



HAL
open science

Contribution à l'algorithmique des graphes: quelques représentations pertinentes de graphes

Pascal Berthomé

► **To cite this version:**

Pascal Berthomé. Contribution à l'algorithmique des graphes: quelques représentations pertinentes de graphes. Modélisation et simulation. Université Paris Sud - Paris XI, 2006. tel-00460292

HAL Id: tel-00460292

<https://theses.hal.science/tel-00460292>

Submitted on 26 Feb 2010

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

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

Mémoire d'Habilitation à Diriger des Recherches

soutenu le 27 octobre 2006
à l'université Paris-Sud

Spécialité : Informatique

Contribution à l'algorithmique des graphes :
Quelques représentations pertinentes de
graphes

Pascal Berthomé

Composition du jury :

- Dominique Gouyou-Beauchamps, Professeur, Université Paris-Sud (Président)
- Michel Habib, Professeur, Université Paris VII (Rapporteur)
- Joseph Peters, Professeur, Université S. Fraser, Vancouver, Canada (Rapporteur)
- Denis Trystram, Professeur, Inst. National Polytechnique de Grenoble (Rapporteur)
- Jean-Claude König, Professeur, Université Montpellier II
- Philippe Mahey, Professeur, Université Clermont II
- Dominique Barth, Professeur, Université Versailles Saint-Quentin (Invité)

Remerciements

Je tiens à remercier en premier en premier lieu Joe Peters, Michel Habib et Denis Trystram d'avoir accepté d'être rapporteurs de mon habilitation, malgré leurs emplois du temps chargés. Leurs commentaires ont été précieux. Je suis reconnaissant à Dominique Gouyou-Beauchamps, Jean-Claude König, Philippe Mahey et Dominique Barth de participer au jury.

Toute l'équipe GrafComm du LRI doit ici être saluée pour sa bonne humeur et ses encouragements au moment délicat de la rédaction. Pierre Fraigniaud m'a fait prendre conscience qu'il était désormais temps de soutenir, Selma Djelloul qui doit aussi préparer son répertoire HDR, Li Hao avec lequel je partage mon bureau depuis quelques années déjà, mais aussi Jean-François, Mekkia, Abdel et David. Je ne voudrais pas oublier les plus anciens de l'équipe : Maryvonne, Odile et Marie-Claude, ni les plus jeunes, en particulier David(s), Nicolas et les autres thésards de l'équipe que j'ai pu croiser durant ces années. Je tiens aussi à saluer mes thésards, Moaiz et Lynda.

De manière plus large, je salue les diverses personnes du LRI qui m'ont enrichi grâce à leur expérience. Je n'oublie pas les diverses personnes avec lesquelles j'ai pu collaborer en particulier à Évry et à Versailles.

Durant les années, l'enseignement a pris une grande place. Je dois remercier l'IUT et le département informatique de m'avoir laissé *un peu plus tranquille* ces dernières années, en me déchargeant en particulier de l'apprentissage, expérience très enrichissante sur beaucoup de plans. Dans le désordre et de manière non exhaustive, merci à Didier, Marianne, Dominique, Lydia, Annie, Sylvie, Colette, Cédric, . . . pour leurs diverses discussions fructueuses autour de la machine à café de l'IUT.

Ces remerciements ne seraient pas complets si j'oubliais la partie la plus importante de ma vie : ma famille. Merci à Aline, Quentin, Justine, Elsa et Marion.

Table des matières

1	Introduction	1
1.1	Contexte de la recherche	1
1.2	Fil d'Ariane	2
1.3	Organisation du document	2
I	Communications dans les réseaux	5
2	Contexte des réseaux	7
2.1	Réseaux d'interconnexion optiques	7
2.1.1	Réseaux d'interconnexion	7
2.1.2	Introduction de l'optique dans ces réseaux	7
2.2	Réseaux de télécommunication	8
2.2.1	Modélisation d'un routeur	9
2.2.2	Communications multipoints	10
2.2.3	Dimensionnement de réseaux	11
2.2.4	Évolution	11
2.3	Apports de l'optique	11
2.4	Organisation de cette partie	12
3	Aspects structurels des communications dans les réseaux	13
3.1	Plongements dans le POPS	13
3.1.1	Définitions et modèles	14
3.1.2	Résultats	15
3.1.3	Conclusions et perspectives de ce travail	16
3.2	Échange total périodique	17
3.2.1	Définitions	17
3.2.2	k -MTA sur des réseaux simples	19
3.2.3	Décomposition d'un réseau général pour l'échange total	21
3.2.4	Décomposition de réseaux particuliers	22
3.2.5	Conclusions et perspectives de ce travail	23
3.3	Routage dans les réseaux de télécommunication optiques	23
3.3.1	Les principes du routage eulérien	23
3.3.2	Résultats	26
3.3.3	Conclusion et perspectives sur le routage eulérien	27
3.4	Conclusions et perspectives de ce chapitre	28

4	Optimisation de ressources dans les réseaux	29
4.1	Communications multipoint	29
4.1.1	Complexité du problème MRS	30
4.1.2	Cas polynomiaux	31
4.1.3	Résolution exacte et heuristiques	32
4.1.4	Conclusion sur les groupes multipoints	32
4.2	Réseaux évolutifs	33
4.2.1	Autres modèles	33
4.2.2	Complexité et approximation du problème	34
4.2.3	Modélisation mathématique	35
4.2.4	Méthodes de résolution	36
4.2.5	Conclusion sur ce problème	37
4.3	Conclusions et perspectives : vers une famille de problèmes évolutifs	37
II	Représentations de graphes	39
5	Les flots multi-terminaux paramétrés	43
5.1	Introduction	43
5.2	Définitions	45
5.2.1	Notion de flots	45
5.2.2	Arbres de flots, arbres de coupes	46
5.2.3	Paramétrisation	47
5.3	Vers une simplification de l'analyse de sensibilité	48
5.3.1	Vision d'Elmaghraby	48
5.3.2	Vision positive	48
5.4	Extensions du théorème 5.1	50
5.4.1	Bornes	50
5.4.2	Plusieurs arêtes paramétrées	51
5.5	Calcul d'arbres de Gomory-Hu couplés	52
5.5.1	Recherche d'un arbre de Gomory-Hu intermédiaire	52
5.5.2	Calcul d'un arbre de Gomory-Hu supplémentaire	53
5.6	Les ε -arbres de Gomory-Hu	56
5.6.1	Définition	56
5.6.2	Propriétés	58
5.6.3	Application : le problème ASF	59
5.7	Conclusions et perspectives de ce chapitre	60
5.7.1	Algorithmes de calcul d'arbres de Gomory-Hu	60
5.7.2	Autres fonctions admettant des arbres de Gomory-Hu et paramétrisation	60
5.7.3	ε -arbres de Gomory-Hu	61
5.7.4	Mineurs de graphes et flots	61
6	L'arbre de cliques augmenté : une autre représentation d'un graphe	63
6.1	Introduction	63
6.2	Définitions	64

6.2.1	Polynômes chromatiques	64
6.2.2	Triangulations, Arbres de cliques	65
6.3	Opérations sur l'arbre de cliques augmenté	67
6.3.1	Fusion de sommets	68
6.3.2	Séparation	69
6.4	Application au calcul du polynôme chromatique	70
6.4.1	Algorithme général	70
6.4.2	Résultats expérimentaux	71
6.5	Conclusions et perspectives	72
6.5.1	Autres calculs à partir de l'arbre de cliques augmenté	72
6.5.2	Quel(s) critère(s) d'évaluation d'une triangulation	73
6.5.3	Modélisation fine de l'arbre de cliques augmenté	73
6.5.4	Autres opérations sur l'arbre de cliques augmenté	74
6.5.5	Extensions de l'arbre de cliques augmenté	74
7	Conclusion et Perspectives	77
	Bibliographie	79
	Annexe	86
A	Collaborations	87
A.1	Télécommunications	87
A.2	Autres collaborations	88
B	Définitions générales et notations	89

Chapitre 1

Introduction

1.1 Contexte de la recherche

Ce document résume une partie des thèmes de recherche que j'ai abordés depuis ma nomination à l'IUT d'Orsay en tant que Maître de Conférences en septembre 1996. Je me suis intégré dans l'équipe *GrafCom* du Laboratoire de Recherche en Informatique (LRI). Au cours de cette période, mes thématiques principales ont évolué de manière progressive, couvrant de fait les divers thèmes abordés par notre équipe. Cette période se divise en deux grandes parties.

Durant la première, entre 1996 et 2002, j'ai principalement abordé les thèmes relatifs au routage, en particulier optique, qui s'inscrivent directement dans la thématique *fondement des communications* de l'équipe. Cette période a été marquée par un investissement important au niveau de l'IUT, en particulier avec la responsabilité pédagogique de la formation par apprentissage. De ce fait, j'ai participé à divers projets de recherche sans y prendre de responsabilité spécifique. C'est durant cette période que j'ai pu collaborer pleinement avec D. Barth, en particulier autour des projets RNRT ROM, IST DAVID dans la thématique du routage optique et du projet RNRT ROCOCO sur la thématique de l'optimisation.

J'ai bénéficié pour l'année 2002-2003 d'un Congé pour Recherche ou Reconversion Thématique (CRCT). Cela m'a permis de me dégager des responsabilités lourdes au sein de l'IUT et de continuer dans de bonnes conditions une reconversion thématique afin d'élargir mes compétences. Cet élargissement rejoint le deuxième thème de recherche de l'équipe : *l'optimisation combinatoire*. Le co-encadrement de la thèse de M. Ben Dhaou avec D. Fayard s'inscrit dans cette optique. Le co-encadrement de celle de L. Gastal avec A. Lisser en est la continuité. Ces deux travaux sont fortement en lien avec les communications optiques car ils s'intéressent à des problèmes de réservation de ressources dans les réseaux.

Le dernier aspect de mes recherches a lui aussi démarré pleinement à cette période charnière consécutive à l'année de CRCT. Il est en lien avec l'algorithmique de graphe, en particulier suite à la collaboration avec M. Diallo sur les problèmes de flots multi-terminaux, et le calcul du polynôme chromatique avec S. Lebresne et K. Nguyen.

1.2 Fil d'Ariane

Plusieurs thématiques et plusieurs problèmes sont abordés dans ce document. Cependant, ils sont tous guidés par les mêmes objectifs. Premièrement, l'outil de modélisation de chacun de ces problèmes est un graphe, les solutions que l'on apporte sont de type algorithmique. Deuxièmement et c'est sans doute le point le plus important, nous avons recherché soit des structures dans les graphes, soit des représentations des propriétés des graphes, qui permettent de présenter chaque problème avec une vision simplifiée, mettant en avant les caractéristiques principales du problème considéré. Ces représentations permettent donc de résoudre relativement efficacement chacun des problèmes posés.

Le fil conducteur de ce document est donc la recherche permanente de structures de graphes simplificatrices adaptées à chacune des situations proposées.

1.3 Organisation du document

Ce document est divisé en deux parties principales, reprenant les deux thématiques énoncées plus haut.

La première partie concerne les résultats que nous avons obtenu au travers de diverses collaborations sur les communications dans les réseaux. Afin de ne pas multiplier les chapitres dans cette partie, nous avons choisi en premier lieu de présenter l'évolution du contexte des réseaux sur lesquels j'ai travaillé durant ces 10 dernières années. En particulier, nous montrons plusieurs facettes que peut recouvrir l'expression **communications optiques**. Dans un deuxième temps, nous avons regroupé les problèmes abordés en deux chapitres :

- le premier s'intéresse à des aspects structurels des graphes utiles pour la construction de protocoles de communication dans les réseaux.
- le second aborde une problématique importante dans le contexte actuel de la recherche de la compétitivité : l'optimisation de ressources.

La deuxième partie s'intéresse à deux **représentations de graphes** qui s'avèrent pertinentes pour les problèmes considérés. Elle présente deux problèmes principaux donnant deux chapitres indépendants. Le choix de l'ordre de présentation de ces problèmes provient de leur développement actuel, les deux ayant été abordés au cours de mon année de CRCT. Le premier est plus abouti, tandis que le second est plus dans une phase que l'on pourrait qualifier d'« exploratoire ».

Le premier problème abordé dans cette partie concerne un très vieux (au sens informatique) problème issu de la théorie de flots : les flots multi-terminaux. Nous nous sommes attachés à montrer la puissance d'un outil permettant de représenter ces types de flots : les arbres de Gomory-Hu, ainsi que leur utilité dans une version paramétrée du problème.

Le second problème présente au travers du calcul du polynôme chromatique une représentation des graphes sous la forme d'arbre de cliques augmenté.

Enfin, une conclusion générale permet d'organiser les diverses perspectives présentées au cours du document. En effet, chaque problématique étant assez spécifique, nous avons

préfére proposer au fur et à mesure les directions de recherche qui nous semblaient les plus pertinentes. Ce chapitre a donc pour but de montrer celles qui nous paraissent les plus prioritaires pour les quelques années à venir.

Première partie

Communications dans les réseaux

Chapitre 2

Contexte des réseaux

Les réseaux en relation avec le monde des communications que nous avons étudiés sont principalement de deux types. Premièrement, nous nous sommes intéressés aux réseaux d'interconnexion pour des machines parallèles, dans la continuité des thèmes abordés durant la thèse [21]. Ensuite, nous avons abordé la modélisation de réseaux de télécommunication. La suite de ce chapitre présente les divers modèles sur lesquels nous avons travaillé au travers de diverses collaborations formelles ou informelles.

2.1 Réseaux d'interconnexion optiques

2.1.1 Réseaux d'interconnexion

Suite à mes travaux de thèse, nous nous sommes penchés sur les réseaux d'interconnexion de machines parallèles. Ceux-ci ont la particularité d'être dédiés à la machine cible et décrivent un certain agencement des processeurs (ou topologie) dont les plus fameux sont l'hypercube, la grille, ou le fat-tree. Dans ce cadre, la problématique principale consiste à concevoir des algorithmes efficaces pour réaliser diverses opérations, soit élémentaires comme la diffusion d'une donnée vers les différents processeurs de la machine, soit de plus haut niveau comme le tri ou des applications numériques comme la multiplication de matrices.

2.1.2 Introduction de l'optique dans ces réseaux

Vers le début des années 1990, les composants optiques ont été introduits dans les réseaux, apportant de nouvelles capacités aux communications en général. Au départ, ce mode de communication n'utilisait qu'une seule longueur d'onde. Il offrait en particulier la possibilité de dupliquer un message « gratuitement », simplement par l'ajout d'un élément physique passif (comme les étoiles passives optiques ou *OPS* [45]). Une branche théorique du parallélisme s'est alors tournée vers la modélisation de machines parallèles obtenues avec ces technologies et a regardé l'apport en terme de puissance de calcul¹ en les comparant aux modèles désormais classiques des PRAMs. Nous pouvons citer par exemple les modèles OCPC ou ORPC [24].

¹c'est-à-dire, le surcoût de simulation d'un algorithme écrit pour un modèle dans l'autre.

C'est durant cette période que nous avons établi une collaboration informelle **Rencontres Optiques-Informatique** avec P. Chavel de l'Institut d'Optique (Laboratoire Charles Fabry) à Orsay afin de comprendre et modéliser les apports de ces *nouvelles* technologies. Plusieurs collaborations que nous avons eues par la suite sont des conséquences plus ou moins directes de cette première prise de contact. Nous revenons en particulier sur le projet RNRT ROM aux paragraphes suivants.

Durant cette première phase, les composants optiques ont principalement servi à améliorer les capacités du réseau d'interconnexion des machines parallèles. En particulier, ils ont permis de mettre en œuvre un certain nombre de topologies difficiles à « câbler » avec des connexions filaires classiques (en cuivre). L'avantage principal de l'optique est la possibilité de densifier les connexions en raison des faibles perturbations intervenant entre deux signaux optiques spatialement proches. Dans le livre que j'ai co-édité avec A. Ferreira sur le sujet [26], à la suite de l'atelier WOCS en 1995 (Workshop on Optics and Computer Science), on s'aperçoit qu'un grand nombre d'applications proposées provenaient de ce domaine. Seules quelques problématiques concernaient les réseaux plus généraux. Dans les réseaux d'interconnexion optiques, on peut citer le réseau POPS, proposé par R. Melhem et son équipe à Pittsburgh [61], pour lequel nous avons étudié des propriétés de plongements de graphes. Nous présentons plus en détail cette problématique et les résultats obtenus au paragraphe 3.1.

L'application de l'optique dans les réseaux dédiés a conduit à l'étude de diverses topologies spécifiques. Par exemple, le réseau d'interconnexion OTIS (Optical Transpose Interconnection System) se compose de deux plans de lentilles permettant de relier une matrice d'émetteurs à une matrice de récepteurs, selon un motif fixé par le nombre et la taille des lentilles. Les travaux effectués par D. Coudert à Nice sur ce dernier modèle sont du même ordre : modélisation de cette topologie par des graphes de Kautz [38].

Cependant, les machines parallèles ont évolué, laissant la place à des grappes de processeurs. L'interconnexion des processeurs ne se fait plus par l'intermédiaire d'un réseau d'interconnexion spécifique, mais grâce à des connexions plus générales. De plus, un nouveau pan de l'algorithmique est revenu au premier plan avec le Global Computing : l'algorithmique distribuée sur un réseau totalement asynchrone. Dans ce cas, le réseau n'est plus du tout dédié, mais les communications utilisent des connexions de type Internet. Cette évolution a été rendue possible grâce à l'introduction des technologies optiques au niveau des routeurs, permettant ainsi d'atteindre des très hauts débits.

2.2 Réseaux de télécommunication

Dans ce paragraphe, nous présentons les divers types de réseaux de télécommunication auxquels je me suis intéressé depuis la fin des années 1990. Dans ce contexte, l'élément qui est rattaché au routeur n'est plus un processeur comme dans les réseaux d'interconnexion de machines parallèles, mais diverses entités qui sont, la plupart du temps, des réseaux à moindre échelle. Ces réseaux communiquent souvent à des débits moins élevés et/ou des technologies différentes. Par exemple, un réseau de type *backbone* pourra utiliser les techniques très haut débit optiques et les sous-réseaux des technologies filaires (cuivre) de moindre débit. Nous nous sommes restreints à l'étude d'un seul niveau de la hiérarchie d'un réseau (aux deux extrêmes, le réseau local et le réseau de type backbone) dans lequel

tous les sommets sont équivalents, i.e., les routeurs sont de même type. Chaque sommet émet un certain nombre de messages à destination d'autres sommets de ce même niveau de modélisation ; il accepte par ailleurs les messages qui lui sont destinés.

2.2.1 Modélisation d'un routeur

De manière simplifiée, on peut représenter le principe d'un routeur par la figure 2.1. Le modèle de routage des messages est simple : ceux-ci arrivent de manière synchrone ou non dans chaque lien du routeur. Celui-ci choisit suivant son algorithme interne le lien de sortie vers lequel il envoie chaque message et, le cas échéant, il récupère les messages qui lui sont destinés. Si on considère qu'un routeur fait l'interface entre le réseau que l'on modélise et un sous-réseau, le message ainsi reçu sera traité dans ce sous-réseau, chargé à ce dernier de l'envoyer vers sa destination réelle finale. Il est quelquefois nécessaire d'effectuer des désencapsulations. Cette dernière partie ne nous concerne pas dans le cadre de nos études.

Nos travaux se situent dans le contexte synchrone. En effet, notre collaboration avec les concepteurs des systèmes optiques dans le cadre du projet RNRT ROM, en particulier avec Alcatel, a montré que cette hypothèse est valide dans le monde des télécommunications, moyennant une resynchronisation au niveau de chaque routeur. Cette opération est gérable « assez » simplement par les composants optiques actuels. Le contexte de travail précise alors les capacités du routeur suivant divers critères.

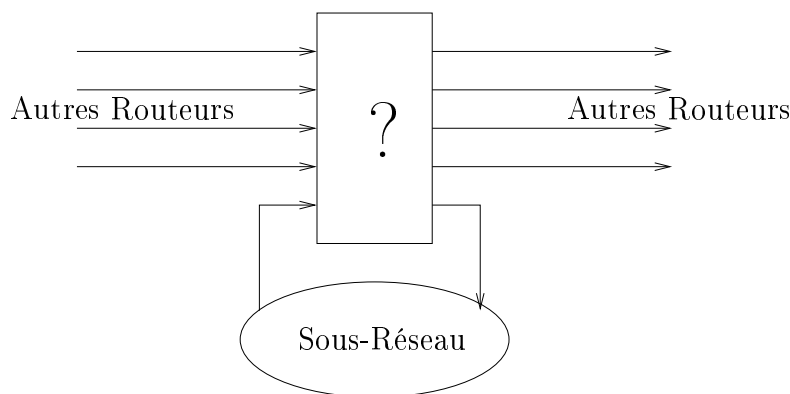


FIG. 2.1 – Modélisation d'un routeur générique

Capacité de stockage du routeur : est-ce que le routeur peut mettre en attente un ou plusieurs messages afin que chaque message prenne la sortie souhaitée ? Dans le contexte classique (non optique), le nombre de messages que l'on peut ainsi stocker est important. Ceci a alors permis de développer des politiques d'attente comme le *store & forward*. Cependant, dans le contexte optique, la taille des messages et le débit sont tels qu'on ne peut stocker qu'un petit nombre de messages en particulier grâce aux lignes à retard. Cette petite capacité est souvent mise à profit pour augmenter un peu le temps de décision de routage, mais elle ne constitue pas un réservoir suffisant de stockage intermédiaire de messages. De ce fait, nous avons considéré cette capacité nulle. Cela a une conséquence non négligeable. L'algorithme de routage ne peut pas envoyer plusieurs messages sur le même lien de sortie. Donc,

en cas de conflit (plusieurs messages voulant emprunter la même sortie), l'algorithme de routage devra expédier au moins l'un des messages concernés vers une sortie non désirée. Nous verrons au chapitre suivant, en particulier au paragraphe 3.3, les options que nous avons choisies dans le cadre du projet RNRT ROM.

Temps de prise de décision de routage : de combien de temps l'algorithme de routage possède-t-il pour prendre sa décision ? Cette question donne d'autres critères d'évaluation d'un algorithme de routage. Elle cherche à résoudre le compromis temps/qualité du routage. On constate que la complexité peut utiliser d'autres critères que ceux communément admis en algorithmique. En effet, si on cherche à mettre en œuvre ces algorithmes sur des cartes dédiées, il faut différencier par exemple les calculs sur des entiers des calculs utilisant des flottants. Une autre opération coûteuse est la lecture directe dans un tableau au travers d'indirections. Cette problématique d'optimisation d'architecture est étudiée dans le cadre du projet RNRT ROM-EO [3]. La problématique principale consiste à quantifier le surcoût induit par une prise de décision non optimale en termes de performances du routage.

Dimensions du problème : le temps de décision du routage est d'autant plus critique que le nombre de messages en entrée est grand. Si le nombre de liens est petit, le nombre de messages différents à router peut être très important si on utilise les technologies multi-longueur d'onde WDM (Wavelength Division Multiplexing). Sur la durée de nos recherches dans le domaine, le nombre de longueurs d'ondes sur les liens des réseaux est passé de quelques unités à quelques dizaines. Si on se réfère aux possibilités actuelles (80 longueurs d'ondes commercialisées, et plus de 300 démontrées²), ce nombre ne constitue plus une ressource critique des réseaux. Cette évolution a pu se faire grâce en particulier à l'évolution des technologies des lasers, et surtout des fibres optiques dont le traitement permet de compenser les effets non linéaires de la déformation du signal sur les longues distances.

Capacités additionnelles des routeurs : avec l'avènement des composants optiques, certains routeurs pourraient intégrer de nouvelles fonctionnalités. Un exemple simple est la duplication de messages : un message arrivant à une entrée peut être envoyé sans surcoût significatif vers plusieurs sorties différentes. Nous étudierons ce modèle au paragraphe 3.2.

2.2.2 Communications multipoints

Au cours des dernières années, les débits offerts par les réseaux de type internet ont permis l'émergence de nouvelles applications. En effet, précédemment, celles-ci étaient principalement basées sur les communications point-à-point entre deux interlocuteurs

²Dans les records (qui sont faits pour être battus), Alcatel a démontré en 2004 qu'il était possible d'atteindre 10 Terabit/s en multiplexant 256 canaux opérant chacun à 40 Gb/s sur une distance de 100 km. Par ailleurs, Alcatel a aussi réussi à transmettre 3 Terabit/s sur une distance de 7300 km, en multiplexant 300 canaux opérant chacun à 10 Gb/s.

particuliers. Maintenant, elles permettent de faire collaborer plusieurs (> 2) interlocuteurs dans un même groupe. La problématique ainsi soulevée évolue de la manière suivante : peut-on faire collaborer plusieurs entités plus efficacement qu'en effectuant des communications de type point-à-point ? Nous présentons au paragraphe 4.1 un problème d'allocation de ressources dans le cadre des communications multipoints.

2.2.3 Dimensionnement de réseaux

Un des problèmes essentiels des opérateurs de télécommunication est de bien dimensionner leurs réseaux : il faut satisfaire les demandes des clients pour un coût global minimum. Par ailleurs, diverses contraintes peuvent s'ajouter, augmentant la diversité des problèmes. Par exemple, un client peut demander à ce que son réseau soit sécurisé : il doit exister deux chemins arc (ou sommet)-disjoints entre chaque paire de sommets.

2.2.4 Évolution

Depuis 2003, nous étudions un nouveau type de problèmes relié aux réseaux : la notion de robustesse dans un contexte de réseau en évolution. Le cadre de la problématique est simple. Le réseau que l'on considère n'est plus fixe comme précédemment, mais en évolution. Des ressources (sommets et/ou liens) apparaissent ou disparaissent à différentes étapes. L'évolution est lente et son utilisation doit être prévue pour les différentes étapes. Dans ce cadre, nous essayons de prévoir une planification de l'utilisation d'un réseau pour des opérations simples d'allocation. De ce fait, une ressource utilisée à une étape donnée peut s'avérer peu intéressante du fait de l'ajout de nouvelles possibilités à l'étape suivante. Cependant, le fait de ne plus utiliser une ressource induit des pénalités dans le coût global d'utilisation. Ainsi, un problème simple dans le contexte classique comme la recherche d'un plus court chemin s'avère plus complexe dès que l'on considère plusieurs étapes.

2.3 Apports de l'optique

Au travers des différents exemples que l'on développe dans cette partie, nous montrons que l'introduction de l'optique dans les communications permet, tant au niveau parallélisme qu'au niveau des réseaux optiques à très haut débit, de renouveler les problématiques de recherche.

D'un point de vue technologique, l'apport est clair : les performances potentielles des systèmes utilisant l'optique pour effectuer leurs communications sont ainsi augmentées de manière considérable. L'enjeu consiste donc à bien utiliser ces moyens de communication.

Plusieurs choix s'offrent alors à l'utilisateur. Premièrement essayer d'utiliser les techniques éprouvées en les simulant sur le nouveau système. C'est souvent l'option choisie dans les systèmes parallèles. Cela conduit à des recherches de plongements d'architectures simples, pour lesquelles on connaît des algorithmes efficaces, dans l'architecture nouvelle. C'est le travail que nous avons effectué dans le cadre du réseau POPS (Paragraphe 3.1). La deuxième option consiste à définir entièrement une solution adaptée au nouveau système. C'est celle qui a été choisie pour le routage eulérien défini dans le cadre des télécommunications optiques (Paragraphe 3.3).

Par ailleurs, on remarque que les ressources apportées par ces nouvelles technologies permettent d'imaginer des applications plus complexes que précédemment, mettant en jeu un grand nombre d'utilisateurs, des volumes de données importants et/ou une puissance de calcul importante. L'émergence des systèmes *pair-à-pair* n'aurait sans doute pas été possible sans l'apport des communications à très haut débit.

2.4 Organisation de cette partie

Dans les deux chapitres suivants, nous présentons nos divers résultats relatifs aux communications dans les réseaux. Afin de ne pas multiplier inutilement le nombre de chapitres, nous avons choisi de les regrouper en deux volets.

Le premier regroupe les problèmes pour lesquels nous avons étudié les aspects structurels dans différents types de réseaux. Le second regroupe divers problèmes d'optimisation de ressources qui interviennent dans ces mêmes réseaux.

Chapitre 3

Aspects structurels des communications dans les réseaux

Le but de ce chapitre est de mettre en évidence un élément important de notre recherche dans le domaine des réseaux. Il s'agit, au travers d'exemples variés, de trouver les structures dans les réseaux adaptées pour réaliser des communications de manière simplifiée. Dans ce cadre, le plongement de l'anneau ou du tore dans le réseau d'interconnexion POPS (paragraphe 3.1) permet d'émuler efficacement n'importe quel algorithme conçu pour des machines parallèles dans l'architecture parallèle POPS. De même, dans l'échange total perpétuel présenté au paragraphe 3.2, nous essayons de partitionner le réseau en différents sous-réseaux pour lesquels le problème initial se subdivise en sous-problèmes élémentaires. Enfin, dans le cadre du projet RNRT ROM, nous avons défini une méthode de routage basée sur les circuits eulériens ; nous nous sommes penchés sur la recherche de circuits eulériens possédant de bonnes propriétés de routage. Ces propriétés s'expriment en terme de diamètre que l'on doit définir de manière spécifique (paragraphe 3.3).

3.1 Plongements dans le POPS

Une des premières applications des technologies optiques dans les communications concerne les machines parallèles. Il s'agissait alors de remplacer les réseaux d'interconnexion des machines parallèles par des structures potentiellement plus efficaces pour l'envoi des données. L'un des composants optiques phare pour effectuer les communications est l'étoile passive optique (OPS) [31]. Ce composant permet d'effectuer des diffusions de messages à moindre coût : un message envoyé par l'expéditeur est reçu simultanément par tous les destinataires. Si on envoie un message de type point-à-point dans un tel réseau d'interconnexion, il suffit de préciser le destinataire dans l'entête du message. Cependant, un réseau n'utilisant qu'un seul OPS ne peut pas relier un grand nombre de machines en particulier pour des problèmes d'atténuation de signal. Diverses propositions ont été suggérées afin de pallier ce problème. Des généralisations des réseaux « intéressants » pour les architectures parallèles comme le réseau de *de Bruijn* ont été proposées. Parmi les propositions, nous nous sommes intéressés au modèle *Partitioned Optical Passive Star* qui avait été défini suite à une collaboration entre des informaticiens et des opticiens à l'université de Pittsburgh [61].

Lorsque l'on définit un nouveau modèle d'architecture parallèle, il est important de

préciser comment on peut y développer des applications. Une grande partie des applications parallèles utilisent des communications dites « régulières », basées sur des topologies simples. Par exemple, les algorithmes de diffusion de données ont été étudiés en profondeur sur des réseaux d'interconnexion simples comme l'anneau, le tore ou l'hypercube [84]. Il est important de voir comment on peut « configurer » la topologie initiale en ces réseaux simples. C'est le but principal des plongements d'une architecture dans une autre. Dans ce cadre, G. Gravenstreter et R. Melhem ont proposé des plongements d'anneaux et de tores dans le POPS [60]. Avec A. Ferreira et J. Cohen, nous avons étudié ce problème au travers d'une autre modélisation du POPS [22,25]. Cette modélisation a permis d'améliorer les résultats précédents de manière significative.

Dans la suite de ce paragraphe, nous présentons rapidement l'architecture POPS et la modélisation choisie (paragraphe 3.1.1), avant de présenter nos différents résultats au paragraphe 3.1.2.

3.1.1 Définitions et modèles

Le cœur de l'architecture du POPS est l'étoile passive optique (OPS). Cet élément optique permet de connecter N nœuds (on parle alors d'un OPS de degré N). Chaque transmetteur envoie son message au travers de l'OPS et chaque destinataire reçoit alors le message directement avec un taux d'atténuation de $1/N$ (cas idéal). Afin d'interconnecter un grand nombre de nœuds, on peut utiliser plusieurs OPS et les configurer de la manière suivante.

Définition 3.1 *Le réseau **Partitioned Optical Passive Star POPS**(d, g) est composé de $N = dg$ nœuds et g^2 OPS de degré d (voir figure 3.1(a)). Les nœuds sont divisés en g groupes de taille d .*

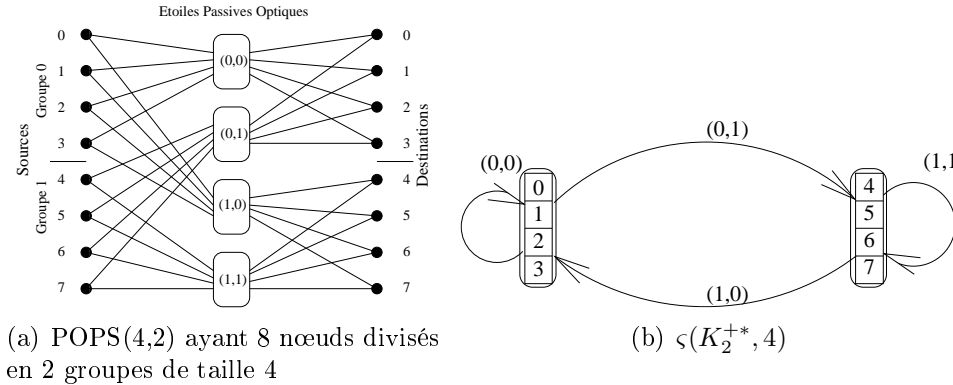


FIG. 3.1 – Illustration de la propriété 3.1 : deux représentations du même réseau.

Ce réseau est modélisé par une classe particulière d'hypergraphes orientés définie par A. Ferreira [33] : les empilements de graphes.

Définition 3.2 *Soit $G = (V, E)$ un graphe et m un entier strictement positif. On appelle **empilement du graphe** G , noté $\varsigma(G, m) = (V_\varsigma, E_\varsigma)$, l'hypergraphe défini par :*

1. $V_\varsigma = \{0, \dots, m - 1\} \times V$;

2. soit π la fonction $V_\zeta \longrightarrow V$ telle que $\pi([i, v]) = v$, alors l'ensemble des hyperarêtes est défini par :

$$E_\zeta = \{\pi^{-1}(a) \cup \pi^{-1}(b) \mid (a, b) \in E\}$$

On appelle alors m le **facteur d'empilement**. Si $G = (V, A)$ est un graphe orienté, on peut définir de même un **empilement d'un graphe orienté** (voir figure 3.1(b)). Les sommets sont définis de la même manière. Chaque hyperarc a_ζ issu d'un arc (u, v) a pour ensemble source $\pi^{-1}(u)$ et ensemble destination $\pi^{-1}(v)$.

La propriété principale des POPS est qu'ils peuvent être modélisés par un empilement de graphe orienté particulier.

Propriété 3.1 ([25]) *Le réseau POPS(d, g) peut être représenté par l'empilement du graphe orienté $K_g^{+*} : \zeta(K_g^{+*}, d)$.*

La notion de plongement est bien définie dans le cadre des graphes [67]. Il peut être défini comme un couple d'applications. La première affecte à chaque sommet du premier graphe un sommet dans le second. La seconde envoie les arêtes du premier graphe vers les chemins du second, et ceci de manière compatible avec la première fonction. On étudie alors le surcoût de l'émulation d'un réseau par un autre. Les principaux paramètres sont la dilatation, la charge et la congestion. La **dilatation** indique la longueur maximale d'un chemin image de la seconde application, tandis que la **charge** correspond au nombre maximum de sommets (charge-sommets) qui sont affectés à un même sommet du réseau cible. La **congestion** correspond au nombre maximum de chemins utilisant une même arête du réseau cible.

Dans notre cas, on cherche à émuler un réseau point-à-point sur un POPS particulier. On définit une extension des plongements d'un graphe dans le POPS. Le réseau étant de diamètre 1, le paramètre de dilatation est toujours égal à 1. Par contre, le paramètre important est la **congestion des OPS**, c'est-à-dire, le nombre de chemins traversant un même OPS. En effet, si k chemins traversent un même OPS, la simulation d'une étape de communication du réseau initial par le POPS nécessitera k étapes sur le POPS.

3.1.2 Résultats

Du fait de la modélisation choisie, la méthodologie générale s'appuie sur le théorème suivant, illustré par la figure 3.2.

Théorème 3.1 ([22]) *Soit α et g deux entiers, G un graphe ayant αg sommets et φ un plongement de G sur K_g^{+*} de charge α et de congestion 1. Il existe alors un plongement Φ de G sur POPS(α, g) de congestion 1 et de charge 1.*

De ce résultat, on en déduit des plongements simples de tout type d'anneau dans n'importe quel POPS. En effet, il suffit de rechercher un plongement dans le graphe complet K_n^{+*} . Comme celui-ci est eulérien, ce plongement peut se faire de manière optimale [25]. La technique mise au point dans [60] est beaucoup plus complexe et met en jeu des algorithmes avancés pour équilibrer le plongement.

Le cas des tores est plus complexe. Ils ont aussi été étudiés dans [60]. Cependant, les auteurs se sont restreints au cas du tore carré dont la longueur du côté est une puissance de

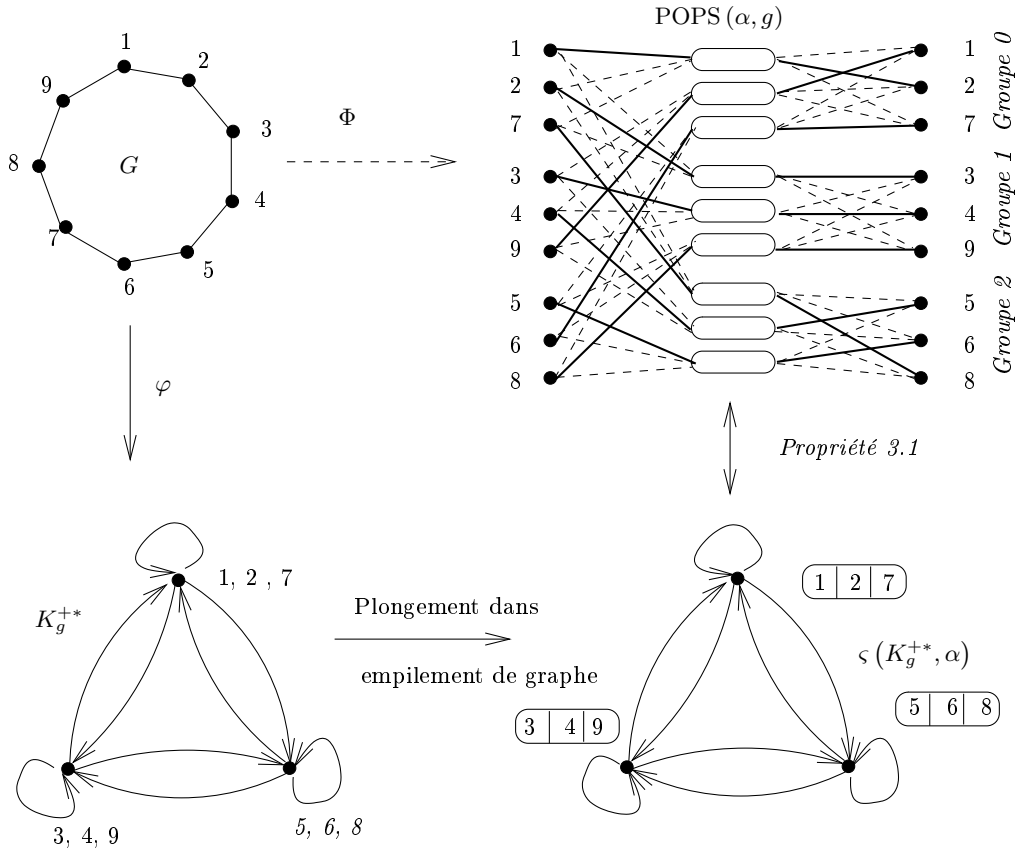


FIG. 3.2 – Illustration du théorème 3.1

2. Nous avons étendu ce résultat à toute dimension de tore carré. Dans notre travail, nous avons cherché à équilibrer les plongements dans les deux directions principales du tore. En effet, le but des plongement est d'émuler des algorithmes parallèles conçus pour des tores (ou des grilles) sur la topologie spécifique POPS. Or, la plupart de ces algorithmes effectuent des communications par phases en privilégiant à chaque fois une direction.

3.1.3 Conclusions et perspectives de ce travail

Dans ce paragraphe, on a mis en évidence une modélisation du réseau d'interconnexion du POPS. Cette modélisation permet en particulier de rechercher des plongements de réseaux classiques dans cette architecture. En utilisant la méthodologie générale présentée au théorème 3.1, nous avons aussi étendu ces résultats aux graphes de *de Bruijn*.

Diverses questions se posent encore. En particulier, les problèmes de décision liés au plongement sont NP-complets [7]. À quelles conditions le restent-ils si on s'intéresse au POPS? En effet, le graphe complet est très simple et offre beaucoup de possibilités. Si la question de la dilatation minimale n'a pas de sens dans notre contexte, ceux de la charge et de la congestion sont plus pertinents.

Cependant, les architectures parallèles basées sur les étoiles passives optiques n'ont pas encore été construites. Les avancées en optique se sont avérées importantes dans le monde des télécommunications. Cela rend sans doute moins critiques les extensions possibles que l'on peut apporter à ce travail. Néanmoins, divers travaux récents ont montré l'intérêt

de l'utilisation du POPS comme architecture parallèle, en montrant comment on peut réaliser diverses opérations haut niveau sur cette architecture, comme le tri [40] ou le routage des permutations [78, 79].

Dans la suite du chapitre, nous nous intéressons à deux autres problèmes structurels davantage basés sur les réseaux optiques généraux, bien que les solutions apportées puissent être utilisées dans des réseaux d'interconnexion.

3.2 Échange total périodique

Comme nous l'avons montré au chapitre précédent, l'introduction de l'optique dans les télécommunications a permis de considérer de nouvelles fonctionnalités dans les capacités d'un routeur. Dans ce paragraphe, nous considérons qu'un routeur peut dupliquer sans surcoût les messages qui transitent par lui. Cette capacité est assez simple à réaliser avec les technologies optiques. Avec D. Barth, nous avons considéré un cas particulier de configuration du routeur : elle reste stable pendant tout le processus. On parle alors de réseau commuté. Cette hypothèse peut paraître un peu irréaliste. Cependant, elle se justifie parfaitement lors d'un processus de contrôle global d'un réseau : une des étapes (TDM), ou une des longueurs d'onde (WDM) est dédiée au contrôle du réseau. Les opérations doivent y être simples. La commutation prédéfinie s'avère donc utile dans ce cas. Les réseaux ciblés par cette application doivent donc supporter une forte synchronisation de l'envoi des messages. Ce problème vise donc les réseaux d'interconnexion de machines parallèles et les réseaux locaux de faible envergure.

Dans ce type de réseau, nous avons étudié une opération de contrôle périodique du réseau : le problème d'échange total périodique. Les contrôles périodiques de réseau sont à la base de la gestion d'un certain nombre d'applications distribuées, comme les applications de vidéo à la demande [68] ou l'équilibrage de charge dans les grilles de calcul [73]. Toutes ces applications ont en commun la recherche d'une qualité de service garantie entre les différents acteurs. Ce paragraphe résume les résultats de [8].

Par la suite, nous précisons le contexte de notre étude au paragraphe 3.2.1, avant de donner nos résultats aux paragraphes suivants.

3.2.1 Définitions

Le modèle de routeur est spécifique à ce problème. On suppose que chaque nœud du réseau envoie périodiquement un message (son état) aux autres nœuds en utilisant les capacités des routeurs. Chaque routeur fonctionne avec le mode suivant :

Synchrone : toutes les communications sont effectuées par étape.

Capacité 1 : les liens ne peuvent envoyer qu'un seul message à la fois.

Δ -port : un routeur peut envoyer des messages sur tous ses liens de sortie.

Unicité : chaque message n'est émis qu'une seule fois.

Contrôle du routeur : les destinations d'un message sont encodées dans son entête.

L'entête est recalculée quand le message est réexpédié. Le message est détruit s'il n'a plus de destinataire.

Absorption : si un message traverse un routeur destinataire, celui-ci en récupère une copie.

Une configuration d'un routeur est représentée dans la figure 3.3.

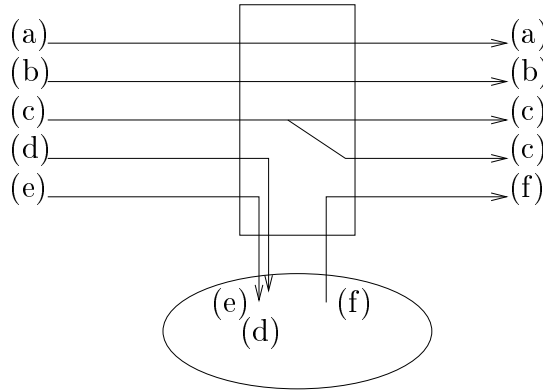


FIG. 3.3 – Commutation d'un routeur. Les messages (d) et (e) sont absorbés par le routeur, le message (f) est inséré dans le réseau et le message (c) est dupliqué vers deux sorties différentes.

Une première solution consiste à réaliser périodiquement un algorithme d'échange total [84]. Cependant, dans la majeure partie des réseaux, cette opération nécessite un réarrangement des routeurs, c'est-à-dire, les messages doivent être réexpédiés et le réseau reconfiguré. Cela n'est donc pas compatible avec le modèle choisi. De plus, nous avons montré que les deux problèmes peuvent être différenciés selon des paramètres décrits plus loin.

Un moyen simple pour résoudre ce problème est de considérer un circuit couvrant tous les nœuds. Chaque nœud doit alors émettre toutes les k étapes, où k est la longueur du circuit. Cette solution n'est pas efficace car elle n'utilise pas tous les liens du réseau. Par ailleurs, le délai d'attente d'un message peut aussi être important.

Dans ce contexte, la problématique consiste à minimiser les deux critères :

la période : la durée maximale entre deux envois de messages par un même nœud ;

la fenêtre : le temps maximum pris par un message pour atteindre toutes ses destinations.

Afin d'atteindre ce but, notre stratégie consiste à décomposer le réseau principal en sous-réseaux indépendants sur lesquels le problème se résout simplement. Si on considère une commutation particulière du réseau, on remarque que ce dernier se décompose naturellement en plusieurs sous-réseaux distincts. On peut prendre deux exemples simples.

- Premièrement, si le réseau contient un circuit eulérien, on peut numéroter les arcs du réseau entre 0 et $m - 1$ (m est le nombre d'arcs). Comme nous l'illustrons au paragraphe suivant (sur le routage eulérien), on peut configurer chaque routeur de sorte que l'arc numéro i soit connecté à l'arc numéroté $(i + 1) \bmod m$. Dans ce cas, il n'y a qu'un seul sous-réseau : le circuit eulérien.
- Un deuxième exemple n'est possible que dans les graphes admettant une décomposition en circuits hamiltoniens, comme certains tores. Il suffit de numéroter indépendamment chaque circuit hamiltonien et effectuer les commutations en suivant

l'orientation des circuits. Dans ce cas, l'ensemble des sous-réseaux correspond à l'ensemble des circuits hamiltoniens.

Ces deux exemples possèdent une propriété essentielle pour effectuer notre opération d'échange total périodique : chacun des sous-réseaux de la décomposition couvre l'ensemble des sommets du réseau. Cette propriété est essentielle. En effet, dans un sous-réseau non couvrant, il est impossible d'atteindre tous les sommets. Comme nous le voyons par la suite, il existe d'autres décompositions admissibles pour notre problème.

Le but de ces décompositions est de donc diviser le problème d'échange total périodique en plusieurs problèmes de type k -MTA (*Many-To-All*). Afin d'effectuer chaque opération, chaque sous-structure doit couvrir tous les nœuds du réseau.

Définition 3.3 *Étant donné un réseau commutable $G = (V, E)$, une **instance d'un k -MTA** est donnée par un ensemble $V_i \subset V$ de taille k contenant les nœuds qui doivent émettre des messages vers tous les autres.*

*Un **schéma périodique** d'un nœud v de V_i est une suite infinie périodique d'entiers qui représente la suite des délais entre deux envois consécutifs de messages par v :*

$$\mathcal{D}(v) = d_0 d_1 \dots d_h (x_1 x_2 \dots x_t)^\infty$$

*Un **schéma périodique global** est la donnée d'une commutation du réseau et des schémas périodiques pour chaque nœud du réseau.*

On peut remarquer que les structures couvrantes recherchées dépendent étroitement de l'opération que l'on souhaite réaliser sur le réseau. En effet, si on souhaite effectuer une diffusion, même périodique, d'une donnée depuis un nœud source vers tous les autres, il suffit de se donner un seul arbre couvrant le réseau. Afin d'améliorer l'efficacité de ces communications, on peut couvrir le réseau par plusieurs arbres de recouvrement disjoints (présentés pour l'hypercube dans [84]). Dans le cas de regroupement de données, on peut utiliser la même structure que pour la diffusion. L'échange total simple, c'est-à-dire non périodique, peut s'effectuer par une séquence de concentration, puis de diffusion. Il est alors important de noter que la taille des messages véhiculés est très importante et ne peut en aucun cas être utilisée avec nos hypothèses de ce travail. En effet, cela supposerait que les routeurs puissent agréger les messages en provenance des différentes entrées de manière efficace. En ajoutant cette possibilité technologique et en supposant que la taille des messages ne soit pas un problème, on peut alors définir un schéma simple d'échange total périodique si on possède deux arbres couvrants disjoints. Le premier permet d'effectuer une concentration des informations vers un nœud particulier, le second étant chargé de la diffusion du message ainsi aggloméré. La période est alors optimale, puisqu'à chaque étape de nouveaux messages peuvent être envoyés par tous les sommets. La fenêtre est alors égale à la somme des hauteurs des deux arbres.

En suivant le modèle choisi, une couverture des sommets par des cycles disjoints permet de pas augmenter la taille des messages envoyés, comme nous allons le voir par la suite.

3.2.2 k -MTA sur des réseaux simples

Les cas simples permettent souvent de définir les entités élémentaires du cas général. Pour notre problème, ces éléments simples sont les anneaux, les arbres et une combinaison

particulière de ces deux structures appelée **poulpe**. Les résultats suivants servent de base aux résultats sur les graphes généraux présentés au paragraphe 3.2.3.

Pour un arbre enraciné tel que toutes les arêtes sont orientées vers les feuilles, on considère l'instance où seule la racine peut émettre. Dans ce cas, la période vaut 1, et la fenêtre correspond à la longueur de la plus grande branche de l'arbre.

En ce qui concerne l'anneau orienté à n éléments, on peut définir des algorithmes périodiques optimaux pour chaque densité d'émetteurs.

Lemme 3.1 ([8]) *Soit $R(n)$ l'anneau orienté avec n sommets et k un entier. Pour n'importe quelle instance V_i du problème k -MTA, il existe un algorithme périodique de fenêtre $n - 1$ (optimal) et de période p tel que :*

- $p = n - 1$ si $k = n$;
- $p = k + \left\lceil \frac{n \bmod k}{n/k} \right\rceil$ si $k < n$.

On définit ensuite les poulpes comme les réseaux combinant les arbres et les anneaux de la manière suivante.

Définition 3.4 *Un poulpe θ de taille n et de hauteur d est un graphe orienté composé de :*

- un anneau orienté de taille n appelé la **tête**, $Head(\theta)$, et
- un ensemble d'arbres orientés, que l'on nomme les **tentacules** de θ , de hauteur au plus d enracinés sur un sommet de la tête. On note $depth(\theta)$ la hauteur maximale de ces arbres.

La famille des poulpes est aussi caractérisée par la famille des graphes dont tous les sommets sauf éventuellement 1 (dans le cas des arbres) ont pour degré entrant 1.

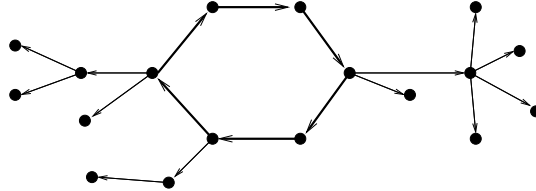


FIG. 3.4 – Un poulpe de taille 6 et de hauteur 2. La tête est en gris.

Une conséquence directe du lemme 3.1 est la résolution du problème k -MTA sur les poulpes en suivant les mêmes principes.

Lemme 3.2 *Soit θ un poulpe de taille n et de hauteur h , et k un entier. Pour n'importe quelle instance V_i du problème k -MTA sur θ où $V_i \subset Head(\theta)$, il existe un algorithme périodique de fenêtre $n + h - 1$ (optimal) et de période p tel que :*

- $p = n - 1$ si $k = n$ et $h = 0$;
- $p = k + \left\lceil \frac{n \bmod k}{n/k} \right\rceil$ si $k < n$.

De ce lemme, on voit que les éléments source ne peuvent se situer que sur la tête des poulpes. En effet, on ne peut atteindre tous les sommets d'un poulpe qu'à partir des sommets de la tête.

3.2.3 Décomposition d'un réseau général pour l'échange total

De ces premiers résultats, on déduit le type de décomposition que l'on recherche pour un réseau donné en vue de réaliser l'échange total (n -MTA) périodique. L'objectif est donc de couvrir le réseau par des poulpes de sorte que :

1. Les têtes des poulpes couvrent tous les sommets du réseau. Un sommet peut éventuellement être couvert par plusieurs têtes.
2. Chaque poulpe couvre tous les sommets du réseau afin d'effectuer localement un k -MTA périodique.

De plus, si un sommet appartient à plusieurs têtes de poulpe, il faut sélectionner le poulpe vers lequel ce sommet va émettre ses données. Pour chaque poulpe, on définit ainsi une instance particulière : si on note f la fonction qui attribue un poulpe à chaque sommet du graphe, l'instance définie pour un poulpe donné θ est $f^{-1}(\theta)$. On appelle cette décomposition une **couverture par des poulpes**, qui est entièrement définie par \mathcal{O} et f , où \mathcal{O} est l'ensemble des poulpes de la couverture. La figure 3.5(b) présente une telle couverture.

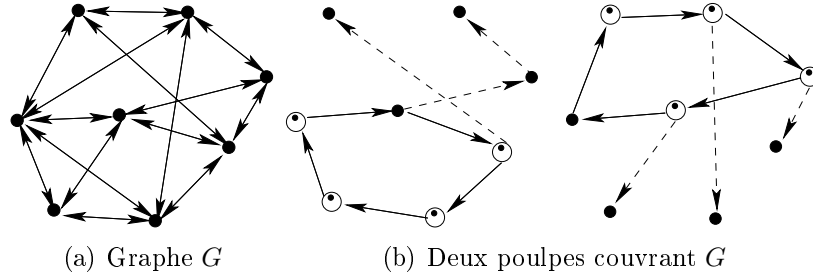


FIG. 3.5 – Une couverture par des poulpes d'un graphe. Chaque sommet de G est marqué dans l'un des deux poulpes (fonction f).

Une fois cette décomposition obtenue, le protocole général consiste à appliquer le protocole défini pour chaque poulpe par le lemme 3.2. En effet, chaque poulpe de la décomposition peut effectuer son protocole indépendamment des autres. On en déduit le résultat général suivant.

Théorème 3.2 ([8]) *Soit $G = (V, E)$ un réseau et (\mathcal{O}, f) une couverture par des poulpes de G . Il existe un algorithme périodique réalisant l'échange total sur G avec une période au plus égale à*

$$\max_{\theta \in \mathcal{O}} (\text{depth}(\theta) + |\text{Head}(\theta)|)$$

et une période égale à

$$\max_{\theta \in \mathcal{O}} \left(|f^{-1}(\theta)| + \left\lceil \frac{|\text{Head}(\theta)| \bmod |f^{-1}(\theta)|}{|\text{Head}(\theta)|/|f^{-1}(\theta)|} \right\rceil \right).$$

La période de l'algorithme est au moins $\lceil |V|/\Delta \rceil$, où Δ est le degré maximum du réseau. Par ailleurs, si on connaît l'ensemble des poulpes de la décomposition (\mathcal{O}), on peut choisir la fonction f de telle sorte que la période soit minimale. Cependant, nous ne savons pas encore si trouver une famille de poulpes qui permet de minimiser la période est un problème polynomial.

3.2.4 Décomposition de réseaux particuliers

Nous résumons les résultats que nous avons obtenus pour quelques classes de graphes orientés symétriques. Ceux-ci sont de deux ordres. Premièrement, ils peuvent être une application du théorème sur les décompositions dans le cas où les graphes possèdent une décomposition hamiltonienne. Le second cas s'intéresse à des solutions *ad hoc*, en particulier dans le cas des tores.

Théorème 3.3 *Soit G un graphe régulier de degré δ de taille n possédant une décomposition hamiltonienne, et k un entier. Il existe un algorithme périodique réalisant un k -MTA pour n'importe quelle instance de fenêtre $n - 1$ et de période :*

$$\left\lceil \frac{k}{\delta} \right\rceil + \left\lceil \frac{n \bmod \lceil k/\delta \rceil}{n/\lceil k/\delta \rceil} \right\rceil.$$

Si on ne considère que le cas de l'échange total périodique, on obtient le tableau suivant.

Réseau	Fenêtre	Période
Hypercube de dimension n , n impair	$2^n - 1$	$\left\lceil \frac{2^n}{n} \right\rceil + \frac{2^n \bmod \lceil 2^n/n \rceil}{2^n/\lceil 2^n/n \rceil}$
Anneau $A(n)$	$n - 1$	$\left\lceil \frac{n}{2} \right\rceil + \frac{n \bmod \lceil n/2 \rceil}{n/\lceil n/2 \rceil}$
Tore $T(n, p)$	$np - 1$	$\left\lceil \frac{np}{4} \right\rceil + \frac{np \bmod \lceil np/4 \rceil}{np/\lceil np/4 \rceil}$
Tore $T(4, q)$, $q \geq 3$	$q + 3$	q
Tore $T(4p\alpha, 4q\alpha)$, $\text{pgcd}(p, q) = 1$	$4pq\alpha^2 + 3$	$4pq\alpha^2$

Dans les deux derniers cas, nous avons déterminé une couverture par des poulpes particulière, pour laquelle les têtes sont disjointes. On arrive à construire ensuite des arborescences à partir de ces têtes qui soient compatibles entre elles. La figure 3.6 présente la décomposition du tore $T(4, 2q)$, d'autres exemples sont donnés dans [8]. Cette décomposition permet de diminuer de manière significative la taille de la fenêtre (divisée environ par 4) sans altérer la qualité de la période.

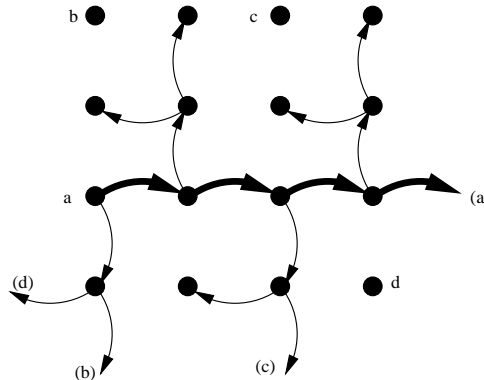


FIG. 3.6 – Poulpe permettant de couvrir le tore $T(4, 2q)$; il suffit de copier 4 fois ce poulpe pour obtenir la couverture.

3.2.5 Conclusions et perspectives de ce travail

Dans ce travail, nous avons exhibé une configuration intéressante pour effectuer l'opération d'échange total perpétuel. Nous avons mis en évidence une nouvelle décomposition d'un graphe en différents poulpes couvrants. Cette décomposition nous a permis d'obtenir un protocole pour effectuer notre opération.

Plusieurs questions ressortent de ce travail. En particulier, est-il possible d'étendre ce dernier type de décomposition à d'autres réseaux réguliers ?

On peut remarquer que notre protocole d'échange total nécessite un réseau régulier, stable. Une évolution possible dans ce thème consiste à rechercher des structures couvrantes moins rigides du graphe qui permettent de simuler un réseau régulier. Des notions particulières de plongement doivent alors être définies.

Une autre voie consiste à rechercher la construction d'une couverture par des moyens algorithmique. En effet, nous n'avons donné ici que des construction ad hoc de cette couverture. Une recherche à plus long terme concernera les aspects algorithmique distribuée pour la construction de cette couverture.

3.3 Routage dans les réseaux de télécommunication optiques

Le monde des télécommunications a toujours essayé d'utiliser les dernières technologies innovantes. En particulier, dès que les composants optiques ont commencé à se miniaturiser, ceux-ci sont naturellement apparus dans ce type de réseau. Dans ce paragraphe, nous présentons quelques résultats issus du projet RNRT ROM (Routage Optique Multiservice) dans lequel les aspects théoriques et pratiques ont été abordés, comme nous l'avons précisé au chapitre précédent [62]. Au sein de ce projet, nous nous sommes plus particulièrement intéressés à la mise en place d'un mode de routage simple, avec des garanties de performances.

Dans ce paragraphe, nous nous limitons à la présentation de problèmes structurels relatifs au routage eulérien *simple* principalement issus de l'étude [9]. Cette technique de routage a été améliorée tout au long du projet, en particulier avec le *routage eulérien avec raccourcis*. Nous avons présenté plusieurs études de performance de ces deux modes de routage, en particulier au travers de la collaboration avec J-M. Fourneau de l'université de Versailles [11, 15, 53, 75].

Dans la suite, nous présentons rapidement le mode de routage que nous avons étudié au paragraphe 3.3.1 avant de présenter les résultats structurels que nous avons obtenus au paragraphe 3.3.2.

3.3.1 Les principes du routage eulérien

Dans le cadre du projet ROM, nous nous sommes concentrés précisément les différentes garanties que l'on pouvait attendre d'un algorithme de routage, et les conséquences que l'on doit en tirer dans le contexte des routeurs optiques. Un des points clé que nous avons cherché à résoudre est le problème de la **garantie de terminaison** : tout message

émis dans le réseau doit arriver à destination au bout d'un temps borné. La contrainte des routeurs optiques oblige à considérer les routages sans stockage intermédiaire : tout message entrant dans un routeur doit être envoyé sans délai vers une sortie (ou éventuellement être *absorbé* par le routeur si le message est arrivé à destination). Dans ce cadre, un routeur ne peut pas dupliquer les messages comme au paragraphe précédent. Reprenons le schéma d'un routeur générique présenté à la figure 2.1. Le routeur devra réaliser une permutation des entrées vers les sorties (si aucun message n'est arrivé à destination et que toutes les entrées sont pleines). Le problème consiste alors à gérer la **congestion** : que faire quand plusieurs messages doivent emprunter le même lien de sortie ?

Diverses méthodes ont été mises au point pour gérer la congestion. La plus simple consiste à donner une priorité plus importante aux messages qui se rapprochent de leur destination finale, les autres messages étant envoyés vers d'autres sorties. C'est le principe de base des routages dits à *déflexion* (ou *hot-potato*) [15, 34, 87]. Cependant, même si les propriétés statistiques de ce mode de routage sont globalement satisfaisantes, celui-ci ne permet pas de garantir la terminaison. Des techniques de priorité peuvent être ajoutées pour améliorer cette propriété, en particulier en fonction du temps déjà passé dans le réseau [34, 48].

Une autre famille d'algorithmes de routage, dans laquelle s'inscrit notre proposition, est le **routage à convergence** [81, 92]. Le principe est de faire suivre aux messages des directions qui vont les rapprocher de leur destination finale suivant une certaine norme. Dans ce cadre, Feige [48] montre qu'il est possible d'assurer une garantie de terminaison en $O(n^{3/2})$ sur n'importe quel graphe. La proposition faite dans le cadre du projet RNRT ROM est d'orienter les messages par un circuit eulérien du graphe sous-jacent au réseau. Dans la version simple de ce routage, les routeurs ont un rôle très réduit : ils peuvent être vus comme commutateurs entre les entrées et les sorties du routeur, la seule fonction consistant alors à filtrer les messages arrivés à destination. Il est cependant possible d'améliorer les performances de ce mode de routage en utilisant pleinement les fonctions du routeur avec le routage eulérien avec raccourcis [53].

Dans la suite, on considère le réseau de routeurs représenté par un graphe connexe $G = (V, E)$ orienté symétrique. Un tel graphe admet un circuit eulérien, c'est-à-dire, un circuit passant une et une seule fois par tous les arcs de G . Soit \mathcal{C} un tel circuit. Chaque paquet émis arrivant à un nœud donné a priorité sur l'arc de sortie correspondant à l'arc suivant dans le circuit. Ainsi, un message émis par le nœud u et à destination du nœud v arrivera nécessairement à destination. Le but est de quantifier le temps maximal pour effectuer cette communication.

Pour un nœud v et un arc $\alpha = (u, x)$ de G , on définit la distance $d_{\mathcal{C}}(\alpha, v)$ comme la longueur du chemin commençant par α et se terminant par la première occurrence de v rencontrée sur le circuit \mathcal{C} .

Définition 3.5 Soit G un graphe orienté symétrique et \mathcal{C} un circuit eulérien de G . Soit u et v deux sommets de G . On appelle **distance eulérienne** entre u et v :

$$d_{\mathcal{C}}(u, v) = \max_{(u, x) \in E} d_{\mathcal{C}}((u, x), v).$$

Par extension, on définit le **diamètre eulérien** par :

$$\text{diam}W_{\mathcal{C}} = \max_{(u, v) \in V \times V} d_{\mathcal{C}}(u, v).$$

Si $Eul(G)$ représente l'ensemble des circuits eulériens de G , on définit le **diamètre eulérien** de G par :

$$\mathcal{E}(G) = \min_{\mathcal{C} \in Eul(G)} diamW_{\mathcal{C}}.$$

Par ailleurs, on définit le **stretch** de \mathcal{C} par

$$S_{\mathcal{C}} = \max_{u \in V} \max_{(u,x) \in E} d_{\mathcal{C}}((u,x), u).$$

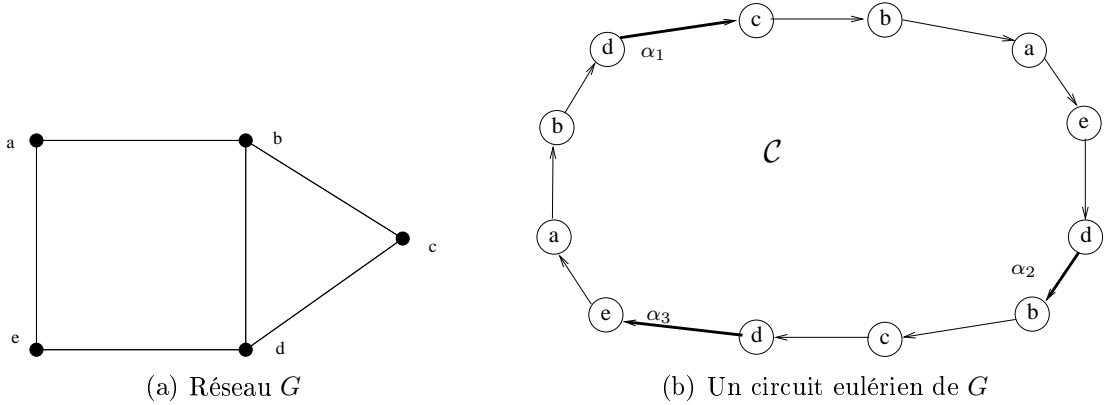


FIG. 3.7 – Un réseau et un de ses circuits eulériens. On a $d_{\mathcal{C}}(\alpha_1, a) = 3$, $d_{\mathcal{C}}(d, a) = d_{\mathcal{C}}(\alpha_2, a) = 5$ et $diamW_{\mathcal{C}} = 6$.

Cette dernière définition représente l'écart maximal entre deux occurrences du même sommet sur l'eulérien. Les deux paramètres relatifs à \mathcal{C} sont simplement reliés par l'équation [9]

$$diamW_{\mathcal{C}} = S_{\mathcal{C}} - 1. \quad (3.1)$$

Cette relation permet de calculer très simplement le diamètre eulérien relatif à circuit donné. Le diamètre eulérien d'un graphe correspond naturellement au minimum sur tous les circuits eulériens possibles du diamètre eulérien. Le nombre de circuits étant très grand (résultats de Fleischner [51, Chapter IX]), il est impensable de les calculer tous dès que le graphe considéré est de « taille raisonnable ».

Étant donné un circuit eulérien, le diamètre eulérien correspond à la garantie de terminaison cherchée. En effet, c'est le chemin le plus long que devra parcourir un message pour atteindre sa destination. Cependant, comme nous le verrons plus loin, ce diamètre est grand ce qui rend en pratique ce mode de routage inutilisable [11]. Plusieurs techniques améliorent les résultats pratiques. De manière paradoxale, elles dégradent la garantie de terminaison. On peut citer :

- l'utilisation du routage eulérien couplé avec le routage à déflexion classique. Les messages partent en mode déflexion, lorsqu'ils ont atteint un seuil du nombre de déflexions, ils passent en mode eulérien [11].
- l'utilisation du routage eulérien avec raccourcis. Le but est de minimiser pour chaque message la distance restant à parcourir entre le routeur où on prend une décision et la destination. Pour un routeur donné u et un message à destination de v , le meilleur choix est de prendre le lien (u, x) qui minimise $d_{\mathcal{C}}((u, x), v)$. Cependant,

cela ne résout pas entièrement le problème de congestion. On transforme le problème initial en la recherche d'un couplage parfait (entre les entrées et les sorties) dans un graphe particulier, prenant en compte les différentes possibilités de sortie entre le meilleur choix défini plus haut et le « pire » qui consiste à rester sur le circuit eulérien. Les sorties qui n'améliorent pas la solution du circuit eulérien ne sont pas à considérer. Conserver le choix de la sortie eulérienne pour chaque message garantit qu'une solution est possible. Dans le cas général, cette solution permet d'améliorer les performances réelles du réseau. Cependant, dans ce dernier contexte, la garantie de terminaison correspond encore au diamètre eulérien tel que nous venons de le définir plus haut. Dans ce cas, on peut définir de la même manière les notions de distance eulérienne avec raccourcis. Il s'avère que cette notion est différente du diamètre au sens classique du graphe. Cependant, dans la plupart des cas, elle s'en approche [75].

3.3.2 Résultats

D'un point de vue général, il existe peu de résultats relatifs au diamètre eulérien de graphes. On peut citer l'encadrement suivant.

Théorème 3.4 ([9]) *Soit G un graphe orienté symétrique connexe de degré minimum δ ayant n sommets et m arêtes, on a alors :*

$$\frac{m}{\delta} - 1 \leq \mathcal{E}(G) \leq m - 2\delta - 3. \quad (3.2)$$

De plus, si G est régulier et $\delta > 2$ alors :

$$\mathcal{E}(G) \geq n + 1.$$

Du point de vue de la complexité, nous ne savons pas si le problème de décision associé à la recherche du diamètre eulérien d'un graphe est NP-complet. Cependant, ceci est fort probable au vu du résultat voisin suivant.

Théorème 3.5 ([9]) *Le problème suivant est NP-complet : étant donné un graphe, un sommet u de G et un entier k , existe-t-il un circuit eulérien \mathcal{C} de G tel que*

$$\max_{(u,x) \in E} d_{\mathcal{C}}((u,x), u) \leq k?$$

La réduction se fait à partir du problème 3-Partition [55].

Étant donné ces résultats, nous avons effectué nos recherches dans plusieurs directions toujours dans le cadre du projet ROM.

1. Chercher des réseaux pour lesquels le diamètre eulérien est de bonne qualité, c'est-à-dire proche de la borne donnée par le théorème 3.4. C'est ce point que nous développons ensuite dans le document.
2. Chercher des *bons* eulériens pour des réseaux connus simples tels que la grille. Il est intéressant de constater qu'un circuit eulérien dont le diamètre eulérien simple est petit possède souvent un diamètre eulérien avec raccourcis grand. D'autres propriétés de circuits eulériens ont aussi été regardées [76].

En ce qui concerne le premier point, nous avons étudié les propriétés du graphe complet. Nous avons mis en évidence une famille de graphes orientés symétriques pour laquelle le diamètre eulérien de cette famille peut être borné par une valeur proche (à une constante près) des bornes obtenues au théorème 3.4. Tous les résultats obtenus sont résumés dans la table 3.1. Le principe général des méthodes utilisées consiste à donner une matrice particulière telle que chaque ligne représente une permutation des entiers compris entre 0 et n . Par ailleurs, si $I(k, l)$ représente l'indice de l'occurrence d'un entier k dans la ligne l , alors $I(k, l+1 \bmod n) \leq I(k, l) + 2$, comme dans l'exemple de la figure 3.8. Une fois cette matrice donnée, on construit le circuit pour le graphe complet, en différenciant les cas pairs et impairs. Afin de donner des valeurs pour les autres degrés, on élimine des parties du circuit ainsi construit.

$$\begin{bmatrix} 0 & 5 & 1 & 4 & 2 & 3 \\ 1 & 0 & 2 & 5 & 3 & 4 \\ 2 & 1 & 3 & 0 & 4 & 5 \\ 3 & 2 & 4 & 1 & 5 & 0 \\ 4 & 3 & 5 & 2 & 0 & 1 \\ 5 & 4 & 0 & 3 & 1 & 2 \end{bmatrix}$$

FIG. 3.8 – Matrice I_6

Famille	borne de $\mathcal{E}(G)$	conjecture pour $\mathcal{E}(G)$
K_n^*	$\begin{cases} = n + 1 & \text{si } n \text{ est impair} \\ \leq n + 4 & \text{sinon} \end{cases}$	$n + 1$
Symétrique, $\lfloor \frac{n-1}{2} \rfloor = p$, $\delta = d$, $p \leq d$	$n + 2 \lfloor \frac{p}{d} \rfloor - 1$?
δ -régulier, non symétrique $q = n \operatorname{div} \delta$, $r = n \bmod \delta$	$\begin{cases} = n + 1 & \text{si } r \leq q \\ \leq n + 3 & \text{sinon} \end{cases}$	$n + 1$

TAB. 3.1 – Diamètre eulériens et graphes réguliers

Si on considère des graphes particuliers, il est très difficile de déterminer la valeur du diamètre eulérien. Ainsi, nous avons orienté nos recherches vers la grille qui possède notamment un diamètre eulérien de l'ordre de m/δ . L'évolution des stratégies de routage montre que ce paramètre n'est pas le plus pertinent. Nous avons montré, au travers d'études expérimentales, que le diamètre eulérien avec raccourcis est une mesure de qualité mieux adaptée pour ce type de routage [53, 76].

3.3.3 Conclusion et perspectives sur le routage eulérien

Dans ce travail, et par extension dans tout le projet ROM, nous avons montré que le routage eulérien est une perspective viable comme routage avec garanties de performance,

en particulier si celle-ci est couplée avec d'autres méthodes de routage. Il faut noter que la propriété de garantie de terminaison dépend directement de ce mode, les autres apportent des moyens efficaces de router les messages.

Bien que nous l'ayons pas développé dans ce document, nous pouvons signaler les résultats de simulations qui renforcent les conclusions précédentes [11, 53]. Lors de ces simulations, nous nous sommes penchés sur le couplage du routage eulérien avec le routage par déflexion : les messages entrent dans le réseau en mode déflexion, puis après un certain nombre de déflexions, ceux-ci terminent leur route suivant le mode eulérien. Elles montrent en particulier des phénomènes de « gel » du réseau quand celui-ci est saturé : une grande majorité des messages suivent le circuit eulérien, ne permettant pas d'écouler les messages assez rapidement. Tous les messages émis sont confinés dans la partie de réseau qui n'utilise pas le routage eulérien et atteignent rapidement le seuil de déflexions, alimentant ainsi la situation de blocage. Cette situation ne se produit pas quand le nombre de déflexions autorisé est très grand, ni si on utilise le routage à déflexion avec raccourcis. Dans ces cas, le nombre de messages en mode eulérien est très réduit et ces messages sont routés de manière prioritaire et efficace vers leur destination finale.

Dans les perspectives de ce travail, il faut maintenant s'intéresser à l'aspect distribué de la construction d'un eulérien. On peut noter [17] où les auteurs définissent un circuit eulérien par un processus auto-stabilisant. Cette technique permet de réparer un circuit en cas de panne du réseau. Cependant, plusieurs questions se posent. En particulier, est-il possible de réparer un circuit eulérien de manière distribuée sans altérer les qualités intrinsèques de celui-ci ? Par ailleurs, comment peut-on signifier à tous les routeurs le nouveau circuit ? Différentes stratégies de réparation des circuits eulériens peuvent être choisies en fonction de la taille du réseau considéré.

3.4 Conclusions et perspectives de ce chapitre

Dans ce chapitre, nous avons rassemblé divers résultats sur les réseaux optiques sous le dénominateur commun de la recherche structurelle des propriétés de ces réseaux. Nous avons soulevé quelques questions ouvertes au cours de notre discussion. Cependant, l'évolution de nos centres d'intérêt font que ces aspects de la recherche sur les réseaux ne font pas partie de nos priorités pour les années à venir.

En effet, bien que ces aspects soient fondamentaux dans la recherche, il me paraît nécessaire d'étudier les contreparties *optimisation* des problèmes que nous avons soulevés. C'est la voie que nous présentons au chapitre suivant. Nous avons choisi d'y présenter des problèmes un peu différents toujours sous le même dénominateur commun : les réseaux d'interconnexion.

Chapitre 4

Optimisation de ressources dans les réseaux

Au chapitre précédent, nous avons exposé plusieurs problèmes structurels reliés au monde des communications, et en particulier, celui des télécommunications. Un deuxième aspect est primordial dans le monde des communications lié à des critères économiques, mais aussi d'efficacité. Il s'agit de diminuer le coût global de création ou d'utilisation d'un réseau.

Dans la suite, nous présentons deux problèmes d'optimisation de ressources. Le premier s'intéresse à la modélisation de l'allocation de groupes multipoints dans un réseau bien dimensionné. La ressource critique est alors la bande passante dont l'utilisation globale doit être réduite. Ce travail, présenté au paragraphe 4.1, a été réalisé en collaboration avec D. Barth et P. Fragopoulou du PRiSM de l'université de Versailles pour la partie complexité du problème ainsi que sur une première heuristique [16]. D'autres heuristiques ont été testées sur ce problème lors du travail de thèse de M. Ben Dhaou [41], thèse que j'ai co-encadrée avec D. Fayard. Le deuxième problème correspond au travail de thèse de L. Gastal, thèse que je co-encadre avec A. Lisser. On se place alors dans le cadre de réseaux évolutifs. Il s'agit d'anticiper les évolutions futures d'un réseau lors de la planification de l'allocation afin que celle-ci coûte le moins cher possible en prenant en compte l'évolution. Nous avons alors étudié diverses implications de ce modèle sur des problèmes classiques. Ce travail est présenté au paragraphe 4.2.

4.1 Communications multipoint

Avec l'évolution des capacités des réseaux de communications, de nouveaux types d'applications sont devenues réalisables. En particulier, les applications mettant en jeu plusieurs partenaires bénéficient de cette évolution. Celles-ci, que l'on appelle **multi-points**, recouvrent des domaines divers. On peut citer par exemple le travail coopératif au travers d'un tableau blanc distribué sur plusieurs sites, les visio-conférences où l'on rassemble sur une page réservée par avance différents interlocuteurs ou, de manière plus ludique, des jeux distribués.

Chacune de ces applications nécessite des garanties propres en terme de performances de communications. Ces garanties peuvent être décrites par exemple en termes de bande passante requise et/ou de délai maximum.

Un certain nombre de protocoles multipoints ont été proposés dans l'Internet, tels que Core-Based-Tree ou Sparse PIM. Une revue des différents problèmes et protocoles décrits pour les applications multipoints est donnée dans [39, 44]. Plusieurs protocoles de gestion de groupes multipoint incluent la possibilité d'évolution du groupe au cours du temps, avec des options différentes si le groupe est constitué de nœuds situés dans le même domaine (protocole intra-domain) ou non (protocoles inter-domaines).

Dans ce paragraphe, nous nous restreignons au cas de la réservation de requêtes multipoints dont les garanties sont simplement décrites en termes de capacité et de latence (ou, en terme de graphe, de distance maximale entre deux points du groupe). Chaque requête est définie comme un ensemble statique de nœuds voulant s'interconnecter. Une application cible de ce mode est la réservation de séances de visio-conférences sur un réseau. Le problème consiste alors à maximiser le nombre de requêtes acceptées étant donné un ensemble de requêtes initiales.

De manière plus formelle, le réseau est considéré comme un graphe $G = (V, E)$ non orienté possédant une fonction de capacité c sur les arêtes, ce qui permet de modéliser le débit offert par le réseau sur chacun de ses liens¹. Un groupe multipoint est décrit par la donnée de trois éléments :

- un sous-ensemble U de V représentant les nœuds à interconnecter ;
- une garantie de débit d pour le groupe ;
- et une latence l maximale autorisée.

Une requête est **satisfaite** s'il existe un sous-graphe connexe $G' = (V', E')$ de G tel que :

- tous les points du groupe sont couverts ($U \subseteq V' \subseteq V$) ;
- toutes les communications intra-groupe sont possibles dans G' ($\forall e \in E' c(e) \geq d$) ;
- toutes les communications ont une latence acceptable ($diam_{G'} \leq l$).

De plus, à l'instar de la majorité des protocoles multipoints, on requiert que ce graphe soit un arbre. Dans ce contexte, vérifier qu'une requête est satisfiable, c'est-à-dire, qu'il existe un sous-arbre du réseau avec les propriétés décrites plus haut, est un problème polynomial [74]. Il diffère du problème de Steiner (Problème ND12 de [55]) car on ne cherche pas dans ce cas à minimiser les ressources utilisées, mais simplement à trouver une solution valide.

De manière naturelle, on dit qu'un ensemble de requêtes est **simultanément satisfiable** dans un graphe donné s'il existe une affectation de chaque requête pour laquelle la capacité de chaque arête du réseau est suffisante pour supporter la totalité des requêtes voulant utiliser cette arête.

4.1.1 Complexité du problème MRS

Nous nous intéressons alors au problème suivant :

Problème : Maximal Request Satisfaction (MRS)

Données : un graphe G possédant une fonction de capacité sur les liens, une collection de requêtes $\mathcal{R} = (R_i)_{0 \leq i < n}$, un entier k ;

¹On peut aussi ajouter une fonction de distance, cela ne change pas la discussion que l'on pourra conduire par la suite.

Question : existe-t-il un sous-ensemble de k requêtes pouvant être satisfaites simultanément dans G ?

Afin de diminuer les contraintes du problème, nous supposons que chaque requête est satisfiable individuellement et nous assimilons la requête à un arbre qui la réalise. Cette relaxation élimine le paramètre de latence dans la requête. On appelle MRS-opt le problème d'optimisation naturellement associé à MRS.

On définit alors les paramètres.

Congestion d'une arête : nombre de requêtes utilisant une arête donnée ($cg(\mathcal{R}, e)$);

Congestion de l'ensemble des requêtes : maximum des congestions sur toutes les arêtes du graphe ($cg(\mathcal{R})$);

Taille d'une requête : nombre d'arêtes du sous-arbre choisi satisfaisant la requête ($ec(R)$);

Taille maximum de l'ensemble des requêtes : maximum de la taille des requêtes ($ec(\mathcal{R})$).

Le théorème suivant présente la difficulté générale du problème MRS.

Théorème 4.1 ([16]) *Le problème MRS est NP-complet, même si la congestion de l'ensemble des requêtes est bornée par une constante.*

De plus, le problème MRS-opt ne peut pas être approximé à plus de $n^{1/2-\epsilon}$ pour $\epsilon > 0$.

La première partie vient d'une réduction à partir du problème du Maximum Set Packing [55]. La seconde partie s'obtient en regardant les relations de notre problème avec le Maximum Independent Set.

On peut compléter ce théorème lorsque la congestion est bornée par une constante. Nous montrons alors une équivalence entre notre problème et le problème du stable de poids maximum dans les hypergraphes (MWIS). À partir d'une instance de notre problème, nous construisons un hypergraphe représentant toutes les solutions admissibles pour chaque arête du graphe. Une solution du problème MRS est alors un sous-ensemble stable de sommets de cet hypergraphe. La taille de l'hypergraphe est bornée par $O(m2^{cg(\mathcal{R})}(cg(\mathcal{R}) + ec(\mathcal{R})))$, où m est le nombre d'arêtes du graphe. Si la congestion est bornée, la taille de l'hypergraphe est proportionnelle à m . On obtient alors le résultat suivant.

Théorème 4.2 ([16]) *Le problème MRS-opt peut être approximé en $O(n/\log n)$ si n est le nombre de requêtes et la congestion est bornée par une constante.*

4.1.2 Cas polynomiaux

Au vu des résultats précédents, nous avons recherché les cas où on pouvait trouver des solutions exactes à notre problème en temps polynomial. La preuve de la NP-complétude au théorème 4.1 utilisant des arbres de degré borné, les cas polynomiaux sont assez peu nombreux.

Dans une chaîne où les capacités sont quelconques et les demandes sont unitaires, le problème MRS se réduit à un problème simple d'affectation sans la contrainte multipoint. En effet, il suffit de ne considérer que les deux éléments du groupe les plus éloignés. Dans ce cas, le problème peut se résoudre par un algorithme glouton, en triant au préalable les requêtes [41]. Cet algorithme peut être étendu aux anneaux.

Si les demandes sont quelconques, même dans une chaîne, on s'aperçoit que le problème devient NP-complet, car il contient les instances du problème de sac à dos. Si les capacités sont bornées, on peut utiliser un algorithme dynamique pour résoudre le problème en $O(n^{q+1})$ où q est le minimum entre la capacité maximale des liens et le nombre maximal de requêtes acceptées passant par un lien donné [41].

Nous avons par ailleurs montré que le problème MRS-opt peut être résolu de manière exacte dans le cas des étoiles de degré borné à capacité unitaire. Ce dernier cas est le seul cas réellement multi-point que l'on sait résoudre [41]. La technique principale consiste à rechercher un problème central qui peut être résolu de manière exacte. D'autres cas simples peuvent sans doute être mis au jour, comme certaines compositions d'étoiles de degré borné.

4.1.3 Résolution exacte et heuristiques

De manière plus prometteuse, nous avons défini diverses heuristiques permettant de résoudre le problème MRS-opt. Premièrement, la preuve du théorème 4.2 a mis en évidence une heuristique donnant avec un rapport d'approximation $O(n/\log n)$ quand la congestion est bornée. L'algorithme est basé sur la recherche d'un ensemble indépendant maximum dans un graphe particulier. Les sommets de ce graphe correspondent aux différentes affectations possibles des requêtes qui ne saturent pas une arête du graphe donné.

On peut alors effectuer une résolution exacte par une évaluation de type Branch & Bound.

Une autre approche, prise dans [41] consiste à reprendre la problème original, où les arbres ne sont pas donnés et de rechercher à maximiser le nombre de requêtes ainsi satisfaites. Les algorithmes génétiques ont été testés en comparaison d'algorithmes gloutons simples. Les arbres correspondant aux différentes requêtes ont été calculés à partir de l'heuristique DNH pour les arbres de Steiner [72]. Cette heuristique a une garantie de performance de 2. Elle a l'avantage d'être facile à mettre en œuvre et d'être de très bonne qualité dans les cas concrets.

Le problème dual de construction d'un réseau de coût minimal à partir de requêtes a aussi été regardé en utilisant cette approche. Ce dernier problème a été étudié par Bienstock et Bley [32] par des méthodes d'optimisation globale.

4.1.4 Conclusion sur les groupes multipoints

Le problème MRS a été une première approche dans la résolution de problèmes d'optimisation. Il nous a servi d'illustration pour aborder ce pan de la recherche que nous ne connaissions que très peu.

Diverses questions restent encore ouvertes sur le sujet. Il manque en effet des comparaisons entre les différents algorithmes et des tests de performance pertinents. Les cas simples s'avèrent très peu nombreux. Il faudrait en particulier pouvoir définir plus finement la frontière avec les problèmes NP-complets.

Étant donné le caractère prospectif de cette étude, nous nous sommes intéressés par la suite à un problème de réseau évolutif, comme nous allons le montrer au paragraphe suivant.

4.2 Réseaux évolutifs

Le cadre de ce dernier travail présenté dans cette partie reste dans le domaine des réseaux. Cependant, nous nous plaçons dans le cadre où la structure même des réseaux peut évoluer au cours du temps. Nous regardons maintenant un problème de réservation de ressources dans la globalité de l'évolution. Nous avons pris comme point de départ un problème simple : les plus courts chemins. Celui-ci est bien connu pour les réseaux « stables ». Cependant, dès que l'on ajoute quelques contraintes complémentaires, ce problème devient rapidement plus complexe. Ce paragraphe résume les travaux effectués dans le travail de thèse de L. Gastal [27, 56] encore en cours. Nous verrons dans les perspectives de ce chapitre des pistes pour généraliser ce modèle à d'autres problèmes.

Le contexte précis de notre étude est le suivant. Considérons une suite de graphes $\mathcal{G} = (G_1, \dots, G_n)$ croissants ($V(G_i) \subseteq V(G_{i+1})$ et $E(G_i) \subset E(G_{i+1})$, la deuxième inclusion étant stricte), possédant une fonction de poids sur les arêtes w . Chaque graphe est appelé **scénario**. Dans le cas du plus court chemin, on se donne deux sommets s et t dans G_1 . Le problème consiste à trouver une suite de chemins $S = (S_1, \dots, S_n)$ telle que chaque S_i relie s à t dans G_i et que le coût global soit minimum. Posé ainsi, ce problème est simple : il suffit en effet de choisir pour chaque graphe le plus court chemin reliant s à t . À l'instar des différents systèmes de location, nous avons ajouté deux coûts supplémentaires qui permettent de coupler les différents chemins. Le premier, ι , correspond au coût d'installation d'une arête : il est ajouté au coût global la première fois qu'une arête est choisie dans un scénario. Le second, δ , correspond au coût de désinstallation d'une arête : il est ajouté à chaque fois qu'une arête n'est plus prise dans le scénario suivant. Les fonctions w , ι et δ sont définies sur les arêtes de chaque graphe. On peut les supposer identiques d'un scénario à l'autre, mais ce n'est pas nécessaire. Le coût global de la solution peut s'écrire sous la forme :

$$\phi(S) = \sum_{i=1}^n w(S_i) + \iota(S_1) + \sum_{i=2}^n \iota(S_i \setminus S_{i-1}) + \sum_{i=2}^n \delta(S_{i-1} \setminus S_i) \quad (4.1)$$

On peut simplifier cette équation en la mettant sous la forme :

$$\phi(S) = \sum_{i=1}^n w(S_i) + \iota(S_1) + \sum_{i=2}^n \Delta(S_{i-1}, S_i), \quad (4.2)$$

la fonction Δ représente alors la fonction de transition entre deux états successifs.

4.2.1 Autres modèles

Les évolutions des réseaux ont déjà été proposées par le passé. Cependant, peu de ces modèles prennent en compte les coûts de l'évolution des réservations.

On peut citer les *graphes évolutifs* introduits par A. Ferreira [50] afin de modéliser les réseaux sans fil et mobiles et leurs évolutions dans le temps. L'idée principale est d'introduire une notion de disponibilité des liens ou des sommets. Chaque instant est alors modélisé par un sous-graphe particulier du graphe global. Les concepts de plus court chemin, de cycle évoluent pour prendre en compte cette temporalité. Des problèmes

simples dans le cas statique deviennent NP-complets dans le contexte dynamique, comme le calcul des composantes connexes de ces graphes [90]. Dans ce modèle, il n’y a pas de pénalité pour l’installation ou la désinstallation des ressources mises en place, il s’agit globalement de problèmes liés au routage dynamique. Par ailleurs, il n’y pas de croissance du modèle comme dans notre cas.

Un autre pan de la recherche, plus axé dans le domaine de la recherche opérationnelle, s’intéresse aussi à des réseaux contenant une certaine incertitude. La thématique générale est la *robustesse*. Il s’agit de trouver une solution qui soit valide pour tous les scénarios possibles. Si on se replace dans notre problème de plus courts chemins, cela revient à travailler sur l’intersection de tous les graphes, c’est-à-dire, on ne considère alors que le graphe le plus petit. Le travail de Bertsimas [30] présente en particulier les divers modèles de robustesse qui ont été choisis ces dernières années. Une caractéristique commune à tous ces modèles est que l’incertitude est uniquement relative aux poids des arêtes, et non à l’existence des arêtes, comme dans notre modèle. Dans ce domaine, on peut remarquer l’introduction des modèles stochastiques pour prendre en compte l’incertitude naturelle dans les réseaux que l’on étudie.

Notre modèle ne peut pas à proprement parler être qualifié de *robuste*². En effet, les évolutions du réseau sont données à l’avance et on doit chercher le meilleur compromis dans cette suite de réseaux. Cependant, nous avons considéré cette étape intermédiaire avant l’introduction de modèles stochastiques, pour lesquels le terme *robuste* s’applique pleinement.

4.2.2 Complexité et approximation du problème

Revenons à la formulation initiale du problème telle que présentée par l’équation 4.1. Le problème de décision lié est alors le suivant :

Problème : Robust Routing in Changing Topologies (RRCT-d)

Données : Une famille $\mathcal{G} = (G_i)_{0 \leq i < n}$ croissante de graphe, une fonction w de poids sur les arêtes des graphes, une fonction d’installation ι et une fonction de désinstallation δ sur les arêtes des graphes, deux sommets s et t de G_1 et k un entier.

Question : Existe-t-il une suite de chemins $S = (S_1, \dots, S_n)$ telle que $\phi(S) \leq k$?

On appellera simplement RRCT le problème d’optimisation associé. En utilisant le problème du circuit hamiltonien, on montre que le problème RRCT-d est NP-complet, même avec des contraintes fortes.

Théorème 4.3 ([27]) *Le problème RRCT-d est NP-complet, même si la famille ne contient que deux graphes et le premier est réduit à un chemin simple.*

²Robuste est défini par Bernard Roy comme « une aptitude à résister à des « à-peu-près » ou à des « zones d’ignorance » afin de se protéger d’impacts jugés regrettables (notamment dégradation de propriétés à préserver) » [83].

L'idée de la preuve consiste à partir d'une instance du problème Circuit Hamiltonien, c'est-à-dire, un graphe. Le premier graphe de la famille modélise la suite des sommets. Pour cela, on associe à chaque sommet du graphe une arête dans le premier scénario de la famille. On relie ensuite tous les sommets pour former un chemin. Le deuxième graphe va refléter la topologie exacte du graphe. L'attribution des poids va favoriser les chemins qui conservent le plus possible les arêtes représentant les sommets du graphe initial.

Afin d'obtenir ce résultat, nous avons créé une instance pour laquelle la fonction de désinstallation n'est pas uniforme et peut être plus grande que le poids initial. Plusieurs remarques peuvent alors être faites.

- Si tous les coûts de désinstallation sont supérieurs aux poids, alors le problème est simple : il suffit de garder un plus court chemin C_1^* dans le premier scénario et considérer la solution $S = (C_1^*, C_1^*)$.
- De même, si tous les coûts de désinstallation sont inférieurs aux poids, le problème avec 2 scénarios et le premier graphe est une chaîne devient polynomial. Le problème consiste alors à trouver un plus court chemin dans le second graphe dont les poids ont été modifiés. La NP-complétude du problème général vient alors du fait que ce dernier graphe possède des cycles absorbants. La question encore ouverte à ce jour est de savoir, si dans ce contexte, le problème général reste NP-complet avec deux scénarios.
- Si ce dernier problème est polynomial, le reste-t-il quel que soit le nombre de scénarios ?

De manière simple, il existe un algorithme d'approximation du problème RRCT.

Lemme 4.1 ([27]) *Il existe une approximation avec un facteur n du problème RRCT.*

Pour cela, il suffit de remarquer que la famille de graphes est croissante et qu'un chemin dans G_1 reste un chemin dans tous les autres graphes. On considère la solution $S = (C_1^*, \dots, C_1^*)$. On peut montrer que cette borne est fine dans le cas général.

4.2.3 Modélisation mathématique

Si on regarde la modélisation mathématique de ce problème avec des variables en 0-1, on s'aperçoit que la fonction objectif introduit des termes quadratiques. Pour cela, on note pour chaque arête (i, j) son poids w_{ij} , son coût d'installation ι_{ij} , son coût de désinstallation δ_{ij} . On appelle x_{ij}^k la variable 0-1 indiquant que l'arête (i, j) est choisie dans le scénario k . Avec ces notations, la fonction objectif devient

$$\begin{aligned} \phi(S) = & \sum_{k=1}^n \sum_{(i,j) \in E_k} w_{ij} x_{ij}^k + \sum_{(i,j) \in E_1} \iota_{ij} + \\ & \sum_{k=2}^n \left(\sum_{(i,j) \in E_k} \delta_{ij} x_{ij}^{k-1} (1 - x_{ij}^k) + \sum_{(i,j) \in E_k} \iota_{ij} x_{ij}^k (1 - x_{ij}^{k-1}) \right) \end{aligned} \quad (4.3)$$

Afin de décrire totalement le problème mathématique, il faut y ajouter les contraintes. Une première série de contraintes décrit que sur chaque scénario, la projection de la solution est un chemin reliant s et t . Pour cela, on écrit les lois classiques concernant

les flots : conservation du flot entre la source et le puits en chaque nœud. Cependant, ces contraintes ne suffisent pas à elles seules. En effet, des circuits peuvent apparaître dans la solution finale, même si on se restreint au cas de la preuve de la NP-complétude (2 scénarios et le premier graphe est un chemin) où la fonction objectif se réduit à une fonction linéaire. Comme dans la résolution du problème du voyageur de commerce, il faut ajouter des contraintes empêchant les cycles dans la solution.

$$\left\{ \begin{array}{l} \min \phi(S) \\ \text{s.c.} \\ \text{Sur chaque scénario, il existe un flot entre } s \text{ et } t \text{ de valeur } 1 \\ \text{Sur chaque scénario, il n'existe pas de sous-cycle induit} \end{array} \right. \quad (4.4)$$

On appelle $RRCT_r$ le problème relâché pour lequel on ne prend pas en compte les contraintes de sous-cycle. Cette formulation est utile car elle est plus simple que la première. Les deux sont en relation directement par le lemme suivant.

Lemme 4.2 *Pour toute instance du problème $RRCT$, si la solution du problème $RRCT_r$ est composée de n chemins entre s et t , alors cette solution est une solution optimale du problème $RRCT$.*

4.2.4 Méthodes de résolution

À partir du lemme 4.2 et de la formulation $RRCT_r$, nous nous sommes intéressés à résoudre cette instance sur différents types de graphes, en particulier sur des instances de différentes tailles simulant des réseaux *réalistes*. Plusieurs techniques ont été mises en œuvre et comparées.

Linéarisation : il s'agit de remplacer les variables quadratiques par de nouvelles variables. La nouvelle fonction objectif ainsi que les diverses contraintes deviennent linéaires. Cela augmente de manière considérable le nombre de variables.

Relaxation continue quadratique : on relâche seulement dans ce cas la contrainte d'intégralité des variables. Les techniques de résolution de programmes avec un objectif quadratique sont implémentés dans les solveurs comme CPLEX.

Relaxation SDP : la relaxation SDP consiste à considérer le problème dans l'espace des matrices semi-définies positives.

Sur les instances testées, il apparaît que la linéarisation permet de résoudre de petites instances. La relaxation quadratique donne de bons résultats, une fois qu'il sont couplés avec des techniques d'arrondi simple. En ce qui concerne la SDP, les sauts de dualité sont plus petits que pour les autres méthodes (la borne inférieure est de meilleure qualité). Cependant, cette méthode ne permet de résoudre que des instances de taille réduite, en comparaison avec les deux autres méthodes.

Dans toutes les instances testées, il apparaît que la relaxation du problème $RRCT$ donne une solution au problème initial.

4.2.5 Conclusion sur ce problème

Ce travail constitue une première approche pour la résolution de notre problème. Nous essayons par ailleurs d'autres techniques de résolution en collaboration avec M. Sebag à l'aide des colonies de fourmis.

Des schémas de résolution doivent être donnés dans le cas où le problème $RRCT_r$ ne donne pas de solution, mais simplement une borne inférieure. La question entière reste de savoir quelle est alors la meilleure formulation du problème. Faut-il faire grossir le nombre de contraintes afin d'éliminer les sous-cycles, comme cela est fait pour la résolution du voyageur de commerce, ou travailler sur une autre forme (quadratique) des contraintes ?

Nous pouvons par ailleurs regarder d'autres problèmes d'allocation dans les réseaux, comme l'arbre couvrant de poids minimum. Nous rediscutons de cette perspective au paragraphe 4.3.

4.3 Conclusions et perspectives : vers une famille de problèmes évolutifs

Dans ce chapitre, nous avons voulu mettre en évidence notre contribution dans le domaine de l'optimisation appliqué principalement à la réservation de ressource dans les réseaux. Si le premier problème peut être vu comme un premier contact avec le monde de l'optimisation en général, le second apporte des ouvertures plus larges que dans le simple cadre des réseaux.

En effet, cette problématique peut être vue dans un contexte plus large que l'on pourrait décrire ainsi. On considère une famille croissante \mathcal{F} de n ensembles, $\mathcal{F} = (F_i)_{0 \leq i < n}$ telle que $F_i \subset F_{i+1}$. On considère alors trois fonctions sur chaque F_i :

fonction de poids : w ;

fonction d'installation : ι ;

fonction de désinstallation : δ .

Pour une propriété \mathcal{P} que l'on peut définir sur chaque ensemble F_i , on note $X_{\mathcal{P}}$ un ensemble de solutions réalisables sur \mathcal{F} relativement à la propriété \mathcal{P} . La fonction objectif globale pour évaluer une solution $X = (X_1, \dots, X_n)$ est donnée par :

$$g(X) = \sum_{i=1}^n w(X_i) + \iota(X_1) + \sum_{i=2}^n (\delta(X_{i-1} \setminus X_i) + \iota(X_i \setminus X_{i-1})) \quad (4.5)$$

Le but de notre problème est de minimiser ou maximiser cette fonction objectif, ainsi que de trouver une solution optimale. Les fonctions d'installation et de désinstallation sont des fonctions pénalisant la qualité de l'objectif. En particulier, si on regarde un problème de maximisation, ces deux fonctions doivent être négatives.

Le problème décrit au paragraphe 4.2 s'inscrit directement dans cette classe de problèmes. Cependant, on peut regarder toute une famille de problèmes de ce type [56].

Élément minimum : c'est l'exemple le plus simple que l'on peut construire dans cette classe. La propriété consiste à choisir un élément dans un ensemble et le but est la

minimisation. Si le nombre de scénarios est fini, le problème global est clairement polynomial en $O(m^n)$ où m est la cardinalité de F_n . Dans ce cas, la solution qui consiste à prendre toujours le minimum du premier ensemble est une n -approximation³.

Élément maximum : c'est aussi un exemple simple. La propriété est la même que dans le cas précédent, mais le but est inverse. Cependant dans ce cas, on ne peut pas trouver de rapport d'approximation simplement.

Arbre de Steiner : la propriété considérée dans ce cas est de trouver un arbre couvrant un ensemble de sommets prédéfinis dans le premier graphe. Le problème est bien sûr NP-complet même dans le cas statique ($n = 1$). Cependant, on peut montrer que l'on obtient une $n\rho$ approximation si ρ est le rapport d'approximation du problème de Steiner.

Arbre couvrant : la propriété considérée est un arbre couvrant dans chaque ensemble F_i . Ce problème a priori plus complexe que le plus court chemin peut être résolu en temps polynomial. En effet, on montre que le problème quadratique sous-jacent est séparable. On obtient ce résultat en constatant qu'une arête qui n'est pas choisie à un instant donné ne sera jamais réutilisée par la suite.

Sac à dos : dans ce cas, le problème consiste à maximiser le coût (gain) global d'une solution : il faut alors prendre les fonctions d'installation et de désinstallation comme des pénalités. Il existe une $n\rho$ approximation de ce problème si ρ est le rapport d'approximation du sac à dos. La solution est un peu plus complexe à construire : on part du dernier ensemble F_n et on considère la *meilleure* solution et ses projections dans les ensembles précédents : il n'y a pas de coût de désinstallation.

Cette classification est une première ébauche. Il faut encore affiner tous les résultats que nous venons de donner. En particulier, toutes les approximations sont données de manière très grossière.

Il existe encore d'autres élargissements à ce travail. En particulier, on peut regarder le cas où la famille de scénarios n'est plus croissante, mais quelconque. Par ailleurs, afin de se rapprocher des modèles de robustesse classique, on peut supposer une incertitude sur l'existence des éléments dans une famille. Ce dernier type d'étude peut être abordé par une approche stochastique.

³On peut améliorer simplement ce rapport en ne considérant qu'une partie des scénarios. Par exemple, en considérant 2 scénarios particuliers, on obtient une $n/2$ approximation.

Deuxième partie

Représentations de graphes

Cette partie regroupe deux aspects de ma recherche à première vue assez éloignés, mais qui ont au moins deux dénominateurs communs. Le premier est non scientifique. Les deux sujets que nous allons aborder font partie des thèmes que j'ai étudiés les plus récemment et pour lesquels mon implication y a plus été motrice. Le deuxième aspect concerne directement l'objet de l'étude. Nous montrons dans ces deux chapitres comment des structures de données représentatives de certains aspects du graphe permettent de résoudre efficacement les questions posées.

De manière plus précise, nous montrons dans un premier temps au chapitre 5 que les arbres de Gomory-Hu, initialement définis pour l'étude des flots multi-terminaux simples, sont précieux pour l'étude d'un problème voisin : les flots multi-terminaux paramétrés. De cette étude résulte une meilleure compréhension de ces arbres.

Le chapitre 6 s'intéresse, au travers du calcul du polynôme chromatique, à une représentation des graphes sous une forme originale : les arbres de cliques augmentés, qui sont une représentation particulière d'une triangulation du graphe considéré. Nous montrons que cette structure s'avère adéquate pour effectuer des opérations d'ajout d'arêtes et fusion de sommets dans des cas qui sont utiles au calcul du polynôme chromatique.

Chapitre 5

Les flots multi-terminaux paramétrés

La thématique présentée dans ce chapitre a été initiée par une collaboration principale lors de la thèse de Madiagne Diallo, suite à un des problèmes que ce dernier avait étudié durant ses années de Master [42]. Ce chapitre résume les travaux suivants [12–14, 23].

5.1 Introduction

La problématique du flot maximal est bien connue et fait partie maintenant de la base des cours de Licence et de Master d’informatique. En particulier, le problème du flot maximum a été résolu par Ford et Fulkerson en montrant en particulier le théorème *du flot maximum et de la coupe minimum*, en extension du théorème de Menger. Même si beaucoup de progrès ont été réalisés dans l’algorithmique du calcul d’un flot maximum, il reste encore des questions ouvertes quant à la complexité réelle de ce problème. Est-il en effet plus facile de calculer une coupe de valeur minimale, et donc la valeur du flot maximal correspondant, que de calculer un flot maximal entier (donner la fonction de flot dans son intégralité)? Ce chapitre ne s’intéresse pas à ce problème, mais suppose que la complexité du calcul du flot maximal est donnée.

En regard, la problématique des *flots maxima multi-terminaux* est moins connue. Elle date du début des années 1960 où elle a été introduite dans des problèmes de recherche de vulnérabilité de réseaux. Contrairement au problème du flot maximal simple, ce dernier ne présuppose pas de source, ni de destination fixe : chaque nœud du réseau pouvant potentiellement jouer l’un de ces deux rôles. Cependant, contrairement au cas des multi-flots¹, on ne considère qu’une seule paire source/destination est active simultanément. De ce problème, il en ressort deux autres reflétant des aspects complémentaires du problème initial :

Faisabilité : étant donnée une matrice carrée $M = (m_{i,j})$ d’ordre n , existe-t-il un réseau R à n sommets tel que la valeur du flot maximum entre les sommets i et j soit égal à $m_{i,j}$? Ce problème est à la base des problèmes de dimensionnement de réseaux. On peut ajouter diverses contraintes sur le réseau cible comme des contraintes de coût global d’installation ou des contraintes de fiabilité et sécurité (au moins deux

¹Le problème des multi-flots possède beaucoup de variantes. Cependant, il s’agit de relier plusieurs couples (source, destination) avec des contraintes complémentaires. Les problèmes de décision associés sont souvent NP-complets. Un certain nombre de résultats sur les multi-flots sont présentés dans [86].

chemins entre toutes les paires de sommet). Ces contraintes sont importantes quand on construit un réseau réel.

Analyse : étant donné un réseau R , déterminer (de manière efficace) l'ensemble des valeurs de flots maxima entre toutes les paires de sommets de ce réseau.

Notre étude s'inscrit dans ce dernier cadre, et de manière plus spécifique dans le cas où le réseau est non orienté. En effet, pour un réseau dont les capacités sont fixes, il existe une manière simple et compacte de représenter toutes les valeurs de flots maximum entre n'importe quelle paire de sommets : l'arbre dit de *Gomory-Hu*. Cette structure possède aussi la propriété de donner une coupe minimum pour chaque paire de sommets. Nous détaillons cette structure dans la suite.

Un certain nombre d'études concerne des réseaux pour lesquels les capacités des arêtes sont fixées à l'avance. Dans ce chapitre, nous nous intéressons particulièrement au cas où les capacités de certaines arêtes sélectionnées peuvent varier au cours du temps. La littérature sur ce sujet est moins abondante que dans le cas fixe. Il est connu sous le terme d'*analyse de sensibilité des flots multi-terminaux*. Le problème a été introduit dans un premier temps par Elmaghraby en 1964 [47]. Il s'agissait de regarder l'effet de la variation d'une seule capacité du réseau sur l'ensemble des flots maxima. Plus tard, en 2000, Diallo [42] a proposé une première généralisation de ce travail en faisant varier plusieurs capacités d'arêtes suivant le même paramètre.

Dans [43], Diallo montre des failles dans la méthode d'Elmaghraby. L'analyse proposée par Elmaghraby suppose le calcul de nombreux arbres de Gomory-Hu. Le point de départ de notre travail a été de simplifier l'analyse d'Elmaghraby en réduisant le nombre de calculs d'arbres de Gomory-Hu à seulement deux. Cette simplification permet de généraliser l'étude de sensibilité au cas de plusieurs arêtes paramétrées de manière indépendante. Dans le cas d'une seule arête paramétrée, les deux arbres que l'on doit considérer concernent deux réseaux qui sont proches : ils ne diffèrent que par une seule arête. Nous nous sommes donc intéressés à étudier la proximité des deux arbres utilisés dans l'étude de sensibilité. Nous en avons déduit une approche algorithmique pour améliorer le calcul de ces deux arbres que l'on peut qualifier de couplés. La suite de l'étude s'intéresse à l'optimalité de l'algorithme précédent en montrant, en particulier, comment choisir les arbres de Gomory-Hu à construire pour que le nombre global d'opérations (en termes de nombre de coupes minimum calculées) soit minimal.

Par ailleurs, d'autres types de paramétrisations de réseaux ont été proposés. Dans [4], Aneja, Chandrasekaran et Nair étudient une autre version dans laquelle les capacités sont toutes paramétrées par une fonction de λ .

Dans la suite de ce chapitre, nous présentons rapidement les diverses définitions complémentaires utiles pour la compréhension de ce chapitre. Nous présentons notre approche de l'analyse de sensibilité au paragraphe 5.3. Nous en déduisons quelques extensions possibles de ces premiers résultats au paragraphe 5.4. De cette première étape, nous présentons des avancées complémentaires dans l'étude des arbres de Gomory-Hu, en particulier comment calculer plus efficacement des arbres de Gomory-Hu pour des réseaux « proches » (paragraphe 5.5), mais aussi comment rechercher des arbres de Gomory-Hu possédant des propriétés particulières (paragraphe 5.6). Dans tout ce chapitre, nous soulevons diverses questions ouvertes directement liées au thème traité. Enfin nous concluons

ce chapitre au paragraphe 5.7 par de nouvelles directions possibles de recherche en lien avec les arbres de Gomory-Hu.

5.2 Définitions

5.2.1 Notion de flots

La notion de flot dans un réseau est définie initialement dans les graphes orientés, plus précisément dans les réseaux orientés [19]. Nous souhaitons préciser ces notions dans le cas où le réseau est non orienté.

Définition 5.1 *On appelle **réseau non orienté** la paire $R = (G, c)$ où $G = (V, E)$ est un graphe non-orienté et c est une fonction de capacité sur les arêtes de G ($c : E \rightarrow \mathbb{R}^+$)².*

Soit G^* le graphe orienté symétrique obtenu à partir de G , en remplaçant chaque arête de G par deux arcs opposés. Le réseau orienté correspondant est alors $R^* = (G^*, c^*)$, où c^* est l'extension naturelle de c sur les arcs de G^* ($c^*(a) = c(e)$, si a et e ont les mêmes extrémités).

Définition 5.2 *Étant donné un réseau non orienté R , et deux sommets s et t de ce réseau, un **flot** dans R entre s et t est défini comme un flot dans R^* entre s et t (ou entre t et s). On notera $f_{s,t} = f_{t,s}$ la valeur du flot.*

Nous précisons ce que nous entendons par la notion de coupe dans un réseau. La définition suivante est compatible avec celle donnée classiquement dans la littérature. Cependant, notre vision permet de tester l'égalité de deux coupes dans deux réseaux différents ayant le même ensemble de sommets. Il suffit alors de faire un test d'égalité d'ensembles.

Définition 5.3 *Un sous-ensemble propre X de V ($\emptyset \subsetneq X \subsetneq V$) est appelé une **coupe** séparant u et v si $u \in X$ et $v \in V \setminus X$. La **capacité de la coupe** est alors définie comme :*

$$c(X) = \sum_{x \in X, y \in V \setminus X, [x,y] \in E} c[x, y];$$

une arête $[x, y]$ telle que $x \in X$ et $y \in V \setminus X$ appartient à la coupe X .

Il faut noter que si on considère un coupe valide sur deux réseaux différents, il n'y a aucune raison pour que les capacités de cette coupe dans chacun de ces deux réseaux soient identiques.

Définition 5.4 *Une **coupe minimum séparant** deux sommets u et v , notée $C_{u,v}$, est une coupe de capacité minimale parmi toutes les coupes séparant u et v .*

²Toute notre étude est applicable quelle que soit la nature des capacités : entières ou réelles.

Le théorème de Ford-Fulkerson [52] relie les deux notions en établissant en particulier l'égalité de la valeur de la coupe minimum et de la valeur du flot maximum. Ce théorème nous a souvent servi lors des diverses preuves de nos résultats.

Il existe un certain nombre d'algorithmes performants pour calculer la valeur du flot maximum entre deux sommets d'un réseau ainsi qu'une coupe minimum. On se référera par exemple à [57] pour une revue des différents algorithmes. On peut noter qu'il existe un algorithme linéaire lorsque la largeur arborescente est bornée par k [64]. Cependant, cet algorithme n'est pas utilisable en pratique étant donnée la constante dépendant de la largeur arborescente de l'ordre de 2^{2^k} . À l'extrême, si le graphe sous-jacent est un arbre, le temps de calcul est linéaire et trivial : il suffit de chercher la capacité minimale entre les deux sommets considérés. Cette dernière remarque sera utile par la suite.

Notation 5.1 *Nous noterons $\mathbf{MF}(n, m)$ la complexité en temps pour calculer un flot maximum dans un réseau ayant n sommets et m arêtes.*

5.2.2 Arbres de flots, arbres de coupes

Nous nous intéressons dans ce chapitre aux flots multi-terminaux, c'est-à-dire, à l'étude d'un réseau pour lequel la source et le puits ne sont pas déterminés à l'avance, mais où chaque sommet peut potentiellement être l'un ou l'autre. Le réseau ne sera utilisé que comme un réseau de flot simple.

Définition 5.5 *Le problème des **flots multi-terminaux** consiste à déterminer pour un réseau non orienté donné R l'ensemble des valeurs de flot maximum entre toutes les paires de sommets.*

Le problème consiste donc à déterminer l'ensemble des $n(n-1)/2$ valeurs de flots possibles. En 1961, Gomory et Hu ont proposé une représentation compacte de l'ensemble de ces flots sous forme d'un arbre de coupes, réduisant à $(n-1)$ le nombre de calculs de flots maximum [59]. La définition ci-dessous est dérivée de celle de Gomory-Hu [59] et de notre définition des coupes.

Définition 5.6 *Étant donné un réseau $R = (V, E, c)$, un **arbre de coupes** ou **arbre de Gomory-Hu** de R est un réseau $T = (V, E', c')$ ayant le même ensemble de sommets que R vérifiant les trois propriétés suivantes :*

Arbre : *le graphe (V, E') est un arbre ;*

Flots : *étant donnés deux sommets s et t les flots $f_{s,t}$ dans R et $f_{s,t}$ dans T sont identiques ;*

Coupes : *une coupe minimum $C_{s,t}$ dans T est aussi minimum entre s et t dans R , le réseau cible³.*

³Bien que l'on considère deux réseaux différents (T et R), on peut définir la même coupe (X, Y) dans ces deux réseaux car l'ensemble des sommets des deux réseaux sont identiques. Dans le cas général, la valeur de cette coupe est différente dans les deux réseaux. L'égalité dans le cas des coupes minimum de T est une propriété des arbres de Gomory-Hu.

Plusieurs points sont à remarquer. Premièrement, pour un réseau donné, il n'y a pas unicité des arbres de Gomory-Hu. Nous utilisons cette propriété plus loin, en particulier au paragraphe 5.6. Par ailleurs, il existe une forme affaiblie, mise en évidence par Gusfield [63], des arbres ne présentant que la propriété de flot. Dans la figure 5.1, nous présentons un réseau simple, un arbre ayant la propriété de flots sans la propriété de coupes et un arbre de Gomory-Hu de ce réseau ; l'arbre de la figure 5.1(b) n'est pas un arbre de Gomory-Hu car la coupe 3/2,1 n'est pas minimale entre 2 et 3.

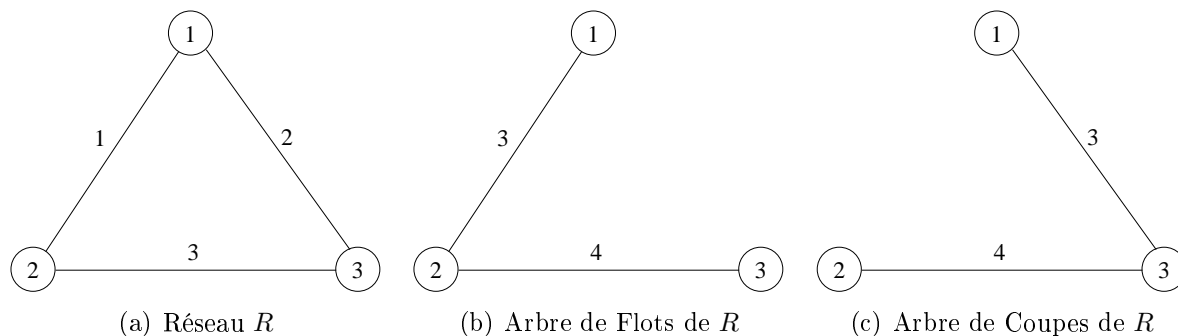


FIG. 5.1 – Réseau, arbre flot équivalent et arbre de Gomory-Hu

Plusieurs algorithmes permettent d'obtenir des arbres de Gomory-Hu. Les deux principaux sont celui de Gomory et Hu [59] et celui de Gusfield [63]. Nous présentons de manière synthétique ces deux algorithmes plus loin dans ce chapitre afin de les utiliser de manière ciblée. Dans [58], Goldberg et Tsioutsoulis présentent une étude expérimentale comparée de ces deux algorithmes.

5.2.3 Paramétrisation

Dans la suite de ce chapitre, nous considérons un certain nombre d'arêtes du réseau pour lesquelles la capacité n'est pas déterminée à l'avance. Étant donné un réseau R , nous sélectionnons k arêtes e_1, e_2, \dots, e_k pour lesquelles la capacité est variable ($c(e_i) = \lambda_i$) (on élimine les indices quand $k = 1$ afin de simplifier les notations).

Pour marquer les différents éléments dépendant des paramètres, nous noterons ceux-ci en exposant. Par exemple, la valeur d'un flot maximum entre s et t sera noté $f_{s,t}^{\lambda_1, \lambda_2, \dots, \lambda_k}$. Cela concernera en particulier le réseau, les arbres de Gomory-Hu, les flots et les coupes.

Pour chaque paramètre, deux cas particuliers sont à noter. Premièrement, $\lambda_i = 0$ signifie simplement que la capacité de l'arête est nulle, en d'autres termes que l'arête peut être éliminée du réseau, ou plus simplement on travaille dans $R \setminus e_i$. Deuxièmement, $\lambda_i = \infty$ est plus difficile à définir de façon pratique, en particulier dans le cadre des calculs de flots. Nous montrons par la suite la validité de cette notation. Afin de faire un parallèle avec ce qui précède, cela peut être vu comme la contraction de l'arête e , c'est-à-dire, travailler dans le réseau R/e .

L'analyse de sensibilité dans le cadre des flots multi-terminaux paramétrés consiste en l'étude du comportement des flots maximaux entre les différents points du réseau en fonction des paramètres. Quand il n'y a qu'un seul paramètre (ou que tous les paramètres sont dépendants d'un seul), elle consiste à chercher les valeurs de ce paramètre pour

lesquelles on constate un changement de comportement du type « une coupe qui était minimum n'est plus minimum ».

Le but de ce chapitre est de montrer comment résoudre ce problème efficacement. Le paragraphe suivant s'intéresse directement à l'analyse de sensibilité, les suivants montrent les conséquences de cette nouvelle analyse.

5.3 Vers une simplification de l'analyse de sensibilité

Dans ce paragraphe, nous présentons le problème de sensibilité tel qu'il a été posé et résolu par Elmaghraby, puis nous en présentons notre vision au paragraphe 5.3.2.

5.3.1 Vision d'Elmaghraby

Dans le problème initialement défini par Elmaghraby [47], l'analyse de sensibilité ne traite que le cas d'une seule arête paramétrée ($k = 1$). On dispose d'un réseau pour lequel une arête e possède des *fuites* : la capacité de cette arête a pour forme $c(e) = c_0 - \epsilon$, $0 \leq \epsilon \leq c_0$. On veut alors savoir à partir de quelle(s) valeur(s) des pertes les différents flots seront affectés.

La technique mise en place par Elmaghraby est basée sur la décroissance de la capacité $c(e)$. Quand cette capacité diminue, certains flots sont affectés, c'est-à-dire, la valeur du flot maximum correspondant diminue. Elmaghraby définit alors la notion de *capacité critique* comme une valeur de la capacité paramétrée pour laquelle un flot commence à être affecté. Afin de déterminer ces capacités critiques, Elmaghraby commence par calculer un arbre de Gomory-Hu pour le paramètre $\epsilon = 0$ (soit T^{c_0}). À partir de la matrice décrivant les coupes minimum dans le réseau original (de taille $m \times (n-1)$) [46], il identifie l'ensemble des coupes minimum qui contiennent l'arête paramétrée, et trouve ainsi la première valeur critique ϵ_0 . Cette technique est répétée à partir de $c(e) = c_1 = c_0 - \epsilon_0$.

Le nombre d'arbres de Gomory-Hu construits par cette méthode correspond au nombre de valeurs critiques distinctes dans le réseau. Dès que le réseau est de taille « raisonnable », cette méthode n'est pas applicable en pratique.

Par ailleurs, une étude fine de cette méthode par Diallo a montré quelques failles dans la caractérisation citée plus haut. Des solutions utilisant une méthodologie de même type y ont été apportées [43], ne faisant qu'augmenter la complexité finale du problème.

5.3.2 Vision positive

Le défaut de la méthode précédente provient essentiellement de la manière de considérer le problème. En effet, le réseau est alors vu au travers de ses modifications élémentaires. Notre contribution principale a été de considérer l'évolution globale d'un flot sur toute la plage des valeurs des capacités de l'arête paramétrée. Le comportement global de la fonction donnant la valeur du flot maximum en fonction de la capacité critique est simple comme le montre la propriété suivante et la figure 5.2. On rappelle que $f_{s,t}^\lambda$ est la valeur du flot maximum entre s et t quand l'arête paramétrée a pour capacité λ .

Propriété 5.1 ([23]) Soit R un réseau, e une arête de capacité paramétrée par λ , s et t deux sommets du réseau. La valeur du flot maximum $f_{s,t}^\lambda$ s'exprime par :

$$f_{s,t}^\lambda = \min(f_{s,t}^0 + \lambda, f_{s,t}^\infty) \quad (5.1)$$

Cette propriété exprime le fait qu'un flot profite linéairement de l'augmentation de capacité de l'arête étudiée (quand celle-ci est dans une coupe minimum), puis, lorsque celle-ci sort de la coupe minimum, le flot n'est plus influencé par les évolutions futures. Le cas particulier où le couple (s, t) correspond aux extrémités de l'arête e conduit simplement à ne considérer que la première partie de l'équation 5.1, la seconde valant par définition $+\infty$. Dans le cas général, il existe un moyen simple de calculer $f_{s,t}^\infty$: il suffit de considérer une capacité suffisamment grande pour s'assurer que l'arête ne fera pas partie d'une coupe minimum comme la somme des capacités des arêtes incidentes à l'une des extrémités de e . Une autre technique consiste à considérer le réseau R/e (ce qui revient à identifier les extrémités de l'arête).

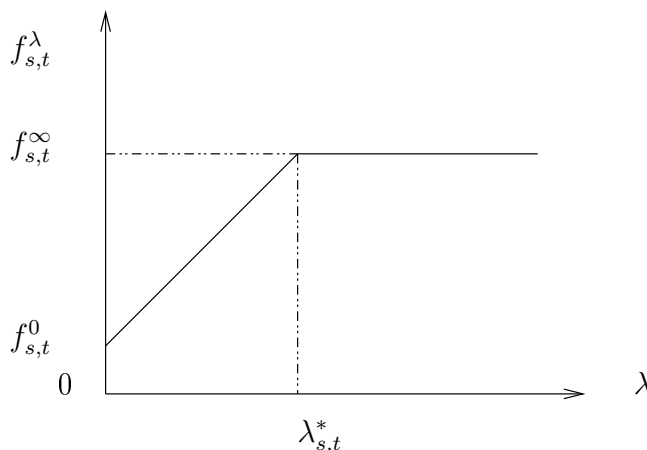


FIG. 5.2 – Modélisation d'un flot paramétré

À partir de cette propriété, nous pouvons simplement retrouver la capacité critique associée à ce flot telle qu'elle est donnée par l'analyse d'Elmaghraby. Celle-ci correspond au changement de comportement de la fonction de flot.

Lemme 5.1 ([23]) Étant donné un réseau R , une arête paramétrée $e = [i, j]$ et deux sommets s et t du réseau, la capacité critique associée au flot $f_{s,t}^\lambda$ est pour $(s, t) \neq (i, j)$:

$$\lambda_{s,t}^* = f_{s,t}^\infty - f_{s,t}^0 \quad (5.2)$$

Nous pouvons faire deux remarques relatives à ce lemme concernant les limites de cette équation. Si les sommets s et t considérés correspondent aux extrémités de l'arête examinée, par extension on peut poser que la capacité critique vaut $+\infty$. Par contre, il faut être plus attentif quand la formule donne $\lambda_{s,t}^* = 0$. En effet, si les deux extrémités de l'arête e se trouvent de part et d'autre d'une coupe minimum, alors, il existe effectivement une capacité critique ; dans l'autre cas, la notion de capacité critique ne peut être définie.

Grâce au théorème de Ford-Fulkerson, la propriété 5.1 peut s'étendre au cas des coupes minimum.

Lemme 5.2 ([12]) *Étant donné un réseau R , une arête paramétrée e et deux sommets du réseau s et t , une coupe minimum entre s et t est donnée par :*

$$C_{s,t}^\lambda = \begin{cases} C_{s,t}^0 & \text{si } \lambda \leq \lambda_{s,t}^* \\ C_{s,t}^\infty & \text{sinon} \end{cases} \quad (5.3)$$

On peut remarquer que pour la capacité critique, les deux coupes sont valides.

Théorème 5.1 ([12, 23]) *Soit R un réseau et e une arête paramétrée de R . Soit T^0 et T^∞ deux arbres de Gomory-Hu pour $\lambda = 0$ et $\lambda = \infty$.*

Pour tout couple de sommets du réseau $[s, t]$, le calcul de la valeur du flot maximum et la recherche d'une coupe minimum $f_{s,t}^\lambda$ peut s'effectuer en un temps linéaire en la taille du réseau ($O(n)$).

De plus, la recherche de toutes les valeurs critiques nécessite $O(n^3)$ étapes.

En comparaison avec la méthode d'Elmaghraby qui nécessite autant de calculs d'arbres de Gomory-Hu que de valeurs critiques dans le réseau, notre méthode ne requiert que le calcul de 2 arbres de Gomory-Hu, chaque arbre nécessitant $(n-1)MF(n, m)$ étapes. Ce théorème constitue donc une avancée importante dans le domaine.

Dans les paragraphes suivants, nous allons présenter les diverses implications et questions que pose ce théorème. En particulier, au paragraphe 5.4, nous présentons deux généralisations de ce théorème. Au paragraphe 5.5, nous considérons le calcul d'arbres de Gomory-Hu couplés dans le but de gagner du temps sur la partie *amont* du théorème 5.1.

5.4 Extensions du théorème 5.1

Dans ce paragraphe, nous proposons deux extensions possibles du théorème 5.1. La première consiste simplement à considérer d'autres bornes pour le théorème. La seconde prend en compte le cas multi-dimensionnel, c'est-à-dire, le cas où plusieurs arêtes peuvent être paramétrées.

5.4.1 Bornes

Afin de rendre plus claire la présentation, nous nous sommes restreints dans ce chapitre au cas où les valeurs extrêmes du théorème 5.1 sont les plus simples : 0 pour l'absence d'arête et $+\infty$ lorsque les deux extrémités de l'arête sont confondues. Le fait de choisir d'autres bornes consiste simplement à limiter la portée de l'équation 5.1 et d'en déduire les autres propriétés.

Supposons donc que la capacité de l'arête paramétrée ne varie qu'entre deux valeurs α et β , $0 \leq \alpha < \beta \leq \infty$. Nous en déduisons la nouvelle formulation de la valeur du flot maximum

$$f_{s,t}^\lambda = \min(f_{s,t}^\alpha + \lambda - \alpha, f_{s,t}^\beta) \quad (5.4)$$

De cette formulation, on remarque qu'il n'existe pas nécessairement de valeur critique dans l'intervalle considéré, en particulier si $f_{s,t}^\beta = f_{s,t}^\alpha + \beta - \alpha$. Par ailleurs, le théorème 5.1 reste vérifié en remplaçant 0 par α et $+\infty$ par β .

Avec cette technique, on peut réécrire tous les théorèmes de ce chapitre. Cependant, cette modification n'apporte pas d'autre intérêt que d'accroître la complexité des formules [12].

5.4.2 Plusieurs arêtes paramétrées

Dans ce paragraphe, nous nous intéressons au cas où les capacités de k arêtes peuvent varier indépendamment ; nous dirons que dans ce cas qu'on travaille sur un *problème de dimension k* . Soit e_1, \dots, e_k les arêtes visées dont la capacité est donnée par $c(e_i) = \lambda_i$. Nos résultats en dimension supérieure proviennent en grande partie de l'application de la propriété suivante.

Propriété 5.2 ([23]) *Étant donné un réseau R possédant k arêtes paramétrées, la fonction partielle $\lambda_i \mapsto f_{s,t}^{\lambda_1^0, \dots, \lambda_i, \dots, \lambda_k^0}$ pour s et t des sommets de R et $\lambda_j^0, j \neq i$ des capacités fixées comprises entre 0 et $+\infty$ est une fonction du même type que l'équation 5.1 :*

$$f_{s,t}^{\lambda_1^0, \dots, \lambda_i, \dots, \lambda_k^0} = \min(f_{s,t}^{\lambda_1^0, \dots, 0, \dots, \lambda_k^0} + \lambda_i, f_{s,t}^{\lambda_1^0, \dots, \infty, \dots, \lambda_k^0})$$

La même proposition peut s'écrire pour les coupes minimum. De cette propriété, on en déduit comment obtenir n'importe quelle valeur de flot maximum et une coupe minimum en fonction des valeurs extrêmes pour chaque capacité. Dans le cas de la dimension 2, cela donne l'expression suivante ; pour les dimensions supérieures, il suffit de développer le calcul (ce qui ne pose pas de problème théorique majeur).

Lemme 5.3 *Étant donné un réseau R , deux sommets s et t du réseau et deux arêtes paramétrées e_1 et e_2 , la valeur du flot maximal entre s et t s'écrit sous la forme :*

$$f_{s,t}^{\lambda_1, \lambda_2} = \min \left(\begin{array}{l} f_{s,t}^{0,0} + \lambda_1 + \lambda_2, \\ f_{s,t}^{0,\infty} + \lambda_1, \\ f_{s,t}^{\infty,0} + \lambda_2, \\ f_{s,t}^{\infty,\infty} \end{array} \right) \quad (5.5)$$

Nous avons le même type d'extension dans le cas des coupes minimum. Celle-ci peut se représenter de manière synthétique par la figure 5.3.

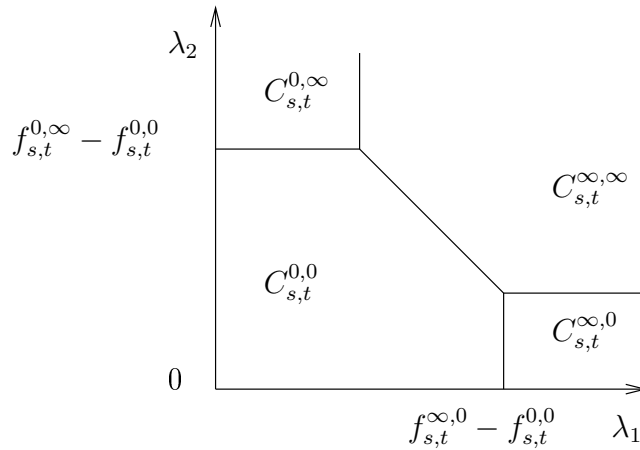


FIG. 5.3 – Zones de validité de chaque coupe minimum

Dans le cas général, le théorème suivant donne la complexité du calcul des valeurs des flots maximum.

Théorème 5.2 ([23]) *Soit R un réseau dont k arêtes sont paramétrées. On peut déterminer à partir de 2^k arbres de Gomory-Hu la valeur d'un flot maximum et d'une coupe minimum entre deux sommets quelconques en temps $O(2^k n)$ où n est le nombre de sommets du réseau.*

Idée de la preuve : On obtient ce résultat par récurrence à partir du cas unidimensionnel. Les arbres de Gomory-Hu considérée sont représentés dans la famille T^{x_1, \dots, x_k} , $x_i \in \{0, \infty\}$, $1 \leq i \leq k$. □

Ce théorème n'améliore la complexité que lorsque le nombre d'arêtes paramétrées est faible. En effet, quand le ce nombre est trop grand, cette recherche requiert plus de calculs que la recherche simple du flot maximum direct dans le réseau.

De manière extrême, si toutes arêtes sont paramétrées, on peut interpréter ce théorème par : la valeur du flot maximum est la somme des capacités des arêtes de la coupe minimum. On retrouve donc la propriété fondamentale des flots.

Cette partie peut apporter de nouvelles directions d'investigation afin de compléter l'analyse dans le cas multidimensionnel. En effet, l'étude précédente se base sur l'indépendance totale de la variation des paramètres. Cependant, les divers paramètres peuvent être liés, comme dans le cas où l'on dispose d'une certaine quantité de capacité à répartir au mieux entre les quelques capacités qui ne sont pas encore fixées. Les questions qui se posent alors sont :

1. Est-il nécessaire de calculer autant d'arbres de Gomory-Hu ? À partir de la figure 5.3, on peut faire la remarque suivante : si la dépendance entre les deux paramètres est du type $\lambda_2 = \alpha \lambda_1 + \beta$, avec $\alpha > 0$, il ne faudra considérer que trois coupes différentes (trois zones d'influence). Dans le cas général, les 4 zones sont nécessaires. Comment ce phénomène se généralise aux dimensions supérieures ?
2. Le domaine de validité du théorème 5.2 en serait-il élargi ?

5.5 Calcul d'arbres de Gomory-Hu couplés

Ce paragraphe s'intéresse en premier lieu à l'étude de la portée des théorèmes 5.1 et 5.2. On remarque en effet que les 2 (2^k) arbres de Gomory-Hu que l'on doit considérer concernent des réseaux qui ne diffèrent que localement : le premier considère le réseau auquel on a supprimé l'arête cible e , le second, le mineur du réseau par contraction de cette arête. On a cherché à améliorer le calcul des 2 arbres de Gomory-Hu en considérant ces deux réseaux simultanément. Dans la suite, nous présentons d'abord un cas plus simple au paragraphe 5.5.1 : calculer un arbre de Gomory-Hu pour une valeur intermédiaire du paramètre. Ensuite, nous présentons le problème général au paragraphe 5.5.2.

L'étude présentée dans ce paragraphe concerne les réseaux ne possédant qu'une seule arête paramétrée.

5.5.1 Recherche d'un arbre de Gomory-Hu intermédiaire

Dans ce paragraphe, il s'agit de résoudre le problème suivant : étant donné T^0 et T^∞ les arbres de Gomory-Hu des réseaux R^0 et R^∞ , est-il possible de calculer efficacement T^λ

correspondant au réseau R^λ ? Dans notre cas, la complexité recherchée doit être meilleure que $(n-1)MF(n, m)$.

Pour ce faire, nous allons examiner l'algorithme alternatif pour calculer les arbres de Gomory-Hu : l'algorithme de Gusfield [63]. On peut écrire de manière très synthétique cet algorithme de la manière suivante. On appelle une étoile à n sommets un arbre réduit à $(n-1)$ feuilles et une racine.

Algorithme 1: Gusfield(R)

▷ R est un réseau à n sommets.

▷ Retourne un arbre de Gomory-Hu T de R .

- 1 Calculer une étoile à n sommets T , étiquetés de 1 à n (la racine portant l'étiquette 1). Les arêtes ne sont pas valuées.
- 2 **pour** $s = 2$ à n
- 3 Soit t le voisin de s dans l'arbre T .
- 4 Calculer une coupe minimum dans R entre s et t .
- 5 Changer l'arbre en étiquetant l'arête $[s, t]$ avec $c(C_{s,t})$, et réarranger l'arbre de sorte que ce nouvel arbre reflète la coupe nouvellement calculée, tout en maintenant la validité des coupes précédentes.
- 6 **fin pour**

En ce qui concerne la complexité de cet algorithme, toutes les étapes peuvent se faire en un temps linéaire, sauf bien sûr l'étape 4 qui nécessite $MF(n, m)$. L'avantage de cet algorithme par rapport à notre problème est que toutes les coupes sont calculées sur le même réseau. Une conséquence directe du théorème 5.1 est alors :

Théorème 5.3 ([12]) *Soit R un réseau dont la capacité d'une arête e est paramétrée. Soit T^0 et T^∞ des arbres de Gomory-Hu pour les paramètres 0 et $+\infty$. Il est alors possible de calculer un arbre de Gomory-Hu du réseau R^λ , $0 < \lambda < \infty$, en temps $O(n^2)$.*

Idée de la preuve : Il suffit d'appliquer l'algorithme de Gusfield. Grâce au théorème 5.1, la complexité de l'étape 4 devient linéaire. □

On remarque que la donnée des deux arbres de Gomory-Hu suffit pour effectuer le calcul. Par ailleurs, ce théorème peut facilement être étendu au cas où k arêtes sont paramétrées.

5.5.2 Calcul d'un arbre de Gomory-Hu supplémentaire

Dans ce paragraphe, nous regardons le problème suivant : étant donné un réseau R possédant une arête e dont la capacité est paramétrée et un arbre de Gomory-Hu T^0 de R^0 , quelle est la complexité du calcul d'un arbre de Gomory-Hu de R^λ , $0 < \lambda \leq \infty$? La complexité qui nous intéresse ici est le nombre d'appels à la méthode de calcul de flot maximum. En effet, comme on a pu le voir dans la description de l'algorithme de Gusfield, la complexité du calcul d'un arbre de Gomory-Hu est bornée par $(n-1) \times M(n, m)$, toutes les autres opérations étant négligeables par rapport à $M(n, m)$. Nous allons résoudre le cas $\lambda = \infty$, cependant, la méthode est générale.

Pour résoudre ce problème, il est important de comprendre à quelle condition on peut réutiliser une partie de l'arbre T^0 dans l'arbre T^∞ . Pour cela, il suffit de regarder la figure 5.2 et de considérer le cas particulier où $f_{s,t}^0 = f_{s,t}^\infty$. Dans le cas général, par

application du lemme 5.2, une coupe minimum $C_{s,t}^0$ ne contient pas l'arête e . Il en résulte que cette coupe reste valide pour toute valeur de la capacité. On en déduit alors le lemme suivant.

Lemme 5.4 ([12]) *Soit R un réseau possédant une arête $e = [i, j]$ dont la capacité est paramétrée par λ . Soit T^0 un arbre de Gomory-Hu de R^0 . Soit s et t deux sommets du réseau. Si le chemin $P_{i,j}^0$ reliant i à j dans T^0 n'a pas d'arête en commun avec $P_{s,t}^0$ (chemin reliant s à t dans T^0), alors :*

$$\forall \lambda \quad f_{s,t}^\lambda = f_{s,t}^0.$$

Par ailleurs, une coupe minimum $C_{s,t}^0$ reste valide pour toutes les valeurs de λ .

En utilisant ce lemme et l'algorithme de Gusfield, on s'aperçoit que l'on peut gagner un certain nombre d'étapes. En effet, l'algorithme de Gusfield examine les sommets dans un ordre prédéfini : les sommets sont passés en revue de 2 à n . En considérant une autre numérotation des sommets du réseau, on peut utiliser les informations contenues dans T^0 afin de construire T^∞ . Nous montrons en particulier par cette méthode [12] qu'il existe un arbre de Gomory-Hu de R^∞ qui contient le plus grand sous-arbre de la forêt $T^0 \setminus P_{i,j}^0$, où $P_{i,j}^0$ est le chemin reliant les extrémités de l'arête paramétrée dans l'arbre T^0 (voir figure 5.4(a)). L'idée principale est de simuler l'algorithme de Gusfield jusqu'au point de construction du sous-arbre souhaité, puis de montrer que ce sous-arbre ne sera plus considéré par l'algorithme par la suite. Par conséquent, on peut commencer l'algorithme de Gusfield de l'état correspondant à la figure 5.4(b). Cependant, cette technique ne permet pas d'éliminer toutes les redondances possibles entre les deux calculs d'arbres. En effet, tous les sous-arbres de la forêt jouent un rôle équivalent. Nous avons donc cherché à en inclure le plus possible dans l'arbre recherché.

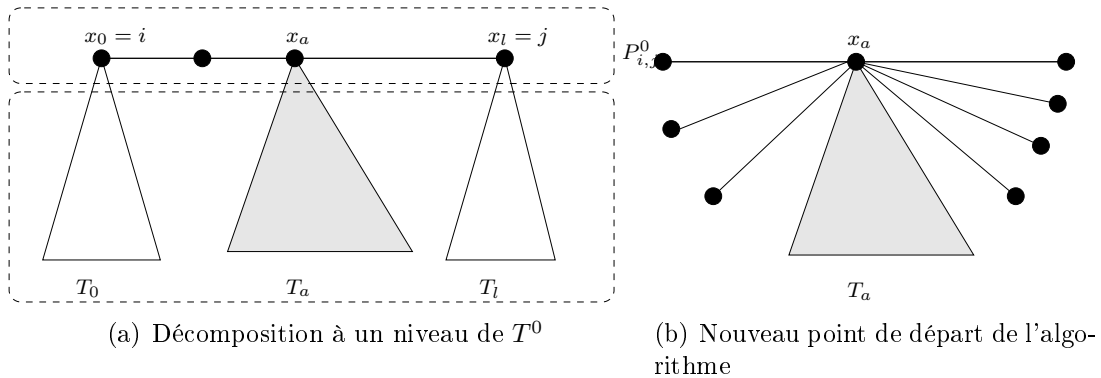


FIG. 5.4 – État initial du calcul de T^∞ par l'algorithme de Gusfield. L'arbre en gris est l'arbre le plus grand

Afin d'améliorer la complexité, nous sommes partis de l'algorithme original de Gomory-Hu [59]. Celui-ci est assez simple à comprendre (plus difficile à mettre en œuvre). Il utilise la notion de super-nœud. Un super-nœud correspond simplement à un regroupement de sommets du réseau.

Algorithme 2: Gomory-Hu(\mathbf{R})

▷ R est un réseau à n sommets.

- ▷ *Retourne un arbre de Gomory-Hu T de R*
- 1 Créer un super-nœud contenant tous les sommets de R .
 - 2 **tant que** il existe un super-nœud SN contenant plus d'un sommet **faire**
 - 3 Choisir s et t dans SN
 - 4 Considérer le réseau R' composé de tous les super-nœuds sauf SN qui a été développé.
 - 5 Calculer une coupe minimum $C'_{s,t}$ dans R' .
 - 6 Séparer SN en deux super-nœuds SN_1 et SN_2 reliés par une arête de poids la capacité de la coupe précédente, de sorte que tous les sommets de SN_1 sont d'un côté de la coupe et tous les sommets de SN_2 sont de l'autre côté. Relier les autres super-nœuds soit à SN_1 ou SN_2 en fonction de leur position dans $C'_{s,t}$.
 - 7 **fin tant que**

Pour cet algorithme, il est aussi intéressant de voir où se trouvent les possibilités de choix :

1. Le super-nœud à diviser. C'est un choix accessoire car il faut faire « disparaître » tous les super-nœuds qui ne sont pas composés d'un seul sommet.
2. Les sommets à l'intérieur du super-nœud (étape 3).
3. La coupe minimum entre les sommets s et t (étape 5).

À partir de ces diverses remarques, on obtient le théorème principal de ce paragraphe.

Théorème 5.4 ([12]) *Soit R un réseau, et $e = [i, j]$ une arête du réseau dont la capacité est paramétrée par λ . Soit T^0 un arbre de Gomory-Hu de R^0 et $P_{i,j}^0$ le chemin dans T^0 entre i et j . On peut construire un arbre de Gomory-Hu T^∞ de R^∞ en n'effectuant que $|P_{i,j}^0| - 1$ calculs de coupes minimum.*

Idée de la preuve : En utilisant les divers degrés de liberté de l'algorithme de Gomory-Hu, on peut simuler celui-ci afin d'obtenir la configuration suivante : un super-nœud composé de tous les éléments du chemin $P_{i,j}^0$ relié aux sous-arbres de $T^0 \setminus P_{i,j}^0$ (que l'on appelle une décomposition à deux niveaux comme le montre la figure 5.5(a)). Pour cela, on donne une nouvelle numérotation des sommets. On effectue ensuite les premières boucles de l'algorithme de Gomory-Hu en précisant les sommets concernés et quelle coupe considérer. On recrée alors tous les sous-arbres se trouvant en dehors du chemin $P_{i,j}^0$.

De ce nouvel état (figure 5.5(b)), il ne reste plus que $|P_{i,j}^0| - 1$ étapes de l'algorithme initial à effectuer. D'où la complexité de notre algorithme. \square

Commentaires généraux et ouvertures. Ce théorème apporte donc un intérêt supplémentaire au théorème 5.1. En effet, le surcoût du calcul de T^∞ à partir de T^0 n'est en général pas très important (le pire cas serait d'obtenir un arbre T^0 de forme linéaire et que les sommets reliés par l'arête étudiée correspondent aux extrémités de l'arbre). La complexité de la recherche d'un flot maximum est-elle aussi réduite du fait que l'on considère un réseau plus petit ? Une étude expérimentale complémentaire serait bienvenue pour étayer la pertinence de ces résultats. Cette étude pourrait apporter des éléments qualitatifs pour l'étude du cas multi-dimensionnel. En effet, calculer de manière efficace les $2^k - 1$ autres arbres à partir de l'arbre de base reste une question ouverte.

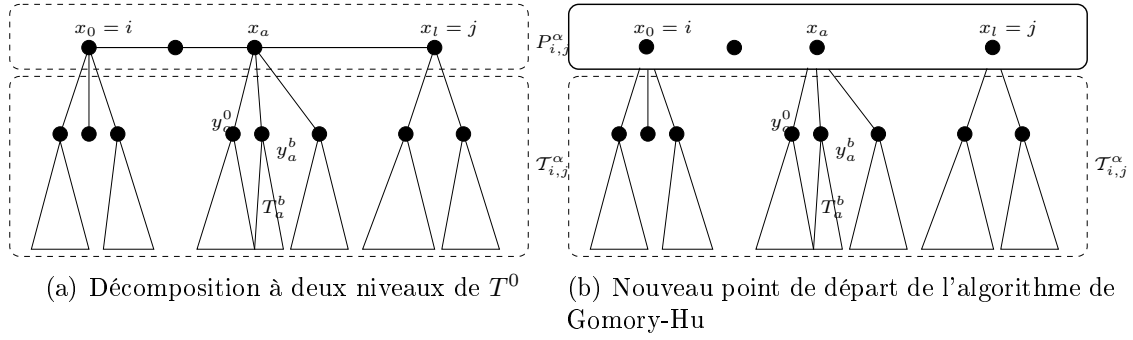


FIG. 5.5 – État initial de construction de T^∞ par l'algorithme de Gomory-Hu

Peut-on mettre une structure de type treillis au dessus de cette famille d'arbres? Est-ce qu'une telle structure apporte quelque-chose du point de vue algorithmique ou du point de vue de la complexité?

Une avancée importante serait de pouvoir construire une structure de donnée permettant de répondre en moyenne plus rapidement à la question du calcul des flots maximum.

Une dernière remarque concerne le sens de reconstruction d'un arbre à partir de l'autre. Nous avons construit T^∞ à partir de T^0 . Si on essaye d'effectuer l'opération inverse, on se heurte à la difficulté rencontrée par Elmaghraby dans son analyse de sensibilité, c'est-à-dire que l'on risque d'être confronté au grand nombre de valeurs critiques.

5.6 Les ε -arbres de Gomory-Hu

Une autre question se pose à partir du théorème 5.4 : est-il possible de construire le premier arbre de Gomory-Hu de sorte à minimiser la distance entre les deux sommets i et j , extrémités de l'arête paramétrée dans le réseau cible? C'est en partie pour répondre à cette question que nous avons mis en place les ε -arbres de Gomory-Hu. Nous présentons ensuite deux applications de ces arbres. Les résultats de ce paragraphe ne sont pas encore publiés. Nous détaillons donc un peu les preuves.

5.6.1 Définition

Comme nous l'avons précisé précédemment, un réseau donné peut avoir plusieurs arbres de Gomory-Hu. Dans l'article de Tucker, Hu et Shing [89], ceux-ci présentent de manière évasive le problème qui nous concerne :

Note that, in general, multiple trees exist which satisfy the GH tree definition. However, for simplicity in discussion we usually prefer the tree of smallest diameter, and pick its internal node of greatest sum of link capacities as the root.

Cependant, il n'est jamais fait mention de techniques permettant de trouver un tel arbre de Gomory-Hu. Comme nous l'avons déjà souligné pour l'algorithme de Gomory-Hu, il existe un certain nombre de degrés de liberté pour construire un arbre de Gomory-Hu. Dans ce paragraphe, nous limitons le nombre de coupes minimum qui peuvent être candidates (lors de l'étape 4 de l'algorithme 1 ou de l'étape 5 de l'algorithme 2) en donnant

une valeur symbolique à la capacité de l'arête paramétrée. Pour cela, nous introduisons une variante des arbres de Gomory-Hu dans laquelle on considère des modifications élémentaires de la capacité paramétrée en un point donné. Suivant la valeur de la capacité et sa position par rapport à la capacité critique, cette modification élémentaire aura ou non une influence sur la valeur du flot maximum.

Définition 5.7 Soit R un réseau et $e = [i, j]$ une arête du réseau. Un ε^+ -**arbre de Gomory-Hu** de R est un arbre de Gomory-Hu de R pour lequel la capacité de l'arête e est $c(e) + \varepsilon$, où ε représente une perturbation positive infinitésimale. Dans ce contexte, il est possible que $c(e) = 0$.

De manière similaire, si $c(e) > 0$, on définit un ε^- -**arbre de Gomory-Hu** de R comme l'arbre de Gomory-Hu de R pour lequel la capacité de e a subi une légère perturbation négative.

Un ε -**arbre de Gomory-Hu** est soit un ε^+ -arbre de Gomory-Hu soit un ε^- -arbre de Gomory-Hu.

On peut alors remarquer que dans de tels arbres, il existe deux types de capacités : celle qui sont constantes et celles qui sont paramétrées par ε .

L'avantage principal de cette structure est de trouver directement les sommets s et t pour lesquels $f_{s,t}$ est sensible à une variation de la capacité de l'arête e . Nous sommes obligés de considérer les deux cas afin de pouvoir considérer la singularité (le changement de pente) de la courbe donnant la valeur du flot maximum en fonction de la capacité (voir la figure 5.2). Avant de donner les propriétés de cette structure, nous précisons quelques points concernant son calcul.

Cette structure peut être calculée de manière aussi efficace que tout autre arbre de Gomory-Hu. Le schéma général suit les algorithmes classiques de Gomory-Hu ou Gusfield. La seule différence provient de l'algorithme de recherche d'une coupe minimum. Si le réseau ne comporte que des capacités entières, on peut choisir $\varepsilon = 1/2$: tous les flots non entiers seront alors sensibles à la variation de la capacité de e . Par contre, si les capacités sont réelles, il faut changer les techniques de comparaison entre différentes capacités en appliquant la règle

$$\forall x, y \quad x < y \implies x < x + \varepsilon < y \text{ et } x < y - \varepsilon < y. \quad (5.6)$$

On peut aussi remarquer qu'il n'y a pas nécessairement unicité des ces nouveaux arbres de Gomory-Hu. En voici des propriétés importantes.

Lemme 5.5 Soit R un réseau et $e = [i, j]$ une arête de R . Soit T^ε un ε -arbre de Gomory-Hu et $P_{i,j}^\varepsilon$ le chemin reliant i à j dans T^ε . Alors, les deux propriétés suivantes sont vérifiées :

1. La capacité d'une arête e' de T^ε dépend de ε si et seulement si e' appartient à $P_{i,j}^\varepsilon$.
2. T^ε fournit un arbre de Gomory-Hu de R en ignorant les ε .

Preuve : Afin de prouver le deuxième point, il suffit de remarquer que l'algorithme calculant les ε -arbres de Gomory-Hu ne diffère des algorithmes classiques que dans le choix des coupes minimum. Par ailleurs, chaque coupe minimum ainsi calculée n'est qu'une coupe minimum spécifique, qui reste valide dans le cas $\varepsilon = 0$.

Le premier point doit être montré pour chaque cas.

ε^+ -arbres de Gomory-Hu. En utilisant le lemme 5.4, une arête n'appartenant pas au chemin $P_{i,j}^\varepsilon$ ne peut pas avoir de capacité dépendant d' ε . Maintenant, éliminer toute arête e' sur le chemin $P_{i,j}^\varepsilon$ dans T^ε définit une coupe minimale, par définition de l'arbre de Gomory-Hu, qui sépare i et j dans T^ε , mais aussi dans R . Par conséquent, l'arête e est dans la coupe minimum et la capacité de cette coupe, qui correspond à la capacité de e' dans T^ε , dépend de ε .

ε^- -arbres de Gomory-Hu. On utilise dans ce cas un argument de même nature. Cependant, le lemme 5.4 ne peut considérer que des perturbations positives. Son extension au cas des perturbations négatives est direct : il suffit d'inverser les arguments en utilisant l' ε -arbre de Gomory-Hu comme cas de base.

□

5.6.2 Propriétés

Afin de trouver l'arbre de Gomory-Hu minimisant la distance entre deux sommets spécifiques, nous donnons un théorème plus général, dont les applications dépassent le cadre de ce paragraphe.

Théorème 5.5 *Soit R un réseau, $e = [i, j]$ une arête de R . Soit $T^{+\varepsilon}$ un ε^+ -arbre de Gomory-Hu de R et $T^{-\varepsilon}$ un ε^- -arbre de Gomory-Hu de R . Soit T un arbre de Gomory-Hu de R . Alors, nous avons l'inégalité⁴*

$$d_{T^{+\varepsilon}}(i, j) \leq d_T(i, j) \leq d_{T^{-\varepsilon}}(i, j) \quad (5.7)$$

Preuve : On note $P_{a,b}$ (respectivement $P_{a,b}^+$ et $P_{a,b}^-$) le chemin entre a et b dans T (resp. $T^{+\varepsilon}$ et $T^{-\varepsilon}$). On appelle X (resp. X^+ et X^-) l'ensemble des sommets de $P_{i,j}$ (resp. $P_{i,j}^+$ et $P_{i,j}^-$).

Commençons par prouver la première inégalité. Soit π^+ l'application $X^+ \rightarrow X$ telle que $\pi^+(v)$ est le plus proche voisin de v dans X . Par exemple, nous avons $\pi^+(i) = i$ et plus généralement, pour tout élément v de $X \cup X^+$ $\pi^+(v) = v$.

Ainsi définie, la fonction π^+ est injective. Supposons en effet le contraire. Alors, il existe deux sommets v et w tels que $\pi^+(v) = \pi^+(w)$. Comme $T^{+\varepsilon}$ et T sont des arbres de Gomory-Hu, et étant donnée la définition de π^+ , le chemin $P_{v,w}$ n'a pas d'arête en commun avec $P_{i,j}$. Donc, d'après le lemme 5.4, la valeur du flot maximum entre v et w n'est pas sensible à une variation positive ε de la capacité de l'arête e . Cependant, si on regarde $T^{+\varepsilon}$, v et w sont sur le chemin $P_{i,j}^+$, et en utilisant le lemme 5.5, la valeur du flot maximum est sensible à une variation positive de la capacité de e . Par conséquent, cela conduit à une contradiction et la fonction π^+ est injective.

On en déduit que, puisque tous les ensembles considérés sont finis, nous avons $|X| \geq |X^+|$, *i.e.*, la distance entre i et j dans T^ε est la plus petite parmi tous les arbres de Gomory-Hu R .

De même, pour la seconde inégalité, nous considérons l'application π^- l'application $X \rightarrow X^-$ qui associe le plus proche voisin d'un point dans X^- . Nous prouvons alors de la même manière que cette nouvelle application est injective.

⁴La notation $d_G(x, y)$ est la distance entre les sommets x et y dans le graphe G .

Tous ces éléments conduisent à la preuve du théorème. \square

L'application principale de ce théorème donne la réponse à la question initiale de ce paragraphe.

Corollaire 5.1 *Soit R un réseau, et e une arête paramétrée. On peut construire les deux arbres de Gomory-Hu T^0 et T^∞ en effectuant un minimum d'appels à la recherche de coupe minimum.*

Preuve : Il suffit de choisir pour T^0 l' ε -arbre de Gomory-Hu de R^0 et d'appliquer ensuite le théorème 5.4 pour calculer T^∞ . Le théorème 5.5 nous assure que le second arbre utilisera le moins d'appels à la méthode $MF(n, m)$. \square

Dans les problèmes de dimension supérieure, cette méthode ne semble effective que dans des cas particuliers : quand les arêtes paramétrées n'ont pas d'influence simultanément sur la valeur du flot maximum entre deux paires. Nous discutons au paragraphe 5.7 des moyens d'étudier ce problème.

5.6.3 Application : le problème ASF

Dans ce paragraphe, nous donnons une application de la deuxième inégalité du théorème 5.5. Cette application donne un nouvel éclairage au problème que nous avons présenté dans [13]. Le problème que nous souhaitons résoudre est le suivant.

Définition 5.8 ([13]) *Soit R un réseau et $e = [i, j]$ une arête de R . L'ensemble $\mathbf{ASF}[e]$ (Always Saturating Flows) est l'ensemble des couples de sommets $\{s, t\}$ tels que tous les flots maximum saturent l'arête e .*

$$ASF[e] = \left\{ \{s, t\} \in V^2 \mid \forall \vec{f}_{s,t} \quad |\vec{f}_{s,t}(e)| = c(e) \right\} \quad (5.8)$$

Cette définition permet de connaître *a priori* les flots qui seront particulièrement sensibles à la variation de la capacité de l'arête, en particulier dans le cas d'une dégradation.

L'ensemble $ASF(e)$ est caractérisé par le théorème suivant.

Théorème 5.6 ([13]) *Soit R un réseau et $e = [i, j]$ une arête de R telle que $c(e) = c_0 > 0$. Soit δ_0 une dégradation infime de la capacité de e . Soit R^{δ_0} le nouveau réseau et T^{δ_0} un arbre de Gomory-Hu de R^{δ_0} . Soit respectivement $f_{s,t}$ et $f_{s,t}^{\delta_0}$ la valeur du flot maximum entre s et t respectivement dans T et dans T^{δ_0} . Alors, on a*

$$\forall s, t \in V, \{s, t\} \in ASF[e] \Leftrightarrow f_{s,t}^{\delta_0} = f_{s,t} - \delta_0$$

On peut réécrire ce théorème en utilisant les ε -arbres de Gomory-Hu.

Théorème 5.7 *Soit R un réseau et $e = [i, j]$ une arête de R telle que $c(e) = c_0 > 0$. Soit $T^{-\varepsilon}$ un ε^- -arbre de Gomory-Hu.*

$$ASF[e] = \{ \{s, t\} \mid \text{le flot entre } s \text{ et } t \text{ dans } T^{-\varepsilon} \text{ dépend de } \varepsilon \}$$

5.7 Conclusions et perspectives de ce chapitre

Dans ce chapitre, nous avons présenté nos différentes contributions dans le domaine des multi-flots paramétrés. Quelques problèmes ouverts sont apparus au fil de la discussion. Ils concernent principalement le cas où plusieurs arêtes peuvent être paramétrées. Voici quelques autres perspectives de ce travail.

5.7.1 Algorithmes de calcul d'arbres de Gomory-Hu

D'un point de vue algorithmique, ou peut être plus pratique, il pourrait être intéressant de fournir un algorithme calculant un arbre de Gomory-Hu intermédiaire entre les deux algorithmes que nous avons présentés. L'algorithme de Gusfield a l'avantage de la simplicité d'écriture et de codage. Cependant, il reste gourmand en opérations : tous les calculs de flot maximum sont effectués sur le réseau initial. Celui original de Gomory-Hu est plus complexe à coder. Il nécessite de mettre en œuvre la notion de super-nœuds. Cependant, les calculs de flots maximum y sont plus simples car les réseaux considérés sont plus petits. Quel serait l'apport d'un algorithme travaillant en 2 phases : la première cherchant à diminuer la taille du réseau avec des opérations de type Gomory-Hu, la seconde travaillant sur des petits réseaux en utilisant l'algorithme de Gusfield. La compréhension synthétique de ces deux algorithmes permettrait de donner une réponse assez rapidement.

Il apparaît plus compliqué d'améliorer la complexité générale du calcul d'un arbre de Gomory-Hu en utilisant la méthodologie de calcul d'une suite d'arbres de Gomory-Hu. En partant d'un sous-réseau sur lequel il est simple (voire très simple) de calculer un arbre de Gomory-Hu, construire une suite d'arbres de Gomory-Hu afin d'obtenir l'arbre final : le calcul d'un sous-réseau *intéressant* (où le calcul de flot est élémentaire et très efficace) peut s'avérer être un problème très complexe en général. Par ailleurs, pour que cette méthode soit efficace, le nombre d'arêtes à rajouter doit au plus être linéaire.

5.7.2 Autres fonctions admettant des arbres de Gomory-Hu et paramétrisation

D'autres fonctions que la fonction de flot ont des représentations compactes de type arbres de Gomory-Hu. Une voie possible est de comprendre quelles sont les propriétés de cette fonction pour laquelle notre étude reste valable.

Un exemple proche des flots est celui des flots multiroutes. Cette notion a été introduite par Kishimoto and Takeushi pour résoudre des problèmes de fiabilité de réseau. Plusieurs études sur ce problème sont parues, en particulier un théorème de Max Flot/Min Cut [71]. La synthèse de réseau pour le problème multiterminal est abordé par Chandrasekaran, Nair, Aneja et Kabadi dans [35]. Dans [70], les mêmes auteurs ont montré l'existence d'arbres de coupes sous certaines conditions sur le nombre de routes que l'on s'autorise.

Les flots simples et les flots multiroutes ont des propriétés communes. Il est naturel de regarder l'extension de notre travail sur ce type de fonction.

5.7.3 ε -arbres de Gomory-Hu

Dans la section 5.6, nous avons montré comment trouver des arbres de Gomory-Hu qui minimisent ou maximisent la distance dans l'arbre entre deux nœuds du réseau. Deux directions principales peuvent être considérées pour étendre ces résultats.

1. **Autres propriétés.** Si on se réfère à l'article [89], il est possible de trouver un arbre de Gomory-Hu de diamètre minimum. Cependant, rien n'atteste de la difficulté du problème. Quelles sont les conditions, que l'on espère le plus simple possible, qui permettent de sélectionner les bons éléments dans les algorithmes de construction d'un tel arbre ? La question inverse se pose aussi : existe-t-il des critères que l'on ne peut satisfaire sans effectuer une recherche exhaustive des coupes minimales ? Sachant qu'il peut y avoir entre deux sommets un nombre exponentiel de coupes minimum [82], on peut supposer que cela donnerait des problèmes NP-complets ?
2. **Plusieurs arêtes paramétrées.** Peut-on donner un équivalent des ε -arbres de Gomory-Hu en dimension supérieure ? Cette question n'est pas si simple. En effet, considérons un flot paramétré par la capacité de 2 arêtes (formule 5.5). On peut distinguer, en dehors de la saturation, trois cas en fonction de la position relative des arêtes paramétrées :
 - (a) **arêtes *parallèles*** : la valeur du flot maximum suit indépendamment les évolutions des deux capacités ;
 - (b) **arêtes *série*** : les deux arêtes bloquent mutuellement l'évolution du flot.
 - (c) **cas général** : on se trouve dans une situation intermédiaire.

On pourrait donc, pour un flot particulier, définir l'association des deux arêtes avec une partie *parallèle* et une autre *série*. Cependant, cette répartition ne concerne qu'une seule paire de sommets du réseau, et devra être recalculée pour toutes les autres paires.

La majeure partie des réponses viendra sans doute dans la meilleure compréhension de la structure de donnée *arbre de Gomory-Hu* et de ses rapports avec le réseau initial.

5.7.4 Mineurs de graphes et flots

La propriété 5.1 peut être considérée avec une vision d'un peu plus haut niveau. On s'aperçoit que l'on travaille avec deux types de réseaux afin d'effectuer les calculs : R^0 et R^∞ . On remarque que du point de vue théorie des graphes, le deuxième réseau consiste à fusionner les deux sommets de chaque côté de l'arête concernée, ou plus simplement effectuer la contraction de l'arête e . Si e est l'arête paramétrée, on travaille donc avec $R \setminus e = R^0$ et $R/e = R^\infty$. Est-ce que ce formalisme peut alors être utilisé pour donner de nouvelles identités ?

Chapitre 6

L'arbre de cliques augmenté : une autre représentation d'un graphe

Ce chapitre présente une direction nouvelle de mes recherches depuis l'année 2003. Je me suis intéressé aux polynômes chromatiques des graphes : au travers d'un projet de maîtrise, j'ai essayé d'en cerner les diverses propriétés, pour rechercher des conditions simples pour que deux graphes aient le même polynôme chromatique. Avec K. Nguyen et S. Lebresne en particulier, nous avons mis au point une méthode de calcul de ces polynômes basée sur la triangulation des graphes. Ce travail a fait l'objet de la publication [29]. Plusieurs autres étudiants ont contribué à faire avancer ce projet afin d'affiner les techniques mises en place dans le travail précédent. Il s'agit d'A. Vieillerivière, M. Sozeau, P-L. Garoche, S. Billard et I. Briquel.

Après quelques études complémentaires, la représentation des graphes choisie est le principal élément qui permet d'effectuer les opérations de calcul du polynôme chromatique de manière assez efficace. Ce calcul devient alors une (principale) application possible de cette représentation avec les opérations que l'on effectue dessus. Nous essayons de montrer dans cette partie le potentiel de cette représentation.

6.1 Introduction

Introduit en 1946 par Birkhoff et Lewis, le polynôme chromatique d'un graphe permet de dénombrer les colorations *propres* de ce graphe. Au travers de ses coefficients ou de la valeur en certains points, ce polynôme capture aussi un certain nombre de propriétés combinatoires des graphes, comme le nombre chromatique, mais aussi le nombre d'orientations acycliques ou d'arbres couvrants [88]. On trouve aussi quelques applications en physique théorique [36]. De nombreuses autres études ont été réalisées sur ce polynôme, en particulier la localisation de ses racines réelles ou complexes [69].

Quelques articles traitent du calcul effectif du polynôme chromatique en utilisant les opérations élémentaires de contraction et la suppression d'arêtes, opérations élémentaires permettant de décrire tous les mineurs d'un graphe donné¹. Ce calcul est basé sur une méthode récursive, les récursions agissant sur des transformations élémentaires du graphe initial. Les cas de base correspondent aux graphes pour lesquels il est facile de calculer

¹Voir l'annexe B pour une définition précise de cette notion.

ce polynôme. Les cas les plus simples sont les graphes vides et les arbres. Haggard [65] a recherché une famille plus large de tels graphes et a poussé ce raisonnement à l'extrême : si un graphe est « suffisamment » petit, on regarde si son polynôme chromatique n'a pas déjà été calculé. On constitue ainsi un dictionnaire des polynômes chromatiques des petits mineurs du graphe cible. Cette méthode utilise par ailleurs la détection de graphes isomorphes [77]. À l'instar de [88], nous avons utilisé les graphes triangulés comme cas de base. Cependant, nous avons utilisé les récursions basées sur l'ajout d'arête et la fusion de deux sommets non voisins.

Un autre élément important de notre étude est la famille des graphes triangulés. Cette famille est caractérisée de différentes manières [66]. Par exemple, un graphe triangulé est un graphe dans lequel tout cycle de longueur > 3 possède une corde. Cette famille possède de bonnes propriétés algorithmiques : un certain nombre de problèmes difficiles (en particulier, NP-complets) deviennent faciles à résoudre (en l'occurrence, polynomiaux) dans cette famille. Une triangulation d'un graphe général consiste à rajouter des arêtes à ce graphe pour qu'il devienne triangulé. Les applications des triangulations se retrouvent dans divers domaines comme pour les multiplications de matrices creuses, en gestion de bases de données ou la vision par ordinateur (voir les références incluses dans [66]).

Dans la suite de ce chapitre, nous présentons les diverses définitions importantes au paragraphe 6.2, en particulier celle de l'arbre de cliques augmenté. Au paragraphe 6.3, nous montrons comment cette structure est utilisable dans le calcul du polynôme chromatique. Au paragraphe 6.4, nous donnons quelques résultats pratiques. Enfin, le paragraphe 6.5 montre le potentiel des arbres de cliques augmentés.

6.2 Définitions

Dans ce paragraphe, nous présentons rapidement les deux outils essentiels de cette étude, à savoir les polynômes chromatiques et les triangulations.

6.2.1 Polynômes chromatiques

Étant donné un graphe $G = (V, E)$, on appelle une **coloration** avec k couleurs de G une fonction $\phi : V \mapsto I_k$, où $I_k = \{1, \dots, k\}$. Une **coloration propre** de G avec k couleurs est une fonction des sommets dans \mathbb{N} telle que deux sommets voisins sont de couleurs différentes.

Définition 6.1 *Le polynôme chromatique du graphe G , noté $P(G, \lambda)$, est une fonction $\mathbb{N} \rightarrow \mathbb{N}$ définie par*

$$P(G, \lambda) = |\{\phi : V \mapsto I_\lambda \mid \phi \text{ coloration propre de } G\}|$$

Comme nous le verrons par la suite, la dénomination « polynôme » de cette fonction est justifiée et nous considérerons son extension naturelle dans le corps des réels. Voici les polynômes chromatiques de quelques graphes simples.

Graphe vide : $P(\bar{K}_n, \lambda) = \lambda^n$;

Graphe complet : $P(K_n, \lambda) = \prod_{i=0}^{n-1} (\lambda - i) = \lambda^{(n)}$;

Arbres : $P(T_n, \lambda) = \lambda(\lambda - 1)^{n-1}$.

Chacune de ces trois familles de polynômes constituent une base de l'espace vectoriel des polynômes sans terme constant. De ce fait, les polynômes chromatiques sont souvent décrits dans l'une de ces trois bases. Shier et Chandrasekaran ont présenté quelques propriétés des coefficients du polynôme chromatique d'un graphe pour chacune de ces bases [88].

Toutes les méthodes de calcul sont basées sur les deux équations suivantes qui sont équivalentes².

$$P(G, \lambda) = P(G + e, \lambda) + P(G/e, \lambda), \text{ si } e \notin E \quad (6.1)$$

$$P(G, \lambda) = P(G - e, \lambda) - P(G/e, \lambda), \text{ si } e \in E \quad (6.2)$$

Ces deux formulations définissent un schéma récursif pour calculer le polynôme. Le cas de base si on utilise la première est le graphe complet. Pour la deuxième formulation, on peut s'arrêter aux graphes vides ou aux arbres. Ces deux équations montrent en particulier que l'objet que l'on considère est bien un polynôme comme somme finie (mais grande) de polynômes élémentaires. Le résultat suivant permet d'améliorer l'efficacité du calcul en diminuant la taille des graphes considérés.

Lemme 6.1 ([69]) *Soit G un graphe, G_1 et G_2 deux sous-graphes de G tels que $G = G_1 \cup G_2$ et $G_1 \cap G_2 \sim K_r$. On a alors :*

$$P(G, \lambda) = \frac{P(G_1, \lambda)P(G_2, \lambda)}{\lambda^{(r)}} \quad (6.3)$$

Ce lemme permet de calculer le polynôme chromatique des arbres en éliminant les feuilles une à une. Par ailleurs, il met en évidence l'importance des séparateurs minimaux complets dans le calcul du polynôme.

Définition 6.2 *Étant donné un graphe $G = (V, E)$, un ensemble S de sommets est appelé un **séparateur** s'il existe deux sommets s et t dans G tels que ces deux sommets sont dans deux composantes connexes distinctes de $G \setminus S$. L'ensemble S est appelé **séparateur minimal** s'il est minimal par inclusion. Il est dit **séparateur minimal complet** s'il induit une clique sur G .*

6.2.2 Triangulations, Arbres de cliques

Un graphe **triangulé** est un graphe qui ne contient pas de cycle induit de longueur strictement supérieure à 3. Cette classe de graphes est intéressante car nombre de problèmes NP-complets dans le cas général deviennent polynomiaux dans cette classe, comme la coloration minimale de graphes ou la clique maximum. Une représentation des graphes triangulés est l'**arbre de cliques** dont on peut donner la définition suivante [6, 20, 54].

Définition 6.3 *Un **arbre de cliques** d'un graphe triangulé $G = (V, E)$ est un arbre $T = (\mathcal{V}, \mathcal{E})$ tel que :*

²Voir l'Annexe B pour les notations.

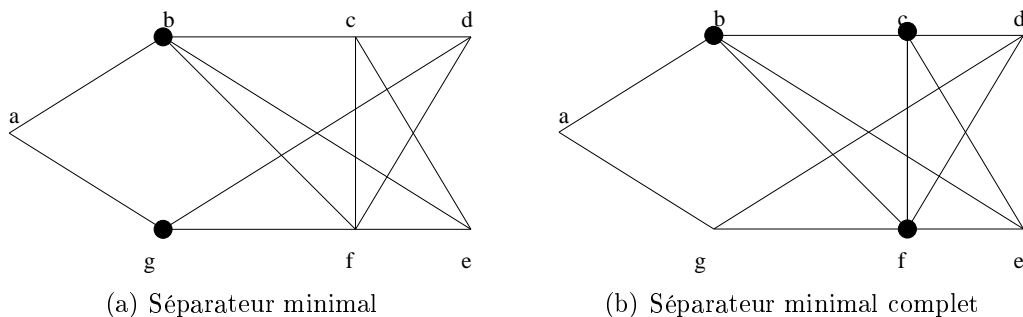


FIG. 6.1 – Deux séparateurs minimaux. Les sommets séparés sont a et e .

- $\mathcal{V} = \{C_1, C_2, \dots, C_j\}$ est l'ensemble des cliques maximales de G ;
- pour chaque sommet v de G , l'ensemble des sommets de T contenant v induit un sous-arbre de T .

Cette définition donne une décomposition arborescente particulière d'un graphe triangulé : ce sont les décompositions dont l'ensemble des sommets représentent les cliques maximales du graphe.

Nous donnons un exemple simple à la figure 6.2. Comme il est montré en particulier dans [54], il n'y a pas unicité des arbres de cliques pour un graphe triangulé donné. Il faut par ailleurs remarquer que chaque arête de l'arbre de cliques induit dans le graphe initial un séparateur minimal complet dans G . Cette caractérisation permet de calculer le polynôme chromatique des graphes triangulés de manière simple en utilisant la formule 6.3.

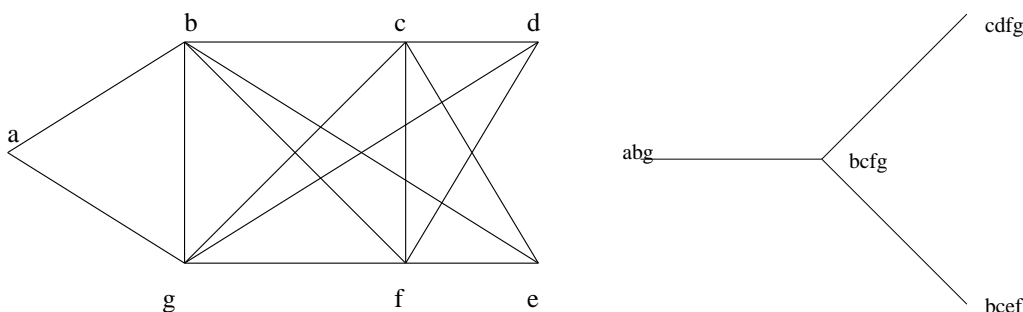


FIG. 6.2 – Un graphe triangulé et un de ses arbres de cliques

Pour un graphe général $G = (V, E)$, une **triangulation** de G est un graphe triangulé $G' = (V, E \cup F)$, avec F un ensemble d'arêtes disjoint de E . Par abus de langage, on appellera aussi F une triangulation³ de G . Une triangulation est dite minimale si elle est minimale par inclusion. On peut trouver une triangulation minimale en temps polynomial [66]. Il existe en effet diverses caractérisations des triangulations qui permettent de tester la minimalité de celle-ci. Il existe deux problèmes classiques d'optimisation reliés aux triangulations :

³L'ensemble F est aussi appelé *fill-in* dans la littérature.

- le problème du **minimum fill-in** cherche à minimiser le nombre d'arêtes à rajouter dans la triangulation ($|F|$). Ce problème est NP-Complet [91] et si l'optimum est k , il existe un algorithme polynomial trouvant un fill-in de taille au plus $8k^2$ [80].
- le problème de la **largeur arborescente minimale** consiste à rechercher une triangulation pour laquelle la taille de la clique maximale est minimale. Ce problème est lui aussi NP-complet [5]. Récemment, Feige *et al.* ont trouvé un algorithme d'approximation en $O(\sqrt{k})$, où k représente la valeur optimale du paramètre [49].

Par ailleurs, il existe un certain nombre d'heuristiques calculant des triangulations minimales [66]. L'une des plus populaires est celle du *degré minimum* qui cherche à compléter les voisinages des sommets de plus petit degré en premier. Ces heuristiques donnent en général des solutions de bonne qualité pour les deux problèmes cités plus haut. Cependant, ce ne sont pas des algorithmes d'approximation.

Afin de représenter les graphes généraux, nous avons proposé dans [29] une représentation des graphes à l'aide d'une de leurs triangulations. L'idée principale est de représenter le graphe par l'une de ses triangulations minimales, et de marquer les arêtes qui ont été rajoutées au graphe d'origine. La triangulation sera représentée par un de ses arbres de cliques. Pour une triangulation minimale, on sait que les arêtes rajoutées font toutes partie de séparateurs minimaux du graphe triangulé. Par ailleurs, tous les séparateurs minimaux d'un graphe triangulé correspondent exactement aux intersections de deux cliques voisines dans l'arbre de cliques. Par conséquent, on peut étiqueter chaque arête de l'arbre de cliques, c'est-à-dire, chaque séparateur minimal, par l'ensemble des arêtes issues de la triangulation contenues dans ce séparateur minimal. Nous donnons une définition plus formelle de cette structure de données ainsi qu'un exemple simple dans la figure 6.3.

Définition 6.4 Soit $G = (V, E)$ un graphe et F une triangulation minimale de G . Soit $T = (\mathcal{V}, \mathcal{E})$ un arbre de cliques de $G' = (V, E \cup F)$. L'**arbre de cliques augmenté** (ou *ACT*) de G par rapport à F est un étiquetage ϕ des arêtes de T tel que :

$$\phi(V_i, V_j) = F \cap E(G'[V_i \cap V_j]),$$

où $G'[U]$ représente le sous-graphe de G' induit par U , $U \subset V(G')$. Par extension, on appellera *arbre de cliques augmenté* la donnée de T et de ϕ .

Dans le paragraphe suivant, nous présentons comment effectuer certaines opérations de modification de graphe directement sur la représentation d'arbre de cliques augmenté.

6.3 Opérations sur l'arbre de cliques augmenté

Afin d'effectuer le calcul du polynôme chromatique d'un graphe, nous avons utilisé la représentation par arbre de cliques augmenté. Comme nous l'avons montré précédemment, ce calcul est basé sur l'équation 6.1 et sur le lemme 6.1 dit de séparation. Nous présentons les conséquences de ces deux résultats sur la structure de données choisie.

Dans tout ce paragraphe, le graphe $G = (V, E)$ admet une triangulation F et un ACT $T = (\mathcal{V}, \mathcal{E}, \phi)$.

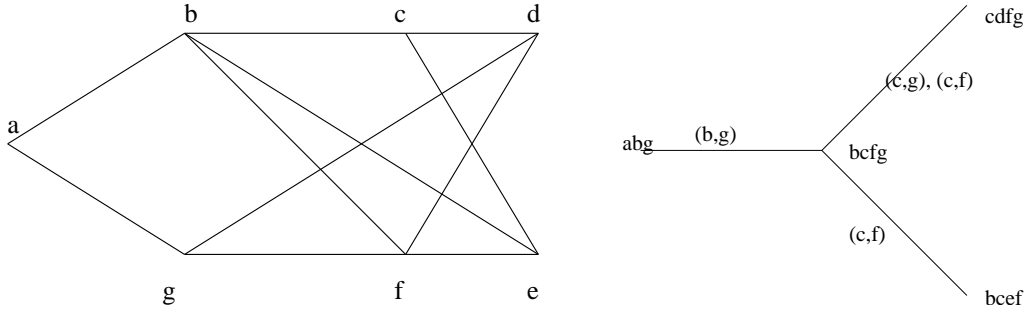


FIG. 6.3 – Un graphe et son arbre de cliques augmenté pour la triangulation $F = \{(b, f), (b, g), (c, f)\}$

6.3.1 Fusion de sommets

L'équation 6.1 induit la création de deux graphes pour effectuer la récursion à partir d'une non-arête donnée $e = (a, b)$: le graphe $G + e$ et le graphe G/e . Si on choisit l'arête dans la triangulation F , on minimise l'impact sur l'ACT associé. En particulier, il est clair qu'un ACT associé à $G + e$ peut être obtenu immédiatement :

- on peut choisir le même arbre de cliques T ;
- la fonction d'étiquetage ϕ' est telle que pour tout séparateur minimal complet S de $G' = (V, E \cup F)$, c'est-à-dire, pour chaque arête de l'arbre de cliques, $\phi'(S) = \phi(S) \setminus S$.

L'apport principal de notre travail a été de montrer que l'opération de fusion de sommets peut être effectuée efficacement si l'arête considérée appartient à la triangulation choisie. Le principe général de cette opération consiste voir que quand on fusionne deux sommets, on crée, dans la représentation ACT, des cliques qui ne sont plus maximales. Il suffit alors de réduire l'ACT pour ne conserver que les cliques maximales. L'algorithme plus précis est donné par la suite. On suppose que tous les sommets ont une étiquette unique.

Algorithme 1: FusionSommets(T, a, b)

- ▷ $T = (\mathcal{V}, \mathcal{E}, \phi)$ est un ACT d'un graphe $G = (V, E)$ suivant une triangulation F .
 - ▷ a et b sont deux sommets de G tels que l'arête $(a, b) \in F$.
 - ▷ renvoie $T' = (\mathcal{V}', \mathcal{E}', \phi')$ est un ACT de G/e
- 1 $T' \leftarrow T$
 - 2 Pour chaque $X \in \mathcal{V}'$, renommer a en b , et éliminer les doublons si nécessaire.
 - 3 Pour chaque arête $e \in \mathcal{E}'$, renommer toutes les occurrences de b par a dans $\phi'(e)$. Éliminer aussi de $\phi'(e)$ les arêtes apparues dans le graphe du fait de la contraction.
 - 4 Réduire l'arbre de cliques pour qu'il ne conserve que des cliques maximales :
 - 5 Tant qu'il existe deux cliques voisines dans T' X et Y telles que $X \subset Y$, contracter l'arête (X, Y) dans T'
 - 6 **retourne** T' .

L'exemple de la figure 6.4 montre les différentes étapes de cet algorithme sur le cycle à 7 sommets.

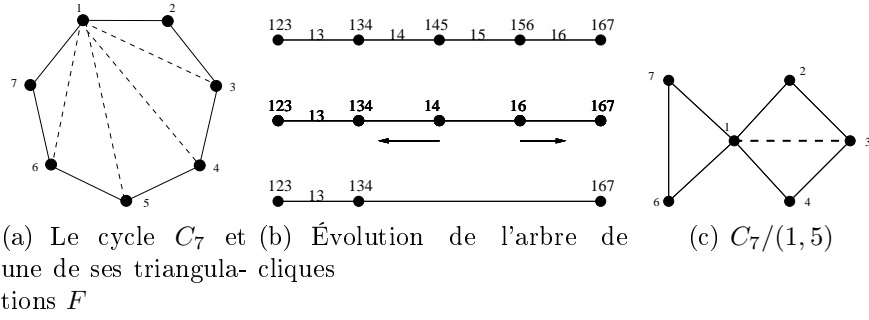


FIG. 6.4 – Fusion de sommets dans le cycle. Les triangulations sont notées en pointillés.

L'algorithme final est un peu plus complexe. En effet, il se peut que la triangulation obtenue après fusion de sommets ne soit plus minimale comme le montre la figure 6.5. Plusieurs techniques peuvent être mises en place :

- conserver un arbre de cliques qui n'est pas réduit. Cette solution est utilisable dans le cas du calcul du polynôme chromatique, mais elle implique des récursions inutiles.
- travailler localement dans l'arbre de cliques augmenté pour obtenir une triangulation minimale. Dans les dernières mises en œuvre de l'algorithme, nous avons proposé une version simplifiée de cette méthode. La version complète et détaillée doit être abordée prochainement (voir paragraphe 6.5.3).

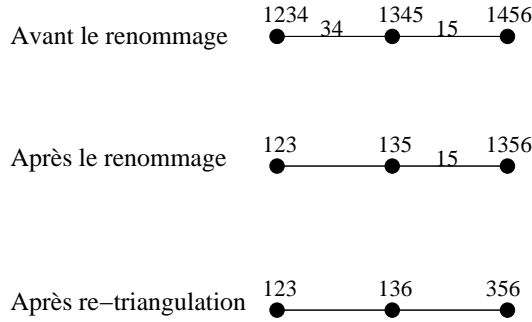


FIG. 6.5 – Évolution d'un ACT donnant une triangulation non minimale. Ici les sommets fusionnés sont 3 et 4. La clique résultante 135 est incluse dans 1356, mais l'arête 15 reste à rajouter.

Dans tous les cas, on peut déduire le théorème suivant.

Théorème 6.1 *Soit G un graphe, F une triangulation de G , et T un ACT de G . Soit $e \in F$. Alors, il existe F_1, F_2, T_1, T_2 tels que :*

1. F_1 est une triangulation de $G + e$ et T_1 un ACT de $G + e$;
2. F_2 est une triangulation de G/e et T_2 un ACT de G/e ;

Ces ensembles peuvent être obtenus en temps quadratique $O(|V(G)|^2)$ à partir de F et T .

6.3.2 Séparation

Le lemme de séparation permet de diminuer efficacement le nombre de récursions à faire lors du calcul effectif du polynôme chromatique.

Le point essentiel dans ce paragraphe est de voir quand ce lemme peut s'appliquer, en considérant la représentation du graphe par un ACT. Cette opération peut s'effectuer quand un séparateur minimal est complet, c'est-à-dire quand l'étiquette d'une arête de l'ACT est vide. Dans ce cas, on peut diviser le calcul en deux parties, ce qui revient à travailler sur les deux parties de l'arbre indépendamment. Cette opération permet un gain en temps significatif car les graphes résultants sont de taille réduite.

Ce résultat se résume dans le théorème suivant :

Théorème 6.2 *Soit G un graphe et G_1, G_2 deux sous-groupes de G tels que $G = G_1 \cup G_2$ et $G_1 \cap G_2 \sim K_r$. Soit F une triangulation de G et T un ACT de G . La triangulation de G peut être divisé en deux triangulations F_1 et F_2 telles que F_i est une triangulation de G_i ($i \in \{1, 2\}$). De plus, les ACT résultants peuvent être obtenus simplement à partir de T .*

En effet, il suffit de considérer les deux sous-arbres obtenus en éliminant l'arête qui représente le séparateur minimal.

6.4 Application au calcul du polynôme chromatique

6.4.1 Algorithme général

À partir de ces résultats théoriques, on en déduit l'algorithme général de calcul du polynôme chromatique. On rappelle que le polynôme chromatique d'un graphe triangulé est simple à calculer, par applications successives du lemme de séparation.

Algorithme 2: Chromatic-Polynomial(G, T)

```

▷  $T$  est un ACT de  $G$ 
▷ Retourne le polynôme chromatique de  $G$ 
1  début
2    si  $G$  est triangulé alors retourne la formule des graphes triangulés.
3    si  $\exists e \in T$  tel que  $\phi(e) = \emptyset$ 
4      Décomposer  $G$  avec le théorème 6.2 :
5        Soit  $G_1, T_1, G_2, T_2$  et  $K_r$  les éléments de cette décomposition
6         $P_1 \leftarrow$  Chromatic-Polynomial( $G_1, T_1$ )
7         $P_2 \leftarrow$  Chromatic-Polynomial( $G_2, T_2$ )
8        retourne  $P_1 \times P_2 / P(K_r)$ 
9    Soit  $e =$  ChoiceFunction( $G, T$ )
10   Avec le théorème 6.1, on calcule
11    $G_1 = G + e$ , et l'arbre de cliques  $T_1 = T$ 
12    $G_2 = G/e$ , et l'arbre de cliques  $T_2$ 
13   retourne Chromatic-Polynomial( $G_1, T_1$ ) + Chromatic-Polynomial( $G_2,$ 
14    $T_2$ )
14  fin

```

Cet algorithme calcule de manière correcte le polynôme chromatique d'un graphe. Cependant, ce schéma est générique et laisse à l'utilisateur une possibilité de réglage importante. Le choix principal se situe au niveau de l'étape 9. L'algorithme sera efficace si l'étape 3 est appliquée régulièrement et qu'elle sépare les graphes en deux parties

équivalentes en taille. Plusieurs stratégies de choix ont été testées lors du Master de S. Billard, chacune d'elle a son champ d'application.

Afin d'illustrer l'importance de ce choix, on a calculé le nombre de nœuds de l'arbre de récursion dans le cas d'un cycle C_n suivant différentes stratégies. La triangulation choisie est celle montrée à la figure 6.4(a) :

arête la plus à gauche : $O(\rho^n)$, où ρ est le nombre d'or ;

arête la plus centrale : $O(n^2)$.

Dans ce cas particulier, la meilleure stratégie est d'utiliser l'équation 6.2, où le nombre de nœuds est linéaire.

6.4.2 Résultats expérimentaux

Cet algorithme a été testé sur un certain nombre d'instances. Le code principal a été mis en place par S. Lebresne et K. Nguyen, une mise au point a été apportée par S. Billard et surtout I. Briquel, lors de stages respectifs de maîtrise et licence. Il a été réalisé en OCAML.

Afin d'évaluer la qualité d'une triangulation, nous avons défini le paramètre d'**épaisseur de triangulation** comme la taille maximale d'une étiquette d'un arbre de clique associé. Dans l'exemple 6.3, l'épaisseur de triangulation est 2. Cette épaisseur est évidemment nulle pour tout graphe triangulé. Pour les cycles, ce paramètre vaut 1.

De manière générale, l'algorithme est efficace pour des instances dont l'épaisseur de triangulation est petite. Dans le cas des graphes aléatoires, nous avons testé les différentes densités de graphes afin de faire une étude statistique. Il apparaît que l'on peut résoudre de manière systématique *tous* les graphes ayant moins de 22 sommets. Le problème principal est lié à la réservation de mémoire engendrée par l'utilisation de très grands entiers. Au delà de 25 sommets, seules les densités faibles et très grandes ont été résolues par notre méthode.

20 sommets								
densité	20	30	40	50	60	70	80	90
#arêtes ACT	14.2	12.00	9.97	8.27	6.9	5.8	4.65	3.52
diamètre ACT	6.42	5.86	5.02	4.45	3.94	3.43	2.83	2.27
Taille max Clique	6.67	8.99	11.00	12.73	14.09	15.19	16.35	17.48
Épaisseur	8.7	15.86	22.26	25.61	25.51	23.17	17.8	10.26
Temps (s)	0.25	3.02	16.30	39.17	36.62	24.29	6.68	0.49
23 sommets								
densité	20	30	40	50	60	70	80	90
#arêtes ACT	16.03	12.87	10.62	8.78	7.23	5.94	4.86	3.72
diamètre ACT	7.04	6.07	5.29	4.71	4.11	3.45	2.91	2.46
Taille Max Clique	7.91	11.13	13.35	15.21	16.76	18.06	19.14	20.28
Épaisseur	13.98	26.82	35.49	39.74	39.23	33.45	25.17	13.96
Temps (s)	2.98	112.95	465.65	1189.29	1540.93	867.54	121.00	4.29
Résolus/10mn	-/100	99/97	85/63	74/24	78/21	100/54	100/98	-/100

25 sommets								
densité	20	30	40	50	60	70	80	90
#arêtes ACT	16.87	13.49	10.99	9.14	7.57	6.27	5.03	3.81
diamètre ACT	7.01	6.13	5.33	4.83	4.32	3.71	3.09	2.43
Taille Max Clique	9.10	12.5	15.01	16.85	18.43	19.73	20.97	22.19
Épaisseur	19.84	35.21	46.2	49.83	48.11	42.63	31.32	18.23
Temps (s)	48.20	474.87	1045	906	982	948	781.61	24.98
Résolus/10mn	100/98	73/53	16/3	3/1	4/1	16/4	87/41	-/100

6.5 Conclusions et perspectives

À partir de cette première étude, nous avons montré l'intérêt d'une modélisation des graphes par les ACT. Plusieurs axes de recherche sont alors possibles.

6.5.1 Autres calculs à partir de l'arbre de cliques augmenté

Les ACT peuvent directement être utilisés pour résoudre toute une famille de problèmes. En effet, tout problème que l'on peut écrire sous la forme :

$$f(G) = g(f(G + e), f(G/e)),$$

avec e n'appartenant pas à $E(G)$, et tel que le calcul du paramètre soit facile sur la famille des graphes triangulés peut utiliser cette méthode de résolution.

L'exemple le plus simple dans cette catégorie est le calcul du nombre chromatique. En effet, nous avons de manière simple à partir de notre étude :

$$\chi(G) = \min(\chi(G + e), \chi(G/e)).$$

Par ailleurs, le lemme de séparation peut se résumer ainsi. S'il existe un séparateur minimal complet, le nombre chromatique est le maximum des nombres chromatiques des deux sous-graphes que l'on obtient en séparant le graphe. La transcription de ces propriétés dans le formalisme des ACT est alors directe. De plus, le nombre chromatique d'un graphe triangulé est très simple à calculer : c'est la taille de la clique maximale. On peut en déduire un algorithme dérivé du nôtre qui détermine le nombre chromatique. Cet algorithme pourrait bénéficier de plus des remarques suivantes :

- à chaque étape de la récursion, il est possible d'évaluer des bornes inférieures et supérieures du nombre chromatique des graphes dérivés, ces évaluations permettent d'élaguer l'arbre de recherche. On peut y associer des heuristiques de coloration simple. L'algorithme obtenu serait alors de type branch-&-bound ;
- la fonction calculée est réduite à un seul entier : le critère de la mémoire utilisée serait moins critique.

De cette étude peut ressortir d'autres questions concernant la complexité exacte du calcul du nombre chromatique. Des bornes sont connues dans le cas général [18], toutes de type exponentiel. Le problème étant NP-complet, il est *a priori* illusoire de chercher à éliminer l'exponentielle, mais il est peut être plus « abordable » de chercher à diminuer le facteur de l'exponentielle. Il serait important de déconnecter ces deux notions ou de montrer qu'elles sont reliées, comme nous l'avons fait pour le minimum fill-in.

6.5.2 Quel(s) critère(s) d'évaluation d'une triangulation

Nous avons relié expérimentalement la qualité d'une triangulation au paramètre d'épaisseur de triangulation. En effet, pour pouvoir séparer une arête de l'ACT, il faut compléter le séparateur, c'est-à-dire, ajouter les arêtes (de la triangulation) contenues dans l'étiquette de l'arête (de l'ACT). Effectuer cette opération induit un arbre de récursion exponentiel en l'épaisseur de cette arête (de l'ACT) avant de pouvoir appliquer le lemme de séparation. Par ailleurs, nous avons montré que le choix de l'arête à rajouter n'influe pas localement sur la taille de l'arbre de récursion. L'impact de la contraction d'une arête (de la triangulation) peut se situer en dehors de l'arête (de l'ACT) choisie. Il serait important de déconnecter ces deux notions ou de montrer qu'elles sont reliées, comme nous l'avons fait pour le minimum fill-in. En effet, d'autres séparateurs minimaux peuvent être complétés plus rapidement.

La forme de l'arbre et la disposition des arêtes semblent donc aussi importantes que l'épaisseur. La répartition des arêtes paraît un meilleur critère : l'impact de la contraction d'une arête peut être très importante. Sans doute le but de l'algorithme associé (calcul du polynôme chromatique, ...) peut déterminer le(s) critère(s) pertinent(s). Cependant, parmi les mesures que l'on peut mettre sur une triangulation en relation avec nos algorithmes, l'épaisseur reste la plus facile à définir.

Dans ce sens, il est important de définir la difficulté de la recherche de la triangulation d'épaisseur minimale, celle-ci étant sûrement NP-dur. En la reliant à la largeur arborescente, qui est aussi une notion locale du graphe, il est possible d'en donner une approximation grossière. Dans [29], nous avons montré que les deux paramètres d'évaluation d'une triangulation que sont le minimum fill-in et l'épaisseur de triangulation ne sont pas minimaux pour les mêmes triangulations. L'exemple de la figure 6.6 illustre cette propriété.

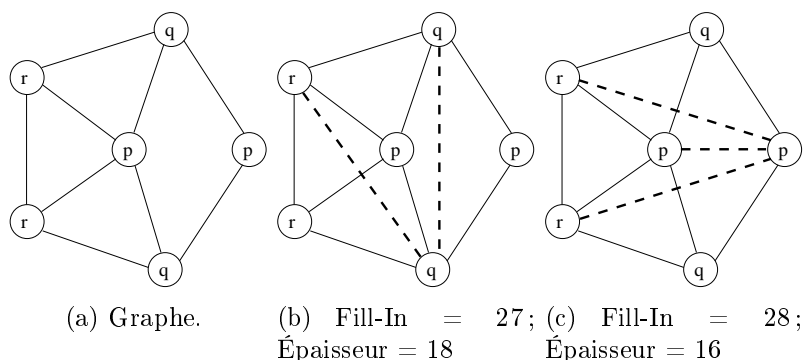


FIG. 6.6 – Une graphe pour lequel le minimum fill-in et l'épaisseur de triangulation ne sont pas minimisés pour les mêmes triangulations. Chaque cercle symbolise un graphe complet et chaque arête représente un graphe biparti complet. Les triangulations proposées sont marquées en pointillé. Ici $p = 2$, $q = 3$ et $r = 6$.

6.5.3 Modélisation fine de l'arbre de cliques augmenté

Nous avons défini l'arbre de cliques augmenté pour des triangulations minimales. L'exemple de la figure 6.5 montre que cette notion peut et doit être étendue à toutes les

triangulations de graphes. Nous proposons d'étudier de manière complète cette notion afin d'en déduire de nouvelles stratégies de triangulation de graphes.

De manière assez simple, si on applique l'algorithme de réduction à notre exemple, il apparaît qu'une arête de la triangulation doit être ajoutée dans une *clique maximale*, en dehors de tout séparateur. Afin d'obtenir une triangulation minimale, on peut éliminer cette arête de cette clique maximale.

De cette modélisation, on peut en déduire des stratégies de triangulation de graphe qui maintiennent un ACT afin d'en obtenir une forme *propre*, en relation avec une triangulation minimale. On peut décrire l'algorithme du degré minimum dans ce cadre : en partant de la triangulation triviale ($F = E$) avec un seul sommet dans l'arbre de cliques, on cherche à éliminer en une seule étape une étoile maximale à cette triangulation, c'est-à-dire, on rend simplicial le sommet de degré minimal. Dans ce type de stratégie, on ne travaille que dans l'espace des graphes triangulés en différenciant deux types d'arêtes : celles qui sont dans le graphe cible, et celles qui appartiennent à la triangulation (minimale ou non).

Avec cette méthode, on peut alors orienter la recherche de la triangulation minimale suivant différents critères. Une voie de recherche consiste à étudier cette modélisation des graphes sous cette forme et le potentiel des techniques sous-jacentes.

6.5.4 Autres opérations sur l'arbre de cliques augmenté

Afin de calculer les polynômes chromatiques, Haggard a fortement utilisé la notion d'isomorphisme pour réduire le nombre de calculs effectifs. Une extension possible de notre stratégie pour le calcul du polynôme chromatique est de rechercher comment apparaissent les isomorphismes dans notre cadre.

Une première approche part de la constatation suivante : si on travaille « à un bout » de l'arbre de cliques, on devra avec une forte probabilité calculer plusieurs fois le polynôme chromatique d'un même graphe. Avec I. Briquel, nous avons commencé cette étude dans le cas des cycles. Nous avons montré que le nombre de graphes regardés (ou le nombre de récursions) peut être linéaire (contre quadratique avec la stratégie actuelle) en fonction de la taille de celui-ci. Cependant, il manque à cette étude préliminaire une modélisation qui permette de réaliser le calcul de manière efficace dans tous les cas.

Notre algorithme utilise une gestion simple de la mémoire. Un algorithme gérant l'isomorphisme doit enregistrer un certain nombre de polynômes chromatiques intermédiaires ; ce qui complique le stockage et la récupération. Ce problème pose donc plusieurs types de défis : algorithmique, structure de données et performance.

6.5.5 Extensions de l'arbre de cliques augmenté

Notre modélisation est basée sur l'ajout d'arêtes. Toutes les autres sont basées sur le calcul du polynôme chromatique dans les mineurs du graphe cible (au travers de l'équation 6.2). Il est donc important de voir comment notre modélisation s'applique pour les opérations classiques des mineurs.

En premier lieu, une stratégie qui utiliserait nos paradigmes travaillerait avec la notion de sous-triangulation, c'est-à-dire un ensemble d'arêtes à enlever pour que le graphe résultant soit triangulé. C'est un peu ce qui est décrit dans [88], sans stratégie de guidage

explicite des récursions. Le problème principal est de décrire la structure équivalente qui permet à l'algorithme de réduire le graphe initial en des graphes triangulés. De manière similaire à ce que l'on vient de décrire, on peut considérer l'arbre de cliques du graphe triangulé associé, et regarder les opérations que l'on peut effectuer avec les arêtes de la sous-triangulation (arêtes à éliminer lors de l'algorithme). Il ne peut s'agir alors que d'une version différente de l'ACT, où les arêtes de la sous-triangulation apparaissent entre les cliques maximales.

Avec cette nouvelle modélisation, les opérations de réduction ne sont pas très simples dans ce cas. Le cas de l'élimination d'arête s'avère simple et efficace, comme dans le cas de l'ajout d'une arête de la triangulation pour un ACT. Cependant, la contraction d'arêtes est généralement plus problématique. Un exemple caractéristique provient du cycle. Les sous-triangulations sont très simples et sont réduites à une seule arête comme le montre la figure 6.7(a). Le problème est que la contraction peut modifier tout l'arbre de cliques, et revient dans nombre de cas à recalculer une autre sous-triangulation minimale.

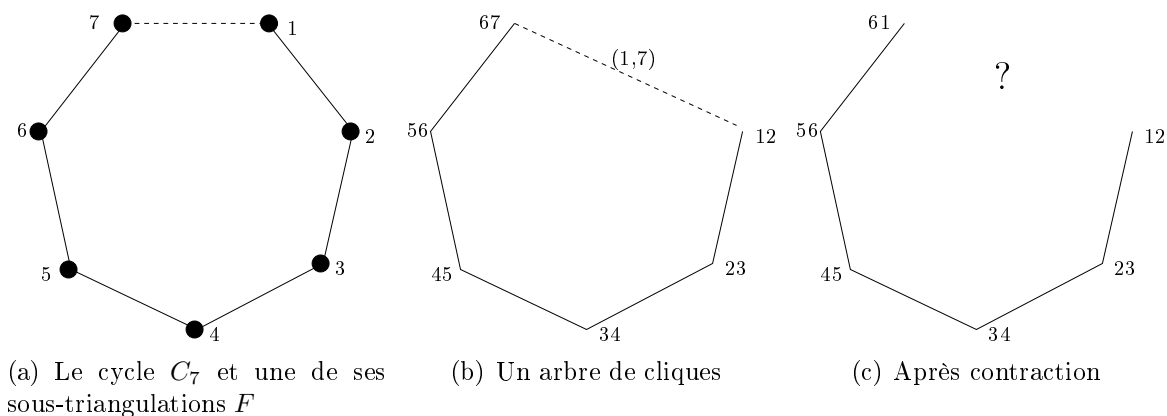


FIG. 6.7 – Fusion de sommets dans le cycle. Les triangulations sont notées en pointillés.

Cette voie semble donc compromise a priori. Cependant, si l'arête à contracter relie deux cliques voisines dans l'arbre de cliques, alors la modification devient réalisable sous une forme similaire à celle que nous avons présentée.

Sur cette thématique, on peut définir clairement deux axes :

1. Donner une version équivalente à notre description avec les sous-triangulations en définissant clairement les limites et l'impact de ces méthodes. On peut noter que cette méthode ne pourrait pas s'appliquer à la recherche du nombre chromatique.
2. Ces deux approches peuvent être combinées afin d'obtenir de nouvelles alternatives et améliorer les temps de calcul effectifs. On obtient alors une triangulation plus complexe, proche du problème général d'édition de graphes.

Chapitre 7

Conclusion et Perspectives

Tout au long de ce document, nous avons cherché à mettre en évidence notre démarche de recherche au travers d'un certain nombre d'exemples dans différents domaines. Nous avons recherché des modélisations pertinentes des réseaux ou des graphes proposés. On peut citer dans ce cadre :

- les empilements de graphes pour représenter le réseau optique POPS ;
- les arbres de Gomory-Hu pour les problèmes des flots multi-terminaux ;
- les arbres de cliques augmentés qui donnent une représentation des graphes adaptée aux opérations d'ajout d'arêtes et de fusion de sommets.

Par ailleurs, nous avons recherché des structures de particulières dans les graphes permettant de résoudre simplement le problème initial. Le meilleur exemple reste encore la couverture d'un graphe par les poulpes afin de réaliser l'échange total périodique.

Nous avons donné pour chaque problème un certain nombre de perspectives. Il ne s'agit pas ici de les reprendre en détail, et nous renvoyons donc le lecteur aux chapitres concernés. Le but est bien ici de replacer les différents thèmes en termes de priorités personnelles.

Communications Ce thème comporte un certain nombre d'aspects que nous avons abordé au cours du temps. L'évolution actuelle des thématiques d'interconnexion se tourne plutôt vers l'étude des systèmes pair à pair, ainsi que l'étude de l'utilisation de nouveaux protocoles comme BGP.

Nos centres d'intérêt dans ce domaine ont évolué durant les 10 dernières années. Nous privilégions maintenant les problématiques relatives à l'optimisation liées en général à la réservation de ressource. Le co-encadrement des thèses de M. Ben Dhaou et L. Gastal vont dans ce sens.

Arbres de Gomory-Hu Cette thématique rejoint des aspects plus fondamentaux de la compréhension des flots en général. Actuellement, nous avons traité un tout cohérent autour de la paramétrisation des arêtes du réseau de flots. Les nouveaux aspects que nous souhaitons mettre en avant dans cette thématique traitent des autres fonctions qui admettent ce type de modélisation, en particulier les flots multi-route.

Arbres de cliques augmentés Cette thématique est une des plus ouvertes dans ce que nous avons pu étudier. Elle constitue un axe prioritaire de notre recherche future. Cette partie nous a apporté une ouverture vers l’algorithmique basée sur les décompositions de graphes, qui est actuellement en plein essor.

Autres thématiques Au cours de nos recherches, nous avons essayé de diversifier les domaines d’application. La motivation du choix d’une thématique réside principalement dans la définition d’un nouveau problème algorithmique sur les graphes. Dans ce sens, nous participons à diverses modélisations de problèmes en bio-informatique. Cette thématique est très vaste, et c’est donc au travers de problèmes très ciblés que nous pouvons intervenir. Nous pouvons citer l’étude des ARNs et la recherche de motifs particuliers dans ces structures.

Bibliographie

- [1] Recherche Opérationnelle et COntstraints pour la COncption de réseau. Projet RNRT, 1999. http://www.telecom.gouv.fr/rnrt/rnrt/projets/pres_d107_ap99.htm.
- [2] DAta and VOice Integration over DWDM networks. IST Project, 2002. <http://david.com.dtu.dk/>.
- [3] Réseau Optique Multi-service exploitant des techniques Electro-optiques et Opto-électroniques. Projet RNRT exploratoire, 2002. http://www.telecom.gouv.fr/rnrt/rnrt/projets/res_02_52.htm.
- [4] Y.P. ANEJA, R. CHANDRASEKARAN et K.P.K. NAIR : Parametric min-cuts analysis in a network. *Discrete Applied Mathematics*, 127(3):679–689, mai 2003.
- [5] S. ARNBORG, D. G. CORNEIL et A. PROSKUROWSKI : Complexity of finding embeddings in a K-tree. *SIAM Journal on Algebraic and Discrete Methods*, 8:277–284, 1987.
- [6] B. ASPVALL et P. HEGGERNES : Finding minimum height elimination trees for interval graphs in polynomial time. *BIT*, 34(4):484–509, 1994.
- [7] D. BARTH : Une approche algorithmique du routage dans les réseaux de télécommunication. Mémoire d’habilitation n. 351 (in French), 1998. Université de Paris-Sud.
- [8] D. BARTH et P. BERTHOMÉ : Periodic gossiping in commuted networks. *Theory of Computing Systems*, 37(5):559–584, 2004.
- [9] D. BARTH, P. BERTHOMÉ et J. COHEN : The eulerian stretch of a network topology and the ending guarantee of a convergence routing. *Journal of Interconnection Networks*, 5(2):93–109, juin 2004.
- [10] D. BARTH, P. BERTHOMÉ et B. COHEN-BOULAKIA : Flots avec des fonctions de coût en escalier. In *ROADEF 2003*, pages 415–416, février 2003. In French.
- [11] D. BARTH, P. BERTHOMÉ, T. CZARCHOSKI, J-M. FOURNEAU, C. LAFOREST et S. VIAL : A mixed deflection and convergence routing algorithm : Design and performance. In *EuroPar’2002*, volume 2400 de *Lecture Notes in Computer Science*, pages 767–774, août 2002.
- [12] D. BARTH, P. BERTHOMÉ et M. DIALLO : Effects of capacities variations on maximum flows, minimum cuts and edge saturation. Rapport technique LRI-1395, LRI, octobre 2004. Available at <http://www.lri.fr/~berthome>.
- [13] D. BARTH, P. BERTHOMÉ et M. DIALLO : Detecting flows congesting a target network link. In P. FEOFILOFF, C.M.H. de FIGUEIREDO et Y. WAKABAYASHI, éditeurs : *GRACO’05*, volume 19 de *Electronic Notes in Discrete Mathematics*, pages 233–239, juin 2005.

- [14] D. BARTH, P. BERTHOMÉ, M. DIALLO et A. FERREIRA : Revisiting parametric multi-terminal problems : Maximum flows, minimum cuts and cut-tree computations. *Discrete Optimization*, 3(3):195–205, septembre 2006. <http://dx.doi.org/10.1016/j.disopt.2006.05.003>.
- [15] D. BARTH, P. BERTHOMÉ, J.-M. FOURNEAU et AL : Performance comparisons of Eulerian routing and deflexion routing in a 2D-mesh all optical network. *In 15TH European Simulation Multiconference*, pages 887–891, juin 2001.
- [16] D. BARTH, P. BERTHOMÉ et P. FRAGOPOULOU : The complexity of the maximal requests satisfaction problem in multipoint communication. *Parallel Processing Letters*, 15(1-2):209–222, 2005.
- [17] J. BEAUQUIER, S. KUTTEN et S. TIXEUIL : Self-stabilization in eulerian networks with cut-through constraints. Rapport technique LRI-1200, LRI-Université Paris-Sud, 1999.
- [18] R. BEIGEL et D. EPPSTEIN : 3-coloring in time $O(1.3289^n)$. *Journal of Algorithms*, 54(2):168–204, février 2005.
- [19] C. BERGE : *Graphes et Hypergraphes*. Dunod, Paris, 1977.
- [20] P.A. BERNSTEIN et N. GOODMAN : The power of natural semijoins. *SIAM Journal of Computing*, 10(4):751–771, février 1981.
- [21] P. BERTHOMÉ : *Contribution à l'Algorithmique des Architectures Parallèles : Des Réseaux Point-à-point aux Réseaux Optiques*. Thèse de doctorat, ENS-Lyon/UCBL-Lyon1, janvier 1995.
- [22] P. BERTHOMÉ, J. COHEN et A. FERREIRA : Embedding tori in partitioned optical passive star networks. *In D. KRIZANC et P. WIDMAYER, éditeurs : International Colloquium on Structural Information & Communication Complexity*, Proceedings in Informatics, pages 40–52. Carleton Scientific, 1997.
- [23] P. BERTHOMÉ, M. DIALLO et A. FERREIRA : Generalized parametric multi-terminal flows problem. *In H.L. BODLAENDER, éditeur : Graph Theoretical Concepts in Computer Science (WG) 2003*, volume 2880 de *Lecture Notes in Computer Science*, pages 71–80, octobre 2003.
- [24] P. BERTHOMÉ, Th. DUBOUX, T. HAGERUP, I. NEWMAN et A. SCHUSTER : Self-simulation for the passive optical star model. *In P. SPIRAKIS, éditeur : European Symposium on Algorithms*, numéro 979 de *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
- [25] P. BERTHOMÉ et A. FERREIRA : Improved embeddings in POPS networks through stack-graph models. *In Third International Workshop on Massively Parallel Processing using Optical Interconnections*, pages 130–136. IEEE CS Press, octobre 1996.
- [26] P. BERTHOMÉ et A. FERREIRA, éditeurs. *Optical Interconnections and Parallel Processing : Trends at the Interface*. Kluwer Academic Publishers, 1998.
- [27] P. BERTHOMÉ, L. GASTAL et A. LISSER : Robust routing in changing topologies. *In INOC 2005*, pages B2–603–608, 2005.
- [28] P. BERTHOMÉ, A. GUEYE, D. NOTT, F. QUESSETTE et A. TROUBNIKOFF : Managing QoS requests in the optical DAVID network. *In Photonics in Switching*, pages 83–85, septembre 2003.

- [29] P. BERTHOMÉ, S. LEBRESNE et K. NGUYỄN : Computation of chromatic polynomials using triangulation and clique trees. In D. KRATSCH, éditeur : *Graph Theoretical Concepts in Computer Science (WG) 2005*, volume 3787 de *Lecture Notes in Computer Science*, pages 262–373, 2005.
- [30] D. BERTSIMAS et M. SIM : The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- [31] S. BHATTACHARYA, D.H.C. DU, A. PAVAN, S.R. TONG, R. VETTER et K. WILLIAMS : A network architecture for distributed high performance heterogeneous computing. In *IEEE International Parallel Processing Symposium*, avril 1994.
- [32] D. BIENSTOCK et A. BLEY : Capacitated network design with multicast commodities. In *Proceedings of 8th International Conference on Telecommunication Systems*, Nashville, mars 2000.
- [33] H. BOURDIN, A. FERREIRA et K. MARCUS : A comparative study of one-to-many WDM lightwave interconnection networks for multiprocessors. In *Second International Workshop on Massively Parallel Processing using Optical Interconnections*, pages 257–253, San Antonio (USA), octobre 1995. IEEE Press.
- [34] J. C. BRASSIL et R. L. CRUZ : Bounds on maximum delay in networks with deflection routing. *IEEE Transactions on Parallel and Distributed Systems*, 6(7):724–732, 1995.
- [35] R. CHANDRASEKARAN, K.P.K. NAIR, Y.P. ANEJA et S.N. KABADI : Multi-terminal multipath flows : synthesis. *Discrete Applied Mathematics*, 143:182–193, 2004.
- [36] S-C. CHANG : Exact chromatic polynomials for toroidal chain of complete graphs. *Physica A*, 313:397–426, 2002.
- [37] B. COHEN-BOULAKIA : Calcul de bornes inférieures pour le dimensionnement réseau. Rapport de DEA Université Versailles-St-Quentin, septembre 2002. in French.
- [38] D. COUDERT : *Algorithmique et optimisation de réseaux de communications optiques*. Thèse de doctorat, Université de Nice Sophia-Antipolis (UNSA), décembre 2001.
- [39] J. CROWCROFT, M. HANDLEY et I. WAKERMAN : *Internetworking Multimedia*. Morgan Kaufmann Publishers, décembre 1999.
- [40] A. DATTA et S. SOUNDARALAKSHMI : Fast merging and sorting on a partitioned optical passive stars network. In R. KLEIN, H-W. SIX et L.M. WEGNER, éditeurs : *Computer Science in Perspective*, volume 2598 de *Lecture Notes in Computer Science*, pages 115–127, 2003.
- [41] M. Ben DHAOU : *Optimisation du placement de tâches dans les systèmes distribués et de l'allocation de ressources pour les communications multipoints*. Thèse de doctorat, Université Paris-Sud, Orsay, décembre 2003. In French.
- [42] M. DIALLO : Parametric multi-terminal networks flow analysis. Mémoire de D.E.A., Dpt of Mathematics, University of Kaiserslautern, Germany, octobre 2000.
- [43] M. DIALLO : *Réseaux de Flots : Flots Paramétrés et Tarification*. Thèse de doctorat, Université de Versailles, France, décembre 2003. In French. Available at <http://www.prism.uvsq.fr/~diallo>.

- [44] C. DIOT, W. DABBOUS et J. CROWCROFT : Multipoint communication : A survey of protocols, functions and mechanisms. *IEEE Journal on Selected Areas in Communications*, 15(3):277–290, avril 1997.
- [45] P.W. DOWD : Random access protocols for high-speed interprocessor communication based on an optical passive star topology. *Journal of Lightwave Technology*, 9(6):799–808, juin 1991.
- [46] S.E. ELMAGHRABY : On the relationship between the cut-tree and the fundamental cut-set of multi-terminal flow networks. *Journal of Franklin Institute*, octobre 1964.
- [47] S.E. ELMAGHRABY : Sensitivity analysis of multi-terminal network flows. *J. ORSA*, 12:680–688, 1964.
- [48] U. FEIGE : Observations on hot potato routing. In *ISTCS : 3rd Israeli Symposium on the Theory of Computing and Systems*, 1995.
- [49] U. FEIGE, M.T. HAJIAGHAYI et J.R. LEE : Improved approximation algorithms for minimum-weight vertex separators. In *ACM Symposium On Theory of Computing*, pages 563–572, 2005.
- [50] A. FERREIRA : Building a reference combinatoriel model for dynamic networks : Initial results in evolving graphs. Rapport technique RR–5041, INRIA, Sophia Antipolis, décembre 2003.
- [51] H. FLEISCHNER : *Eulerian graphs and related topics*, volume 45 de *Annals of Discrete Mathematics*. North-Holland, 1990.
- [52] L.R. FORD et D.R. FULKERSON : *Flows in Networks*. Princeton Univ. Press, Princeton, NJ, 1973.
- [53] J-M. FOURNEAU, D. BARTH, P. BERTHOMÉ, S. VIAL et C. LAFOREST : Performance evaluation of short-cut Eulerian routing. In *NGI 2005, 1st Conference on Next Generation Internet Networks*, Rome, avril 2005. Poster session.
- [54] P. GALINIER, M. HABIB et C. PAUL : Chordal graphs and their clique graph. In M. NAGL, éditeur : *Graph Theoretical Concepts in Computer Science (WG)*, volume 1017 de *Lecture Notes in Computer Science*, pages 358–371, Aachen, Germany, juin 1995. 21st International Workshop WG’95, Springer.
- [55] M.R. GAREY et D.S. JOHNSON : *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [56] L. GASTAL : Une nouvelle définition de la robustesse. Soumis, 2005.
- [57] A.V. GOLDBERG : Recent developments in maximum flow algorithms (invited lecture). In Stefan ARNBORG et Lars IVANSSON, éditeurs : *Scandinavian Workshop on Algorithm Theory*, volume 1432 de *Lecture Notes in Computer Science*, pages 1–10, 1998.
- [58] A.V. GOLDBERG et K. TSIOUTSIOLIKLIS : Cut tree algorithms : An experimental study. *Journal of Algorithms*, 38:51–83, 2001.
- [59] R. E. GOMORY et T. C. HU : Multi-terminal network flows. *SIAM Journal of Computing*, 9(4):551–570, décembre 1961.
- [60] G. GRAVENSTRETER et R.G. MELHEM : Realizing common communication patterns in partitioned optical passive stars (POPS) networks. *IEEE Transactions on Computers*, 47(9):998–1013, 1998.

- [61] G. GRAVENSTRETER, R.G. MELHEM, D.M. CHIARULLI, S.P. LEVITAN et J.P. TEZA : The partitioned optical passives stars (POPS) topology. *In IEEE International Parallel Processing Symposium*, pages 4–10, 1995.
- [62] P. GRAVEY, S. GOSSELIN, C. GUILLEMOT, D. CHIARONI, N. LE SAUZE, A. JOURDAN, E. DOTARO, D. BARTH, P. BERTHOMÉ, C. LAFOREST, S. VIAL, T. ATMACA, G. HÉBUTERNE, H. EL BIAZE, R. LAALAOUA, E. GANGLOFF et I. KOTULIAK : Multiservice optical network : main concepts and first achievements of the ROM program. *Journal of Lightwave Technology*, 19:23–31, janvier 2001.
- [63] D. GUSFIELD : Very simple methods for all pairs network flow analysis. *SIAM Journal of Computing*, 19:143–155, 1990.
- [64] T. HAGERUP, J. KATAJAINEN, N. NISHIMURA et P. RAGDE : Characterizations of k -terminal flow networks and computing network flows in partial k -trees. *In ACM Symposium On Discrete Algorithms*, pages 641–649, 1995.
- [65] G. HAGGARD : Using thresholds to compute chromatic polynomials. *Ars Combinatoria*, 58:85–95, 2001.
- [66] P. HEGGERNES : Minimal triangulations of graphs : A survey. *Discrete Mathematics*, 2006. Article in Press, available On-line at www.sciencedirect.com.
- [67] M.-C. HEYDEMANN, J. OPATRYNY et D. SOTTEAU : Embeddings of hypercubes and grids into de Bruijn graphs. *Journal of Parallel and Distributed Computing*, 23:104–111, 1994.
- [68] A. HU : Video-on-demand broadcasting protocols : a comprehensive study. *In IEEE INFOCOM 2001*, volume 1, pages 508–517, Anchorage, Alaska, avril 2001.
- [69] B. JACKSON : Zeros of chromatic and flow polynomials of graphs. *Journal of Geometry*, 76:95–109, 2003.
- [70] S.N. KABADI, R. CHANDRASEKARAN et K.P.K. NAIR : Existence of cut-trees and the realizability problem for multiroute flows. *In INOC 2005*, pages B2–373–377, 2005.
- [71] W. KISHIMOTO : A method for obtaining the maximum multi-route flows in a network. *IEEE Networks*, 27:279–291, 1996.
- [72] L. KOU, G. MARKOWSKY et L. BERMAN : A fast algorithm for Steiner trees. *Acta Informatica*, 15(2):141–145, 1981.
- [73] K. KRAUTER, R. BUYYA et M. MAHESWARAN : A taxonomy and survey of grid resource management systems for distributed computing. *Software - Practice & Experience*, 32(2):135–164, février 2002.
- [74] D. KRUMME et P. FRAGOPOULOU : Minimum eccentricity multicast trees. *Discrete Mathematics and Theoretical Computer Science*, 4:157–172, 2001.
- [75] C. LAFOREST : Quelques résultats algorithmique sur la construction de structures de communications. Mémoire d’habilitation à diriger des recherches, avril 2004. In French.
- [76] C. LAFOREST et S. VIAL : Short cut eulerian routing of datagrams in all optical point-to-point networks. *In International Parallel and Distributed Processing Symposium*, avril 2002.

- [77] B.D. MCKAY : Practical graph isomorphis. *Congressus Numerantium*, (30):45–87, 1981. 10th. Manitoba Conference on Numerical Mathematics and Computing (Winnipeg, 1980).
- [78] A. MEI et R. RIZZI : Mapping hypercube computations onto partitioned optical passive star networks. In T.M. PINKSTON et V.K. PRASANNA, éditeurs : *High Performance Computing - HiPC 2003, 10th International Conference, Hyderabad, India*, volume 2913 de *Lecture Notes in Computer Science*, pages 95–104, décembre 2003.
- [79] A. MEI et R. RIZZI : Routing permutations in partitioned optical passive stars networks. *Journal of Parallel and Distributed Computing*, 63(9):847–852, 2003.
- [80] A. NATANZON, R. SHAMIR et R. SHARAN : A polynomial approximation algorithm for the minimum fill-in problem. In ACM, éditeur : *ACM Symposium On Theory of Computing*, pages 41–47, New York, NY, USA, 1998. ACM Press.
- [81] Y. OFEK et M. YUNG : Principles for high speed network control : loss-less and deadlock-freeness, self-routing and a single buffer per link. In *ACM Symposium On Principles of Distributed Computing*, pages 161–175, 1990.
- [82] J-C. PICARD et M. QUEYRANNE : On the structure of all minimum cuts in a network and applications. *Mathematical Programming Study*, 13:8–16, 1980.
- [83] B. ROY : *Flexibilité et robustesse en ordonnancement*, chapitre 2 : A propos de robustesse en recherche opérationnelle et aide à la décision, pages 35–50. Hermes - Lavoisier, 2005. sous la direction de J.-C. Billaut, A. Moukrim, E. Sanlaville.
- [84] J. de RUMEUR : *Communication dans les Réseaux de Processeurs*. Masson, 1994. In French.
- [85] A. SCHRIJVER : *Combinatorial Optimization — Polyhedra and Efficiency, Volume A : Paths, Flows, Matchings*, volume 24 de *Algorithms and Combinatorics*. Springer-Verlag, Berlin-Heidelberg-New York-Hong Kong-London-Milan-Paris-Tokyo, 2003.
- [86] A. SCHRIJVER : *Combinatorial Optimization — Polyhedra and Efficiency, Volume C : Disjoint Paths, Hypergraphs*, volume 24 de *Algorithms and Combinatorics*. Springer-Verlag, Berlin-Heidelberg-New York-Hong Kong-London-Milan-Paris-Tokyo, 2003.
- [87] A. SCHUSTER : *Optical Interconnections and Parallel Processing : The Interface*, chapitre Bounds and analysis techniques for greedy hot-potato routing, pages 284–354. Kluwer Academic Publishers, 1998.
- [88] D.R SHIER et N. CHANDRASEKHARAN : Algorithms for computing the chromatic polynomial. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 4:213–222, 1988.
- [89] P.A. TUCKER, T.C. HU et M.T. SHING : Minimum cuts without path packing. Rapport technique CS 99-625, UCSD, juin 1999.
- [90] B. Bui XUAN, A. FERREIRA et A. JARRY : Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267–285, avril 2003.
- [91] M. YANNAKAKIS : Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, mars 1981.

- [92] B. YENER, T. BOULT et Y. OFEK : Hamiltonian decompositions of regular topology networks with convergence routing. Rapport technique CUCS-011094, Columbia University, Computer Science Department, 1994.

Annexe A

Collaborations

Cette annexe décrit les collaborations principales auxquelles j'ai pu participer au cours de ces dernières années, principalement dans le domaine des communications optiques.

A.1 Télécommunications

Dans le contexte des télécommunications, nous avons travaillé sur divers modèles au travers de plusieurs projets en collaboration avec le monde industriel et universitaire, avec des partenaires français et européens.

ROM (Routage Optique Multiservice) : ce projet RNRT était géré par Alcatel CIT avec plusieurs partenaires industriels (France Télécom R&D, Alcatel), et universitaires (INT, LRI) durant la période 1999-2001. Le but principal de ce projet était de montrer la faisabilité d'un réseau optique multi-service où l'optique intervient à tous les niveaux du routage. Il fallait aussi prévoir la gestion de la qualité de service. Les verrous technologiques à lever par ce projet étaient d'identifier les diverses classes de service, de définir un format de paquets compatible avec l'architecture proposée. Nous avons présenté le premier modèle du réseau dans [62]. C'est dans le cadre de ce projet que nous avons défini le routage eulérien dont quelques propriétés sont montrées au paragraphe 3.3.

Ce projet a connu une suite dans le cadre RNRT avec le projet ROM-EO [3], en regardant une faisabilité à moins long terme. En effet, l'option tout-optique n'est pas actuellement financièrement viable, l'utilisation d'interfaces Électronique/Optique s'avère une solution intermédiaire acceptable. Cependant, je n'ai que très superficiellement participé à ce dernier projet.

DAVID (DAta and Voice Integration over DWDM networks) [2] : ce projet européen du 5ème PCRD IST a regroupé une quinzaine de partenaires sous la direction scientifique de L. Ditmann de l'université technique du Danemark. Le but était là aussi de proposer un réseau optique travaillant dans le domaine multi-longueur d'onde, allant du réseau de type mondial (WAN) ou métropolitain (MAN). Le projet s'intéressait à tous les aspects, depuis l'électronique jusqu'à la gestion des messages. Dans ce contexte, j'ai encadré le stage de DEA d'A. Gueye sur des problèmes d'allocation de requêtes sur le réseau proposé [28].

A.2 Autres collaborations

ROCOCO (Recherche Opérationnelle et COntraintes pour la COnception de réseau) [1] : Ce projet RNRT était géré par Ilog en partenariat avec France Telecom R&D. Ce projet vise à développer des techniques logicielles intégrant l'efficacité de l'optimisation combinatoire et la flexibilité de la programmation par contraintes pour optimiser le dimensionnement et le re-dimensionnement de réseaux de télécommunication, d'entreprise en particulier. Dans ce contexte, j'ai encadré le stage de DEA de B. Cohen-Boulakia [10, 37] dont le but était de modéliser le coût de revient d'une ligne simple en fonction du débit demandé. Ces travaux ne sont pas présentés dans ce document.

BQR Fourmis Le travail sur les évolutions de contexte dans le monde des télécommunications a été initié au cours d'un BQR de l'université Paris-Sud en collaboration avec M. Sebag de l'équipe Inférence et Apprentissage du LRI, il a été prolongé dans le cadre de la thèse de L. Gastal en collaboration avec A. Lisser. Nous en présentons les premiers résultats au paragraphe 4.2.

Annexe B

Définitions générales et notations

Nous présentons dans cette annexe divers rappels de la théorie des graphes utiles pour la compréhension du document. Il ne s'agit pas de faire un cours, mais de poser les définitions. Toutes ces notions peuvent être trouvées dans plusieurs livres de référence comme [19, 85].

Un **graphe non orienté** $G = (V, E)$ est constitué de deux ensembles : un ensemble de **sommets** V , et un ensemble d'**arêtes** E . Chaque arête relie deux sommets qui sont alors dits voisins. Un graphe est dit simple si, pour chaque couple de sommets, il existe au plus une seule arête les reliant. Dans le cas contraire, on parlera de multi-graphe. Si l'arête (a, a) n'existe pas, alors le graphe est dit sans boucle.

Un **chemin** dans un graphe est une séquence (finie) de sommets telle que deux sommets successifs de la séquence sont reliés par une arête. Le chemin est dit **simple** s'il ne passe pas deux fois par le même sommet. La **longueur** d'un chemin est le nombre d'arêtes le composant. La **distance** entre deux sommets est la longueur du plus court chemin les reliant. Un graphe est dit **connexe** s'il existe pour toute paire de sommets un chemin les reliant.

Le **degré** d'un sommet est le nombre de voisins que celui-ci possède dans le graphe. On note $\delta(G)$ (ou simplement δ) le minimum des degrés du graphe et $\Delta(G)$ (ou Δ) le maximum des degrés.

Pour un graphe non orienté $G = (V, E)$ et $e = (a, b)$ une arête de G , on note $G \setminus e$ le graphe $(V, E \setminus e)$ (le graphe dans lequel e est éliminé) et G/e le graphe simple sans boucle (V', E') tel que $V' = V \setminus b$ et $E' = \{(\pi_a^b(x), \pi_a^b(y)) \mid (x, y) \in E\} \setminus (a, a)$, où π_a^b est la fonction de V dans V' qui correspond à l'identité sur V' et tel que $\pi_a^b(b) = a$. Le graphe G/e est le résultat de la contraction de G suivant e . Il revient simplement à identifier les deux extrémités de l'arête e et d'identifier les deux voisinages. Par analogie, si $e \notin E$, on définit $G + e = (V, E \cup \{e\})$ et $G/e = (G + e)/e$. Nous illustrons ces notions dans la figure B.1. Les deux premières opérations sont les opérations de **mineur** sur un graphe. Par extension, un graphe H est un mineur de G s'il peut être obtenu par une suite (finie) de contractions et de suppressions d'arêtes.

On dit que deux graphes G et H sont isomorphes, ce qu'on note $G \sim H$, s'il existe un renommage des sommets de G tel que les deux graphes soient identiques (mêmes sommets et mêmes arêtes).

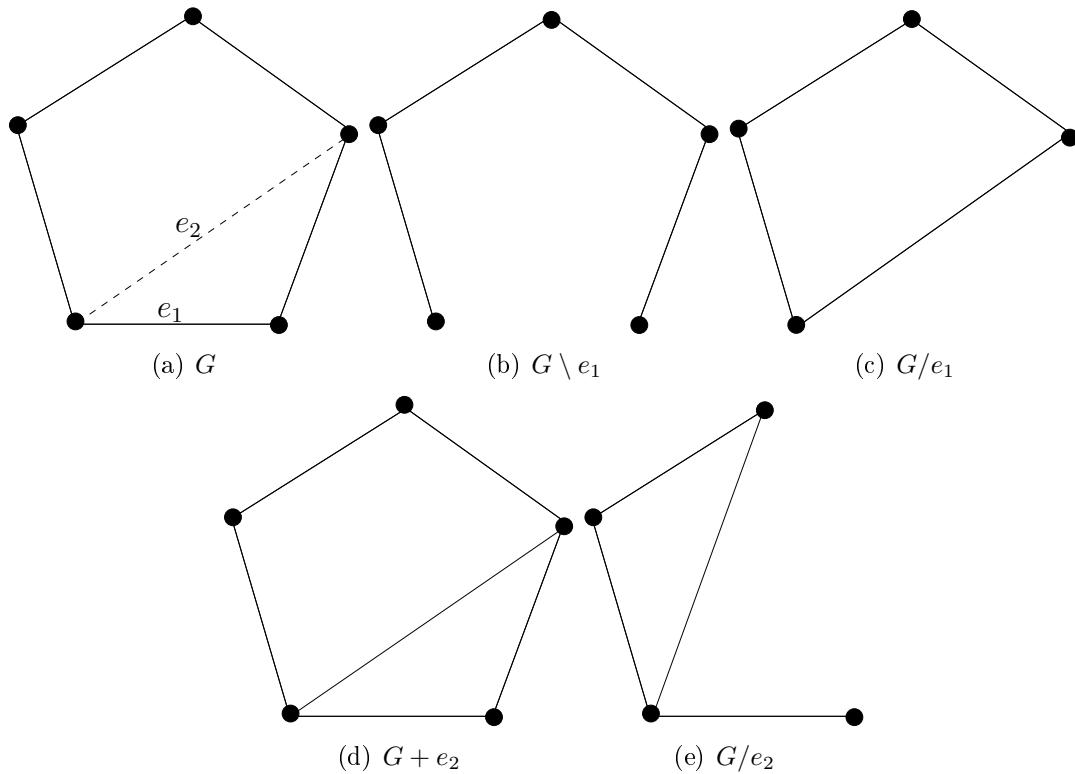


FIG. B.1 – Un graphe et ses transformés élémentaires. e_1 est une arête de G tandis que e_2 n'en est pas une.

Pour terminer, voici quelques exemples de graphes qui sont utilisés dans ce document. Un **arbre** à n sommets est un graphe connexe ayant $n - 1$ arêtes. Une **forêt** est une collection d'arbres. Un **anneau** est un chemin simple pour lequel les deux extrémités sont identiques. On note K_n le graphe complet à n sommets, c'est-à-dire quand tous les couples de sommets sont reliés par une arête. Dans le cas des graphes orientés, on note K_n^+ le graphe complet sans boucle, et K_n^{+*} quand toutes les boucles sont aussi présentes.