square$

On appelle également *1-mur généralisé* le graphe obtenu à partir d'un 1-mur par des identifications de sommets non adjacents. Les 1-murs généralisés seront utilisés dans les preuves de cette section. La remarque suivante permet de donner une description d'un 1-mur généralisé à partir des chaînes qui le composent. Cette remarque sera utilisée pour montrer qu'un ensemble de chaînes bien choisi induit un 1-mur généralisé.

Remarque 5.3.7 Soient $w_0, w_1, ..., w_q, z_1, ..., z_{q-1}$ des sommets distincts de G. Pour simplifier l'écriture, posons $z_0 = w_0$ et $z_q = w_q$. Pour $i \in \{0, 1, ..., q-1\}$, soient $P(w_i, z_i)$ (resp. $P(w_i, w_{i+1})$, $P(z_i, z_{i+1})$) une chaîne reliant w_i et z_i (resp. w_i et w_{i+1}, z_i et z_{i+1}), de telle façon que chacun des cycles formés par les arêtes de $P(w_i, z_i)$, $P(z_i, z_{i+1})$, $P(w_{i+1}, z_{i+1})$ et $P(w_{i+1}, w_i)$ soit impair. Alors l'ensemble des chaînes $P(w_i, z_i)$, $P(w_i, w_{i+1})$ et $P(z_i, z_{i+1})$, $i \in \{0, ..., q-1\}$, forme un 1-mur d'ordre q.

Pour deux sommets u, v de V, soit $\mathcal{P}^{uv} = \mathcal{P} \cup \{L^e_{uu}, L^e_{vv}\}$, i.e. $\mathcal{P}^{uv}$ est l'ensemble des chaînes de $\mathcal{P}$ augmenté de deux chaînes vides. Soit G^{uv}_P le graphe obtenu de G_P en ajoutant deux sommets supplémentaires dans V^1 correspondant aux éléments L^e_{uu} et L^e_{vv} de $\mathcal{P}^{uv}$ et en reliant ces sommets à tous sommets de V_1 par des arêtes dans E_e et à tous sommets de V^2 par des arêtesde E_o. Soit f le vecteur sur les arêtes de G^{uv}_P défini de la même façon. On peut montrer comme dans le lemme 5.3.3 que f est non négatif.

Le lemme suivant peut être démontré de manière similaire au lemme 5.3.4.

Lemme 5.3.8 *Soit $H = (W, F)$ un chemin-cycle (généralisé) d'ordre q de G avec $q \geq 2$. Pour $j \in \{1, ..., q\}$, soit t_j un des deux sommets de V_P correspondant à l'élément $L^{r_j}_{w_j,z_j}$ de $\mathcal{P}(w_0, w_q)$, où r_j est la parité de la chaîne P_j reliant w_j et z_j. Soit également t_0 et t_q les sommets de $G^{w_0w_q}_P$ correspondant respectivement aux éléments $L^e(w_0, w_0)$ et $L^e(w_q, w_q)$. Alors l'inégalité de chemin-cycle (généralisé) (5.2.1) associée à H peut s'écrire sous la forme*

$$\sum_{j=0}^{q-1} \Omega(t_j, t_{j+1}) + \omega(P^*) \geq 1$$

où,

$$\begin{aligned}\Omega(t_j, t_{j+1}) = \; & \tfrac{1}{2}\left(\omega(P_j) + \omega(P_{j+1})\right) + \omega(w_j \sim w_{j+1}) + \omega(z_j \sim z_{j+1}) \\ & -\frac{1}{4}\left(\bar{x}(z_j) + \bar{x}(w_j) + \bar{x}(z_{j+1}) + \bar{x}(w_{j+1})\right) - \frac{1}{2}.\end{aligned}$$

De plus, $\displaystyle\sum_{j=0}^{q-1} \Omega(t_j, t_{j+1}) \geq \sum_{j=0}^{q-1} f(t_j, t_{j+1})$.

Par le théorème 4.7.3, les inégalités de chemins-cycle définissent des facettes pour $P_B(G)$ seulement si p est pair. Le théorème suivant montre que dans ce cas, ces contraintes peuvent être séparées en temps polynomial.

Théorème 5.3.9 *La classe des inégalités de chemin-cycles généralisés d'ordre pair peut être séparée en $O(|V|^6)$.*

Preuve. L'idée de l'algorithme est de rechercher une inégalité de chemin-cycle violée pour tout couple de sommets de G jouant le rôle des sommets w_0 et w_q, s'il y en a une. Pour cela, considérons deux sommets u et v de G. On distingue deux cas suivant que, dans le chemin-cycle généralisé recherché, la chaîne P^* reliant u à v soit paire ou impaire. On note par r cette parité et, par conséquent, la chaîne $u \sim v$ du chemin-cycle recherché doit être de parité $opp(r)$.
Soient t_0 et t_q les sommets de $G_P^{u,v}$ qui correspondent respectivement aux éléments L^e_{uu} et L^e_{vv} de $\mathcal{P}^{uv}$. On calcule dans G_P^{uv} une chaîne de parité $opp(r)$ entre t_0 et t_q qui soit de poids minimum par rapport au vecteur-poids f. Soit $R^r_{uv} = (t_0t_1, ..., t_{q-1}t_q)$, où q est pair, la chaîne de poids minimum selon f parmi tous les chaînes calculées. D'après la remarque 5.3.7, la chaîne R^r_{uv} correspond à un 1-mur généralisé H d'ordre q de faces intérieures, disons $C_1, ..., C_q$. De plus, les sommets u et v sont des sommets intérieurs des chaînes $C_1 \setminus P_1$ et $C_q \setminus P_{q-1}$. D'autre part, par construction de R^r_{uv}, la sous-chaîne $u \sim v$ de H' est de parité $opp(r)$. Si l'on ajoute à H la chaîne L^r_{uv}, alors $H' \cup L^r_{uv}$ est un chemin-cycle généralisé d'ordre pair tel que les sommets u et v correspondent aux sommets w_0 et w_q et la chaîne P^*, c'est-à-dire L^r_{uv}, est de parité r.
Soit H^* le chemin-cycle généralisé correspondant à la plus courte chaîne $R^* = (t^*_0t^*_1, ..., t^*_{q^*-1}t^*_{q^*})$ parmi les chaînes R^r_{uv}, $u, v \in G$ et $r \in \{o, e\}$. Considérons que le chemin-cycle H^* a été obtenu pour la parité r^* et pour les sommets w^*_0 et $w^*_{q^*}$. L'inégalité de chemin-cycle associé à H^* peut être décrite de la façon suivante. Soit $L^{r_j}_{w^*_j, z^*_j}$ l'élément de $\mathcal{P}(w^*_0, w^*_{q^*})$ qui correspond au sommet t^*_j, pour $j \in \{0, ..., q^*\}$. Si t^*_j et t^*_{j+1}, $j \in \{1, ..., q^*-1\}$, sont deux sommets de V^1 ou deux sommets de V^2 (resp. un sommet de V^1 et un sommet de V^2), on pose $p_j = e$ (resp.$p_j = o$). Soit $q_j = opp(r_j + r_{j+1})$ (resp. $q_j = r_j + r_{j+1}$), si p_j est paire (resp. impaire) pour $j = \{1, ..., q^*-1\}$. Alors, par le lemme 5.3.6, la contrainte

$$\sum_{j=1}^{q^*-1} \omega(L^{r_j}_{w^*_j z^*_j}) + \sum_{j=0}^{q^*-1} \Big(\omega(L^{p_j}_{w^*_j w^*_{j+1}}) + \omega(L^{q_j}_{z^*_j z^*_{j+1}})\Big) - \frac{1}{2}\sum_{j=1}^{q^*-1}(\bar{x}(w^*_j) + \bar{x}(z^*_j))$$

$$+\omega(L^{r^*}_{w^*_0 w^*_{q^*}}) - \frac{1}{2}(\bar{x}(w^*_0) + \bar{x}(w^*_{q^*})) \geq \frac{q^*}{2} + 1. \tag{5.10}$$

est l'inégalité de chemin-cycle généralisée associée à H^*. Si $\sum_{j=0}^{q^*-1} f(t_j, t_{j+1}) + \omega(L^{r^*}_{w^*_0 w^*_{q^*}}) - \frac{1}{2}(\bar{x}(w^*_0) + \bar{x}(w^*_{q^*})) < 1$, alors l'inégalité (5.10) est violée par x^*. Dans le cas contraire, toutes les inégalités de chemins-cycles d'ordre pair de G sont satisfaites par x^*. En effet, si l'on suppose qu'il existe une telle inégalité qui ne soit pas violée par x^*, alors, en utilisant le lemme 5.3.8, on obtient facilement une contradiction.

L'algorithme présenté ci-dessus consiste à rechercher pour tout couple u, v de sommets de G et pour les deux parités, une chaîne paire dans G_P^{uv}. La recherche d'une telle chaîne paire peut être réalisée par l'algorithme de Dijkstra dans un graphe biparti composé de deux copies de G_P. Comme le graphe G_P contient $4|V|^2$ sommets, une telle recherche se fait en $O(|V|^4)$. Par conséquent, l'algorithme est en $O(|V|^6)$. □

5.4 conclusion

Dans ce chapitre, nous avons introduit la classe des inégalités cellulaires qui sont valides pour le polytope des sous-graphes bipartis induits. Nous avons donné des conditions nécessaires et suffisantes pour que ces inégalités définissent des facettes pour $P_B(G)$. Pour deux sous-classes de cette famille d'inégalités, nous avons montré que le problème de séparation associé peut être résolu en temps polynomial.

Comme la séparation de certaines sous-classes d'inégalités cellulaires est un problème polynomial, il serait intéressant d'étudier le problème pour la classe des inégalités cellulaires en général. Nous conjecturons que ces inégalités peuvent être en général séparées en temps polynomial.

Les résultats présentés dans ce chapitre et le chapitre précédent nous ont permis de développer un algorithme de coupes et branchements que nous présenterons dans le chapitre suivant. Cet algorithme sera utilisé pour résoudre des instances du problème de Via Minimization contraint sur 2 couches (chapitre 6) et du problème dit d'assemblage SNP d'haplotypes (chapitre 8).

Chapitre 6

Un algorithme de coupes et branchements pour le PSBI

Ce chapitre présente un algorithme de coupes et branchements pour le problème du sous-graphe biparti induit. Notre but est d'apporter une application algorithmique aux résultats polyédraux présentés dans les chapitres 4 et 5.

Nous discutons de certains résultats numériques pour des instances réelles et aléatoires des problèmes du sous-graphe biparti induit et de Via Minimization contraint avec 2 couches.

6.1 Séparation heuristiques des contraintes pour le PSBI

Dans cette section, nous proposons des heuristiques pour séparer des classes d'inégalités valides du polytope des sous-graphes bipartis. Si pour certaines d'entre elles il existe des algorithmes de séparation polynomiaux, notre but est d'avoir des routines de séparation rapides pour notre algorithme de coupes et branchements.

Les algorithmes qui seront présentés s'appliquent à un graphe $G = (V, E)$ et à un poids $x^*(v), v \in V$. Dans ce qui suit, on note $\bar{x}$ le vecteur $\mathbf{1}$-x^*, où $\mathbf{1}= (1, ..., 1)^T$. On note également par G_{x^*} le graphe support de x^*. Tous les algorithmes s'effectuent sur ce graphe support G_{x^*}.

6.1.0.1 Séparation des inégalités de cliques

Notre séparation des inégalités de cliques (4.4) utilise une simple heuristique gloutonne introduite par Nemhauser et Sigismondi [88] pour le problème du stable. L'heuristique fonctionne comme suit. On choisit un sommet, disons v, de poids maximum par rapport à x^* et on pose $K = \{v\}$. On répète alors l'opération suivante

> Déterminer un sommet de poids maximum parmi les sommets universel à K, s'il en existe, et ajouter ce sommet à K.

Si $x^*(K) > 2$ alors l'inégalité de clique correspondant à K est violée.

6.1.0.2 Séparation des inégalités de roues liftées

Dans cette section, on s'intéresse à une séparation efficace des inégalités de p-roues liftées (4.10). Ces inégalités peuvent être séparées en temps polynomial comme il a été montré dans la section 4.5.2. Cependant cet algorithme, qui est en $O(|V|^4)$, nécessite que toutes les inégalités de cycles impairs soient satisfaites. Pour cela, nous avons développé une heuristique de compléxité plus faible. Celle-ci peut être présentée comme suit: on choisit arbitrairement un sommet, disons u, de degré $k \geq 3$, et l sommets $u_1, .., u_l \in N(u)$ avec $l \leq k$ et l impair. Pour tout $i = 1, ..., l$, on recherche la plus courte chaîne P_i (par rapport à $\bar{x}$) entre u_i et u_{i+1} tel que $u \notin V(P_i)$ et que tout sommet de P_i apparaît dans au plus une des chaînes $P_1, ..., P_{i-1}, P_{i+1}, ..., P_l$. Le but ici est que chacune des chaînes P_i corresponde à un cycle d'une k-roue liftée, et que tout sommet différent de u n'appartienne qu'à au plus deux de ces cycles. On pose alors $W = \bigcup_{i=1,...,l} V(P_i) \cup \{u\}$. Si toutes les chaînes P_i sont élémentaires, alors W induit une k-roue liftée (autrement W pourrait produire une inégalité redondante). Si on a obtenu une k-roue liftée, on teste alors si l'inégalité correspondante est violée.

6.1.0.3 Séparation des inégalités cellulaires

Nous proposons ici deux heuristiques pour séparer deux sous-classes d'inégalités cellulaires en utilisant des idées similaires à celles présentées dans la section précédente.

Tout d'abord, nous présentons une heuristique pour les inégalités de chemin-cycle (4.11). Nous avons montré que cette classe peut être séparée en temps polynomial mais cet algorithme n'est pas assez efficace pour être utilisé dans un algorithme de coupes et branchements pour le problème.

On définit le poids $\omega(e) = \frac{\bar{x}(u)+\bar{x}(v)}{2}$ pour toute arête $e = uv$ de G_{x^*}. On recherche tout d'abord un arbre couvrant de poids minimum T dans G_{x^*} par rapport au poids ω.

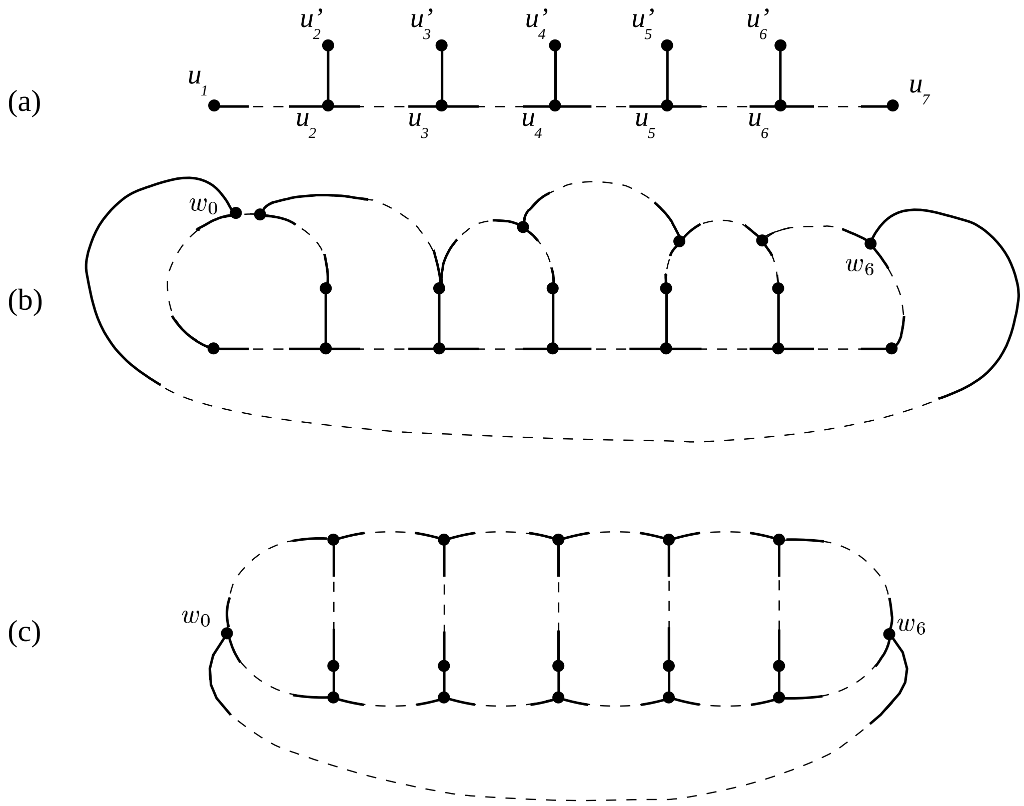


FIG. 6.1 –

On sélectionne alors arbitrairement trois sommets, disons u, v, w de T, et on détermine le sous-arbre T' de T qui couvre ces sommets et qui a deux sommets parmi u, v, w, disons u et w, pour feuilles. On peut remarquer que T' peut prendre deux formes, soit une chaîne avec u et w pour extrémités, soit un arbre ayant les sommets u, v, w pour feuilles. Une fois T' obtenu, s'il contient plus de 5 sommets, on choisit q sommets dans T', $u_1, ..., u_q$, avec $q \geq 5$, de telle façon que $u_1 = u, ..., u_t = v, ..., u_q = w$, q soit impair et que $u_1, ..., u_{t-1}, u_{t+1}, ..., u_q$ soient dans cet ordre sur la chaîne reliant u à w. Pour chaque sommet u_j, $j = 2, ..., q-1$, on prend u'_j un voisin de u_j dans $V \setminus V(T')$ et on pose $u'_1 = u_1, u'_q = u_q$. A cette étape, on obtient un graphe similaire à celui de la figure 6.1 (a) (Ici $q = 7$). On recherche alors une plus courte chaîne P_i entre toute paire de sommets (u'_i, u'_{i+1}), $i = 1, ..., q$ (les indices sont pris modulo q). Ces chaînes sont calculées de telle façon que $P_i \cup T'$ contienne un cycle impair, pour $i = 1, ..., p$ et que deux chaînes P_i et P_j avec $|j - i| > 1$ soient sommet-disjointes. Ainsi on obtient un graphe du même type que celui de la figure 6.1 (b). Maintenant les chaînes P_i avec

T' et l'ensemble d'arêtes $\{u_i u_i', i = 2, ..., q-1\}$ forment un chemin-cycle C d'ordre $q-1$. Ceci est illustré dans la figure 6.1 (c). On peut remarquer que la chaîne P_q correspond à la chaîne P^* du chemin-cycle. Par construction, C' est impair. Le cycle C'' est aussi impair. En effet, l'algorithme produit un graphe planaire ayant q faces intérieures qui sont toutes impaires. De plus, chaque arête est partagée par au plus deux faces. Comme q est impair, la face extérieure, qui est C'', doit être aussi impaire. Si $x^*(C) > |V(C)| - (\frac{q-1}{2}+1)$, alors une inégalité de chemin-cycle violée a été détectée.

Notre deuxième heuristique s'intéresse à un deuxième type d'inégalités cellulaires, que l'on nomme ici arbre-cellulaires. On appelle un graphe *arbre-cellulaire* s'il est cellulaire et si, en ôtant tous les sommets de degré 2 de la face extérieure, on obtient un arbre. Un tel graphe est représenté par la figure 6.2.

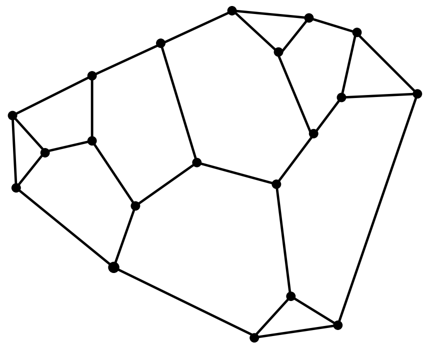


FIG. 6.2 – *Un graphe arbre-cellulaire*

Les graphes arbre-cellulaires peuvent être vus comme une généralisation des chemins-cycle et des 2-murs. La motivation de cette heuristique est d'essayer de trouver un sous-graphe arbre-cellulaire qui contient un ensemble $u_1, ..., u_k$ de sommets fractionnaires par rapport à x^* avec $k \geq 5$. On espère ainsi identifier des inégalités cellulaires violées dans les cas où l'algorithme précédent n'en aurait pas trouvées.

L'heuristique de séparation des inégalités arbre-cellulaires repose sur le même principe que l'heuristique proposée pour la séparation des inégalités de chemin-cycles. On recherche tout d'abord un arbre dont les sommets ont un degré au plus trois. Puis on relie les feuilles de cet arbre par des chaînes de façon à contruire un sous-graphe arbre-cellulaire.

Soit $u_1, ..., u_k$ de sommets fractionnaires par rapport à x^* avec $k \geq 5$. On recherche un arbre, de poids minimum par rapport à ω, qui couvre les sommets $u_1, ..., u_k$ et dont les sommets ont un degré au plus trois. Cet arbre peut être obtenu par l'heuristique gloutonne suivante. On recherche tout d'abord une plus courte chaîne entre u_1 et u_2 par rapport à ω. Soit T cette chaîne. On relie ensuite u_3 à T par une chaîne P de poids minimum par rapport à ω, de manière à ce que $T \cup P$ reste un arbre dont les sommets ont un degré au plus trois. On pose alors $T \leftarrow T \cup P$ et on réitère la procèdure pour les sommets $u_4, ..., u_k$.

Soient $v_1, ..., v_p$ les feuilles de l'arbre T obtenue. On suppose que $v_1, ..., v_p$ sont données dans le sens de la montre dans une représentation planaire de T. Si $p \geq 3$, on peut alors relier les sommets v_i à v_{i+1}, $i \in \{1, ..., p\}$ de manière à former un sous-graphe arbre-cellulaire, c'est-à-dire de façon à ce que toutes les faces du graphe planaire obtenu soient impaires. Si $p = 3$, on aura obtenu une 3-roue liftée. Si $p = 5$, on a obtenu un chemin-cycle d'ordre 4. On teste alors si l'inégalité cellulaire associée au sous-graphe est violée par x^*.

6.2 Description de l'algorithme

6.2.1 Implémentation de l'algorithme

Dans cette section, nous présentons un algorithme de coupes et branchements pour le problème du sous-graphe biparti induit. On considère un graphe $G = (V, E)$ et un vecteur-poids $c \in I\!R_+^V$ associé aux sommets de G. L'algorithme débute en résolvant un programme linéaire de la forme.

$$\begin{array}{ll} \max \sum_{v \in V} c(v)x(v) & \\ 0 \leq x(v) \leq 1, & \text{pour tout } v \in V, \\ x(T) \leq 2, & \text{pout tout } T \in \zeta, \end{array}$$

où ζ est un ensemble de triangles. La solution optimale $x^* \in I\!R^V$ de cette relaxation est réalisable pour le problème si x^* est entier et si l'ensemble de sommets $W = \{u \mid y(u) = 1\}$ induit un sous-graphe biparti de G. Ce test de réalisabilité consiste à construire le graphe G' induit par W puis à effectuer un parcours du graphe G' en donnant alternativement le label 0 ou 1 aux sommets de W. Si, à la fin de ce parcours, tous les sommets adjacents dans G' sont de labels différents, alors G' est bien biparti.

En général, la solution x^* n'est pas réalisable, et alors, à chaque itération de l'algorithme de coupes et branchements, il est nécessaire de générer de nouvelles inégalités valides pour $P_B(G)$ et violées par la solution courante x^*.

La séparation des inégalités est réalisée dans l'ordre suivant.

1. Inégalités de cliques,
2. Inégalités de cycles impairs,
3. Inégalités de roues impaires liftées,

4. Inégalités de chemin-cycles,
5. Inégalités arbre-cellulaires.

On peut remarquer que toutes ces inégalités sont globales (i.e. valides dans tout l'arbre de branchement) et que plusieurs contraintes peuvent être ajoutées à chaque itération. Pour chacune de ces classes d'inégalités, on associe une borne sur le nombre de contraintes qui peuvent être générées à chaque itération.

Certaines inégalités peuvent appartenir à plus d'une classe parmi les classes 1,...,5 (comme par exemple les inégalités de cliques induites par K_4 qui sont aussi des inégalités de roues impaires). Afin de ne pas générer une telle inégalité plus d'une fois dans la même itération, on modifie le graphe pour le reste de l'étape de séparation.

Les algorithmes de séparation utilisés pour les classes d'inégalités 1, 3, 4 et 5 ont été donnés dans la section précédente. Il s'agit en fait d'algorithmes de séparation heuristiques. En revanche, pour séparer les inégalités de cycles impairs, nous avons utilisé l'algorithme de séparation exact décrit dans la section 4.2. Tous les algorithmes de séparation sont effectués sur le graphe support G_{x^*} induit par les sommets u tels que $x^*(u) \neq 0$.

Nous avons également développé une heuristique primale, dite d'arrondissement, dans le but de produite une borne inférieure réalisable. Cette heurisitique peut être présentée de la façon suivante. On calcule tout d'abord un arbre couvrant H de poids maximum par rapport au le vecteur-poids $\omega \in I\!R^E$ défini par $\omega(e) = \max\{y(u)c(u), y(v)c(v)\}$ pour tout arête $e = uv$ de G_{x^*}. L'objectif est d'obtenir une solution ayant un nombre maximum de sommets de poids important. On partitionne alors V en deux ensembles V_1 et V_2 de telle façon que toute arête de H ait une extrémité dans V_1 et l'autre dans V_2. On pose $\widetilde{V} = V$ et $\tilde{v}$ un sommet de $\widetilde{V}$ tel que $x^*(\tilde{v})c(\tilde{v})$ soit minimum. Soit $i \in \{1, 2\}$ tel que $\tilde{v} \in V_i$. Si $\tilde{v}$ a un voisin dans V_i, alors un cycle impair est détecté et, ainsi on pose $\widetilde{V} := \widetilde{V} \setminus \{\tilde{v}\}$. Si le graphe induit par $\widetilde{V}$ ne contient pas de cycle impair, alors on s'arrête, sinon on choisit un sommet $\tilde{v}$ de $\tilde{V}$ et on réitère le processus.

Notre algorithme de coupes et branchements a été programmé en C++ en utilisant ABACUS 2.4 alpha [1] et CPLEX 8 [34]. Il a été testé sur un Pentium IV 1.8 GHz avec 1 GO RAM, sous système Linux.

Pour toutes les instances, nous avons autorisé un temps maximal d'exécution. Si, pour une instance, la limite de temps est atteinte, alors l'heuristique est appliquée pour obtenir une solution approchée.

6.2.2 Résultats expérimentaux pour le PSBI

La table 6.1 rassemble des résultats expérimentaux concernant des instances du problème du sous-graphe biparti induit qui ont été générées aléatoirement. La table est organisée de manière à ce que les dix premiers graphes soient plus denses que les derniers.

Inst.	$\|V\|$	$\|E\|$	Ncl	Nci	Nrl	Ncel	SB	Gap	Copt	CPU
c20a	20	169	21	9	2	0	0	0	20844	0:01
c50a	50	266	340	501	890	258	420	14.57	70026	3:17
c50b	50	1082	353	32	3	0	8	1.56	75123	0:19
c80a	80	2005	43945	138	9	2	2350	*6.47*	*156748*	–*
c100a	100	200	1	326	1631	38	100	2.89	80208	4:19
c100b	100	477	317	3567	11539	131	5388	*12.8*	*119964*	–*
c150a	150	295	1	902	6542	98	642	2.42	122627	61:40
c150b	150	392	16	2025	14351	205	2404	*5.51*	*131266*	–*
n200a	200	299	0	196	276	42	0	0	138427	0:08
c200a	200	347	0	1484	11838	119	2672	*1.83*	*153992*	–*
n300a	300	399	0	426	1397	52	78	0.39	192174	14:12
n500a	500	600	0	442	930	74	34	0.16	291922	15:10
n700a	700	800	0	543	1140	54	122	0.18	380339	56:54
n800a	800	900	0	240	146	33	0	0	428223	0:10
n1000a	1000	1100	0	462	529	79	4	0.03	542411	7:22
n1000b	1000	1200	0	1353	1714	394	166	*1.43*	*551834*	–*

-*: *temps CPU de 5h dépassé*

TAB. 6.1 – *Instances aléatoires du PSBI*

La première colonne de la table 6.1 contient les noms des instances. Les autres entrées sont les suivantes.

$|V|$: Nombre de sommets.
$|E|$: Nombre d'arêtes.
Nci: Nombre d'inégalités de cycles impairs générées.
Nrl: Nombre d'inégalités de roues liftées générées.
Ncel: Nombre d'inégalités cellulaires générées.
SB: Nombre de sous-problèmes générés dans la phase de branchement.

Gap: L'erreur relative, donnée en pourcentage, entre la solution obtenue par notre heuristique (dans le cas où la solution optimale n'a pu être trouvée dans le temps limite) et la borne obtenue par l'algorithme.

Copt: la valeur optimale.

CPU: Temps CPU total en min:sec.

On peut noter que lorsque les graphes deviennent plus denses, le problème devient plus difficile à résoudre. En fait, seulement trois des instances traitées ont été résolues à l'optimalité dans le premier noeud de l'arbre de branchement. De plus, certaines des instances n'ont pas été résolues dans le temps imparti de 5 heures. Comme il est indiqué dans la table, ces instances ont été souvent résolues par notre heuristique primale avec un gap très petit. On peut aussi remarquer que, bien que les inégalités de cycles impairs soient nécessaires à la formulation du problème, elles jouent en fait un rôle mineur. Les inégalités de roues liftées et les inégalités cellulaires semblent très utiles pour cette gamme d'instances. En fait, dans les graphes les plus denses, les inégalités de cycles impairs sont dominées par les inégalités de cliques, les inégalités de roues liftées et les inégalités cellulaires.

6.3 Résultats expérimentaux pour le 2-PVMC

Dans cette section, nous présentons des résultats expérimentaux pour le problème de Via Minimization contraint sur 2 couches. Toutes les instances non aléatoires proviennent d'applications industrielles.

6.3.1 Instances du problème de Via Minimization contraint

S'il existe des banques d'instances pour les problèmes de placement et de routage, à notre connaissance, il n'existe pas de banques de données pour le problème de Via Minimization. Nous avons donc rassemblé plusieurs types d'instances du problème dans le but de tester le mieux possible notre algorithme de coupes et branchements.

Nous avons tout d'abord généré aléatoirement des instances de circuits selon le principe suivant. Au début, on choisit aléatoirement un point du plan qui sera un point terminal d'un réseau. On construit ensuite progressivement le réseau point par point en choisissant aléatoirement le degré de chacun de ces points: degré 1, un point terminal; degré 2, un coude; degré 3 ou 4, une jonction de degré 3 ou 4. Si un bord de la surface est atteint, le point est alors point terminal du réseau. On réitère ensuite ce processus pour un autre réseau en faisant attention à conserver une densité de réseaux raisonnable.

Nous avons également collecté un ensemble d'intances réelles provenant de différentes sources industrielles et universitaires. Ces instances possèdent des degrés de jonction 2, 3 ou 4. Certaines d'entre elles sont originellement routées sur plus de deux couches.

La figure 6.3 présente une instance, nommée a50-7, que nous avons générées aléatoirement. Les nuances de gris représentent les 142 réseaux et on peut remarquer les nombreux croisements que forment ces réseaux.

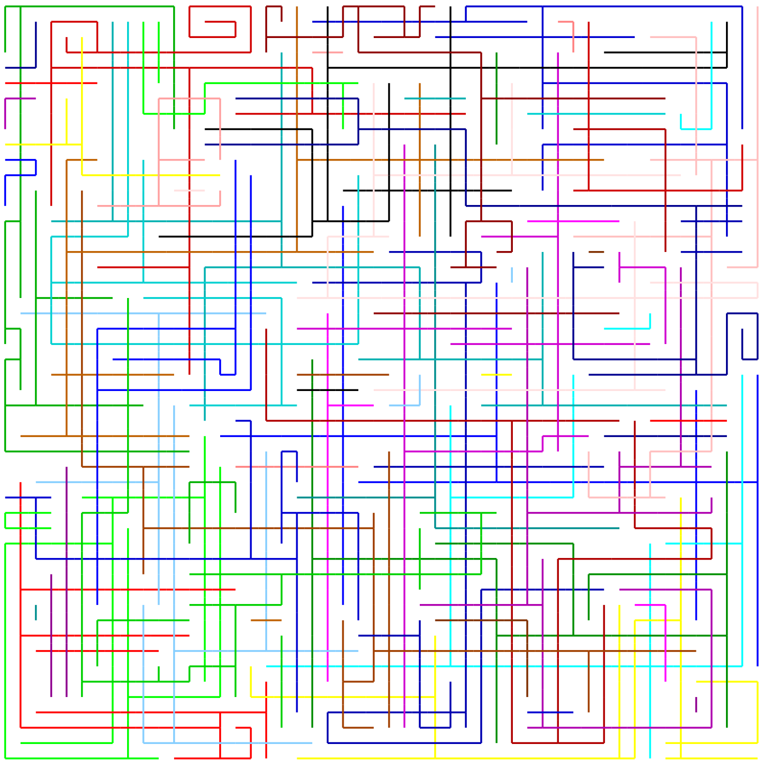


Fig. 6.3 – *L'instance a50-7*

Chaque type d'instances possède des caractériques dépendant de leur provenance. Entre autres, elles n'existent pas originellement sous le même format. Nous avons donc dû implémenter des logiciels de traduction pour ces différents formats (par exemple les formats LEF/DEF [56] ou MAG [100]). Ainsi chacune de nos instances aléatoires ou réelles a pu être encodée sous un même format. Celui-ci consiste en une description de chacun des réseaux sous la forme d'un arbre dont les feuilles sont les points terminaux et les nœuds sont les jonctions. Ce format permet de manipuler facilement les instances du PSBI et leurs solutions.

Afin d'obtenir le graphe de modélisation correspondant à chacune des instances, nous avons implémenté un logiciel repérant tous les chevauchements entre les réseaux. Enfin, nous avons utilisé certaines procédures de réduction de graphes pour accélérer le processus de séparation. La figure 6.4 donne le graphe de modélisation correspondant à l'instance alea50-7.

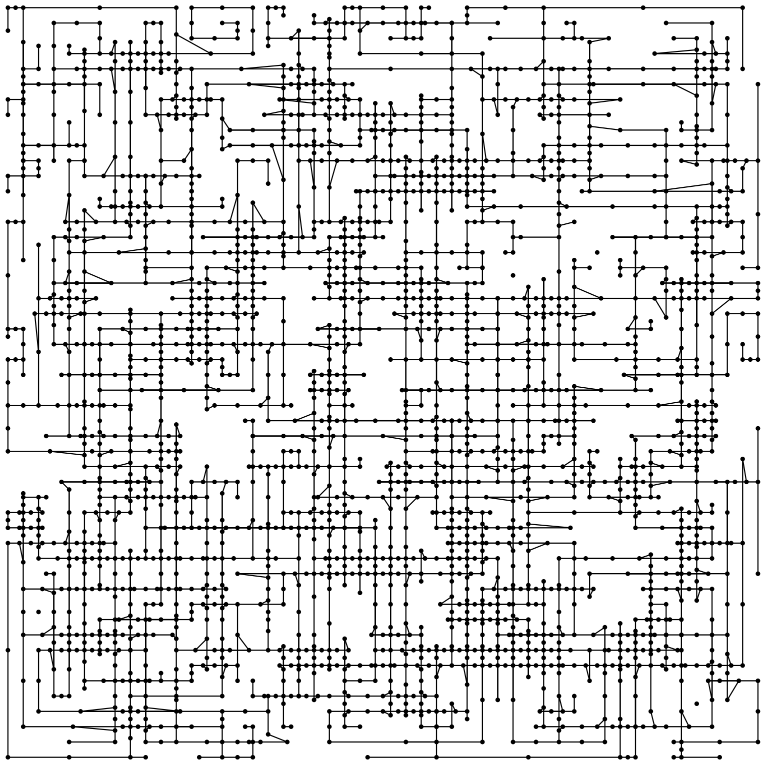


FIG. 6.4 – *Le graphe de modélisation correspondant à l'instance a50-7*

Soit $G = (V, E)$ le graphe de modélisation associé à une instance du 2-PVMC. D'après le théorème 3.2.6, tout sous-graphe biparti induit de poids maximum de G contient tous les sommets-branche. Il est donc suffisant de limiter le programme linéaire aux variables correspondant aux sommets-jonctions. Ainsi le nombre de variables du programme linéaire à résoudre est donné dans le champ J, qui correspond au nombre de jonctions de l'instance. D'autre part, par construction, le graphe de modélisation G ne contient pas de clique de plus de deux sommets. Il est donc inutile d'utiliser l'algorithme de séparation des contraintes de cliques pour ces problèmes-test.

Le graphe de modélisation G obtenu pour une instance du 2-PVMC est très peu dense. En fait, on constate que son nombre d'arêtes ne dépasse pas le double de son nombre de sommets.

6.3.2 Résultats expérimentaux

Dans ce paragraphe, nous présentons certains résultats expérimentaux pour le problème de Via Minimization contraint sur deux couches. Ceux-ci sont donnés dans les tables 6.2 et 6.3. Nous avons utilisé un temps maximum CPU pour l'exécution du programme. Si ce temps est dépassé, une heuristique propose une solution approchée qui est alors indiquée en italique. La première colonne de chaque table donne les noms des instances. Les autres entrées sont les suivantes.

NN: Nombre de réseaux.
J: Nombre de jonctions.
J3: Nombre de jonctions de degré 3.
J4: Nombre de jonctions de degré 4.
$|V|$: Nombre de sommets du graphe de modélisation.
Nci: Nombre d'inégalités de cycles impairs générées.
Nrl: Nombre d'inégalités de roues liftées générées.
Ncel: Nombre d'inégalités cellulaires générées.
SB: Nombre de sous-problèmes générés dans la phase de branchement.
Gap: L'erreur relative, donnée en pourcentage, entre la solution obtenue par notre heuristique (dans le cas où la solution optimale n'a pu être trouvée dans le temps limite) et la borne obtenue par l'algorithme.
NV: Nombre optimal de vias.
CPU: Temps CPU total en min:sec.

Notre première série de problèmes-test concerne des instances générées aléatoirement. Nous avons fixé un temps CPU limite de 5 heures. Les résultats numériques obtenus sont présentés dans la table 6.2.

Inst.	NN	J	J3	J4	$\|V\|$	Nci	Nrl	Ncel	SB	Gap	NV	CPU
a33-9	38	424	47	10	953	441	85	0	0	0	380	3:51
a50-7	71	1173	127	16	2576	4591	1032	452	3	0	127	60:01
a70-147	88	2211	213	23	4769	3457	194	36	0	0	213	17:37
a100-847	142	5648	363	53	11907	4861	286	38	0	0	511	13:26
a150-200	244	13806	758	114	28842	9585	1312	369	0	0	1159	190
a200-147	288	24388	1208	196	50664	14522	1932	156	0	0.6	*2019*	-*

-*: *temps CPU de 5h dépassé*

TAB. 6.2 – *Instances aléatoires du 2-PVMC*

Les instances aléatoires de la table 6.2 semblent assez difficiles à résoudre. En effet, l'algorithme ne peut résoudre que des instances de moins de 250 réseaux. Néanmoins, pour l'instance a200-147 qui contient 388 réseaux, l'algorithme a généré une solution approchée avec un gap inférieur à 0.6%.

Pour ces instances, les graphes de modélisation sont très peu denses et, comme nous l'avons noté pour les graphes générés aléatoirement, les inégalités de cycles impairs sont très importantes. En revanche, l'algorithme a moins utilisé d'inégalités de roues liftées et d'inégalités cellulaires.

La figure 6.5 montre une solution optimale pour l'instance a50-7. Les deux types de traits correspondent aux segments positionnés sur les deux couches. Les cercles noirs repèrent les 127 vias nécessaires.

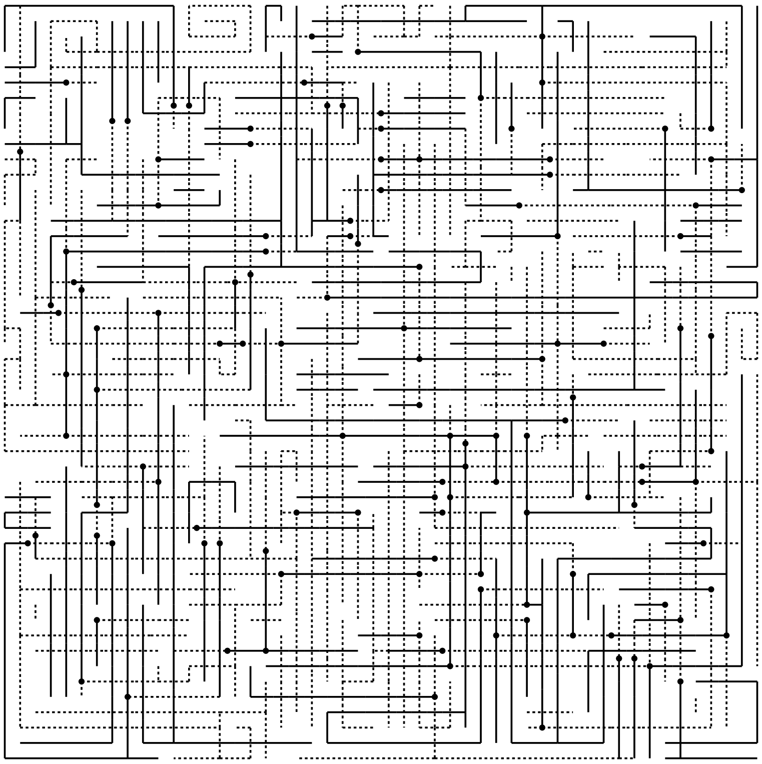


FIG. 6.5 – *Une solution optimale de l'instance a50-7*

Les instances de la deuxième série rassemblées dans la table 6.3 concernent des instances réelles du 2-PVMC.

Les instances c1, c2, c3, c4 et c5 sont des instances de type canal (channel routing) provenant de l'industrie. Ces instances de tailles réduites correspondent à des circuits sur deux couches et ont un degré maximum de jonction limité à 3.

Les instances test1, test2, test3, mcc1 et mcc2 sont issues des travaux de Chang et Cong [22]. Les instances test1, test2, test3 et mcc1 sont sur 4 couches alors que mcc2 est sur 6 couches. Les instances test1, test2 et test3 sont des instances réalistes de tailles croissantes et de degrés limités à 3. Les instances mcc1 et mcc2 correspondent à des circuits intégrés conçu par MCC (Microelectronics and Computer Technology Corporation). Ce sont deux instances de tailles importantes. Elles possèdent quelques jonctions de degré 4.

L'instance p4 nous a été fournie par le laboratoire de Physique Corpusculaire de Clermont-Ferrand [65]. Elle correspond à un circuit de grandes dimensions sur 3 couches conçu par ce laboratoire dans un projet d'analyseur statistique. Elle contient une centaine de jonctions de degré 4.

L'instance ibm01 est issue de la banque de données "ISPD02 IBM-MS Mixed-size Placement Benchmarks" [55] pour les problèmes de placement et de routage. En fait, cette instance a été routée sur 3 couches grace au logiciel Cadence [56]. L'instance obtenue est de très grandes dimensions et possède 70 jonctions de degré 4.

Certaines de ces instances réelles n'ont pu être traitées pour cause de dépassement mémoire. Dans ce cas, nous avons alors traité des instances extraites de plus petites tailles. Ces dernières sont indiquées dans la table en ajoutant une lettre à la fin du nom de l'instance. La figure 6.6 présente l'instance ibm01c extraite de ibm01, elle contient 10000 réseaux au lieu de 19787.

Inst.	NN	J	J3	J4	$\|V\|$	Nci	Nrl	Ncel	SB	Gap	NV	CPU
c1	198	6448	30	0	13124	2743	0	0	0	0	265	1:36
c2	153	4191	23	0	8558	5015	6	0	0	0	228	1:39
c3	300	9947	77	0	20271	10351	0	0	0	0	483	7:03
c4	351	14043	46	0	28483	12803	50	0	0	0	459	13:12
c5	360	11412	50	0	23234	25224	2300	9	0	0	562	38:27
test1	500	50992	0	0	102484	5957	0	0	0	0	976	42:04
test2	957	164091	0	0	329139	12131	0	0	0	0	2142	225:25
test3	1254	261136	0	0	523526	14141	100	0	0	0	2821	388:10
mcc1	907	200909	364	3	403095	32328	9650	0	0	0	2810	854:24
mcc2a	1500	282783	44	0	567110	19364	280	0	0	0	3529	726:42
p4	6914	251452	4773	102	514887	93629	24800	0	0	9.97	*41133*	*48h*
ibm01a	1000	1224	493	1	3943	240	10	0	0	0	72	10:52
ibm01b	5000	32658	2868	17	73216	16470	2519	0	0	0.49	*4173*	*24h*
ibm01c	10000	144839	6187	35	305932	59043	1753	0	0	1.25	*18370*	*24h*

TAB. 6.3 – *Instances réelles du 2-PVMC*

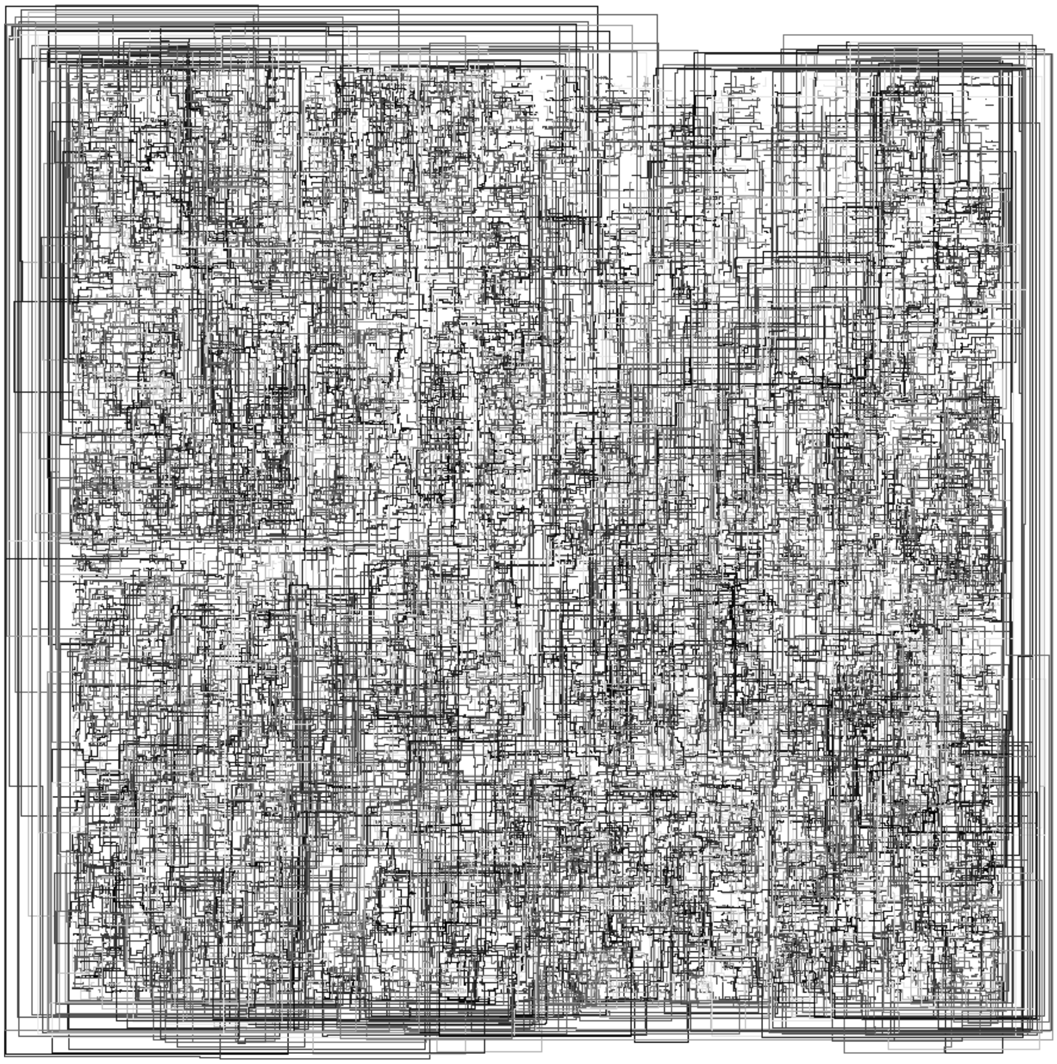


FIG. 6.6 – *L'instance ibm01c*

Pour ces instances réelles, nous n'avons pas fixé de temps CPU limite mais, pour les instances non résolues, nous avons exécuté le programme sur une longue période pré-définie qui est alors indiquée en italique.

Les résultats obtenus pour les instances industrielles de la table 6.3 dépendent fortement de leurs origines industrielles. On peut néanmoins globalement faire les remarques suivantes. L'algorithme n'a pas utilisé de branchements, que les instances soient résolues à l'optimum ou non. D'autre part, quasiment aucune inégalité cellulaire n'a été produite et les inégalités de roues liftées ont été peu utilisées. En fait, le temps imparti (ou la solution optimale) est atteint avant que ces inégalités ne soient davantage sollicitées. Ceci s'explique par le rôle prépondérant que jouent les inégalités de cycles impairs. En effet, elles sont produites en très grandes quantités et, pour les instances non résolues, leur séparation n'est pas finie à la fin du temps imparti.

On peut noter également que l'heuristique primale est très efficace. En effet, pour les instances non résolues, elle donne des solutions approchées avec un gap d'au plus 1.25%.

On peut noter que les instances réelles sont plus faciles à résoudre que les instances générées alétoirement. Pour ces dernières instances, les inégalités de roues liftées et les inégalités cellulaires ont été davantage utilisées.

Les instances c1, c2, c3, c4 et c5 ont été facilement résolus en utilisant principalement les inégalités de cycles impairs. Les instances test1, test2, test3, mcc1 et mcc2a ont également pu être résolues dans un temps CPU maximum de 14 heures. Elles ont nécessité davantage d'inégalités de roues liftées.

L'instance p4 est apparue la plus difficile à traiter. Bien qu'elle ne contienne que 6914 réseaux, elle n'a pas pu être résolue au bout de 48 heures. La meilleure solution obtenue correspond à un gap de 9.97%.

Les problèmes-tests extraits de l'instance ibm01 ont pu être résolu jusqu'à 1000 réseaux en une dizaine d'heures. D'autre part, nous avons pu résoudre une instance possédant 10000 réseaux avec un gap de 1.25%.

6.4 conclusion

Dans ce chapitre, nous avons décrit un algorithme de coupes et branchements pour le problème du sous-graphe biparti induit. Nous avons appliqué cet algorithme à des instances du problème de Via Minimization contraint sur 2 couches. Nous avons pu résoudre exactement des instances industrielles de tailles importantes.

Si notre algorithme a pu résoudre avec un gap très faible toutes les instances testées, nous n'avons pu résoudre les très grandes instances qui correspondent à des graphes de plusieurs millions de sommets. Il serait intéressant d'étudier comment résoudre de telles instances, par exemple en utilisant des opérations de réduction de graphes

Chapitre 7

Un algorithme de génération de colonnes et branchements pour le k-PSPI

Comme il a été montré dans le chapitre 3, le problème de Via Minimization contraint sur k couches se ramène au problème du sous-graphe k-parti induit dans un graphe G^k approprié. Nous nous sommes intéressés à résoudre ce problème afin de déterminer une solution exacte au problème de Via Minimization multicouche.

A notre connaissance, le k-PSPI n'a jamais été traité dans la littérature quand $k \geq 3$. Dans ce chapitre, nous discutons du problème du sous-graphe k-parti induit pour k quelconque. Pour cela, nous donnons une première formulation du problème en le ramenant au problème du stable. Ensuite nous discutons d'une deuxième formulation en nombres entiers pour laquelle nous développons un algorithme de génération de colonnes. Nous appliquons ensuite cet algoritme pour le k-PSPI à des instances du k-PVMC.

7.1 Formulations en programmes en nombres entiers

7.1.1 Formulation 1

Dans ce qui suit, nous montrons que le k-PSPI peut être ramené au problème du stable de poids maximum dans un graphe approprié.

Soient $G = (V, E)$ un graphe et $\bar{c}$ un vecteur-poids associé aux sommets de G. On associe à tout sommet u de V un vecteur en 0-1 de dimension k, $x_u = (x^1(u), ..., x^k(u))$. Le k-PSPI est équivalent au programme linéaire en nombres entiers suivant.

$$\text{Max} \sum_{u \in V^k} \sum_{l=1}^{k} \bar{c}(u) x^l(u)$$

$$\sum_{l=1}^{k} x^l(u) \leq 1, \qquad \text{pour tout } u \in V^k, \tag{7.1}$$

$$x^l(u) + x^l(v) \leq 1, \qquad \text{pour tout } e = uv \in E^k \text{ et } 1 \leq l \leq k, \tag{7.2}$$

$$x^l(u) \in \{0, 1\}, \qquad \text{pour tout } u \in V^k \text{ et } 1 \leq l \leq k. \tag{7.3}$$

On note $W_i = \{u \in V \mid x^i(u) = 1\}$ pour tout $i \in \{1, ..., k\}$. On affirme que les ensembles W_i, $i \in \{1, ..., k\}$ forment une k-partition d'un plus grand sous-graphe k-parti induit de G. En effet, les inégalités (7.1) expriment le fait qu'un sommet u de V ne peut appartenir à plus d'un élément parmi $W_1, ..., W_k$. Et par les inégalités (7.2), deux sommets adjacents ne peuvent appartenir au même élément parmi $W_1, ..., W_k$. Les inégalités (7.3) sont les contraintes d'intégrité. Ainsi le graphe induit par l'ensemble de sommets $W = \bigcup_{i=1}^{n} W_i$ est un sous-graphe k-parti de poids maximum.

On peut remarquer que x_u^l peut être égal à 0 pour un sommet u et pour tout $l = 1, ..., k$. Dans ce cas, le sommet u n'est pas pris dans la solution.

On définit le graphe G_S à partir de G en faisant k copies de G et en reliant par une arête toutes les paires de copies d'un même sommet de G. Le programme en nombres entiers précédent est en fait équivalent au problème du stable de poids maximum dans le graphe G_S.

Ainsi, le k-PSPI peut être résolu en déterminant un stable de poids maximum dans G_S. Comme le problème du stable est NP-difficile, une méthode efficace pour le résoudre est la technique de coupes et branchements [88]. Malheureusement, le graphe G_S, dans lequel est recherché un stable de poids maximum, contient k fois plus de sommets que le graphe G. Cette augmentation de taille ainsi que la structure du graphe G_S rend difficile la résolution d'une instance du k-PSPI.

Récemment, Campêlo, Corrêa et Frota [21] se sont intéressés à une modélisation similaire pour le problème de coloration. Ils proposent un algorithme de coupes et branchements en utilisant une formulation très proche de celle du problème du stable. Mendez-Diaz et Zabala [79] proposent également un algorithme de coupes et branchements pour le problème de coloration. Les auteurs ont ainsi pu résoudre des instances de graphes de quelques centaines de sommets.

7.1.2 Formulation 2

Soit $G = (V, E)$ un graphe et $\bar{c}$ un vecteur-poids associé aux sommets de G. Si G est k-parti alors chaque élément V_i d'une k-partition $(V_1, ..., V_k)$ de G est un stable de G. Cette observation conduit à la formulation présentée ci-dessous.

Soit $\mathcal{S}$ l'ensemble des stables non vides de G. On associe à chaque stable $S \in \mathcal{S}$ une variable binaire t_S. Le k-PSPI est alors équivalent au programme en nombres entiers $(\mathcal{P})$ suivant.

$$\text{Max} \sum_{S \in \mathcal{S}} \bar{c}(S) t_S$$

$$\sum_{S \in \mathcal{S}} t_S = k, \tag{7.4}$$

$$\sum_{S \in \mathcal{S} \ | \ u \in S} t_S \leq 1, \quad \text{pour tout } u \in V, \tag{7.5}$$

$$t_S \in \{0, 1\}, \qquad \text{pour tout } S \in \mathcal{S}. \tag{7.6}$$

En fait, une solution de $(\mathcal{P})$ consiste en un ensemble $\{S_1, ..., S_k\}$ de k stables disjoints deux à deux. Par conséquent, $(S_1, ..., S_k)$ forme une k-partition de $W = \bigcup_{i=1,..,k} S_i$ dans G. De plus, comme $\sum_{S \in \mathcal{S}} c(S) t_S = \sum_{u \in W} c(u)$ est maximum, W induit un sous-graphe k-parti de poids maximum.

7.2 Génération de colonnes

Malheureusement, si la formulation 2 contient $|V| + 1$ contraintes, elle comporte un nombre exponentiel de variables (une pour chaque stable de G). Cette difficulté peut être surmontée en considérant une méthode de génération de colonnes qui permet d'utiliser un nombre réduit de variables. Cette méthode pour le programme $(\mathcal{P})$ peut être décrite comme suit.

1) Trouver un ensemble $J = \{S_1^0, ..., S_k^0\}$ de stables disjoints de G.
2) Trouver une solution optimale $\bar{x}$ de la relaxation linéaire de $(\mathcal{P})$ restreint aux variables associées à J, déterminer la valeur duale $\bar{y}_0$ de l'inégalité (7.4) et la valeur duale $\bar{y}(u)$ de l'inégalité (7.5) correspondant au sommet u, pour tout $u \in V$.
3) Si $r(S) = c(S) - \bar{y}_0 - \sum_{u \in S} \bar{y}(u) \leq 0$ pour tout stable $S \in \mathcal{S} \setminus J$ alors terminé, une solution optimale est trouvée.
4) Sinon ajouter dans J un stable S_0 tel que $r(S_0) > 0$.

5) Aller en 2).

L'étape 3) de la méthode de génération de colonnes est appelée *problème de génération de variables* (*pricing problem*).

Des approches similaires ont été développées pour un problème de gestion forestière [16], pour le problème de coloration [78] et pour un problème d'affectation de fréquences de télécommunications [53].

Pour plus de détails sur les techniques de génération de colonnes, voir le survey de Lübbecke et Desrosiers [73] ainsi que Thienel [101]. Dans [104], Vanderbeck analyse comment les techniques de génération de colonnes peuvent être adaptées efficacement pour les programmes linéaires en nombres entiers.

7.2.1 Résolution du problème de génération de variables

Si S^* est un stable de poids maximum par rapport à la fonction poids $w(u) = c(u) - \bar{y}(u)$, $u \in V$, alors $w(S^*) = \sum_{u \in S^*}(c(u) - \bar{y}(u)) = c(S^*) - \sum_{u \in S^*} \bar{y}(u)$. Par conséquent, le problème de génération de colonnes revient à déterminer un tel stable S^* et tester si $w(S^*) - \bar{y}_0 \leq 0$. Dans ce cas la solution optimale est trouvée, sinon, S^* est ajouté à J.

Ainsi résoudre le problème de génération de variables se ramène à la résolution du programme en nombres entiers $(\mathcal{P}_\mathcal{G})$ suivant.

$$
\begin{aligned}
\text{Max} \sum_{u \in V} & w(u)x(u) \\
x(u) + x(v) \leq 1, & \quad \text{pour tout } uv \in E, \\
x(u) \in \{0, 1\}, & \quad \text{pour tout } u \in V.
\end{aligned}
\tag{7.7}
$$

Le programme $(\mathcal{P}_\mathcal{G})$ est équivalent au problème du stable dans G et peut être résolu de manière efficace par une méthode de coupes et branchements.

On a également implémenté une heuristique pour la génération de variables en s'appuyant sur l'algorithme donné dans [16]. Il s'agit de construire itérativement un stable S de grand poids dans le graphe G selon le poids w. On peut le présenter de la manière suivante: on définit tout d'abord le poids $\omega(u) = w(u) - \lambda \sum_{v \in N(u)} w(v)$ où λ est un entier positif inférieur à 1. On choisit un sommet de plus grand poids selon ω que l'on place dans S, on ôte les voisins de ce sommet puis on recalcule le poids ω pour le nouveau graphe obtenu. On réitère cette opération jusqu'à ce qu'il ne reste plus de sommet dans le graphe.

Le nombre de variables générées peut être très grand. Il est néanmoins possible d'éliminer les variables devenues non actives pour le programme maître. En effet, une variable dont le coût réduit est très négatif peut être temporairement éliminée du programme maître. Afin de ne pas regénérer plusieurs fois les mêmes variables, on stocke les variables devenues non actives dans une structure de données appelée pool. Avant chaque nouvelle génération de colonnes, on parcourt le pool de variables de manière à replacer dans le programme maître les variables de coûts réduits positifs.

7.2.2 Procédure de branchement

La solution optimale de la relaxation linéaire de $(\mathcal{P})$ peut être fractionnaire. Dans ce cas, on considère deux sous-problèmes suivant la stratégie de branchement suivante. Soient u et v deux sommets non-adjacents de G, considérons la contrainte

$$\sum_{S\in\mathcal{S}:\{u,v\}\subset S} t_S \leq 1.$$

Cette contrainte est valide pour le problème. On définit alors deux sous-problèmes $(\mathcal{P}_1)$ et $(\mathcal{P}_2)$ à partir de $(\mathcal{P})$ en ajoutant respectivement les contraintes $\sum_{S\in\mathcal{S}:\{u,v\}\subset S} t_S = 1$ et $\sum_{S\in\mathcal{S}:\{u,v\}\subset S} t_S = 0$. Comme les variables t_S sont binaires, pour tout $S \in \mathcal{S}$, $(\mathcal{P}_1)$ et $(\mathcal{P}_2)$ partitionnent l'espace des solutions de $(\mathcal{P})$. Pour un sous-problème contenant la contrainte $\sum_{S\in\mathcal{S}:\{u,v\}\subset S} t_S = 1$, chacune des variables est associée à un stable qui doit contenir u s'il contient v et inversement. Pour un sous-problème contenant la contrainte $\sum_{S\in\mathcal{S}:\{u,v\}\subset S} t_S = 0$, chacune des variables est associée à un stable qui ne peut contenir à la fois u et v.

Cette règle de branchement est similaire à celle dite de Ryan et Foster [94] qui est utilisée pour le problème de partionnement d'ensembles.

On applique alors la procédure de génération de colonnes pour chacun des sous-problèmes.

Ajouter une telle contrainte de branchement au problème maître entraîne la gestion de nouvelles variables duales dans le problème de génération de colonnes. Plutôt que d'ajouter ces contraintes, nous avons choisi d'appliquer la procédure suivante.

Pour le sous-problème $(\mathcal{P}_1)$, on élimine du programme maître les variables correspondant à des stables contenant uniquement u (resp. v) et non v (resp. u). Pour le sous-problème $(\mathcal{P}_2)$, on élimine du programme maître les variables correspondant à des stables contenant à la fois u er v. On adapte ensuite la procédure de génération de variables présentée dans la section précédente pour qu'elle puisse résoudre le problème de génération de variables associé à ce sous-problème. Pour cela, on ajoute la contrainte

$x_u = x_v$ (resp. $x_u + x_v \leq 1$) au programme $P_{\mathcal{G}}$ associé $(\mathcal{P}_1)$ (resp. $(\mathcal{P}_2)$). Cet ajout de contraintes revient en fait à identifier les sommets u et v dans le graphe G pour la génération de colonnes associée à $(\mathcal{P}_1)$ et à ajouter l'arête uv dans le graphe G pour la génération de colonnes associée à$(\mathcal{P}_2)$.

Nous avons développé une heuristique qui convertit une solution fractionnaire de la relaxation linéaire de $(\mathcal{P})$ en une solution réalisable pour le k-PSPI. Si $\mathcal{S}^*$ est l'ensemble des stables S de la solution fractionnaire tels que $t_S > 0$, l'heuristique consiste à construire k stables $S_1, ..., S_k$ de V en attribuant itérativement les sommets de V à chaque stable. Pour cela, on calcule, pour tout sommet v de V le nombre d'éléments $f(v)$ de $\mathcal{S}^*$ contenant v. De même, on calcule, pour toute paire de sommets u et v dans V, le nombre d'éléments $b(u, v)$ de $\mathcal{S}^*$ contenant à la fois u et v. L'heuristique peut être présentée comme suit.

1) $S_i \leftarrow \emptyset$, pour $i = 1, ..., k$.
   $T \leftarrow V$.
2) Pour tout $i = 1, ..., k$ faire
      Oter le sommet u de T de valeur $f(u)$ maximum.
      $S_i \leftarrow S_i \cup \{u\}$.
      $L \leftarrow T$.
      Tant que $L \neq \emptyset$ faire
         Oter l'élément v de L de valeur $b(u, v)$ maximum.
         Si v non adjacent à un sommet de S_i faire
            Oter v de T.
            $S_i \leftarrow S_i \cup \{v\}$.

7.2.3 Procédure d'ajout de coupes

Le programme linéaire $(\mathcal{P})$ associé au graphe G est en fait une formulation entière proche de celle du problème de stable.

On construit le graphe $G_{\mathcal{S}} = (V_{\mathcal{S}}, E_{\mathcal{S}})$ suivant. Pour chaque variable t_S, $S \in \mathcal{S}$, on associe un sommet u_S dans $V_{\mathcal{S}}$. D'autre part, pour toute contrainte (7.5) où $u \in V$, les sommets $\{u_S \mid S \in \mathcal{S} \text{ avec } u \in S\}$ forment une clique de $G_{\mathcal{S}}$.

En fait, résoudre le programme en nombres entiers $(\mathcal{P})$ revient à rechercher un stable du graphe $G_{\mathcal{S}}$ qui soit de taille k et dont le poids est maximum selon le vecteur-poids $\bar{c}$.

On note alors $P_{stab}(G_{\mathcal{S}})$ le polytope du stable associé au graphe $G_{\mathcal{S}}$. Ainsi, toute inégalité valide pour $P_{stab}(G_{\mathcal{S}})$ peut être ajoutée au programme maître $(\mathcal{P})$. L'ajout de ces inégalités dans le cadre d'une méthode de coupes permet d'améliorer la solution courante. Nous avons considéré uniquement les inégalités de cliques dont font d'ailleurs parties les inégalités (7.5). Nous avons également tenté d'ajouter des inégalités de cycles impairs au programme $(\mathcal{P})$, mais, dans les cas d'instances que nous avons traitées, les seules inégalités de cycles impairs utiles sont les inégalités de triangles qui sont également des inégalités de cliques.

Par construction, le graphe $G_{\mathcal{S}}$ possède un nombre exponentiel de sommets par rapport à $|V|$. Il est néanmoins possible de produire des contraintes de cliques pour $P_{stab}(G_{\mathcal{S}})$ en manipulant simplement le sous-graphe de $G_{\mathcal{S}}$ correspondant aux variables du programme maître courant.

Dans ce sous-graphe, on applique l'heuristique gloutonne décrite par Nemhauser et Sigismondi [88] pour les contraintes de clique du polytope du stable. On ajoute alors les contraintes de cliques générées au programme maître.

Comme ces contraintes ne concernent que des variables déjà générées, la variable duale associée à ces nouvelles contraintes n'entre pas en compte dans le calcul du coût réduit des nouvelles variables. Ainsi le problème de génération de colonnes est inchangé.

L'appel à la procédure d'ajout de coupes est réalisé dès que le problème de génération de colonnes ne trouve plus de nouvelles variables à ajouter et la solution obtenue n'est pas entière.

7.3 Implémentation de l'algorithme

7.3.1 Cadre expérimental

Nous avons implémenté notre algorithme de génération de colonnes et branchements en utilisant le logiciel ABACUS [1]. Il a été testé sur un Pentium IV 2.4 GHz avec 1 GO RAM.

Notre série de problème-tests concerne le problème de Via Minimization sur k couches pour k égal à 2, 3 et 4. En effet, nous avons créé à partir d'instances de circuits VLSI les graphes de modélisation G^2, G^3 et G^4 issus de la modélisation du k-PVMC pour 2, 3 et 4 couches. Nous avons ensuite appliqué notre algorithme de génération de colonnes et branchements à ces instances. Ces résultats sont présentés dans la table 7.1. Les instances concernent des circuits de 5 à 12 réseaux qui ont produit, après transformation, des graphes de 71 à 492 sommets.

Dans notre algorithme de génération de colonnes et branchements, le problème de génération de colonnes consiste à trouver un stable de poids maximum. Cette recherche est réalisée par le solveur en nombres entiers du logiciel CPLEX 7.1 [34]. En effet, vues les tailles réduites des instances, ce logiciel était très efficace (quelques centièmes de secondes) pour déterminer une solution optimale. Dans le cas d'instances de tailles plus importantes, l'utilisation d'un algorithme de coupes et branchements pour le problème du stable serait alors nécessaire.

Une méthode de génération de colonnes doit être initialisée par une solution réalisable. Pour pouvoir initialiser l'algorithme avec une bonne solution, nous avons implémenté une heuristique de type GRASP [60] pour le problème de Via Minimization contraint sur k couches. Une heuristique de type GRASP est une méthode itérative composée de deux étapes. La première consiste à construire une solution réalisable à l'aide d'une méthode gloutonne qui s'appuie sur la dernière solution générée. Cette solution est ensuite améliorée dans une deuxième étape de recherche locale. Ces deux étapes peuvent être répétées jusqu'à ne plus trouver d'amélioration ou atteindre un temps CPU limite.

La méthode gloutonne choisie pour le k-PSPI consiste à affecter les réseaux les uns après les autres sur les k couches en plaçant les vias nécessaires. Nous avons choisi une méthode de type tabou [41, 51] pour la méthode de recherche locale. Une méthode tabou est définie par le voisinage utilisé. Pour notre heuristique, nous avons implémenté le voisinage suivant. Pour une solution donnée du k-PSPI, on choisit aléatoirement une jonction portant un via, on change l'affectation des branches incidentes à ce via de manière à ce qu'elles soient toutes sur la même couche. Ce choix de la couche commune à toutes les branches est fait de manière à provoquer l'ajout d'un nombre minimum de vias.

7.3.2 Résultats expérimentaux pour le k-PVMC

Nous avons utilisé l'algorithme de générations de colonnes et branchements pour résoudre des instances du k-PVMC.

Les résultats expérimentaux sont donnés dans la table 7.1. La première colonne de la table contient les noms des instances. Le nombre contenu dans le nom correspond en fait au nombre de réseaux de l'instance. Les autres entrées sont les suivantes.

$|V|$: Nombre de sommets.
Ncli: Nombre d'inégalités de cliques générées.
Nvar: Nombre de variables générées.
NV: Nombre optimal de vias.
CPU: Temps CPU total en min:sec pour résoudre le problème.

Nous avons fixé un temps maximum CPU de 8 heures. Si ce temps limite est atteint, nous indiquons en italique la meilleure solution obtenue.

Instance	2 couches				
	$\|V\|$	Ncli	Nvar	NV	CPU
t5	49	0	49	2	0:00
t10a	82	0	269	3	0:03
t10b	94	0	846	3	0:05
t11a	167	0	4119	6	0:22
t11b	143	0	7292	2	1:53
t12a	252	0	57501	*7*	-*
t12b	180	0	45426	3	147:58

Instance.	3 couches				
	$\|V\|$	Ncli	Nvar	NV	CPU
t5	71	0	1345	1	0:19
t10a	118	46	5460	1	1:15
t10b	136	0	7361	0	00:29
t11a	245	0	222621	3	180:03
t11b	209	0	110348	0	299:03
t12a	372	0	119874	*4*	-*
t12b	264	0	63231	1	86:04

Instance.	4 couches				
	$\|V\|$	Ncli	Nvar	NV	CPU
t5	93	11	1461	0	0:07
t10a	154	0	0	0	0:09
t10b	178	0	2472	0	00:12
t11a	323	1732	29738	3	387:47
t11b	275	19825	19572	0	77:34
t12a	492	0	105336	*2*	-*
t12b	348	0	85975	*1*	-*

*-: *temps CPU de 8h dépassé*

TAB. 7.1 – *Instances de Via Minimization sur 2, 3 et 4 couches*

Notre algoritme n'a pu résoudre toutes les instances de circuits VLSI. En fait, seules les instances de moins de 11 réseaux ont pu être résolues à l'optimum. Le nombre de variables générées au cours de l'algorithme de génération de colonnes étant très important, la limite de temps de 8h CPU est fréquemment atteinte. Dans ce cas, l'algorithme fournit une solution approchée indiquée en italique.

On peut remarquer que pour toutes les instances résolues, le nombre de vias diminue bien avec le nombre de couches.

Pour cette table d'instances, la procédure de branchement n'a pas été appelée. En fait, l'ajout des inégalités de cliques permet d'éviter l'utilisation de la procédure de branchement pour certaines intances.

7.4 conclusion

Ce chapitre et le chapitre précédent rassemblent nos résultats expérimentaux concernant le problème de Via Minimization contraint sur k couches. Dans ces deux chapitres, nous avons utilisé la modélisation présentée au chapitre 2 qui ramène ce problème au problème du sous-graphe k-parti induit. Pour $k = 2$, notre algorithme de coupes et branchements a pu résoudre des instances réelles de très grandes dimensions. En revanche, lorsque $k = 3$ ou $k = 4$, le problème du sous-graphe k-parti est apparu plus difficile à résoudre et notre algorithme de générations de colonnes et branchements n'a pu résoudre que des instances de tailles réduites.

Dans ce chapitre, nous avons proposé une formulation en termes de programme en nombres entiers pour le problème du sous-graphe k-parti de poids maximum. En utilisant cette formulation, nous avons proposé un algorithme de génération de colonnes et branchements. Ce dernier a été appliqué pour résoudre des instances du problème de Via Minimization contraint multicouche.

Pour pouvoir utiliser l'algorithme de génération de colonnes pour les problèmes de grandes tailles, il serait nécessaire d'apporter certaines améliorations à l'algorithme. Comme la limitation de ses performances est principalement due au nombre important de variables générées, il serait utile de déterminer quelles sont les variables nécessaires au problème et comment limiter la génération de colonnes à ces variables.

Chapitre 8

Le problème d'assemblage SNP d'haplotypes

Dans cette section, nous présentons le problème d'assemblage SNP d'haplotypes qui constitue la dernière étape du séquençage du génome pour les organismes diploïdes. Dans [71], Lippert et al. ont montré que, pour le critère d'enlèvement minimum de fragments, ce problème se ramène au problème du sous-graphe biparti induit de cardinalité maximum. En utilisant cette modélisation, nous donnons des résultats expérimentaux utilisant notre algorithme de coupes et branchements pour le PSBI.

Nous proposons également une modélisation du problème d'assemblage SNP d'haplotypes pour le critère dit du nombre minimum de corrections. Nous montrons que ce problème se ramène lui-aussi à la recherche d'un sous-graphe biparti induit de poids maximum.

8.1 Le problème du séquençage

L'*information génétique* est essentielle à la construction et au fonctionnement des organismes vivants. Elle détermine les caractères d'un être vivant et se transmet des parents à leurs descendants. Cette information se trouve principalement dans le noyau de chacune des cellules et plus particulièrement dans les molécules d'ADN formant les chromosomes.

L'*acide désoxyribonucléique* (ADN) est le support universel de l'information génétique chez les êtres vivants. Il est constitué de deux chaînes enroulées en double hélice. Les deux brins de l'ADN sont l'assemblage de molécules élémentaires : les *nucléotides*. Il existe 4 nucléotides nommés d'après le nom de leurs bases azotées respectives: adénine, guanine, cytosine et thymine. On symbolise ces nucléotides par les lettes A,G,C et T. Ainsi, l'information génétique est codée sous forme chimique dans un alphabet de 4 lettres. Un *gène* est un fragment d'ADN qui correspond à un caractère héréditaire et constitue l'unité d'information génétique. L'ensemble des gènes d'un individu forme le *génome*.

Connaître le génome revient en fait à connaître l'enchaînement des gènes, c'est-à-dire les séquences codantes de l'ADN. On appelle *problème du séquençage* le fait de rechercher et connaître les molécules d'ADN et leurs gènes.

Il est possible de traiter indépendamment chaque molécule d'ADN, c'est-à-dire chaque chromosome. En revanche, il est impossible de connaître directement toute la molécule d'ADN d'un chromosome donné (molécule de plusieurs millions à plusieurs dizaines de millions de nucléotides). Par conséquent, la première étape du séquençage consiste à obtenir des copies partielles de la molécule, appelées des *fragments*. Cette opération est réalisée par des procédés biochimiques et biomécaniques. Elle produit plusieurs millions des fragments d'une longueur de l'ordre de quelques centaines de nucléotides. En fait, ces fragments se chevauchent en possédant deux à deux des parties communes de la molécule.

La seconde étape du séquençage s'appelle le *problème d'assemblage*. Elle consiste à reconstituer la molécule d'ADN à partir des fragments. Cette étape peut se faire par des tâtonnements avec des tests biochimiques, ce qui est très lent. Des méthodes informatiques ont permis de réaliser ce séquençage pour des organismes entiers, dont l'être humain [105]. Ces méthodes consistent à recomposer la molécule en utilisant les chevauchements des fragments. En effet, les fragments qui se chevauchent correspondent à plusieurs copies des mêmes positions de la molécule pour cet assemblage. La plus célèbre des méthodes heuristiques pour cet assemblage est le séquençage aléatoire (shotgun sequencing). En fait, ces méthodes peuvent être divisées en deux phases successives que l'on nommera ici, étape d'alignement et étape de concensus. L'*étape d'alignement* consiste à repérer les positions absolues des fragments sur la molécule. La figure 8.1 montre trois fragments alignés selon leur position sur la molécule d'ADN.

f_1 ATCGATGCATGA
f_2 GATGGATGAAT
f_3 GGATGAATCG

FIG. 8.1 – *Trois fragments alignés*

En fait, parmi les fragments produits, certains peuvent être partiellement erronés. De plus, l'étape informatique d'alignement peut ne pas produire l'alignement exact pour certains fragments. Par conséquent, des fragments alignés peuvent différer sur un ou plusieurs nucléotides. On appelle *étape de concensus* le fait de décider quel nucléotide doit être choisi parmi les nucléotides possibles. En général, ce choix est fait en fonction du nombre de fragments portant le même nucléotide.

Le séquençage du génôme a été permis par l'effort scientifique de chercheurs venant du monde médical et de la biologie mais aussi de l'informatique ou des mathématiques. Le séquençage des organismes a ouvert l'horizon à de nombreuses perspectives de recherche. Dans la section suivante, nous présentons comment un séquençage plus affiné est possible pour les organismes diploïdes.

8.2 Le problème d'assemblage pour les organismes diploïdes

Les organismes diploïdes, comme l'être humain, possèdent en fait deux exemplaires de chaque chromosome, c'est-à-dire deux exemplaires de chaque molécule d'ADN. Chacune de ces copies est appelée un *haplotype*. Ces deux exemplaires sont globalement identiques mais diffèrent parfois de quelques nucléotides, on appelle cette disparité des *polymorphismes génétiques*. Ces derniers sont parfois codants, c'est-à-dire qu'ils forment une manière supplémentaire de coder de l'information. Ils sont généralement dus à un processus naturel, mais peuvent aussi être induits par des agents externes (comme un virus ou des radiations). Dans ce dernier cas, on parle plutôt de mutations génétiques, et leurs détections intéressent la recherche et l'industrie médicale pour le dépistage et le traitement de certaines maladies génétiques. Un gène peut donc se trouver en deux versions alternatives qui diffèrent en séquence, on appelle ces versions des *allèles*.

La forme la plus simple et la plus fréquente de polymorphisme génétique est la variance d'un ou plusieurs nucléotides non contigüs dans une séquence. On appelle chacun de ces polymorphismes portant sur un nucléotide, un *SNP (Single Nucleotid Polymorphism)*. Ils forment 90% des polymorphismes de l'ADN.

On considère connue la liste de fragments ainsi que leurs positions absolues sur la molécule. Ceci permet d'obtenir l'alignement des fragments de manière à faire apparaître les éventuelles différences de nucléotides. Par exemple, la figure 8.2 montre l'alignement de cinq fragments. On appelle ici SNP un emplacement de la molécule où deux fragments diffèrent. La figure 8.2 porte trois SNP.

Pour un organisme diploïde, l'étape d'assemblage doit en fait reconstruire les deux

```
              s1  s2
f1  ATCGATGCATGT
f2   TCGATGCATGAAT     s3
f3      ATGGATGTATGCAA
f4          ATGAATGCATGCA
f5                 CATGCA
```

FIG. 8.2 – *Cinq fragments avec trois SNP*

haplotypes formant une paire de chromosomes. Si on se limite au polymorphisme de type SNP, ce problème s'appelle alors *problème d'assemblage SNP d'haplotypes* (PASH).

En fait, pour un individu donné, un des haplotypes est hérité de la mère et l'autre du père. Néanmoins des récombinaisons par blocs des deux haplotypes peuvent se produire, c'est-à-dire que des échanges de blocs de séquence d'ADN peuvent avoir lieu. Pour cette raison, la possibilité de reconstituer les deux haplotypes a longtemps été ignorée dans le processus du séquençage. Le PASH a été pris en considération à la suite des travaux de Gabriel et al. [39] qui ont montré que les haplotypes d'une même molécule respectent une structure par blocs, et qu'ainsi reconstituer les deux haplotypes avait un sens.

On dispose donc d'un ensemble de fragments issus des deux molécules d'ADN d'une même paire de chromosomes, mais on ignore à quel exemplaire du chromosome chacun de ces fragments appartient. Le PASH revient à regrouper les fragments d'ADN en deux ensembles, chacun correspondant à l'un des deux haplotypes. Malheureusement, lors des procédés biochimiques qui ont permis d'obtenir les fragments, un certain nombre de fragments faux ou partiellement faux, ont été générés. Il ne faut éventuellement pas tenir compte de certaines parties des fragments ou même de certains fragments.

Lippert et al. [71] ont proposé la modélisation suivante pour le PASH.

Soient $\mathcal{F}$ l'ensemble des n fragments et $\mathcal{S}$ l'ensemble des m SNP. On dit que deux fragments f_1 et f_2 sont en conflit si leurs valeurs diffèrent sur un de leurs nucléotides. On suppose que, pour un SNP donné, il n'y a que deux valeurs possibles pour un fragment que l'on notera 0 et 1. Il existe néanmoins une troisième valeur, notée "-", lorsque le fragment ne *couvre* pas le SNP, c'est-à-dire que l'on ignore la valeur de ce nucléotide pour un fragment. On obtient ainsi une *matrice de conflits M* dont les lignes correspondent aux fragments, les colonnes aux SNP et dont les coefficients de la matrice sont 0, 1 ou "-". On dit que la matrice de conflits est *sans erreur* s'il existe une partition des lignes de la matrice en deux matrices qui ne contiennent chacune aucun fragment en conflit. On dit aussi que deux SNP s_1 et s_2 sont en conflit s'il existe deux fragments qui sont en conflit selon s_1 et non selon s_2.

La figure 8.3 donne la matrice de conflits associée aux fragments de la figure 8.2. Comme les fragments 1 et 3 ont des valeurs 0-1 différentes pour le SNP 1, ces deux fragments sont en conflit. On peut remarquer aussi, que les fragments 1 et 2 sont en conflit ainsi que les fragments 1 et 3 et les fragments 1 et 4. Par conséquent, les 5 fragments de l'instance ne peuvent être partitionnés entre deux haplotypes. Ce qui est équivalent à dire que la matrice de la figure 8.3 n'est pas sans erreur. Enfin, on peut noter que les fragments 1 et 3 sont en conflit sur le SNP 1 mais qu'ils ne le sont pas sur le SNP 2. Ce qui correspond au fait que les SNP 1 et 2 sont en conflit.

	s_1	s_2	s_3
f_1	0	1	–
f_2	0	0	–
f_3	1	1	1
f_4	–	0	0
f_5	–	–	0

FIG. 8.3 – *La matrice de conflits correspondant aux fragments de la figure 8.2*

Pour répondre au problème PASH, il faut décider quel critère doit être choisi pour réaliser l'assemblage des haplotypes. Dans Lippert et al. [71], les critères suivants sont proposés:

Enlèvement minimum de fragments (EMF): il consiste à rechercher le plus grand sous-ensemble de fragments qui peut être partitionné de manière à ce que deux fragments d'un même ensemble ne soient en conflit.

Enlèvement minimum de SNP (EMS): il consiste à ôter un nombre minimum de SNP de manière à ce que l'ensemble de tous les fragments puissent être partionné en deux sous-ensembles qui soit chacun sans conflit.

Reconstruction de l'haplotype maximum (RHM): il consiste à rechercher un ensemble de fragments pouvant être partitionné de manière à ce que deux fragments d'un même ensemble ne soient en conflit et que les haplotypes ainsi induits soient de longueurs maximum.

Dans [17], Bonizzoni et al. proposent d'ajouter le critère suivant.

Nombre minimum de corrections (NMC): il consiste à rechercher le nombre minimum de corrections à faire sur les coefficients de la matrice pour obtenir une matrice sans erreur.

Ainsi, le problème d'assemblage SNP d'haplotypes se divise en plusieurs sous-problèmes différents pour lesquels différentes modélisations sont proposées. Nous verrons dans la section suivante les modélisations, proposées par Lippert et al. dans [71], pour les critères EMF et EMS. En revanche, comme aucune modélisation n'a été proposée pour le critère NMC, nous en proposons une modélisation dans la section 8.4.

On dit que deux SNP sont *consécutifs* s'il n'existe pas de SNP situés entre eux sur la molécule. On dit qu'un fragment est *sans lacune* s'il couvre un ensemble de SNP consécutifs. Au contraire, un fragment possède k *lacunes* s'il couvre $k+1$ blocs disjoints de SNP consécutifs. C'est-à-dire qu'un fragment à k lacunes est un fragment composé de k fragments sans lacune qui fonctionnent tous ensemble comme un même fragment. Les cas les plus fréquents sont les cas où $k = 0$ et $k = 1$. Dans [71], il est montré que, dans le cas sans lacune, le PASH avec le critère EMF, EMS ou RHM peut être résolu en temps polynomial. En revanche, pour $k \geq 1$, le PASH avec le critère EMF est NP-difficile. De même, le PASH avec le critère EMS est NP-difficile pour $k = 2$.

Dans [66], Lancia et al. proposent des heuristiques pour traiter quelques uns de ces problèmes. De même, Panconesi et Sozio [90] donnent une heuristique pour le critère EMF et l'utilisent sur des données réalistes. Récemment, Bonizzoni et al. [17] ont étudié des problèmes liés aux problèmes d'assemblage d'haplotypes.

8.3 Modélisation du PASH avec critères EMF et EMS

Dans [71], Lippert et al. proposent deux modélisations pour le problème d'assemblage SNP d'haplotypes avec critère d'enlèvement minimum de fragments et de SNP.

A une instance du PASH, on associe le *graphe de conflits des fragments* $G_F = (V_F, E_F)$ en associant un sommet de V_F à chaque fragment de $\mathcal{F}$ et en liant par une arête deux sommets correspondant à des fragments en conflit. La figure 8.4 donne une matrice de conflits et le graphe G_F correspondant. Cette matrice est associée à une instance de 5 fragments et de 6 SNP. On peut remarquer que cette instance comporte des fragments ayant un nombre variable de lacunes.

	s_1	s_2	s_3	s_4	s_5	s_6
f_1	0	1	–	0	1	0
f_2	0	–	1	–	–	0
f_3	1	0	–	1	1	1
f_4	–	1	1	–	0	–
f_5	1	1	–	1	–	1

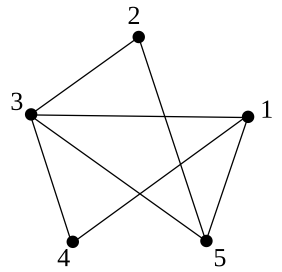


FIG. 8.4 – *Une matrice de conflits et son graphe de conflits des fragments G_F associé*

Les auteurs établissent le résulat suivant pour le PASH avec le critère EMF.

Théorème 8.3.1 [71] *Le problème d'assemblage SNP d'haplotypes avec le critère EMF est équivalent au problème du sous-graphe biparti induit de cardinalité maximum dans le graphe G_F.* □

Ainsi, résoudre le PASH sous critère EMF revient à rechercher un sous-graphe biparti induit de cardinalité maximum dans un graphe dont le nombre de sommets est égal au nombre de fragments. En fait, rechercher un sous-graphe biparti induit de G_F revient à rechercher un ensemble de fragments dont les lignes correspondantes dans la matrice de conflits induisent une matrice sans erreur. On a donc aussi le

Corollaire 8.3.2 *La matrice de conflits M est sans erreur si et seulement si le graphe G_F est biparti.*

Le graphe de conflits des fragments de la figure 8.4 contient par exemple un cycle impair entre les sommets correspondant aux fragments f_1, f_3 et f_4. Ce cycle symbolise le fait que ces fragments sont en conflit deux à deux. Par conséquent, la matrice ne peut pas être sans erreur.

On associe de même à l'instance le *graphe de conflits des SNP* $G_S = (V_S, E_S)$ dont l'ensemble V_S des sommets correspond aux SNP. De plus, il existe une arête v_1v_2 dans E_S si les SNP correspondant à v_1 et v_2 sont en conflit.

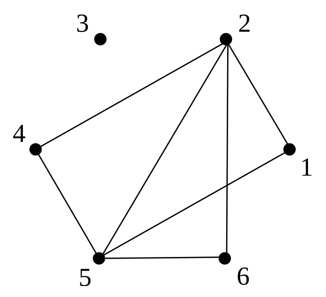


FIG. 8.5 – *Le graphe de conflits des SNP G_S associé à la matrice de la figure 8.4*

Le graphe de conflits des SNP G_S ainsi construit possède la propriété suivante.

Théorème 8.3.3 [71] *G_F est biparti si et seulement si G_S est un stable.* □

Par le théorème précédent, le problème d'assemblage SNP d'haplotypes sous critère EMS revient à rechercher un stable de cardinalité maximum dans le graphe G_S. En effet, en recherchant un tel stable, on détermine un ensemble maximum de SNP tel que la matrice de conflits induite par les colonnes correspondantes soit sans erreur. On peut donc énoncer le

Corollaire 8.3.4 [71] *Le problème d'assemblage SNP d'haplotypes avec le critère EMS est équivalent au problème du stable de cardinalité maximum dans le graphe G_S.* □

Proposée par Lippert et al. [71], cette modélisation du PASH pour les critères EMS et EMF permet de résoudre efficacement des instances réalistes. Dans la section 8.5, nous donnons les résultats obtenus par notre algorithme de coupes et branchements pour le PSBI pour traiter des instances du PASH sous critère EMF.

8.4 Modélisation du PASH avec critère NMC

Nous proposons dans cette section une modélisation pour le problème d'assemblage SNP d'haplotypes avec le critère du nombre minimum de corrections dans la matrice de conflits. On appelle ici *correction* le fait de remplacer la valeur 0 ou 1, par la valeur "-" sur un coefficient de la matrice. En effet, cette correction sur le coefficient d'indices (i, j) permet, en un seul changement de valeurs, d'éviter tous les conflits entre le fragments f_i et tous les autres fragments sur le SNP j. En revanche, une correction qui serait un basculement d'une valeur 0 à la valeur 1 ou inversement, pourrait éventuellement créer d'autres conflits entre les fragments.

On considère une instance du PASH et sa matrice de conflits M associée. On note $m_{i,j}$ les coefficients de la matrice M, $i \in \{1, ..., n\}$ et $j \in \{1, ..., k\}$.

On associe à M le *graphe de corrections* $G_M = (V_M, E_M)$ suivant. L'ensemble des sommets V_M est composé de l'ensemble V_F des sommets associés aux fragments et d'un ensemble V_m de sommets associés aux coefficients de M différents de "-". L'ensemble des arêtes E_M est aussi divisé en deux sous-ensembles E_{Fm} et E_m. Une arête de E_{Fm} lie le sommet correspondant au fragment f_i au sommet correspondant au coefficient $m_{i,j}$, $i \in \{1, ..., n\}$, $j \in \{1, ..., m\}$. Une arête de E_m lie deux sommets correspondant aux coefficients $m_{i,j}$ et $m_{i',j}$ si $m_{i,j} \neq m_{i',j}$, pour $j \in \{1, ..., m\}$ et $i, i' \in \{1, ..., n\}$. Notons que, par construction, le sous-graphe de G_M induit par les sommets de V_m est formé d'un ensemble disjoint de sous-graphes bipartis.

La figure 8.6 donne une matrice de conflits sur trois fragments et trois SNP. On peut remarquer que les trois fragments sont en conflit deux à deux, donc la matrice n'est pas sans erreur. La figure donne également le graphe de corrections associé. Sur ce graphe, les sommets indiqués f_1, f_2, f_3 correspondent aux sommets de V_F et les autres aux coefficients de la matrice. On a positionné ces derniers sommets conformément à la disposition de la matrice. De plus, les arêtes de E_{Fm} entre sommets de V_F et sommets de V_m ont été dessinées en pointillés.

	s_1	s_2	s_3
f_1	1	0	0
f_2	1	1	0
f_3	1	–	1

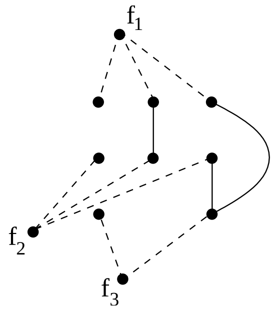


FIG. 8.6 – *Une matrice de conflits et son graphe de corrections G_M associé*

Le graphe G_M a les propriétés suivantes.

Lemme 8.4.1 *Soient deux sommets v et v' de V_m correspondant aux coefficients de la matrice de conflits $m_{i,j}$ et $m_{i',j}$, pour $i, i' \in \{1, ..., n\}$ et $j \in \{1, ..., m\}$.*

i) S'il existe une chaîne impaire entre v et v' comportant uniquement des sommets de V_m alors v et v' sont liés par une arête.

ii) Si v et v' sont tous deux adjacents à un sommet distinct v'' de V_m, alors les fragments f_i et $f_{i'}$ appartiennent au même haplotype.

Preuve. i) Par construction du graphe G_M, s'il existe un telle chaîne entre v et v', alors les valeurs des coefficients correspondant à v et v' sont opposées et ainsi il existe une arête entre v et v'.
ii) Notons $m_{i'',j}$ le coefficient correspondant au sommet v'', $i'' \in \{1, ..., n\}$ et $j \in \{1, ..., m\}$. Comme le sommet v'' est lié aux sommets v et v', le fragment $f_{i''}$ est en conflit avec les fragments f_i et $f_{i'}$. Par conséquent, si les fragments f_i et $f_{i'}$ sont séparés entre les deux haplotypes, alors le fragment $f_{i''}$ ne peut être affecté à aucun des deux haplotypes, une contradiction. □

Lemme 8.4.2 *Soient deux sommets u et u' de V_F associés aux fragments f_i et $f_{i'}$. S'il existe dans G_M une chaîne impaire (resp. paire) liant u et u', alors les fragments f_i et $f_{i'}$ ne peuvent appartenir au même haplotype (resp. doivent appartenir au même haplotype).*

Preuve. Par construction du graphe G_M, une chaîne $P_{ii'}$ liant les sommets u et u' peut s'écrire $(u_1v_1, P_1, v'_1u_1, P_2, ..., u_pv_p, P_p, v'_pu'_p)$ où P_l est une chaîne dont les sommets sont tous dans V_m, u_l est un sommet de V_F, et v_l et v'_l sont des sommets de V_m, pour $l = \{1, .., p\}$. On peut remarquer qu'alors $u = u_1$ et $u' = u_{p+1}$. D'après le lemme 8.4.1 i), si la sous-chaîne de P_l, $l \in \{1, ..., p\}$, liant v_l et v'_l est de longueur $|P_l| \geq 3$, alors il existe une sous-chaîne P'_l liant v_l et v'_l qui soit de longueur 2 si $|P_l|$ est pair, ou de longueur 1 si $[P_l|$ est impair. On note également $P'_l = P_l$ si $|P_l| \leq 3$ et $P'_{ii''}$ la chaîne obtenue depuis $P_{ii''}$ en remplaçant la sous-chaîne P_l par le sous-chaînes P'_l, pour $l = 1, ..., p$. On note alors p_1 (resp. p_2) le nombre de sous-chaînes de $\{P'_l \mid l \in \{1, ..., p\}\}$ qui sont de cardinalité 1 (resp. 2). La chaîne $P'_{ii''}$ a donc pour longueur $2p + p_1 + 2p_2$ et, par conséquent, p_1 est impair si et seulement si $|P_{ii'}|$ est impair. Remarquons que si $|P'_l| = 1$, les fragments correspondant à u_l et u'_l ne peuvent pas appartenir au même haplotype. De plus, d'après le lemme 8.4.1 ii), si $|P'_l| = 2$, alors ces fragments doivent être dans le même haplotype. Si $|P_{ii'}|$ est pair, alors p_1 est pair, et ainsi, d'après les deux remarques précédentes, f_i et $f_{i''}$ doivent appartenir au même haplotype. Inversement, si $|P_{ii'}|$ est impair, alors p_1 est impair, et f_i et $f_{i''}$ ne peuvent appartenir au même haplotype. □

On définit un vecteur poids c sur les sommets de G_M tel que $c(u) = L$, pour tout sommet u de V_F, et $c(u') = 1$, pour tout sommet u' de V_m. Ici L est une valeur suffisamment grande. On a le résulat suivant.

Théorème 8.4.3 *Le problème d'assemblage SNP d'haplotypes avec le critère NMC se ramène au problème du sous-graphe biparti induit de poids maximum dans le graphe G_M muni du vecteur-poids c.* □

Preuve. Soit $G'_M = (V'_M, E(V'_M))$ un sous-graphe biparti induit de G_M qui soit de poids maximum selon le poids c. On veut prouver qu'une solution optimale pour le PASH sous critère NMC peut être déduite de G'_M.
Comme G'_M est maximum selon le poids c, E'_M contient tous les sommets de V_F. En effet, par construction, le sous-graphe de G_M induit par V_F est biparti et la somme des poids de ses sommets est supérieure à la somme des poids des sommets d'un sous-graphe de G_M ne contenant pas tous les sommets de V_F. Par conséquent, tous les sommets de $V_M \setminus V'_M$ sont des sommets de V_m et ainsi, G'_M est un sous-graphe biparti induit de G_M obtenu en ôtant un nombre minimum $|V_M \setminus V'_M|$ de sommets de V_m.
Considérons la matrice de conflits M associée à l'instance. On va corriger la matrice M de la manière suivante. Pour tout sommet de $V_M \setminus V'_M$ associé à un coefficient de M, on effectue la correction de ce coefficient en changeant la valeur 0 ou 1 de $m_{i,j}$ en la valeur "-". Par cette opération, on effectue ainsi $|V_M \setminus V'_M|$ corrections sur

M. La matrice M' obtenue par ces corrections est sans erreur. En effet, supposons le contraire. Tout d'abord, notons que, d'après le théorème 8.3.2, le graphe de conflits sur les fragments $G'_F = (V'_F, E'_F)$ associé à la matrice M' n'est donc pas biparti. Soit $C = (u_1u_2, u_2u_3, ..., u_{p-1}u_p)$ un cycle impair de $\widetilde{G}_F$. Remarquons que les sommets de $\widetilde{G}_F$ correspondent à des fragments et sont des sommets de V_F, donc de G'_M. Les fragments correspondant à u_i et u_{i+1} sont par conséquent en conflit, pour $i \in \{1, ..., p\}$ (les indices sont pris modulo p). Par construction, il existe dans G'_M une chaîne $(u_iv_i, v_iv'_i, v'_iu_{i+1})$ où les sommets v_i et v'_i sont dans V_m, pour $i \in \{1, ..., p\}$. Par conséquent, le graphe G'_M contient le cycle $(u_1v_1, v_1v'_1, v'_1u_2, u_2v_2, ..., v_{p-1}u_p, u_pv_p, v_pv'_p, v'_pu_1)$ qui possède donc $3p$ sommets et qui est impair. Comme G'_M est biparti, on obtient une contradiction. En fait, rechercher un sous-graphe biparti partiel de poids minimum de G_M permet d'obtenir une matrice de conflits sans erreur à partir de M en faisant $|V_M \setminus V'_M|$ corrections. Réciproquement, on veut montrer que ν corrections rendant M sans erreur correspondent à un sous-graphe biparti induit de G_M obtenu en ôtant ν sommets de V_m. On pose $\widetilde{V}_m$ l'ensemble des sommets de V_m correspondant aux coefficients de M qui ont été corrigés. Supposons que le graphe $\widetilde{G}_M$ induit par l'ensemble de sommets $\widetilde{V}_M = V_M \setminus \widetilde{V}_m$ contienne un cycle impair $\widetilde{C}$. Par construction de G_M, $\widetilde{C}$ contient au moins deux sommets u et u' de V_F. Donc il existe dans $\widetilde{G}_M$ une chaîne paire et une chaîne impaire reliant u_1 à u_2. Par le lemme 8.4.2, les fragments correspondant à u et à u' doivent être à la fois dans le même haplotype et ne pas y être. Ceci contredit le fait que la matrice M corrigée est sans erreur. □

D'après le théorème précédent, résoudre le problème d'assemblage SNP d'haplotypes en faisant un nombre minimum de corrections sur la matrice de conflits revient à rechercher un sous-graphe biparti induit de poids maximum du graphe de corrections associé à la matrice. En fait, le nombre de sommets ôtés au graphe pour le bipartiser est égal au nombre de corrections nécessaire pour rendre la matrice de conflits sans erreur. Le graphe de corrections de la figure 8.6 contient un unique cycle impair, on peut briser ce cycle en ôtant un des sommets correspondant à un coefficient de la matrice. En mettant le coefficient de la matrice correspondant à la valeur "-", on obtient alors une matrice sans erreur, c'est-à-dire que l'ensemble des fragments peut alors se répartir en deux haplotypes.

8.5 Résultats expérimentaux pour le PASH avec critère EMF

Dans [90], Panconesi et Sozio constatent l'absence d'instances réelles disponibles librement pour les problèmes de séquençage et proposent la méthode suivante pour construire des instances réalistes pour le PASH.

On génére aléatoirement une chaîne de taille n composée aléatoirement de 0 et de 1, puis on recopie cette chaîne en produisant un taux d'erreur 0.2 entre les deux. Ces deux chaînes vont bien entendu correspondre aux deux haplotypes à reconstituer. Pour chacun des deux haplotypes, on casse la chaîne en k fragments contigüs de tailles aléatoires mais proches de n/k. On recommence c fois cette génération de fragments. Enfin, on échange les valeurs 0 et 1 pour chaque valeur de chaque fragment avec une probabilité 0.05.

Dans [90], les auteurs construisent ainsi des instances en utilisant des haplotypes de taille $n = 100$, coupés $c = 20$ fois en $k = 20$ fragments. Ils obtiennent ainsi des instances comportant entre 400 et 800 fragments chacun de tailles entre 3 et 7 caractères.

Nous avons transformé les instances de Panconesi et Sozio en instances pour le PSBI. Les graphes obtenus possèdent tous 400 sommets et un nombre d'arêtes qui dépend du nombre de conflits entre les fragments. Nous avons utilisé notre algorithme de coupes et branchements pour résoudre ces instances. Les résultats numériques sont résumés dans les tables 8.1 et 8.2.

La première colonne des tables contient les noms des instances. Les autres entrées sont les suivantes.

- $|V|$: Nombre de sommets, c'est-à-dire le nombre de fragments.
- n: Taille des haplotypes à reconstituer.
- k: Taille moyenne des fragments.
- $|E|$: Nombre d'arêtes, c'est-à-dire le nombre de conflits entre les fragments.
- Ncl: Nombre d'inégalités de clique générées.
- Nci: Nombre d'inégalités de cycle impair générées.
- Nrl: Nombre d'inégalités de roue liftée générées.
- Ncel: Nombre d'inégalités cellulaires générées.
- SB: Nombre de sous-problèmes générés dans la phase de branchement.
- Gap: L'erreur relative, en pourcentage, entre la solution obtenue par notre heuristique (dans le cas où la solution optimale n'a pu être trouvée dans le temps limite) et la borne obtenue par l'algorithme.

Inst.	$\|V\|$	n	k	$\|E\|$	Ncl	Nci	Nrl	Ncel	SB	Gap	NF	CPU
00b	400	100	5	2083	128	29	48	0	0	0	361	0:02
17b	400	100	5	2111	183	20	34	0	0	0	357	0:01
22b	400	100	5	1997	284	24	66	0	0	0	357	0:02
27b	400	100	5	2006	105	10	15	0	0	0	365	0:01
29b	400	100	5	1939	183	26	33	0	0	0	371	0:01
33b	400	100	5	2161	155	24	54	0	0	0	363	0:02
51b	400	100	5	1771	143	8	15	0	0	0	367	0:01
63b	400	100	5	1942	218	18	51	0	3	0	361	0:03
77b	400	100	5	1387	121	5	27	0	0	0	364	0:01
95b	400	100	5	2118	178	25	51	0	0	0	365	0:02
99b	400	100	5	2100	133	23	25	0	0	0	361	0:01

TAB. 8.1 – *Première série d'instances du PASH*

NF: Solution optimale, c'est-à-dire le nombre de fragments pouvant être conservés pour construire deux haplotypes.

CPU: Temps CPU total en min:sec pour résoudre le problème.

La table 8.1 présente les résultats concernant les instances de Panconesi et Sozio. Comme on peut le remarqer, toutes les instances issues des travaux de Panconesi et Sozio ont été résolues à l'optimalité en quelques secondes. De plus, à l'exception d'une instance, le branchement n'a pas été nécessaire. En fait, les graphes issus de ces instances sont quasiment bipartis. De plus, la taille de ces instances est assez réduite.

Pour ces instances, l'algorithme n'utilise que les contraintes de cycles impairs et de roues liftées. La solution optimale est atteinte avant que les contraintes cellulaires ne soient utilisées.

Dans le but de se rapprocher des dimensions des tailles réelles d'instances du PASH, nous avons à notre tour généré des instances pour des tailles plus importantes. Cette démarche nous permet également de tester l'efficacité de notre algorithme de coupes et branchements. Les instances traitées sont rassemblées dans la table 8.2.

Les instances de la table 8.2 sont rangées selon la taille des haplotypes à reconstituer. La première instance, nommée a, illustre le fait que nous avons généré des instances ayant les mêmes caractéristiques que celles de Panconesi et Sozio. Nous avons choisi des tailles réalistes de fragments d'une centaine d'éléments. Malheureusement, pour des instances possédant beaucoup de conflits, les graphes obtenus possèdent beaucoup d'arêtes. Les instances de tailles supérieures que nous avons produites saturent rapidement la mémoire.

Inst.	$\|V\|$	n	k	$\|E\|$	Ncl	Nci	Nrl	Ncel	SB	Gap	NF	CPU
a	441	100	5	1558	50	20	0	0	0	0	440	0:01
b	21340	800	9	467193	600	164	7	0	0	0	21249	137:96
c	9980	10000	200	1685202	1916	1689	0	3	0	0	1471	26:05
d	20077	10000	100	2887173	3155	2648	4	1	0	0	7572	53:43
e	20077	10000	100	2711641	5165	3399	2	7	0	0	7581	95:49
f	9972	100000	100	2773776	820	265	0	0	0	0	98	32:50
g	11972	100000	100	4016646	803	271	0	0	0	0	98	75:12

TAB. 8.2 – *Deuxième série d'instances du PASH*

Les graphes issus de ces instances sont assez denses, néanmoins elles ont pu être tous résolues en quelques heures. Malgré la tailles importante des graphes, peu de contraintes ont été nécessaires. La structure des graphes issus des instances du PASH est en fait très proches des graphes d'intervalles, ces graphes semblent contenir peu de roues liftées ou de graphes cellulaires. En fait, les inégalités de cliques et de cycles impairs dominent les autres contraintes.

8.6 Conclusion

Dans ce chapitre, nous avons présenté le problème d'assemblage SNP d'haplotypes qui est une étape du processus du séquençage du génome pour les organismes diploïdes. Nous avons discuté des modélisations du problème suivant certains critères. Pour un de ses critères, le problème se ramène à la recherche d'un sous-graphe biparti induit de cardinalité maximum. Nous avons testé notre algorithme de coupes et branchements présenté dans le chapitre 6 pour résoudre des instances réalistes du problème d'assemblage SNP d'haplotypes. Toutes les instances testées ont été résolues par notre algorithme dans des marges de temps réduites et sans branchement, ce qui montre que notre algorithme est efficace pour ce type de problème.

Nous avons aussi proposé une modélisation pour le problème d'assemblage SNP d'haplotypes pour le critère du nombre minimum de corrections. Nous avons ramené également ce problème à la recherche d'un sous-graphe biparti induit de poids maximum dans un graphe approprié.

Conclusion

Dans cette thèse, nous nous sommes d'abord intéressés au problème de Via Minimization contraint sur k couches (k-PVMC). Nous avons montré que ce problème se ramène au problème du sous-graphe k-parti induit.

Nous avons dans un premier temps étudié une approche polyédrale pour le problème du sous-graphe biparti induit (PSBI). Nous avons exhibé plusieurs classes d'inégalités valides pour le polytope $P_B(G)$ associé et nous avons donné des conditions nécessaires et des conditions suffisantes pour que ces inégalités définissent des facettes. Nous avons montré que le problème de séparation pour certaines de ces classes peut être résolu en temps polynomial. Nous avons également proposé des procédures de construction de facettes à partir d'autres facettes. Cette étude faciale nous a permis d'implémenter un algorithme de coupes et branchements pour le PSBI et de résoudre des instances réelles du problème de Via Minimization contraint sur 2 couches.

Nous avons également donné une formulation en termes de programme linéaire en nombres entiers du problème du sous-graphe k-parti induit. Et nous avons développé un algorithme de générations de colonnes et branchements pour le problème. Nous avons utilisé cet algorithme pour résoudre des instances du problème de Via Minimization contraint sur k couches.

Enfin, nous avons utilisé l'algorithme de coupes et branchements pour le PSBI pour résoudre le problème d'assemblage SNP d'haplotypes, PASH, avec le critère d'enlèvement minimum de fragments. Nous avons ainsi pu résoudre des instances réalistes de ce problème. Nous avons aussi montré que, pour le critère dit du nombre minimum de corrections, le problème d'assemblage SNP d'haplotypes se ramène également au PSBI.

Ce travail de recherche ouvre plusieurs perspectives de recherche liées à la fois à la théorie et à l'expérimentation numérique.

Pour pouvoir résoudre des instances du 2-PVMC et du PASH de très grandes dimensions, il serait très intéressant d'améliorer les performances de l'algorithme de coupes et branchements pour le PSBI. Une piste serait d'améliorer l'efficacité des algorithmes de séparation pour les contraintes valides présentées, lorsque les graphes sont de grandes tailles. Pour cela, il serait utile d'étudier des opérations de réductions de graphe afin d'obtenir des graphes réduits sur lesquels la séparation d'inégalités valides serait efficace. En particulier, l'étude des points extrêmes de $P_B(G)$ lors de ces opérations de réductions permettrait d'évaluer l'efficacité des procédures de séparation sur les graphes réduits.

Aussi, comme la séparation de certaines sous-classes d'inégalités cellulaires est un problème polynomial, il serait intéressant d'étudier le problème de séparation pour la classe des inégalités cellulaires en général. Nous conjecturons que ces inégalités peuvent être séparées en temps polynomial.

Une question naturelle qui peut se poser est de caractériser les graphes dans lesquels le polytope $P_B(G)$ est décrit par les contraintes triviales et les contraintes de cycles impairs. Dans cette classe, le problème du sous-graphe biparti induit peut être résolu en temps polynomial. Cette caractérisation peut permettre d'identifier de nouvelles classes de contraintes valides. Il serait également intéressant d'étudier le polytope $P_B(G)$ pour les graphes cellulaires. D'autres familles de contraintes seraient nécessaires, en plus des contraintes cellulaires, pour décrire le polytope $P_B(G)$ dans cette classe de graphes. Aussi, une question qui mérite également d'être étudiée est la cacaractérisation de $P_B(G)$ dans les graphes d'intervalles et dans les graphes série-parallèles. Dans [15], Barahona et Mahjoub ont donné un algorithme polynomial basé sur des techniques de composition pour résoudre le problème dans la classe des graphes série-parallèles. Il serait intéressant d'avoir un algorithme polynomial de coupes pour le problème dans cette classe de graphes.

Une autre problématique consisterait à poursuivre l'étude de notre formulation entière pour le problème du sous-graphe k-parti induit. La performance de l'algorithme de générations de colonnes et branchements basé sur cette formulation dépend principalement du nombre de variables générées. Il serait donc utile de déterminer quelles sont les variables nécessaires au problème et comment limiter la génération de colonnes à ces variables.

D'autre part, notre modélisation du k-PVMC pourrait être utilisée pour produire une bonne heuristique pour le k-PSPI. De plus, en combinant une telle heuristique avec notre algorithme de générations de colonnes, on pourrait alors estimer la précision des solutions approchées.

Annexe

Deuxième preuve pour le théorème 4.7.5

Dans la section 4.7, nous proposons des conditions nécessaires et suffisantes pour que les inégalités de chemin-cycle définissent des facettes de $P_B(G)$. Ce théorème est en fait un cas particuliers du théorème 5.2.5 pour les inégalités cellulaires. Nous proposons dans cette annexe une autre preuve pour le théorème 4.7 qui n'utilise pas le théorème 5.2.5.

Théorème 4.7.5 *Soit H^* le sous-graphe de G induit par $W \cup N(W)$. Supposons que H^* est de degré maximum trois et que les voisins dans H^* de chaque sommet $u \in N(W)$ appartiennent tous à la même face de H. Soit H_u, où $u \in N(W)$, le sous-graphe induit par $W \cup \{u\}$. Alors l'inégalité (4.11) définit une facette de $P_B(G)$ si et seulement si pour chaque sommet $u \in N(W)$ de degré trois dans H^*, au moins une des faces de H_u est paire.*

Preuve. Tout d'abord, remarquons que, pour tout $u \in N(W)$, H_u est planaire et tout sommet de H_u est de degré ≤ 3.
($\Leftarrow$) Comme l'inégalité (4.11) définit une facette de $P_B(H)$, pour prouver qu'elle définisse aussi une facette de $P_B(G)$, il suffit de prouver que pour tout sommet $u \in V \setminus W$, il existe un ensemble de sommets B, avec $B \cap (V \setminus W) = \{u\}$, qui induit un sous-graphe biparti de G et dont le vecteur d'incidence satisfait (4.11) à l'égalité. Pour cela, prenons arbitrairement un sommet u de $V \setminus W$. Notons que u est contenu dans l'une des faces de H. (Cette face peut être la face extérieure de H). Supposons que u est contenu dans C' et que ses voisins dans H^* appartiennent tous à $V(P')$. (Les autres cas peuvent être traités d'une façon similaire.) Comme C' est impair, il s'ensuit qu'exactement une des faces de H_u contenant u est impaire. Soit u_1 un voisin de u qui appartient à cette face

de telle façon que $u_1 \in V(C_j)$, où $j \in \{1, ..., p\}$. Si j est impair (resp. pair), considérons les ensembles de sommets

$B_u = (W \setminus \{z_1, z_3, ..., z_{j-2}, u_1, z_j, z_{j+2}, ..., z_{p-1}\}) \cup \{u\}$.

(resp. $B_u = (W \setminus \{z_1, z_3, ..., z_{j-1}, u_1, z_{j+1}, ..., z_{p-1}\} \cup \{u\}$.)

L'ensemble B_u induit un sous-graphe biparti de G, et son vecteur d'incidence satisfait (4.11) à l'égalité.

($\Rightarrow$) Supposons qu'il existe un sommet $u \in V \setminus W$ de $N(W)$ de degré trois, dont les voisins, disons u_1, u_2, u_3, sont dans W et tel que les trois faces de H_u contenant u, sont toutes impaires. Par hypothèse, on sait que les sommets u_1, u_2, u_3 appartiennent tous à l'une des faces C', C'', C_i $i = 1, ..., p$. Comme H^* est de degré maximum trois, alors u_1, u_2, u_3 sont de degré deux dans H.

Dans ce qui suit, nous montrons que tout ensemble de sommets B induisant un sous-graphe biparti de G et dont le vecteur d'incidence x^B satisfait (4.11) à l'égalité, ne peut contenir u. Comme (4.11) induit une contrainte non triviale, elle ne peut donc définir une facette. En effet, supposons le contraire et considérons un ensemble de sommets B qui contient u, qui induit un sous-graphe biparti de G et dont le vecteur d'incidence x^B satisfait (4.11) à l'égalité.

Soit $\bar{B} = W \setminus B$. Etant donnés deux entiers r, s, $1 \leq r \leq s \leq p$, on note par $\bar{B}_{r,s}$ l'ensemble des sommets de $\bar{B}$ qui sont dans $\bigcup_{i=r,r+1,...,s} V(C_i)$, et on note par $\bar{H}_{r,s}$ le graphe obtenu à partir de $H_{r,s}$ en ôtant les sommets de $\bar{B}$. On peut remarquer que si $s - r + 1$ est pair (resp. impair), alors $|\bar{B}_{r,s}| \geq \frac{s-r+1}{2}$ (resp. $|\bar{B}_{r,s}| \geq \frac{s-r+2}{2}$). On peut également remarquer que dans le cas où $s - r + 1$ est pair, et $|\bar{B}_{r,s}| = \frac{s-r+1}{2}$, les sommets de $\bar{B}_{r,s}$ appartiennent aux chaînes $P_r, P_{r+2}, ..., P_{s-1}$ et $|\bar{B}_{r,s} \cap V(P_i)| = 1$ pour $i = r, r+2, ..., s-1$.

Assertion 1 *Soient $r, s \in \{1, ..., p\}, r \leq s$. Supposons que $s - r + 1$ est pair et $|\bar{B}_{r,s}| = \frac{s-r+1}{2}$. Soient v_1 et v_2 deux sommets respectivement de $V(w_{r-1} \sim w_r) \setminus \{w_r\}$ et $V(w_{s-1} \sim w_s) \setminus \{w_{s-1}\}$. Alors il existe une chaîne R dans $\bar{H}_{r,s}$ entre v_1 et v_2 de même parité que $v_1 \sim v_2$.*

Preuve. Comme les cycles $C_r, ...C_s$ sont tous impairs et $|\bar{B}_{r,s}| = \frac{s-r+1}{2}$, on doit avoir $|V(P_i) \cap \bar{B}| = 1$ pour $i = r, r+2, ..., s-1$. S.p.d.g., on peut supposer que $V(P_i) \cap \bar{B} \subset \{z_i, w_i\}$ pour $i = r, r+2, ..., s-1$. Par le lemme 4.7.1, il existe donc une chaîne R' dans $\bar{H}_{r,s}$ entre w_{r-1} et w_s qui a la même parité que $w_{r-1} \sim w_s$.

Si $w_r \in R'$ (resp. $w_r \notin R'$), alors soit R'' la chaîne $R' \setminus (w_{r-1} \sim v_1)$ (resp. $R' \cup (w_{r-1} \sim v_1)$). Maintenant, si $w_{s-1} \in R''$ resp. $w_{s-1} \notin R''$), considérons R la chaîne $R'' \setminus (v_2 \sim w_s)$ (resp. $R'' \cup (v_2 \sim w_s)$). Il est clair que R relie v_1 et v_2, et a la même parité que $v_1 \sim v_2$. $\diamondsuit$

Assertion 2 *Soient $r, s \in \{1, ..., p\}$ tel que $r < s$ et $s - r + 1$ est pair. Soient v_1 et v_2 deux sommets respectivement de $V(P') \cap V(C_{r'})$ et $V(P') \cap V(C_{s'})$ où $r' \leq r + 1$ et $s' \geq s$. Supposons qu'il existe deux chaînes R_1 et R_2 de $\bar{H}_{r',s'}$ qui relient respectivement v_1 à w_r et v_2 à w_s. Si $(V(P_r) \cup V(z_r \sim z_s) \cup V(P_s)) \subset B$, alors il existe une chaîne R de $\bar{H}_{r',s'}$, entre v_1 et v_2 qui a une parité opposée à la chaîne $(R_1, w_r \sim w_s, R_2)$.*

Preuve. On considère le cas où $z_r \in V(R_1)$ et $z_s \notin V(R_2)$. (Les autres cas sont similaires.) Comme $z_r \in V(R_1)$, P_r est une sous-chaîne de R_1 et $P_s \cap R_2 = \emptyset$. On peut alors poser $R = (R_1 \setminus P_r, z_r \sim z_s, P_s, R_2)$. Comme les cycles $C_r, ..., C_s$ sont tous impairs et $s - r$ est impair, les chaînes $(P_r, z_r \sim z_s, P_s)$ et $w_r \sim w_s$ ont des parités opposées. Comme la chaîne R peut être obtenu à partir de la chaîne $(R_1, w_r \sim w_s, R_2)$ en remplaçant la sous-chaîne $w_r \sim w_s$ par la chaîne $(P_r, z_r \sim z_s, P_s)$, alors R et $(R_1, w_r \sim w_s, R_2)$ ont des parités opposées. $\diamondsuit$

On considère maintenant deux cas.

Cas 1. u est contenu dans la face C_j pour un $j \in \{1, ..., p\}$. Alors $u_1, u_2, u_3 \in V(C_j)$. On peut supposer que j est impair. Le cas où j est pair se déduit par symétrie. (voir la figure 8.7).

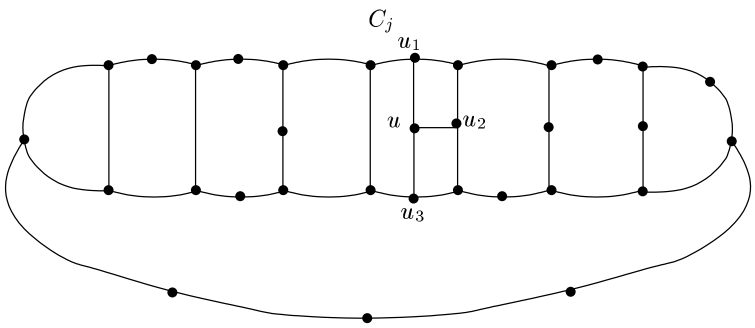


FIG. 8.7 –

On peut aisément montrer que pour tout $i \in \{1, ..., p\}$, $|V(C_i) \cap \bar{B}| \leq 2$. Comme les trois faces de H_u contenant u sont toutes impaires et, par conséquent, doivent être couvertes par au moins deux sommets de $\bar{B}$, on a $|V(C_j) \cap \bar{B}| = 2$ et $|\{u_1, u_2, u_3\} \cap \bar{B}| \geq 1$.
On suppose que $u_1 \in V(z_{j-1} \sim z_j)$, $u_2 \in V(P_j)$ et $u_3 \in V(w_{j-1} \sim w_j)$ avec $u_3 \in \bar{B}$ (les autres cas peuvent être traités d'une manière similaire), alors la sous-chaîne de C_j entre u_1 et u_2 qui ne passe pas par u_3 contient exactement un sommet de $\bar{B}$.
- Si $(V(u_1 \sim z_j) \setminus \{z_j\}) \cap \bar{B} \neq \emptyset$, alors $|\bar{B}| \geq |\bar{B}_{1,j-1}| + 2 + |\bar{B}_{j+1,p}| \geq \frac{p}{2} + 2$, une contradiction.
- Si $V(u_1 \sim z_j) \cap \bar{B} = \emptyset$, alors $|\bar{B}_{1,j-1}| = \frac{j-1}{2}$ et $|\bar{B}_{j+2,p}| = \frac{p-(j+2)+1}{2}$. Ainsi, par

l'assertion 1, il existe deux chaînes R_1 et R_2 qui ont, respectivement, mêmes extrémités et parité que $w_0 \sim w_{j-1}$ et $w_{j+1} \sim w_p$. Comme $V(P_{j-1}) \cup V(z_{j-1} \sim z_{j+1}) \cup V(P_{j+1}) \subset B$, on peut construire, à partir de R_1 et R_2, une chaîne R entre w_0 et w_p ayant la même parité que P'. Alors $R' \cup P^*$ est un cycle impair dont les sommets sont tous dans B, une contradiction.
Par conséquent, on a $z_j \in \bar{B}$. En utilisant les chaînes R_1, $(P_{j-1}, z_{j-1} \sim z_j, P_j)$ et $(w_j \sim w_{j+1}, R_2)$ avec des arguments similaires à ceux utilisés dans la preuve de l'assertion 2, on peut montrer qu'il existe une chaîne R' entre w_0 et w_p avec une parité opposée à celle de P'. De plus, R' passe par u_1, z_j et u_2. Soit R la chaîne obtenu à partir de R' en ôtant la sous-chaîne entre u_1 et u_2 et en ajoutant les arêtes u_1u, uu_2. Comme $V(w_j \sim w_{j+1}) \cap \bar{B} = \emptyset$, R et P^* forment un cycle impair dont tous les sommets sont dans B, une contradiction.

Cas 2. u est contenu dans la face C'. (On peut remarquer que ce cas est le même que celui où u est la face extérieure de H. On doit juste bouger la chaîne P^* pour le voir.) Supposons que les sommets u_1, u_2, u_3 appartiennent tous à la chaîne P'. (Les cas où certains des sommets u_1, u_2, u_3 sont dans la chaîne P^* peuvent être étudiés selon les mêmes principes.) On peut supposer que u_1, u_2, u_3 n'appartiennent pas au même cycle C_i. Si u_1, u_2, u_3 sont tous dans le même cycle, alors on retrouve le cas 1. On pose ainsi j, k, l, $1 \leq j \leq k \leq l \leq p$ de telle façon que u_1, u_2, u_3 appartiennent respectivement à $V(C_j), V(C_k), V(C_l)$ (voir la figure 8.8).
Supposons d'abord que $\{u_1, u_2, u_3\} \subset \bar{B}$. Comme $|V(C_i) \cap \bar{B}| \geq 1$ pour $i = 1, ..., j-1, j+1, ..., k-1, k+2, ..., l-1, l+1, ..., p$, $|\bar{B}| \geq \frac{p}{2} + 2$, on a une contradiction. Ainsi, $|\bar{B} \cap \{u_1, u_2, u_3\}| \leq 2$.
On peut considérer le cas où j est impair, k est pair et l est impair (les autres cas peuvent être traités de façon similaire). De plus, s.p.d.g., on peut supposer que $j > 1$, $j + 1 < k < l - 1$ et $l < p - 1$.

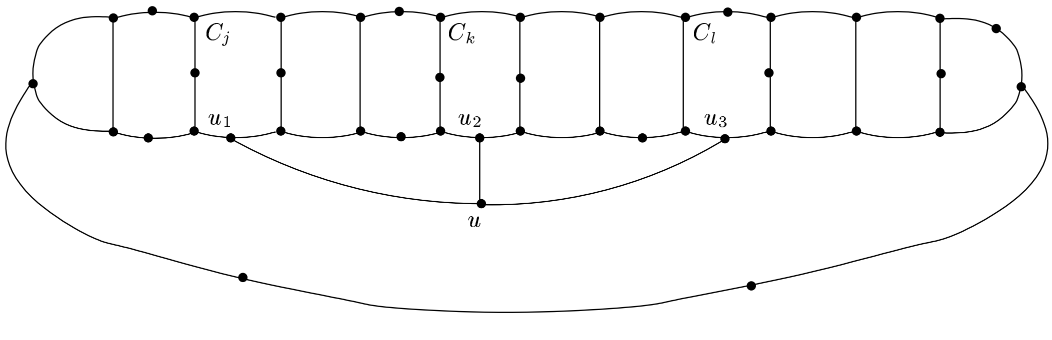


FIG. 8.8 –

Cas 2.1. $|\bar{B} \cap \{u_1, u_2, u_3\}| = 2$.
Si $\bar{B} \cap \{u_1, u_2, u_3\} = \{u_1, u_3\}$ ou $\bar{B} \cap \{u_1, u_2, u_3\} = \{u_2, u_3\}$, alors $|\bar{B}| \geq \frac{p}{2} + 2$,

une contradiction. Donc $\bar{B} \cap \{u_1, u_2, u_3\} = \{u_1, u_2\}$. Par conséquent, $|\bar{B}_{1,j-1}| = \frac{j-1}{2}$, $|\bar{B}_{j+1,k-1}| = \frac{(k-1)-(j+1)+1}{2}$ et $|\bar{B}_{k+1,p}| = \frac{p-(k+1)+1}{2}$. Par l'assertion 1, il existe trois chaînes R_1, R_2 et R_3 ayant, respectivement, mêmes extrémités et parité que $w_0 \sim w_{j-1}$, $w_j \sim w_{k-1}$ et $w_k \sim w_p$. Alors, par l'assertion 2 par rapport à R_1 et R_2, il existe une chaîne R entre w_0 et w_{k-1} ayant une parité opposée à celle de $w_0 \sim w_{k-1}$. Maintenant, en réutilisant l'assertion 2 par rapport à R et R_3, on obtient une chaîne R' entre w_0 et w_p de même parité que P'. Par conséquent, R' et P^* forment un cycle impair. Comme l'ensemble de sommets de ce cycle est dans B, on a une contradiction.

Cas 2.2. $|\bar{B} \cap \{u_1, u_2, u_3\}| = 1$.
On peut supposer que $u_1 \in \bar{B}$ (les cas où $u_2 \in \bar{B}$ et $u_3 \in \bar{B}$ sont similaires). Alors $|V(C_k) \cap \bar{B}| = 1$. Soit $V(C_k) \cap \bar{B} = \{z\}$.
Si $z \notin V((u_2 \sim w_k) \cup P_k)$, alors $|\bar{B}_{k+1,l-1}| = \frac{l-k-1}{2}$. De plus, par l'assertion 1, il existe une chaîne R entre w_k et w_{l-1} de même parité que $w_k \sim w_{l-1}$. Ainsi, le cycle $(uu_2, u_2 \sim w_k, R, w_{l-1} \sim u_3, u_3u)$ est impair et son ensemble de sommets est dans B, une contradiction.

On peut donc supposer que $z \in V((u_2 \sim w_k) \cup P_k)$. Alors $|\bar{B}_{1,j-1}| = \frac{j-1}{2}$ et $|\bar{B}_{j+1,k-1}| = \frac{k-j-1}{2}$. Par conséquent, par l'assertion 1, il existe deux chaînes R_1 et R_2 ayant, respectivement, mêmes extrémités et parité que $w_0 \sim w_{j-1}$ et $w_j \sim w_{k-1}$. Par l'assertion 2 par rapport à R_1 et R_2, il existe une chaîne, disons R^*, entre w_0 et w_{k-1} ayant une parité opposée à celle de $w_0 \sim w_{k-1}$.

Assertion 3 $|\bar{B}_{l,p}| = \frac{p-l+1}{2} + 1$.

Preuve. Il est clair que $|\bar{B}_{l,p}| \geq \frac{p-l+1}{2}$. Supposons que l'assertion ne soit pas vraie. Alors $|\bar{B}_{l,p}| = \frac{p-l+1}{2}$, et, par l'assertion 1, il existe une chaîne R avec les mêmes extrémités et parité que $u_3 \sim w_p$. Cependant, comme les cycles $(uu_1, u_1 \sim u_2, u_2u)$ et $(uu_1, u_1 \sim w_0, P^*, w_p \sim u_3, u_3u)$ sont impairs, il s'ensuit que le cycle $(uu_2, u_2 \sim w_0, P^*, w_p \sim u_3, u_3u)$ est pair. De plus, comme R^* a une parité opposé à celle de $w_0 \sim w_{k-1}$ et que R a la même parité que $u_3 \sim w_p$, le cycle $(uu_2, u_2 \sim w_{k-1}, R^*, P^*, R, u_3u)$ est impair, une contradiction. $\diamondsuit$

Assertion 4 $\{w_{l-1}, w_l\} \subset \bar{B}$.

Preuve. Si $w_{l-1} \notin \bar{B}$, alors $V(u_3 \sim w_{l-2}) \cap \bar{B} = \emptyset$. De plus, comme $|\bar{B}_{k,l-2}| = \frac{l-k-1}{2}$, par l'assertion 1, il existe une chaîne R entre u_2 et w_{l-2} de même parité que $u_2 \sim w_{l-2}$. Alors le cycle $(uu_2, R, w_{l-2} \sim u_3, u_3u)$ est impair, une contradiction. Ainsi $w_{l-1} \in \bar{B}$.

Maintenant, pour montrer que $w_l \in \bar{B}$, nous allons montrer d'abord que $|V(P_l) \cap \bar{B}| = 1$ et $|\bar{B}_{l+2,p}| = \frac{p-l-1}{2}$. Si $V(P_l) \cap \bar{B} = \emptyset$, alors $|\bar{B}_{k,l}| = \frac{l-k+1}{2}$, et ainsi par l'assertion 1 il existe une chaîne R entre u_2 et u_3 avec la même parité que $u_2 \sim u_3$. Alors le cycle (uu_2, R, u_3u) est impair, une contradiction. Suppose que $w_l \notin \bar{B}$. Alors $V(u_3 \sim w_{l+1}) \cap \bar{B} = \emptyset$. Par l'assertion 1, il existe une chaîne R entre w_{l+1} et w_p avec la même parité que $w_{l+1} \sim w_p$. Cependant, comme les cycles $(uu_1, u_1 \sim u_2, u_2u)$ et $(uu_1, u_1 \sim w_0, P^*, w_p \sim u_3, u_3u)$ sont tous impairs, il s'ensuit que le cycle $(uu_2, u_2 \sim w_0, P^*, w_p \sim u_3, u_3u)$ est pair. Comme R^* a une parité opposée à celle de $w_0 \sim w_{k-1}$, le cycle $(uu_2, u_2 \sim w_{k-1}, R^*, P^*, R, w_{l+1} \sim u_3, u_3u)$ est impair, une contradiction. $\Diamond$

Par les assertions 3 et 4, $|\bar{B}_{k,l-2}| = \frac{l-k-1}{2}$ et $|\bar{B}_{l+2,p}| = \frac{p-l-1}{2}$. Alors par l'assertion 1, il existe deux chaînes R_1 et R_2 avec les mêmes extrémités et parité que, respectivement, $w_{k-1} \sim w_{l-2}$ et $w_{l+1} \sim w_p$. Par conséquent, par l'assertion 2 par rapport à R_1 et R_2, il existe une chaîne R entre w_{k-1} et w_p avec une parité opposée à celle de $w_{k-1} \sim w_p$. On peut remarquer que si on remplace dans C' la chaîne $w_0 \sim w_{k-1}$ par R^*, comme R^* a une parité opposée à celle de $w_0 \sim w_{k-1}$, le cycle obtenu, disons $\widetilde{C}'$, est pair. Si on remplace aussi dans $\widetilde{C}'$, $w_{k-1} \sim w_p$ par R, comme les deux chaînes ont des parités opposées, on obtient un cycle impair. Cependant, tous les sommets de ce dernier cycle sont tous dans B, ce qui est impossible.

Cas 2.3. $\bar{B} \cap \{u_1, u_2, u_3\} = \emptyset$.
Soient $U_1 = V((u_1 \sim w_j) \cup P_j) \cup V(P_{k-1} \cup (w_{k-1} \sim u_2))$,
$U_2 = V((u_2 \sim w_k) \cup P_k) \cup V(P_{l-1} \cup (w_{l-1} \sim u_3))$.

Si $U_1 \cap \bar{B} = \emptyset$ (le cas où $U_2 \cap \bar{B} = \emptyset$ se déduit par symétrie), alors $|\bar{B}_{j+1,k-1}| = \frac{k-j-1}{2}$. Par l'assertion 1, il s'ensuit donc qu'il existe une chaîne R avec mêmes extremités et parité que $w_j \sim w_{k-1}$. Comme $V(u_1 \sim w_j) \cap \bar{B} = \emptyset$ et $V(w_{k-1} \sim u_2) \cap \bar{B} = \emptyset$, le cycle $(uu_1, u_1 \sim w_j, R, w_{k-1} \sim u_2, u_2u)$ est impair et son ensemble de sommets est dans B, une contradiction.

Supposons maintenant que $U_1 \cap \bar{B} \neq \emptyset \neq U_2 \cap \bar{B}$.

Assertion 5 *i)* $|\bar{B}_{1,j-1}| = \frac{j-1}{2}$,
ii) $|\bar{B}_{j,k}| = \frac{k-j+1}{2} + 1$,
iii) $|\bar{B}_{l,p}| = \frac{p-l+1}{2} + 1$.

Preuve. i) On observe tout d'abord que $\frac{j-1}{2} \leq |\bar{B}_{1,j-1}| \leq \frac{j-1}{2} + 1$. Supposons que $|\bar{B}_{1,j-1}| = \frac{j-1}{2} + 1$. Si $V((u_1 \sim w_j) \cup P_j) \cap \bar{B} \neq \emptyset$, alors $|\bar{B}_{j+2,p}| \geq \frac{p-j-1}{2}$. Ceci implique que $|V(P_i) \cap \bar{B}| = 1$ pour $i = j+2, j+4, ..., p-1$ et alors $U_2 \cap \bar{B} = \emptyset$, une contradiction. Si ce n'est pas le cas, alors $V(P_{k-1} \cup (w_{k-1} \sim u_2)) \cap \bar{B} \neq \emptyset$, et $|\bar{B}_{j+1,k-2}| \geq \frac{k-j-1}{2} + 1$.

Par conséquent, $|\bar{B}_{k+1,p}| = \frac{p-k}{2}$ et donc $|V(P_i) \cap \bar{B}| = 1$ pour $i = k+1, k+3, ..., p-1$. On a alors $U_2 \cap \bar{B} = \emptyset$, une contradiction.
ii) On sait que $\frac{k-j+1}{2} \leq |\bar{B}_{j,k}| \leq \frac{k-j+1}{2}+1$. Supposons que $|\bar{B}_{j,k}| = \frac{k-j+1}{2}$. Par l'assertion 1, il existe une chaîne R avec mêmes extrémités et parité que $u_1 \sim u_2$. Alors le cycle (uu_1, R, u_2u) est impair, une contradiction.
iii) On sait que $\frac{p-l+1}{2} \leq |\bar{B}_{l,p}| \leq \frac{p-l+1}{2} + 1$. Supposons que $|\bar{B}_{l,p}| = \frac{p-l+1}{2}$. Alors, comme $|\bar{B}_{1,j-1}| \leq \frac{j-1}{2}$, par l'assertion 1, il existe deux chaînes R_1 et R_2 avec mêmes extrémités et parité que, respectivement, $w_0 \sim u_1$ et $u_3 \sim w_p$. Comme le cycle $(uu_1, u_1 \sim w_0, P^*, w_p \sim u_3, u_3u)$ est impair, le cycle $(uu_1, R_1, P^*, R_2, u_3u)$ est aussi impair. Comme l'ensemble de sommets de ce dernier cycle est dans B, on a une contradiction. $\diamondsuit$

Assertion 6 *i)* $|V(P_k) \cap \bar{B}| = 1$,
ii) $|V(P_{l-1}) \cap \bar{B}| = 1$.

Preuve. Nous donnons la preuve de i), celle de ii) est similaire. Il suffit de montrer que $V(P_k) \cap \bar{B} \neq \emptyset$. Si $|V(P_k) \cap \bar{B}| \geq 2$, comme par l'assertion 5 ii), $|\bar{B}_{j,k}| = \frac{k-j+1}{2} + 1$, il s'ensuit que $\bar{B} \cap V(C_t) = \emptyset$ pour un certain $t \in \{j, ..., k-1\}$, une contradiction. Maintenant, supposons, au contraire, que $V(P_k) \cap \bar{B} = \emptyset$. Comme $k < l-1$, k est pair et l est impair, on a $l - k \geq 3$. Si $l - k = 3$, alors par l'assertion 5, on a $|\bar{B}_{1,j-1}| + |\bar{B}_{j,k}| + |\bar{B}_{l,p}| = \frac{p}{2}+1$ (Notons que par l'assertion 5 i), $\bar{B}_{1,j-1} \cap \bar{B}_{j,k} = \emptyset$). Par conséquent $V(C_{k+1}) \cap \bar{B} = \emptyset$, une contradiction. Si $l - k > 3$, alors $\frac{(l-2)-(k+2)+1}{2} \leq |\bar{B}_{k+2,l-2}| \leq \frac{p}{2} + 1 - |\bar{B}_{1,j-1}| - |\bar{B}_{j,k}| - |\bar{B}_{l,p}| \leq \frac{(l-2)-(k-2)+1}{2} - 1$, et nous avons à nouveau une contradiction. $\diamondsuit$

Assertion 7 *Si* $|V(P_{k-1}) \cap \bar{B}| = 1$, *alors* $w_{k-1} \in \bar{B}$.
Si $|V(P_l) \cap \bar{B}| = 1$, *alors* $w_l \in \bar{B}$.

Preuve. Supposons que $|V(P_{k-1}) \cap \bar{B}| = 1$ avec $w_{k-1} \notin \bar{B}$. Alors $V(u_2 \sim w_{k-2}) \cap \bar{B} = \emptyset$. Par les assertions 5 ii) et 6 i) il s'ensuit que $|\bar{B}_{j,k-2}| = \frac{(k-2)-j+1}{2}$. Ainsi par l'assertion 1, il existe une chaîne R entre u_1 et w_{k-2} avec la même parité que $u_1 \sim w_{k-2}$. Mais ceci implique que le cycle $(uu_1, R, w_{k-2} \sim u_2, u_2u)$ est impair, une contradiction.
Maintenant, on suppose que $|V(P_l) \cap \bar{B}| = 1$ avec $w_l \notin \bar{B}$. Alors $V(u_3 \sim w_{l+1}) \cap \bar{B} = \emptyset$. Observons que par l'assertion 5 i), $|\bar{B}_{1,j-1}| = \frac{j-1}{2}$, et par les assertions 5 iii) et 6 ii), $|\bar{B}_{l+2,p}| = \frac{p-(l+2)+1}{2}$. Alors par l'assertion 1, il existe deux chaînes R_1 et R_2 avec mêmes extrémités et parité que, respectivement, $w_0 \sim w_{j-1}$ et $w_{l+1} \sim w_p$. Alors le cycle $(uu_1, u_1 \sim w_{j-1}, R_1, P^*, R_2, u_3 \sim w_{l+1}, u_3u)$ est impair, une contradiction. $\diamondsuit$

On distingue maintenant deux cas.

Cas 2.3.1. Soit $w_k \notin \bar{B}$, soit $w_k \in \bar{B}$ et $V(P_{k-1}) \cap \bar{B} = \emptyset$.
On suppose tout d'abord que soit $w_{l-1} \notin \bar{B}$, soit $w_{l-1} \in \bar{B}$ et $V(P_l) \cap \bar{B} = \emptyset$. Dans les deux cas, il est facile de prouver qu'il existe une chaîne R entre u_2 et u_3 avec la même parité que $u_2 \sim u_3$. Alors (uu_2, R, u_3u) est un cycle impair dont l'ensemble des sommets est dans B, une contradiction.
Par conséquent, $w_{l-1} \in \bar{B}$ et $|V(P_l) \cap \bar{B}| = 1$. Par les assertions 5 et 1, il est facile de prouver qu'il existe trois chaînes R_1, R_2 et R_3 ayant mêmes extrémités et parité que, respectivement, $w_0 \sim u_1$, $u_2 \sim w_{l-2}$ et $w_{l+1} \sim w_p$. Alors, par l'assertion 2 par rapport à R_2 et R_3, il existe une chaîne R entre u_2 et w_p ayant une parité opposée à celle de $u_2 \sim w_p$. Comme les faces $(uu_2, u_2 \sim u_3, u_3u)$ et $(uu_1, u_1 \sim w_0, P^*, w_p \sim u_3, u_3u)$ sont impaires, alors le cycle $(uu_1, R_1, P^*, w_p \sim u_2, u_2u)$ est pair. Ainsi, le cycle $(uu_1, R_1, P^*, R, u_2u)$ est impair, une contradiction.

Cas 2.3.2. $w_k \in \bar{B}$ et $|V(P_{k-1}) \cap \bar{B}| = 1$.
Alors par l'assertion 7, $w_{k-1} \in \bar{B}$. Supposons que soit $w_{l-1} \notin \bar{B}$, soit $w_{l-1} \in \bar{B}$ et $V(P_l) \cap \bar{B} = \emptyset$. Par l'assertion 5, on peut facilement prouver qu'il y a deux chaînes R_1 et R_2 avec mêmes extremités et parité que, respectivement, $u_1 \sim w_{k-2}$ et $w_{k+1} \sim u_3$. Alors, par l'assertion 2, il existe un chaîne R entre u_1 et u_3 avec une parité opposée à celle de $u_1 \sim u_3$. Ainsi, le cycle $(uu_1, u_1 \sim u_3, u_3u)$ est pair, et le cycle (uu_1, R, u_3u) est impair, une contradiction.
Par conséquent $w_{l-1} \in \bar{B}$ et $|V(P_l) \cap \bar{B}| = 1$. Alors, par l'assertion 7, $w_l \in \bar{B}$. De plus, par les assertions 5 et 1, il est facile de voir qu'il existe trois chaînes R_1, R_2 et R_3 avec mêmes extrémités et parité que, respectivement, $w_0 \sim w_{k-2}$, $w_{k+1} \sim w_{l-2}$ et $w_{l+1} \sim w_p$. Alors, par l'assertion 2 par rapport à R_1 et R_2, il existe une chaîne R entre w_0 et w_{l-2} ayant une parité opposée à celle de $w_0 \sim w_{l-2}$. D'une manière similaire, par l'assertion 2 par rapport à R et R_3, il existe une chaîne, disons $\widetilde{R}$, entre w_0 et w_p ayant la même parité que P'. Comme la chaîne $\widetilde{R}$ et P^* forment un cycle impair dont l'ensemble des sommets est dans B, on a une contradiction, ce qui termine la preuve de notre théorème. $\square$

Bibliographie

[1] Abacus, "http://www.informatik.uni-koeln.de/abacus",.

[2] A. Abdullah and S. Sastry, "Topological via minimization and routing", *Proc. First Great Lakes Symposium on VLSI*, (1991) 220-224.

[3] K. Ahn and S.Sahni, "Constrained via minimization", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, (1993) 273-282.

[4] C.J. Alpert and A.B. Kahng, "Recent directions in netlist partitioning. integration", *the VLSI Journal*, (1995) 1-81.

[5] C.S. Rim andT. Kashiwabara and K. Nakajima, "Exact algorithms for multilayer topological via minimization", *TR 88-88, Univ. of Maryland*, 1988.

[6] C.S. Rim andT. Kashiwabara and K. Nakajima, "A note on the np-hardness of the topological via minimization problem", *TR 89-18, Univ. of Maryland*, 1989.

[7] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, "Implementing the dantzigfulkerson -johnson algorithm for large traveling salesman problems", *Mathematical Programming*, (2003) 91-153.

[8] F. Barahona, "On the complexity of max-cut", *Rapport de recherche no. 186, IMAG, Université Scientifique et Médicale de Grenoble*, (1980).

[9] F. Barahona, "On via minimization", *Report No. 87491-OR, Univ Bonn*, 1987.

[10] F. Barahona, "On some applications of the chinese postman problem", *Algorithm and Combinatoricd 9, Springer Verlag*, (1990) 1-16.

[11] F. Barahona, "On via minimization", *IEEE Trans. on Circuits and Systems 37*, (1990) 527-530.

[12] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt, "An application of combinatorial optimization to statistical physics and circuit layout design", *Operations Research 36*, (1988) 493-513.

[13] F. Barahona and A.R. Mahjoub, "On the cut polytope", *Mathematical Programming 36*, (1986) 157-173.

[14] F. Barahona and A.R. Mahjoub, "Facets of the balanced (acyclic) induced subgraph polytope", *Mathematical Programming 45*, (1989) 21-34.

[15] F. Barahona and A.R. Mahjoub, "Composition of graphs and polyhedra i : Balanced induced subgraphs and acyclic subgraphs", *SIAM Journal on Discrete Mathematics 7*, (1994) 344-358.

[16] F. Barahona, A. Weintaub, and R. Epstein, "Habitat dispersion in forest planning and stable set problem", *Operations Research*, (1992) 14-21.

[17] P. Bonizzoni, G. Della Vedova, R.Dondi, and J.Li, "The haplotyping problem: An overview of computational models and solutions", *Journal of Computer Science and Technology*, (2003) 675-688.

[18] M.L. Brady and D.J. Brown, "Vlsi routing: four layers suffice", *Advances in Computing research*, (1984) 245-257.

[19] M.L. Brady and M. Sarrafzadeh, "Streching a knock-knee layout for multilayer wiring", *IEEE Transactions on computer*, (1990) 148-152.

[20] M.A. Breuer, "Design automation of digital system", *vol.1, Englewood CLiffs, NJ:Prentice Hall*, (1972) 321-329.

[21] M. Campêlo, R. Corrêa, and Y. Frota, "Cliques, holes and the vertex coloring polytope", *Information Processing Letters*, (2004) 159-164.

[22] C.C Chang and J. Cong, "An efficient approach to multi-layer layer assignment with application to via minimization", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, (1999) 608-620.

[23] K.C. Chang and D.H.C. Du, "Efficient algorithm for layer assignment problem", *IEEE Trans. on Computer-Aided Design 6*, (1987) 67-78.

[24] R. Chen, Y. Kajitani, and S. Chan, "A graph theoretic via minimization algorithm for two layer printed circuit boards", *IEEE Trans. on Circuits and Systems*, (1983) 284-299.

[25] Z. Chen and I. Koren, "Layer reassignment for antenna effect minimization in 3-layer channel routing", *Proc. IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI Systems*, (1996) 77-85.

[26] E. Cheng and W.H. Cunningham, "Wheel inequalities for stable set polytope", *Mathematical programming*, (1997) 389-421.

[27] E. Cheng and S. de Vries, "Antiweb-wheel inequalities and their separation problems over the stable set polytopes", *Mathematical programming*, (2002) 153-175.

[28] H.A. Choi, K. Nakajima, and C.S. Rim, "Graph bipartization and via minimization", *SIAM Journal on Discrete Mathematics 2*, (1989) 38-47.

[29] S. Chopra, "Comparison of formulations and a heuristic for packing steiner trees in a graph", *Ann. of Operation Research 50*, (1994) 143-171.

[30] Y.C. Chou and Y.L. Lin, "A graph-partioning-based approach for multi-layer constrained via minimization", *Int. conf. on computer-aided design ICCAD*, (1998).

[31] M.J. Ciesielski and Kinnen, "An optimum layer assignment for routing in ics and psbs", *18th IEEE Design Automation Conference*, (1981) 733-737.

[32] J. Cong, J. Fang, and K.Y. Khoo, "Via design rule consideration in multi-layer maze routing algorithms", *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, (2000) 215-223.

[33] J. Cong and C. L. Liu, "On the k-layer planar subset and topological via minimization problems", *IEEE Trans. on Computer-Aided Design 10*, (1991) 972-981.

[34] Cplex, "http://www.ilog.com",.

[35] J. Edmonds, "Maximum matching and a polyhedron with 0,1-vertices", *Journal of Research National Bureau of Standards*, (1965) 125-130.

[36] M. Grötschel F. Barahona and A.R. Mahjoub, "Facets of the bipartite subgraph polytope", *Mathematics of Operations Research*, (1985) 340-358.

[37] S.C. Fang, K.E. Chang, W.S. Feng, and S.J. Chen, "Constrained via minimization with practical considerations for multi-layer vlsi/pcb routing problems", *Proc. 28th conf. on ACM/IEEE design automation conference*, (1991) 60-65.

[38] J. Fonlupt, A.R. Mahjoub, and J.P. Uhry, "Composition in the bipartite subgraph polytope", *Discrete Mathematics*, (1992) 73-91.

[39] S.B. Gabriel, S.F. Schaffner, H. Nguyen, and et al., "The structure of haplotype blocks in the human genome", *Science*, (2002) 2225-2229.

[40] M.RM Garey and D.S. Johnson, "Computers and intractability. a guide to the theory of np-completeness", *W. H Freeman and Company*, (1979).

[41] F. Glover, "Future paths for integer programming and links to arti®cial intelligence", *Computers and Operation Research*, (1986) 533-549.

[42] M. Grötschel, M. Jünger, and G. Reinelt, "Via minimization with pin preassignment and layer preference", *Zeitschrift für Angewandte Mathematik und Mechanik 69*, (1989) 393-399.

[43] M. Grötschel, A. Martin, and R. Weismantel, "Packing steiner trees: polyhedral investigations", *Mathematical Programming 72*, (1996) 101-123.

[44] M. Grötschel, A. Martin, and R. Weismantel, "Packing steiner trees: A cutting plane algorithm and computational results", *Mathematical Programming*, (1996) 125-145.

[45] M. Grötschel, A. Martin, and R. Weismantel, "Packing steiner trees: separation algorithms", *SIAM Journal on Discrete Mathematics 9*, (1996) 233-257.

[46] M. Grötschel, A. Martin, and R. Weismantel, "Packing steiner trees: further facets", *European Journal on Combinatorics 17*, (1996) 39-52.

[47] M. Grötschel, A. Martin, and R. Weismantel, "The steiner tree packing problem in vlsi-design", *Mathematical Programming 78*, (1997) 265-282.

[48] M. Grötschel and W. R. Pulleyblank, "Weakly bipartite graphs and the max-cut problem, operations research", *Operations Research*, (1981) 23-27.

[49] B. Guenin, "A characterization of weakly bipartite graphs", *Journal of Combinatorial Theory B*, (2001) 112-168.

[50] F. O. Hadlock, "Finding a maximum cut of a planar graph in polynomial time", *SIAM Journ. Computing 4*, (1975) 221-225.

[51] P. Hansen, "The steepest ascent mildest descent heuristic for combinatorial programming", *Congress on Numerical Methods in Combinatorial Optimization*, (1986).

[52] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment with large apertures", *Proc. 8th Design Automation Workshop*, (1971) 155-169.

[53] T.D. Hemazro, B. Jaumard, and O. Marcotte, "A column generation and branch-and-cut algorithm for the channel assignment problem", *Les Cahiers du GERAD G-2003-86, year = (2003).*

[54] http://vlsicad.cs.ucla.edu/ cheese/ispd98.html.

[55] http://vlsicad.eecs.umich.edu/BK/ISPD02bench/.

[56] http://www.cadence.com.

[57] http://www.mentor.com.

[58] D.G. Jorgensen and M. Meyling, "A branch-and-price algorithm for switch-box routing", *TR DIKU-TR-00/10, dep. of Comp. Science, Univ of Copenhagen*, 2000.

[59] Y. Kajitani, "On via hole minimization of routings on a 2-layer board", *Proc. IEEE ICCC*, (1980) 295-298.

[60] J. Kao, P. Galinier, and M. Habib, "Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes", *Revue d'Intelligence Artificielle*, (2000) 283-324.

[61] M. Kaufmann and K. Mehlhorn, "Routing problems in grid graphs", *Algorithms and Combinatorics 9, Springer-Verlag*, (1991) 165-184.

[62] K. Y. Khoo and J. Cong, "An efficient multilayer mcm router based on four-via routing", *IEEE Trans. on Computer-Aided Design*, (1995) 1277-1290.

[63] B. Korte, H.J. Prömel, and A. Steger, "Steiner trees in vlsi-layout", *Paths, Flows, and VLSI-Layout, Springer-Verlag*, (1990) 185-214.

[64] Y.S. Kuo, T.C Chern, and W.K. Shih, "Fast algorithm for optimal layer assignment", *25th ACM/IEEE Design Automation Conference*, (1988) 554-559.

[65] Université Blaise Pascal Laboratoire de Physique Corpusculaire de Clermont-Ferrand, "http://clrwww.in2p3.fr/",.

[66] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz, "Snps problems, complexity, and algorithms", *ESA*, (2001) 182-193.

[67] M. Laurent, "A generalization of antiwebs to independence systems and their canonical facets", *Mathematical Programming*, (1989) 97-108.

[68] M. Laurent, "A generalization of antiwebs to independence systems and their canonical facets", *Mathematical programming*, (1989) 97-108.

[69] T. Lengauer and M. Lügering, "Integer program formulations of global routing and placement problems", *Reihe Informatik Nr. 95, Univ. Gesamthochschule-Paderborn*, 1991.

[70] V. Letrouit, "Optimisation du réseau des routes aériennes en europe", *Thèse de l'INPG de Grenoble*, 1998.

[71] R. Lippert, R. Schwartz, G. Lancia, and S. Istrail, "Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem",.

[72] W. Lipski, "On the structure of three-layer wirable layouts", *Advances in computing research*, (1984) 231-243.

[73] M.E. Lübbecke and J. Desrosiers, "Selected topics in column generation", *Les Cahiers du GERAD G-2002-64*, (2002).

[74] L. Lovasz M. Grötschel and A. Schrijver, "Geometric algorithms and combinatorial optimization", *Springer-Verlag*, (1985).

[75] A.R. Mahjoub, "A min-max relation for K_3-covers in graphs non-contractible to $K_5 \setminus e$", *Discrete Applied Mathematics*, (1995) 209-219.

[76] M. Marek-Sadowska, "An unconstrained topological via minimization problem for two-layer routing", *IEEE Trans. on Computer-Aided Design 3*, (1984) 184-190.

[77] A. Martin and R. Weismantel, "Packing paths and steiner trees: routing of electronic circuits", *CWI Quarterly, Centrum voor Wiskunde en Informatica 6*, (1993) 185-204.

[78] A. Mehrotra and M. A. Trick, "A column generation approach for graph coloring", *INFORMS, Journal on Computing*, (1996) 344-354.

[79] I. Mendez-Diaz and P. Zabala, "Application of mathematical programming to graph-coloring", *International Symposium on Mathematical Programming*, (2003.

[80] R.H. Möhring, D. Wagner, and F. Wagner, "Vlsi network design, a survey", *TR No. 323, Univ. Berlin*, 1992.

[81] P. Molitor, "On the contact mminimization problem", *Proc. of the 4th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes on Computer Science 247*, (1987) 420-431.

[82] P. Molitor, "Constrained via minimization for systolic arrays", *IEEE Trans. on Computer-Aided Design*, (1990) 537-542.

[83] P. Molitor, "A survey on wiring", *EIK Journal of Information Processing and Cybernetics 27*, (1991) 3-19.

[84] P. Molitor, "A hierachy preserving hierarchical bottom-up 2-layer wiring algorithm with respect to via minimization", *Integration, the VLSI journal 15*, (1993) 73-95.

[85] P. Molitor, "Recent developments in wiring and via minimization", *Advanced Routing of Electronic Modules, CRC Press*, (1996) 285-311.

[86] P. Molitor, U. Sparmann, and D. Wagner, "Two layer assignment with pin preassignment is easier if power supply is already generated", *Proc. of 7th Int. Conf. on VLSI Design, Calcutta*, (1994) 149-154.

[87] N.J. Naclerio, S. Masuda, and K. Nakajima, "Via minimization for gridless layouts", *24th ACM/IEEE Design Automation Conference*, (1987) 159-165.

[88] G.L. Nemhauser and G. Sigismondi, "A strong cutting plane/branch-and-bound algorithm for node packing", *Journ. of the Operations Research Society*, (1992) 443-457.

[89] G.L. Nemhauser and L.E. Trotter, "Properties of vertex packing and independence system polyhedra", *Mathematical Programming*, (1974) 48-61.

[90] A. Panconesi and M. Sozio, "Fast hare: a fast heuristic for single individual snp haplotype reconstruction", *To appear in Proceedings of WABI 2004*, (2004).

[91] R.Y. Pinter, "Optimal layer assignment for interconnect", *Proc. International Symposium on Circuits and Systems.*

[92] W. R. Pulleyblank, "Polyhedral combinatorics", *Handbooks in Operations Research and Management Science*, (1989) 371-446.

[93] M. Jünger R. Euler and G. Reinelt, "Generalizations of cliques, odd cycles and anticycles and their relation to independence system polyhedra", *Mathematics of Operations Research*, (1987) 451-462.

[94] D. Ryan and D.B. Foster, "An integer programming approach to scheduling", *Computer Scheduling of Public Transport, A. Wren (Ed.)*, (1981) 269-280.

[95] M. Sarrafzadeh and D. T. Lee, "A new approach to topological via minimization", *IEEE Transactions on Computer-Aided Design*, (1989) 890-900.

[96] A. Schrijver, "Combinatorial optimization - polyhedra and efficiency", *Springer*, (2003).

[97] C.J. Shi, "Constrained via minimization and signed hypergraph partitioning", *Algorithmic Aspects of VLSI Layouts, World Scientific Publishing Company*, (1993) 337-356.

[98] C.J. Shi and J.A. Brzozowski, "A characterization of signed hypergraphs and its applications to vlsi via minimization and logic synthesis", *Discrete Applied Mathematics 90*, (1999) 223-243.

[99] C.J Shi, A. Vannelli, and J. Vlach, "Performance-driven layer assignment by integer linear programming and path-constrained hypergraph partitionning", *Journal of heuristics 3*, (1997) 225-243.

[100] Magic A VLSI Layout System, "http://vlsi.cornell.edu/magic/",.

[101] Stefan Thienel, "Abacus - a branch-and-cut system", *doctoral thesis, Universität zu Köln,*, (1995).

[102] L.E. Trotter, "A class of facet producing graphs for vertex packing polyhedra", *Discrete Mathematics*, (1975) 373-388.

[103] E. Uchoa and M. Poggi de Aragão, "Vertex-disjoint packing of two steiner trees: Polyhedra and branch-and-cut", *Mathematical Programming 90*, (2001) 537-557.

[104] F. Vanderbeck, "Decomposition and column generation for integer programs", *Universit'e Catholique de Louvain*, (1994).

[105] J.C. Venter, M.D. Adams, E.W. Myers, and et al., "The sequence of the human genome", *Science*, (2001) 1304-1351.