



HAL
open science

Solutions optimales des problèmes de recouvrement sous contraintes sur le degré des nœuds

Massinissa Merabet

► **To cite this version:**

Massinissa Merabet. Solutions optimales des problèmes de recouvrement sous contraintes sur le degré des nœuds. Algorithmes et structure de données [cs.DS]. Université Montpellier II - Sciences et Techniques du Languedoc, 2014. Français. NNT : 2014MON20138 . tel-01684758

HAL Id: tel-01684758

<https://theses.hal.science/tel-01684758>

Submitted on 15 Jan 2018

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.

THÈSE

Pour obtenir le grade de
Docteur

Délivré par l'Université Montpellier II

Préparée au sein de l'école doctorale **I2S***
Et de l'unité de recherche **LIRMM, UMR 5506**

Spécialité: **Informatique**

Présentée par **Merabet Massinissa**

**Solutions optimales des problèmes de
recouvrement sous contraintes sur le
degré des nœuds**

Soutenue le <05/12/2014> devant le jury composé de :

CRISTINA BAZGAN	Prof.	LAMSADE, U. Paris-Dauphine	Rapporteur
DAVID COUDERT	CR, HDR	CNRS, INRIA Sophia Antipolis	Rapporteur
JEAN-CLAUDE KÖNIG	Prof.	LIRMM, U. Montpellier II	Examineur
CHRISTIAN LAFOREST	Prof.	LIMOS, U. Blaise Pascal	Examineur
MIKLÓS MOLNÁR	Prof.	LIRMM, U. Montpellier II	Directeur
SYLVAIN DURAND	MCF	LIRMM, U. Montpellier III	Encadrant

Table des matières

Table des figures	5
Remerciements	7
Introduction	9
1 Notations et état de l'art	13
1.1 Introduction	13
1.2 Notations et définitions	13
1.2.1 Notations	13
1.2.2 Éléments de la théorie de la complexité	16
1.2.2.1 Problèmes d'optimisation	17
1.2.2.2 Classes de complexité	17
1.2.3 Éléments de la théorie de l'approximation	18
1.2.4 Éléments de la programmation linéaire	19
1.3 Domaine d'application : les réseaux optiques	19
1.3.1 Avantages de la transmission par fibre optique	20
1.3.2 Le multiplexage	21
1.3.3 Architecture des réseaux tout optique	21
1.3.4 Le routage multicast/broadcast tout optique	22
1.4 État de l'art	25
1.4.1 Arbre de recouvrement avec une borne supérieure sur le nombre de sommets de branchement	25
1.4.1.1 Définition et intérêt pratique	25
1.4.1.2 Complexité	26
1.4.1.3 Résolution exacte et approchée	28
1.4.1.3.a Utilisation de la programmation linéaire	28
1.4.1.3.b Utilisation d'heuristiques	32
1.4.2 Arbre de recouvrement tel que la somme des degrés des sommets de branchement est minimum	33

1.4.2.1	Définition et intérêt pratique	33
1.4.2.2	Résolution exacte et approchée	34
1.4.2.3	Relation entre le problème MBVST et le problème MDST	36
1.4.3	Arbre de recouvrement avec une borne supérieure sur le degré des sommets	38
1.4.3.1	Définition et intérêt pratique	38
1.4.3.2	Résolution exacte et approchée	39
1.5	Conclusion	41
2	Recouvrement total avec une borne supérieure sur le degré des sommets	43
2.1	Introduction	43
2.2	Arbre de recouvrement sous contrainte sur le degré des sommets	44
2.2.1	Une nouvelle formulation PLNE du problème DCMST	45
2.3	Hierarchie de recouvrement sous contrainte sur le degré des sommets	47
2.3.1	Propriétés des solutions optimales du problème DCMSH	50
2.3.2	Résolution exacte du problème DCMSH	54
2.3.2.1	Première étape : Construction d'un k-graphe	55
2.3.2.2	Deuxième étape : Formulation du PLNE	56
2.3.2.3	Troisième étape : Construction d'une hiérarchie optimale	59
2.3.3	Résultats expérimentaux	61
2.3.4	Simulations	61
2.3.5	Résultats des simulations	61
2.4	Résolution approchée du problème DCMSH	64
2.4.1	Recouvrement d'étoiles par les hiérarchies	65
2.4.2	Un algorithme d'approximation pour le problème DCMSH	67
2.4.2.1	La décomposition en étoiles de l'arbre de coût minimum	68
2.4.2.2	L'heuristique d'approximation du problème DCMSH	68
2.4.3	Discussion sur le ratio d'approximation	70
2.5	Conclusion	72
3	Recouvrement total avec une borne supérieure sur le nombre de sommets de branchement	73
3.1	Introduction	73
3.2	Arbre de recouvrement : Structure utilisée pour minimiser le nombre de sommets de branchement dans les réseaux optiques	74
3.2.1	Les problèmes MBVST et MDST	74
3.2.2	Relation entre le problème MBVST et le problème MDST	76
3.2.3	Arbre de recouvrement de graphe ayant un minimum de feuilles	79
3.3	Hierarchie de recouvrement : une alternative à la structure d'arbre pour minimiser le nombre de sommets de branchement dans les réseaux optiques	81
3.4	Propriétés des solutions optimales du problème BVCMSH	83
3.5	Résolution exacte du problème BVCMSH	85

3.5.1	Formulation ILP	86
3.5.2	Résultats expérimentaux	87
3.6	Résolution approchée du problème BVCMSH	89
3.6.1	Première étape : Construction d'un arbre de recouvrement de coût minimum	90
3.6.2	Deuxième étape : Contraction des chaînes qui composent le MST	90
3.6.3	Troisième étape : Décomposition en étoiles de l'arbre de recouvrement de coût minimum contracté T'	91
3.6.4	Quatrième étape : Recouvrement des étoiles par les hiérarchies	91
3.6.5	Cinquième étape : Connexion des hiérarchies de recouvrement des étoiles .	92
3.7	Conclusion	94
4	Routage multicast dans les réseaux WDM sans capacité de division	95
4.1	Introduction	95
4.2	Définition des problèmes	96
4.3	Formulation PLNE du problème MWRS	102
4.4	Résultats expérimentaux	106
4.4.1	Les paramètres des simulations	106
4.4.2	Les résultats des simulations	107
4.5	Conclusion	110
	Conclusion	111
	Bibliographie	115

Table des figures

1.1	Illustration d'une hiérarchie	16
1.2	Schéma permettant d'illustrer le découpage en temps entre les différentes connexions 21	
1.3	Schéma permettant d'illustrer le découpage en longueur d'onde entre les différentes connexions	22
1.4	Architecture d'un réseau tout optique WDM	23
1.5	Illustration de la preuve pour le MBV	26
1.6	Illustration de la preuve pour le MBV	27
1.7	Illustration de l'utilité de minimiser le degré des sommets de branchement	34
1.8	Illustration de la preuve pour le MDS	35
1.9	Exemple exhibé dans l'article [CGI09]	36
1.10	Instance contenant un arbre optimal pour le MDST qui n'est pas optimal pour le MBVST	37
2.1	Arbres de recouvrement de G respectant la contrainte sur le degré $B = 4$	44
2.2	Exemple illustrant la nécessité de garantir la connexité	47
2.3	Instance non satisfiable pour le problème DCMST quelle que soit la valeur de B .	48
2.4	Réduction du problème DCMSH au problème chemin hamiltonien ($B = 7$)	50
2.5	Construction du graphe G' à partir du graphe G ($B = 3$)	55
2.6	Schéma de résolution exacte du problème DCMSH	55
2.7	Construction du k-Graphe G_k à partir du graphe G	56
2.8	Construction de la hiérarchie optimale pour le problème DCMSH avec $B = 3$ à partir du graphe G_s	60
2.9	Arbre optimal versus hiérarchie optimale du réseau NSF	62
2.10	Arbre de recouvrement optimal versus hiérarchie de recouvrement optimale	64
2.11	Nombre d'instances en fonction de l'amélioration du coût par les hiérarchies	64
2.12	Hiérarchie de recouvrement d'étoile calculé avec l'heuristique proposée.	67
2.13	Décomposition en étoiles d'un arbre	70
2.14	Le graphe utilisé dans le théorème 2.4.3	71

3.1	Différence entre le problème MBVST et le problème MDST	75
3.2	Instance prouvant la proposition 3.2.2	77
3.3	Arbre de recouvrement $T1$ du graphe G	78
3.4	Arbre de recouvrement $T2$ du graphe G	79
3.5	Le problème ML n'est pas équivalent aux problèmes MBVST et MDST	80
3.6	Instance prouvant que les problèmes MBVST, MDST et MLST ne sont pas liés .	81
3.7	Réduction du problème BVCMST au problème chemin hamiltonien de coût minimum	83
3.8	Instance montrant que la borne sur le nombre de duplications des sommets ne dépend pas uniquement de k	85
3.9	Arbres de recouvrement optimaux versus hiérarchies de recouvrement optimales	88
3.10	Contraction des arêtes d'une chaîne	90
3.11	Recouvrement d'une étoile par une hiérarchie	91
3.12	Les sommets h_1 et h'_1 sont des sommets de branchement	92
3.13	Seul h_1 est un sommet de branchement	93
3.14	Seul h'_1 est un sommet de branchement	93
3.15	Aucun des deux sommets h_1 et h'_1 ne sont des sommets de branchement	93
4.1	Illustration d'une hiérarchie optique simple	98
4.2	Instance couverte par deux hiérarchies optiques	98
4.3	Relation entre le problème MWRS et le problème chemin hamiltonien orienté. . .	100
4.4	Transformation polynomiale du graphe G pour $k = 3$	101
4.5	Solution pour le problème MWRS en utilisant les structures HOS et HOE.	102
4.6	Recouvrement d'une étoile par une hiérarchie	102
4.7	Minimiser le nombre de longueurs d'onde nécessaire au routage peut augmenter considérablement le coût.	103
4.8	Dans une solution optimale pour le problème MWRS-HOS, la source n'a pas de prédécesseur.	104
4.9	Solutions optimales du problème MWRS-HOS versus solutions optimales du problème MWRS-HOE	109

Remerciements

Mes premiers remerciements vont à Christina Bazgan et à David Coudert pour avoir accepté d'être rapporteurs de cette thèse. Vos remarques pertinentes et vos conseils avisés m'ont permis d'améliorer le contenu de ce manuscrit. Je remercie tout aussi chaleureusement Jean-Claude König ainsi que Christian Laforest pour avoir accepté d'être membres de mon jury.

Un grand merci à Miklos Molnar, mon directeur de thèse, que j'ai eu la chance d'avoir comme enseignant en Master. Lors d'une présentation qui constituait l'évaluation de son module, j'avais préparé 60 diapos pour 15 minutes de temps de parole (questions comprises), je n'ai pas pu aller au-delà de l'introduction. Quinze jours plus tard, il était mon tuteur de stage. Six mois plus tard, il était mon directeur de thèse et trois ans plus tard il est et restera à mes yeux une référence scientifique et humaine. Miklos m'a poussé à donner le meilleur de moi-même en trouvant le juste équilibre entre liberté d'investigation et la pression nécessaire pour respecter les délais.

Cette thèse n'aurait pas été la réussite qu'elle est sans la rigueur et la minutie de Sylvain Durand, mon encadrant, qui était présent au quotidien. Sylvain s'est impliqué dans toutes les étapes techniques qu'on a pu rencontrer. Son expérience et ses conseils m'ont été d'une utilité inestimable. Ce fut un honneur de travailler avec Sylvain dont la gentillesse n'a d'égal que son rigorisme scientifique.

Je remercie tous les membres du LIRMM qui, en alliant fraternité, modestie et professionnalisme, créent une atmosphère scientifiquement stimulante et humainement intéressante. Un merci particulier à mes co-bureaux Remi, dont les connaissances en théorie de l'approximation et en complexité paramétrée impressionnent. Danh dont j'admire le courage d'être loin de sa femme et de leur bébé. Ceci donne une idée sur sa détermination et de son abnégation... Guillaume, toujours prêt à aider et à partager.

Les remerciements sont un exercice agréable mais compliqué, car on doit se poser une question à laquelle il est difficile de donner une réponse exhaustive : Grâce à qui me suis-je retrouvé en mesure de faire une thèse et grâce à qui l'ai-je réussi? Pour être bref, je vais remercier les personnes les plus chères à mon cœur, sans lesquelles je n'aurais pas été ce que je suis.

Mon père m'a toujours dit « Mon fils, tu es le meilleur. » même quand je ne l'étais pas réellement, il ne doutait pas de ma réussite. J'ai maintenant compris que les études - comme la vie d'ailleurs - ne sont pas un sprint mais une course de fond. L'important n'est pas d'être en tête à chaque virage mais d'être au rendez-vous à l'arrivée. Mon père a toujours eu plus d'ambitions pour moi que je n'en avais pour moi-même. Et si c'était ça avoir un père formidable ? Il est ma plus grande fierté et j'espère que grâce à ce modeste travail, j'ai réussi à lui procurer le dixième de la fierté que j'ai pour lui.

Dans un moment de très grand doute, comme les doctorants en rencontrent nécessairement sur leur chemin, ma mère a eu pour moi cette phrase qui m'a ému « Mon fils, ce n'est pas grave si tu ne réussis pas, nous t'aimons ». Quoi de plus réconfortant dans un moment difficile que d'avoir la permission d'échouer de la part de ceux que l'on aime. C'est sans doute grâce à cette phrase que j'ai réussi. Nous avons tous une maman formidable, la mienne l'est davantage encore !

Le plus lourd sacrifice qu'ont nécessité mes études est de n'avoir pas vu mon petit frère grandir. On s'appelle régulièrement et chaque appel est une bouffée d'oxygène. La pudeur de l'adolescent l'empêche de se livrer mais ses « allô » enthousiastes sonnent à mon oreille comme des « tu me manques ».

Je termine par Nejma, qui m'a accompagné au plus près durant cette thèse. Elle a su rendre mon quotidien agréable. Subir les difficultés, le stress, la fatigue et les doutes pour un diplôme n'est rien devant quelqu'un qui le subit pour rien, rien que pour moi. Les mots manquent à l'émotion. Nejma est la seule personne qui fait Science Politique et qui a des notions de théorie des graphes et de complexité, tellement ses lectures du manuscrit ont été nombreuses et méticuleuses. Cette fin de thèse n'est que le début d'autres péripéties, et je suis impatient de les découvrir avec elle. Les challenges de la vie ne me font plus peur depuis que je l'ai dans le cœur.

Introduction

Nous évoluons dans un environnement urbanisé autour de réseaux divers, de plus en plus denses et inter-connectés. L'acheminement de manière fiable et à moindre coût de l'information au sein d'un réseau a toujours été une préoccupation permanente de la recherche scientifique ; du développement technologique en général et de la recherche opérationnelle en particulier. L'acheminement de l'information se fait par une structure couvrant les terminaux et les liens du réseau utilisés à cet effet. Le travail que nous développons dans le cadre de cette thèse s'articule autour des problèmes de recherche de structure de recouvrement d'un réseau modélisé par un graphe sous contrainte sur le degré des sommets.

Étant la structure connexe permettant de couvrir les sommets en utilisant un minimum de liens, l'arbre est le plus utilisée pour ce type de problèmes. Cependant, cette seule caractéristique ne suffit pas ; il est également utile de minimiser l'ensemble des coûts des liens utilisés par l'arbre. On parlera alors d'arbre de recouvrement de coût minimum. La minimisation du coût de la structure de recouvrement est l'objectif majeur considéré dans cette thèse. Le recouvrement peut être total et ne couvrir l'ensemble des sommets du graphe ou partiel et couvrir alors qu'un sous ensemble des sommets de celui-ci. Chercher un arbre de coût minimum dans le cas du recouvrement total est polynomial. Il est cependant NP-difficile dans le cas du recouvrement partiel (arbre de Steiner).

Bon nombre de cas pratiques exigent une limitation du degré des sommets de l'arbre. Cette limitation peut avoir plusieurs formes. Elle peut, par exemple, exprimer la nécessité de borner le degré des sommets de l'arbre ou bien la capacité ou non de ces derniers à avoir plus de deux voisins. De manière générale, deux types de contraintes sur le degré des sommets peuvent être relevés. L'une dite homogène, identique à tous les sommets et l'autre hétérogène pouvant différer d'un sommet à un autre. Deux problèmes largement étudiés dans la littérature ont suscité notre intérêt :

1. Le problème Degree Constrained Minimum Spanning Tree (DCMST), qui consiste à chercher un arbre de recouvrement total d'un graphe tel que le degré de chacun de ses sommets a un degré inférieur ou égal à un entier k ;

2. Le problème Branch Vertices Constrained Minimum Spanning Tree (BVCMST), qui consiste à chercher un arbre de recouvrement total de coût minimum d'un graphe ayant au plus k sommets de branchement.

Ces deux problèmes sont NP-difficiles et n'admettent pas de ratio d'approximation avec un facteur constant.

Bien qu'en l'absence de contraintes sur le degré, l'arbre de recouvrement de coût minimum soit incontestablement la structure offrant un moindre coût, en présence de ceux-ci, il montre ses limites en matière de respectabilité des contraintes et de réduction du coût. Il existe cependant une autre structure capable de mieux répondre aux exigences et aux contraintes imposées. Notre action première est de mettre en évidence de manière concrète la plus-value apportée par cette nouvelle structure. Nous verrons, par la suite, que la hiérarchie est une structure nettement plus "flexible" que l'arbre. Il est alors nécessaire d'exploiter cette flexibilité pour minimiser le coût du recouvrement.

La structure d'arbre de recouvrement suscite un intérêt particulier dans le domaine des réseaux. L'introduction du broadcast/multicast par Deering dans les années 90 a considérablement augmenté son impact. Deering avait constaté que l'envoi d'un seul message de la source à chaque sommet consommait beaucoup de ressources (bande passante). Il a alors proposé l'envoi par la source d'un unique message, puis de le dupliquer au cours de son acheminement [Dee92]. Dans bon nombre de cas pratiques, borner le degré de chaque sommet de l'arbre de recouvrement de coût minimum est essentiel. La diffusion partielle ou totale dans les réseaux de communication classiques se fait grâce à la duplication de chaque message arrivant à un sommet par le nombre de ses successeurs. Afin de réaliser le routage en un temps acceptable ou d'augmenter la vitesse de celui-ci, il est souhaitable de limiter le nombre de duplications du message routé [ZOK93]. La robustesse du routage dépend également du degré des sommets. En effet, si un sommet de l'arbre a un degré élevé, alors en cas de panne, un grand nombre de sommets se retrouveraient isolés et il serait nécessaire de les relier rapidement à d'autres sommets de l'arbre [DH68].

Le routage broadcast/multicast dans les réseaux optiques constitue l'essentiel de notre préoccupation. La transmission des messages dans ce type de réseaux se fait grâce à la technologie du multiplexage en longueur d'onde communément appelée WDM (Wavelength Division Multiplexing). Cette technologie consiste en la division du spectre optique en plusieurs sous-canaux où chacun est associé à une longueur d'onde spécifique [Gre92]. Il est ainsi possible de faire transiter, dans une même fibre, plusieurs longueurs d'onde en parallèle ou encore ce que l'on peut considérer comme étant plusieurs couleurs en même temps.

Dans les réseaux tout optique, les fonctions de commutation et de routage sont fournies par les brasseurs optiques OXC (optical cross-connect) et permettent la mise en œuvre de commu-

nications de bout en bout entre les nœuds d'accès. Grâce au démultiplexage du signal optique entrant, un brasseur optique OXC peut commuter chacune des longueurs d'onde d'un port d'entrée vers un port de sortie quelconque. Certains OXC particuliers, appelés OXC-MC, peuvent également diviser une longueur d'onde entrante vers plusieurs ports de sortie à la fois grâce à un coupleur optique, afin d'offrir un service multicast.

Dans les réseaux équipés de switch OXC-MC, le routage multicast/broadcast se fait grâce à des arbres de recouvrements partiels, chacun prenant en compte une seule longueur d'onde. Si les switches OXC-MC sont suffisamment disponibles et bien positionnés dans le réseau, alors un seul arbre de recouvrement (une seule longueur d'onde) peut être suffisant. Bien que la contrainte principale imposée par la technologie WDM soit que deux canaux optiques de même longueur d'onde ne peuvent pas emprunter une même fibre, la contrainte sur le degré des sommets de l'arbre de recouvrement est également justifiée. Effectivement, les sommets munis de coupleurs optiques sont capables de diviser le spectre lumineux arrivant, mais cette capacité est limitée. De plus, la division du spectre lumineux entraîne une perte de puissance.

La technologie développée aujourd'hui dans les réseaux utilise la structure de l'arbre de recouvrement pour le routage. Cependant, un certain nombre de ses limites structurelles vient se greffer à l'environnement global de fonctionnement qui tend à devenir rigide. En effet, l'arbre de recouvrement contient une et une seule fois les sommets du graphe. En matière de routage, ceci est apparenté à un unique passage par chaque sommet. Or, dans certains cas, il est possible et même nécessaire d'avoir un passage multiple par un même sommet. Le routage dans les réseaux optiques multifibres, dans lesquels deux sommets peuvent être reliés par plusieurs fibres optiques, permet ainsi un passage multiple par un sommet avec une même longueur d'onde.

La capacité limitée des switches à diviser la lumière a toujours été considérée comme étant une valeur globale à ne pas dépasser. Dans le cas d'un passage multiple, cette valeur est décrétementée à chaque passage. Cependant, la contrainte sur le degré n'exprime pas toujours le fait qu'un sommet a un "budget" global afin de connecter ses sommets voisins (cette approche de budget global est considérée dans [RMR⁺01]). La contrainte sur le degré peut aussi exprimer une action qui est réalisée ou un budget qui est réhabilité à chaque visite du sommet. Il est donc nécessaire de trouver une autre structure qui prenne en considération le passage multiple et la réhabilitation de la contrainte sur le degré à chacun des passages. Cette nouvelle approche parfaitement adaptée au recouvrement de graphes a été introduite et développée par M. Molnar [Mol08]. Notre travail de thèse repose sur ses fondements et traite de la hiérarchie de recouvrement comme une structure flexible permettant d'optimiser une valeur telle que la minimisation du coût du routage. Notre contribution s'articule donc autour de cette nouvelle structure. Nous avons reconsidéré les problèmes classiques de recherche de structure (arbre) de recouvrement sous contrainte sur le degré des sommets. Nous nous appuyons sur les hiérarchies de recouvrement afin de répondre à ces problèmes de manière plus efficace.

Nous introduisons dans cette optique deux nouveaux problèmes :

1. Le problème Degree Constrained Minimum Spanning Hierarchy (DCMSH), qui consiste à chercher une hiérarchie de recouvrement de coût minimum d'un graphe ayant au plus k sommets de branchement.
2. Le problème Branch Vertices Constrained Minimum Spanning Hierarchy (BVCMSH), qui consiste à chercher une hiérarchie de recouvrement de coût minimum d'un graphe tel que le degré de chaque sommet est inférieur ou égal à un entier k .

Nous effectuons une analyse large de ces deux problèmes. La complexité et la résolution exacte et approchée sont ainsi abordées. Une partie expérimentation effectuée sur des graphes aléatoires complète cette analyse théorique et offre une vision générale mais précise sur la contribution des hiérarchies dans l'amélioration du coût du routage.

Pour illustrer l'applicabilité de cette structure, nous étudions un troisième problème qui prend davantage en compte la réalité des réseaux optiques. Ce problème, traitant du routage multicast dans les réseaux optiques dépourvus de switch OXC-MC, consiste en un calcul d'une structure de recouvrement de coût minimum ayant la source comme racine et couvrant toutes les destinations. Cette structure doit cependant respecter non seulement la contrainte sur le degré mais également celle sur la distinction des longueurs d'onde [ZP05].

Nous résolvons ce nouveau problème NP-difficile en utilisant la structure de hiérarchie optique élémentaire et la structure de hiérarchie optique simple.

Cette thèse est constituée de quatre chapitres. Le premier donne les pré-requis et les notations nécessaires à la bonne lecture des chapitres suivants ainsi que l'état de l'art des problèmes de recouvrement sous contrainte sur le degré des sommets. Le deuxième est consacré à l'étude du problème de recouvrement total avec une borne supérieure homogène sur le degré des sommets. Le troisième est dédié à l'étude du problème de recouvrement total avec une borne supérieure sur le nombre de sommets de branchement. Enfin, le dernier chapitre est consacré à l'étude du problème du routage multicast dans les réseaux WDM sans capacité de division.

Chapitre 1

Notations et état de l'art

1.1 Introduction

Dans ce premier chapitre, nous donnons les notions élémentaires requises pour une bonne lecture de ce document. Nous commençons par préciser les notations et les définitions qui accompagneront nos travaux tout au long des chapitres suivants. Nous présentons également les éléments de la théorie de la complexité et de l'approximation qui constituent des outils d'analyse efficaces des problèmes traités dans cette thèse. La deuxième partie de ce chapitre est consacrée à la présentation de l'essentiel des travaux antérieurs sur les problèmes de recouvrement sous contrainte sur le degré des sommets. Cette étude de l'état de l'art illustre l'évolution du domaine, permettant ainsi de situer les travaux menés durant cette thèse et d'estimer notre valeur ajoutée à l'ensemble de l'effort scientifique précédemment fourni.

1.2 Notations et définitions

Dans cette section, nous introduisons les notations préliminaires nécessaires à la bonne compréhension des chapitres suivants. Nous rappelons également des définitions issues de la théorie des graphes et présentons quelques concepts fondamentaux de l'optimisation combinatoire et de l'approximation.

1.2.1 Notations

On note $G = (V_G, E_G)$ un graphe pouvant être orienté ou non orienté. L'ensemble V_G est un ensemble fini non vide d'éléments, les éléments de V_G sont appelés **sommets**. E_G est un ensemble tel que $E_G \subseteq V_G \times V_G$, les éléments de E_G sont appelés **arcs** (notés $e = (u, v)$) dans le cas orienté et **arêtes** (notées $e = \{u, v\}$) dans le cas non-orienté. On note $n = |V_G|$ le nombre de sommets (la taille du graphe) et $m = |E_G|$ le nombre d'arêtes du graphe G . Si une fonction $C : E_G \rightarrow \mathbb{R}_*^+$ associe un coût $C_G(e)$ à chaque arête $e \in E_G$ alors le graphe G est dit arêtes pondérées. Si une fonction $C' : V_G \rightarrow \mathbb{R}_*^+$ associe un coût $C'_G(v)$ à chaque sommet $v \in V_G$ alors

le graphe est dit sommets pondérés.

Degré et voisinage Un sommet $v \in V_G$ est voisin du sommet $u \in V_G$ dans G si l'arête $\{v, u\} \in E_G$. On note $\lambda_G(\mathbf{v}) \subseteq V_G$ l'ensemble des voisins de v dans G . On appelle degré d'un sommet v le cardinal de l'ensemble de ses voisins et le note $\mathbf{d}_G(\mathbf{v}) = |\lambda_G(v)|$. Plus généralement, soit un ensemble de sommets $X \subseteq V_G$, $\mathbf{d}_G(\mathbf{X}) = \sum_{v \in X} d_G(v)$. On note $\mathbf{\Delta}(G) = \max\{d_G(v) \mid v \in V_G\}$ le plus grand degré dans G et par $\mathbf{\delta}(G) = \min\{d_G(v) \mid v \in V_G\}$ le plus petit degré dans G . On note $\mathbf{V}_d(G)$ l'ensemble des sommets de degré d dans G . Un graphe G est **d-régulier** si tous les sommets de G ont un degré égal à d ($V_d(G) = V(G)$). Un graphe 3-régulier est aussi appelé **graphe cubique**. Si $d_G(v) = 1$ alors le sommet v est appelé **feuille** et si ce sommet a un degré $d_G(v) > 1$ alors il est dit **sommet interne**. On note $\mathbf{F}(G)$ l'ensemble des feuilles de G et $\mathbf{IN}(G)$ l'ensemble des sommets internes. On a par définition $F(G) \cup IN(G) = V_G$. On note $\mathbf{L}(G) = |F(G)|$

Chemins et parcours Un chemin de u à v dans G , noté $\mathbf{CH}_G(\mathbf{u}, \mathbf{v})$, est une séquence de sommets $\langle v_0, \dots, v_k \rangle$ avec $v_0 = u$ et $v_k = v$ et telle que $\forall i, 0 \leq i < k, \{v_i, v_{i+1}\} \in E_G$. Le chemin $\mathbf{CH}_G(u, v) = \langle v_0, \dots, v_k \rangle$ est dit de **longueur** $k - 1$. Les sommets u et v sont les **extrémités** de $\mathbf{CH}_G(u, v)$. On note par $\ell(G)$ la longueur du chemin le plus long dans G . Un chemin est élémentaire si chacun des sommets de la séquence le composant y est présent une seule fois, il est non-élémentaire sinon. Un chemin est dit **hamiltonien** s'il contient tous les sommets du graphe et qu'il est élémentaire. Un graphe contenant un chemin hamiltonien est appelé graphe hamiltonien. Un parcours $\mathbf{P}(G)$ d'un graphe G est un chemin qui contient tous les sommets du graphe une ou plusieurs fois et qui peut passer plusieurs fois par chaque arête. Si un parcours passe au plus une fois par chaque arête alors il est élémentaire. Si, par contre, il passe plus d'une fois par chacune des arêtes de G alors il est dit non-élémentaire. S'il passe une et une seule fois par chaque arête de G , le **parcours est dit eulérien**. Un graphe ayant un tel parcours est un **graphe eulérien**. Un graphe est **connexe** s'il existe un chemin entre chaque paire de ses sommets.

Cycles Un cycle dans G est un chemin $\mathbf{CH}_G(u, v)$ tel que $v = u$. Si un cycle contient tous les sommets du graphe, alors ce cycle est dit hamiltonien. Un graphe sans cycle est dit acyclique.

Sous-graphes Un sous graphe G' de G est un graphe tel que $V_{G'} \subseteq V_G$ et $E_{G'} \subseteq E_G$. Si $V_{G'} \subset V_G$ et $E_{G'} = \{\{u, v\} \in E_G \mid u \in V_{G'} \text{ et } v \in V_{G'}\}$ alors G' est un **sous-graphe induit** de G .

Arbres Un arbre $T = (V_T, E_T)$ est un graphe connexe acyclique. Si $V_T = V_G$ et $E_T \subseteq E_G$ alors T est dit **arbre de recouvrement** de G .

Sommets de branchement Un sommet de branchement est un sommet de degré strictement supérieur à deux ($d_G(v) > 2$). On appelle $\mathbf{SB}(G)$ l'ensemble des sommets de branchement du graphe G . On note $\mathbf{s}(G) = |\mathbf{SB}(G)|$ le nombre de sommets de branchement de G . On note $\mathbf{q}(G)$ la somme des degrés des sommets de branchement de G ($q(G) = \sum_{v \in \mathbf{SB}(G)} d_G(v)$). On dénote par $\mathbf{s}^*(G)$ le plus petit nombre de sommets de branchement que peut avoir un arbre de recouvrement de G . On note par $\mathbf{q}^*(G)$ la somme minimum des degrés des sommets de branchement que peut avoir un arbre de recouvrement de G . Enfin, nous notons par $\mathbf{L}^*(G)$ le nombre de feuilles minimum que peut avoir un arbre de recouvrement de G .

Araignées Une araignée est un arbre de recouvrement de celui-ci ayant au plus un sommet de branchement. Si, de plus, la distance entre toute paire de feuilles est au plus égale à deux, alors on parlera **d'étoile**.

Sommets indépendants et graphes bipartis Deux sommets $(v, u) \in V_G^2$ sont dits indépendants dans G s'il n'existe pas d'arête $\{v, u\} \in E_G$ les reliant. Un ensemble de sommets est dit indépendant si chaque paire de sommets de l'ensemble est indépendante. On note par $\delta_k(G)$ la somme minimum des degrés de k sommets indépendants dans G . On note $\delta_k^u(G) = \min_{I \in \mathcal{I}_k} \sum_{v \in I} d_G(v)$ tel que I est un ensemble de sommets de G de taille k et $I \cup \{u\}$ est indépendant. Notons que $\delta_1^u(G)$ est simplement le degré minimum des sommets non-adjacents à u . Un graphe $G = (V_G, E_G)$ dont l'ensemble de sommets peut être partitionné en deux ensembles $V1_G$ et $V2_G$, tel que $V1_G = V2_G \cup V1_G \cap V2_G = \emptyset$, chacun indépendant est appelé graphe biparti. On note $G = (V1_G, V2_G, E_G)$ un tel graphe. Remarquons que tout arbre est un graphe biparti.

Graphes complets Un graphe $G = (V_G, E_G)$ est dit complet si pour tout $(u, v) \in V_G^2$, $\{u, v\} \in E_G$. On note un tel graphe K_n , où n représente sa taille. Un graphe biparti $G = (V1_G, V2_G, E_G)$ est dit complet si $\forall u \in V1_G, \forall v \in V2_G, \{u, v\} \in E_G$. Un graphe biparti complet est noté K_{n_1, n_2} , où n_1 et n_2 représentent respectivement les cardinaux de ses deux ensembles de sommets.

Homomorphismes et hiérarchies de recouvrement Un parcours non-élémentaire d'un graphe G n'est pas un sous-graphe car il existe au moins une arête de G qui s'y trouve plus d'une fois. En appliquant cette remarque aux arbres de recouvrement, on peut envisager une nouvelle structure qui serait dans le cas élémentaire un arbre de recouvrement classique et dans le cas non-élémentaire un arbre qui contiendrait plusieurs fois les sommets et les arêtes du graphe couvert. Dans le cas non-élémentaire, cette structure n'est pas un sous-graphe du graphe couvert. Cette structure, appelée hiérarchie et introduite dans [Mol08], est un homomorphisme d'un arbre dans le graphe couvert. En effet, un homomorphisme est une application $h : W \rightarrow V$ entre deux graphes $Q = (W_Q, F_Q)$ et $G = (V_G, E_G)$ qui associe un sommet de V_G à chaque sommet de W_Q en préservant l'adjacence. Le triplet (Q, h, G) définit une **structure de recouvrement** dans G . Si Q est un arbre alors le triplet définit une **hiérarchie** dans G . On note une telle

hiérarchie $H = (Q_H, h_H, G)$ où Q_H est appelé arbre hiérarchique de H .

Une hiérarchie est caractérisée par son arbre hiérarchique. Pour simplifier, on pose $V_H = V_{Q_H}$ et $E_H = E_{Q_H}$. Un arbre hiérarchique Q_H contient une ou plusieurs fois chaque sommet $v \in G$. Si un sommet $v \in V_G$ est présent une seule fois dans Q_H alors on dira que ce sommet est dupliqué une seule fois dans Q_H et qu'il existe une seule occurrence de v dans Q_H . Si ce même sommet est présent n fois, alors on dira qu'il est dupliqué n fois dans Q_H et qu'il existe n occurrences de v dans Q_H . Il est utile d'étiqueter les différentes duplications des sommets de G dans Q_H . On note v^i la duplication i du sommet $v \in V_G$ dans Q_H (Figure 1.1). Le degré $d_H(v^i)$ d'un sommet v^i dans la hiérarchie H est égal au nombre de voisins de v^i dans l'arbre hiérarchique Q_H . Chaque duplication d'un sommet de G a donc un degré distinct dans H .

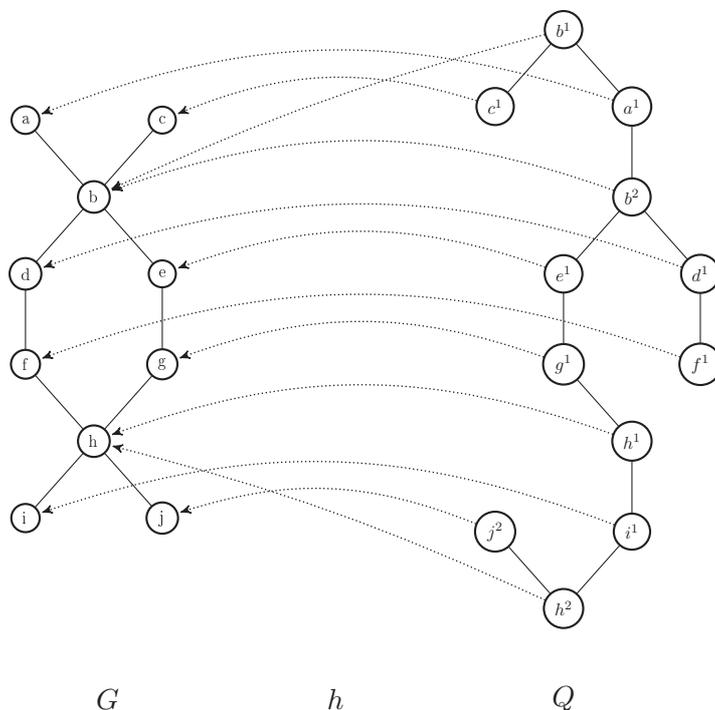


Figure 1.1 – Illustration d'une hiérarchie

1.2.2 Éléments de la théorie de la complexité

Il existe deux types de problèmes algorithmiques. Les problèmes de décision, qui consistent à répondre par l'affirmation ou par la négation à une question donnée. Le problème est dit décidable si la réponse peut être fournie par un algorithme en un nombre fini d'étapes. Les problèmes d'existence quant à eux concernent la recherche d'un élément en particulier, si celui-ci existe. Le problème est calculable si cet élément peut être fourni par un algorithme. Nous nous intéressons aux problèmes d'existence. Parmi ces problèmes, nous étudions exclusivement certains problèmes d'optimisation qui consistent à trouver la meilleure solution parmi l'ensemble des solutions réalisables (solutions qui satisfont des contraintes établies dans le problème). Cet

ensemble peut être fini mais peut compter un très grand nombre d'éléments. La qualité d'une solution est estimée grâce à une fonction objective, la meilleure solution est celle dont la valeur correspond le plus à l'objectif recherché (généralement maximiser ou minimiser une quantité).

1.2.2.1 Problèmes d'optimisation

Les problèmes d'optimisation dans leur forme générale peuvent être définis comme suit :

Définition 1.2.1. *Un problème d'optimisation \mathcal{P} est donné par le quadruplet $(I_{\mathcal{P}}, SOL_{\mathcal{P}}, m_{\mathcal{P}}, But_{\mathcal{P}})$, où :*

1. $I_{\mathcal{P}}$ est l'ensemble des instances de \mathcal{P} ;
2. $SOL_{\mathcal{P}}$ est la fonction qui associe à chaque instance $x \in I_{\mathcal{P}}$ l'ensemble $SOL_{\mathcal{P}}(x)$ des solutions réalisables de x ;
3. $m_{\mathcal{P}}$ est la fonction objective, définie pour les paires (x, y) telle que $x \in I_{\mathcal{P}}$ et $y \in SOL_{\mathcal{P}}(x)$. Pour chaque paire (x, y) , $m_{\mathcal{P}}(x, y)$ détermine une valeur dans \mathbb{R} qui est la valeur de la solution réalisable y .
4. $But_{\mathcal{P}} \in \{MIN, MAX\}$ spécifie si \mathcal{P} est un problème de minimisation ou de maximisation.

1.2.2.2 Classes de complexité

Nous présentons les classes de complexité dans lesquelles nos problèmes de décision et d'existence se situent. Pour plus d'informations sur la théorie de la complexité, nous renvoyons le lecteur vers l'ouvrage de Garey et Johnson [GJ79].

Définition 1.2.2. *Un problème est dans la classe de complexité P s'il peut être résolu par un algorithme déterministe en temps polynomial par rapport à la taille de l'instance. On qualifie alors le problème de polynomial, c'est un problème de complexité $O(n^k)$, tel que k est une constante.*

Définition 1.2.3. *La classe NP des problèmes Non-déterministes Polynomiaux réunit les problèmes de décision qui peuvent être décidés sur une machine non déterministe en temps polynomial. Plus simplement, ce sont les problèmes qui peuvent être résolus en énumérant l'ensemble des solutions possibles et en les testant à l'aide d'un algorithme polynomial. Le nombre de solutions peut néanmoins être exponentiellement grand.*

On a l'inclusion $P \subseteq NP$, mais on ne sait pas si cette inclusion est stricte, ou si $P = NP$. Une grande part de la théorie de la complexité s'est construite sous l'hypothèse que $P \neq NP$.

Définition 1.2.4. *La classe NP -complet est une sous-classe de NP constituée des problèmes de décision qui sont les plus difficiles de NP . Un problème de décision \mathcal{P} est NP -complet si et seulement s'il vérifie ces deux conditions :*

1. $\mathcal{P} \in NP$
2. $\forall \mathcal{P}' \in NP\text{-complet}, \mathcal{P}'$ se réduit à \mathcal{P} par un algorithme polynomial.

Un problème qui vérifie la condition 2 (sans pour autant la condition 1) est appelé *NP*-difficile. On peut donc dire qu'un problème est *NP*-complet si et seulement s'il appartient à *NP* et est *NP*-difficile. Ainsi, pour démontrer qu'un problème \mathcal{P} est *NP*-complet, il suffit de démontrer les deux propriétés suivantes :

1. $\mathcal{P} \in NP$,
2. Un problème quelconque \mathcal{P}' *NP*-complet se réduit à \mathcal{P} par un algorithme polynomial.

1.2.3 Éléments de la théorie de l'approximation

Nous résumons ici le concept d'algorithme d'approximation et de ratio d'approximation. Étant donné une instance x , on note $SOL_{\mathcal{P}}^*(x)$ l'ensemble des solutions optimales de x . C'est l'ensemble des solutions dont la valeur est optimale (minimum, si $But_{\mathcal{P}} = MIN$ ou maximum si $But_{\mathcal{P}} = MAX$). Plus formellement, pour chaque $y^*(x)$ tel que $y^*(x) \in SOL_{\mathcal{P}}^*(x)$ on a : $m_{\mathcal{P}}(x, y^*(x)) = But_{\mathcal{P}} \{v : v = m_{\mathcal{P}}(x, z) \wedge z \in SOL_{\mathcal{P}}(x)\}$. On note $m_{\mathcal{P}}^*(x)$ la valeur optimale de la fonction objective $y^*(x)$ de x .

Il est possible de comparer n'importe quelle solution réalisable $y \in SOL_{\mathcal{P}}(x)$ à la solution optimale et d'exprimer « l'éloignement » de cette solution de la solution optimale. Pour cela il suffit calculer le ratio de performance de y .

Définition 1.2.5. *Étant donné un problème d'optimisation \mathcal{P} , pour n'importe quelle instance x de \mathcal{P} et solution réalisable y de x , le ratio de performance de y est défini comme suit :*

$$R(x, y) = \frac{m_{\mathcal{P}}(x, y)}{m_{\mathcal{P}}^*(x)}$$

pour un problème de minimisation, et

$$R(x, y) = \frac{m_{\mathcal{P}}^*(x)}{m_{\mathcal{P}}(x, y)}$$

pour un problème de maximisation.

On peut désormais définir le ratio d'approximation comme suit :

Définition 1.2.6. *Étant donné un problème d'optimisation $\mathcal{P} = (I_{\mathcal{P}}, SOL_{\mathcal{P}}, m_{\mathcal{P}}, But_{\mathcal{P}})$ et un algorithme $\mathcal{A}_{\mathcal{P}}$ pour \mathcal{P} qui retourne pour une instance $x \in I_{\mathcal{P}}$ une solution réalisable $\mathcal{A}_{\mathcal{P}}(x) \in Sol_{\mathcal{P}}(x)$. On dit que $\mathcal{A}_{\mathcal{P}}$ est un algorithme $f(|x|)$ -approché pour \mathcal{P} si et seulement si quelle que soit une instance x de \mathcal{P} on a : $R(x, \mathcal{A}_{\mathcal{P}}(x)) \leq f(|x|)$ et $\mathcal{A}_{\mathcal{P}}$ est polynomial en la taille de x . La fonction $f(|x|)$ est alors appelée ratio de performance ou ratio d'approximation de $\mathcal{A}_{\mathcal{P}}$.*

Il existe de nombreuses classes d'approximations. Deux d'entre elles nous serviront à classer les problèmes traités dans cette thèse, la classe des problèmes approximables à facteurs constants et la classe des problèmes non approximables à facteurs constants.

Définition 1.2.7. *APX est la classe d'approximation de tous les problèmes d'optimisation \mathcal{P} tel que pour une certaine constante $r \geq 1$, il existe un algorithme r -approché pour \mathcal{P} .*

Définition 1.2.8. *Non - APX est la classe d'approximation de tous les problèmes d'optimisation \mathcal{P} tel que quelle que soit la constante $r > 1$, il n'existe pas d'algorithme r -approché pour \mathcal{P} .*

1.2.4 Éléments de la programmation linéaire

La programmation linéaire nous fournit des outils très puissants pour la conception et l'analyse des algorithmes exacts et approchés pour de nombreux problèmes combinatoires. De plus, elle fournit un cadre unifié dans lequel la résolution exacte et l'approximation des problèmes pondérés et non-pondérés peuvent être traitées de façon uniforme.

Considérons qu'un problème discret Π est un problème de minimisation (pour plus de clarté, on le note $\min \Pi$), alors une formulation à l'aide d'un programme linéaire en variables binaires (avec n variable et $m + n$ contraintes) est un ensemble d'équations linéaires de la forme :

$$\min \Pi = \begin{cases} \min \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{i,j} x_j \geq b_i & i=1, \dots, m \\ x_j \in \{0, 1\} & i=1, \dots, n \end{cases} \quad (1.1)$$

Si Π est un problème de maximisation ($\max \Pi$), sa formulation est :

$$\max \Pi = \begin{cases} \max \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{i,j} x_j \leq b_i & i=1, \dots, m \\ x_j \in \{0, 1\} & i=1, \dots, n \end{cases} \quad (1.2)$$

Le relaxé de ces PL est obtenu en remplaçant $x_j \in \{0, 1\} \forall i = 1, \dots, n$ par $x_j \in [0, 1] \forall i = 1, \dots, n$.

Notons $\text{opt}(\min \Pi)$, $\text{opt}(\max \Pi)$ les valeurs optimales des formes 1.1 et 1.2, et $\text{opt}(\min \Pi_r)$, $\text{opt}(\max \Pi_r)$ respectivement les valeurs optimales de leur forme relaxée. Nous avons alors les inégalités suivantes :

$$\begin{aligned} \text{opt}(\min \Pi) &\geq \text{opt}(\min \Pi_r) \\ \text{opt}(\max \Pi) &\leq \text{opt}(\max \Pi_r) \end{aligned}$$

Ces deux inégalités sont très souvent utilisées pour obtenir des bornes sur la valeur optimale du problème (discret) que l'on veut résoudre approximativement.

1.3 Domaine d'application : les réseaux optiques

Avec une vitesse de transmission pouvant atteindre 1 petabit/s, la fibre optique est considérée comme le support permettant les plus hauts débits. Les réseaux optiques utilisent la fibre op-

tique comme support de transmission de l'information pour transporter des signaux sous forme optique et non électrique.

Dans les fils métalliques servant de support aux réseaux classiques, on transmet les informations par l'intermédiaire d'un courant électrique modulé. Avec la fibre optique, on utilise un faisceau lumineux modulé. Une connexion optique nécessite un émetteur et un récepteur ; l'émetteur pouvant être une diode électroluminescente, une diode laser, ou bien un laser modulé. L'importance du phénomène de dispersion dépend du type de l'émetteur. Ainsi, l'utilisation d'un laser permet d'avoir moins de dispersion et offre une puissance optique supérieure aux diodes. Le récepteur peut également être de différents types. On en distingue, cependant, deux principaux modèles. La photodiode PIN est un composant semi-conducteur de l'optoélectronique ayant comme particularité une jonction composée d'une zone intrinsèque intercalée entre une région fortement dopée P et une autre fortement dopée N. La diode à avalanche est également un composant semi-conducteur de l'optoélectronique ayant pour particularité de détecter des intensités de lumière extrêmement faibles. Le faisceau lumineux est véhiculé à l'intérieur d'une fibre optique à une très grande vitesse cependant réduite par ses composants extrémités [Puj08].

1.3.1 Avantages de la transmission par fibre optique

La fibre optique offre des avantages importants en matière de vitesse et de fiabilité de la transmission. La qualité de celle-ci se voit considérablement améliorée. Notons principalement la très large passante de l'ordre de 1 GHz pour 1 km permettant ainsi le multiplexage de très nombreux canaux sur un même support ; le poids d'un câble optique au kilomètre est neuf fois plus faible que celui d'un câble conventionnel et il bénéficie d'une grande flexibilité. Ainsi, le rayon de courbure peut descendre en dessous de 1 cm. Le câble conventionnel est fortement sensible aux éléments extérieurs ce qui peut engendrer une perturbation de la transmission. La fibre optique est insensible aux éléments climatiques, au bruit électromagnétique et aux variations de températures. La transmission est alors régulière et de bonne qualité. Le matériau privilégié dans l'élaboration de fibres optiques est la silice très pure. Ce matériau bon marché présente, en effet, des pertes optiques très faibles. Quand l'atténuation n'est pas le principal critère de sélection, on peut également mettre en œuvre des fibres en matière plastique. Le principal atout des fibres optiques est l'atténuation extrêmement faible qui est associée à l'affaiblissement du signal lumineux au cours de la propagation. Cette perte dépend de la qualité de la fibre. Cependant, la maîtrise des procédés de fabrication permet actuellement d'atteindre une atténuation aussi faible que 0,2 dB/km après 100 km de propagation, il restera donc encore 1 % de la puissance initialement injectée dans la fibre, ce qui peut être suffisant pour une détection par un terminal telle qu'une diode à avalanche. Toutes ces raisons donnent à la technologie optique une place de choix dans le domaine de la transmission de l'information.

1.3.2 Le multiplexage

Le multiplexage permet de faire passer plusieurs informations sur un seul support de transmission. L'utilisation d'un minimum de câble pour passer la même quantité d'information permet de faire d'importantes économies d'installation et d'exploitation. Nous distinguons deux principaux types de multiplexage :

Le multiplexage en temps (Time Division Multiplexing) consiste à découper la bande passante de la fibre optique en unités de temps, que vont se partager les différentes communications. Cette répartition du temps d'utilisation entre les communications permet la commutation à grande vitesse de chaque signal (figure 1.2).

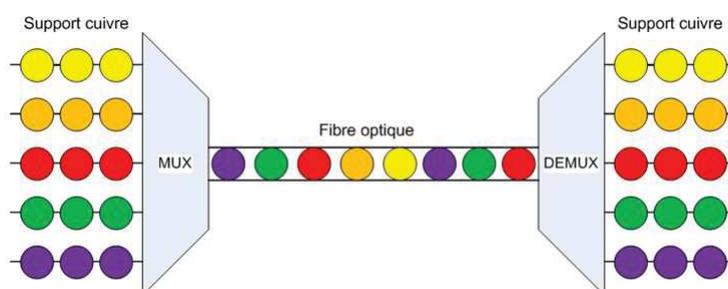


Figure 1.2 – Schéma permettant d'illustrer le découpage en temps entre les différentes connexions

Le multiplexage en longueur d'onde (Wavelength Division Multiplexing) est comparable au multiplexage en fréquence [Cur07] dans les réseaux cellulaires téléphoniques où chaque fréquence est utilisée comme un canal de communication. Cette technologie multiplexe en même temps plusieurs longueurs d'onde sur une même fibre optique. Chacune des longueurs d'onde est considérée comme un canal distinct pour la transmission des données (figure 1.3). Différents types de multiplexage en longueur d'onde existent. Le plus utilisé étant le Coarse-WDM qui permet de prendre en compte jusqu'à 16 longueurs d'onde avec un débit de transmission de 1,25 à 2,5 Gbit/s. Le Dense-WDM quant à lui, permet de multiplexer de 8 à 128 longueurs d'onde avec un débit de 10Gbit/s à 40 Gbit/s et enfin le Ultra-Dense-WDM permet de multiplexer un nombre de longueurs d'onde supérieur à 400 avec un débit dépassant les 40 Gbit/s.

1.3.3 Architecture des réseaux tout optique

Les réseaux tout optique utilisent la fibre optique comme support de transmission et la technologie de multiplexage en longueur d'onde (WDM) comme mode de transmission. L'idée de base est de réutiliser au maximum la même longueur d'onde pour acheminer l'information. Les réseaux WDM correspondent à une topologie maillée où les noeuds sont inter-connectés point-à-point. Ils sont principalement constitués de noeuds d'accès, d'OXC et de fibres optiques (figure 1.4). Les noeuds d'accès fournissent l'interface entre le cœur optique et les sous-réseaux non optiques des clients [Pro01]. Ces noeuds peuvent être des sources ou des destinations dans un chemin optique.

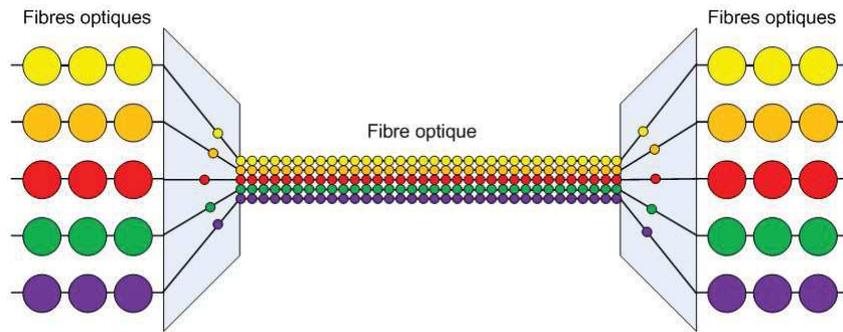


Figure 1.3 – Schéma permettant d’illustrer le découpage en longueur d’onde entre les différentes connexions

Ils sont équipés d’émetteurs mais aussi de récepteurs optiques afin de réaliser au mieux leurs fonctions d’envoi et de réception de signaux. Dans le cœur du réseau, les fonctions de commutation et de routage sont fournies par les OXC et permettent la mise en œuvre d’une communication de bout en bout entre les nœuds d’accès. Un OXC utilise la technologie du multiplexage de longueur d’onde. Il peut donc commuter chacune des longueurs d’ondes d’un port d’entrée vers un port de sortie quelconque. Enfin, le signal est acheminé via une fibre optique ayant le même ensemble de longueurs d’onde. Ceci permet de garantir qu’une longueur d’onde spécifique soit supportée tout au long d’un chemin optique et d’éviter ainsi la perte d’information.

1.3.4 Le routage multicast/broadcast tout optique

Le mode de routage broadcast/multicast a été introduit par Deering au début des années 90. Il avait constaté que l’envoi d’un seul message de la source à chaque sommet consommait beaucoup de ressources (bande passante). Il a alors proposé l’envoi par la source d’un unique message, puis sa duplication au cours de son acheminement [Dee92]. Ce mode de routage est appliqué dans plusieurs types de réseaux. Il est cependant nécessaire que le réseau soit muni d’équipements capables d’effectuer cette dite duplication. Dans les réseaux optiques, le routage broadcast/multicast est possible grâce à des OXC particuliers intégrant un coupleur optique et pouvant non seulement commuter chacune des longueurs d’onde entrantes vers un port de sortie quelconque, mais également diviser une longueur d’onde entrante et commuter chacun des sous-signaux obtenus vers plusieurs ports de sortie. Les OXC ayant cette capacité de division sont dits OXC-MC (OXC multicast capable) et ceux qui en sont dépourvus sont dits OXC-MI (OXC multicast incapable). De plus, la capacité de prélèvement de puissance est intégrée dans les OXC-MC afin de mieux soutenir le multicast. Compte tenu de sa rentabilité, le dispositif Tap-and-Continue (TaC) [AD00] permet de consommer une fraction du signal optique entrant pour l’usage local et de transmettre le reste du signal vers l’OXC suivant. Les noeuds MI ne prennent pas en charge la duplication optique mais emploient un dispositif Tac qui est plus simple et moins coûteux. La capacité TaC permet à un noeud MI d’exploiter une petite quantité de la puissance lumineuse d’entrée pour la détection des signaux et d’envoyer le reste à une seule sortie. Ainsi, nous supposons dans cette thèse que tous les OXC (OXC-MC et OXC-MI) du réseau WDM utilise la

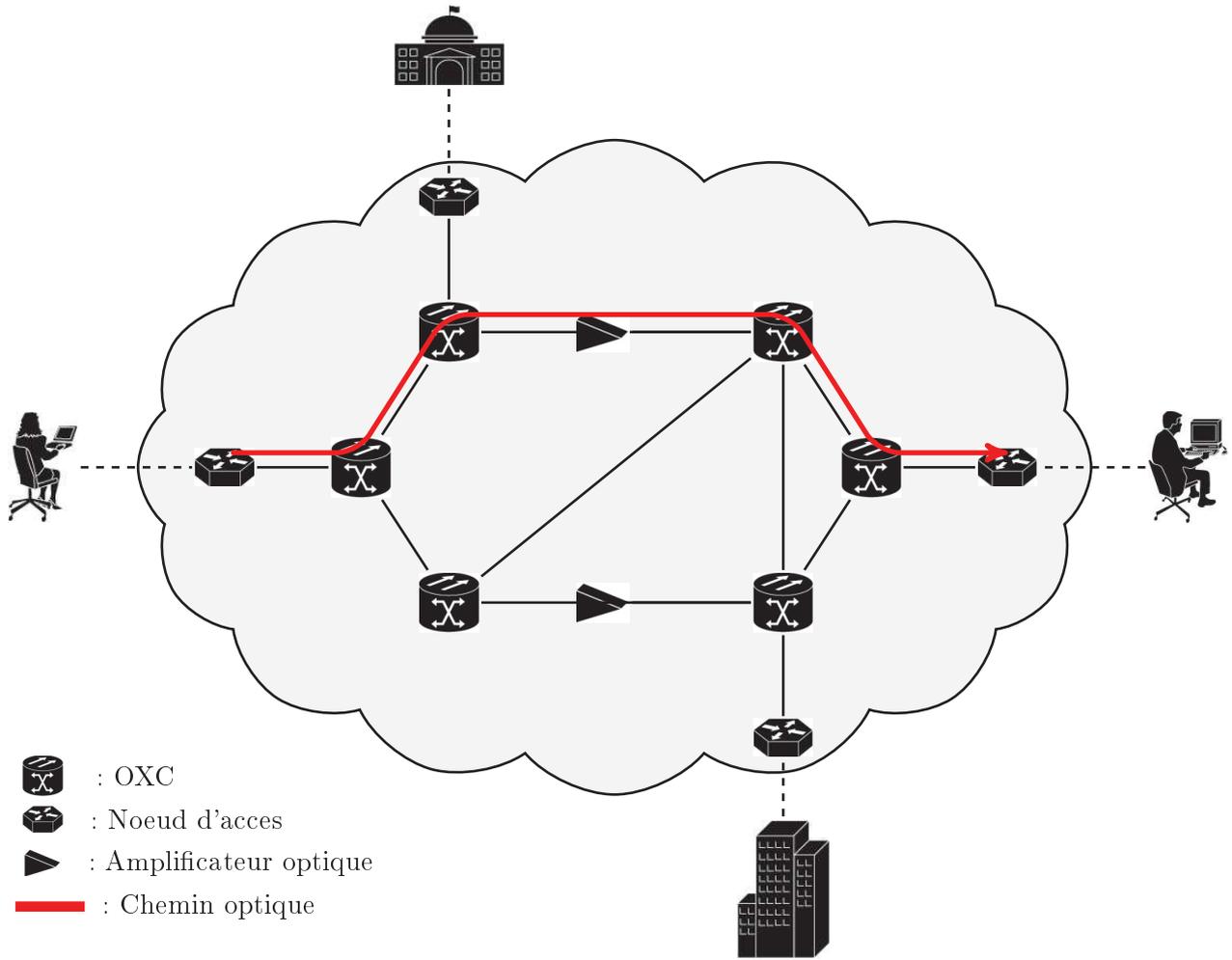


Figure 1.4 – Architecture d'un réseau tout optique WDM

technologie TaC.

Dans les réseaux équipés de switch OXC-MC, le routage multicast/broadcast se fait grâce à des arbres de recouvrements partiels, chacun prenant en compte une seule longueur d'onde. Si les switches OXC-MC sont suffisamment disponibles et bien positionnés dans le réseau, alors un seul arbre de recouvrement (une seule longueur d'onde) peut être suffisant. Bien que la contrainte principale imposée par la technologie WDM est que deux canaux optiques de même longueur d'onde ne peuvent emprunter une même fibre, la contrainte sur le degré des sommets de l'arbre de recouvrement est également justifiée. Effectivement, les sommets munis de coupleurs optiques sont capables de diviser le spectre lumineux arrivant, mais cette capacité est limitée. De plus, la division du spectre lumineux entraîne une perte de puissance. Plusieurs facteurs propres à la configuration du réseau optique peuvent ainsi impacter le recouvrement :

Impact de la conversion optique. Le convertisseur de longueur d'onde permet à une onde de changer de longueur. Des commutateurs équipés de convertisseurs offrant la possibilité de convertir les longueurs d'onde sont aujourd'hui technologiquement réalisables [RM98]. Ces derniers permettent à un chemin de changer de longueur d'onde au passage dans le commutateur, optimisant et simplifiant le processus d'allocation des longueurs d'onde. Cependant, le coût d'un tel équipement, largement supérieur aux coûts des commutateurs optiques sans conversion, limite sa production et sa commercialisation. Nous pouvons donc supposer que ce type d'équipements soit absent du réseau. La continuité de la longueur d'onde est donc exigée [Muk00]. Une information transitant sur un chemin optique allant de la source à une destination spécifique doit utiliser une unique longueur d'onde.

Impact de la distinction de longueur d'onde. Pour chaque longueur d'onde, la fibre optique ne contient qu'un seul sous-canal capable de l'acheminer et chaque sous-canal n'est capable d'acheminer qu'une seule longueur d'onde. Par conséquent, deux chemins optiques passant au moins une fois par une fibre commune ne peuvent avoir la même longueur d'onde.

Impact de la capacité de duplication du signal optique. Tous les commutateurs du réseau ne sont pas nécessairement des OXC-MC. Certains d'entre eux ne peuvent pas diviser le signal optique. Il est donc nécessaire de limiter le degré d'un tel nœud à deux dans la structure de recouvrement servant au routage. Il est également utile de limiter le degré d'un nœud équipé de commutateur OXC-MI, car le coupleur optique divise la puissance du signal entrant par le nombre de branches de sortie et achemine chaque fraction de lumière (chacune contenant la totalité du message de départ) vers une sortie spécifique. Cette division préserve le message original mais provoque une perte de puissance du signal. Cette perte de puissance peut, certes, être comblée par un amplificateur optique. Cependant, en l'absence de cet équipement, il est important de limiter la capacité de division de chaque commutateur afin de préserver une puissance du signal suffisante à la bonne réception par les destinations des informations transmises.

Impact du nombre de longueurs d'onde Les réseaux optiques WDM sont donc des réseaux multi-canaux. Chaque canal peut prendre en charge une longueur d'onde spécifique. Le nombre de canaux correspond au nombre de longueurs d'onde prises en charge par une fibre optique. La technologie WDM permet la division d'une fibre optique et peut atteindre 400 canaux pour multiplexer autant de longueurs d'onde. L'absence de conversion optique et la nécessité de distinguer les longueurs d'onde limitent l'utilisation de cette multitude de canaux. En effet, certaines longueurs d'onde disponibles peuvent ne pas être utilisables dans une session multicast, car la continuité de la longueur d'onde l'exige.

Cette multitude de contraintes ouvre un vaste champ d'étude sur les différents problèmes de routage dans les réseaux optiques selon la possibilité (ou pas) de conversion optique, la présence ou non d'amplificateurs optiques dans le réseau, ou encore selon de la disponibilité de commutateurs OXC-MC. Il est également intéressant de considérer une limitation sur le nombre de divisions

de la lumière par ces commutateurs. Il est possible de varier le nombre de canaux contenus dans une fibre optique. Nous pouvons envisager que le réseau soit multi-fibres avec plusieurs fibres optiques pouvant relier deux nœuds du réseau. Nous pouvons également mettre une borne sur le temps nécessaire pour effectuer le routage (ce temps pouvant être considéré comme la durée de la transmission de l'information de la source au dernier destinataire atteint). Plusieurs problèmes d'optimisation peuvent en découler. Citons par exemple, la minimisation du coût du routage, du nombre total de division du signal lumineux, ou encore, du nombre de OXC-MC nécessaires au routage multicast. Nous pouvons enfin envisager de minimiser le nombre de longueurs d'onde utilisées pour le routage. Les possibilités sont donc considérables ; mais nous avons décidé d'axer notre recherche sur une contrainte spécifique : la contrainte sur le degré des sommets.

1.4 État de l'art

Le travail que nous développons dans le cadre de cette thèse s'articule autour des problèmes de recherche de structure de recouvrement de graphes sous contrainte sur le degré des sommets. La minimisation du coût de la structure de recouvrement est l'objectif principal considéré. Comme l'arbre de recouvrement couvre les sommets d'un graphe connexe avec un minimum de liens, il est généralement proposé comme solution à ce type de problèmes. De nombreux cas pratiques exigent une limitation du degré des sommets de l'arbre. Cette limitation peut avoir plusieurs formes. Deux d'entre elles sont étudiées : la première exprime la nécessité de borner le degré des sommets de l'arbre et la deuxième exprime la capacité ou non de ces derniers à avoir plus de deux voisins. Dans cette section, l'essentiel des travaux antérieurs menés sur les arbres de recouvrement soumis à ces deux formes de contraintes est présenté. La connaissance de ces travaux permet de mieux estimer la plus-value apportée par la structure de hiérarchie de recouvrement présentée dans les chapitres suivants.

1.4.1 Arbre de recouvrement avec une borne supérieure sur le nombre de sommets de branchement

1.4.1.1 Définition et intérêt pratique

Nous rappelons que dans les réseaux optiques, il existe deux types de commutateurs. Les commutateurs munis de coupleur optique sont dits OXC-MC et ceux qui en sont démunis sont dits OXC-MI. Contrairement aux seconds, les premiers ont une capacité de division du signal lumineux. Naturellement, les commutateurs OXC-MC ont un coût de fabrication supérieur aux autres. Il est donc préférable de minimiser leur nombre dans le réseau. De plus, afin de permettre un routage multicast, il ne suffit pas de trouver un arbre de recouvrement du graphe modélisant le réseau ; il est également nécessaire que les sommets de degré strictement supérieurs à deux (sommets de branchement) correspondent à un nœud muni d'un commutateur OXC-MC. Enfin, minimiser le nombre de sommets de branchement peut contribuer à réduire la division du signal lumineux dans le réseau et, par conséquent, à réduire la perte de puissance de ce dernier. Gagano et al. ont formellement défini dans [GHSV02] le problème de recherche d'arbre de recouvrement

d'un graphe tel que le nombre de sommets de branchement $s(T)$ soit minimum :

Problème : Minimum Branch Vertices Spanning Tree - MBVST
Données : Graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$, $c(e) \forall e \in E_G$
Résultat : Arbre de recouvrement T du graphe G .
Objectif : Minimiser le nombre de sommets de branchement de T .

1.4.1.2 Complexité

Décider qu'un graphe admet un arbre de recouvrement T tel que $s^*(T) = 0$ revient à décider que ce graphe est hamiltonien, ce qui est NP-complet [GJ79]. L'araignée de recouvrement est un arbre particulier qui contient au plus un sommet de branchement. Décider qu'un graphe admet une araignée de recouvrement est également NP-complet [GHSV02]. En effet, posons $G = (V_G, E_G)$ un graphe connexe et $v \in V_G$. On construit un nouveau graphe G' en dupliquant 3 fois le graphe G . On ajoute un nouveau sommet z , on choisit un sommet quelconque $v \in V_G$, et on relie le sommet v au sommet z par une arête dans les 3 duplications (figure 1.5). Le graphe G' admet une araignée de recouvrement si et seulement si les 3 duplications du graphe G admettent un chemin hamiltonien ayant v pour extrémité. Or, décider qu'un graphe $G = (V_G, E_G)$ admet un chemin hamiltonien ayant $v \in V_G$ pour extrémité est NP-complet [Kar72].

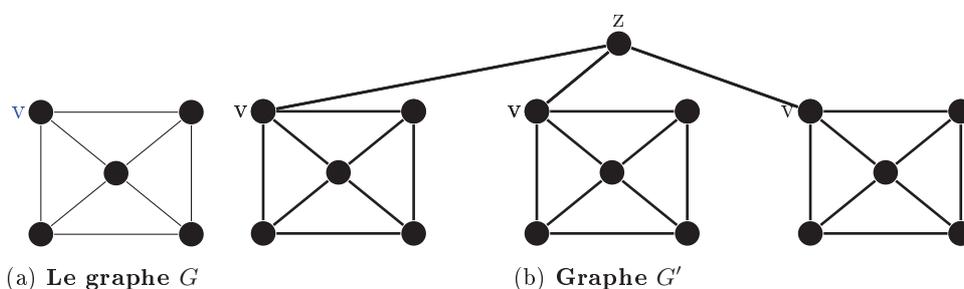


Figure 1.5 – Il existe un chemin hamiltonien dans les 4 duplications du graphe G . Le graphe (arbre) G' contient un seul noeud de branchement (le sommet z).

Plus généralement, soit k un entier positif. Il est NP-complet de décider qu'un graphe $G = (V_G, E_G)$ admet un arbre de recouvrement T tel que $s(T) \leq k$. Ceci se démontre facilement en généralisant la preuve précédente. Les cas $k = 0$ et $k = 1$ étant démontrés par les résultats précédents, nous pouvons poser $k \geq 2$. Étant donné un graphe $G = (V_G, E_G)$, on construit un nouveau graphe G' en dupliquant $2 \cdot k$ fois le graphe G et en ajoutant un graphe complet K_k . On choisit un sommet quelconque v de V_G , et on relie chaque sommet du graphe complet à deux duplications distinctes de G à partir de leur sommet v (figure 1.6). Comme dans tout arbre

de recouvrement de G' , les k sommets du graphe complet sont nécessairement des sommets de branchements, il est trivial de constater que le graphe G' admet un arbre de recouvrement ayant k sommets de branchement si et seulement si le graphe G admet un chemin hamiltonien ayant v pour sommet initial.

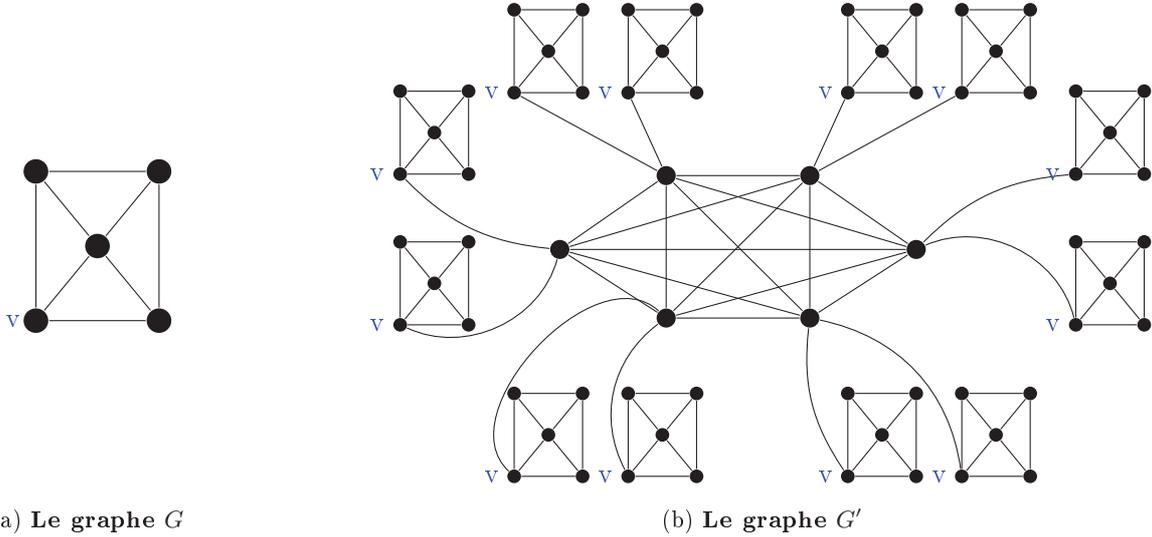


Figure 1.6 – Construction du graphe G' pour $k = 6$.

Dans la classe des graphes cubiques, le problème MBVST reste NP-complet [GHSV02]. Cette affirmation résulte de la relation entre le problème MBVST et le problème LONGEST PATH qui consiste à trouver un chemin de longueur maximum d'un graphe. Notons $\ell(G)$ la longueur d'un tel chemin. Les deux problèmes sont liés par l'équation la suivante :

$$\ell(G) \geq \frac{|V|}{s^*(G) + 1} \tag{1.3}$$

Étant donné un graphe $G = (V_G, E_G)$ et un entier positif k , Bazgan et al. démontrent dans [BST99] qu'il n'existe pas d'algorithme polynomial qui permette de trouver un chemin de longueur maximum de G tel que $\ell(G) \geq |V|/k$ même si G est cubique. Par conséquent, la valeur de $s^*(G)$ dans l'inégalité 1.3 ne peut être trouvée en temps polynomial.

Décider de l'existence d'une araignée de recouvrement devient facile sous certaines conditions. On note $\delta(G)$ le plus petit degré dans G et $\delta_k(G)$ la somme minimum des degrés de k sommets indépendants. Il est démontré dans [GHH⁺04] que si $\delta(G) \geq (n - 1)/3$ (plus généralement, $\delta_3(G) \geq (n - 1)$), alors G contient une araignée de recouvrement et il existe un algorithme polynomial ($O(|V_G|^3)$) qui la trouve. Dans [Dir52], Dirac prouve que si la somme des degrés de n'importe quelle paire de sommets indépendants de G est au moins égale à $|V_G| - 1$, alors G contient un chemin hamiltonien. Dans [GH03], Gargano et al. se réfèrent aux travaux de Dirac pour prouver que si la somme des degrés de n'importe quel triplet de sommets est au plus égale

à $|V_G| - 1$, alors G contient une araignée de recouvrement et il existe un algorithme polynomial ($O(|V_G|^3)$) qui la trouve. On note $\delta_k^u(G) = \min_I \sum_{v \in I} d(v)$, où I est un ensemble de taille k tel que $I \cup \{u\}$ est indépendant. Étant donné un graphe connexe $G = (V_G, E_G)$ et un sommet $u \in V_G$, Flandrin et al. prouvent dans [FKK⁺08] que si $\delta_2^u(G) \geq |V_G| - 1$, alors G contient une araignée de recouvrement centrée sur u . Dans [GHSV02], une conjecture généralisant le théorème de Dirac est proposée : si chaque sommet d'un graphe $G = (V_G, E_G)$ est de degré au moins $\frac{|V_G|-1}{k+2}$ (plus généralement, si la somme des degrés de chaque ensemble de $k+1$ sommets indépendants de G est d'au plus $|V_G| - 1$), alors $s^*(G) \leq k$. Cette conjecture est prouvée pour $k = 0$ et $k = 1$, mais reste cependant ouverte pour $k \geq 2$.

Il est démontré que la borne sur le degré qui est suffisante pour l'existence d'un chemin hamiltonien peut être améliorée pour les graphes bipartis [Ber85]. Il en est de même pour l'existence de l'araignée de recouvrement [GH03]. En effet, soit $G(V1_G, V2_G, E_G)$ un graphe biparti connexe tel que $|V1_G| \leq |V2_G|$, si pour tout $u \in V1_G$ et $v \in V2_G$, $d_G(u) + d_G(v) \geq |V2_G|$ alors, pour tout $u \in V1_G$, le graphe G contient une araignée de recouvrement centrée sur u .

Un graphe $K_{1,3}$ est appelé *claw*. Un graphe est dit *claw-free* s'il ne contient pas $K_{1,3}$ comme sous-graphe induit. Il est prouvé dans [MOY14] que si dans un graphe *claw-free* de taille n , la somme des degrés de tout ensemble de cinq sommets indépendants est supérieure ou égale à $n-2$, alors il existe un arbre de recouvrement de ce graphe avec au plus un sommet de branchement. Cette condition de densité d'un graphe *claw-free*, qui permet de répondre par l'affirmative à l'existence d'une araignée de recouvrement de celui-ci, est la meilleure possible.

1.4.1.3 Résolution exacte et approchée

La résolution exacte du problème MBVST est réalisée en utilisant la programmation linéaire en nombres entiers. La résolution approchée est réalisée en effectuant des relaxations des programmes linéaires ou bien à l'aide d'algorithmes d'approximations classiques. Ce problème n'appartient pas à la classe d'approximation APX. Il est de ce fait non-approximable à facteur constant près [BGPea97]. Les heuristiques proposés dans la littérature sont donc sans garantie de performance.

1.4.1.3.a Utilisation de la programmation linéaire

Plusieurs formulations mathématiques du problème MBVST ont été proposées. Une formulation en programme linéaire en nombres entiers avec un nombre exponentiel de contraintes d'une part ; une deuxième formulation en programme linéaire en nombres entiers basée sur le concept de flot d'autre part ; et enfin une formulation en programme linéaire en nombres entiers basée sur le concept de flots multiples.

1. **Formulation en programme linéaire en nombres entiers (PLNE) avec un nombre exponentiel de contraintes [CCGG13]** : Soit un graphe connexe $G = (V_G, E_G)$. L'ensemble des variables est le suivant : une variable binaire x_e pour chaque arête $e \in E_G$,

qui vaut 1 si l'arête e est sélectionnée, 0 sinon. Une autre variable binaire y_v pour chaque sommet $v \in V_G$ qui est égale à 1 si le sommets v est un sommet de branchement, 0 sinon. Le programme se présente comme suit :

$$\min \sum_{v \in V_G} y_v \quad (1.4)$$

$$\sum_{e \in E_G} x_e = |V_G| - 1 \quad (1.5)$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \quad \forall S \subseteq V_G \quad (1.6)$$

$$\sum_{e \in \lambda_G(v)} x_e - 2 \leq d_G(v) y_v \quad \forall v \in V_G \quad (1.7)$$

$$y_v \in \{0, 1\} \quad \forall v \in V_G \quad (1.8)$$

$$x_e \in \{0, 1\} \quad \forall e \in E_G \quad (1.9)$$

Pour chaque sous ensemble $S \subseteq V_G$, on note $E(S)$ l'ensemble des arêtes ayant leurs deux extrémités dans S . La fonction objective (1.4) vise à minimiser le nombre total de sommets de branchement. La contrainte (1.5) garantit la présence de exactement $|V| - 1$ arêtes dans l'arbre de recouvrement. La contrainte (1.6) garantit que l'arbre de recouvrement est sans cycle. La contrainte (1.7) assure que la variable y_v est égale à 1 si et seulement si le degré du sommet v est strictement supérieur à 2 dans l'arbre de recouvrement.

2. **Formulation en programme linéaire en nombres entiers basée sur le concept de flot (PLNE-CP) [CGI09]** : Pour la formulation basée sur le flot, on considère la version orientée symétrique du graphe où deux arcs (u, v) et (v, u) sont associés à chaque arête $\{u, v\} \in E_G$. On note par E' ce nouvel ensemble d'arcs. On note $A^+(v)$ et $A^-(v)$ l'ensemble des arcs de E' sortant et entrant dans v , respectivement. Des variables de flot sont également introduites. On note par $f_{u,v}$ la quantité de flot transitant dans l'arc $(u, v) \in E'$. Une source est choisie de manière arbitraire parmi tous les sommets du graphe et une unité de flot est envoyée à partir de cette source à destination de chaque sommet $v \in V_G \setminus \{s\}$. Les arcs faisant transiter une quantité de flot non nulle correspondent aux arêtes de l'arbre de recouvrement. Ceci permet de garantir la connexité de l'arbre de recouvrement construit.

$$\min \sum_{v \in V_G} y_v \quad (1.10)$$

$$\sum_{(u,v) \in A^-(v)} x_{u,v} = 1 \quad (1.11)$$

$$\sum_{(s,v) \in A^+(s)} f_{s,v} - \sum_{(v,s) \in A^-(s)} f_{v,s} = |V_G| - 1 \quad (1.12)$$

$$\sum_{(v,u) \in A^+(v)} f_{v,u} - \sum_{(u,v) \in A^-(v)} f_{u,v} = -1 \quad \forall v \in V_G \setminus \{s\} \quad (1.13)$$

$$x_{u,v} \leq f_{u,v} \leq (|V_G|) \cdot x_{u,v} \quad \forall (u,v) \in E'_G \quad (1.14)$$

$$\sum_{(v,u) \in A^+(v)} x_{v,u} + \sum_{(u,v) \in A^-(v)} x_{u,v} - 2 \leq d_G(v) y_v \quad \forall v \in V_G \quad (1.15)$$

$$y_v \in \{0, 1\} \quad \forall v \in V_G \quad (1.16)$$

$$x_e \in \{0, 1\} \quad \forall (u,v) \in E'_G \quad (1.17)$$

$$f_{u,v} \geq 0 \quad \forall (u,v) \in E'_G \quad (1.18)$$

La fonction objective (1.10) vise toujours à minimiser le nombre de sommets de branchement de l'arbre de recouvrement. La contrainte (1.11) assure que chaque sommet de l'arbre de recouvrement a exactement un arc entrant. Les contraintes (1.12) et (1.13) assurent qu'une unité de flot soit envoyée depuis la source à chaque sommet de l'arbre de recouvrement, garantissant ainsi la connexité de toute solution réalisable. La contrainte (1.14) relie les variables de flot f aux variables binaires x . La contrainte (1.15) garantit que la variable y_v est égale à 1 si et seulement si le degré du sommet v est strictement supérieur à 2 dans l'arbre de recouvrement.

3. **Formulation en programme linéaire en nombres entiers basée sur le concept de flots multiples (PLNE-CPM) [CCGG13]** : Le problème MBVST peut également être formulé en utilisant le concept de flots multiples [BMMN95]. Pour ce faire, en introduisant le même ensemble d'arc E' , nous introduisons, pour chaque arc $(u,v) \in E'$ un ensemble de variables de flot $f_{u,v}^k$, où $k \in V_G$ représente le flot passant par l'arc (u,v) relatif au chemin allant de la source arbitraire s au sommet k .

$$\min \sum_{v \in V_G} y_v \quad (1.19)$$

$$\sum_{(u,v) \in A^-(v)} x_{u,v} = 1 \quad \forall v \in V_G \setminus \{s\} \quad (1.20)$$

$$\sum_{(v,u) \in A^+(v)} f_{v,u}^k - \sum_{(u,v) \in A^-(v)} f_{u,v}^k = 0 \quad \forall v \in V_G, v \in V_G \setminus \{s\}, v \neq s \quad (1.21)$$

$$\sum_{(s,v) \in A^+(s)} f_{s,v}^k - \sum_{(v,s) \in A^-(s)} f_{v,s}^k = 1 \quad \forall v \in V_G \setminus \{s\} \quad (1.22)$$

$$\sum_{(k,u) \in A^+(k)} f_{k,u}^k - \sum_{(i,k) \in A^-(k)} f_{i,k}^k = -1 \quad \forall k \in V_G \setminus \{s\} \quad (1.23)$$

$$f_{u,v}^k \leq x_{u,v} \quad \forall (u,v) \in E'_G, k \in V_G \quad (1.24)$$

$$\sum_{(v,u) \in A^+(v)} x_{v,u} + \sum_{(u,v) \in A^-(v)} x_{u,v} - 2 \leq d_G(v)y_v \quad \forall v \in V_G \quad (1.25)$$

$$y_v \in \{0, 1\} \quad \forall v \in V_G \quad (1.26)$$

$$x_{u,v} \in \{0, 1\} \quad \forall (u,v) \in E'_G \quad (1.27)$$

$$f_{u,v}^k \geq 0 \quad \forall (u,v) \in E_G, k \in V_G \quad (1.28)$$

L'objectif est identique à celui des modèles précédents. La contrainte (1.20) assure que chaque sommet de l'arbre de recouvrement a exactement un arc entrant. Les contraintes de conservation de flot (1.21)-(1.23) garantissent que la source s est connectée à tous les sommets du graphe. La contrainte (1.24) relie les variables de flot aux variables binaires d'arcs. La contrainte (1.25) est la même que dans le modèle précédent.

En raison du nombre exponentiel de contraintes, la première formulation n'est pas appropriée pour la résolution exacte d'instances de grande taille. Cette formulation est néanmoins utilisée pour la relaxation lagrangienne. Le dual de cette relaxation donne une solution réalisable qui est une borne supérieure pour le problème MBVST [CGI09]. Dans [CGI09], une méthode de résolution exacte est appliquée à la formulation **PLNE-CP** sur un jeu de graphes aléatoires pour avoir une solution exacte au problème MBVST. Cette résolution montre ses limites à partir d'instances de taille 100. Afin d'obtenir une borne inférieure à ce problème, Carrabs et al., dans [CCGG13] proposent une relaxation en variables mixtes de la formulation **PLNE-CP**. Cette relaxation est obtenue en remplaçant la contrainte (1.17) par la contrainte $0 \leq x_{u,v} \leq 1, \forall (u,v) \in E'$, notée (1.21'). La seconde relaxation proposée dans ce même article est une relaxation continue. Elle est obtenue en remplaçant la contrainte (1.17) par (1.21') mais aussi la contrainte (1.16) par la contrainte $0 \leq y_v \leq 1, \forall v \in V_G$, notée (1.20').

Aucune méthode de résolution exacte n'a été appliquée à la formulation **PLNE-CPM**, on ne connaît donc pas les performances que peut apporter cette formulation. Il est également proposé dans [CCGG13] une relaxation continue de ce modèle en remplaçant les contraintes (1.27) et (1.26) par les contraintes (1.21') et (1.20'), respectivement.

1.4.1.3.b Utilisation d'heuristiques

Dans [CGI09], trois heuristiques gloutonnes sont proposées pour résoudre de façon approchée (sans garantie de performance) le problème MBVST.

1. **Heuristique basée sur le poids des arêtes :** L'idée de cette heuristique est d'ajouter une fonction $\omega : E_G \rightarrow \mathbb{N}^+$ qui attribue un poids à chaque arête $\{u, v\} \in E_G$. Le poids $\omega(u, v)$ de l'arête $\{u, v\}$ est utilisé comme indicateur de la probabilité que cette arête crée un sommet de branchement si elle est sélectionnée pour être dans l'arbre de recouvrement. L'algorithme commence par assigner le poids 1 à chaque arête du graphe G . À chaque itération, une arête $\{u^*, v^*\}$ ayant un poids minimum est sélectionnée à condition que cette arête ne soit pas préalablement dans l'arbre T , et que le sommet u^* et le sommet v^* soient dans deux composantes connexes différentes. Une fois cette arête sélectionnée, elle est ajoutée à l'arbre T et supprimée du graphe G . Le poids de chaque arête incidente à u^* ou à v^* est alors incrémenté de 1, ce qui a pour effet de diminuer leur probabilité d'être sélectionnée à l'itération suivante. L'algorithme s'arrête quand $|V_G| - 1$ arêtes sont sélectionnées.
2. **Heuristique basée sur les couleurs des sommets :** Cette approche est basée sur l'assignation d'une couleur à chaque sommet du graphe $c : v \rightarrow \{V(ert), B(leu), J(aune), R(ouge)\}$. La couleur $c(v)$ d'un sommet v indique si ce sommet deviendra sommet de branchement quand une arête incidente à celui-ci est sélectionnée pour être insérée dans l'arbre de recouvrement. Une couleur est attribuée à un sommet comme suit :

$$V, \text{ si } d_T(v) = 0 \tag{1.29}$$

$$B, \text{ si } d_T(v) = 1 \tag{1.30}$$

$$J, \text{ si } d_T(v) = 2 \tag{1.31}$$

$$R, \text{ si } d_T(v) \geq 3 \tag{1.32}$$

L'algorithme assigne la couleur verte à chaque sommet $v \in V_G$. À chaque itération, une arête est sélectionnée en fonction des couleurs de ces extrémités pour être insérée dans l'arbre de recouvrement. Notons que le nombre de nouveaux sommets de branchement créés après l'insertion de l'arête $\{u, v\}$ est le nombre de sommets jaunes constituant ses deux extrémités. L'algorithme choisit ainsi une arête ayant un nombre minimum d'extrémités jaunes. Afin de minimiser le nombre de nouveaux sommets jaunes potentiels, l'algorithme sélectionne, parmi les arêtes ayant un nombre minimum d'extrémités jaunes, celles ayant le moins de sommets bleus comme extrémités. L'algorithme vérifie avant d'ajouter une arête à l'arbre de recouvrement que celle-ci ne crée pas de cycle. Ainsi, avant l'ajout d'une arête $\{u, v\}$, les sommets u et v doivent être dans deux composantes connexes différentes. Quand une arête est ajoutée, les couleurs des sommets extrémités de celle-ci sont mis à jour en fonction de leurs degrés. Enfin, une procédure consistant à ajouter à l'arbre de recouvrement (sans créer de cycles) les arêtes reliant deux sommets (deux sommets de

couleur rouge), car cet ajout ne crée pas de sommets de branchement supplémentaires. L'algorithme s'arrête quand $|V_G| - 1$ arêtes sont sélectionnées.

3. **Heuristique basée sur la combinaison des deux approches précédentes :** Cette approche propose une sélection d'arêtes encore plus précise en combinant les deux approches précédentes. À chaque itération, l'ensemble des arêtes ayant un poids minimum est sélectionné. Parmi ces arêtes, celles ayant un nombre minimum d'extrémités coloriées en jaune sont gardées. Parmi ces dernières, celles ayant un nombre minimum d'extrémités coloriées en bleu sont enfin sélectionnées pour être dans l'arbre de recouvrement.

Des séries de tests sur des graphes aléatoires ont permis d'évaluer les performances des trois heuristiques. Sans surprise, l'heuristique basée sur la combinaison des deux approches donne les meilleurs résultats. Malgré le fait qu'il n'existe pas de garantie de performance, elle retourne, dans le pire des cas, une solution approchée de coût égal à trois fois celui de la solution optimale. Elle retourne une solution optimale dans 22% des instances testées.

Dans [SSM⁺11], une heuristique basée sur les travaux de [DK97] est proposée. Cette heuristique commence par calculer un arbre de poids minimum T du graphe G . Cet arbre peut contenir plusieurs sommets de branchement. L'arbre T sera modifié de manière itérative pour réduire le nombre de sommets de branchement. À chaque itération, une liste d'arêtes L_{cut} à remplacer est constituée. Cette liste est formée des arêtes dont au moins une extrémité est un sommet de branchement. Une arête de cette liste est sélectionnée en se basant sur deux critères. Le premier critère α est le nombre d'extrémités de l'arête qui sont sommets de branchement. Le deuxième critère σ est la somme des degrés de ses extrémités. L'arête de L_{cut} ayant la plus grande valeur de α (et la plus grande valeur σ en cas d'égalité) est retirée de l'arbre T , ceci divisant l'arbre T en deux composantes connexes. Chaque arête retirée est sauvegardée dans une liste L_{list} afin de ne pas la réinsérer dans l'arbre et d'éviter ainsi les cycles. La liste L_{rep} contient toutes les arêtes qui peuvent connecter les deux composantes connexes et qui ne sont pas dans L_{list} . L'arête de L_{rep} ayant la plus petite valeur de α (et la plus petite valeur de σ en cas d'égalité) est ajoutée à l'arbre T permettant ainsi de connecter les deux composantes.

Sans garantie de performance, il est difficile de hiérarchiser les performances des heuristiques. Cette heuristique retourne cependant un meilleur résultat que les trois précédentes dans 78% des instances testées.

1.4.2 Arbre de recouvrement tel que la somme des degrés des sommets de branchement est minimum

1.4.2.1 Définition et intérêt pratique

Réduire le nombre de commutateurs OXC-MC nécessaires au routage multicast/broadcast ne suffit pas. Il faut également réduire le nombre de nœuds voisins de ceux-ci. En effet, chaque nœud muni d'un commutateur OXC-MC et ayant m voisins dans l'arbre de recouvrement di-

visé le signal optique arrivant en $m - 1$ sous-signaux. Chaque sous-signal est transmis à un nœud voisin distinct parmi les $m - 1$ existants (en excluant le voisin par lequel le signal est arrivé) [AD06]. Plus le nombre de voisins est grand, plus le signal est divisé et plus la perte de puissance du signal est importante. Afin de minimiser cette perte, il est donc nécessaire de minimiser la somme des voisins de tous les nœuds munis de commutateur OXC-MC participant au routage multicast/broadcast. Cela revient à minimiser la somme des degrés des sommets de branchement dans l'arbre de recouvrement du graphe modélisant le réseau optique.

Observons la figure 1.7 et supposons que le graphe G modélise un réseau optique et que l'on veuille envoyer en mode multicast une information de la source v à tous les autres nœuds du réseau. Un arbre de recouvrement de ce graphe ayant un minimum de sommets de branchement est représenté par la figure 1.7b. Cet arbre contient un seul sommet de branchement. Cependant, le signal lumineux qui arrive au sommet u est divisé en 7 sous-signaux. La figure 1.7c illustre un arbre de recouvrement de G ayant également un seul sommet de branchement mais tel que son degré est minimum. Dans ce cas, le signal sera divisé seulement par 2, provoquant ainsi une perte de puissance du signal moins importante que dans le cas précédent.

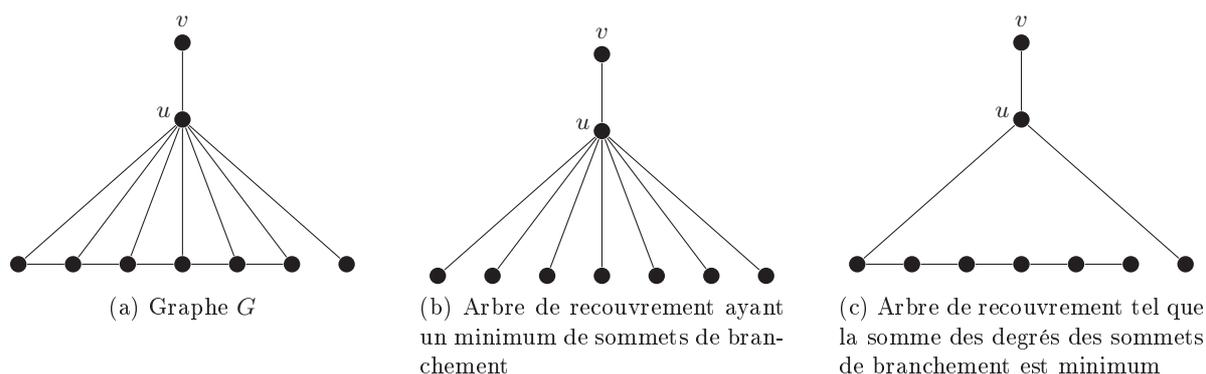


Figure 1.7 – Illustration de l'utilité de minimiser le degré des sommets de branchement

Le problème MDST est encore plus récent que le problème MBVST. Il a été initialement évoqué et défini dans [CGI09].

Problème : Minimum Degree Sum Spanning Tree - MDST
Données : Graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$
Résultat : Arbre de recouvrement T du graphe G .
Objectif : Minimiser la somme des degrés des sommets de branchement de T .

1.4.2.2 Résolution exacte et approchée

À l'instar du problème MBVST, le problème MDST est NP-difficile [CGI09]. En effet, s'il existe un arbre de recouvrement T d'un graphe G sans aucun sommet de branchement alors cet arbre est optimal pour le problème MDST et cet arbre est aussi un chemin hamiltonien de G . Décider de l'existence d'un arbre de recouvrement d'un graphe tel que la somme des degrés de ses

sommets de branchement est égale à 0 revient à décider que le graphe est hamiltonien. Sachant qu'un sommet de branchement a un degré au moins égal à 3, vérifier si G admet un arbre de recouvrement tel que $q(T) < 6$ revient à vérifier si celui-ci admet une araignée de recouvrement. Soit $k \geq 6$. Il est NP-complet de décider qu'un graphe $G = (V_G, E_G)$ admet un arbre de recouvrement T tel que $q(T) \leq k$. En effet, étant donné un graphe $G(V_G, E_G)$, on construit un nouveau graphe G' en dupliquant $k - 2$ fois le graphe G , on crée deux sommets $z1$ et $z2$ et on relie ces deux derniers, on choisit un sommet quelconque v , et on relie le sommet v de deux duplications de G au sommet $z1$ et le sommet v des $k - 4$ autres duplications au sommet $z2$ (figure 1.8). Le graphe G' admet un arbre de recouvrement T tel que $q(T) \leq k$ si et seulement si les $k - 2$ duplications du graphe G admettent un chemin hamiltonien ayant v pour sommet initial.

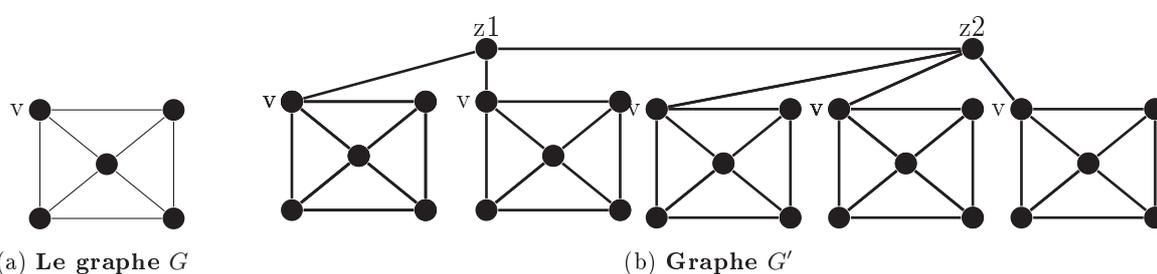


Figure 1.8 – Dans cet exemple on pose $k = 7$. Il existe un chemin hamiltonien dans G . L'arbre couvrant le graphe G' contient deux sommets de branchement (les sommets $z1$ et $z2$).

Une seule formulation mathématique du problème MDST a été proposée dans la littérature. Il s'agit d'un programme linéaire en nombres entiers basé sur le concept de flot proposé dans [CGI09]. Cette formulation est identique à celle du problème MBVST évoquée dans ce même article, mis à part l'ajout d'une variable entière qui représente le degré des sommets de branchement de la solution :

$$z_v = \begin{cases} d_{T_G}(v), & \text{si } v \text{ est un sommet de branchement,} \\ 0, & \text{sinon.} \end{cases}$$

Le modèle mathématique du MDST vise à minimiser la fonction objective :

$$\min m(T_G) = \sum_{v \in V} z_v \tag{1.33}$$

On ajoute au PL du MBVST la contrainte suivante :

$$\sum_{e=(u,v) | u \in \omega(v)} x_e - 2 + y_v \leq z_v \quad \forall v \in V \tag{1.34}$$

On observe que la partie gauche de la contrainte (1.34) est négative quand y_v est égale à zéro. Par conséquent et du fait de la minimisation de la fonction objective, si le sommet v n'est pas un sommet de branchement alors z_v est égale à 0. En revanche, si v est un sommet de branchement alors le variable z_v est au moins égale à $d_T(v)$.

L'heuristique basée sur le poids des arêtes, l'heuristique basée sur les couleurs des sommets ainsi que l'heuristique hybride (basée sur la combinaison des deux approches) qui ont été proposées dans [CGI09] pour résoudre le problème MBVST, sont légèrement modifiées pour résoudre le problème MDST. Le changement réside dans la modification du critère de sélection des arêtes qui doivent être dans l'arbre de recouvrement. Ainsi, il ne suffit plus de sélectionner une arête par hasard parmi toutes celles qui ont un même poids minimum, mais il faut aussi s'assurer de prendre celle qui a un degré minimum. Si un sommet devient sommet de branchement, la stratégie qui consiste à essayer d'augmenter son degré car il est déjà sommet de branchement ne fonctionne pas. Certes, augmenter le degré des sommets qui sont déjà sommets de branchement n'augmente pas le nombre de sommets de branchement de l'arbre de recouvrement. Mais cela augmente la valeur de la somme des degrés des sommets de branchement de l'arbre de recouvrement.

1.4.2.3 Relation entre le problème MBVST et le problème MDST

Il existe une corrélation entre le problème MBVST et le problème MDST. En effet, la somme des degrés des sommets de branchement d'un arbre optimal pour le problème DCMST est toujours supérieur ou égal au nombre de sommets de branchement d'un arbre optimal pour le problème MBVST fois trois ($q^*(G) \geq 3 \cdot s^*(G)$) [CGI09]. S'il existe un arbre de recouvrement sans sommet de branchement, alors cet arbre est optimal pour les deux problèmes. S'il existe un arbre de recouvrement T optimal pour le problème MBVST dont tous les sommets de branchements sont de degré trois alors cet arbre est aussi optimal pour le problème MDST. À l'inverse, s'il existe une solution pour le problème MDST avec un seul sommet de branchement, cette solution est aussi optimale pour le problème MBVST. Pour montrer que les deux problèmes sont distincts, [CCGG13] Cerulli et al. décrivent une instance (figure 1.9) pour laquelle il existe un arbre de recouvrement optimal pour le problème MBVST qui n'est pas optimal pour le problème MDST. L'arbre T' est quant à lui optimal pour les deux problèmes.

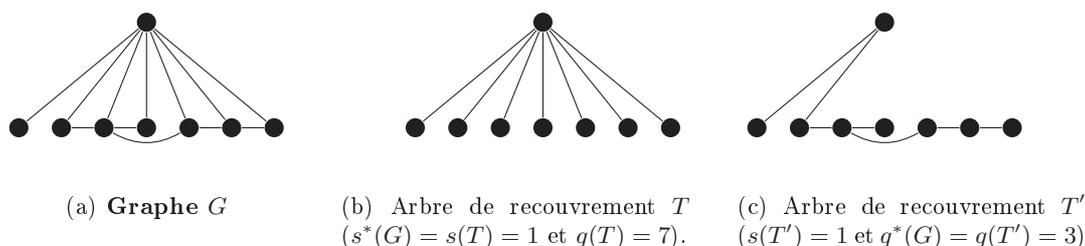


Figure 1.9 – Exemple exhibé dans l'article [CGI09]

Dans [CCR14], une instance est donnée pour laquelle il existe un arbre de recouvrement optimal pour le problème MDST, qui n'est pas optimal pour le problème MBVST. En effet, dans la Figure 1.10, l'arbre de recouvrement T est bien optimal pour le problème MDST mais pas pour le problème MBVST. L'arbre T' , quant à lui, est optimal pour les deux problèmes.

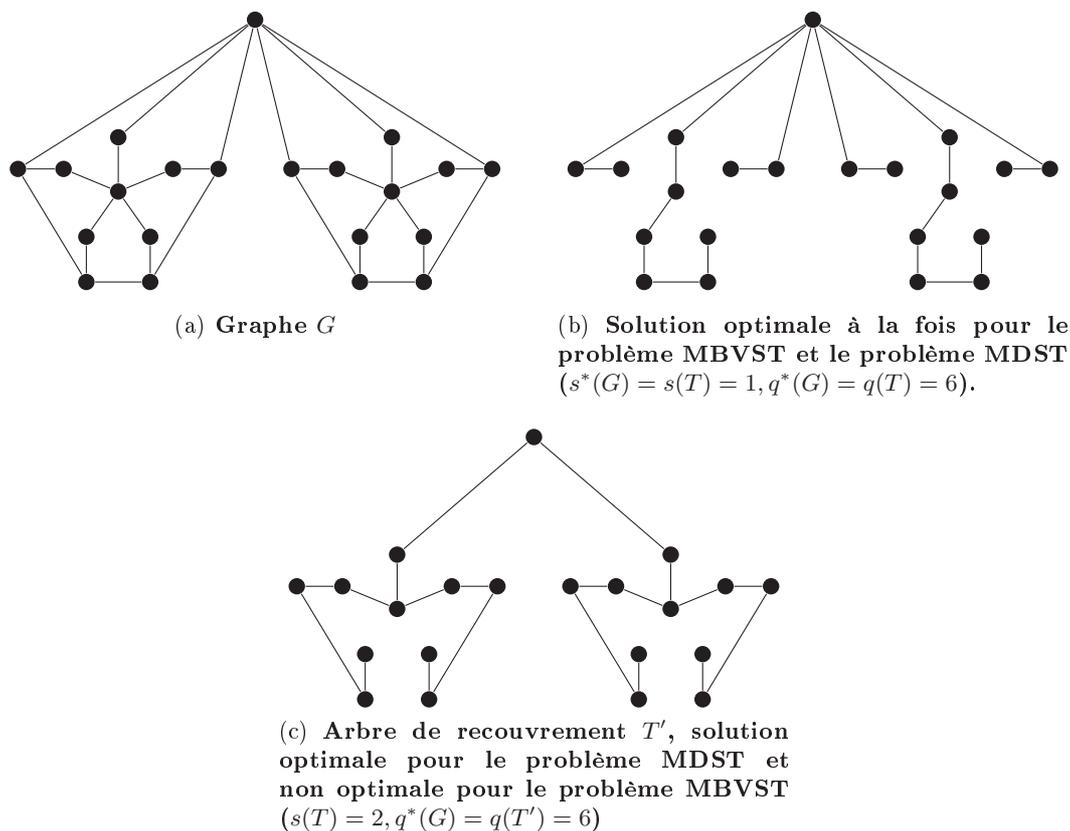


Figure 1.10 – Instance contenant un arbre optimal pour le MDST qui n'est pas optimal pour le MBVST

Les instances des figures 1.9 et 1.10 prouvent que :

1. Il existe un graphe G dans lequel il existe un arbre de recouvrement T de G pour lequel $s(T) = s^*(G)$ et $q(T) \neq q^*(G)$,
2. Il existe un graphe $G' \neq G$ dans lequel il existe un arbre de recouvrement T' de G' pour lequel $q(T') = q^*(G)$ et $s(T') \neq s^*(G')$.

Bien que ces deux exemples prouvent que les deux problèmes sont distincts et que les solutions optimales pour l'un ne sont pas nécessairement optimales pour l'autre, ils ne prouvent pas qu'il n'existe pas toujours au moins une solution optimale pour les deux problèmes à la fois.

1.4.3 Arbre de recouvrement avec une borne supérieure sur le degré des sommets

1.4.3.1 Définition et intérêt pratique

Le problème DCMST est un problème NP-difficile [GJ79] largement étudié dans la littérature et ayant de nombreuses applications pratiques. Il est présenté initialement en 1968 dans [DH68] et évoqué brièvement la même année dans [Obr68].

Deo et Hakimi expliquent d'ores et déjà l'intérêt pratique de ce problème en évoquant la nécessité de réduire le nombre de liens reliés à chaque point dans les réseaux de communication, ceci car les hubs ont une capacité limitée. Ils se sont inspirés entre autres des travaux de [FY62] dans le domaine de la fiabilité et la robustesse des réseaux de communication pour insister sur la nécessité d'éviter d'avoir un point du réseau avec un degré élevé ; car en cas de panne de celui-ci, la reconnexion des nombreux points adjacents serait problématique.

On peut actualiser la liste des cas pratiques en évoquant la duplication limitée de la lumière dans les réseaux optiques. En effet, le degré des nœuds des arbres optiques ne peut dépasser la capacité de duplication du commutateur optique correspondant.

Le problème DCMST trouve également un intérêt dans les réseaux de capteurs. Dans [LHL08], un protocole basé sur le DCMST est proposé. Ce protocole permet d'augmenter la durée de vie du réseau et donne un bon compromis entre efficacité énergétique et latence.

Une des principales problématiques dans les réseaux ad-hoc est le coût des communications. Si l'infrastructure de communication contient des sommets de forts degrés, des effets indésirables peuvent être observés. Parmi ces effets, on peut citer le phénomène de congestion ou la perte de messages due à leur collision. De plus, les sommets de fort degré sont des cibles privilégiées lors d'attaques de sécurité visant un réseau. En effet, la paralysie des sommets de fort degré perturbe, voire empêche l'acheminement des informations dans un réseau. Dans tous ces cas pratiques, il est donc souhaitable de construire des structures de communication où le degré maximum des sommets est borné.

Le routage multicast/broadcast dans les réseaux tout optique constitue notre principale motivation. Dans ce cas de figure, l'utilité de limiter le degré d'un nœud équipé de commutateur OXC-MI est avérée. En effet, le coupleur optique divise la puissance du signal entrant par le nombre de branches de sortie et achemine chaque fraction de lumière (chacune contenant la totalité du message de départ) vers une sortie spécifique. Cette division préserve le message original mais provoque une perte de puissance du signal. En l'absence d'amplificateurs optiques ou en leur présence de manière insuffisante dans le réseau, il est important de limiter la capacité de division de chaque commutateur afin de préserver une puissance du signal suffisante à la bonne réception par les destinataires des informations transmises. Remarquons également que si un réseau tout optique est dépourvu de commutateur OXC-MI, alors le degré de tous les sommets

appartenant à l'arbre de recouvrement du graphe modélisant ce réseau doit être au plus égal à deux.

Deo et Hakimi ont défini le problème DCMST comme suit [DH68] :

Problème : Degree Constrained Minimum Spanning Tree - DCMST
Données : Graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$, une fonction $C : E_G \rightarrow \mathbb{R}_*^+$ associant un coût $c(e)$ à chaque arête $e \in E_G$ et un entier positif $B \geq 2$.
Résultat : Arbre de recouvrement T de G tel que le degré de tous les sommets de T est inférieur ou égal à B .
Objectif : Minimiser le coût de l'arbre T .

1.4.3.2 Résolution exacte et approchée

Dans [DH68], il est précisé que le problème DCMST est NP-difficile car les problèmes chemin hamiltonien et voyageur du commerce sont des cas particuliers de celui-ci. En effet, si l'on fixe la borne sur le degré B à deux, alors un arbre de recouvrement de poids minimum qui respecte cette borne est bien un chemin hamiltonien. Si de plus le graphe G est complet (s'il ne l'est pas, alors il est complété par des arêtes de poids $+\infty$) alors cet arbre est bien une solution optimale pour le voyageur du commerce. Par contre, pour $B = |V_G| - 1$, tout arbre de poids minimum respecte la contrainte sur le degré et est calculable en temps polynomial. Par la suite, Garey et Johnson montrent que le problème DCMST est NP-difficile pour toute valeur fixée de $2 \leq B < |V_G| - 1$ [GJ79]. Dans le cas où la borne sur le degré n'est pas homogène (elle est spécifique à chaque sommet v du graphe et elle est notée B_v), il est mentionné dans [Cie00] que s'il existe un et un seul sommet v du graphe pour lequel $B_v < |V_G| - 1$ alors le problème DCMST est solvable en temps polynomial. En effet, dans ce cas le problème peut être réduit linéairement au problème de recherche d'arbre de recouvrement de coût minimum [GT84].

Deo et Hakimi, dans [DH68], proposent un premier schéma de résolution basé sur la programmation linéaire. Le problème est formulé en programme linéaire en nombres entiers mais la contrainte de connexité est relâchée. La méthode du simplexe est appliquée pour résoudre le PL. Si le graphe résultant est connexe alors il s'agit bien d'une solution optimale. Sinon, un algorithme glouton est appliqué pour trouver une solution approchée (si elle existe).

La première formulation de ce problème en programme linéaire en nombres entiers a été proposée par Narula et al. dans [NC80]. Soit $G = (V_G, E_G)$ un graphe complet arête pondéré et des variables $x_{i,j}$ égales à 1 si l'arête $e_{i,j} \in E_G$ est incluse dans l'arbre de recouvrement et à 0 sinon. Le programme linéaire se présente comme suit :

$$\text{Minimiser } \sum_{\substack{i,j \in V_G \\ i \neq j}} c_{i,j} x_{i,j} \quad (1.35)$$

$$\sum_{\substack{j \in V_G \\ i \neq j}} x_{i,j} \leq b \quad \forall i \in V_G \quad (1.36)$$

$$\sum_{\substack{j \in V_G \\ i \neq j}} x_{i,j} \leq 1 \quad \forall i \in V_G \quad (1.37)$$

$$\sum_{i,j \in N} x_{i,j} \leq |N| - 1 \quad \forall N \subset V_G \quad (1.38)$$

$$x_{i,j} \in \{0, 1\} \quad i, j \in V_G \quad (1.39)$$

Le problème DCMST est non-approximable à un facteur constant du coût de l'arbre optimal [RMR⁺93a] (il n'est pas dans la classe APX). Les efforts ont donc été orientés vers l'élaboration d'algorithmes d'approximation bi-critères. Soit opt le coût de la solution optimale du problème DCMST. Un algorithme d'approximation $(\alpha, f(B_v))$ -approché est un algorithme qui retourne un arbre de recouvrement de coût au plus $\alpha * opt$ et tel que $d_T(v) \leq f(B_v)$ pour tout $v \in V_T$. Ravi et al. [RMR⁺93a] donnent un algorithme d'approximation $(\log \frac{|N|}{B}, B * \log \frac{|V_G|}{B})$ -approché basé sur la technique de la comparaison améliorante (démarrer d'une solution non réalisable (non connexe) et appliquer une procédure de sélection d'arêtes avec un coût borné pour construire une solution réalisable). Konemann et Ravi [KR00] obtiennent une $(O(1), B_v + \log |V_G|)$ approximation en utilisant une approche basée sur la relaxation Lagrangienne et un algorithme initialement proposé dans [FR94] qui donne une solution approchée au problème DCST de $opt + 1$ où opt est le degré maximum de tous les arbres de recouvrement du graphe. Dans [RS06], Ravi et Singh considèrent une variante du problème pour lequel l'arbre obtenu doit être un arbre de coût minimum et donnent un algorithme qui retourne un MST dans lequel le degré de chaque sommet v est au plus $B_v + p$, où p est le nombre de coûts distincts de n'importe quel MST du graphe.

En 1991, Geomans formule la conjecture suivante : en temps polynomial, on peut trouver un arbre de recouvrement de coût opt dans lequel le degré de chaque sommet est au plus $B + 1$, où opt est le coût minimum de tous les arbres de recouvrement de degré maximum B . Dans le cas des graphes arêtes pondérées avec un coût unitaire, Furer et Raghavachari [RMR⁺93b], en se basant sur les travaux de [Win89], proposent un algorithme qui retourne un arbre de recouvrement dans lequel le degré de chaque sommet v est au plus $B_v + 1$ où bien un certificat de non existence de solution réalisable. La conjecture reste cependant ouverte dans le cas général. En 2006, Geomans présente dans [Goe06] un algorithme d'approximation $(1, B_v + 2)$ approchée pour le problème DCMST en utilisant une méthode d'intersection de matroïdes. Il faut attendre 2007 pour voir cette conjecture confirmée. En effet, Singh et al. proposent dans [SL07] un algorithme d'approximation $(1, B_v + 1)$ -approchée pour le problème DCMST. Cet algorithme utilise la méthode de la relaxation itérative comme introduit dans [Jai01]. L'idée fondamentale est que si l'algorithme ne peut trouver une arête à sélectionner, alors il peut retirer ou relaxer une des contraintes sur le degré. Le cœur de l'analyse est de montrer qu'un sommet ayant un degré au plus égal à $B_v + 1$ existe toujours. Ceci est prouvé en s'appuyant sur le fait qu'un point extrême

est caractérisé par une laminar family de contraintes strictes [Goe06]. La condition par laquelle un sommet est sélectionné assure que la contrainte sur le degré ne peut être violée que d'au plus un.

Quelques algorithmes d'approximation sans garantie de performance ont été proposés dans la littérature. Dans [NC80], un algorithme primal, un algorithme dual, ainsi qu'une méthode branch-and-bound sont proposés pour résoudre de manière approchée le problème DCMST. L'algorithme primal vise à démarrer d'une solution réalisable, qui est considérée comme une borne supérieure au problème, et de l'améliorer de façon gloutonne jusqu'à converger vers une solution qui s'approche de l'optimale tout en maintenant la réalisabilité à chaque étape. Pour leur algorithme primal, Narula et al. ont donc commencé par construire un arbre qui respecte la borne sur le degré (arbre réalisable DCST), ceci en appliquant l'algorithme de Prim [Pri57] modifié de façon à prendre en considération la contrainte sur le degré à chaque itération. Une fois l'arbre réalisable DCST (qui n'est pas forcément un DCMST) construit, un algorithme vérifiant si chaque arête de l'arbre peut être remplacée par une autre arête de plus faible coût en respectant la connexité et la contrainte sur le degré est appliqué. L'algorithme dual quant à lui vise à démarrer d'une borne inférieure non réalisable et à modifier cette borne de sorte qu'elle respecte la contrainte sur le degré. L'algorithme dual proposé dans [NC80] considère le coût de l'arbre de coût minimum comme borne inférieure pour le problème DCMST. Si l'arbre de coût minimum ne respecte pas la contrainte sur le degré des sommets, alors un algorithme glouton basé sur la supposition que beaucoup d'arêtes dont les deux extrémités ne sont pas des sommets de branchement doivent figurer dans le DCMST. Ainsi, les autres arêtes sont remplacées pour obtenir une solution réalisable. L'algorithme de branch-and-bound utilise le coût de l'arbre de coût minimum comme borne inférieure et le coût d'un arbre qui respecte la contrainte sur le degré comme borne supérieure. La procédure de branchement est une réadaptation de la méthode présentée dans [HK71] pour le problème voyageur du commerce.

1.5 Conclusion

Le travail que nous développons dans le cadre de cette thèse s'articule autour des problèmes de recherche de structure de recouvrement d'un réseau modélisé par un graphe sous contrainte sur le degré des sommets. L'arbre étant la structure connexe permettant de couvrir les sommets en utilisant un minimum de liens, il est le plus utilisé pour ce type de problèmes. Afin de valoriser nos travaux axés sur la structure de hiérarchie de recouvrement que nous présentons dans les chapitres suivants, nous avons mentionné ici les travaux effectués sur quelques problèmes de recherche d'arbre de recouvrement d'un graphe sous contrainte sur le degré des sommets ainsi que leur intérêt pratique. Afin de mesurer de manière concrète la plus-value apportée par la hiérarchie de recouvrement, nous comparons les deux structures. Cette comparaison est principalement basée sur les résultats préalables obtenus sur les arbres de recouvrement.

Recouvrement total avec une borne supérieure sur le degré des sommets

2.1 Introduction

Les problèmes de couverture de graphe ont toujours été au centre de la recherche opérationnelle ; les problèmes Vertex Cover et Set Cover étant les plus connus. L'introduction du routage multicast/broadcast dans les années 90 par Deering [Dee92] a changé l'orientation d'étude de ce genre de problèmes. En effet, pour permettre le routage dans un réseau, il ne suffit pas de couvrir les sommets du graphe avec un ensemble d'arêtes quelconques. Il faut également que le graphe formé par cet ensemble de sommets et d'arêtes soit connexe. L'arbre de recouvrement répond à cette problématique lorsque la solution est un sous-graphe. Lorsqu'il est également utile de minimiser le coût des liens participant au recouvrement, l'arbre de recouvrement de coût minimum est alors une solution. Des motivations situées dans différents types de réseaux de communications (voir chapitre 1) ont incité les chercheurs à étudier le problème de recherche d'arbre de recouvrement de coût minimum avec une borne supérieure sur le degré des sommets. Malgré l'intérêt particulier apporté à ce problème NP-difficile, la résolution exacte de celui-ci n'a pas été suffisamment explorée. Nous proposons, dans ce chapitre une nouvelle formulation en programme linéaire en nombres entiers de ce problème. Nous proposons dans un second temps, de reconsidérer ce problème classique en utilisant la structure de hiérarchie de recouvrement qui, de par sa flexibilité, permet un passage multiple par les sommets. Nous proposons une analyse de ce problème. La complexité et la résolution exacte et approchée sont ainsi abordées. Enfin, une partie expérimentation complète cette analyse théorique et offre une vision sur la plus-value apportée par la hiérarchie de recouvrement par rapport aux solutions classiques (arbre).

2.2 Arbre de recouvrement sous contrainte sur le degré des sommets

Deo et Hakimi ont été les premiers à identifier l'utilité de borner le degré des sommets d'un arbre de recouvrement d'un graphe [DH68] en évoquant la nécessité de réduire le nombre de liens reliés à chaque point dans les réseaux de communication, car les concentrateurs (hubs) ont une capacité limitée. L'introduction du multicast au début des années quatre-vingt-dix a augmenté de manière considérable l'étude des problèmes de recouvrement de graphe sous contrainte sur le degré des sommets. Le problème Degree Constrained Minimum Spanning Tree (DCMST) est défini dans le chapitre 1, section 1.4.3.1, page 39.

Les figures 2.1b, 2.1c, 2.1d, 2.1e constituent l'ensemble des arbres de recouvrement du graphe G (figure 2.1a). Si l'on fixe la borne sur le degré B à 4, alors les quatre arbres de recouvrement respectent cette contrainte. Si de plus, on pose un coût identique pour toutes les arêtes du graphe G , alors ces arbres de recouvrement sont tous optimaux pour le problème DCMST (pour tout $4 \leq B$).

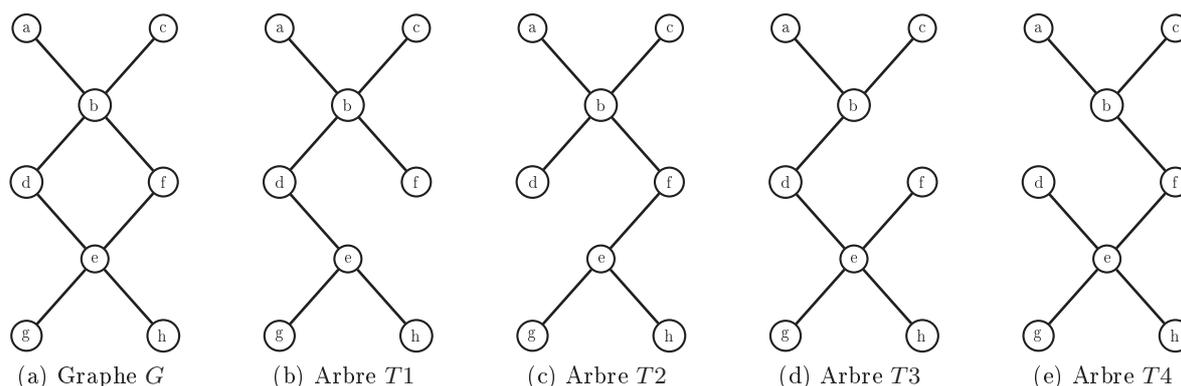


Figure 2.1 – Arbres de recouvrement de G respectant la contrainte sur le degré $B = 4$

Il s'est avéré, après l'étude de l'état de l'art, que le volet résolution approchée du problème NP-difficile [GJ79] DCMST ait été intensivement étudié, au détriment du volet résolution exacte de celui-ci. En effet, seule une formulation mathématique du problème DCMST en programme linéaire en nombres entiers a été proposée dans [NC80]. Pour un graphe G de taille n , ce programme génère $n(n-1)/2$ variables et $2^n + n - 1$ contraintes, ce qui permet seulement la résolution de très petites instances. Afin de garantir la connexité de l'arbre de recouvrement, chaque sous-graphe induit de taille n_i de l'arbre de recouvrement doit ainsi avoir exactement $|n_i| - 1$ arêtes. Le graphe G contient $2^{|V_G|}$ sous-graphes induits, ce qui implique $2^{|V_G|}$ contraintes nécessaires pour assurer la connectivité.

Ces dernières années, la formulation linéaire en nombres entiers se base davantage sur le principe de flot afin de garantir la connexité des arbres de recouvrement [BJS09]. Celui-ci permet de réduire considérablement le nombre de contraintes.

2.2.1 Une nouvelle formulation PLNE du problème DCMST

Nous proposons ici une nouvelle formulation du problème DCMST basée sur le principe de flot. Ce principe a notamment été utilisé dans la formulation PLNE du problème de recherche d'arbre de recouvrement ayant un minimum de sommets de branchement [CGI09]. Ainsi, la connexité de l'arbre de recouvrement est garantie grâce à l'envoi d'une unité de flot depuis un sommet « source » choisi de manière arbitraire s vers chacun des autres sommets du graphe. Les arêtes faisant transiter le flot sont sélectionnées pour appartenir à l'arbre de recouvrement. Des contraintes sur le degré des sommets sont ajoutées afin que tous les sommets de l'arbre de recouvrement respectent la borne sur le degré B . Parmi tous les arbres de recouvrement qui respectent cette contrainte, un arbre ayant un coût minimum sera retourné à l'issue de la résolution. Remarquons que le programme linéaire n'est pas réalisable lorsqu'aucun arbre de recouvrement ne respecte la contrainte sur le degré. La formulation de flot nécessite le passage à un concept orienté. Ainsi le graphe G est transformé en un graphe orienté où chaque arête de E_G est représentée par deux arcs orientés dans des sens opposés ayant chacun un coût égal à celui de l'arête.

Les paramètres du graphe :

- $In(u)$: L'ensemble des sommets prédécesseurs du sommet u .
- $Out(u)$: L'ensemble des sommets successeurs du sommet u .
- $C(u, v)$: Le coût de l'arc allant de u vers v .

Les variables en nombres entiers :

- $L(u, v)$: Variable binaire. Elle est égale à 1 si l'arc (u, v) se trouve dans l'arbre de recouvrement, 0 sinon.
- $F(u, v)$: Variable de flot. Représente le nombre entier d'unités de flot transitant sur l'arc (u, v) .

L'objectif est de minimiser le coût total des arêtes. La fonction objective peut être exprimée comme suit :

$$\text{Minimiser : } \sum_{u \in V_G} \sum_{v \in Out(u)} L(u, v) \cdot C(u, v) \quad (2.1)$$

La solution doit respecter un certain nombre de contraintes de degré et de connexité qui sont listées comme suit :

Contraintes de degré :

$$\sum_{v \in \text{Out}(s)} L(s, v) \leq B \quad (2.2)$$

$$\sum_{v \in \text{In}(s)} L(v, s) = 0 \quad (2.3)$$

La contrainte 2.2 assure que la source s a au plus B successeurs. La contrainte 2.3 garantit que la source n'a aucun prédécesseur.

$$\sum_{v \in \text{In}(u)} L(v, u) = 1 \quad \forall u \in V_G - \{s\} \quad (2.4)$$

La contrainte 2.4 assure que chaque sommet, excepté la source, a un seul prédécesseur.

$$\sum_{v \in \text{out}(u)} L(u, v) \leq (B - 1) \quad \forall u \in V_G \setminus \{s\} \quad (2.5)$$

La contrainte 2.5 assure que chaque sommet $u \in V_G \setminus \{s\}$ peut avoir au plus $B - 1$ successeurs. Ainsi, le nombre de successeurs additionné au seul prédécesseur d'un sommet est bien inférieur ou égal à B .

$$\sum_{(v,u) \in E_G} L(v, u) = |V_G| - 1 \quad \forall u, v \in V_G \quad (2.6)$$

La contrainte 2.6 assure que le nombre d'arêtes de l'arbre de recouvrement est égal à $|V_G| - 1$.

Contraintes de connexité :

Les contraintes précédentes ne suffisent pas à assurer la connexité de l'arbre de recouvrement. En effet, les deux graphes des figures 2.2.b et 2.2.c respectent toutes les contraintes précédentes mais ils ne sont pas pour autant connexes. Afin de garantir la connexité de l'arbre de recouvrement, des contraintes à base de flot sont introduites :

$$\sum_{v \in \text{Out}(s)} F(s, v) = |V_G| - 1 \quad (2.7)$$

$$\sum_{v \in \text{In}(s)} F(v, s) = 0 \quad (2.8)$$

Les contraintes de flot 2.7 et 2.8 assurent que la source envoie $|V_G| - 1$ unités de flot et qu'elle ne reçoit aucune unité de flot.

$$\sum_{v \in \text{In}(u)} F(v, u) = \sum_{v \in \text{Out}(u)} F(u, v) + 1 \quad \forall u \in V_G - \{s\} \quad (2.9)$$

La contrainte 2.9 assure que chaque sommet $u \in V_G \setminus \{s\}$ consomme une seule unité de flot et

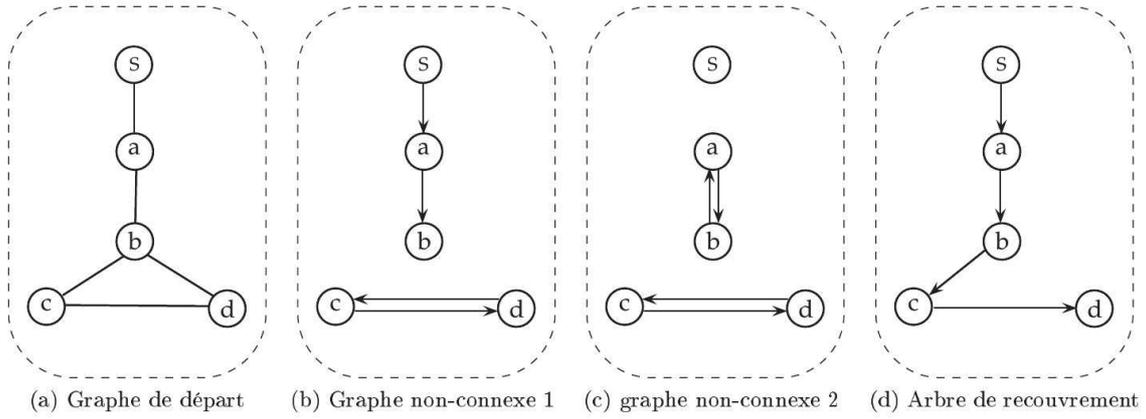


Figure 2.2 – Exemple illustrant la nécessité de garantir la connectivité

envoie les unités de flot restantes, si elles existent, à ses successeurs. Cette contrainte garantit que chaque sommet est effectivement connecté à la source.

$$F(u, v) \geq L(u, v) \quad \forall u, v \in V_G \quad (2.10)$$

$$F(u, v) \leq (|V_G| - 1) * L(u, v) \quad \forall u, v \in V_G \quad (2.11)$$

Les contraintes 2.10 et 2.11 permettent de lier les variables $F(u, v)$ et $L(u, v)$. Elles assurent ainsi que chaque arc (u, v) transportant une quantité non-nulle de flot est sélectionné pour être dans l'arbre de recouvrement ($F(u, v) > 0 \Rightarrow L(u, v) = 1$) et que chaque arc sélectionné pour être dans l'arbre transporte bien une quantité non-nulle de flot ($L(u, v) = 1 \Rightarrow F(u, v) > 0$).

2.3 Hiérarchie de recouvrement sous contrainte sur le degré des sommets

Il n'existe pas toujours de solution réalisable pour le problème DCMST, particulièrement si la borne sur le degré B est petite. En effet, reconsidérons le graphe de la figure 2.1, qui contient quatre arbres de recouvrement distincts. Certes, ils respectent la contrainte sur le degré si celle-ci est fixée à 4, mais aucun de ces arbres ne respecte cette borne si celle-ci est fixée à $B = 3$ (chaque arbre contient un sommet de degré 4). Plus généralement, quelle que soit la valeur de la borne B , on peut construire des graphes dans lesquels il n'existe pas de solution réalisable pour le problème DCMST. Il suffit, par exemple, de considérer une étoile avec un sommet central de degré $B + 1$ (figure 2.3). La résolution du programme linéaire en nombres entiers sur un jeu de graphes aléatoires a montré que le pourcentage d'instances non-réalisables est important quand la contrainte sur le degré est forte. La contrainte sur le degré peut exprimer une action qui est réalisée ou un budget qui est réhabilité à chaque visite du sommet. En effet, les switchs OXC déployés dans les réseaux optiques ont une capacité limitée de division du signal lumineux entrant (borné, par exemple, par une valeur m). À chaque passage du signal par un switch, cette

capacité est renouvelée et le signal peut ainsi être de nouveau divisé par m .

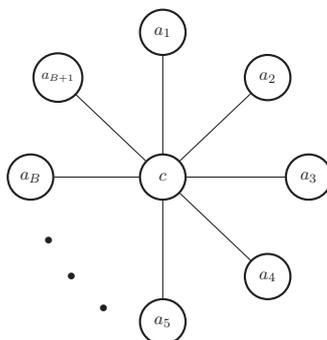


Figure 2.3 – Instance non satisfiable pour le problème DCMST quelle que soit la valeur de B .

Les sommets d'un arbre de recouvrement sont visités une et une seule fois. Ceci ne permet pas de prendre en compte la possibilité d'un passage multiple par un sommet. Afin de prendre en considération cette borne sur le degré effective à chaque visite d'un sommet, nous utilisons la structure de hiérarchie de recouvrement. Celle-ci permet de passer plusieurs fois par les sommets du graphe couvert. Nous introduisons le problème Degree Constrained Minimum Spanning Hierarchy (DCMSH) que nous définissons comme suit :

Problème : Degree Constrained Minimum Spanning Hierarchy problem - DCMSH
Données : Graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$, une fonction $C : E_G \rightarrow \mathbb{R}_*^+$ associant un coût $c(e)$ à chaque arête $e \in E_G$ et un entier positif $B \geq 2$
Résultat : Hiérarchie de recouvrement H de G telle que le degré de tous les sommets de H est inférieur ou égal à B .
Objectif : Minimiser le coût de la hiérarchie H .

Si un arbre de recouvrement de coût minimum T d'un graphe G respecte la contrainte sur le degré B , alors cet arbre est une solution optimale pour le problème DCMST. Cet arbre est-il également optimal pour le problème DCMSH ? Peut-il exister une hiérarchie de recouvrement d'un graphe avec un coût strictement inférieur à celui de l'arbre de recouvrement de coût minimum ? Un arbre de recouvrement optimal pour le problème DCMST est-il nécessairement une solution optimale pour le problème DCMSH ? Existe-t-il une solution réalisable pour le problème DCMSH quelle que soit l'instance ?

Lemme 2.3.1. *Pour tout graphe connexe G et pour toute borne $B \geq 2$, le problème DCMSH a une solution réalisable.*

Démonstration. Un parcours est une hiérarchie de recouvrement particulière, dans laquelle le degré de chaque occurrence des sommets est au plus égal à deux. Sachant qu'un graphe connexe

admet toujours un parcours, il existe toujours une hiérarchie de recouvrement du graphe qui respecte la contrainte sur le degré $B = 2$. Cette solution respecte donc toutes les bornes supérieures ou égales à 2. \square

Lemme 2.3.2. *S'il existe un arbre de recouvrement de coût minimum d'un graphe G respectant la contrainte sur le degré, c'est une solution optimale pour le problème DCMSH et toutes les solutions optimales sont des arbres de G .*

Démonstration. La structure de recouvrement de coût minimum qui connecte tous les sommets du graphe sans aucune contrainte sur le degré est l'arbre de recouvrement de coût minimum (Minimum Spanning Tree - MST), car il ne contient aucune redondance (tous les sommets de G y figurent une seule fois et ne contient aucune arête pouvant être supprimée ou remplacée). Donc, si l'un des MST du graphe respecte la contrainte sur le degré alors il est une solution optimale pour le problème DCMST et également pour le problème DCMSH.

Supposons maintenant qu'en plus de contenir un MST (notons le T^*), l'ensemble des solutions optimales du problème DCMSH contient également une hiérarchie $H = (Q_H, h_H, G)$ ayant au moins un sommet visité deux fois. Le coût $c(H)$ de la hiérarchie optimale doit être égal au coût $c(T^*)$ de l'arbre de coût minimum. Trivialement, le coût d'une hiérarchie est supérieur ou égal au coût de son image dans G : $c(I) \leq c(H)$, où I est l'image (le sous-graphe généré par H dans G). La hiérarchie H contient alors au moins un cycle (un sommet dupliqué crée un cycle). L'image I couvre tous les sommets de V_G . Il existe deux possibilités :

1. L'image I est un arbre et son coût est borné inférieurement par le coût de T^* : $c(T^*) \leq c(I)$. Dans ce cas, il existe au moins une arête dupliquée dans H (rappelons que H n'est pas un arbre), donc $c(I) < c(H)$. Finalement, nous obtenons $c(T^*) < c(H)$. Par conséquent, la hiérarchie H ne peut pas être optimale.
2. L'image I n'est pas un arbre. En éliminant les arêtes redondantes, nous obtenons un arbre T' de recouvrement de G . Nous avons alors $c(T') < c(I)$ (les arêtes ont un coût strictement positif) et $c(I) < c(H)$. Finalement, $c(T') < c(T^*) < c(H)$.

\square

Le problème DCMST est NP-difficile car la recherche d'un chemin hamiltonien est un cas particulier de ce dernier (si $B = 2$ et que toutes les arêtes du graphe ont un coût identique). Cependant, si la borne B est fixée à deux, la solution optimale du problème DCMSH n'est pas nécessairement un chemin hamiltonien. En effet, il existe une solution pour ce problème même si le graphe n'est pas hamiltonien (proposition 2.3.1). Une question s'impose alors : Le problème DCMSH est-il NP-difficile ?

Théorème 2.3.3. *Le problème DCMSH est NP-difficile pour tout $B \geq 2$.*

Démonstration. Soit $G = (V_G, E_G)$ un graphe connexe arête pondéré tel que $c(e) = 1, \forall e \in E_G$. Soit $G' = (V'_G, E'_G)$ le graphe obtenu en ajoutant $B - 2$ sommets de degré 1 à chaque sommet

$v \in V_G$ (voir figure 2.4) . Nous avons alors $|V'_G| = |V_G| + |V_G| \cdot (B - 2) = (B - 1) \cdot |V_G|$. Tout arbre de recouvrement de G' a un coût égal à $(B - 1) \cdot |V_G| - 1$. Il existe une hiérarchie de recouvrement du graphe G' qui respecte la contrainte sur le degré B de coût $(B - 1) \cdot |V_G| - 1$ si et seulement si le graphe G contient un chemin Hamiltonien (rappelons que ce chemin hamiltonien contient $|V_G| - 1$ arêtes).

Supposons qu'il existe une hiérarchie de recouvrement H du graphe G' de coût $(B - 1) \cdot |V_G| - 1$. Étant donné son coût, la hiérarchie H est nécessairement un arbre. En effet, s'il existait au moins une arête dupliquée dans H alors le coût de H serait au moins égal à $(B - 1) \cdot |V_G|$. Si l'on retire de H les $(B - 2) \cdot |V_G|$ sommets qui appartiennent à $V'_G \setminus V_G$, nous obtenons alors un sous-graphe connexe dans lequel tous les sommets ont un degré inférieur ou égal à deux, qui est donc un chemin hamiltonien de G .

Réciproquement, en reliant $B - 2$ feuilles à chaque sommet d'un chemin hamiltonien de G , nous obtenons un arbre qui satisfait la contrainte sur le degré B et qui est, en considérant le lemme 2.3.2, une solution optimale pour le problème DCMSH dans G' . \square

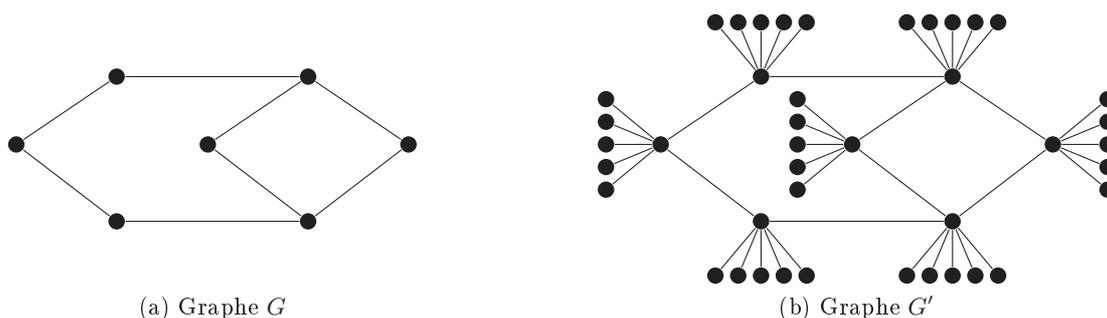


Figure 2.4 – Réduction du problème DCMSH au problème chemin hamiltonien ($B = 7$)

Nous pouvons remarquer une différence dans la difficulté de résolution entre le problème DCMST et le problème DCMSH. La difficulté de résolution (exacte) du problème DCMST réside dans la contrainte sur le degré. En effet, trouver une solution réalisable quelle que soit la valeur de B est NP-difficile. Dans la preuve du théorème 2.3.3, on peut constater que le graphe G' admet une solution réalisable pour le problème DCMST si et seulement si le graphe G admet un chemin hamiltonien. Par contre, trouver une solution réalisable pour le problème DCMSH est trivial (proposition 2.3.1). La difficulté du problème DCMSH réside dans la minimisation du coût. En effet, la condition de trouver une solution dans G' de coût exactement $(B - 1) \cdot |V_G| - 1$ nécessite que le graphe G soit hamiltonien.

2.3.1 Propriétés des solutions optimales du problème DCMSH

Afin de réduire le nombre de variables et de contraintes du programme linéaire en nombres entiers modélisant le problème DCMSH que nous allons proposer, il est nécessaire de borner le

nombre de duplications des sommets dans la solution optimale.

Notations Pour tout graphe $G = (V_G, E_G)$, on note $\lambda_m(G)$ le nombre de sommets de degré m dans G . Soit $H = (Q_H, h_H, G)$ une hiérarchie de recouvrement du graphe G , on note $\theta(v)$ l'ensemble des occurrences du sommet v dans H . Une hiérarchie optimale pour le problème DCMSH est notée H^* . Nous commençons par donner une borne sur le nombre de feuilles dans un arbre de recouvrement en fonction du nombre de sommets de degré m que cet arbre peut contenir.

Lemme 2.3.4. *Pour tout arbre $T = (V_T, E_T)$ tel que $|V_T| \geq 2$, $\lambda_m(T) \geq \alpha \Rightarrow \lambda_1(T) > \alpha * (m - 2) + 2$.*

Démonstration. Par induction :

1. Le cas de base : Trivialement vrai pour $\alpha = 0$ et $|V_T| \geq 2$ puisque tout arbre avec au moins 2 sommets a au moins 2 feuilles.
2. Supposons que la proposition soit vraie pour tout $\alpha' < \alpha$. Soit $T = (V_T, E_T)$ un arbre avec $\lambda_m(T) = \alpha$. Soit $v \in V_T$ un sommet tel que $d_T(v) = m$ avec $\{s_1, s_2, s_3, \dots, s_m\} \in V_T$ ses voisins. On note $F = (T_1, \dots, T_m)$ la forêt obtenue en supprimant le sommet v de l'arbre T et en ajoutant un sommet s'_i et une arête $\{s_i, s'_i\}$ à chaque composante connexe C_i . Soit α_i le nombre de sommets de degré m de chaque composante connexe G_i .

$$\begin{aligned}
\lambda_1(T) &\geq \sum_i^m \lambda_1(T_i) - m \\
&\geq \sum_i^m (\alpha_i * (m - 2) + 2) - m \\
&\geq \sum_i^m (\alpha_i) * (m - 2) + m \\
&\geq (\alpha - 1) * (m - 2) + m \\
&\geq \alpha * (m - 2) + 2
\end{aligned}$$

□

Lemme 2.3.5. *Pour toute hiérarchie $H^* = (Q_{H^*}, h_{H^*}, G)$ optimale pour le problème DCMSH on a : $\forall v \in V_G, \sum_{i=1}^{|\theta(v)|} d_{Q_{H^*}}(v^i) > B * (|\theta(v)| - 1) + 1$.*

Démonstration. Nous procédons par contradiction. Supposons qu'il existe une hiérarchie optimale $H^* = (Q_{H^*}, h_{H^*}, G)$ et un sommet $v \in V_G$ tel que $\sum_{i=1}^{|\theta(v)|} d_{Q_{H^*}}(v^i) \leq B * (|\theta(v)| - 1) + 1$. Dans ce cas, le nombre d'arêtes qui pourraient être ajoutées aux $|\theta(v)|$ occurrences sans violer la contrainte sur le degré est supérieur ou égal à $B - 1$. Orientons la hiérarchie H^* . Pour cela, choisissons une racine arbitraire $r \notin \theta(v)$ et orientons Q_{H^*} de r jusqu'aux feuilles. Il est possible de supprimer une occurrence v^i de v et relier les (au plus $B - 1$) successeurs de v^i aux $|\theta(v)| - 1$

autres occurrences de v sans dépasser la borne sur le degré B . Après suppression de l'orientation des arcs, nous obtenons une hiérarchie réalisable H' avec un coût strictement inférieur à celui de H^* (H' a au moins une arête de moins que H^*). Ceci est absurde. \square

Lemme 2.3.6. *Soit $G = (V_G, E_G)$ un graphe connexe. Il existe une hiérarchie optimale $H^* = (Q_{H^*}, h_{H^*}, G)$ telle que pour chaque $v \in V_G$ ayant un degré strictement supérieur à B et un nombre d'occurrences strictement supérieur à 1, il existe au moins $|\theta(v)| - 1$ occurrences de v avec un degré égal à B dans Q_{H^*} .*

Démonstration. Nous prouvons ici qu'il est possible de transformer toute hiérarchie H^{**} optimale pour le problème DCMSH en une hiérarchie H^* également optimale et qui respecte les conditions de la proposition. Soit $H^{**} = (Q_{H^{**}}, h_{H^{**}}, G)$ une hiérarchie optimale pour le problème DCMSH et v^i une occurrence arbitraire du sommet v . La Proposition 2.3.5 implique que :

$$\begin{aligned} B * (|\theta(v)| - 1) + 1 &< \sum_{j=1, j \neq i}^{|\theta(v)|} d_{Q_{H^{**}}}(v^j) + d_{Q_{H^{**}}}(v^i) \\ B * (|\theta(v)| - 1) - \sum_{j=1, j \neq i}^{|\theta(v)|} d_{Q_{H^{**}}}(v^j) &< d_{Q_{H^{**}}}(v^i) - 1 \\ &\leq d_{Q_{H^{**}}}(v^i) - 2 \end{aligned}$$

On choisit une racine arbitraire $r \notin \theta(v)$. On oriente $Q_{H^{**}}$ de r vers les feuilles. Puisque H^{**} est optimale, v^i ne peut pas être une feuille (sinon celle-ci pourrait être supprimée, ce qui diminuerait la coût de la solution), donc $d_{Q_{H^{**}}}(v^i) \geq 2$. On peut déconnecter $B * (|\theta(v)| - 1) - \sum_{j=1, j \neq i}^{|\theta(v)|} d_{Q_{H^{**}}}(v^j)$ successeurs de v^i et les reconnecter à d'autres occurrences de v sans dépasser la borne sur le degré. La hiérarchie H^* obtenue satisfait les conditions de la proposition. \square

Théorème 2.3.7. *Pour toute hiérarchie optimale $H^* = (Q_{H^*}, h_{H^*}, G)$ avec $B > 2$ et pour tout sommet $v \in V_G$, le nombre d'occurrences de v dans H^* est inférieur ou égal à $\frac{|V_G| - 2}{B - 2}$.*

Démonstration. Nous procédons par contradiction. Nous supposons que dans une hiérarchie optimale $H^* = (Q_{H^*}, h_{H^*}, G)$ il existe un sommet $v \in V_G$ pour lequel le nombre d'occurrence soit strictement supérieur à $\frac{|V_G| - 2}{B - 2}$. En suivant le même raisonnement que dans la démonstration de la proposition 2.3.6, H^* peut être transformée en une hiérarchie $H^{**} = (Q_{H^{**}}, h_{H^{**}}, G)$ ayant les mêmes occurrences de sommets mais telle qu'il existe au moins $|\theta(v) - 1|$ occurrences de v avec un degré égal à B . Nous avons donc :

$$\begin{aligned} \lambda_B(Q_{H^{**}}) &\geq |\theta(v)| - 1 \\ &> \frac{|V_G| - 2}{B - 2} - 1 \\ &\geq \frac{|V_G| - 2}{B - 2} \end{aligned}$$

En utilisant la proposition 2.3.4, on obtient :

$$\begin{aligned}\lambda_1(Q_{H^{**}}) &\geq \left(\frac{|V_G| - 2}{B - 2}\right) * (B - 2) + 2 \\ &\geq |V_G| \quad (\text{Absurde})\end{aligned}$$

□

Théorème 2.3.8. *Dans toute hiérarchie $H^* = (Q_{H^*}, h_{H^*}, G)$ optimale pour le problème DCMSH on a : $|V_Q| < \frac{|V_G| - 2}{B - 2} + |V_G| - 1$.*

Démonstration. Soit $H^* = (Q_{H^*}, h_{H^*}, G)$ une hiérarchie optimale pour le problème DCMSH. En suivant le même raisonnement que dans la démonstration de la proposition 2.3.6, H^* peut être transformée en une hiérarchie $H^{**} = (Q_{H^{**}}, h_{H^{**}}, G)$ ayant les mêmes occurrences de sommets mais tel qu'il existe au moins $|\theta(v) - 1|$ occurrences de v avec un degré égal à B . Nous avons donc :

$$\begin{aligned}\lambda_B(Q_{H^{**}}) &\geq \sum_{v \in V_G} (|\theta(v)| - 1) + 1 \\ &\geq \sum_{v \in V_G} |\theta(v)| - |V_G| + 1 \\ |V_G| \geq \lambda_1(Q_{H^{**}}) &\geq \left(\sum_{v \in V_G} |\theta(v)| - |V_G| + 1\right)(B - 2) + 2 \\ \frac{|V_G| - 2}{B - 2} &> \sum_{v \in V_G} |\theta(v)| - |V_G| + 1 \\ \frac{|V_G| - 2}{B - 2} + |V_G| - 1 &> \sum_{v \in V_G} |\theta(v)|\end{aligned}$$

□

Théorème 2.3.9. *Dans toute hiérarchie optimale pour le problème DCMSH avec $B = 2$, le nombre d'occurrences de chaque arête est inférieur ou égal à 2*

Démonstration. Soit $H^* = (Q_{H^*}, h_{H^*}, G)$ une hiérarchie optimale pour le problème DCMSH avec $B = 2$. Nous supposons qu'il existe une arête $\{a, b\}$ avec au moins 3 occurrences dans H . Puisque $B = 2$ alors H^* est un parcours (éventuellement non-élémentaire). Orientons arbitrairement H^* . Soient $\{a^1, b^1\}$, $\{a^2, b^2\}$, et $\{a^3, b^3\}$ les trois premières occurrences de l'arête $\{a, b\}$ dans le parcours orienté. Supposons sans perte de généralité que l'arête $\{a^1, b^1\}$ soit orientée de a^1 vers b^1 . quatre cas sont possibles :

1. Les arcs (a^1, b^1) , (a^2, b^2) , et (a^3, b^3) sont tous orientés de a^i vers b^i . Le parcours orienté est $W_1(a^1, b^1)W_2(a^2, b^2)W_3(a^3, b^3)W_4$, où W_i sont des sous parcours potentiellement vides.
Soit $H' = W_1(a^1, b^1)W_2\overline{W_3}W_4$ où $\overline{W_3}$ est le parcours W_3 orienté dans le sens opposé.

2. Les arcs (a^1, b^1) et (a^2, b^2) sont orientés de a^i vers b^i et (a^3, b^3) est orienté de b^3 vers a^3 .
Le parcours orienté est $W_1(a^1, b^1)W_2(a^2, b^2)W_3(b^3, a^3)W_4$. Soit $H' = W_1(a^1, b^1)W_3W_2W_4$.
3. Les arcs (a^1, b^1) , (a^3, b^3) sont orientés de a^i vers b^i et (a^2, b^2) est orienté de b^2 vers a^2 . Le parcours orienté est $W_1(a^1, b^1)W_2(b^2, a^2)W_3(a^3, b^3)W_4$. Soit $H' = W_1W_3(a^1, b^1)W_2W_4$.
4. Les arcs (a^2, b^2) et (a^3, b^3) sont orientés de a^i vers b^i et (a^1, b^1) est orienté de b^1 vers a^1 .
Le parcours orienté est $W_1(b^1, a^1)W_2(a^2, b^2)W_3(a^3, b^3)W_4$. Soit $H' = W_1(b^1, a^1)W_2\overline{W_3}W_4$.

Dans les 4 cas précédents, on peut construire un parcours H' avec un coût strictement inférieur à H^* . Ceci contredit l'optimalité de H^* . \square

2.3.2 Résolution exacte du problème DCMSH

Une méthode de résolution exacte intuitive du problème DCMSH consiste à construire un graphe G' à partir du graphe $G = (V_G, E_G)$ incluant toutes les duplications de sommets et d'arêtes possibles dans toute hiérarchie de recouvrement de G . Il faut alors sélectionner les sommets et arêtes appartenant à la solution optimale. Ainsi, on construit le graphe G' en dupliquant $\frac{|V_G|-2}{B-2}$ fois chaque sommet $v \in V_G$ (théorème 2.3.7) et en reliant chaque duplication $v^1, v^2, \dots, v^{\frac{|V_G|-2}{B-2}}$ de v aux autres duplications de v . Si une arête $\{v, u\} \in E_G$, alors quels que soient i et j supérieurs à 1 et inférieurs à $\frac{|V_G|-2}{B-2}$, $\{v^i, u^j\} \in E_{G'}$. Enfin, pour chaque sommet $v \in V_G$, on crée un sommet $v^* \in V_{G'}$ que l'on appelle sommet « fictif ». Chaque sommet fictif v^* est relié aux duplications de v avec des arêtes de coût nul (voir la figure 2.5). Trouver une hiérarchie de recouvrement du graphe G optimale pour le problème DCMSH revient à trouver un arbre de coût minimum dans G' couvrant tous les sommets fictifs et dont les sommets respectent la contrainte sur le degré B . En effet, pour couvrir un sommet fictif v^* il faut nécessairement passer par au moins une duplication du sommet v , car v^* est uniquement relié aux duplications de v .

Nous pouvons alors envisager de résoudre le programme linéaire en utilisant une méthode de résolution exacte. Cependant, la taille de G' est conséquente. En effet, pour un graphe $G = (V_G, E_G)$ tel que $|V_G| = n$ et $|E_G| = m$, le graphe G' contient $n \cdot \left(\frac{|V_G|-2}{B-2} + 1\right)$ sommets et $\frac{|V_G|-2}{B-2} \cdot m + 2 \left(\frac{|V_G|-2}{2}\right) \cdot m + n \cdot \left(\frac{|V_G|-2}{2} + 1\right)$ arêtes. Ainsi, pour une instance ayant n sommets et m arêtes, le programme linéaire contient $2 \cdot \left(\frac{|V_G|-2}{B-2} \cdot m + 2 \left(\frac{|V_G|-2}{2}\right) \cdot m + n \cdot \left(\frac{|V_G|-2}{2} + 1\right)\right)$ variables. Chaque contrainte du programme linéaire est appliquée aux n^2 sommets. Le nombre très important de contraintes et de variables ($O(n^2m)$) générées pour des instances de taille et de densité élevées rend cette méthode de résolution inefficace.

Nous proposons dans [MDM12] une autre approche moins intuitive mais plus efficace (générant moins de variables et de contraintes). Elle consiste à ne pas dupliquer les sommets mais à dupliquer uniquement les arêtes. Ainsi, nous introduisons un k -graphe G_k correspondant au graphe G tel que chaque arête de G est dupliquée k fois dans le graphe G_k . Nous construisons un programme linéaire en nombres entiers ayant des variables correspondant aux sommets et arêtes du k -graphe et calculant un flot de coût minimum dans G_k . Ce PLNE a des inégalités

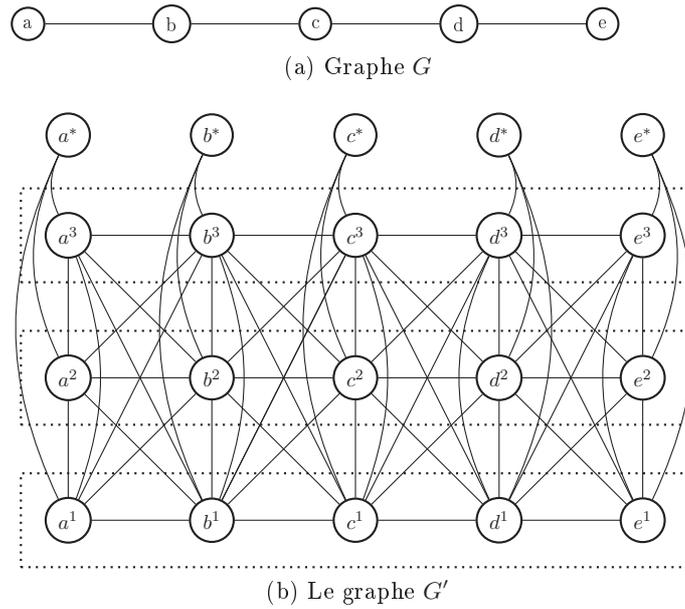


Figure 2.5 – Construction du graphe G' à partir du graphe G ($B = 3$)

valides trouvées grâce à l'étude des propriétés des solutions optimales du problème DCMSH. Le PLNE est résolu en utilisant le solveur CPLEX. Enfin, une hiérarchie optimale est construite à partir des valeurs des variables retournées par le PLNE et à l'aide des propriétés des solutions optimales. La figure 2.6 montre les étapes nécessaires pour la résolution exacte du problème DCMSH.

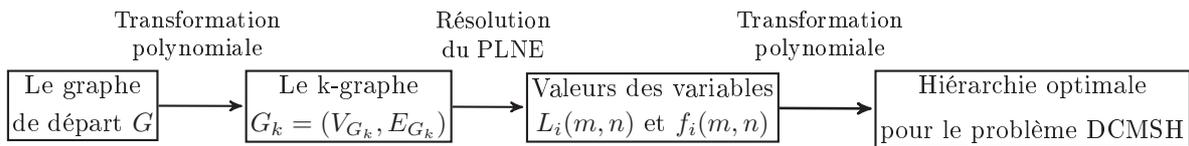


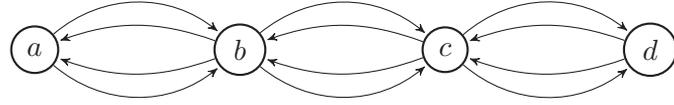
Figure 2.6 – Schéma de résolution exacte du problème DCMSH

2.3.2.1 Première étape : Construction d'un k-graphe

Soit $G = (V_G, E_G)$ un graphe connexe arête pondéré, nous introduisons un k-graphe orienté $G_k = (V_{G_k}, E_{G_k})$ tel que $V_{G_k} = V_G$ et E_{G_k} contient k arcs dans chaque sens correspondant à chaque arête appartenant à E_G . Le nombre de duplications de chaque arête est borné par le résultat du Théorème 2.3.9 si B est égal à 2 et par le résultat du Théorème 2.3.7 sinon (figure 2.7). Les propriétés des contraintes du programme linéaire permettront, par la suite, de déduire le nombre de duplications de chaque sommet à partir des arêtes entrantes et sortantes présentes dans la solution.



(a) Graphe G



(b) Instance d'entrée du PLNE avec $B = 2$ obtenue en modifiant le graphe G

Figure 2.7 – Construction du k -Graphe G_k à partir du graphe G

2.3.2.2 Deuxième étape : Formulation du PLNE

Notre modèle se base sur une formulation de flot. Afin de garantir la connexité, une unité de flot est envoyée d'une source choisie de manière arbitraire à chacun des $|V_{G_k}| - 1$ autres sommets de G_k . Une quantité de flot transitant dans une hiérarchie peut passer plus d'une fois entre deux sommets adjacents u et v , mais chaque passage du flot doit être fait par une occurrence distincte de l'arête initiale entre u et v . Si le flot transite dans n occurrences d'une arête alors les n occurrences de cette arête doivent être présentes dans la hiérarchie. Ce modèle permet de respecter la contrainte sur le degré. En effet, chaque sommet réceptionnant (destination ou relai) du flot arrivant d'un prédécesseur différent est autorisé à avoir $B - 1$ successeurs supplémentaires. Ainsi, chaque sommet peut avoir au plus un nombre de successeurs égal au nombre de ses prédécesseurs multiplié par la borne sur le degré B moins 1. Résoudre notre programme linéaire ne donne pas directement une solution pour le problème DCMSH. En effet, la relation entre les duplications d'arêtes et les duplications de sommets doit être réalisée. Cette relation permet de construire une hiérarchie pour le problème DCMSH qui respecte les conditions de la proposition 2.3.6. Cette étape supplémentaire est réalisée en temps polynomial (Section 2.3.2.3).

Les paramètres du graphe restent inchangés (voir Section 2.2). Seules les variables en nombres entiers changent car nous avons une variable par duplication d'arêtes. Cependant, le paramètre β représentant le nombre de duplications de chaque arc dans le k -graphe est ajouté. Il est égal à 2 si B est égal à 2, $\frac{|V_G|-2}{B-2}$ sinon.

Les variables en nombres entiers :

- $L_i(u, v)$: Variable binaire. Elle est égale à 1 si l'occurrence i de l'arc (u, v) se trouve dans la hiérarchie de recouvrement, 0 sinon.
- $F_i(u, v)$: Variable de flot. Elle représente le nombre d'unités de flot transitant sur l'occurrence i de l'arc (u, v) .

L'objectif est de minimiser le coût total des arêtes appartenant à la hiérarchie de recouvrement. La fonction objective peut être exprimée comme suit :

$$\text{Minimiser : } \sum_{u \in V_G} \sum_{v \in \text{Out}(u)} \sum_{i=1}^{\beta} L_i(u, v) * C(u, v) \quad (2.12)$$

La solution doit respecter les contraintes suivantes :

Contraintes de degré :

$$\sum_{v \in \text{out}(s)} \sum_{i=1}^{\beta} L_i(s, v) \leq B + (B - 1) * \sum_{v \in \text{In}(s)} \sum_{i=1}^{\beta} L_i(v, s) \quad (2.13)$$

$$\sum_{v \in \text{out}(u)} \sum_{i=1}^{\beta} L_i(u, v) \leq (B - 1) * \sum_{v \in \text{In}(u)} \sum_{i=1}^{\beta} L_i(v, u) \quad (2.14)$$

Les contraintes (2.13) et (2.14) assurent que pour chaque sommet excepté la source, le nombre des successeurs est au plus égal au nombre de prédécesseurs multiplié par la borne sur le degré B moins 1 (chaque flot arrivant peut être divisé en $B - 1$ flots sortants). Pour la source, le nombre des successeurs est au plus égal à B plus le nombre de ses prédécesseurs multiplié par $B - 1$ car la première occurrence de la source peut avoir B successeurs bien qu'elle n'ait pas de prédécesseur.

$$(B - 1) * \sum_{v \in \text{In}(u)} \sum_{i=1}^{\beta} L_i(v, u) - \sum_{v \in \text{Out}(u)} \sum_{i=1}^{\beta} L_i(u, v) \leq B - 1 \quad (2.15)$$

La contrainte (2.15) assure que chaque sommet peut avoir k prédécesseurs si et seulement si ce sommet a au moins $B * (k - 1)$ successeurs. Ceci permet de construire une hiérarchie optimale pour laquelle chaque sommet u a au plus 1 occurrence avec un degré strictement inférieur à B (Proposition 2.3.6). Cette contrainte n'est pas indispensable mais contribue à réduire l'espace de recherche de la solution.

$$\sum_{v \in \text{In}(u)} L_1(v, u) \geq 1 \quad \forall u \in V_G \setminus \{s\} \quad (2.16)$$

La contrainte (2.16) garantit que chaque sommet excepté la source a au moins un prédécesseur. Cette contrainte assure qu'aucun sommet ne soit isolé.

Contrainte de connexité :

Afin de garantir la connexité de la hiérarchie, on introduit dans notre programme des contraintes

de flot adaptées aux spécificités des hiérarchies.

$$\sum_{i=1}^{\beta} \sum_{v \in Out(s)} F_i(s, v) = \sum_{i=1}^{\beta} \sum_{v \in In(s)} F_i(v, s) + |V_G| - 1 \quad (2.17)$$

La source, à l'instar des autres sommets, peut être dupliquée dans la hiérarchie optimale. Cependant, la première duplication est celle qui joue véritablement le rôle de source puisque c'est celle par laquelle le flot est émis. Les autres duplications de la source ne sont que des relais. C'est la raison pour laquelle nous permettons aux occurrences de la source autres que celle qui émet le flot d'avoir des prédécesseurs. Naturellement, la somme du flot sortant de la source moins le flot entrant dans la source est égale à $|V_G| - 1$, ce qui correspond au flot émis par cette source.

$$\sum_{i=1}^{\beta} \sum_{v \in In(u)} F_i(v, u) = \sum_{i=1}^{\beta} \sum_{v \in Out(u)} F_i(u, v) + 1 \quad \forall u \in V_G - \{s\} \quad (2.18)$$

L'équation (2.18) assure que chaque sommet, exceptée la source, consomme une seule unité de flot. Cette contrainte assure également que chaque sommet est bien relié à la source s .

$$F_i(u, v) \geq L_i(u, v) \quad \forall u, v \in V_G \quad (2.19)$$

$$F_i(u, v) \leq (|V_G| - 1) * L_i(u, v) \quad \forall u, v \in V_G \quad (2.20)$$

Les contraintes 2.19 et 2.20 permettent de relier les variables $F_i(u, v)$ et $L_i(u, v)$. Elles assurent que chaque arc (u, v) transportant du flot ($F_i(u, v) > 0$) soit sélectionné pour être dans la hiérarchie de recouvrement ($L_i(u, v) = 1$) et que chaque arc sélectionné pour être dans la hiérarchie de recouvrement ($L_i(u, v) = 1$) transporte bien du flot ($F_i(u, v) > 0$).

Inégalités valides :

Ces contraintes ne changent aucunement la solution de notre PLNE mais améliorent le temps de résolution de notre problème.

$$L_{i-1}(u, v) \geq L_i(u, v) \quad \forall u \in V_G, \quad \forall v \in Out(u), \quad 2 \leq i \leq \beta \quad (2.21)$$

La contrainte (2.21) assure que l'occurrence i de l'arc (u, v) puisse être sélectionnée pour figurer dans la hiérarchie si et seulement si l'occurrence $i - 1$ est déjà sélectionnée.

$$\sum_{u \in V_G} \sum_{v \in In(u)} L_1(v, u) \geq |V_G| - 1 \quad \forall u \in V_G \quad (2.22)$$

Cette contrainte assure que le nombre total des arcs d'occurrence 1 soit supérieur ou égal à $|V_G|$ moins 1.

$$\sum_{u \in V_G} \sum_{v \in \text{Out}(u)} \sum_{i=1}^{\beta} L_i(u, v) \leq \frac{|V_G| - 2}{B - 2} + |V_G| - 1 \quad (2.23)$$

Conformément à la proposition 2.3.8, cette contrainte assure que le nombre total des arcs de la hiérarchie optimale doit être inférieur ou égal à $\frac{|V_G| - 2}{B - 2} + |V_G| - 1$.

2.3.2.3 Troisième étape : Construction d'une hiérarchie optimale

Notons $G_s = (V_{G_s}, E_{G_s})$ le graphe connexe orienté tel que $V_{G_s} = V_G = V_{G_k}$ et que E_{G_s} contienne uniquement les arêtes de E_{G_k} traversées par un flot non-nul. Afin de construire la hiérarchie optimale H^* , il est nécessaire de distinguer les duplications des sommets. Ceci est possible grâce aux propriétés des contraintes du programme linéaire. En effet, les contraintes 2.13 et 2.14 assurent que chaque sommet u ayant p prédécesseurs peut avoir au plus $p \cdot (B - 1)$ successeurs. Ainsi, nous pouvons dupliquer le sommet u p fois et relier à chacune des duplications un prédécesseur parmi les p existants et $B - 1$ successeurs parmi les $p \cdot (B - 1)$ existants. Notons que les $p - 1$ premières duplications d'un sommet u ont $(B - 1)$ successeurs et la dernière duplication en a au moins 1, et au plus $(B - 1)$ (proposition 2.3.6). Ainsi, tous les sommets de la hiérarchie optimale respectent la contrainte sur le degré B . Elle est également connexe car notre programme linéaire garantit l'envoi de flot de la source à tous les sommets.

La figure 2.8 montre un exemple de construction de la hiérarchie optimale pour le problème DCMSH à partir de G_s avec $B = 3$. Le sommet u du graphe G_k a 4 prédécesseurs, par conséquent, ce sommet est dupliqué 4 fois dans la hiérarchie optimale H^* . Les 3 premières duplications ont $B - 1$ successeurs, et la dernière n'en a qu'un seul.

Pour chaque sommet $m \in V_{G_s}$, nous posons $d_{G_s}^-(u) = \sum_{i=1}^{\beta} L_i(v, u)$ le nombre de ses prédécesseurs et $d_{G_s}^+(u) = \sum_{i=1}^{\beta} L_i(u, v)$ le nombre de ses successeurs. La construction de la hiérarchie optimale H^* à partir du graphe G_s est la suivante :

1. Pour chaque sommet $m \in V_{G_s} - \{s\}$, on crée $d_{G_s}^-(u)$ duplications de ce sommet dans H^* . Le sommet qui a été considéré comme étant la source dans le programme linéaire est dupliqué $d_{G_s}^-(s) + 1$ fois car sa première occurrence n'a pas de prédécesseur.
2. Nous faisons un parcours en largeur du graphe G_s à partir de la source s jusqu'aux feuilles. À chaque sommet u de G_s correspondent $d_{G_s}^-(u)$ sommets dans H^* . Soit $N_{u(G_s)}^+$ l'ensemble des successeurs de u dans G_s , nous construisons $N_{u(H^*)}^+$ l'ensemble des successeurs de $u^1, u^2, \dots, u^{d_{G_s}^-(u)}$ dans H^* tel que s'il existe p arêtes entre u et un successeur v dans G_s alors les sommets v^1, v^2, \dots, v^p sont ajoutés à l'ensemble $N_{u(H^*)}^+$. Nous divisons l'ensemble $N_{u(H^*)}^+$ en $d_{G_s}^-(u) - 1$ sous-ensembles disjoints de taille $B - 1$ et un dernier sous ensemble non vide de taille au plus $B - 1$. Nous relierons chaque occurrence du sommet u dans H^*

à tous les éléments (sommets) d'un sous-ensemble distinct. Ainsi, chaque sommet de la hiérarchie optimale H^* aura au plus $B-1$ successeurs et un seul prédécesseur. La contrainte sur le degré est alors respectée par tous les sommets.

3. Nous supprimons les orientations des arcs.

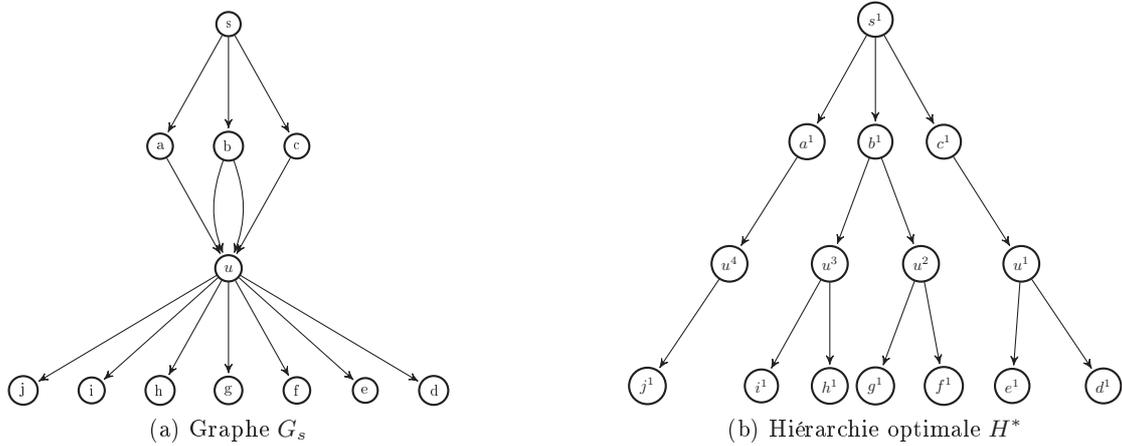


Figure 2.8 – Construction de la hiérarchie optimale pour le problème DCMSH avec $B = 3$ à partir du graphe G_s

Lemme 2.3.10. *La hiérarchie H^* obtenue à l'issue de la troisième correspond à une hiérarchie optimale pour le problème DCMSH dans G_k non-orienté.*

Démonstration. Procédons par contradiction. Supposons que la hiérarchie H^* (construite à partir du graphe G_s) ne soit pas optimale pour le problème DCMSH. Il existe alors une hiérarchie de recouvrement H' ayant un coût strictement inférieur à celui de H^* et qui respecte la contrainte sur le degré des sommets. Nous choisissons un sommet arbitraire de H' et nous orientons la hiérarchie à partir de ce sommet. Nous construisons le graphe G_s où toutes les occurrences d'un sommet dans la hiérarchie sont représentées par un seul sommet dans le graphe G_s et pour tout arc (u^i, v^j) entre une occurrence du sommet u et une occurrence du sommet v , nous mettons un nouvel arc de u vers v dans le graphe G_s (l'inverse de l'étape trois). Le graphe $G_{s'}$ est connexe, il respecte alors les contraintes de connexité du programme linéaire. Le nombre de successeurs de chaque sommet du graphe $G_{s'}$ est au plus égal au nombre de prédécesseurs fois $B - 1$, excepté la source, pour laquelle son nombre de successeurs est au plus égal à B plus le nombre de prédécesseurs fois $B - 1$. Les contraintes 2.13 et 2.14 sont alors respectées. Chaque sommet excepté la source a au moins un prédécesseur, la contrainte 2.16 est également respectée. Le graphe $G_{s'}$ respecte les contraintes de degré et de connexité du PL. Sachant que le programme linéaire a pour objectif de minimiser le coût de la solution et qu'il retourne les variables correspondant au graphe G_s , il est absurde que le coût de G_s soit strictement supérieur à celui de $G_{s'}$. \square

Lemme 2.3.11. *La hiérarchie H^* obtenue à l'issue de l'étape trois est optimale pour le problème DCMSH dans le graphe G .*

Démonstration. On peut faire correspondre à toute hiérarchie dans G_k une hiérarchie dans G . Le coût d'une arête $\{u, v\}$ dans G est identique au coût de l'arête $\{u, v\}$ dans G_k . Une hiérarchie optimale pour le problème DCMSH dans G_k correspond alors à une hiérarchie optimale pour ce même problème dans G . En considérant la proposition 2.3.10, nous pouvons alors affirmer que la hiérarchie H^* obtenue à l'issue de l'étape trois est optimale pour le problème DCMSH dans le graphe G . □

2.3.3 Résultats expérimentaux

Cette section vise à montrer les avantages du problème DCMSH par rapport au problème DCMST.

2.3.4 Simulations

Des simulations sont réalisées pour comparer la structure de hiérarchie de recouvrement aux arbres de recouvrement. Les *PLNE* des problèmes DCMST et DCMSH sont implémentés en langage C en incluant le package CPLEX. Les instances seront d'une part le graphe modélisant le réseau NSF (National Science Foundation Network), et d'autre part des graphes aléatoires générés en utilisant NetGen [KNS74].

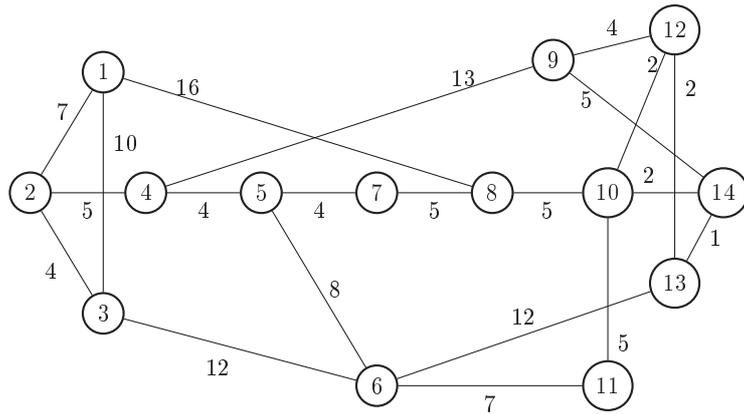
On considère 9 valeurs différentes pour le nombre de sommets des graphes aléatoires G : $|V| \in \{20, 30, 40, 50, 60, 70, 80, 90, 100\}$. La densité des graphes aléatoires est fixée à 2 (Nous avons repris la définition de la densité formulée dans [CGI09] qui est le ratio entre le nombre d'arêtes et le nombre de sommets). Ils sont donc peu denses. Les graphes peu denses sont plus appropriés pour la représentation de la réalité des réseaux optiques. Afin d'avoir un jeu de tests significatif, 100 instances réalisables pour le problème DCMST et ayant des coûts distribués aléatoirement sur les arêtes sont générées pour chaque valeur de $|V_G|$.

Les PLNE des problèmes DCMST et du DCMSH sont résolues pour cet ensemble de jeu d'instances avec la borne sur le degré B fixée à 2 et à 3.

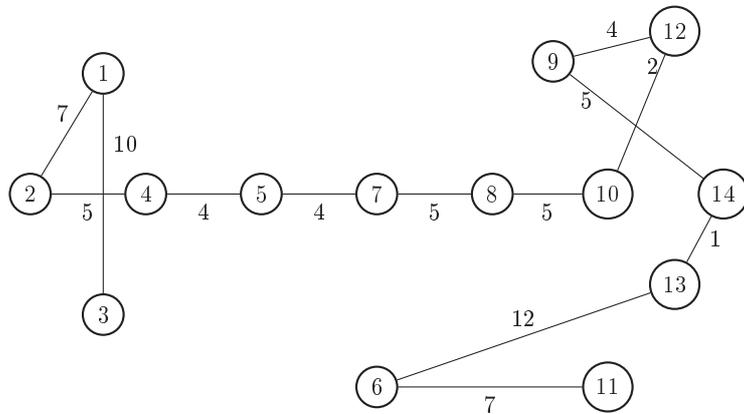
2.3.5 Résultats des simulations

La figure 2.9a montre la modélisation du réseau NSF avec un graphe G arêtes pondérées. Le coût de la hiérarchie de recouvrement de G optimale pour le problème DCMSH (figure 2.9c) est strictement inférieur au coût de l'arbre de recouvrement de G optimal pour le problème DCMST (figure 2.9b).

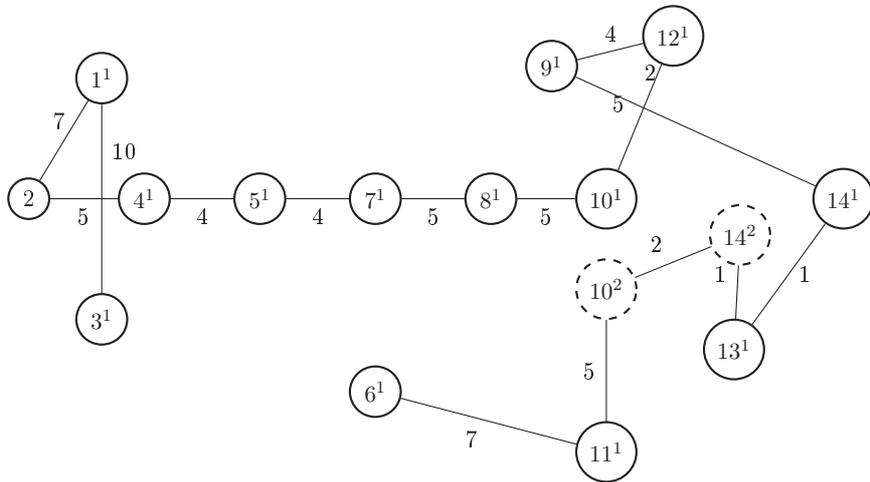
Afin d'analyser les résultats de manière pertinente, le pourcentage des instances non réalisables pour chaque valeur de $|V_G|$ est tout d'abord indiqué. Notons que toute instance est réalisable pour le problème DCMSH et ceci pour tout $B \geq 2$. Les pourcentages d'instances non réalisables pour le problème DCMST avec borne sur le degré B fixée à 2 et à 3 sont présentés dans la table



(a) Modélisation du réseau NSF



(b) Solution optimale pour le problème DCMST ($R = 2$), coût total égal à 68.



(c) Solution optimale pour le problème DCMSH ($R = 2$), coût total égal à 64.

Figure 2.9 – Arbre optimal versus hiérarchie optimale du réseau NSF

2.1.

Si un graphe est réalisable pour le problème DCMST avec $B = 2$ alors ce graphe est nécessai-

$ V $	20	30	40	50	60	70	80	90	100
$R = 2$	52.03%	78.81%	89.62%	91.67%	92.05%	94.28%	96.01%	96.89%	98.11%
$R = 3$	9.20%	33.44%	46.82%	55.12%	60.67%	63.07%	65.41%	66.81%	69.03%

TABLE 2.1 – Proportion d’instances non réalisables pour le problème DCMST

rement hamiltonien. Cependant, les graphes aléatoires générés sont des graphes peu denses, par conséquent, la probabilité de générer des graphes hamiltoniens est faible. La proportion d’instances non réalisables pour le problème DCMST est importante et augmente avec la taille du graphe.

$B = 2$									
$ V_G $	20	30	40	50	60	70	80	90	100
Coût moyen DCMST	7487	12706	18985	22536	25195	29026	32380	35926	39148
Coût moyen DCMSH	5961	9556	13825	15874	17281	19221	21053	23341	24729
Amélioration	20.37%	24.79%	27.18%	29.56%	31.41%	33.78%	34.98%	35.03%	36.83%
$B = 3$									
$ V_G $	20	30	40	50	60	70	80	90	100
Coût moyen DCMST	6477	10623	14327	18101	21358	23269	25905	28504	30197
Coût moyen DCMSH	5671	8725	11408	14006	16967	17282	18641	19924	20696
Amélioration	11.80%	17.20%	20.37%	22.62%	24.11%	25.73%	28.04%	30.10%	31.60%

TABLE 2.2 – Coût moyen des hiérarchies versus coût moyen des arbres

Les résultats numériques concernant les coûts des hiérarchies et des arbres de recouvrement, pour les instances pour lesquelles le problème DCMST a une solution, sont présentés dans la table 2.2. Une solution optimale pour le problème DCMST est une solution réalisable pour le problème DCMSH, le coût de la hiérarchie optimale est donc toujours inférieur ou égal au coût de l’arbre optimal (si celui-ci existe). Le pourcentage moyen d’amélioration du coût varie entre 20% et 36% quand $B = 2$ et entre 11% et 31% quand $B = 3$. Le pourcentage d’amélioration du coût augmente en fonction de la taille du graphe. Quand la taille du graphe est grande, la probabilité qu’un sommet soit dupliqué dans la hiérarchie augmente et ceci accroît la probabilité d’amélioration du coût. Cependant, l’amélioration diminue lorsque la borne B augmente. Cela est dû à l’augmentation de la probabilité que l’arbre de coût minimum sans contraintes soit réalisable pour le problème DCMST. Même s’il n’est pas réalisable, le coût de l’arbre avec contraintes s’en approche et donc l’intérêt des hiérarchies diminue. Sachant que l’arbre de recouvrement de coût minimum est une borne inférieure à la fois pour le problème DCMST et le problème DCMSH, il est normal que l’amélioration moyenne baisse.

L’amélioration est plus claire dans les figures 2.10a et 2.10b qui représentent le coût moyen des arbres de recouvrement et des hiérarchies de recouvrement pour $B = 2$ et $B = 3$, respectivement. Dans les deux figures, l’écart type n’est pas très significatif. Les figures 2.11a et 2.11b apportent plus de précisions à nos résultats. Ces figures montrent, pour $B = 2$ et pour $B = 3$, le nombre d’instances pour lesquelles il y a une amélioration du coût incluse dans un intervalle spécifique en fonction de la valeur de $|V_G|$. Pour $B = 2$ et $|V_G| \geq 70$, les hiérarchies améliorent le coût de plus de 30% pour plus de 50 instances parmi 100. L’amélioration augmente en fonction de la taille des instances. En effet, pour $|V| = 90$, les hiérarchies améliorent de plus de 30% pour plus

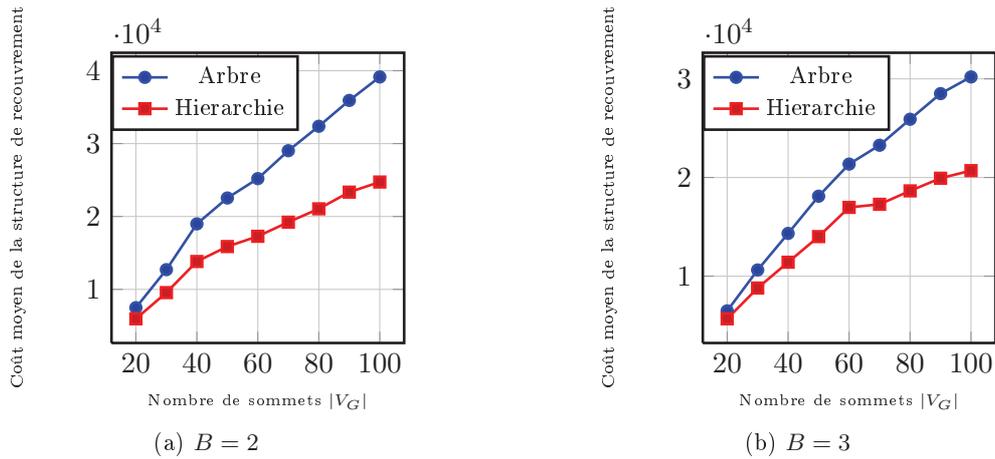


Figure 2.10 – Arbre de recouvrement optimal versus hiérarchie de recouvrement optimale

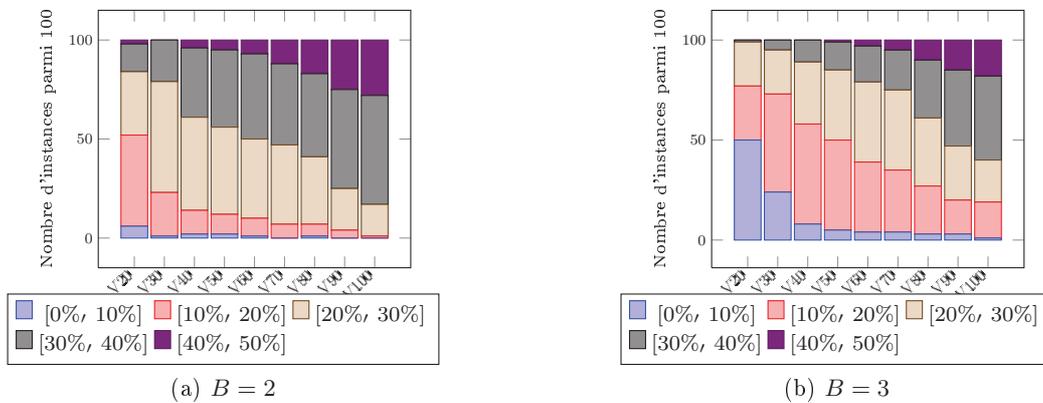


Figure 2.11 – Nombre d’instances en fonction de l’amélioration du coût par les hiérarchies

de 75 instances parmi 100. Pour $B = 3$, l’amélioration est moins marquée. On peut cependant observer dans la figure 2.11b que pour $|V| \geq 70$, les hiérarchies améliorent le coût de plus de 20% pour plus de 65 instances parmi 100.

Étant donné que toutes les instances générées ont un ratio constant entre le nombre d’arêtes et le nombre de sommets, le nombre d’arêtes augmente en fonction de l’augmentation de la taille de l’instance. Sachant que les hiérarchies ont la possibilité d’utiliser plusieurs fois une même arête, le nombre de hiérarchies de recouvrement du graphe est donc supérieur au nombre d’arbres de recouvrement de celui-ci. Ceci explique l’amélioration du coût observée.

2.4 Résolution approchée du problème DCMSH

Dans cette section, nous nous intéressons à la résolution approchée du problème DCMSH et nous prouvons que ce problème est approximable avec un ratio $R = \frac{B}{B-1}$. Rappelons que le problème DCMST est non approximable à un facteur constant du coût de l’arbre optimal [RMR⁺93a] (il n’est pas dans la classe APX). Les principaux travaux qui ont été menés sur l’approximation du

problème DCMST ont été orientés vers l'élaboration d'algorithmes d'approximation bi-critères. Singh et al. proposent dans [SL07] un algorithme d'approximation $(1, B + 1)$ approchée pour le problème DCMST qui est le meilleur ratio d'approximation possible. Un algorithme d'approximation trivial du problème DCMSH avec ratio de deux consiste à trouver un arbre de recouvrement de coût minimum du graphe et de parcourir l'arbre avec un parcours en longueur. Le coût du parcours est ainsi égal à deux fois le coût de l'arbre de recouvrement de coût minimum.

L'algorithme proposé dans [MDM14] et permettant d'approximer le problème DCMSH avec un ratio $R = \frac{B}{B-1}$ est construit en trois étapes. Étant donné un graphe $G = (V_G, E_G)$ connexe arêtes pondérées et une borne sur le degré B :

1. Nous considérons comme point de départ de notre algorithme (borne inférieure) l'arbre de recouvrement de coût minimum (cet arbre peut ne pas respecter la borne sur le degré).
2. Dans le cas où l'arbre de coût minimum respecte la borne sur le degré B , c'est une solution à notre problème. Dans le cas contraire, nous le décomposons en un ensemble d'étoiles arêtes-disjointes. Nous couvrons chaque étoile en utilisant une hiérarchie de recouvrement avec un coût borné et en respectant la borne sur le degré B .
3. Enfin, nous construisons une hiérarchie en connectant toutes les hiérarchies couvrant les étoiles.

2.4.1 Recouvrement d'étoiles par les hiérarchies

Soit S_k une étoile ayant k sommets et $k - 1$ arêtes. Notons c le sommet central et $c(S)$ la somme des coûts des arêtes la composant. Supposons que $B < k - 1$. La hiérarchie de recouvrement de S_k de coût minimum respectant la borne sur le degré contient plusieurs fois le sommet central. Certaines feuilles peuvent également être dupliquées. Afin d'assurer le recouvrement de tous les sommets de l'étoile, chaque sommet de celle-ci doit apparaître au moins une fois dans la hiérarchie de recouvrement et chaque arête est donc utilisée au moins une fois dans la hiérarchie de recouvrement. Résoudre le problème DCMSH pour une étoile revient donc à minimiser le coût des arêtes dupliquées.

Dans ce qui suit, nous proposons le calcul d'une hiérarchie de recouvrement d'étoile en respectant la borne sur le degré B . Notons que l'algorithme permettant de le faire ne garantit pas l'optimalité de la hiérarchie couvrant l'étoile, mais permet néanmoins de garantir le ratio d'approximation de notre heuristique.

Dans notre algorithme, quand une duplication d'arête est nécessaire, les arêtes de moindre coût sont utilisées dans un ordre croissant. De plus, chaque arête sélectionnée est dupliquée au plus une fois. Plus formellement, partitionnons les arêtes de l'étoile comme suit. Nous créons $\lfloor k/(B - 1) \rfloor + 1$ ensembles dans la partition. Chaque ensemble, sauf le dernier, contient $B - 1$ arêtes (si $k \bmod (B - 1) = 0$, alors le dernier ensemble est vide). Les $\lfloor k/(B - 1) \rfloor$ arêtes de plus faibles coûts sont distribuées dans la partition : chacune de ces arêtes est mise dans

un ensemble différent (si le dernier ensemble est vide alors aucune arête de plus faible coût n'est ajoutée à celui-ci). Ainsi, chaque ensemble d'arêtes correspond à une "sous-étoile" qui respecte la borne sur le degré (chaque sous-étoile contient au plus $B - 1$ arêtes). Pour obtenir la hiérarchie H_{S_k} qui couvre toutes les feuilles de l'étoile, les sous-étoiles sont connectées en dupliquant les arêtes de plus faibles coûts. Le sommet central de l'étoile S_k est dupliqué dans la hiérarchie de recouvrement finale autant de fois qu'il existe d'ensembles dans la partition. Ainsi, chaque occurrence du sommet central respecte la borne sur le degré B et la structure obtenue est une hiérarchie. La figure 2.12 illustre la hiérarchie de recouvrement pour $B = 4$ avec $k/(B - 1) \bmod (B - 1) = 0$ (l'intérêt de la dernière occurrence du sommet c sera justifié par la suite).

Lemme 2.4.1. *La hiérarchie de recouvrement H_{S_k} calculée par l'algorithme proposé contient $N_c = \lfloor k/(B - 1) \rfloor + 1$ fois le sommet central c et chaque occurrence respecte la borne sur degré B . Si $N_c \geq 2$, la première et la dernière occurrence ont un degré strictement inférieur à B . Le ratio du coût $r = c(H_{S_k})/c(S)$ est borné par $B/(B - 1)$.*

Démonstration. Preuve par construction. Chaque occurrence de c dans H_{S_k} a un degré au plus B . Dans chaque sous-étoile de la partition, on trouve une occurrence de c et le nombre de feuilles exclusivement couvertes est au plus égal à $B - 1$ (une feuille est couverte de manière exclusive s'il n'existe qu'une seule occurrence de cette feuille dans une seule sous-étoile de la partition). Ce nombre est égal à $B - 1$ pour toutes les occurrences de c exceptée éventuellement une occurrence (la dernière occurrence n'a pas obligatoirement $B - 1$ sommets adjacents). Il y a au plus $\lfloor k/(B - 1) \rfloor$ arêtes dupliquées. Soit D l'ensemble des arêtes dupliquées. En choisissant de dupliquer les arêtes de plus faibles coûts, le coût total de l'ensemble des arêtes de l'étoile dupliquées $c(D) = \sum_{e \in D} c(e)$ est borné par :

$$c(D) \leq \frac{\lfloor k/(B - 1) \rfloor}{k} c(S) \leq \frac{k/(B - 1)}{k} c(S) = \frac{1}{B - 1} c(S)$$

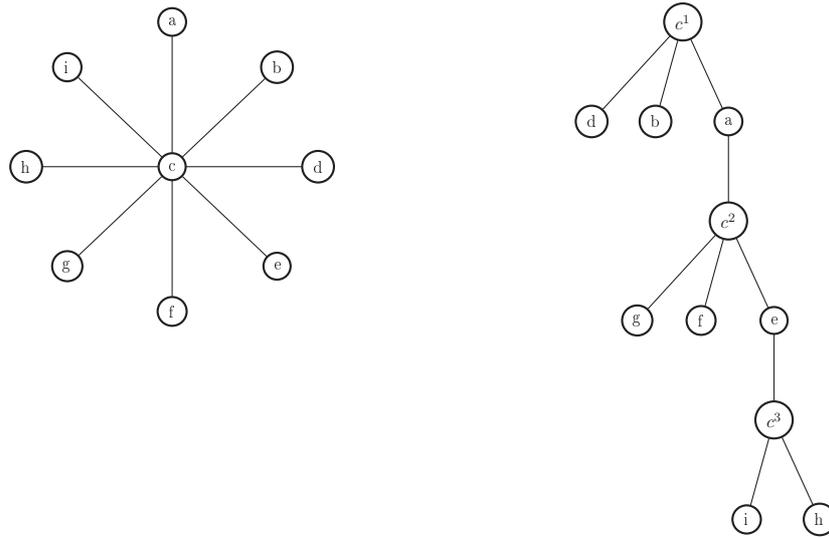
□

Une borne supérieure sur le ratio du coût peut être formulée ainsi :

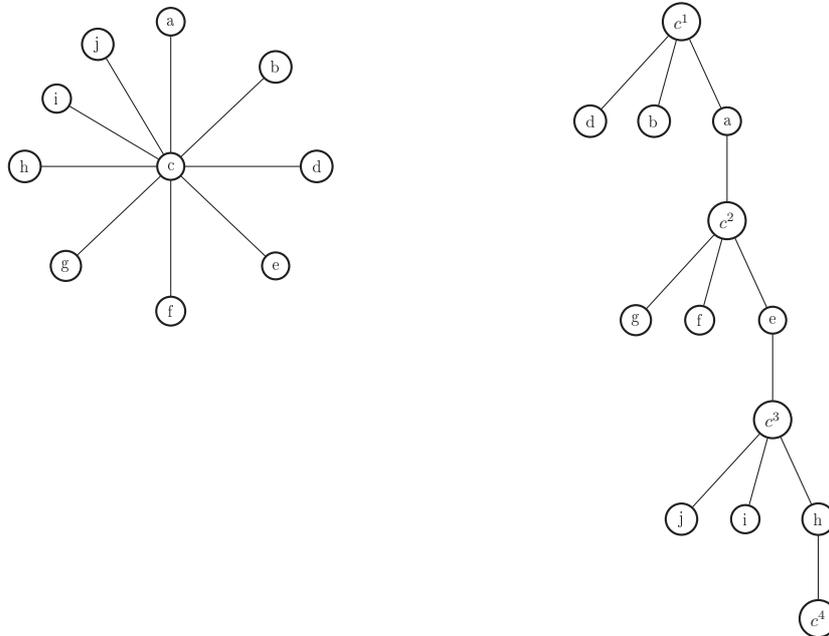
$$r = \frac{c(H_S)}{c(S)} = \frac{c(S) + c(D)}{c(S)} \leq \frac{B}{B - 1}$$

Remarque : Si $N_c = 1$ (le cas $k < B - 1$), alors le sommet central a un degré strictement inférieur à $B - 1$.

La hiérarchie de recouvrement H_{S_k} correspond à un « caterpillar » (un arbre dont tous les sommets sont à une distance 1 du chemin central), chaque sommet dans le chemin central a un degré au plus B . De plus, les occurrences du sommet central dans la première et dans la dernière sous-étoile ont un degré inférieur à B (s'il n'y a qu'une seule étoile, $d_{H_{S_k}} < B - 1$, cf. Remarque).



(a) $B = 4, d_{S_k}(c) \bmod (B - 1) > 0$



(b) $B = 4, d_{S_k}(c) \bmod (B - 1) = 0$

Figure 2.12 – Hiérarchie de recouvrement d'étoile calculé avec l'heuristique proposée.

2.4.2 Un algorithme d'approximation pour le problème DCMSH

Le coût de l'arbre de coût minimum donne une borne inférieure pour le problème DCMSH, une borne supérieure du coût de la hiérarchie renvoyée par l'algorithme d'approximation peut être calculée à partir du coût de l'arbre de coût minimum. Dans ce qui suit, nous proposons un algorithme d'approximation basé sur la décomposition de l'arbre de coût minimum du graphe.

2.4.2.1 La décomposition en étoiles de l'arbre de coût minimum

L'arbre de coût minimum peut être décomposé en un ensemble d'étoiles comme suit. Soit $T = (V_T, E_T)$ un arbre de coût minimum (MST) avec $|V_T| > 2$ et v_1 un sommet arbitraire de T . Le sommet v_1 peut être considéré comme le sommet central d'une étoile S_1 . Certains voisins de v_1 dans S_1 peuvent correspondre à des feuilles dans T , certains autres voisins de v_1 peuvent correspondre à des sommets de branchement. Les sommets de branchement peuvent être considérés comme des sommets centraux des étoiles correspondantes. Récursivement, l'arbre peut être entièrement couvert par des étoiles arête disjointe. La décomposition est illustrée par la Figure 2.13. Les étoiles sont arête disjointe et couvrent toutes les arêtes de T , trivialement : $c(T) = \sum_{i=1}^k c(S_i)$ ou les $S_i, i = 1, \dots, k$ sont les étoiles dans la décomposition.

2.4.2.2 L'heuristique d'approximation du problème DCMSH

Afin de calculer une approximation pour le problème DCMSH, on propose l'algorithme suivant.

Algorithm 1: Algorithme d'approximation du problème DCMSH

Entrée: Graphe $G = (V_G, E_G)$, une borne sur le degré B .

Sortie : Hiérarchie de recouvrement $H_A = (G, h, Q)$.

1. Calculer un arbre de recouvrement de coût minimum T de G .
 2. Décomposer l'arbre T en étoiles S_1, S_2, \dots, S_k .
 3. Pour chaque étoile S_i , calculer une hiérarchie de recouvrement de S_i comme proposé dans la section précédente.
 4. « Reconnecter » les sous-hiérarchies H_i pour former la hiérarchie de recouvrement connexe H_A . Une connexion entre deux étoiles est nécessaire si une feuille dans une étoile S_i coïncide avec le sommet central d'une autre étoile S_j . Dans H_i , les feuilles de S_i ne sont pas dupliquées et ont un degré 1 ou 2. Notons par l_i une feuille dans S_i qui correspond au sommet central c_j de S_j et au sommet v_k dans le graphe original. Rappelons que c_j peut être répété dans H_j mais sa première occurrence a un degré égal à $B - 1$ au plus.
 - Si l_i a un degré 1 dans H_i , alors l_i peut être fusionné avec la première occurrence du sommet c_j dans H_j , ainsi, un seul sommet représente la fusion des deux sommets dans la hiérarchie finale et respecte la borne sur le degré B .
 - Si l_i a un degré 2 dans H_i (ce n'est pas une feuille), la connexion peut être réalisée comme suit.
 - Si le sommet central c_j correspond à une seule occurrence dans H_j , alors cette occurrence est de degré strictement inférieur à $B - 1$. Par conséquent, l_i et c_j peuvent être fusionnés dans la hiérarchie finale et le sommet issu de cette fusion respecte la borne sur le degré B .
 - S'il existe plusieurs occurrences de c_j dans H_j , la première et la dernière occurrence ont un degré au plus $B - 1$ et les deux arêtes adjacentes de l_i peuvent être rattachées à ces deux occurrences sans violation de la borne sur le degré par ces deux sommets.
 5. La hiérarchie H_A peut contenir des arêtes inutiles (des arêtes qui retournent à une occurrence du sommet central dont le degré est égal à un). Les arêtes inutiles peuvent être supprimées.
-

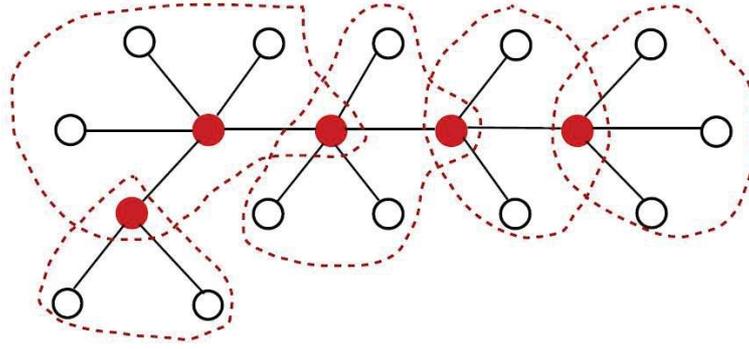


Figure 2.13 – Décomposition en étoiles d'un arbre

Théorème 2.4.2. *L'algorithme 1 donne une $R \leq \frac{B}{B-1}$ approximation de la solution optimale.*

Démonstration. L'algorithme est basé sur la décomposition de l'arbre de coût minimum T^* en un ensemble d'étoiles arête disjointe. Soit $c(S_i)$ le coût des étoiles $S_i, i = 1, \dots, k$ dans la décomposition. Trivialement $c(T^*) = \sum_{i=1}^k c(S_i)$.

En considérant le résultat du Lemme 2.4.1, le coût de la hiérarchie de recouvrement est borné par :

$$c(H) = \sum_{i=1}^k c(H_{S_i}) \leq \sum_{i=1}^k \frac{B}{B-1} c(S_i) = \frac{B}{B-1} c(T^*)$$

Ainsi, le ratio d'approximation peut être formulé par :

$$R = \frac{c(H)}{c(H^*)} \leq \frac{c(H)}{c(T^*)} \leq \frac{B}{B-1}$$

□

Remarque : Si pour tout sommet $v \in V_{T^*}$, $d_{T^*}(v) < B$ alors l'algorithme retourne l'arbre de coût minimum qui est, dans ce cas, une solution optimale.

Nous proposons de discuter le fait que le ratio d'approximation de l'heuristique n'est pas obtenu en fonction de la hiérarchie optimale mais à partir de l'arbre de coût minimum du graphe.

2.4.3 Discussion sur le ratio d'approximation

Notre heuristique n'utilisant que les arêtes appartenant à l'arbre de coût minimum, la hiérarchie de recouvrement résultante, pour des petites valeurs de B , peut avoir un coût largement supérieur à celui de la hiérarchie optimale. Le théorème suivant démontre cependant que le coût de la hiérarchie renvoyé par notre heuristique est le meilleur que l'on puisse obtenir en se basant sur l'arbre de coût minimum.

Théorème 2.4.3. *Il n'existe pas de ratio d'approximation inférieur à $B/(B-1)$ pouvant être obtenu par une heuristique uniquement basée sur l'arbre de coût minimum.*

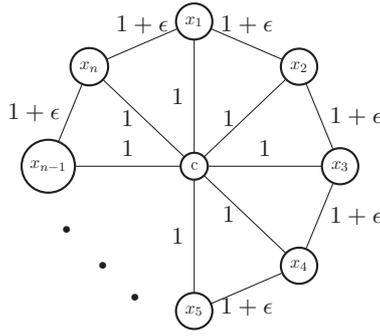


Figure 2.14 – Le graphe utilisé dans le théorème 2.4.3

Démonstration. Soit $G = (V, E)$ le graphe formé d'une étoile à n branches et des arêtes reliant les extrémités des branches consécutives (voir Figure 2.14). Supposons que $c(c, x_i) = 1$ pour $i = 1, \dots, n$, $c(x_i, x_{i+1}) = 1 + \epsilon$ pour $i = 1, \dots, n - 1$, et $c(x_n, x_1) = 1 + \epsilon$.

Trivialement, le chemin $P = \langle y, x_1, x_2, x_3, \dots, x_{n-1}, x_n \rangle$ est une hiérarchie de recouvrement de G qui respecte la borne sur le degré pour tout $B > 1$ et ayant un coût $c(P) = 1 + (n - 1)(1 + \epsilon)$.

L'étoile S avec un sommet central c et n feuilles est l'arbre de recouvrement de coût minimum de G . Soit $H^* = (T^*, h, S)$ la hiérarchie optimale couvrant l'étoile S et respectant la borne sur le degré. Dans l'arbre T^* , il ne peut y avoir qu'une seule occurrence de chaque sommet correspondant à une feuille dans S . Si une feuille x_i de S a au moins deux occurrences dans T^* alors :

- Si l'une des deux est une feuille dans T^* , alors elles peuvent être retirées de T^* . Nous obtenons ainsi une hiérarchie de recouvrement contenant le même ensemble de sommets et ayant un coût strictement inférieur ;
- Sinon, chaque occurrence est un sommet interne dans T^* . Soit x_i^1 et x_i^2 deux occurrences de x_i . Soit x_j une feuille dans T^* et soit $T^{*'}$ l'arbre construit à partir de T^* en supprimant la feuille x_j et en remplaçant le label x_i^1 par x_j . Étant donné que tous les voisins de x_i^1 dans T^* sont des occurrences de c , il existe toujours un homomorphisme $h^{*'}$ entre $T^{*'}$ amenant à la même contradiction.

Donc, T^* est un graphe biparti particulier où la partition des sommets peut être réalisée comme suit : un ensemble contient les n_c occurrences de c et l'autre ensemble contient les n sommets correspondant aux feuilles de S . Puisque T^* est un arbre, son nombre d'arêtes est égal à son nombre de sommets moins 1. Par conséquent, $c(H^*) = n + n_c - 1$. Toute occurrence de c a au plus B voisins dans T^* . Donc, le nombre d'arêtes de H^* est au plus égal à $n_c * B$ et nous avons $n + n_c - 1 \leq n_c * B$ ce qui implique $\frac{n-1}{B-1} \leq n_c$. Le coût de H^* est alors au moins $c(H^*) \geq (n - 1) + \frac{n-1}{B-1} = \frac{B(n-1)}{B-1}$. Ainsi, le ratio d'approximation de toute heuristique basée sur l'arbre de recouvrement de coût minimum est supérieur ou égal à $\frac{c(H^*)}{c(P)} = \frac{\frac{B(n-1)}{B-1}}{1+(n-1)(1+\epsilon)}$ et $\frac{(n-1)}{1+(n-1)(1+\epsilon)}$ peut être aussi proche de 1 que l'on veut pour n assez grand et ϵ assez petit. \square

Dans le cas où le calcul de la hiérarchie de recouvrement n'est pas basé sur l'arbre de recouvrement de coût minimum, des résultats plus intéressants peuvent être obtenus. Par exemple, pour $B = 2$, le ratio d'approximation peut être amélioré. En effet, dans ce cas, notre ratio d'approximation est égal à 2, ce qui est le pire cas. Cependant ici, le problème est équivalent à trouver un chemin hamiltonien de coût minimum dans la fermeture métrique de G . Il peut donc être approximé avec un ratio de $3/2$ en utilisant par exemple les techniques présentées dans [Hoo91].

2.5 Conclusion

Tant sur le volet résolution exacte que sur celui de la résolution approchée, la hiérarchie de recouvrement apporte une plus-value considérable au problème classique qui se base sur l'arbre de recouvrement. Nous avons ainsi montré, grâce aux expérimentations effectuées sur des graphes aléatoires, que l'amélioration du coût de la solution optimale du problème DCMSH par rapport à celle du problème DCMST peut atteindre 36%. Sur un cas réel, comme le réseau NSF, l'amélioration est présente mais de manière moins importante. Le taux de réalisabilité des instances est également en faveur de la hiérarchie de recouvrement. En effet, avec une borne égale à deux, ce taux peut être inférieur à 5% pour le problème classique. Alors que, toute instance connexe admet une hiérarchie de recouvrement respectant toute borne $B \geq 2$. Alors que le problème DCMST est non-approximable à facteur constant, nous avons présenté un algorithme qui retourne une hiérarchie d'un coût au plus égal à $\frac{B}{B-1}$ fois celui de la hiérarchie optimale pour le problème DCMSH. Nous prouvons également que cette garantie de performance est la meilleure que l'on puisse obtenir en utilisant une heuristique qui se base uniquement sur l'arbre de coût minimum.

Recouvrement total avec une borne supérieure sur le nombre de sommets de branchement

3.1 Introduction

Dans le chapitre précédent, nous avons montré l'intérêt de la structure de hiérarchie par rapport à l'arbre de recouvrement lorsqu'une borne supérieure sur le degré des sommets est imposée. Nous proposons ici de continuer la comparaison entre la hiérarchie et l'arbre en considérant un autre problème : la recherche de structure de recouvrement avec une borne sur le nombre de sommets de branchement.

La recherche d'arbre de recouvrement d'un graphe ayant un minimum de sommets de branchement est un problème récent qui tire son intérêt pratique principalement du routage multi-cast/broadcast dans les réseaux tout optique. En effet, dans ce type de réseaux, seuls les nœuds équipés de commutateurs optiques OXC-MC sont capables de diviser le signal lumineux entrant. Par conséquent, seuls les sommets du graphe correspondant à ce genre de nœuds sont autorisés à avoir un degré strictement supérieur à deux dans l'arbre de recouvrement. De plus, minimiser les sommets de branchement de l'arbre contribue à réduire les divisions du signal lumineux dans le réseau. Ceci permet d'atténuer la perte de puissance du signal lumineux (voir chapitre 1). Un autre problème visant à réduire cette division du signal lumineux a été introduit récemment. Ce problème consiste à minimiser la somme des degrés des sommets de branchement de l'arbre de recouvrement. Il permet donc de prendre en compte non seulement le nombre des sommets de branchement mais également de minimiser leur degré. Ceci contribue à réduire davantage les divisions du signal.

Dans ce chapitre, nous commençons par introduire et discuter la relation entre ces deux problèmes. Par la suite, nous proposons de les reconsidérer en utilisant la structure de hiérarchie.

Trouver une hiérarchie de recouvrement ayant un minimum de sommets de branchement est trivial car il suffit de faire un parcours d'un arbre de recouvrement quelconque du graphe. Nous introduisons donc un nouveau problème qui consiste à chercher une hiérarchie de recouvrement de coût minimum telle que le nombre de sommets de branchement est borné supérieurement par un entier k . Le schéma d'analyse de ce problème est similaire à celui du problème DCMSH. La complexité est d'abord étudiée. Par la suite, les résolutions exacte et approchée sont abordées. Enfin, une partie expérimentation vient conforter les résultats théoriques en donnant une vision sur l'apport des hiérarchies de recouvrement.

3.2 Arbre de recouvrement : Structure utilisée pour minimiser le nombre de sommets de branchement dans les réseaux optiques

Dans le cadre du routage multicast/broadcast dans les réseaux tout optique, le signal lumineux doit être contrôlé en raison de la perte de puissance de celui-ci. En effet, un signal lumineux arrivant à un nœud équipé de commutateur OXC-MC et ayant m voisins dans l'arbre de recouvrement est divisé en $m - 1$ sous-signaux. Ainsi, la puissance de chaque sous-signal est réduite à une valeur inférieure ou égale à $1/(m - 1)$ du signal original [AD06]. La figure 3.1 montre la nécessité de prendre en compte le degré des sommets de branchement. Quel que soit un arbre de recouvrement du graphe G , il contiendra nécessairement le sommet central comme sommet de branchement (ce sommet relie trois composantes connexes). L'arbre T a donc un nombre minimum de sommets de branchement (un seul sommet de branchement, le sommet central). Le degré du sommet central est égal à 15. Le signal lumineux arrivant à ce sommet sera divisé par 14. L'arbre T' contient également le sommet central comme sommet de branchement mais le degré de ce dernier est minimisé. Le signal lumineux arrivant à ce sommet sera divisé seulement par 2, ce qui générera beaucoup moins de pertes.

3.2.1 Les problèmes MBVST et MDST

Le problème MBVST a été introduit par Gargano et al. dans [GHSV02]. Le problème MDST a été introduit quelques années plus tard par [CGI09]. Le problème MBVST est défini dans le chapitre 1, section 1.4.1.1, page 25. La définition du problème MDST se présente comme suit :

Problème : Minimum Degree Sum Spanning Tree - MDST
Données : Graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$, $c(e) \quad \forall e \in E_G$.
Résultat : Arbre de recouvrement T du graphe G .
Objectif : Minimiser la somme des degrés des sommets de branchement de T .

Trouver un arbre de recouvrement d'un graphe ayant un minimum de sommets de branchement est NP-difficile [GHSV02]. Trouver un arbre de recouvrement dont la somme des degrés

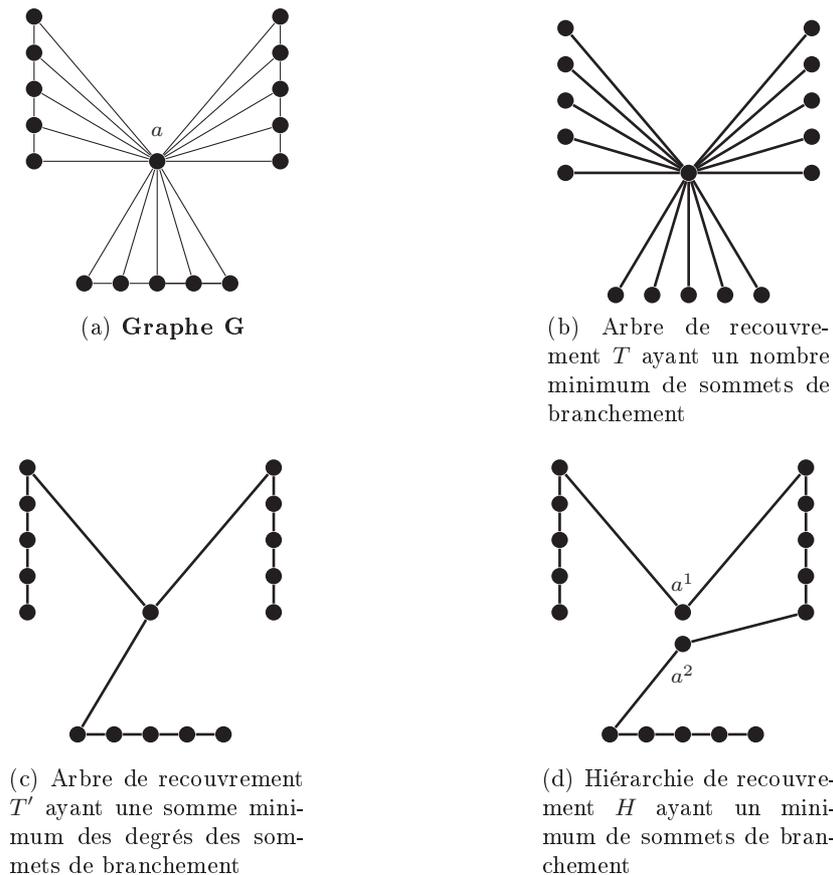


Figure 3.1 – Différence entre le problème MBVST et le problème MDST

des sommets de branchement est minimale est également NP-difficile [CGI09] (les preuves sont données dans les parties 1.4.1.2 et 1.4.2.2). Il suffit pour s'en convaincre de considérer le problème de l'hamiltonicité. En effet, trouver un arbre de recouvrement d'un graphe hamiltonien qui minimise le nombre de sommets de branchement revient à trouver un chemin hamiltonien dans ce graphe. Ceci est NP-difficile. Bien évidemment, le chemin hamiltonien répond également à l'objectif du problème MDST car un tel chemin ne contient aucun sommet de branchement et par conséquent, la somme des degrés de ceux-ci est égale à zéro.

Le problème de recherche de chemin hamiltonien peut alors être considéré comme un cas particulier du problème MBVST. En effet, pour un graphe connexe G donné, si l'arbre de recouvrement T^* de G optimal pour le problème MBVST contient k sommets de branchement, alors on considère que le graphe G s'éloigne d'une « distance » de k de l'état d'hamiltonicité.

Le théorème de Dirac [Dir52], qui est la meilleure condition de densité existante impliquant l'hamiltonicité d'un graphe, peut être généralisé au problème MBVST. En effet, considérons le théorème de Dirac :

Théorème 3.2.1. [Dir52] *Soit G un graphe connexe et k un entier positif. Si tous les sommets*

de G ont un degré supérieur ou égal à $\frac{n-1}{2}$ alors le graphe G est Hamiltonien.

Dans [GHSV02], le théorème de Dirac a été généralisé au problème MBVST en posant la conjecture suivante :

Conjecture 3.2.1. [GHSV02] Soit G un graphe connexe et k un entier positif. Si tous les sommets de G ont un degré supérieur ou égal à $\frac{n-1}{k+2}$ alors le graphe G admet un arbre de recouvrement T ayant au plus k sommets de branchement ($s(T) \leq k$).

Cette conjecture est vérifiée par le théorème de Dirac pour $k = 0$. Gargano et al. la vérifient pour $k = 1$. Elle est cependant ouverte pour $k > 1$.

Cette généralisation du théorème de Dirac au problème MBVST donne une borne inférieure sur le degré que doivent avoir les sommets d'un graphe afin que ce dernier contienne un arbre de recouvrement ayant un nombre limité de sommets de branchement. Elle représente une condition de densité importante pour l'existence d'un tel arbre.

3.2.2 Relation entre le problème MBVST et le problème MDST

La figure 3.1 illustre la différence entre le problème MBVST et le problème MDST. Ainsi, La figure 3.1b représente un arbre de recouvrement T optimal pour le problème MBVST et la figure 3.1c représente un arbre de recouvrement T' optimal pour le problème MDST d'un même graphe. Ces deux arbres sont différents. On est alors tenté de croire que les deux problèmes admettent des solutions différentes. L'arbre de recouvrement T' est optimal pour les deux problèmes. En effet, cet arbre contient un seul sommet de branchement et ce sommet a un degré égal à trois (le mieux que l'on puisse avoir dans le cas où le graphe est non hamiltonien).

Nous définissons, comme suit, la notion de liaison et de non-liaison de deux problèmes d'optimisation :

Définition 3.2.1. Deux problèmes d'optimisation, portant sur les mêmes instances et les mêmes solutions réalisables, ne sont pas liés s'il existe au moins une instance où les deux ensembles de solutions optimales sont disjoints. Ils sont liés sinon.

Malgré l'existence de plusieurs instances présentées dans la littérature montrant que les deux problèmes ont des ensembles de solutions différents (au moins un élément non-commun) [CCR14, CGI09], aucune ne montre que les deux problèmes ne sont pas liés (Voir la partie 1.4.2.3).

Nous proposons dans ce qui suit une instance qui montre que les deux problèmes ne sont pas liés.

Lemme 3.2.2. Le problème MDST et le problème MBVST ne sont pas liés : Il existe un graphe G tel que : quel que soit T un arbre de recouvrement de G alors :

1. Si $s(T) = s^*(G)$ alors $q(T) > q^*(G)$,
2. si $q(T) = q^*(G)$ alors $s(T) > s^*(G)$.

Preuve. La figure 3.2 présente un graphe G qui respecte les conditions de la proposition 3.2.2 :

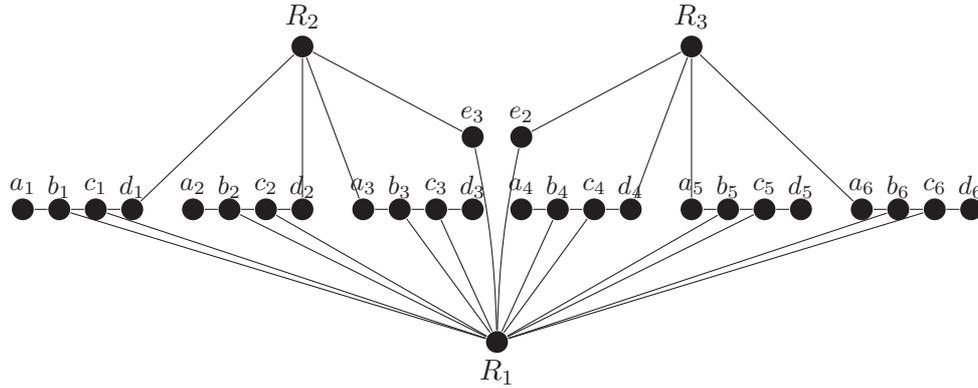


Figure 3.2 – Instance prouvant la proposition 3.2.2

Étape 1 : Si le graphe G de la figure 3.2 est hamiltonien, alors il ne permet pas de distinguer les deux solutions ($s^*(G) = q^*(G) = 0$). On doit donc d'abord prouver que le graphe G ne contient pas de chaîne hamiltonienne. Pour cela, on utilise le lemme suivant :

Lemme 3.2.3. Soit $G = (V, E)$ un graphe. Si G a une chaîne hamiltonienne, alors $\forall S \subseteq V$, le graphe $(G - S)$ a au plus $|S| + 1$ composantes connexes.

Dans le graphe G , il suffit de prendre $S = \{R_1, R_2, R_3\}$, pour conclure que G n'est pas hamiltonien.

Étape 2 : Prouvons que $s^*(G) = 1$. Le graphe G n'est pas hamiltonien, alors $s^*(G) \geq 1$. L'arbre T_1 de la figure 3.3 est un arbre de recouvrement du graphe G et $s(T_1) = 1$. On a donc $s^*(G) = 1$.

Étape 3 : Prouvons que $\forall T$ un arbre de recouvrement, si $s(T) = s^*(G) = 1$ alors R_1 est l'unique sommet de branchement dans T . Le graphe induit par $V_G - \{R_1, R_2, R_3\}$ est composé de 8 composantes connexes. Par conséquent, tout arbre de recouvrement de G doit avoir au moins R_1 ou R_2 ou R_3 comme sommet de branchement.

Notons $G' = (V', E')$ et $G'' = (V'', E'')$ les sous-graphes induits de G tels que $V' = \{a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, a_3, b_3, c_3, d_3, e_3, R_2, R_1\}$ et $V'' = \{a_4, b_4, c_4, d_4, a_5, b_5, c_5, d_5, a_6, b_6, c_6, d_6, e_2, R_3, R_1\}$. Le sommet R_1 est le seul sommet qui connecte les deux sous-graphes induits. Supposons que R_2 soit le seul sommet de branchement d'un arbre de recouvrement de G . Dans ce cas, le

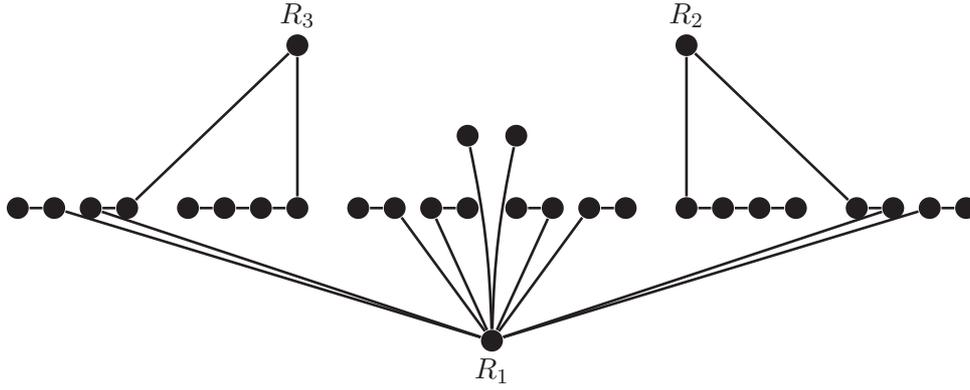


Figure 3.3 – Arbre de recouvrement T_1 du graphe G

sous-graphe induits G' doit contenir un chemin hamiltonien se terminant en R_1 . Or, il suffit de prendre $S = \{R_1, R_3\}$ pour conclure en appliquant le lemme 3.2.3 que G' n'est pas hamiltonien. De manière symétrique, R_3 ne peut pas être le seul sommet de branchement d'un arbre couvrant G .

Dans la figure 3.3, nous avons un arbre ayant uniquement le sommet R_1 comme sommet de branchement. Nous pouvons alors affirmer que seul le sommet R_1 peut être unique sommet de branchement d'un arbre de recouvrement de G .

Étape 4 : Prouvons que si R_1 est l'unique sommet de branchement d'un arbre de recouvrement T alors $q(T)$ est au moins égal à 10. Le nombre de sommets du graphe G est égal à 29. Tout arbre de recouvrement de G a donc 28 arêtes. Supposons qu'il existe un arbre de recouvrement T de G ayant le sommet R_1 comme unique sommet de branchement et tel que $d_T(R_1) = q(T) \leq 9$. Les 40 arêtes de E_G peuvent être partitionnées en 3 sous-ensembles : E_{R_1} l'ensemble des arêtes de G incidentes à R_1 , E_{R_2, R_3} l'ensemble des arêtes incidentes à R_2 ou R_3 et E_r l'ensemble des arêtes restantes. L'arbre T a au plus 9 arêtes dans E_{R_1} car $d_T(R_1) \leq 9$ et au plus 4 arêtes dans E_{R_2, R_3} , car R_1 et R_2 ont un degré au plus égal à 2 dans T . L'arbre T a donc au moins 15 arêtes dans E_r . Comme les arêtes $\{a_i, b_i\}$ sont obligatoirement dans T , il y a au moins 9 arêtes de T parmi $\{\{b_i, c_i\}, \{c_i, d_i\}\}$, $i \in [1, 6]$. Il y a donc au moins 3 ensembles $\{\{b_i, c_i\}, \{c_i, d_i\}\}$ dont les deux arêtes sont dans T . Soit R_2 , soit R_3 est incident à deux d'entre eux (supposons sans perte de généralité que ce soit R_2 et que $\{b_1, c_1\}, \{c_1, d_1\}, \{b_2, c_2\}$ et $\{c_2, d_2\}$ soient dans T . Comme b_i et c_i ne sont pas des sommets de branchement, $\{R_1, b_i\}$ et $\{R_1, c_i\}$ n'appartiennent pas à T , donc $\{d_1, R_2\}$ et $\{d_2, R_2\}$ appartiennent à T . Cependant, cette composante connexe ne peut pas être connectée aux autres composantes de T car le degré de R_2 est au plus égal à 2. Par conséquent, si $s^*(G) = 1$, alors $q(G) \geq 10$.

Étape 5 : Trouvons un arbre de recouvrement T tel $s(T) > 1$ et $q(T) < 10$. Dans la figure 3.4 nous avons un arbre de recouvrement de G ayant deux sommets de branchement tel que $q(T) = 8$.

Nous pouvons ainsi conclure que : quel que soit T un arbre de recouvrement de G tel que $q(T) = q^*(G) = 8$ on a $s(T) > 2$, donc $s(T) \neq s^*(G)$. Quel que soit T un arbre de recouvrement de G tel que $s(T) = s^*(G) = 1$ on a $q(R_1) > 8$, donc $q(T) \neq q^*(G)$.

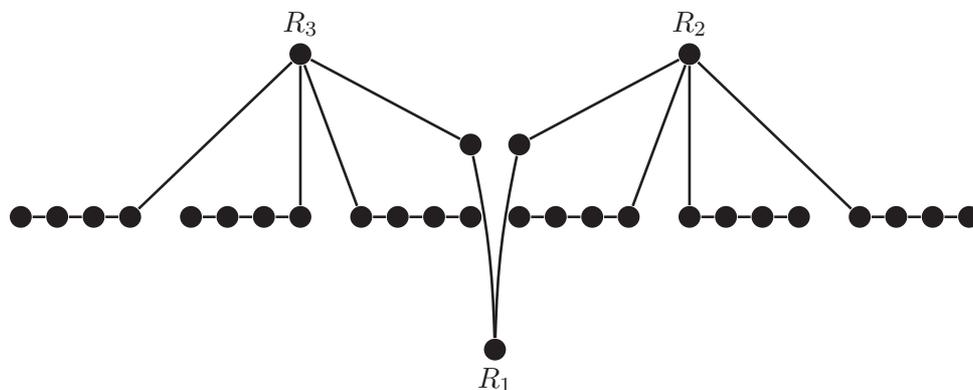


Figure 3.4 – Arbre de recouvrement T_2 du graphe G

□

3.2.3 Arbre de recouvrement de graphe ayant un minimum de feuilles

On remarque, en observant 3.1, qu'il existe une corrélation entre la somme des degrés des sommets de branchement et le nombre de feuilles de l'arbre de recouvrement. En effet, chaque sommet de branchement v va « générer » $d_T(v) - 2$ feuilles dans l'arbre de recouvrement T . Le nombre de feuilles d'un arbre est ainsi égal à la somme des degrés de ses sommets de branchement moins deux fois le nombre de sommets de branchement plus deux $L(T) = q(T) - 2 \cdot s(T) + 2$ [CCR14].

Supposons qu'un arbre T de recouvrement d'un graphe G modélisant un réseau optique soit utilisé pour une communication multicast. Soit v un sommet de l'arbre traversé par le signal lumineux. Si $d_T(v) = 1$ ou $d_T(v) = 2$ alors aucune division de la lumière n'est nécessaire à ce niveau (le sommet v est soit une feuille soit un sommet relai). Si v est un sommet de branchement, alors la transmission du signal lumineux à ses fils va nécessiter $d_T(v) - 2$ divisions du signal [CCR14]. Le nombre total de divisions du signal lumineux dans le réseau est alors égal à la somme des degrés des sommets de branchement moins leur nombre $q(T) - s(T)$. Le nombre total de divisions du signal lumineux dans le réseau est ainsi égal au nombre de feuilles dans ce réseau moins 2. On peut alors considérer que trouver un arbre de recouvrement avec un minimum de feuilles permet de minimiser le nombre total de divisions de la lumière, ce qui incite à considérer le problème suivant :

Problème : Minimum Leaves Spanning Tree - MLST
Données : Graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$
Résultat : Arbre de recouvrement T du graphe G .
Objectif : Minimiser le nombre de feuilles de l'arbre T

Dans [CCR14], une instance G est donnée (figure 3.5) pour laquelle, quel que soit un arbre de recouvrement de G optimal pour le problème MLST, cet arbre n'est ni optimal pour le problème MBVST ni pour le problème MDST. En effet, l'arbre de recouvrement T de G de la figure 3.5b, qui contient un seul sommet de branchement de degré 5, est optimal pour les problèmes MBVST et MDST mais non optimal pour le problème MLST. L'arbre de recouvrement T' de G de la figure 3.5c est, quant à lui, optimal pour le problème MLST et non optimal pour le MBVST et le MDST.

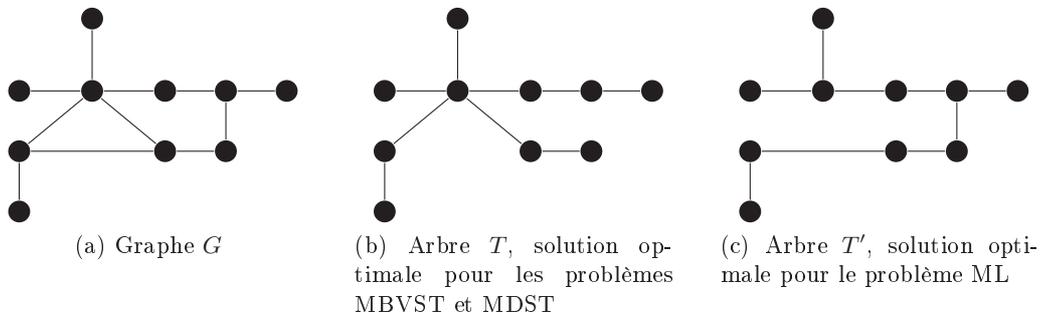


Figure 3.5 – Le problème ML n'est pas équivalent aux problèmes MBVST et MDST

En combinant les instances des figures 3.2 et 3.5, nous pouvons créer une instance qui montre que les trois problèmes sont deux à deux non liés.

Lemme 3.2.4. *Les problèmes MBVST, MDST et MLST sont deux à deux non liés : il existe au moins un graphe G tel que : quel que soit T un arbre de recouvrement de G alors :*

1. Si $s(T) = s^*(G)$ alors $q(T) > q^*(G)$ et $L(T) > L^*(G)$,
2. si $q(T) = q^*(G)$ alors $s(T) > s^*(G)$ et $L(T) > L^*(G)$,
3. si $L(T) = L^*(G)$ alors $s(T) > s^*(G)$ et $q(T) > q^*(G)$.

La preuve est obtenue en considérant les graphes combinant ceux des figures 3.5b et 3.3 et ceux des figures 3.5c et 3.4.

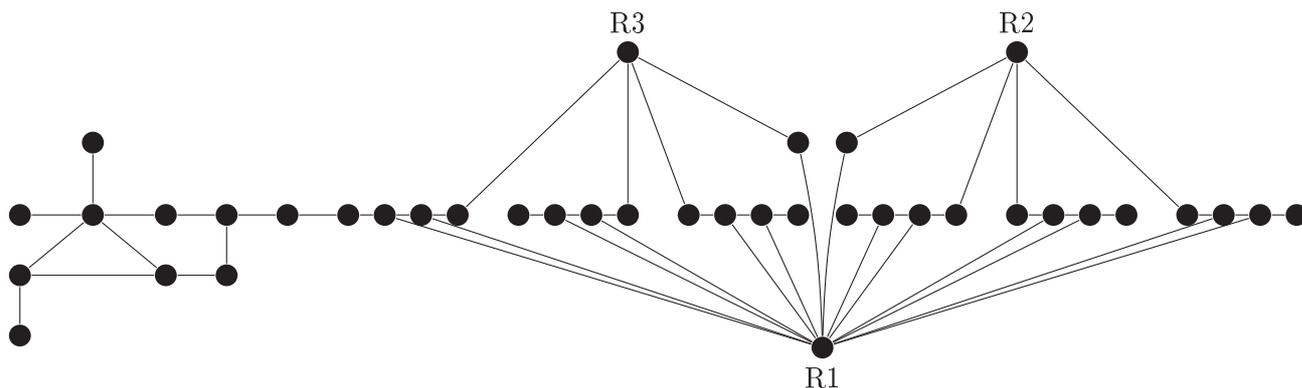


Figure 3.6 – Instance prouvant que les problèmes MBVST, MDST et MLST ne sont pas liés

3.3 Hiérarchie de recouvrement : une alternative à la structure d'arbre pour minimiser le nombre de sommets de branchement dans les réseaux optiques

L'absence de contraintes sur le degré des sommets dans la solution du problème MBVST induit la réalisabilité de toute instance connexe. En effet, il suffit que le graphe soit connexe pour qu'un arbre de recouvrement ayant un minimum de sommets de branchement existe. Cette absence de contraintes sur le degré des sommets rend triviale la recherche d'une hiérarchie de recouvrement ayant un minimum de sommets de branchement. Il suffit de faire un parcours du graphe où chaque sommet peut être visité plusieurs fois. On obtient alors une hiérarchie de recouvrement n'ayant aucun sommet de branchement. Dans la figure 3.1d, on remarque que la hiérarchie de recouvrement H du graphe G ne contient aucun sommet de branchement, car le sommet central a a été « visité » deux fois (deux occurrences de a sont présentes dans la hiérarchie).

Minimiser le nombre de sommets de branchement dans l'arbre de recouvrement d'un graphe modélisant un réseau optique permet, certes, de minimiser le nombre de splitters nécessaires au routage optique et de réduire la division de la lumière, mais ceci ne permet pas toujours d'être en deçà du nombre de splitters que peut effectivement contenir le réseau. En effet, si un réseau peut contenir au plus k splitters (pour des raisons de coûts) et que tout arbre de recouvrement du graphe modélisant le réseau contient au moins $k + 1$ sommets de branchement, alors le routage multicast ne peut être pleinement réalisé sur ce réseau.

On introduit deux nouveaux problèmes qui permettent de prendre en considération le nombre de splitters pouvant être présents dans le réseau et qui minimise le coût total de la communication. Le premier se base sur la structure d'arbre de recouvrement et le deuxième sur la structure de hiérarchie de recouvrement.

Le problème BVCMST est défini comme suit :

Problème : Branch Vertices Constrained Minimum Spanning Tree - BVCMST
Données : Un graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$, une fonction $c : E_G \rightarrow \mathbb{R}_*^+$ associant un coût $c(e)$ à chaque arête $e \in E_G$ et un entier positif $k \geq 0$
Résultat : un arbre de recouvrement T du graphe G ayant au plus k sommets de branchement.
Objectif : Minimiser le coût de T .

Le problème BVCMST est défini comme suit :

Problème : Branch Vertices Constrained Minimum Spanning Hierarchy - BVCMSTH
Données : Un graphe connexe non-orienté arêtes pondérées $G = (V_G, E_G)$, une fonction $c : E_G \rightarrow \mathbb{R}_*^+$ associant un coût $c(e)$ à chaque arête $e \in E_G$ et un entier positif $k \geq 0$
Résultat : Une hiérarchie de recouvrement H du graphe G ayant au plus k sommets de branchement.
Objectif : Minimiser le coût de H .

Il est clair que le problème BVCMST est NP-difficile, car si l'on fixe la borne sur le nombre de sommets de branchement k à zero, alors trouver une solution réalisable revient à trouver un chemin hamiltonien dans le graphe. Le problème BVCMST est en fait NP-difficile pour toute valeur de k (preuve identique à celle présentée dans la section 1.4.1.2).

Trouver un arbre de recouvrement de coût minimum d'un graphe ayant au plus k sommets de branchement est donc NP-difficile. Remarquons que trouver un arbre de recouvrement réalisable l'est également. En effet, dans un graphe contenant des arêtes de coûts identiques, un arbre de recouvrement réalisable pour le problème BVCMST est aussi optimal. Par conséquent, le fait que ce problème soit NP-difficile n'est pas dû à la minimisation du coût mais à la contrainte sur le nombre de sommets de branchement. Une question se pose alors : pour une instance et un entier k donnés, la donnée des k sommets de branchement de la solution optimale permet-elle de trouver cette solution en temps polynomial ?

Considérons l'instance présentée dans la figure 3.7 et fixons la borne sur le nombre de sommets de branchement à k . Il est indispensable que les sommets a^1, a^2, \dots, a^k soient des sommets de branchement. Sachant que les k cliques admettent un chemin hamiltonien, toutes les solutions optimales pour le problème BVCMST contiennent comme seuls sommets de branchement les sommets a^1, a^2, \dots, a^k . Trouver une solution optimale revient à trouver un chemin hamiltonien de coût minimum dans k cliques et ayant respectivement pour sommets de départ les sommets b^1, b^2, \dots, b^k . Il est alors NP-difficile de trouver une solution optimale au problème BVCMST même si l'on connaît les sommets de branchement appartenant à celle-ci.

Par ailleurs, trouver une hiérarchie de recouvrement d'un graphe ayant au plus k sommets

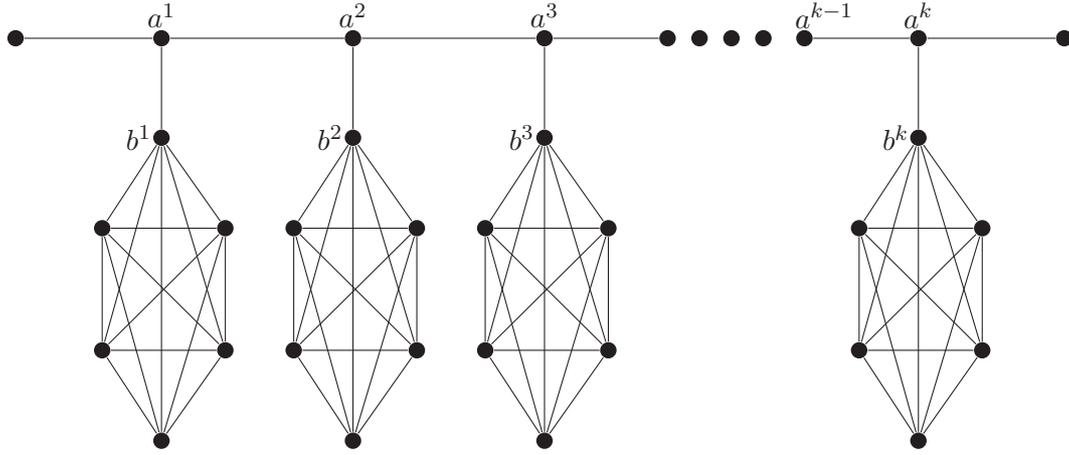


Figure 3.7 – Réduction du problème BVC MST au problème chemin hamiltonien de coût minimum

de branchement est polynomial (il suffit de faire un parcours du graphe). Par contre, trouver une hiérarchie de recouvrement d'un graphe de coût minimum ayant au plus k sommets de branchement est NP-difficile. En effet :

Lemme 3.3.1. *Le problème BVCMSH est NP-difficile pour tout $k \geq 0$.*

Démonstration. Pour $k = 0$: Soit $G = (V_G, E_G)$ un graphe connexe arête pondéré. Supposons que le coût des arêtes soit uniformément égal à 1. Trouver une hiérarchie de recouvrement de G de coût $|V_G| - 1$ et n'ayant aucun sommet de branchement équivaut à trouver un chemin hamiltonien dans G .

Pour $k \geq 1$: Soit $G = (V_G, E_G)$ un graphe connexe arête pondéré et un sommet $v \in V_G$ choisi arbitrairement. Nous faisons $2 \cdot k$ copies de G , toutes les arêtes ont un coût de 1 dans ces copies. Nous construisons une chaîne formée de k sommets connectés par des arêtes de coût 1. Chacun de ces sommets est connecté au sommet v de deux copies distinctes de G grâce à deux arêtes de coût M , tel que $M > (2 \cdot k \cdot |V_G|)$. Notons G' le graphe obtenu. Dans toute hiérarchie optimale pour le problème BVCMSH avec une borne k , tous les sommets de la chaîne sont des sommets de branchement. Sinon, les arêtes de coût M doivent être utilisées plus d'une fois. Or, le coût d'une hiérarchie ayant au moins $k+1$ arêtes de coût c est strictement supérieur à celui d'une hiérarchie ayant k arêtes de coût M . Il existe donc une solution de coût $(k-1) + 2 \cdot (k \cdot M + |V| - 1)$ si et seulement s'il existe un chemin hamiltonien de G ayant v comme extrémité. \square

3.4 Propriétés des solutions optimales du problème BVCMSH

Nous proposons ici des propriétés des solutions optimales du problème BVCMSH :

Lemme 3.4.1. *Les sommets de branchement appartenant à une hiérarchie $H^* = (Q_{H^*}, h_{H^*}, G)$ optimale pour le problème BVCMSH ne sont jamais dupliqués.*

Démonstration. Procédons par contradiction. Supposons qu'il existe une hiérarchie optimale $H^* = (Q_{H^*}, h_{H^*}, G)$ ayant un sommet de branchement v dupliqué deux fois (v^1 et v^2). Nous choisissons une racine arbitraire $r \in V_{Q_{H^*}}$ et nous enracinons Q_{H^*} en r . Nous pouvons supprimer v^2 et relier ses successeurs à v^1 . L'arc qui reliait v^2 à son prédécesseur est supprimé. Nous obtenons donc une hiérarchie réalisable H^{**} ayant un coût strictement inférieur à celui de H^* (H^{**} contient une arête de moins que H^*). Ceci contredit l'optimalité de H^* . \square

Théorème 3.4.2. *Si une hiérarchie $H^* = (Q_{H^*}, h_{H^*}, G)$ optimale pour le problème BVCMSH est telle que $s(H^*) < k$, alors $\forall (a, b) \in V_G^2$, il existe au plus une arête entre les occurrences de a et les occurrences de b dans Q_{H^*} .*

Démonstration. Procédons par contradiction. Supposons qu'il existe une hiérarchie optimale $H^* = (Q_{H^*}, h_{H^*}, G)$ telle que $s(H^*) < k$ et qu'il existe deux sommets a et b de V_G , il existe deux arêtes entre les occurrences de a et les occurrences de b dans Q_{H^*} .

Si a et b étaient tous les deux des sommets de branchement alors d'après la proposition 3.4.1, ils n'ont qu'une seule occurrence dans Q_{H^*} et il ne peut donc y avoir qu'une seule arête entre eux.

Si un seul des deux sommets a ou b est un sommet de branchement. Supposons sans perte de généralité que le sommet de branchement soit a . Le sommet a n'est pas dupliqué (proposition 3.4.1). Supposons que les deux duplications de l'arête $\{a, b\}$ soient $\{a, b^1\}$ et $\{a, b^2\}$. Supprimons l'arête $\{a, b^2\}$ et le sommet b^2 et relier son voisin (s'il en a) au sommet b^1 . Le sommet b^1 peut éventuellement devenir sommet de branchement. Nous obtenons une hiérarchie réalisable H^{**} ayant au plus k sommets de branchement et ayant un coût strictement inférieur à celui de H^* (H^{**} contient une arête de moins que H^*). Ceci contredit l'affirmation de départ.

Si ni a ni b ne sont des sommets de branchement. Choisissons arbitrairement une racine $r \in E_{H^*}$ et orientons Q_{H^*} de r vers les feuilles. Nous distinguons alors trois cas :

- Une occurrence de a a comme seuls voisins deux duplications de b . Le même raisonnement s'applique.
- Les deux duplications de $\{a, b\}$ relient deux duplications distinctes a^i et a^j de a à deux duplications b^i et b^j de b et ces deux arcs sont respectivement orientés de a^i vers b^i et de a^j vers b^j . Les arcs (a^i, b^i) et (a^j, b^j) connectent trois composantes connexes de H^* . Notons les $H1, H2$ et $H3$ ($H1(a^i, b^i)H2(a^j, b^j)H3$). Nous supprimons les sommets a^j et b^j et nous relier $H3$ à b^i . Le sommet b^i devient alors sommet de branchement.
- Les deux duplications de l'arête $\{a, b\}$ relient deux duplications a^i et a^j de a à deux duplications b^i et b^j de b et ces deux arcs sont respectivement orientés de a^i vers b^i et de b^j vers a^j . Ces deux arcs connectent trois composantes connexes de H^* . Notons les $H1, H2$ et $H3$ ($H1(a^i, b^i)H2(b^j, a^j)H3$). Nous supprimons a^j et b^j et nous relier $H3$ à a^i . Le sommet a^i devient alors un sommet de branchement.

Dans les trois cas précédents, nous avons construit une hiérarchie réalisable H^{**} ($s(H^{**}) = s(H^*) + 1 \leq k$) ayant un coût strictement inférieur à celui de H^* (H^{**} contient une arête de moins que H^*). Ceci contredit l'affirmation de départ. \square

Notre objectif à ce niveau était de trouver une borne supérieure sur le nombre de duplications des sommets dans une solution optimale pour le problème BVCMSH ayant k sommets de branchement (une borne supérieure plus intéressante que $|V_G|$ moins une constante). Cependant, il existe un exemple qui montre que cette borne peut ne dépendre que de $|V_G|$. En effet, soit $G = (V_G, E_G)$ un graphe composé d'un sommet v ayant $|V_G| - 1$ voisins de degré 1 (une étoile). Pour $k = 0$, la hiérarchie optimale pour le problème BVCMSH contient nécessairement $|V_G| - 2$ occurrences du sommet v .

Considérons le graphe présenté dans la figure 3.8. Supposons que le sommet c ait un degré $d_G(c)$ très grand et que toutes les arêtes reliant le sommet c à ses voisins aient un coût égal à 1 (sauf l'arête le reliant à la chaîne qui a un coût de M). Les autres arêtes du graphe ont un coût $M > d_G(c)$. La hiérarchie de recouvrement optimale pour le problème BVCMSH contient alors k sommets de branchement et le sommet c est dupliqué $d_G(c) - 2$ fois. Par conséquent, le nombre de duplications d'un sommet dans une hiérarchie optimale pour le problème BVCMSH peut tendre vers le nombre de sommets du graphe quelle que soit la valeur de la borne k .

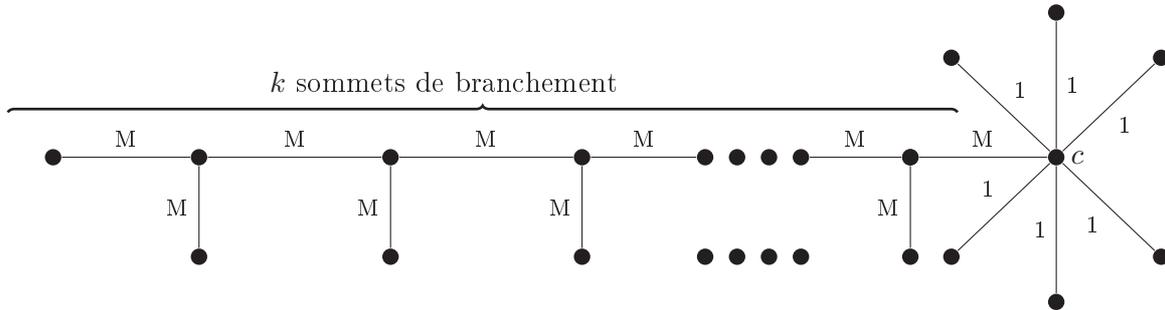


Figure 3.8 – Instance montrant que la borne sur le nombre de duplications des sommets ne dépend pas uniquement de k

3.5 Résolution exacte du problème BVCMSH

Le modèle proposé pour résoudre le problème BVCMSH est identique au modèle proposé pour résoudre le problème DCMSH dans le chapitre 2. Cependant, le nombre de duplications des arêtes (qui induit dans notre modèle le nombre de duplications des sommets) n'est cette fois borné que par $\beta = |V_G| - 1$. Ceci a pour effet d'augmenter le nombre de variables et de contraintes du programme linéaire du problème BVCMSH par rapport à celui du DCMSH (ce dernier ayant une borne sur le nombre de duplications des sommets de $\frac{|V_G| - 2}{B - 2}$). Notons cependant que des inégalités valides sont introduites afin de réduire le temps de résolution.

3.5.1 Formulation ILP

Les paramètres de graphe et les variables du programme linéaire sont également identiques à ceux du problème DCMSH. Une variable binaire $Y(u)$ est ajoutée pour tout $u \in V_G$. Cette variable est égale à 1 si le sommet u est un sommet de branchement, 0 sinon.

L'objectif consiste également à minimiser le coût total des arêtes présentes dans la hiérarchie de recouvrement. La formulation de la fonction objective est donc identique à celle du problème DCMSH (2.12). Afin de garantir la connexité de la hiérarchie, on a introduit dans 2.3.2.2 des contraintes de flot. Ces contraintes sont également ajoutées à l'ILP. Seules les contraintes de degré sont différentes. Celles-ci se présentent comme suit :

Contraintes de degré :

$$\sum_{v \in Out(s)} \sum_{i=1}^{\beta} L_i(s, v) - \sum_{v \in In(s)} \sum_{i=1}^{\beta} L_i(v, s) - 2 \leq (|V_G| - 1) * Y(s) \quad (3.1)$$

$$\sum_{v \in Out(u)} \sum_{i=1}^{\beta} L_i(u, v) - \sum_{v \in In(u)} \sum_{i=1}^{\beta} L_i(v, u) \leq (|V_G| - 1) * Y(u), \quad \forall u, v \in V_G \setminus \{s\} \quad (3.2)$$

Les contraintes (3.1) et (3.2) assurent que si $Y(u)$ est égal à 0 alors pour chaque sommet exceptée la source, le nombre des successeurs est au plus égal au nombre des prédécesseurs. Sinon, le nombre de successeurs peut être aussi grand que nécessaire (cependant borné par $|V_G| - 1$).

$$\sum_{u \in V_G} Y(u) \leq k \quad (3.3)$$

La contrainte (3.3) assure que le nombre de sommets de branchement est inférieur ou égal à la borne k .

Inégalités valides :

$$L_i(u, v) \leq 1 - Y(u), \quad \forall v \in V_G, \quad \forall v \in Out(u), \quad \forall i > 2 \quad (3.4)$$

La contrainte (3.4) permet de prendre en considération la proposition 3.4.1. En effet, elle assure que chaque sommet de branchement ne soit pas dupliqué. Ainsi, la première occurrence de u est indépendante de la valeur de $Y(u)$. Par contre, les duplications suivantes ne sont autorisées que si $Y(u)$ est égal à 0.

$$L_i(v, u) \leq 2 - Y(v) - Y(u), \quad \forall u \in V_G, \quad \forall v \in In(u), \quad \forall i \geq 2 \quad (3.5)$$

La contrainte (3.5) permet de prendre en considération la proposition 2.3.5. Elle assure ainsi que si les sommets $\forall v, u \in V_G$ sont des sommets de branchement, alors il y aura au plus une arête entre les occurrences de v et les occurrences de u .

3.5.2 Résultats expérimentaux

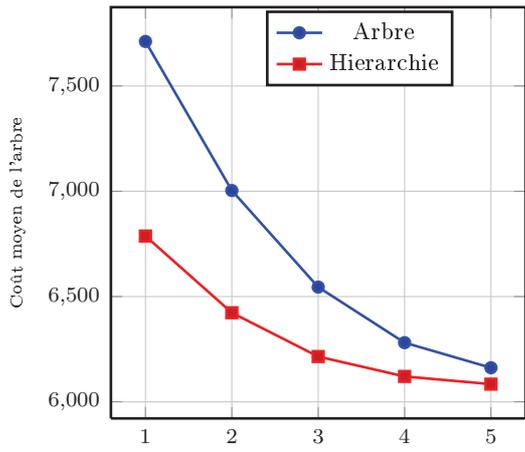
Dans cette section, nous montrons que les hiérarchies de recouvrement peuvent être une alternative aux arbres de recouvrement pour les problèmes de recherche de structure de recouvrement de coût minimum avec une borne sur le nombre de sommets de branchement. Pour cela, des simulations sont réalisées pour comparer les deux structures. Les formulations en *PLNE* des problèmes BVC MST et BVCMSH sont implémentées en langage C et ils sont résolus avec le solveur de programmes linéaires CPLEX sur des instances générées aléatoirement en utilisant le générateur NetGen [KNS74].

Le protocole expérimental est le même que celui du chapitre 2. Le nombre de duplications des arêtes impliquant le nombre de duplications des sommets est borné par $|V_G| - 1$. Nous nous limitons à des graphes d'au plus 60 sommets. Nous considérons cinq valeurs de la borne supérieure sur le nombre de sommets de branchement $k \in \{1, 2, 3, 4, 5\}$.

k	1	2	3	4	5
$ V_G = 20$					
Coût moyen BVC MST	7711.31	7003.42	6544.86	6280.93	6084.01
Coût moyen BVCMSH	6787.38	6423.11	6215.03	6120.51	5925.82
Amélioration	11.98%	8.28%	5.03%	2.55%	1.26%
$ V_G = 30$					
Coût moyen BVC MST	13075.17	11041.47	10045.07	9500.05	9214.98
Coût moyen BVCMSH	10537.10	9645.13	9005.65	8900.74	8850.68
Amélioration	19.41%	12.64%	10.35%	6.31%	3.95%
$ V_G = 40$					
Coût moyen BVC MST	19921.41	14703.60	13588.13	12602.45	12079.18
Coût moyen BVCMSH	15403.22	12306.91	11585.28	11298.09	11168.40
Amélioration	22.68%	16.30%	14.74%	10.35%	7.01%
$ V_G = 50$					
Coût moyen BVC MST	22501.89	17942.16	15871.91	14833.04	14020.51
Coût moyen BVCMSH	16811.14	14226.35	13151.46	12897.32	12582.00
Amélioration	25.29%	20.71%	17.14%	13.04%	10.30%
$ V_G = 60$					
Coût moyen BVC MST	25438.00	20542.36	19032.39	17632.11	16599.78
Coût moyen BVCMSH	18536.67	15813.50	15262.07	14865.63	14581.24
Amélioration	27.13%	23.02%	19.81%	15.69%	12.16%

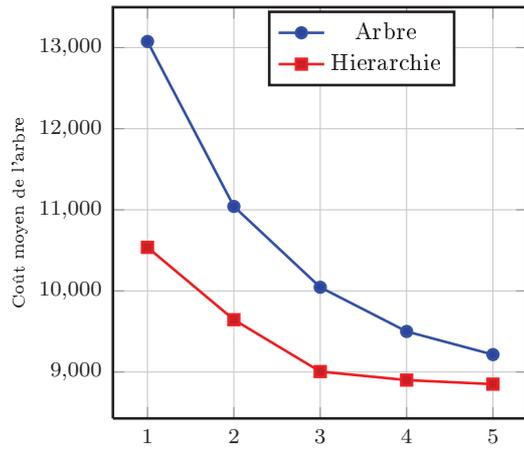
TABLE 3.1 – Les coûts des hiérarchies comparés aux coût des arbres

Les résultats numériques sont présentés dans le tableau 3.1. L'amélioration du coût par les hiérarchies est plus visible dans les figures 3.9a à 3.9e. Ces figures représentent le coût moyen des arbres optimaux et des hiérarchies optimales pour $|V_G| \in \{20, 30, 40, 50, 60\}$. On peut ainsi remarquer que l'amélioration augmente en fonction de l'augmentation de la taille des instances et diminue quand la borne supérieure sur le nombre de sommets de branchement augmente. Le pourcentage moyen de l'amélioration du coût dépasse 25% quand $k = 1$ et $|V_G| = 60$. Le pour-



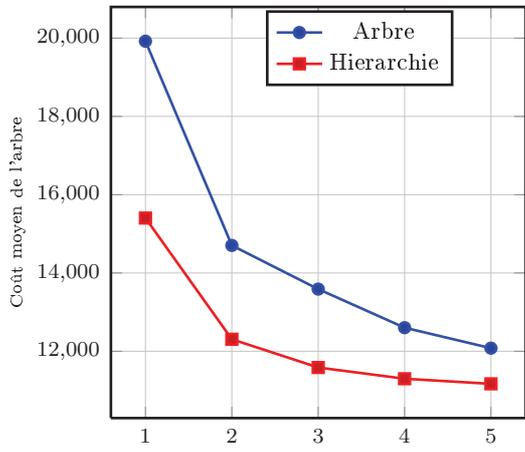
Nombre de sommets de branchement autorisés k

(a) $|V_G| = 20$



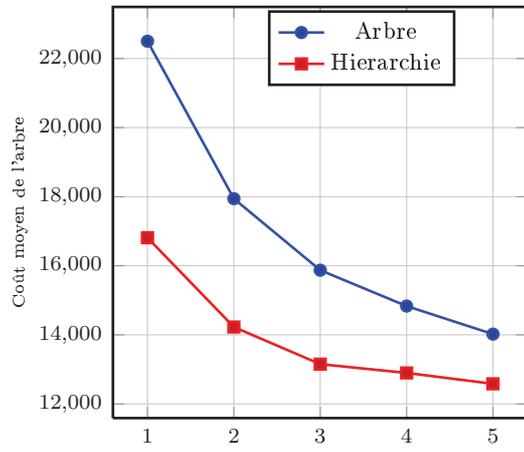
Nombre de sommets de branchement autorisés k

(b) $|V_G| = 30$



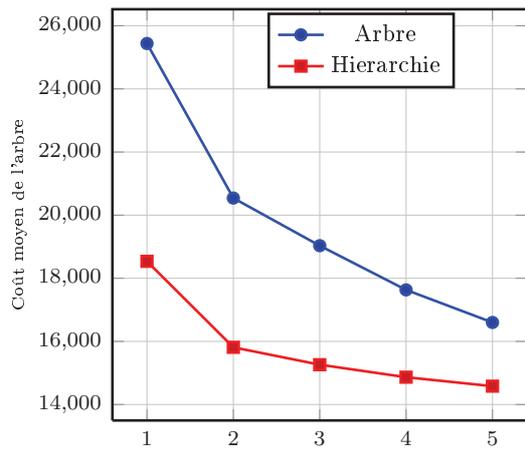
Nombre de sommets de branchement autorisés k

(c) $|V_G| = 40$



Nombre de sommets de branchement autorisés k

(d) $|V_G| = 50$



Nombre de sommets de branchement autorisés k

(e) $|V_G| = 60$

Figure 3.9 – Arbres de recouvrement optimaux versus hiérarchies de recouvrement optimales

centage d'amélioration du coût augmente en fonction de la taille du graphe. Quand la taille du graphe est grande, la probabilité qu'un sommet soit dupliqué dans la hiérarchie augmente. Ceci accroît la probabilité d'amélioration du coût. Cependant, l'amélioration diminue en fonction de la réduction de la borne k . En effet, plus k est grand et plus le coût de la solution optimale pour le problème BVC MST s'approche de celui de l'arbre de coût minimum, diminuant ainsi l'intérêt des hiérarchies.

3.6 Résolution approchée du problème BVCMSH

Pour démontrer l'avantage des hiérarchies de recouvrement par rapport aux arbres, nous avons calculé et comparé, dans la section précédente, les coûts des solutions optimales des problèmes BVC MST et BVCMSH pour des graphes aléatoires. Nous avons ainsi montré que l'amélioration du coût peut être significative en fonction de la borne k et de la taille du graphe. Nous avons également montré que toute instance connexe contient une solution réalisable pour le problème BVCMSH, quel que soit k . Ce n'est pas le cas pour le problème BVC MST. Ce comparatif concerne le volet "résolution exacte" des deux problèmes. Qu'en est-il de la résolution approchée ?

Le problème BVC MST est non approximable à un facteur constant. Il suffit de fixer la borne sur le nombre de sommets de branchement à zero pour s'en convaincre. En effet, dans ce cas, trouver un algorithme d'approximation pour le problème BVC MST garantissant un ratio constant R revient à trouver un algorithme d'approximation pour le problème du voyageur du commerce garantissant le même ratio. Or il a été démontré que le problème voyageur du commerce (STP) est non approximable [CLRS02]. Le problème BVCMSH est, quant à lui, approximable. En effet, il suffit de faire un parcours en largeur de l'arbre de recouvrement de coût minimum de graphe pour obtenir une hiérarchie réalisable avec un coût inférieur ou égal à deux fois le coût de la hiérarchie optimale. Dans cette section, nous donnons un algorithme permettant d'approximer le problème BVCMSH avec un ratio $\rho = 2 \cdot (1 - \frac{k}{n})$.

Notre algorithme contient 6 étapes qui se présentent comme suit :

1. Nous considérons comme point de départ de notre algorithme (borne inférieure) un arbre de coût minimum. Notons que cet arbre peut contenir un nombre de sommets de branchement strictement supérieur à la borne k .
2. Dans le cas où l'arbre de coût minimum respecte la borne k , cet arbre est considéré comme une solution à notre problème.
3. Dans le cas contraire, nous contractons toutes les chaînes parfaites présentes dans l'arbre (définition d'une chaîne parfaite, sous-section 3.6.2).
4. Nous décomposons cet arbre en un ensemble d'étoiles en utilisant l'algorithme 1 présenté dans le chapitre deux.

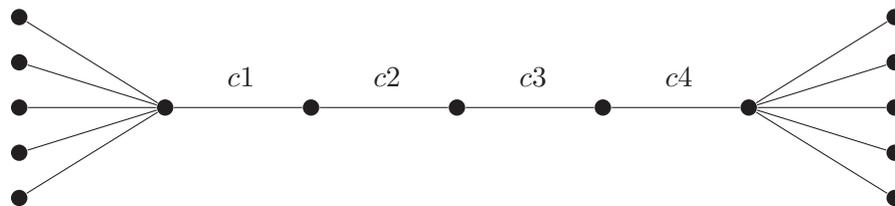
5. Nous ne modifions pas les k étoiles de plus grand coût dont le sommet central est un sommet de branchement ; par contre, nous parcourons chacune des autres étoiles en utilisant une hiérarchie de recouvrement.
6. Enfin, nous construisons une hiérarchie composée de toutes les "sous-hiérarchies" couvrant les étoiles.

3.6.1 Première étape : Construction d'un arbre de recouvrement de coût minimum

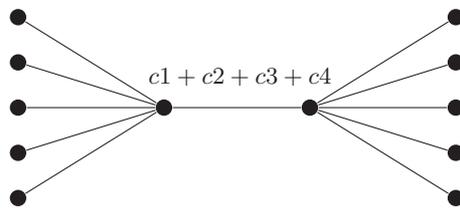
Pareillement que pour l'heuristique que nous proposons pour résoudre le problème DCMSH, nous posons ici le coût de l'arbre de recouvrement de coût minimum comme une borne inférieure pour estimer le coût de notre solution approchée. La première étape de notre algorithme est donc de calculer un tel arbre. Le calcul est polynomial et peut être réalisé grâce à l'algorithme de Prim [Pri57].

3.6.2 Deuxième étape : Contraction des chaînes qui composent le MST

Dans un graphe donné, toute chaîne de longueur au moins égal à deux ayant des sommets internes de degré deux et ses deux extrémités de degré différent de deux est dite chaîne parfaite. Après la construction de l'arbre de coût minimum, nous contractons toutes les chaînes parfaites : chaque chaîne parfaite est remplacée par une arête entre ses deux extrémités de coût égal à la somme des coûts des arêtes supprimées. Nous obtenons ainsi un arbre T' ayant un coût égal à celui du MST T . Notons que cette opération ne modifie pas le degré des sommets qui sont des extrémités dans la chaîne parfaite. La figure 3.10 illustre cette contraction.



(a) Arbre T



(b) La seule chaîne parfaite de T est remplacée par une seule arête qui a pour coût la somme des coûts des arêtes qui composent la chaîne

Figure 3.10 – Contraction des arêtes d'une chaîne

3.6.3 Troisième étape : Décomposition en étoiles de l'arbre de recouvrement de coût minimum contracté T'

Nous pouvons décomposer l'arbre de recouvrement en étoiles en utilisant l'algorithme 1 proposé dans le chapitre 2 et en prenant comme sommet de départ un sommet de branchement s'il existe (sinon, le MST est une chaîne et c'est donc une solution optimale). À l'issue de cette étape, l'arbre de recouvrement sera décomposé en étoiles. La seule différence due à la contraction est que chaque étoile a nécessairement comme sommet central un sommet de branchement.

3.6.4 Quatrième étape : Recouvrement des étoiles par les hiérarchies

L'arbre T' est maintenant décomposé en étoiles. Tout arbre $T = (V_T, E_T)$ contient au plus $\frac{n}{2}$ sommets de branchement [CCR14]. Chaque étoile de l'arbre T' contient comme sommet central un sommet de branchement. Le nombre d'étoiles de l'arbre T' est donc borné par $\frac{n}{2}$. Les k étoiles de plus grand coût sont conservées dans la hiérarchie approchée. Les autres étoiles (au plus $\frac{n}{2} - k$) sont parcourues avec un parcours en profondeur.

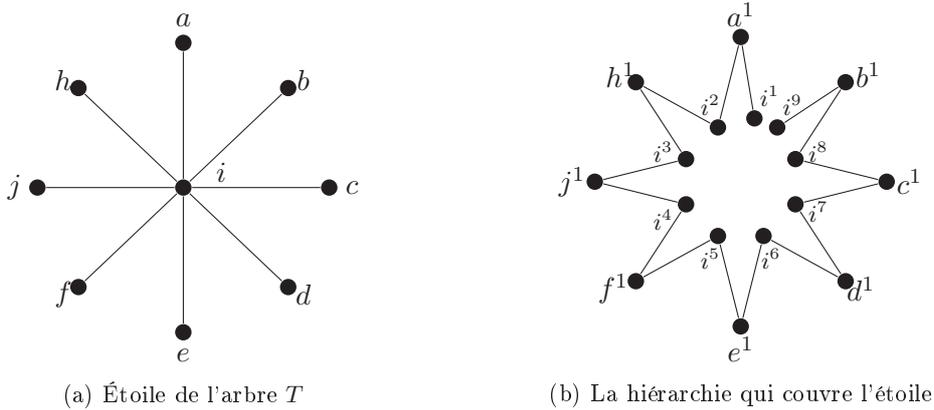


Figure 3.11 – Recouvrement d'une étoile par une hiérarchie

La figure 3.11 montre le parcours d'une étoile. Ainsi, chaque arête de l'étoile est utilisée (dupliquée) exactement deux fois dans la hiérarchie. Le coût de la hiérarchie de recouvrement de l'étoile est égal à deux fois le coût de l'étoile. Notons que les deux feuilles de la hiérarchie sont deux duplications du sommet central de l'arbre, ces duplications ne sont pas nécessaires car le sommet central est déjà couvert. Elles sont néanmoins utiles pour connecter les sous-hiérarchies couvrant les étoiles (étape 5).

Une fois les étoiles de plus faible coût parcourues par des hiérarchies, on connecte ces hiérarchies et les k étoiles de plus fort coût afin de former une seule hiérarchie H contenant k sommets de branchement. Notons $C(AD)$ le coût de la partie dupliquée dans H et posons $n = |V_T|$. Ce coût peut être borné par $C(AD) \leq \frac{\frac{n}{2} - k}{\frac{n}{2}} \cdot C(T)$. Ainsi nous obtenons : $\frac{C(H)}{C(T)} \leq 2 \cdot (1 - \frac{k}{n})$.

3.6.5 Cinquième étape : Connexion des hiérarchies de recouvrement des étoiles

Il est nécessaire de reconnecter les hiérarchies de recouvrement des étoiles afin de former une seule hiérarchie connexe. Une connexion est nécessaire si une feuille dans une étoile coïncide avec le sommet central de l'autre étoile. Notons S et S' deux étoiles telles que c et c' soient leurs sommets centraux et que le sommet c' soit une feuille dans l'étoile S . Notons par H et par H' les hiérarchies de recouvrement de S et de S' , respectivement. Pour des raisons de clarté de la preuve, les étiquettes des sommets dans les hiérarchies ne correspondront pas à celles du graphe initial. Notons par h_i la i -ème occurrence du sommet c dans H et h'_i la i -ème occurrence du sommet c' dans H' . Notons par f_1 la seule occurrence du sommet c' dans la hiérarchie H .

Rappelons que si le sommet central d'une étoile est un sommet de branchement dans la solution, alors la hiérarchie de recouvrement de cette étoile est identique à l'étoile. Plusieurs cas peuvent se présenter :

1. Les sommets h_1 et h'_1 sont des sommets de branchement : dans ce cas, les deux hiérarchies H et H' sont reliées de la même manière qu'étaient reliées les étoiles S et S' avant la décomposition. Ainsi, nous fusionnons le sommet f_1 et le sommet h_1 (figure 3.12).

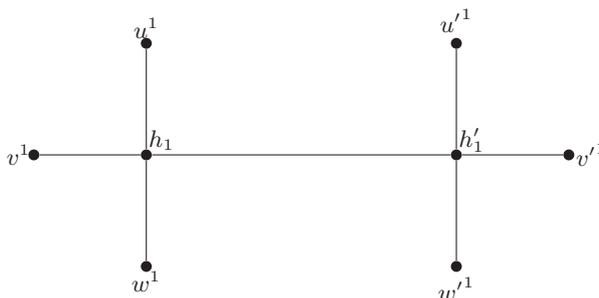


Figure 3.12 – Les sommets h_1 et h'_1 sont des sommets de branchement

2. Le sommet h_1 est un sommet de branchement et le sommet h'_1 n'est pas un sommet de branchement : le sommet f_1 peut être fusionné avec h'_1 sans créer de sommets de branchement, car h'_1 et f_1 ont un degré 1.

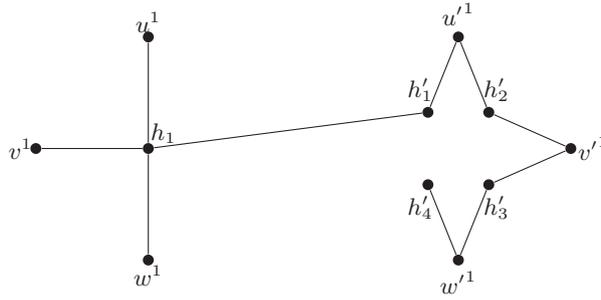


Figure 3.13 – Seul h_1 est un sommet de branchement

3. Le sommet h_1 n'est pas un sommet de branchement et le sommet h'_1 est un sommet de branchement : le sommet h'_1 est alors fusionné avec le sommet f_1 . Ceci ne crée pas de sommet de branchement supplémentaire car le sommet h'_1 est déjà un sommet de branchement.

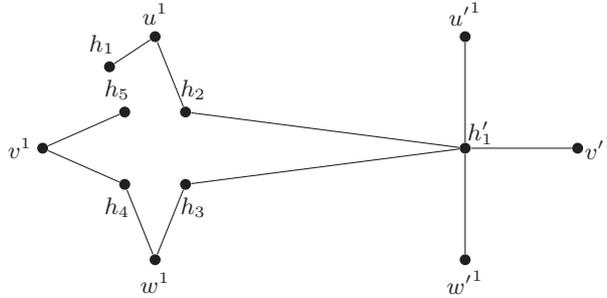


Figure 3.14 – Seul h'_1 est un sommet de branchement

4. Aucun des deux sommets h_1 et h'_1 ne sont des sommets de branchement : notons h_i et h_{i+1} les deux occurrences du sommet central c reliées au sommet f_1 . Nous supprimons le sommet f_1 ainsi que les deux arêtes qui le reliaient respectivement à h_i et h_{i+1} . Nous ajoutons deux arêtes qui relient h_i à h'_1 et h'_{i+1} à la dernière occurrence du sommet c_i dans H , ce qui ne modifie pas le coût de la solution et n'ajoute pas de sommet de branchement.

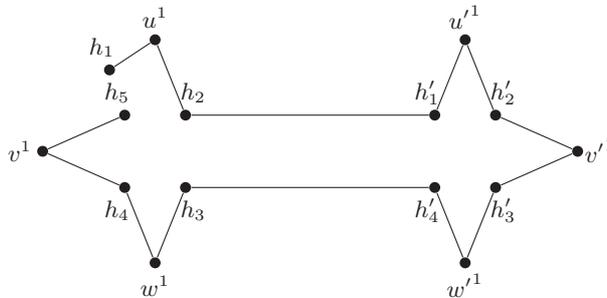


Figure 3.15 – Aucun des deux sommets h_1 et h'_1 ne sont des sommets de branchement

3.7 Conclusion

Nous avons commencé par prouver que les problèmes MBVST, MDST et MLST ne sont pas liés [MDM13b]. Nous avons, par la suite, introduit le problème de recherche de hiérarchie de recouvrement de coût minimum tel que le nombre de ses sommets de branchement est borné supérieurement par un entier k et nous avons prouvé que ce nouveau problème est NP-difficile. Une formulation en programme linéaire en nombres entiers a été donnée. Ce problème a été résolu de manière exacte en utilisant le solveur CPLEX sur des jeux de graphes aléatoires. Le coût des solutions optimales du problème DCMSH et celui des solutions optimales du problème DCMST ont été comparés. Dans certains cas, l'amélioration du coût peut atteindre 27% [MDM13a]. Alors que le problème MBVST est non-approximable à facteur constant, nous avons présenté un algorithme qui retourne une hiérarchie d'un coût au plus $2 \cdot (1 - \frac{k}{n})$ de celui de la hiérarchie optimale pour le problème BVCMSH, où k est la borne sur le nombre de sommets de branchement et n la taille de l'instance.

Routage multicast dans les réseaux WDM sans capacité de division

4.1 Introduction

La transmission des messages dans les réseaux tout optique se fait grâce à la technologie du multiplexage en longueur d'onde communément appelée WDM (Wavelength Division Multiplexing). Cette technologie consiste à diviser le spectre optique en plusieurs sous-canaux et à associer à chaque sous-canal une longueur d'onde spécifique [Gre92]. Ainsi, il est possible de faire transiter, dans une même fibre, plusieurs longueurs d'onde en parallèle. Dans les réseaux WDM, une communication de bout en bout entre une paire de nœuds d'accès (source, destination) est mise en œuvre par une séquence de liens logiques. Cette séquence est appelée chemin optique. Un chemin optique est un chemin entre deux nœuds d'accès, où une seule longueur d'onde est utilisée de bout en bout. Les fibres optiques portent le même ensemble de longueurs d'onde et fonctionnent en mode WDM afin de fournir une transmission à haut débit.

Dans les réseaux tout optique, les fonctions de commutation et de routage sont fournies par les brasseurs optiques (OXC) et permettent la mise en œuvre de communications de bout en bout entre les nœuds d'accès. Grâce au démultiplexage du signal optique entrant, un OXC peut commuter chacune des longueurs d'onde d'un port d'entrée vers un port de sortie quelconque. Certains OXC particuliers, équipés de coupleurs optiques, peuvent également diviser une longueur d'onde entrante vers plusieurs ports de sortie simultanément et offrir ainsi un service multicast. Les switches OXC ayant la capacité de diviser une longueur d'onde sont dits Multicast Capable (OXC-MC) et les autres sont dits Multicast Incapable (OXC-MI). Les réseaux tout optique équipés de ces deux types de switch sont des réseaux WDM avec capacité clairsemée de division (sparse splitting) [QJG⁺99]. Les switches OXC-MC ont un coût élevé en raison d'un procédé de fabrication compliqué et délicat. De plus, la division de la lumière diminue considé-

ablement son niveau d'énergie. Il est donc nécessaire de réduire leur nombre dans les réseaux tout optique.

Dans ce chapitre, nous traitons un cas particulier mais important : nous supposons que le réseau est dépourvu de switch OXC-MC. Nous supposons également que la conversion de longueurs d'onde n'est pas possible dans le réseau. En effet, les convertisseurs de longueur d'onde sont des dispositifs utilisant une technologie très complexe. Ils sont donc coûteux et difficiles à fabriquer [SvSR14, SS12]. Par conséquent, nous supposons qu'ils ne sont pas disponibles dans les réseaux que nous étudions. En l'absence de convertisseurs, une seule longueur d'onde doit ainsi être conservée sur toutes les fibres optiques d'un chemin allant de la source à une destination. Cette contrainte est appelée la contrainte de continuité de la longueur d'onde. De plus, deux chemins optiques partageant au moins une fibre optique commune doivent impérativement utiliser deux longueurs d'onde distinctes [Muk00]. En effet, une fibre optique contient certes plusieurs sous-canaux, mais chaque sous-canal est associé à une longueur d'onde spécifique. Il ne peut donc y avoir deux canaux transportant la même longueur d'onde.

Notons que la majorité des études menées sur le routage optique concernent les réseaux symétriques dans lesquels, si deux sommets sont reliés, ils le sont grâce à deux fibres orientées dans les deux sens opposés. Les réseaux ayant cette particularité sont modélisés par des graphes bi-directionnels [AD00]. Afin de prendre en compte le cas général, nous considérons dans notre étude que les réseaux ne sont pas symétriques. Nous modélisons donc les réseaux optiques par des graphes orientés où chaque arc représente une fibre orientée reliant une paire de switches.

Notre objectif premier est de minimiser la somme des longueurs d'onde utilisées par l'ensemble des chemins optiques allant de la source et atteignant toutes les destinations. Sachant qu'un coût est attribué à chaque fibre, un second objectif consiste à minimiser la somme des coûts des fibres appartenant aux chemins optiques.

4.2 Définition des problèmes

Un réseau optique est donc modélisé par un graphe orienté $G = (V_G, E_G)$ où chaque arc $e \in E_G$ représente une fibre entre deux nœuds (switches) dans le réseau (nous supposons qu'il existe au plus une fibre dans chaque sens entre une paire de nœuds). Une fonction de coût c attribue à chaque arc e un coût $c(e)$. Nous supposons que le réseau est dépourvu de switch OXC-MC. Nous supposons également que toutes les fibres portent le même ensemble W de longueurs d'onde. Nous notons par $RM = (s, D)$ une requête multicast dans laquelle $s \in E_G$ représente la source et $D \subseteq V_G \setminus \{s\}$ représente l'ensemble des destinations. Le problème de routage consiste à calculer une structure de recouvrement (par exemple un ensemble de chemins, un arbre) ayant la source comme racine et couvrant toutes les destinations. Cette structure doit respecter un certain nombre de contraintes optiques [ZP05] :

1. *Contrainte sur la conversion optique* : En l'absence de convertisseurs optiques, chaque

chemin optique de la structure de recouvrement doit utiliser une unique longueur d'onde.

2. *Contrainte de distinction des longueurs d'onde* : Pour chaque longueur d'onde, la fibre optique ne contient qu'un seul sous-canal capable de l'acheminer et chaque sous-canal n'est capable d'acheminer qu'une seule longueur d'onde. Par conséquent, deux chemins optiques utilisant au moins une fibre commune ne peuvent avoir la même longueur d'onde.
3. *Contrainte de degré* : En l'absence de switchs capables de diviser une longueur d'onde entrante vers plusieurs ports de sortie, tous les sommets (sauf la source) composant un chemin de la structure de recouvrement allant de la source à une destination doivent avoir un degré au plus égal à deux. La source, en sa qualité d'émetteur du signal, peut avoir un degré quelconque (trivialement borné par $|D|$).

En se basant sur la définition de la hiérarchie de recouvrement présentée dans le premier chapitre, nous pouvons définir une version « orientée » de celle-ci. En effet, si $H = (Q_H, h_H, G)$ est une hiérarchie de recouvrement telle que G et Q sont orientés alors la hiérarchie H est une hiérarchie de recouvrement orientée (l'homomorphisme, en préservant l'adjacence, prend en compte l'orientation des arcs). Une hiérarchie optique est une hiérarchie orientée ne contenant pas d'arcs dupliqués. Seuls les sommets peuvent l'être. En terme de routage, une telle hiérarchie n'utilise qu'une seule longueur d'onde.

Un chemin élémentaire est un chemin ne passant pas deux fois par un même sommet, c'est-à-dire dont tous les sommets sont distincts. Un chemin simple est un chemin ne passant pas deux fois par un même arc, c'est-à-dire dont tous les arcs sont distincts (notons qu'un chemin élémentaire est nécessairement simple). Nous distinguons deux types de hiérarchie optique :

Définition 4.2.1. *Une Hiérarchie Optique Élémentaire (HOE) est une hiérarchie optique dont le graphe Q (qui est une araignée enracinée et orientée) est composé d'un ensemble de chemins sommets-disjoints allant de la source à chaque feuille, chaque chemin correspondant (via l'application h) à un chemin élémentaire dans G .*

Définition 4.2.2. *Une Hiérarchie Optiques Simple (HOS) est une hiérarchie optique dont le graphe Q est composé d'un ensemble de chemins sommets-disjoints allant de la source à chaque feuille, chaque chemin correspondant à un chemin simple dans G .*

La figure 4.1 montre un exemple de hiérarchie optique pour laquelle le graphe de base Q correspond à deux chemins simples arcs-disjoints dans le graphe G (le premier allant de s à e et le second, de s à h). Ces chemins peuvent donc contenir plusieurs fois un sommet de G . Chaque sommet de G peut ainsi correspondre à plusieurs sommets dans Q et un sommet de Q ne peut correspondre qu'à un seul sommet dans G . Observons également qu'aucun arc du graphe G n'est présent plus d'une fois dans Q .

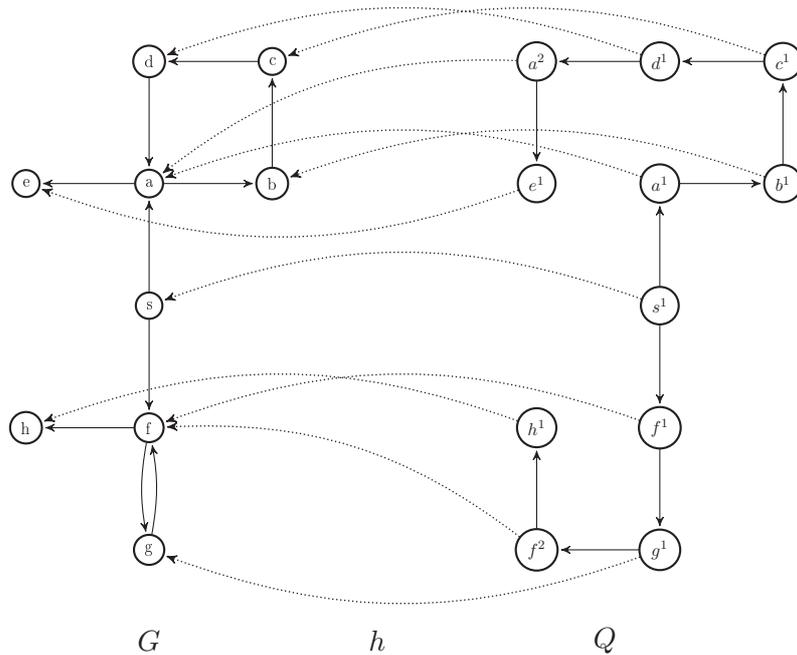


Figure 4.1 – Illustration d'une hiérarchie optique simple

Notons $CO(RM)$ l'ensemble des chemins optiques composant la structure de recouvrement correspondant à une requête multicast RM et par $CO_i, i = 1, 2, 3, \dots, k$ les chemins appartenant à cet ensemble. Une hiérarchie optique, qu'elle soit élémentaire ou simple, utilise une seule longueur d'onde. Toutefois, il est possible qu'une seule hiérarchie (même si elle est simple) ne suffise pas pour couvrir tous les sommets destinations. Ceux-ci sont alors couverts par plusieurs hiérarchies optiques. Par exemple, l'instance illustrée par la figure 4.2 dans laquelle $D = \{b, c\}$, ne peut être couverte par une seule hiérarchie optique car pour atteindre les sommets b et c il faut passer deux fois par l'arc (s, a) .

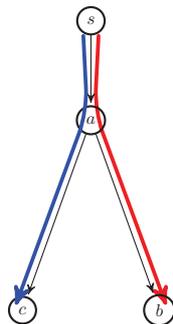


Figure 4.2 – Instance couverte par deux hiérarchies optiques

Le nombre de longueurs d'onde $No.Ondes(RM)$ nécessaire pour le routage multicast tel que la structure de recouvrement contient $|CO| = k$ hiérarchies optiques est inférieur ou égal à k . S'il existe un arc commun aux k hiérarchies optiques, alors le nombre de longueurs d'onde nécessaire

est au moins égal à k . À l’opposé, si elles ne contiennent aucun arc en commun alors le nombre minimum de longueurs d’onde est égal à 1. Le coût total du routage est égal à la somme des coûts des arcs appartenant aux hiérarchies optiques impliquées dans ce routage. Ainsi, nous avons :

$$c(CO(RM)) = \sum_{i=1}^k c(CO_i) = \sum_{i=1}^k \sum_{a \in CO_i} c(a)$$

Nous introduisons un problème générique aux deux types de structures précédemment définis :

Définition 4.2.3. *Étant donné un graphe connexe orienté et arête-pondéré $G = (V_G, E_G)$ et une requête multicast $RM = (s, D)$, le problème Minimum Wavelength Routing Structure (MWRS) consiste à trouver un ensemble de hiérarchies optiques couvrant les sommets de D dans G qui respecte les contraintes optiques décrites précédemment et tel que le nombre de longueurs d’onde soit minimum. Si plusieurs solutions admettent le même nombre minimum de longueurs d’onde, alors nous sélectionnons la solution ayant un coût minimum.*

Soit $D = V_G - \{s\}$ une requête multicast. Si le graphe est hamiltonien à partir de s alors il est possible de couvrir tous les sommets de G en utilisant une seule longueur d’onde. En effet, par définition, le chemin hamiltonien passe une seule fois par tous les sommets de G . Si le graphe n’est pas hamiltonien alors il est parfois possible de couvrir les sommets de G en utilisant une longueur d’onde. En effet, on peut passer plusieurs fois par un sommet sans emprunter les arêtes plus d’une fois, comme le montre la figure 4.3a.

Divisons le sommet a en sommets $a1$ et $a2$ tous deux reliés par un arc allant de $a1$ à $a2$. Les successeurs de a deviennent successeurs de $a2$ et les prédécesseurs de a deviennent prédécesseurs de $a1$ et on ajoute un arc allant du sommet $a1$ au sommet $a2$ (figure 4.3b). Passer m fois par le sommet a du graphe G revient à passer m fois par l’arc $(a1, a2)$ dans le graphe G' . Par conséquent, le graphe G' peut être couvert à partir de la source s en utilisant une seule longueur d’onde si et seulement si le graphe G peut être couvert en passant une seule fois par le sommet a .

Construisons le graphe G'' en appliquant la même opération à tous les sommets de G mise à part la source. Le graphe G'' peut être couvert en utilisant une seule longueur d’onde si et seulement si le graphe G peut être couvert en passant une seule fois par ses sommets (si chemin hamiltonien orienté existe).

Théorème 4.2.1. *Soit k un entier strictement positif. Si $P \neq NP$, alors il n’existe pas d’algorithme polynomial décidant si $W^* \leq k$, où W^* est le nombre minimum de longueurs d’onde nécessaire pour une requête multicast donnée.*

Démonstration. Nous réduisons le problème chemin hamiltonien orienté [GJ79] au problème MWRS. Considérons un graphe connexe orientée et arête-pondéré $G = (V_G, E_G)$. Nous construisons le graphe $G' = (V_{G'}, E_{G'})$ tel que chaque sommet de $v \in V_G$ correspond à deux sommets $v1$ et $v2$ dans $V_{G'}$. Chaque paire de sommets est reliée par un arc allant de $v1$ vers $v2$. Chaque

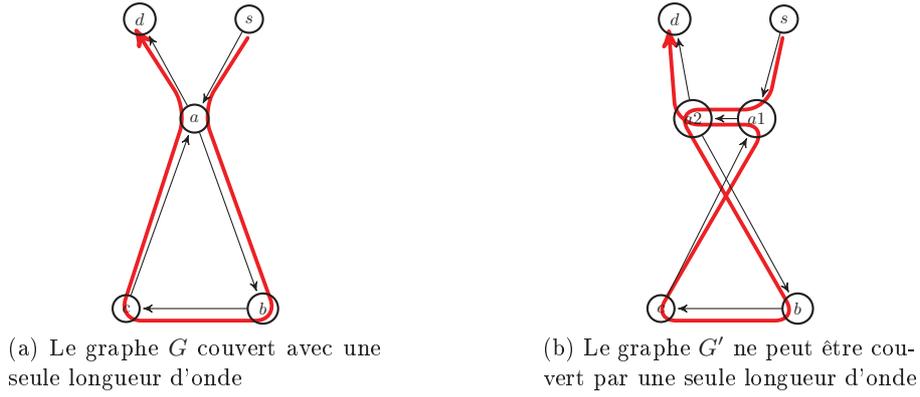


Figure 4.3 – Relation entre le problème MWRS et le problème chemin hamiltonien orienté.

prédécesseur de v devient un prédécesseur de v_1 et chaque successeur de v devient un successeur de v_2 . Nous construisons le graphe H en réalisant k copies du graphe G' , $G'_1 = (V_{G'_1}, E_{G'_1})$, $G'_2 = (V_{G'_2}, E_{G'_2})$, ..., $G'_k = (V_{G'_k}, E_{G'_k})$ et en ajoutant deux nouveaux sommets s et z reliés par l'arc (s, z) . Le sommet s est considéré comme étant la source du routage multicast. Nous ajoutons une arête allant de z vers le sommet v_1 de chaque duplication du graphe G' (voir figure 4.4). Enfin, posons $D = V_{G'_1} \cup V_{G'_2} \cup \dots \cup V_{G'_k}$. Nous constatons alors que H admet un ensemble de hiérarchies optiques couvrant les sommets destinataires et utilisant $W^* \leq k$ longueurs d'onde si et seulement si le graphe G contient un chemin hamiltonien orienté. \square

Notons que l'ensemble des solutions minimisant le nombre de longueurs d'onde peut contenir uniquement des hiérarchies optiques simples non-élémentaires. En effet, considérons la topologie donnée dans la figure 4.1, supposons que tous les arcs aient un coût de 1, que la source soit le sommet s et que $D = V_G \setminus \{s\}$. Les figures 4.5a et 4.5b illustrent respectivement une solution optimale $H1$ au problème MWRS en considérant uniquement les hiérarchies optiques simples non-élémentaires et une solution optimale $H2$ en considérant uniquement les hiérarchies optiques simples. La solution $H1$ est composée de deux hiérarchies optiques simples n'ayant aucun arc en commun. Dans ce cas, une seule longueur d'onde est nécessaire pour effectuer le routage. La hiérarchie $H2$ est composée de deux hiérarchies élémentaires, chacune composée de deux chemins élémentaires. L'arc (s, a) et l'arc (s, f) sont communs à deux de ces chemins. Deux longueurs d'onde sont donc nécessaires pour effectuer le routage.

Remarquons également que le nombre de longueurs d'onde d'une solution optimale pour le problème MWRS est borné inférieurement par 1 et supérieurement par $|V_G| - 2$. La borne supérieure est obtenue dans le cas où la requête multicast est de la forme $RM = (s, D = V_G - \{s, h\})$ et le réseau est une topologie en étoile telle que la source est l'une de ses feuilles (voir figure 4.6b). Le nombre de longueurs d'onde nécessaire pour le routage est alors égal au nombre des sommets destinations ($|V_G| - 2$).

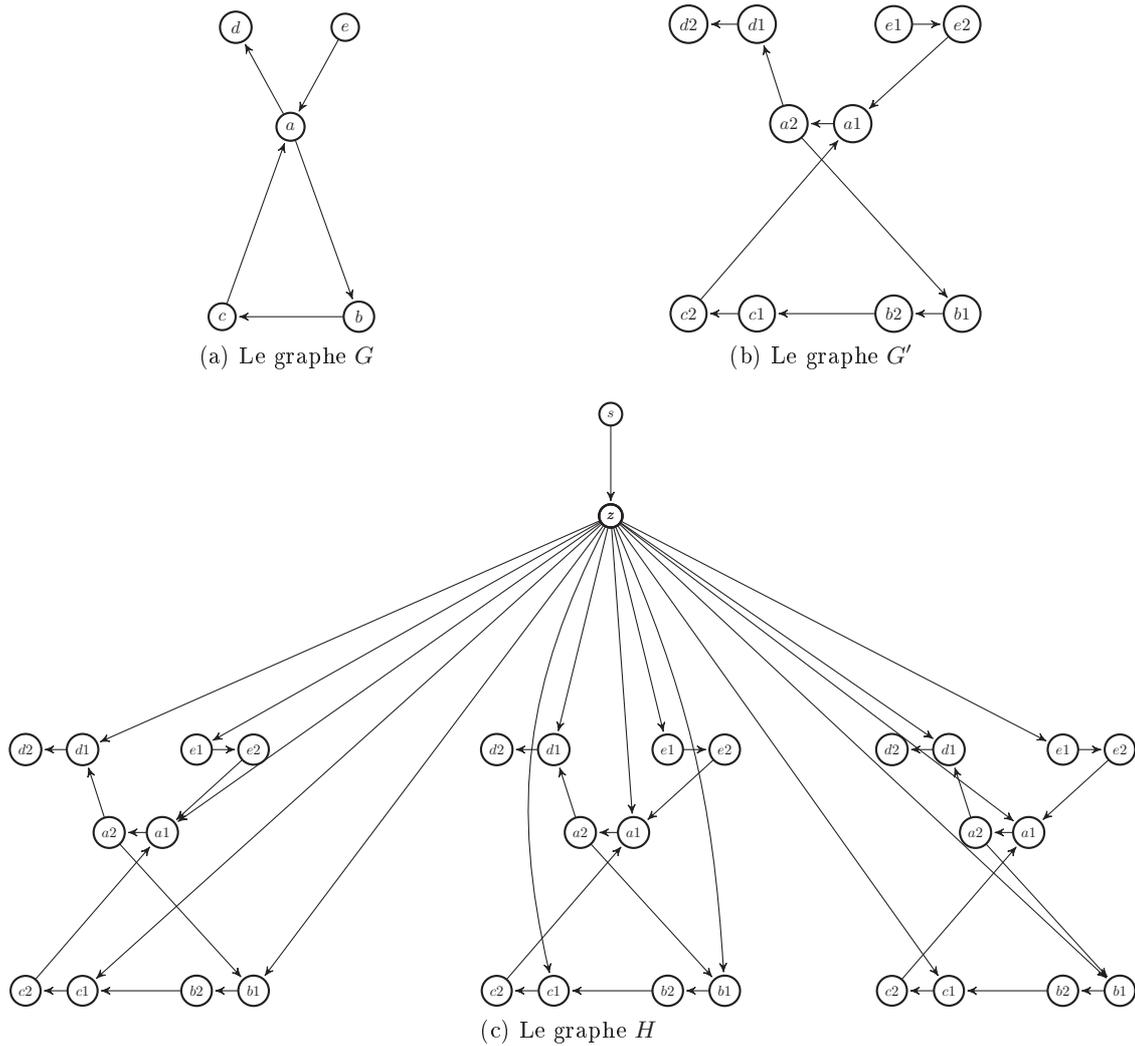


Figure 4.4 – Transformation polynomiale du graphe G pour $k = 3$

Le positionnement de la source dans le réseau peut donc influencer fortement le nombre de longueurs d'onde nécessaire pour effectuer le routage. Dans une même topologie en étoile (figure 4.6) et pour une requête multicast ($RM = (s, D = V_G \setminus \{s\})$), le positionnement de la source change radicalement le nombre de longueurs d'onde. Dans le graphe illustré par la figure 4.6a, ce nombre est égal à 1 et dans le graphe illustré par la figure 4.6b, ce nombre atteint $|V_G| - 2$.

Le problème MWRS, tel que nous l'avons défini, sélectionne parmi toutes les solutions minimisant le nombre de longueurs d'onde celle ayant un coût minimum. Notons que le coût d'une solution optimale pour le problème MWRS peut être largement supérieur à celui de la solution qui minimise uniquement le coût. Pour l'instance G illustrée par la figure 4.7 et la requête $RM = (r, D = V_G \setminus \{r\})$, la solution $H1$ optimale pour le problème MWRS nécessite une seule longueur d'onde (figure 4.7a) mais a un coût arbitrairement supérieur à celui de la solution $H2$ (qui n'est pas optimale pour le problème MWRS), si M est grand.

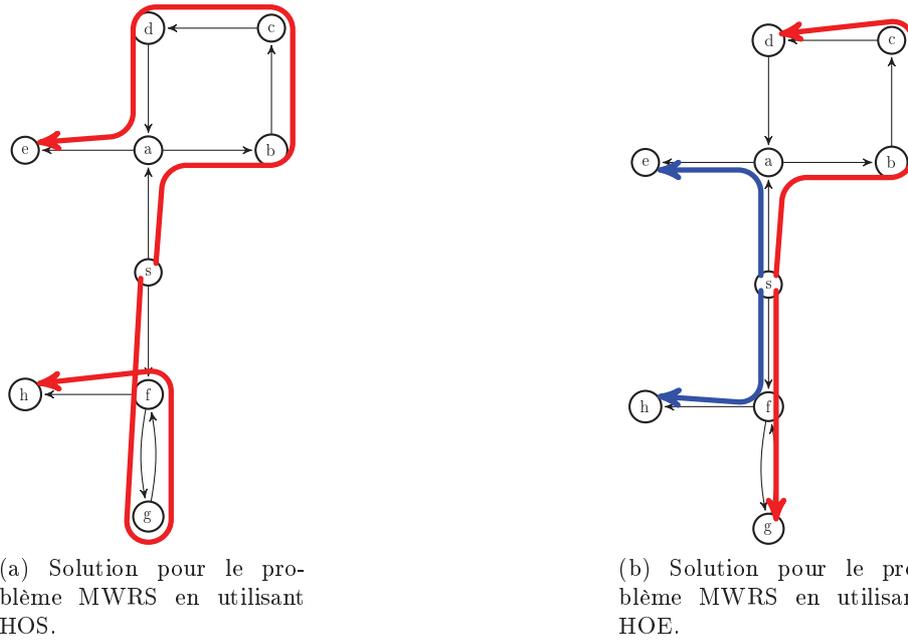


Figure 4.5 – Solution pour le problème MWRS en utilisant les structures HOS et HOE.

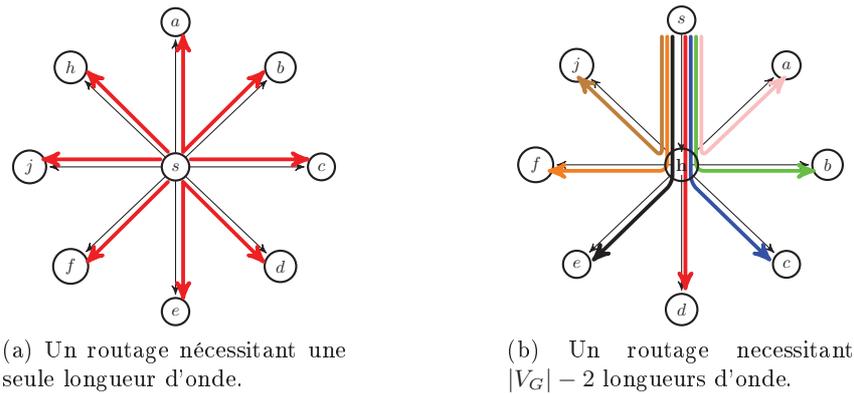
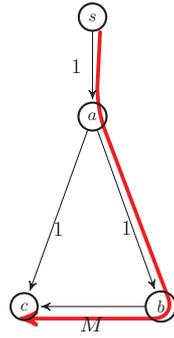


Figure 4.6 – Recouvrement d'une étoile par une hiérarchie

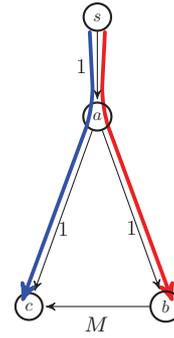
4.3 Formulation PLNE du problème MWRS

Dans ce qui suit, nous comparons la performance des solutions basées sur les chemins élémentaires (HOE) et celles basées sur les chemins simples (HOS). Pour cela, nous commençons par diviser le problème MWRS en deux sous-problèmes : le problème MWRS-HOE, dans lequel les solutions sont des hiérarchies optiques élémentaires et le problème MWRS-HOS dans lequel les solutions sont des hiérarchies optiques simples. Nous proposons une formulation en programme linéaire en nombres entiers du problème MWRS-HOS (et nous en déduisons celle du MWRS-HOE). Comme pour les problèmes étudiés dans les deux chapitres précédents, la connexité est garantie grâce à une formulation basée sur des flots. Le PLNE du problème MWRS-HOS se présente comme suit :

Les paramètres :



(a) Solution optimale $H1$ pour le problème MWRS



(b) Solution $H2$ minimisant uniquement le coût

Figure 4.7 – Minimiser le nombre de longueurs d’onde nécessaire au routage peut augmenter considérablement le coût.

- $G = (V_G, E_G)$: Le graphe orienté arête pondérée modélisant le réseau optique.
- W : L’ensemble des longueurs d’onde qui peut transiter dans chaque arc.
- Δ : Un entier positif tel que $\Delta > |V_G| \cdot \sum_{(u,v) \in E_G} c_{u,v}$.
- $In(u)$: L’ensemble des sommets prédécesseurs du sommet $u \in V_G$.
- $Out(u)$: L’ensemble des sommets successeurs au sommet $u \in V_G$.
- (s, D) : Une requête multicast spécifiant le sommet source (émetteur) et les sommets destinations.
- $d^-(u)$: Le nombre de sommets prédécesseurs au sommet u .
- $d^+(u)$: Le nombre de sommets successeurs du sommet u .
- $a(u, v)$: L’arc allant de u vers le sommet v .
- $c(u, v)$: Le coût de l’arc (u, v) .

Variables du PLNE :

- $L^\gamma(u, v)$: Variable binaire. Égale à 1 si la longueur d’onde γ est utilisée dans l’arc $a(u, v)$, 0 sinon.
- $F^\gamma(u, v)$: Variable entière de flot représentant la quantité de flot transitant sur l’arc (u, v) et utilisant la longueur d’onde γ .
- $w(\gamma)$: Variable binaire. Égale à 1 si la longueur d’onde γ est utilisée dans au moins un chemin, 0 sinon.

Formulation du PLNE

L’objectif principal est de minimiser le nombre de longueurs d’onde nécessaire afin d’atteindre toutes les destinations à partir de la source. L’objectif second est de choisir parmi toutes les solutions ayant un nombre de longueurs d’onde minimal celle ayant le coût le plus faible. Afin de respecter l’ordre des objectifs, nous attribuons un coefficient important Δ à la minimisation du nombre de longueurs d’onde par rapport à la minimisation du coût dans la fonction objective. Celle-ci est exprimée comme suit :

$$\text{Minimiser : } \Delta \cdot \sum_{\gamma \in W} w(\gamma) + \sum_{\gamma \in W} \sum_{v \in V_G} \sum_{u \in \text{Out}(v)} c_{u,v} \cdot L^\gamma(u,v) \quad (4.1)$$

Les contraintes du PLNE se présentent comme suit :

Contraintes liées à la structure HOS :

Contraintes sur la source :

$$\sum_{\gamma \in W} \sum_{u \in \text{In}(s)} L^\gamma(u,s) = 0 \quad (4.2)$$

$$1 \leq \sum_{\gamma \in W} \sum_{v \in \text{Out}(s)} L^\gamma(s,v) \leq |D| \quad (4.3)$$

La contrainte (4.2) assure que le sommet source s ne peut avoir aucun prédécesseur. En effet, un chemin passant deux fois par la source peut être divisé en deux chemins en supprimant l'arc reliant la source à son prédécesseur (figure 4.8). La contrainte (4.3) assure que la source doit avoir au moins un successeur. Le nombre de ses successeurs ne doit naturellement pas dépasser le nombre de sommets destinations $|D|$.

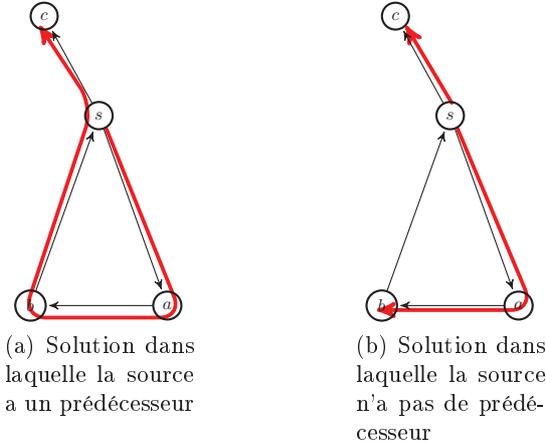


Figure 4.8 – Dans une solution optimale pour le problème MWRS-HOS, la source n'a pas de prédécesseur.

Contraintes sur les sommets destinations :

$$1 \leq \sum_{\gamma \in W} \sum_{u \in \text{In}(d)} L^\gamma(u,d) \leq |D| - 1, \forall d \in D \quad (4.4)$$

La contrainte (4.4) assure que chaque sommet destination est visité au moins une fois et au plus $|D| - 1$ fois par la hiérarchie optique simple optimale.

$$\sum_{v \in \text{Out}(u)} L^\gamma(u, v) \leq \sum_{v \in \text{In}(u)} L^\gamma(v, u), \forall \gamma \in W, \forall u \in D \quad (4.5)$$

Les sommets destinations peuvent être des feuilles mais également des sommets internes traversés une ou plusieurs fois. Le réseau étant dépourvu de switches OXC-MI, la contrainte (4.5) assure que le nombre d'arcs sortants d'un sommet ne doit pas excéder le nombre d'arcs entrants dans celui-ci.

Contraintes sur les sommets non-destinations :

$$\sum_{v \in \text{Out}(u)} L^\gamma(u, v) = \sum_{v \in \text{In}(u)} L^\gamma(v, u), \forall \gamma \in W, \forall u \in V_G \setminus (s \cup D) \quad (4.6)$$

La contrainte (4.6) assure que les sommets n'appartenant pas à l'ensemble des sommets destinations peuvent être visités plusieurs fois mais ne peuvent servir que d'intermédiaires. Ainsi, le nombre d'arcs sortants d'un tel sommet est égal au nombre de ses arcs entrants.

Relation entre $L^\gamma(m, n)$ et $w(\gamma)$:

$$w(\gamma) \geq L^\gamma(u, v), \forall u, v \in V_G, \forall \gamma \in W \quad (4.7)$$

$$w(\gamma) \leq \sum_{u \in V_G} \sum_{v \in V_G} L^\gamma(u, v), \forall \gamma \in W \quad (4.8)$$

Les contraintes (4.7) et (4.8) garantissent qu'une longueur d'onde γ est utilisée dans la hiérarchie optique optimale si et seulement si cette longueur d'onde est utilisée sur au moins un arc.

Contraintes de connexité

Contrainte de flot pour la source :

$$\sum_{\gamma \in W} \sum_{v \in \text{Out}(s)} F^\gamma(s, v) = |D| \quad (4.9)$$

La contrainte (4.9) indique que la quantité de flot émise par la source est égale au nombre de destinations de la requête multicast.

Contrainte de flot sur les sommets destinations :

$$\sum_{\gamma \in W} \sum_{v \in \text{In}(d)} F^\gamma(n, d) = \sum_{\gamma \in W} \sum_{v \in \text{Out}(d)} F^\gamma(d, v) + 1, \forall d \in D \quad (4.10)$$

$$\sum_{v \in \text{In}(d)} F^\gamma(v, d) - 1 \leq \sum_{v \in \text{Out}(d)} F^\gamma(d, v) \leq \sum_{v \in \text{In}(d)} F^\gamma(v, d), \forall \gamma \in W, \forall d \in D \quad (4.11)$$

Les contraintes (4.10) et (4.11) assurent qu'une unité de flot doit être consommée par chaque sommet destination. Cette contrainte garantit que chaque sommet destination est bien rattaché au sommet source.

Contraintes de flot sur les sommets non-destinations :

$$\sum_{v \in In(u)} F^\gamma(v, u) = \sum_{v \in Out(u)} F^\gamma(u, v), \forall \gamma \in W, \forall u \in V_G \setminus (s \cup D) \quad (4.12)$$

L'équation (4.12) assure qu'un sommet non-destination ne consomme aucune unité de flot et ne sert que de relai au flot arrivant.

Relation entre les variables $L^\gamma(u, v)$ et $F^\gamma(u, v)$:

$$F^\gamma(u, v) \geq L^\gamma(u, v), \forall u, v \in V_G, \forall \gamma \in W \quad (4.13)$$

$$F^\gamma(u, v) \leq |D| \cdot L^\gamma(u, v), \forall u, v \in V_G, \forall \gamma \in W \quad (4.14)$$

Comme mentionné dans les chapitres précédents, la relation doit être faite entre les variables d'arcs et les variables de flot. Ainsi, les contraintes (4.13) et (4.14) indiquent qu'un arc est présent dans la solution optimale si et seulement s'il supporte un flot non nul.

Remarque : La formulation PLNE du problème MWRS-HOE diffère légèrement de celle du problème MWRS-HOS. En effet, chaque sommet exceptée la source ne peut avoir au plus qu'un seul arc entrant pour chaque longueur d'onde. Il faut donc ajouter une contrainte supplémentaire au PLNE. Elle est formulée comme suit :

$$\sum_{v \in In(u)} L^\gamma(v, u) \leq 1, \forall \gamma \in W, \forall u \in V_G \setminus \{s\} \quad (4.15)$$

4.4 Résultats expérimentaux

Dans cette section, nous présentons les résultats expérimentaux pour les deux sous problèmes MWRS-HOS et MWRS-HOE. Nous comparons le nombre de longueurs d'onde (et le coût) du routage multicast en utilisant d'une part, une hiérarchie de recouvrement simple et d'autre part, une hiérarchie de recouvrement élémentaire.

4.4.1 Les paramètres des simulations

Les deux programmes linéaires en nombres entiers sont implémentés en C++ en utilisant le solveur CPLEX. Nous avons effectué une série de simulations sur des graphes orientés aléatoires générés avec la librairie LEDA v.6.3 [MN10]. Nous prenons en compte quatre tailles de graphes différentes $|V_G| = \{20, 30, 40, 50\}$. La densité (ratio entre le nombre d'arcs et le nombre de sommets) est fixée à deux pour les raisons évoquées dans les chapitres précédents.

Le coût de chaque arc est aléatoirement fixé dans l'ensemble $\{1, 2, \dots, 20\}$. La source est choisie aléatoirement. L'ensemble des destinations D est également choisi de manière aléatoire. Pour chaque taille de graphe, nous considérons 5 tailles différentes pour D : $|D| \in \{0.1 \cdot |V_G|, 0.2 \cdot |V_G|, \dots, 0.5 \cdot |V_G|\}$. Afin de s'assurer de l'existence d'une solution réalisable, les graphes sont sélectionnés aléatoirement parmi l'ensemble des graphes connexes contenant au moins un che-

min orienté de la source vers chaque destination. Nous effectuons 100 simulations pour chaque taille de graphe et pour chaque taille de requête multicast. Pour chaque simulation, le nombre de longueurs d'onde et le coût du routage sont les critères d'évaluation de la performance des structures HOE et HOS.

Enfin, pour réduire le temps de résolution, nous utilisons l'heuristique Farthest First proposée dans [LMP13] ainsi que l'heuristique Farthest Greedy proposée dans [Din09]. Ces heuristiques permettent, respectivement, d'obtenir une borne supérieure sur le nombre de longueurs d'onde que peut utiliser une hiérarchie de recouvrement optimale pour le problème MWRS-HOE et pour le problème MWRS-HOS. Ainsi, toutes les solutions ayant un nombre de longueurs d'onde supérieur à la borne seront directement exclues dans le calcul des solutions optimales des deux problèmes.

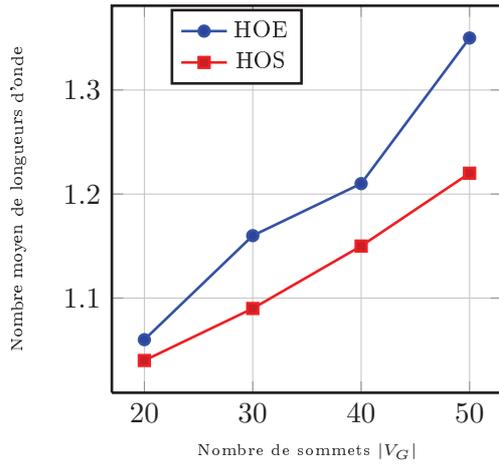
4.4.2 Les résultats des simulations

Les résultats des simulations sont présentés dans la table 4.1 et dans la figure 4.9. Comme prévu, les hiérarchies optiques simples donnent de meilleurs résultats que les hiérarchies optiques élémentaires, à la fois en terme de longueurs d'onde mais aussi en terme de coût. Concernant le nombre de longueurs d'onde utilisé, la solution optimale pour le problème MWRS-HOS utilise toujours moins de longueurs d'onde que la solution optimale pour le problème MWRS-HRE. Le ratio d'amélioration du nombre de longueurs d'onde utilisé par la solution HOS optimale pour le problème MWRS-HOS par rapport à la solution HOE optimale pour le problème MWRS-HOE est en moyenne égal à 12.71%, pour une taille de graphes égale à 50. Ce ratio atteint 24,35% quand $|V| = 20$ et $|D| = 10$. Cette amélioration est prévisible car la possibilité de visiter plusieurs fois un sommet par la structure HOS permet d'augmenter le nombre de sommets visités par chaque chemin la composant par rapport à la structure HOE. Ceci permet naturellement de réduire le nombre de longueurs d'onde nécessaire pour le routage.

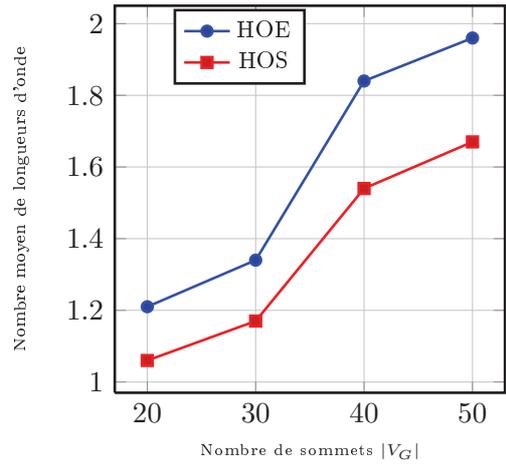
Concernant le coût, il existe, certes, quelques instances pour lesquelles le coût de la solution optimale pour le problème MWRS-HOE est meilleur que le coût de la solution optimale pour le problème MWRS-HOS, en utilisant cependant plus de longueurs d'onde. Toutefois, la moyenne d'amélioration du coût calculée, pour chaque taille de graphe et sur les 100 instances, est toujours positive. Cette amélioration reste faible car l'objectif principal est de minimiser le nombre de longueurs d'onde utilisé.

<i>Taille</i>	Nombre moyen de longueurs d'onde			Coût total		
$ V_G =20$						
$ D $	HOE	HOS	\searrow	HOE	HOS	\searrow
2	1.06	1.04	12,39%	8740	8586	1,8%
4	1.21	1.06	18,12%	9762	9543	2,24%
6	1.55	1.21	21,93%	11499	11236	2,3%
8	1.75	1.38	21,14%	14165	13748	2,9%
10	1.93	1.46	24,35%	16222	16160	0,4%
AVG			16,34%			1,86%
$ V_G =30$						
$ D $	HOE	HOS	\searrow	HOE	HOS	\searrow
3	1.16	1.09	12,68%	9017	13516	2,7%
6	1.34	1.17	4,39%	13735	11253	1,6%
9	1.89	1.55	17,98%	19435	18793	3,3%
12	2.07	1.57	6,21%	21379	21316	0,3%
15	2.38	1.86	21,84%	26345	26273	0,3%
AVG			16,54%			1,63%
$ V_G =40$						
$ D $	HOE	HOS	\searrow	HOE	HOS	\searrow
4	1.21	1.15	4,95%	13105	12932	1,3%
8	1.84	1.54	16,3%	20060	19709	1,7%
12	2.18	1.72	21,10%	27456	27135	1,2%
16	2.39	1.99	16,73%	32210	32322	1,24%
20	2.58	2.25	12,79%	36511	35790	2,0%
AVG			14,378%			1,17%
$ V_G =50$						
$ D $	HOE	HOS	\searrow	HOE	HOS	\searrow
5	1.35	1.22	9,629%	15894	15249	4,1%
10	1.96	1.67	14,79%	26920	26803	0,4%
15	2.17	1.94	10,59%	34077	33329	2,2%
20	2.85	2.51	11,92%	42700	41649	2,5%
25	2.85	2.60	8,77%	47939	47407	1,1%
AVG			11,145%			2,051%

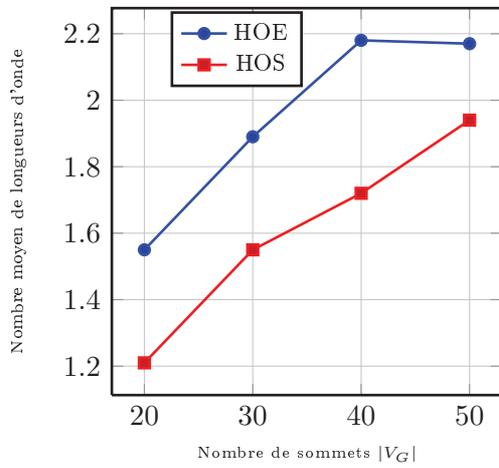
TABLE 4.1 – Comparaison de performances entre la structure HOE et la structure HOS.



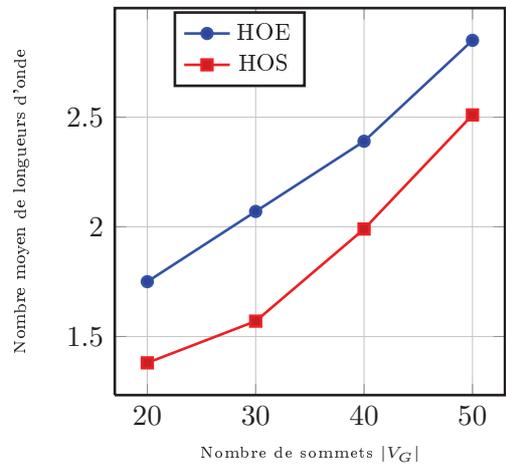
(a) $|D| = 0.1 \cdot |V_G|$



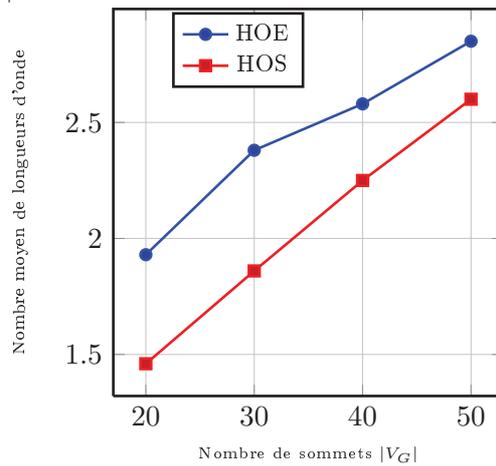
(b) $|D| = 0.2 \cdot |V_G|$



(c) $|D| = 0.3 \cdot |V_G|$



(d) $|D| = 0.4 \cdot |V_G|$



(e) $|D| = 0.5 \cdot |V_G|$

Figure 4.9 – Solutions optimales du problème MWRS-HOS versus solutions optimales du problème MWRS-HOE

4.5 Conclusion

En étudiant un problème qui prend en compte plusieurs exigences concrètes, nous avons montré dans ce chapitre que la hiérarchie de recouvrement est une structure applicable et adaptée à un problème tiré directement de la réalité des réseaux optiques. Nous avons ainsi introduit le problème MWRS (Wavelength Routing Structure), traitant du routage multicast dans les réseaux optiques dépourvus de switch OXC-MC et consistant à calculer une structure de recouvrement tel que le nombre de longueurs d'onde utilisé est minimum. Si plusieurs solutions admettent le même nombre de longueurs d'onde W^* , alors celle ayant un coût minimum est sélectionnée. La structure doit cependant respecter non seulement la contrainte sur le degré mais également celle sur la distinction des longueurs d'onde. Nous avons, par la suite, prouvé qu'il n'existe pas d'algorithme polynomial décidant si W^* est inférieur ou égal à un entier strictement positif k . Nous avons donné une formulation PLNE du problème MWRS. La solution à ce problème peut être un ensemble de hiérarchies optiques élémentaires ou bien un ensemble de hiérarchies optiques simples. Afin de distinguer les deux, nous avons introduit deux sous-problèmes du MWRS. Le premier (MWRS-HOE) accepte uniquement des hiérarchies optiques élémentaires comme solution et le second (MWRS-HOS) n'accepte que des hiérarchies optiques simples comme solution.

Nous savons théoriquement que la solution optimale du problème MWRS-HOS utilise toujours moins de longueurs d'onde que celle du problème MWRS-HOE. Les simulations sur des graphes aléatoires nous ont permis de quantifier l'amélioration apportée. Nous avons ainsi observé que pour des graphes de taille égale à 50, l'amélioration du nombre de longueurs d'onde est en moyenne égale à 12.71%. Sur certaines instances, cette amélioration peut dépasser 25% [LM].

Conclusion

Depuis l'apparition de l'Internet, l'évolution effrénée et permanente des réseaux de communication entraîne l'émergence de nouveaux problèmes d'optimisation. La minimisation du coût de la transmission de l'information tout en prenant en compte des contraintes propres à chaque type de réseau est un challenge de plus en plus omniprésent. Cette transmission d'information est réalisée à travers des routes reliant la source aux différentes destinations. L'ensemble de ces routes forme la structure de recouvrement dont dépendent le coût et la qualité de service.

Dans cette thèse, nous avons proposé l'étude des problèmes de recouvrement en utilisant une structure mieux adaptée aux exigences des réseaux à très haut débit. La structure d'arbre de recouvrement a toujours été la plus utilisée pour ce type de problèmes. Celle-ci tire son attractivité du fait qu'elle permette de couvrir les sommets d'un graphe connexe en utilisant le minimum de liens. Cette spécificité permet de faciliter considérablement le routage d'informations dans les différents types de réseau. Si l'utilité de minimiser le coût des liens utilisés par l'arbre est avérée, on parlera alors d'arbre de recouvrement de coût minimum.

Nous avons clairement justifié, en évoquant les spécificités des réseaux de communication en général et des réseaux optiques en particulier, la nécessité de limiter le degré des sommets de la structure de recouvrement. Selon la topologie du réseau, il n'existe pas toujours un arbre de recouvrement qui respecte cette limitation. Il suffit pour s'en convaincre, de considérer un réseau en étoile ayant p terminaux et une limitation fixée à $p - 2$. De plus, le coût de l'arbre de recouvrement respectant cette limitation peut être largement supérieur à celui de l'arbre de recouvrement de coût minimum (il suffit de relier les feuilles de ce réseau en étoile par des liens ayant un coût élevé et d'attribuer aux autres liens un coût unitaire).

L'arbre de recouvrement d'un graphe étant un sous-graphe de celui-ci, l'acheminement de l'information est réalisé exclusivement dans les routes appartenant à ce sous-graphe. Certaines applications ne l'exigent pas. En effet, dans les réseaux optiques à passage multiple, le routage de l'information peut emprunter des chemins qui passent une ou plusieurs fois par un sommet. La limite sur le degré d'un sommet est alors réinitialisée à chaque passage. La structure de recouvrement d'un graphe n'est donc pas nécessairement un sous-graphe.

Une nouvelle structure appelée « hiérarchie » est proposée dans [Mol08]. De par sa flexibilité, elle est parfaitement adaptée à ce type de routage et permet d'élargir l'espace de recherche et des solutions. Notre contribution est basée sur cette structure. Notre objectif principal étant de montrer son intérêt vis-à-vis de l'arbre en terme de coût et de faisabilité du routage, nous avons reconsidéré les problèmes classiques de recherche de structures (arbre) de recouvrement sous contrainte sur le degré des sommets en s'appuyant sur les hiérarchies de recouvrement.

Après avoir étudié les travaux antérieurs sur les problèmes de recouvrement sous contrainte sur le degré des sommets, nous nous sommes spécifiquement intéressés au problème du recouvrement total avec une borne supérieure sur le degré des sommets. Constamment inscrits dans une démarche comparative entre la hiérarchie et l'arbre, nous avons reconsidéré le problème classique NP-difficile DCMST (Degree Constrained Minimum Spanning Tree) et avons introduit un nouveau problème nommé DCMSH (Degree Constrained Minimum Spanning Hierarchy) qui consiste à chercher une hiérarchie de recouvrement d'un graphe de coût minimum tel que le degré de ses sommets soit inférieur ou égal à un entier B . Nous avons prouvé que ce nouveau problème est également NP-difficile. Nous avons donné un algorithme de résolution exacte du problème DCMSH articulé autour de trois étapes. D'abord, nous construisons un k -graphe à partir du graphe de départ. Ensuite, un Programme Linéaire en Nombres Entiers ayant des variables correspondant aux sommets et arêtes du k -graphe se basant sur une formulation de flot est résolu. Enfin, en se basant sur notre étude des propriétés des solutions optimales du problème, une hiérarchie optimale est construite à partir des valeurs des variables retournées par le PLNE. Grâce à cet algorithme, nous réussissons à résoudre de manière exacte le problème DCMSH sur des graphes de taille largement supérieure à celle des réseaux optiques existants. Une expérimentation sur des graphes aléatoires permet de montrer concrètement la diminution du coût des solutions en utilisant des hiérarchies plutôt que des arbres.

Alors que le problème DCMST n'accepte pas d'algorithme de résolution polynomial avec garantie de performance, nous avons démontré que le problème DCMSH est dans APX et nous avons proposé un algorithme $\frac{B}{B-1}$ approché. Cette heuristique a comme point de départ l'arbre de coût minimum (MST). Le ratio d'approximation est calculé en considérant le coût d'un MST comme borne inférieure. Nous avons démontré que ce ratio est le meilleur que l'on puisse espérer en construisant une heuristique uniquement basée sur le MST.

Toujours positionné dans cette démarche comparative, nous nous sommes penchés sur deux autres problèmes NP-difficiles visant à limiter le nombre de sommets de branchement dans la structure de recouvrement. Ils trouvent leur intérêt pratique dans la réduction du nombre de switches et de divisions du signal lumineux pour les routages multicast/broadcast optique. Nous avons d'abord étudié la relation entre le problème MBVST (Minimum Vertices Spanning Tree) et le problème MDST (Minimum Degree Sum Spanning Tree). Nous avons prouvé qu'ils n'étaient pas « liés » en montrant qu'il existe au moins une instance pour laquelle les deux ensembles de

solutions optimales de ces problèmes sont disjoints. Il est évident que trouver une hiérarchie de recouvrement d'un graphe connexe ayant un minimum de sommets de branchement est trivial, puisqu'il suffit de le parcourir pour obtenir une hiérarchie sans sommets de branchement. Nous nous sommes alors orientés vers la minimisation du coût tout en bornant le nombre de sommets de branchement. Nous avons ainsi introduit le problème BVCMST (Branch Vertices Constrained Minimum Spanning Tree), pour les arbres et le problème BVCMSH (Branch Vertices Constrained Minimum Spanning Hierarchy), pour les hiérarchies de recouvrement. Ces deux problèmes restent NP-difficiles mais des algorithmes de résolution exacte sont donnés. Des jeux de tests permettent de montrer, tout aussi concrètement que pour le problème DCMSH, la nette amélioration du coût en utilisant la structure de hiérarchie de recouvrement. Enfin, une heuristique avec garantie de performance est donnée pour le problème MBVMSH.

Afin d'illustrer l'applicabilité des hiérarchies de recouvrement, nous avons considéré un problème prenant davantage en compte la réalité des réseaux optiques. Le problème MWRS (Wavelength Routing Structure), traitant du routage multicast dans les réseaux optiques dépourvus de switch OXC-MC, consiste en un calcul d'une structure de recouvrement tel que le nombre de longueurs d'onde utilisées est minimum. Si plusieurs solutions admettent le même nombre de longueurs d'onde W^* , alors celle ayant un coût minimum est sélectionnée. La structure doit cependant respecter non seulement la contrainte sur le degré mais également celle sur la distinction des longueurs d'onde. Nous avons prouvé qu'il n'existe pas d'algorithme polynomial décidant si W^* est inférieur ou égal à un entier strictement positif k . Nous avons donné une formulation PLNE du problème MWRS et nous avons montré grâce à des simulations sur des graphes aléatoires que la solution optimale à ce problème est presque toujours une hiérarchie de recouvrement qui n'est pas un arbre.

Les schémas de résolution exacte et approchée que nous avons développés pour le problème DCMSH ont été réutilisés pour la résolution du problème BVCMSH. Ainsi, des modifications mineures de ceux-ci permettraient de résoudre d'autres problèmes de recouvrement. Nous pensons au problème de recherche de hiérarchie de recouvrement de coût minimum tel que le nombre de feuilles est inférieur ou égal à un entier fixé et à celui de la recherche de hiérarchie de recouvrement de coût minimum tel que la somme des degrés des sommets de branchement est bornée. Nous avons également la certitude que ces algorithmes peuvent être adaptés au recouvrement partiel (Steiner) sous contrainte sur le degré. Nous projetons, par ailleurs, d'explorer d'autres méthodes de résolution exacte telles que la programmation dynamique, le Branch and bound, la génération de colonnes ou encore les méta-heuristiques.

Les heuristiques que nous avons élaborées ont un arbre de coût minimum comme point de départ. Le coût de celui-ci pouvant être largement inférieur à celui de la hiérarchie optimale du DCMSH et du BVCMSH, nous envisageons de développer des heuristiques ayant comme point de départ une structure dont le coût s'approcherait davantage de celui de la hiérarchie optimale. Cette démarche permettrait d'améliorer le ratio d'approximation que nous avons établi. Nous pouvons

également orienter nos efforts vers l'élaboration d'algorithmes d'approximation bi-critères. En effet, Singh et al. proposent dans [SL07] un algorithme d'approximation $(1, B + 1)$ -approchée pour le problème bicritère consistant à minimiser le coût et le degré maximum d'un arbre couvrant. Trouver un algorithme d'approximation $(1, B + \rho)$ approchée (tel que ρ soit le plus petit possible) pour le problème DCMSH reste un challenge à relever.

Bibliographie

- [AD00] M. Ali and J. S. Deogun. Cost-effective implementation of multicasting in wavelength-routed networks. *IEEE/OSA Journal of Lightwave Technology*, 18 :1628–1638, 2000.
- [AD06] M. Ali and J. S. Deogun. Power-efficient design of multicast wavelength-routed networks. *IEEE J.Sel. A. Commun.*, 18(10) :1852–1862, September 2006.
- [Ber85] C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd, 1985.
- [BGPea97] B. Beauquier, L. Gargano, S. Perennes, and et al. Graph problems arising from wavelength-routing in all-optical networks. Proc. 2nd Workshop on Optics and Computer Science, Geneva, 1997.
- [BJS09] S. M. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. Wiley, 2009.
- [BMMN95] M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser. Optimal trees. In *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 503 – 615. Elsevier, 1995.
- [BST99] C. Bazgan, M. Santha, and Z. Tuza. On the approximation of finding a(nother) hamiltonian cycle in cubic hamiltonian graphs. *Journal of Algorithms*, 31(1) :249–268, 1999.
- [CCGG13] F. Carrabs, R. Cerulli, M. Gaudioso, and M. Gentili. Lower and upper bounds for the spanning tree with minimum branch vertices. *Computational Optimization and Applications*, 56(2) :405–438, 2013.
- [CCR14] C. Cerrone, R. Cerulli, and A. Raiconi. Relations, models and a memetic approach for three degree-dependent spanning tree problems. *European Journal of Operational Research*, 232(3) :442 – 453, 2014.
- [CGI09] R. Cerulli, M. Gentili, and A. Iossa. Bounded-degree spanning tree problems : models and new algorithms. *Comput. Optim. Appl*, 42 :353–370, April 2009.
- [Cie00] D. Cieslik. The vertex degrees of minimum spanning trees. *European Journal of Operational Research*, 125(2) :278 – 282, 2000.

- [CLRS02] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction à l'algorithmique*. Dunod, 2002.
- [Cur07] W. Curt. *Data communication and computer networks*. MA : Thomson Course Technology, 2007.
- [Dee92] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford, CA, USA, 1992. UMI Order No. GAX92-21608.
- [DH68] N. Deo and L. Hakimi. The shortest generalized hamiltonian tree. In *6th Annual Allerton Conference*, pages 879–888, Illinois, USA, 1968.
- [Din09] D. Din. Heuristic Algorithms for Finding Light-Forest of Multicast Routing on WDM Network. *Information Science and Engineering*, 25 :83–103, 2009.
- [Dir52] G. A. Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, s3-2(1) :69–81, 1952.
- [DK97] N. Deo and N. Kumar. Computation of constrained spanning trees : A unified approach. In P. Pardalos, D. Hearn, and W. Hager, editors, *Network Optimization*, volume 450 of *Lecture Notes in Economics and Mathematical Systems*, pages 196–220. Springer Berlin Heidelberg, 1997.
- [FKK⁺08] E. Flandrin, T. Kaiser, R. Kužel, H. Li, and Z. Ryjáček. Neighborhood unions and extremal spanning trees. *Discrete Mathematics*, 308(12) :2343 – 2350, 2008.
- [FR94] M. Fürer and B. Raghavachari. Approximating the minimum-degree steiner tree to within one of optimal. *J. Algorithms*, 17(3) :409–423, November 1994.
- [FY62] Y. Fu and S. S. Yau. A note on the reliability of communication networks. *Journal of the Society for Industrial and Applied Mathematics*, 10(3) :469–474, 1962.
- [GH03] L. Gargano and M. Hammar. There are spanning spiders in dense graphs (and we know how to find them). In J. Baeten, J. Lenstra, J. Parrow, and G. Woeginger, editors, *Automata, Languages and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 802–816. Springer Berlin Heidelberg, 2003.
- [GHH⁺04] L. Gargano, M. Hammar, P. Hell, L. Stacho, and U. Vaccaro. Spanning spiders and light-splitting switches. *Discrete Mathematics*, 285(1–3) :83 – 95, 2004.
- [GHSV02] L. Gargano, P. Hell, L. Stacho, and U. Vaccaro. Spanning trees with bounded number of branch vertices. In P. Widmayer, S. Eidenbenz, F. Triguero, R. Morales, R. Conejo, and M. Hennessy, editors, *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 355–365. Springer Berlin Heidelberg, 2002.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman Co, New York, NY, USA, 1979.
- [Goe06] M. X. Goemans. Minimum bounded degree spanning trees. In *Foundations of Computer Science, 2006. FOCS '06. 47th Annual IEEE Symposium on*, pages 273–282, 2006.

- [Gre92] P. E. Green. *Fiber Optic Networks*. Prentice-Hall, Cambridge, 1992.
- [GT84] H. N. Gabow and R. E. Tarjan. Efficient algorithms for a family of matroid intersection problems. *J. Algorithms*, 5(1) :80–131, 1984.
- [HK71] M. Held and R. Karp. The traveling-salesman problem and minimum spanning trees : Part ii. *Mathematical Programming*, 1(1) :6–25, 1971.
- [Hoo91] J. A. Hoogeveen. Analysis of christofides’ heuristic : Some paths are more difficult than cycles. *Oper. Res. Lett.*, 10(5) :291–295, July 1991.
- [Jai01] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21 :39–60, 2001.
- [Kar72] R. M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
- [KNS74] D. Klingman, A. Napier, and J. Stutz. A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20(5) :814–821, 1974.
- [KR00] J. Könemann and R. Ravi. A matter of degree : Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM Journal on Computing*, 31 :537–546, 2000.
- [LHL08] C. Liang, Y. Huang, and J. Lin. An energy efficient routing scheme in wireless sensor networks. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 916–921, March 2008.
- [LM] D.D. Le and M. Merabet. Light-trail based hierarchies : The optimal solution for multicast routing problem in wdm networks without splitters. In *International conference on Photonics, Optics and Laser technology - Photonics*, pages 66–73.
- [LMP13] D. D. Le, M. Molnar, and J. Palaysi. Multicast Routing in WDM without Splitters. *Infocommunications Journal*, 5(2) :1–10, June 2013.
- [MDM12] M. Merabet, S. Durand, and M. Molnár. Exact solution for bounded degree connected spanning problems. Technical report, 2012.
- [MDM13a] M. Merabet, S. Durand, and M. Molnár. Exact solution for branch vertices constrained spanning problems. *Electronic Notes in Discrete Mathematics*, 41(0) :527 – 534, 2013.
- [MDM13b] M. Merabet, S. Durand, and M. Molnár. Minimization of branching in the optical trees with constraints on the degree of nodes. In *The Eleventh International Conference on Networks - ICN*, pages 235–240, 2013.
- [MDM14] M. Molnár, S. Durand, and M. Merabet. Approximation of the degree-constrained minimum spanning hierarchies. In Magnús M. Halldórsson, editor, *Structural Information and Communication Complexity*, volume 8576 of *Lecture Notes in Computer Science*, pages 96–107. Springer International Publishing, 2014.

- [MN10] K. Mehlhorn and S. Naeher. LEDA-a Library of Efficient Data Types and Algorithms, November 2010.
- [Mol08] M. Molnár. Hierarchies for Constrained Partial Spanning Problems in Graphs. Technical report PI-1900, 2008.
- [MOY14] H. Matsuda, K. Ozeki, and T. Yamashita. Spanning trees with a bounded number of branch vertices in a claw-free graph. *Graphs and Combinatorics*, 30(2) :429–437, 2014.
- [Muk00] B. Mukherjee. Wdm optical communication networks : progress and challenges. *Selected Areas in Communications, IEEE Journal on*, 18(10) :1810–1824, Oct 2000.
- [NC80] C. Subhash Narula and A. Ho Cesar. Degree-constrained minimum spanning tree. *Computers and Operations Research*, 7(4) :239 – 249, 1980.
- [Obr68] A. K. Obruca. Spanning tree manipulation and the travelling salesman problem. *The Computer Journal*, 10(4) :374–377, 1968.
- [Pri57] R. C. Prim. Shortest connection networks and some generalizations. *The Bell Systems Technical Journal*, 36(6) :1389–1401, 1957.
- [Pro01] J. Proakis. *Wiley encycloedia of telecommunication*. John Wiley and Sons, Inc., 2001.
- [Puj08] G. Pujolle. *Les Réseaux*. 5ème édition, Eyrolles, 2008.
- [QJG⁺99] C. Qiao, M. Jeong, A. Guha, X. Zhang, and J. Wei. Wdm multicasting in ip over wdm networks. In *Network Protocols, 1999. (ICNP '99) Proceedings. Seventh International Conference on*, pages 89–96, October 1999.
- [RM98] B. Ramamurthy and B. Mukherjee. Wavelength conversion in wdm networking. *IEEE journal on selected areas in communications*, 16 :1061–1073, 1998.
- [RMR⁺93a] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt. Many birds with one stone : multi-objective approximation algorithms. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 438–447, New York, NY, USA, 1993. ACM.
- [RMR⁺93b] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt. Many birds with one stone : multi-objective approximation algorithms. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 438–447, New York, NY, USA, 1993. ACM.
- [RMR⁺01] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III. Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica*, 31(1) :58–78, 2001.
- [RS06] R. Ravi and M. Singh. Delegate and conquer : An lp-based approximation algorithm for minimum degree msts. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 169–180. Springer Berlin Heidelberg, 2006.

- [SL07] M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 661–670, New York, NY, USA, 2007. ACM.
- [SS12] A.A.M. Saleh and Jane M. Simmons. All-optical networking x2014 ;evolution, benefits, challenges, and future vision. *Proceedings of the IEEE*, 100(5) :1105–1117, May 2012.
- [SSM⁺11] D. Silva, R. Silva, G. Mateus, J. Gonçalves, M. Resende, and P. Festa. An iterative refinement algorithm for the minimum branch vertices problem. In P. Pardalos and S. Rebennack, editors, *Experimental Algorithms*, volume 6630 of *Lecture Notes in Computer Science*, pages 421–433. Springer Berlin Heidelberg, 2011.
- [SvSR14] M. Savi, H. Øverby, N. Stol, and C. Raffaelli. Cost evolution model to design optical switching fabrics with wavelength converters. *Computer Networks*, 72(0) :99 – 112, 2014.
- [Win89] S. Win. On a connection between the existence of k -trees and the toughness of a graph. *Graphs and Combinatorics*, 5(1) :201–205, 1989.
- [ZOK93] W. De Zhong, Y. Onozato, and J. Kaniyil. A copy network with shared buffers for large-scale multicast. *ATM Switching," IEEE/ACM Transactions on Networking*, pages 157–165, 1993.
- [ZP05] Y. Zhou and G. S. Poo. Optical multicast over wavelength-routed WDM networks : A survey. *Optical Switching and Networking*, 2(3) :176 – 197, 2005.

Résumé

Le travail que nous développons dans le cadre de cette thèse s'articule autour des problèmes de recherche de structure de recouvrement de graphes sous contrainte sur le degré des sommets. Comme l'arbre de recouvrement couvre les sommets d'un graphe connexe avec un minimum de liens, il est généralement proposé comme solution à ce type de problèmes. Cependant, pour certaines applications telles que le routage dans les réseaux optiques, les solutions ne sont pas nécessairement des sous-graphes. Nous supposons dans cette thèse que la contrainte sur le degré est due à une capacité limitée instantanée des sommets et que la seule exigence sur le recouvrement est sa connexité. Dans ce cas, la solution peut être différente d'un arbre. Nous reformulons ces problèmes de recouvrement en nous appuyant sur une extension du concept d'arbre appelée hiérarchie de recouvrement. Notre objectif principal est de démontrer son intérêt vis-à-vis de l'arbre en termes de faisabilité et de coût du recouvrement. Nous considérons deux types de contraintes sur le degré : des bornes sur le degré des sommets ou une borne sur le nombre de sommets de branchement et cherchons dans les deux cas un recouvrement de coût minimum. Nous illustrons aussi l'applicabilité des hiérarchies en étudiant un problème prenant davantage en compte la réalité du routage optique. Pour ces différents problèmes NP-difficiles, nous montrons, tant sur le coût des solutions optimales que sur la garantie de performance des solutions approchées, l'intérêt des hiérarchies de recouvrement. Ce constat se voit conforté par des expérimentations sur des graphes aléatoires.

Abstract

The work conducted in this thesis is focused on the minimum spanning problems in graphs under constraints on the vertex degrees. As the spanning tree covers the vertices of a connected graph with a minimum number of links, it is generally proposed as a solution for this kind of problems. However, for some applications such as the routing in optical networks, the solution is not necessarily a sub-graph. In this thesis, we assume that the degree constraints are due to a limited instantaneous capacity of the vertices and that the only pertinent requirement on the spanning structure is its connectivity. In that case, the solution may be different from a tree. We propose the reformulation of this kind of spanning problems. To find the optimal coverage of the vertices, an extension of the tree concept called hierarchy is proposed. Our main purpose is to show its interest regarding the tree in term of feasibility and costs of the coverage. Thus, we take into account two types of degree constraints : either an upper bound on the degree of vertices and an upper bound on the number of branching vertices. We search a minimum cost spanning hierarchy in both cases. Besides, we also illustrate the applicability of hierarchies by studying a problem that takes more into account the reality of the optical routing. For all those NP-hard problems, we show the interest of the spanning hierarchy for both costs of optimal solutions and performance guarantee of approximate solutions. These results are confirmed by several experimentations on random graphs.