



HAL
open science

Algorithmique et optimisation dans les réseaux de télécommunications

David Coudert

► **To cite this version:**

David Coudert. Algorithmique et optimisation dans les réseaux de télécommunications. Réseaux et télécommunications [cs.NI]. Université Nice Sophia Antipolis, 2010. tel-00466400

HAL Id: tel-00466400

<https://theses.hal.science/tel-00466400>

Submitted on 23 Mar 2010

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

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

UNIVERSITÉ de NICE-SOPHIA ANTIPOLIS – UFR SCIENCES

École Doctorale STIC
Département d'Informatique

Habilitation à Diriger des Recherches

de l'Université de Nice-Sophia Antipolis

Spécialité : **INFORMATIQUE**

par

David COUDERT

Algorithmique et optimisation dans les réseaux de télécommunications

Soutenue publiquement le 5 mars 2010 devant le jury composé de :

Rapporteurs	M. Pierre	FRAIGNIAUD	DR CNRS
	M. Christos	KAKLAMANIS	Professeur
	M. Arie M.C.A.	KOSTER	Professeur
Président	M. David	SIMPLLOT-RYL	Professeur
Examineurs	M. Jean-Claude	BERMOND	DR CNRS
	M. Pierre	FRAIGNIAUD	DR CNRS
	M. Arie M.C.A.	KOSTER	Professeur
	M. Philippe	NAIN	DR INRIA
	M. Dimitri	PAPADIMITRIOU	Alcatel-Lucent

INRIA Sophia Antipolis - Méditerranée, Salle Euler violet à 10h30

Table des matières

1	Introduction	1
2	Introduction (in english)	9
3	Groupage de trafic	15
3.1	Echange total sur l'anneau unidirectionnel	16
3.1.1	Exemple	16
3.1.2	Modèle	17
3.1.3	Comment établir des bornes inférieures	18
3.1.4	Résultats connus	21
3.2	Chemin orienté	22
3.2.1	Ensemble complet de requêtes	23
3.2.2	Ratio maximum	25
3.3	Trafic général	26
3.3.1	Complexité et (in)approximabilité	26
3.3.2	Le cas $g = 1$	28
3.3.3	Anneau bidirectionnel	28
3.4	Discussion	29
4	Groupe de ressources partageant un risque	31
4.1	Modélisation	32
4.2	Problèmes d'optimisation combinatoire colorés	33
4.2.1	Chemin coloré	33
4.2.2	Coupe colorée	35
4.2.3	Chemins colorés disjoints	36
4.2.4	Arbre couvrant	36
4.3	Conclusion	36
5	Reconfiguration de routages	39
5.1	Introduction	39
5.2	Modélisation	41
5.2.1	Graphe de dépendances	42
5.2.2	Stratégie de traitement	43
5.3	Métriques sans pénalité	44
5.3.1	Indice de traitement <i>vs</i> indice de transmission	46
5.3.2	Complexité et inapproximabilité	46
5.4	Modèles avec pénalités et interdictions	47
5.4.1	Interdictions – Requêtes qui ne peuvent pas être interrompues	47
5.4.2	Pénalité par interruption	49

5.4.3	Pénalité par étape	50
5.4.4	Nombre d'étapes	52
5.5	Relations avec les jeux de capture	52
5.5.1	Indice de traitement <i>vs</i> indice d'échappement	52
5.5.2	Graphes d'indice de traitement inférieur à 2	54
5.5.3	Graphes orientés d'indice de traitement inférieur à 2	56
5.5.4	Arbres	56
5.5.5	Graphes planaires extérieurs	57
5.6	Conclusion	58
6	Conclusion et perspectives	61
	Bibliographie	65
	CV complet	77

Chapitre 1

Introduction

Ce document retrace mes activités scientifiques après thèse, certains travaux étant la suite de travaux engagés au cours de ma thèse.

Pour rappel, ma thèse portait sur l’algorithmique et l’optimisation dans les réseaux de communications optiques. La première partie portait sur la conception et la caractérisation de réseaux d’interconnexions optique en espace libre utilisant l’architecture OTIS (*Optical Transpose Interconnection System*). Je renvoie à [T1, A2, A3, A4, Ci15, Ci16, Ci17, Ci19, Cn39, Cn41] pour l’ensemble de mes travaux sur les « *OTISeries* ». La deuxième partie portait sur la sécurisation par protection dans les réseaux optiques à multiplexage en longueur d’onde (*wavelength division multiplexing*, WDM) [T1, Cis36, Ci18, Cn40, Cn41], et je continue depuis à m’intéresser à ces questions.

Le contexte général de mes travaux se situe dans les réseaux orientés connexions, que ce soit des réseaux optiques à multiplexage en longueur d’onde (WDM), des réseaux MPLS (multi-protocol label switching), ou encore des réseaux à faisceaux hertziens (*backhaul networks*). Je m’intéresse principalement à des flux d’informations en transit dans le réseau et peu aux paquets qui les constituent. Dans ces réseaux, je m’intéresse à router les flux d’information, à agréger des flux d’information bas débits dans des flux de plus hauts débits, à faire évoluer le routage en cas de variations dans la quantité de trafic à transporter ou dans la topologie du réseau, et à assurer la continuité du trafic en cas de panne simple ou multiple. Pour aborder ces questions, j’utilise des outils variés de l’algorithmique, de la théorie des graphes et de l’optimisation combinatoire.

L’ensemble des résultats présentés dans ce document est le fruit de travaux collaboratifs avec les membres de l’équipe-projet Mascotte, des collègues d’autres universités, française ou étrangères, et des collègues de France Télécom, Alcatel-Lucent et 3-ROAM. Dans cette introduction, je résume nos travaux sur le routage, le groupage de trafic, la tolérance aux pannes et la reconfiguration. Je résume également des travaux plus récents sur la minimisation du nombre d’étiquette dans les réseaux MPLS, le dimensionnement de réseaux de collecte IP sans fil, ainsi que des travaux sur le routage disjoints d’ensembles particuliers de requêtes. Ensuite, je détaille nos travaux sur le groupage de trafic dans le chapitre 3, sur la notion de groupes de ressources partageant un risque dans le chapitre 4, et sur la reconfiguration de routages dans le chapitre 5.

Routage et affectation de longueurs d’ondes

Le problème du routage et de l’affectation de longueurs d’ondes (RWA) dans les réseaux optiques WDM, aussi appelé *routage optique*, est au cœur de nombreux problèmes d’optimisation dans ces réseaux. Il a donc suscité de nombreuses études au cours de 20 dernières années [212, 213, 214, 215, 218], le critère à optimiser étant généralement le nombre de longueurs d’ondes utilisées. Parmi ces travaux, (trop) nombreux sont ceux qui ont proposé des algorithmes heuristiques. Les

plus courants consistent à résoudre d’abord le problème du routage en utilisant une heuristique, puis à résoudre le problème de l’affectation de longueurs d’ondes (qui se modélise par un problème de coloration des sommets d’un graphe) en utilisant une autre heuristique. Rappelons que les problèmes du routage et de l’affectation de longueurs d’ondes sont NP-difficiles et difficiles à approcher [59, 60].

Afin de mieux comprendre pourquoi tant d’efforts étaient investis dans la conception d’algorithmes heuristiques pour le problème RWA, et quelle était la difficulté réelle de calculer des solutions optimales ou proche de l’optimale, j’ai étudié de plus près ce problème avec Hervé Rivano. Nous avons utilisé des techniques classiques de recherche opérationnelle pour résoudre le problème RWA. Les principes généraux sont d’une part de transformer les problèmes de routage et de l’affectation de longueurs d’ondes en un unique problème de multiflot entier dans un graphe auxiliaire, et d’autre part de considérer l’ensemble des requêtes issues d’une même source comme un unique flot entier afin de réduire le nombre de commodités à considérer (d’un facteur n). Nous avons de plus ajouté un *gadget* au graphe modélisant le réseau pour modéliser différents types de conversion en longueur d’onde. Nous avons également utilisé la capacité des arcs du graphe pour modéliser le nombre de fibres optiques sur chaque lien du réseau et donc le nombre de fois qu’une longueur d’onde peut être utilisée sur un même lien. Avec cette modélisation, le temps de résolution du problème, qui croît avec le nombre de longueurs d’ondes, se résout en quelques minutes avec un solveur comme CPLEX avec des instances standards (50 nœuds, 160 liens, 100 longueurs d’ondes). Ensuite, nous avons utilisé la technique de l’arrondi aléatoire de la relaxation fractionnaire du multiflot pour résoudre plus rapidement le problème [Ci20, Cn42]. Nous obtenons alors une solution de bonne qualité en quelques secondes. Enfin, nous avons utilisé un algorithme combinatoire d’approximation de multiflot fractionnaire pour approcher la relaxation fractionnaire du problème [Ci23, Cn43, Ci24]. Ceci nous permet d’aborder des problèmes de grandes tailles en temps raisonnable et avec une faible complexité en espace. Je renvoie à la thèse de Rivano [215] pour plus de détails sur ces travaux. Notons que nos algorithmes sont compétitifs vis-à-vis des formulations en programmes linéaires en nombres entiers avec génération de colonnes les plus performantes telles que celles proposées par Jaumard et al. [218], à la fois en temps de calculs et en qualité des solutions.

D’autre part, nous observons régulièrement que des auteurs, lorsqu’ils travaillent sur des instances “réelles” (topologie, trafic), remplacent le calcul de multiflot (entier) par de simples calculs de plus courts chemins. De façon surprenante, cette technique donne de bons résultats, mais personne ne sait aujourd’hui expliquer pourquoi. Aussi, une direction de recherche importante est d’étudier les spécificités structurelles des instances “réelles” afin de concevoir des algorithmes plus performants (temps de calcul, qualité de la solution).

Ces travaux ont été effectués en collaboration avec Jean-François Lalande (thèse Mascotte, puis MdC ENSI Bourges), Hervé Rivano (Mascotte, puis SWING, CITI, Lyon), et Mohamed Bouklit et Christophe Paul du LIRMM dans le cadre de la Color DYNAMIC.

J’ai co-encadré avec Hervé Rivano les stages de Marc Martinez de Albeniz (M2 UPC Barcelone en 2003, 6 mois) et Xavier Roche (L3 ENS Lyon en 2003, 6 semaines) sur le sujet.

Ceci a donné lieu à 5 publications dans des conférences [Ci20, Ci23, Ci24, Cn42, Cn43].

Groupage de trafic (chapitre 3)

La notion de *groupage de trafic* dans les réseaux correspond au fait de multiplexer des flux de données bas débits dans des flux de plus hauts débits afin d’optimiser l’utilisation des ressources des réseaux, et en particulier de simplifier les opérations de routage dans les nœuds intermédiaires. L’objectif est donc d’une part d’optimiser l’utilisation de la bande passante disponible

et d'autre part de minimiser la quantité de traitement à effectuer dans les nœuds du réseau et donc la complexité, le nombre et le coût des équipements à installer.

Nous nous sommes principalement intéressés au cas des réseaux en anneaux utilisant la technologie SONET/SDH, c'est-à-dire utilisant une hiérarchie digitale synchrone sur chaque longueur d'onde (multiplexage temporel de la bande passante). Dans ces réseaux, chaque longueur d'onde peut transporter g flux distincts (un flux correspond à une connexion), chacun utilisant une fraction $1/g$ de la capacité de cette longueur d'onde. Pour insérer et extraire les flux bas débits d'une longueur d'onde, nous utilisons un multiplexeur d'insertion/extraction (*add/drop multiplexer*, ADM). Nous utilisons un ADM sur une longueur d'onde à chaque nœud du réseau où il faut insérer ou extraire des flux. Il peut donc y avoir plusieurs ADMs par nœuds, un par longueur d'onde. Dans ce contexte, nous avons cherché à minimiser le nombre total d'ADMs à placer dans le réseau pour satisfaire un ensemble donné de connexions, en étudiant plus particulièrement le cas de l'échange total (une connexion entre chaque paire de nœuds du réseau). Nous avons montré que le problème du groupage d'un trafic d'échange total sur l'anneau unidirectionnel correspond à une partition des arêtes du graphe complet en sous-graphes à au plus g arêtes (où g est le facteur de groupage) et où il faut minimiser le nombre total de sommets. En utilisant des outils de la théorie des graphes et de la théorie des configurations (*design theory*), nous avons obtenu des constructions optimales pour les valeurs pratiques de N et de g , et pour des classes infinies de congruences pour des g donnés. Nous avons donné des constructions optimales pour $g \geq N(N-1)/6$ et pour $g = 4, 6, 12$. Nous avons également proposé des constructions approchées. D'autre part, nous avons obtenu des constructions optimales pour le chemin orienté avec $g = 2$. Par ailleurs, nous avons montré comment améliorer les bornes inférieures en utilisant des techniques de comptage adaptées, et comment déterminer le nombre maximum de connexions pouvant être établies sur une longueur d'onde d'un chemin de taille N .

Nos contributions sur ce sujet sont présentées dans le chapitre 3 au travers d'un état-de-l'art des contributions dans ce domaine.

Mes contributions sur le groupage de trafic ont été réalisées principalement en collaboration avec : Jean-Claude Bermond (Mascotte), Charles J. Colbourn (ASU, Tempe, USA), Michel Cosnard (Univ. Nice Sophia & INRIA), Gennian Ge (Zhejiang Univ., Chine), Alan Ling (Univ. Vermont, Burlington, USA), Xavier Muñoz (UPC, Barcelone, Espagne), Stéphane Pérennes (Mascotte). et Joseph Peters (SFU, Vancouver, Canada). De plus, des travaux sur la protection par cycles réalisés avec Lilian Chacon (Orange Labs, Lannion), François Tillerot (Orange labs, Lannion) et Min-Li "Joseph" Yu (SFU, Vancouver, Canada) répondent à certains problèmes de groupage.

J'ai co-encadré avec X. Muñoz la thèse d'Ignasi Sau (2006-2009, puis post-doc Technion, Haifa, Israel) sur le sujet, et j'ai encadré ou co-encadré 3 stages sur le sujet : Laurent Braud (L3 ENS Lyon, 2004, 6 semaines), Benjamin Lévêque (L3 ENS Lyon, 2002, 6 semaines), et Ignasi Sau (M2 UPC Barcelone, 2005-2006, 7 mois, co-encadrement avec J-C. Bermond).

Ceci a donné lieu aux publications suivantes : 4 revues [A5, A6, A7, A10], 3 chapitres de livres [B13, B12, B14] et 8 conférences [Cis36, Ci18, Ci21, Ci22, Ci25, Ci26, Ci27, Cn40].

Tolérance aux pannes (chapitre 4)

Assurer la continuité du trafic en cas de pannes d'équipements (nœuds ou liens) est une nécessité pour les réseaux de télécoms. Pour ce faire, deux types d'approches se distinguent : les méthodes pro-actives (protection) qui planifient les chemins de secours de chaque connexion au moment du dimensionnement du réseau ou de l'établissement d'une nouvelle connexion, et les méthodes réactives (restauration) qui déterminent au moment de la panne les chemins de

secours en fonction des ressources disponibles. Je me suis intéressé aux méthodes pro-actives, qui me semblent les plus adaptées dans les réseaux WDM.

Au cours de ma thèse, et pour répondre à une question posée par François Tillerot de France Télécom, nous avons abordé le problème de la protection par cycles. Il s'agit de protéger un ensemble de connexions routées le long d'un cycle dans le réseau en utilisant une longueur d'onde circulant en sens inverse. Cette notion a été introduite sous le nom de p -cycle dans [168] et a connue de nombreux développements par la suite (voir l'état-de-l'art [186]). Nous avons proposé un dimensionnement optimal pour l'instance all-to-all sur l'anneau [A5, Cis36, Ci18, Cn40]. Ces résultats ont également des applications pour le groupage de trafic sur l'anneau avec facteur de groupage $g = 1$ (voir chapitre 3). Je me suis également intéressé à la question de déterminer deux chemins arcs disjoints entre toute paire de sommets et dont la somme des coûts soient minimale. L'algorithme que j'ai proposé [Cn41] a la même complexité que celui de Suurballe et Tarjan [167], mais utilise plus directement la notion de chaîne augmentante.

Depuis, nous nous sommes intéressés à la notion de groupe de ressources partageant un risque (*shared risk resource group*, SRRG), c'est-à-dire un ensemble de ressources du réseau pouvant tomber en panne simultanément. Dans ce contexte, nous ne cherchons plus à optimiser le nombre d'arêtes d'un chemin ou d'une coupe, mais le nombre de SRRG auxquels il appartient. Nous avons étudié la complexité et l'(in)approximabilité de problèmes classiques d'optimisation combinatoire (chemin, coupe, arbre, chemins disjoints) dans ce contexte, en fonction de la structure des SRRG. Nous avons d'abord montré que de nombreux problèmes deviennent alors NP-complets et difficiles à approcher [A8, Ci28], puis nous avons caractérisé de nombreux cas où ces problèmes peuvent être résolus en temps polynomial [S49]. Ces travaux font l'objet du chapitre 4.

Les contributions du chapitre 4 ont été obtenues en collaboration avec : Pallab Datta (NSI, San Diego, USA), Florian Huc (thèse Mascotte, puis post-doc CUI Genève, Suisse), Fabrice Peix (Mascotte), Stéphane Pérennes (Mascotte), Hervé Rivano (Mascotte) et Marie-Emilie Vogé (thèse Mascotte puis MdC LIFL, Lille).

Elles ont donné lieu aux publications suivantes : 1 article de revue [A8], 1 article soumis à une revue [S49], et 2 articles de conférences [Ci29, Ci28].

Reconfiguration de routages (chapitre 5)

La configuration d'un réseau (WDM par exemple) correspond principalement au routage (et affectation de longueurs d'ondes) des requêtes de connexions, et peut aussi inclure le paramétrage courant des divers composants (routeurs, amplificateurs,...) du réseau. Pour optimiser l'utilisation des ressources suite à des évolutions de la matrice de trafic, ou pour interdire l'utilisation de certaines ressources devant subir une opération de maintenance, il peut-être nécessaire de changer la configuration du réseau. Il faut alors d'une part déterminer la nouvelle configuration et d'autre part planifier les changements à effectuer pour passer de la configuration courante à la nouvelle, tout en limitant les perturbations induites pour les clients (interruptions de trafic). Ce problème se retrouve dans tous les types de réseaux orientés connexions et est étudié au moins depuis les années 70 pour les réseaux téléphoniques [190].

Ce problème étant très difficile à résoudre, nous nous sommes intéressés uniquement à la deuxième partie du problème qui consiste, étant données les configurations courante et finale, à planifier l'ordre dans lequel sont effectués les changements de routes de chaque connexion afin de minimiser les interruptions de services. A notre connaissance, seuls Jose et Somani [200] s'étaient intéressés à cette question en proposant une heuristique pour minimiser le nombre total d'interruptions au cours de la reconfiguration. Comme dans [200], nous modélisons les dépendances entre connexions par un graphe orienté (le graphe de dépendances) dont les sommets correspondent aux connexions devant changer de route, et il y a un arc d'un sommet u vers un

sommet v si la connexion v utilise dans la configuration initiale des ressources que doit utiliser u dans la configuration finale. Lorsque ce graphe contient des cycles, il est nécessaire d’interrompre temporairement certaines requêtes. Nous modélisons une interruption par le placement d’un agent sur le sommet correspondant, et nous étudions un jeu similaire aux jeux des gendarmes et du voleur, où l’objectif est de minimiser soit le nombre total de sommets visité par un agent au cours de la stratégie (nombre total d’interruptions), soit le nombre d’agents nécessaires à l’existence d’une stratégie (nombre simultané d’interruptions), soit des compromis [S55]. Nous avons entre autre montré la complexité et l’inapproximabilité du problème [Cn44], caractérisé les graphes pour lesquels il existe un stratégie utilisant au plus 2 agents [S50], et proposé des algorithmes heuristiques [Ci33, S52].

Ces travaux nous ont amené à étudier des invariants de graphe comme la largeur de chemin (*pathwidth*) et l’indice d’échappement (*node search number*). En particulier, nous avons montré que la largeur de chemin d’un graphe planaire extérieur et celle de son dual géométrique diffèrent d’un facteur au plus 2 [A9], infirmant ainsi plusieurs conjectures sur cet écart [77, 80]. Nous avons également proposé un algorithme distribué permettant de calculer plusieurs invariants (largeur de chemin, indice d’échappement, indice de traitement,...) dans les arbres, ainsi qu’une version dynamique de cet algorithme permettant l’ajout ou la suppression d’arêtes [Cn45, S51, Cis38, Cis37].

Les travaux présentés dans le chapitre 5 ont été effectués en collaboration avec : Florian Huc (thèse Mascotte puis post-doc CUI Genève), Dorian Mazauric (thèse Mascotte), Nicolas Nisse (Mascotte), Stéphane Pérennes (Mascotte) et Jean-Sébastien Séréni (thèse Mascotte puis CR CNRS au LIAFA, Paris).

J’ai co-encadré avec J-C. Bermond la thèse de Florian Huc (2005-2008) dont une partie des travaux portaient sur le sujet, et j’ai encadré ou co-encadré les stages de Arnaud Daver (stage M2 en 2004, 4 mois), Quang-Cuong Pham (stage L3 en 2004, 2 mois, co-encadrement avec S. Pérennes), Dorian Mazauric (stage M1/EPU2 en 2007, 3 mois, co-encadrement avec F. Huc), Ronan Soares (stage M2 en 2009, 3 mois, co-encadrement avec N. Nisse) et Saber Ben Nejma (PFE Sup’Com Tunis en 2009, 4 mois, co-encadrement avec N. Nisse).

Mes travaux ont donné lieu aux publications suivantes : 1 revue [A9], 7 conférences [Ci34, Ci33, Cn46, Cis37, Cis38, Cn45, Cn44], 3 articles en cours de soumissions [S50, S51, S55], et [S52, S48].

Réseaux MPLS

La technologie MPLS (*MultiProtocol Label Switching*) est un mécanisme conçu pour fournir un service unifié de transport de données pour les clients en utilisant une technique de commutation de paquets. Avec cette technologie, les paquets d’une même connexion sont transmis de nœud en nœud le long d’un chemin, appelé LSP (*Labeled Switched Path*), vers la destination. Chaque nœud intermédiaire utilise l’étiquette du paquet pour décider du prochain lien à traverser. Cependant cette étiquette est propre à la table de routage de chaque nœud. Aussi, un objectif important est de réduire le nombre total d’étiquettes différentes à utiliser dans le réseau. Ceci est crucial avec les technologies telles que AOLS (*All-Optical Label Switching*) [217, 216] qui proposent d’utiliser le même principe de routage que MPLS en restant dans le domaine optique. Chaque étiquette nécessite alors de placer un équipement particulier dans le nœud.

Plusieurs méthodes ont été proposées pour réduire le nombre d’étiquettes et nous nous sommes intéressés à l’empilement d’étiquettes (*label stacking*) qui permet d’empiler plusieurs étiquettes. Dans ce cas, un routeur ne peut cependant en lire qu’une (celle du sommet de la pile). Avec cette méthode, l’ensemble du chemin peut par exemple être encodé dans un paquet. Ceci simplifie les opérations de routage aux nœuds intermédiaires, mais induit un surcoût important de signalisation. De plus, cette méthode permet de créer des tunnels dans le réseau afin

d'agréger plusieurs LSPs le long d'un chemin et ainsi de réduire le nombre d'étiquettes.

Nous avons étudié [Ci30, Ci31, S53] le problème de choisir les tunnels à utiliser pour minimiser le coût du routage, où le coût d'un tunnel est alors associé à sa longueur en nombre d'arêtes et au nombre de LSPs qu'il transporte, dans le cas où la taille des piles est limitée à 2. En utilisant une modélisation des tunnels à base d'hypergraphes orientés, nous avons montré que ce problème est NP-difficile en général et difficile à approcher à un facteur $c \log n$, où c est une constante positive et n le nombre de nœuds du réseau. Lorsque le graphe est orienté symétrique, nous avons d'abord montré que le problème est difficile à approcher à un facteur constant, puis nous avons proposé un algorithme $\log n$ -approché. Nous nous sommes ensuite concentré sur le cas où le réseau est un chemin orienté et proposé un algorithme exact en $O(n^{k+1})$ à base de programmation dynamique, où k est le nombre de sources dont est issue le trafic.

Ces travaux apportent les premiers résultats théoriques dans ce domaine. Ils ont été effectués en collaboration avec : Jean-Claude Bermond (Mascotte), Joanna Moulierac (Mascotte), Stéphane Perennes (Mascotte), Hervé Rivano (Mascotte puis SWING, Lyon) et Ignasi Sau de Mascotte et Fernando Solano Donado (Univ. Warsaw, Pologne).

Ils ont donné lieu à 2 publications dans des conférences [Ci30, Ci31] et 1 article actuellement soumis à une revue [S53].

Réseaux de collecte IP sans fil

Dans le cadre d'une collaboration avec la PME 3-ROAM, basée à Sophia Antipolis, nous travaillons sur des problèmes d'optimisation dans des réseaux de collecte IP sans fil, appelé *wireless backhaul networks*. Les liens de ces réseaux sont des faisceaux hertziens, c'est-à-dire que les informations sont transmises par micro-ondes d'une antenne vers une autre sur une distance pouvant aller jusqu'à 50km. La bande passante offerte par ces liens est de l'ordre de 500Mbit/s. Ces réseaux sont faciles à déployer et permettent de relier des zones où il est difficile, voire prohibitif, d'apporter l'accès à l'internet haut débit par fibre optique (e.g., village reculé, zone montagneuse). De plus, ces réseaux utilisent une technologie peu coûteuse. Ils ont donc un intérêt économique et territorial certain.

Sur ces réseaux, nous étudions des problèmes similaires à ceux que nous étudions sur les réseaux WDM ou MPLS : dimensionnement, routage, tolérance aux pannes, optimisation dynamique du routage,... Toutefois, il y a une différence fondamentale entre ces réseaux et les réseaux filaires : la bande passante des liens du réseau est sujette à variation. Tout d'abord, les changements météorologiques (pluie, brouillard, chaleur) peuvent dégrader la qualité de la transmission, ce qui peut entraîner une diminution de la bande passante de certains liens, voire les rendre inopérants. D'autre part, il est possible d'adapter dynamiquement la bande passante de ces liens en fonction des besoins en en modifiant la configuration (choix de la modulation, de la puissance d'émission du signal, du codage ou encore du taux d'erreurs). Ceci ouvre des perspectives très intéressantes dans l'étude des problèmes d'optimisation habituels.

Avec Napoleão Nepomuceno et Hervé Rivano, nous avons tout d'abord étudié le problème de déterminer la meilleure configuration d'un lien pour offrir une bande passante donnée. Puis, nous avons étudié le problème de déterminer la configuration de chaque lien et le routage des informations afin d'optimiser la consommation globale d'énergie du réseau. Ces gains sont modestes en comparaison de la consommation des routeurs, mais c'est un début.

J'encadre actuellement les thèses de Napoleão Nepomuceno (depuis décembre 2007) et d'Issam Tahiri (depuis novembre 2009) sur le sujet. Nos travaux ont déjà donné lieu à 1 publication dans une revue [A11] et 2 publications dans des conférences [Ci35, Cn47]. Notons également que le modèle de lien que nous avons proposé et que Napoleão a implémenté est actuellement utilisé par 3-ROAM.

Je m'intéresse également à d'autres questions. Par exemple, au cours de ma thèse, nous avons cherché à déterminer le plus grand k tel que tout sous ensemble de k sommets ordonnés d'une grille torique (suffisamment grande) soit inclus dans un circuit arête simple (une arête n'est utilisée qu'une fois). Nous avons montré que $k \leq 11$ et conjecturé que $k = 11$ [T1, section 2.4]. Nous avons depuis infirmé cette conjecture avec Ignasi Sau et Frédéric Giroire (Mascotte) en montrant que $k = 10$ [Ci32, S56].

Chapitre 2

Introduction (in english)

This document describes my scientific activities after my PhD thesis, but some results were undertaken during my PhD.

As a reminder, my PhD thesis was about algorithms and optimization in optical communication networks. The first part focused on the design and characterization of free space optical interconnection networks using the OTIS architecture (*Optical Transpose Interconnection System*). I refer to [T1, A2, A3, A4, Ci15, Ci16, Ci17, Ci19, Cn39, Cn41] for all our results on the “*OTISeries*”. The second part concerns protection mechanisms in WDM networks [T1, Cis36, Ci18, Cn40, Cn41], and I continue to be interested in these issues.

The general contexts of my research activity are the connection oriented networks, either WDM (wavelength division multiplexing) optical network, or MPLS (Multi-protocol label switching), or wireless backhaul networks using microwave links. I am more interested in the data flows in transit in the network rather than the packets they carry. In such networks, I am interested in routing data flows, in aggregating low rate traffic streams into higher speed one, in updating the routing after traffic or topological modifications, and in ensuring the traffic continuity under single or multiple failures scenarios. To address these issues, I use a variety of tools from algorithmics, graph theory and combinatorial optimization.

All results presented in this document are the result of collaborative work with members of the project-team Mascotte, colleagues from other universities (in France and abroad), and colleagues from France Telecom, Alcatel-Lucent and 3-ROAM. This introduction summarizes our work on routing, traffic grooming, fault tolerance and reconfiguration. It also summarizes more recent work on minimizing the number of labels in MPLS networks, the design of wireless backhaul networks, as well as the disjoint routing of particular set of requests. I present in more details our work on traffic grooming in Chapter 3, on of shared risk resource groups in Chapter 4, and on the reconfiguration of routing in Chapter 5.

Routing and wavelength assignment

The routing and wavelength assignment (RWA) problem in WDM optical networks, also known as *optical routing* problem, is central to many optimization problems. It has motivated numerous studies over the past 20 years [212, 213, 214, 215, 218], one of the main criteria to optimize being the number of wavelengths used. Among these works, (too) many have proposed heuristic algorithms. The most common approach is to solve first the routing problem using heuristics, and then solve the wavelength assignment problem (which is modeled as a graph coloring problem) using another heuristic. Recall that both problems are NP-hard and difficult to approximate [59, 60].

With Hervé Rivano, we have studied the RWA problem with the aim of better understanding why so many efforts were invested in the design of heuristic algorithms for the RWA problem, and what was the real difficulty encountered in the computation of optimal and near-optimal solutions. We used standard techniques of operational research to solve the RWA problem. The general principles are : to transform the routing and the wavelength assignment problems into a single multi-commodity flow problem on an auxiliary graph, and to consider the set of requests issued from a same node as a single integral commodity in order to reduce (by a factor N) the number of commodities to consider. We have added a *gadget* to model various types of wavelength conversions functionalities. We have also used the capacity of the arcs to model the number of optical fibers on each link of the network, and therefore the number of times a given wavelength can be used on a same link. Clearly, with this model the computation time increases with the number of wavelengths. However, solvers like CPLEX are able to compute optimal solutions in a few minutes on standard instances (50 nodes, 160 links, 100 wavelengths). We then used randomized rounding techniques on the fractional relaxation of the multi-commodity flow to obtain faster solutions [Ci20, Cn42]. We then get good solutions in a few seconds. Finally, we used a combinatorial approximation algorithm for the fractional multi-commodity flow in order to approximate the fractional relaxation of the problem [Ci23, Ci24, Cn43]. This allows us to tackle large problems in reasonable time and with a low space complexity. I refer to the thesis of Rivano [215] for more details on this work. Note that our algorithms are competitive with respect to the most powerful integer linear programming formulations using column generation proposed by Jaumard et al. [218] in both computation time and quality of the solutions.

On the other hand, we observe that many authors replace the computation of the integral multi-commodity flow by simple shortest paths computation, when working on “real” instances (topology, traffic). Surprisingly, this technique gives good results, but today no one is able to explain why. Also, an important research direction is to study the structural characteristics of “real” instances in order to design more efficient algorithms (computation time, quality of the solution).

These studies have been done in collaboration with Jean-François Lalande (PhD thesis Mascotte, now MdC ENSI Bourges), Hervé Rivano (Mascotte, now SWING, CITI, Lyon), and Mohamed Bouklit and Christophe Paul of the LIRMM laboratory in the context of the Color grant DYNAMIC.

I have co-supervised with Hervé Rivano on this subject the internships of : Marc Martinez de Albeniz (M2 UPC Barcelona in 2003, 6 months); Xavier Roche (L3 ENS Lyon in 2003, 6 weeks).

This work has been published in 5 conferences [Ci20, Ci23, Ci24, Cn42, Cn43].

Traffic grooming (chapter 3)

The notion of *traffic grooming* refers to techniques used to combine low speed traffic streams onto high speed wavelengths in order to minimize the network-wide cost in terms of electronic switching. The main objectives are to optimize the usage of available bandwidth and to minimize the number of switching operations to perform in network nodes, and so the complexity, the number and the cost of the equipments to be installed in router nodes.

We have mainly considered SONET/SDH rings that use a synchronous digital hierarchy (time division multiplexing) on each wavelength of a WDM ring network. In these networks, each wavelength can carry up to g distincts streams (each using $1/g$ of the bandwidth of a wavelength). We use add/drop multiplexers (ADMs) to add and drop low rate traffic streams from the wavelength. We place one ADM on a wavelength at each node in which traffic is added or dropped from that wavelength. Also, we may install several ADMs at the same node, one per wavelength.

In this context, we are interested in minimizing the total number of ADMs to install in the network in order to satisfy a given traffic requirement. We have particularly studied the case of all-to-all traffic (one connection request between each pair of nodes) on networks with path and ring topologies. For ring topology, we have shown that the traffic grooming problem corresponds to a partition of the edges of the complete graph into subgraphs, where each subgraph has at most g edges (where g is the grooming ratio) and where the total number of vertices has to be minimized [Ci22, B12]. Using tools of graph and design theory, we optimally solved the problem for practical values [Ci21] and infinite congruence classes for a given g . We gave optimal constructions on unidirectional rings when $g \geq N(N-1)/6$ and when $g = 4, 6, 12$ [Ci22, A5, A6], and on paths when $g = 2$ [A7, Ci25]. We provided approximate constructions on unidirectional rings for all values of g [A10]. We also showed how to improve lower bounds by using refined counting techniques [A6], and how to determine the maximum number of connections which can be established in a wavelength of a path of size N [Ci26].

Our contributions on traffic grooming are presented in Chapter 3.

My contributions on traffic grooming have been done in collaboration with : Jean-Claude Bermond (Mascotte), Charles J. Colbourn (ASU, Tempe, USA), Michel Cosnard (Univ. Nice Sophia & INRIA), Gennian Ge (Zhejiang Univ., China), Alan Ling (Univ. Vermont, Burlington, USA), Xavier Muñoz (UPC, Barcelona, Spain), Stéphane Pérennes (Mascotte) and Joseph Peters (SFU, Vancouver, Canada). Furthermore, our studies on protection by cycles [A5, Cis36, Ci18, Cn40] done with Lilian Chacon (Orange Labs, Lannion), François Tillerot (Orange labs, Lannion) and Min-Li “Joseph” Yu (SFU, Vancouver, Canada) answer some traffic grooming problems (all-to-all traffic grooming in uni/bidirectional ring with $g = 1$).

On this topic, I have co-supervised with X. Muñoz the PhD thesis of Ignasi Sau (2006-2009, now post-doc Technion, Haifa, Israel), and I have supervised or co-supervised 3 internships : Laurent Braud (L3 ENS Lyon, 2004, 6 weeks), Benjamin Lévêque (L3 ENS Lyon, 2002, 6 weeks), and Ignasi Sau (M2 UPC Barcelone, 2005-2006, 7 months, co-supervision with J-C. Bermond).

These results have been published in : 4 journals [A5, A6, A7, A10], 3 book chapters [B12, B13, B14] and 8 conferences [Cis36, Ci18, Ci21, Ci22, Ci25, Ci26, Ci27, Cn40].

Fault tolerance (Chapter 4)

Ensuring traffic continuity in the event of equipment failures (nodes or links) is a necessity for telecom operators. To do this, two approaches emerge : the pro-active methods (protection) that reserve backup paths for each connection when designing the network or establishing a new connection, and reactive methods (restoration) that determine backup paths using available resources when the failure occurs. I am interested in pro-active methods that seem more appropriate in WDM networks.

During my PhD thesis, and in order to answer a question from François Tillerot (France Telecom), we addressed the problem of protection by cycles. It consists in protecting a set of connections routed along a cycle in the network using a wavelength traveling in opposite directions. This concept is known as the p -cycles, introduced in [168] and widely studied (see the survey [186]). We have proposed an optimal design for all-to-all traffic on the ring topology [A5, Cis36, Ci18, Cn40]. These results have also applications for traffic grooming on rings with grooming factor $g = 1$ (see Chapter 3). I was also interested in the question of determining two arcs disjoint paths between any pair of vertices whose total cost are minimal. The algorithm that I proposed [Cn41] has the same complexity than the one of Suurballe and Tarjan [167], but it uses more directly the concept of augmenting path.

Then, we have addressed the notion of shared risk resource group (SRRG), that is a set of resources of the network that may fail simultaneously. In this context, we are no longer

interested in minimizing the number of edges of a path or a cut, but the number of SRRG to which it belongs. We have established complexity and hardness results for some classical combinatorial optimization problems (path, cut, tree, disjoint paths) in this context. Our results take into account the structural properties of the SRRG. We have also proposed approximation algorithms. We have first showed that many problems become NP-complete and difficult to approximate in the SRRG context [A8, Ci28]; then we have characterized many cases where these problems can be solved in polynomial time [S49]. These studies are detailed in Chapter 4.

The contributions of Chapter 4 have been done in collaboration with : Pallab Datta (NSI, San Diego, USA), Florian Huc (PhD thesis Mascotte under my (co)direction, now post-doc CUI Geneva, Switzerland), Fabrice Peix (Mascotte), Stéphane Pérennes (Mascotte), Hervé Rivano (Mascotte) and Marie-Emilie Vogt (PhD thesis Mascotte under my (co)direction, now M&C LIFL, Lille).

This work has been published in : 1 journal [A8], 1 paper submitted to a journal [S49], and 2 conferences [Ci28, Ci29].

Routing reconfiguration (Chapter 5)

The configuration of a network (e.g. WDM) corresponds mainly to the routing (and wavelengths assignment) of connection requests, and may also include the current setting of the devices (routers, amplifiers,...) of the network. To optimize the usage of resources with the evolution of the traffic matrix, or to avoid using particular resources subject to maintenance operation, it may be necessary to change the configuration of the network. It is then required to first determine the new configuration and then to schedule necessary changes to switch from the current configuration to the new one, while limiting possible traffic perturbations to customers (traffic disruption). This problem exists in all types of connection oriented networks and is studied since at least the 70th for telephone networks [190].

Since this problem is very difficult to solve, we have considered only the second part of the problem : given the current and final configurations, how to schedule the route changes of each connection in order to minimize service disruptions. To the best of our knowledge, this problem had only been considered by Jose and Somani [200]. They have proposed a heuristic to minimize the total number of disruption during the reconfiguration. As in [200], we model the dependencies between connections with a directed graph (the dependency digraph) whose vertices correspond to the connections that have to change of routes, and there is an arc from a vertex u to a vertex v if connection v uses in the initial configuration resources that must be used by u in the final configuration. When the graph contains cycles, it is necessary to suspend (disrupt) temporarily some connections. We model a disruption by placing an agent on the corresponding node, and we study a cops-and-robber like game, in which the objective is to minimize either the total number of vertices visited by an agent during the strategy (total number of disruptions), or the number of agents needed for the existence of a strategy (number of simultaneous disruptions), or tradeoffs [S55]. We have among others shown the complexity and the hardness of the problem [Cn44], characterized graphs for which there is a strategy using at most 2 agents [S50] and proposed heuristic algorithms [Ci33, S52].

This work led us to study graph invariants such as the pathwidth and the node search number. In particular, we have shown that the pathwidth of a biconnected outerplanar graph and that of its geometric dual differ by a factor of at most 2 [A9], thus disproving several conjectures on this gap [77, 80]. We have also proposed a distributed algorithm to compute several invariants (pathwidth, node search number, process number,...) in trees, and a dynamic version of this algorithm allowing to add or delete any tree edges [Cis37, Cis38, Cn45, S51].

These studies are reported in Chapter 5 and have been done in collaboration with : Florian Huc (PhD thesis Mascotte, now post-doc CUI Geneva), Dorian Mazauric (PhD thesis Mascotte),

Nicolas Nisse (Mascotte), Stéphane Pérennes (Mascotte) and Jean-Sébastien Séréni (PhD thesis Mascotte, now CR CNRS, LIAFA, Paris).

I have co-supervised with J-C. Bermond the PhD thesis of Florian Huc (2005-2008), part of which were on this topic, and I have supervised or co-supervised the following internships : Arnaud Daver (DEA RSD, 2004, 4 months), Quang-Cuong Pham (L3 ENS ULM, 2004, 2 months, co-supervision with S. Pérennes), Dorian Mazauric (M1/EPU2, 2007, 3 months, co-supervision with F. Huc), Ronan Soares (M2 Fortaleza, 2009, 3 months, co-supervision with N. Nisse) and Saber Ben Nejma (PFE Sup'Com Tunis, 2009, 4 months, co-supervision with N. Nisse) .

This work has been published in : 1 journal [A9], 7 conferences [Cis37, Cis38, Ci33, Ci34, Cn44, Cn45, Cn46], 3 articles under submission [S50, S51, S55], and 2 articles in preparation [S52, S48].

MPLS networks

MPLS (Multiprotocol Label Switching) is a mechanism designed to provide a unified data transport service for clients using packet switching. With this technology, the packets of a same connection are transmitted from node to node along a path, called LSP (Labeled Switched Path), toward the destination. Each intermediate node uses the packet label to determine the next link to follow. But this label is specific to the routing table of each node. Also, an important goal is to reduce the total number of labels used in the network. This is crucial with technologies such as AOLS (All-Optical Label Switching) [216, 217] that aim at using the same forwarding principle than MPLS, but remaining in the optical domain. Thus, each label used requires to install a specific device in the node.

Several methods have been proposed to reduce the number of labels and we are interested in label stacking which allows to stack multiple labels in each packets. In this case, a router can read only the label at the top of the stack. With this method, the entire path may for example be encoded in a packet. This simplifies the routing operations for intermediate nodes, but induces a significant signaling overhead. Moreover, label stacking allows to create tunnels in the network in order to aggregate multiple LSPs along a path and thus reduce the number of labels.

We have studied [Ci30, Ci31, S53] the problem of choosing the best set of tunnels to use in order to minimize the cost of routing, when the size of the stacks is limited to 2, and where the cost of a tunnel is associated with its length in number of edges and with the number of LSPs it carries. We showed that this problem is NP-hard in general and difficult to approximate within a factor $c \log n$, where c is a positive constant and n the number of network nodes. When the digraph is symmetric, we first showed that the problem is difficult to approximate within a constant factor, and then we have proposed a $\log n$ -approximation algorithm. When the network is a directed path, we proposed an exact dynamic programming algorithm with time complexity $O(n^{k+1})$, where k is the number of sources sending traffic.

These studies provide the first theoretical results in this context. They have been done in collaboration with : Jean-Claude Bermond (Mascotte), Joanna Moulrierac (Mascotte), Stéphane Pérennes (Mascotte), Hervé Rivano (Mascotte, now SWING, Lyon), Ignasi Sau (PhD thesis Mascotte under my (co)direction, now post-doc Technion, Haifa, Israel) and Fernando Solano Donado (Univ. Warsaw, Pologne).

This work has been published in : 2 conferences [Ci30, Ci31] and 1 paper submitted to a journal [S53].

Wireless backhaul networks

In the context of collaboration with the SME-3 ROAM, based in Sophia Antipolis, we are working on optimization problems in wireless backhaul networks. These networks use directed

microwave links to transmit data from one antenna to another over a distance of up to 50km. The bandwidth offered by these links is around 500Mbit/s. These networks are easy to deploy and allow to reach remote places where it is difficult or prohibitive to provide access to broadband Internet via optical fibers (e.g., remote villages, islands,...). Moreover, these networks use a relatively cheap technology. They are therefore interesting in both economic and territorial terms.

On these networks, we study similar problems to those we are studying on the WDM and MPLS networks : design, routing, fault tolerance, dynamic optimization of the routing,... However, there is a fundamental difference between these networks and wired networks : the bandwidth of network links is subject to variations. First, weather changes (rain, fog, heat) can affect (reduce) the quality of the transmission, which may lead to a reduction of the bandwidth of some links, or even to make them unusable. On the other hand, it is possible to dynamically adjust the bandwidth of these links by modifying their configuration (choice of the modulation, transmission power, coding, or bit error rate). This opens interesting research perspectives in the study of classical optimization problems.

With Napoleão Nepomuceno and Hervé Rivano, we have first studied the problem of determining the best configuration of a link to provide a given bandwidth. Then, we studied the problem of determining jointly the configuration of each link and the routing in order to optimize the overall energy consumption of the network. Our gains are modest in comparison with the power consumption of routers, but it's a beginning.

I currently supervise the PhD thesis of Napoleão Nepomuceno (since December 2007) and Issam Tahiri (since November 2009) on this topic. Our studies have been published in 1 journal publication [A11] and 2 conferences [Ci35, Cn47]. Note also that the modeling of the link that we have proposed has been implemented by Napoleão and is currently used by 3-ROAM.

I'm also interested in several questions. For example, during my PhD thesis, we tried to determine the largest k such that every subset of k ordered vertices of a sufficiently large torus is included in an edge simple circuit (an edge is used only once). We showed that $k \leq 11$ and conjectured that $k = 11$ [T1, section 2.4]. We have recently disproved this conjecture with Ignasi Sau and Frédéric Giroire (Mascotte) and proved that $k = 10$ [Ci32, S56].

Chapitre 3

Groupage de trafic

La notion de *groupage de trafic* dans les réseaux correspond à la façon de multiplexer des flux de données bas débits dans des flux de plus hauts débits afin d'optimiser l'utilisation des ressources des réseaux. L'objectif est d'une part d'optimiser l'utilisation de la bande passante disponible et d'autre part de minimiser la quantité de traitement à effectuer dans les nœuds du réseau et donc la complexité et le coût des équipements à installer. Cette notion se décline de plusieurs façons selon la technologie considérée.

Dans les réseaux SONET/SDH (*Synchronous Optical Networking / Synchronous Digital Hierarchy*) qui utilisent une hiérarchie digitale synchrone sur chaque longueur d'onde (multiplexage temporel de la bande passante), il s'agit de partager la bande passante d'une longueur d'onde entre au plus g flux distincts, chacun utilisant une fraction $1/g$ de la capacité de cette longueur d'onde. Pour insérer et extraire les flux bas débits d'une longueur d'onde, nous utilisons un multiplexeur d'insertion/extraction (*add/drop multiplexer*, ADM). Nous utilisons un ADM sur une longueur d'onde et à chaque nœud du réseau où il faut insérer ou extraire des flux (voir figure 3.1). Il peut donc y avoir plusieurs ADMs par nœuds, un par longueur d'onde. Dans ce contexte, nous cherchons à minimiser le nombre total d'ADM à placer dans le réseau pour satisfaire un ensemble donné de connexions, comme nous le verrons dans la suite de ce chapitre.

Dans les réseaux MPLS (*MultiProtocol Label Switching*), c'est principalement l'utilisation de tunnels qui permet d'agréger des flux de trafic [Ci30, Ci31, S53]. En effet, un tunnel permet de router des flux distincts d'un point à un autre du réseau en utilisant la même étiquette, au lieu d'une étiquette distincte pour chacun de ces flux. Ceci réduit le nombre d'opérations distinctes à effectuer aux nœuds intermédiaires, et donc simplifie le routage.

D'autres propositions ont été faites pour exploiter efficacement la bande passante des longueurs d'ondes en agrégeant des flux de granularités plus faibles. Par exemple, un *light-trail* permet d'utiliser une longueur d'onde sur un chemin comme un bus [139, 150]. L'ensemble des nœuds d'un *light-trail* peut transmettre des informations vers les nœuds suivants. Les difficultés sont alors d'une part de contrôler l'accès au *light-trail* (collisions, équité) et d'autre part de choisir les *light-trails* à établir dans le réseau pour satisfaire les besoins en communication. Le concept de *light-trail* a été étendu à la technologie MPLS avec les FAMAs (*forwarding adjacency multiple access*) [146]. Il s'agit alors d'utiliser des LSPs (*labeled switched path*) comme des bus. Notons que les tunnels MPLS peuvent également être considérés comme des sortes de bus. En effet, un tunnel MPLS est à entrées multiples (il est possible d'insérer un LSP depuis n'importe quel point du tunnel) mais à destination unique (tous les LSPs sortent du tunnel au même nœud). Ces concepts conduisent à ne plus considérer des communications point-à-point, mais des ensembles de communications à modéliser par des hyper-arêtes ou des hyper-arcs comme nous l'avons fait dans [S53].

Un autre niveau de groupage de trafic est la notion de *waveband switching* [129, 140, 143,

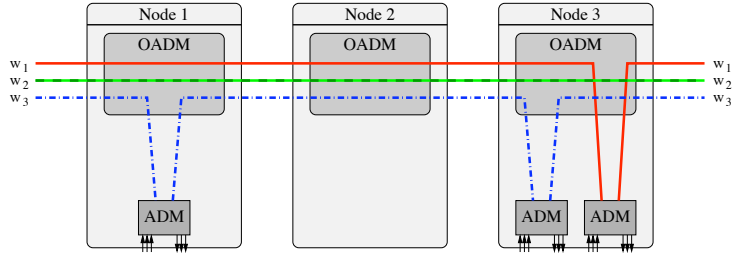


FIG. 3.1 – Placement d’ADMs dans le réseau : un ADM pour chaque longueur d’onde utilisée dans un nœud.

147, 151] qui consiste à effectuer une même opération de routage sur un ensemble de longueurs d’ondes, appelé bandes. Une bande est par exemple un ensemble de 4 ou 8 longueurs d’ondes contiguës (les longueurs d’ondes 1 à 4 ou 1 à 8) ou formant un motif régulier (longueurs d’ondes paires). Toutes les longueurs d’ondes d’une même bande traversent un nœud du réseau de la même façon et l’opération de routage est réalisée par un équipement unique. Il s’agit alors de résoudre le problème du routage et de l’affectation des longueurs d’onde en prenant en compte la notion de bandes afin de réduire le coût des opérations de routage.

Je renvoie le lecteur aux états-de-l’art auxquels j’ai contribué [B13, B12, B14], à ceux écrits par d’autres chercheurs [120, 124, 126, 130, 133], ainsi qu’aux ouvrages [166, 165, 164] pour de plus amples informations sur la notion de groupage de trafic.

Dans la suite, je présente les travaux que nous avons effectué sur le groupage de trafic dans les réseaux WDM utilisant la technologie SONET/SDH. Nous nous sommes concentrés sur le problème de minimiser le nombre total d’ADMs à installer dans des réseaux simples tels que le chemin orienté et l’anneau. Je présente nos travaux au travers d’un état-de-l’art des techniques utilisées et des principaux résultats. Je commence dans la section 3.1 par le cas du groupage d’un trafic d’échange total sur l’anneau unidirectionnel. J’y présente la modélisation que nous avons proposée pour aborder ce problème, les techniques utilisées pour établir des bornes inférieures et les résultats optimaux connus. Ensuite, dans la section 3.3, je présente les résultats de complexité et d’(in)approximabilité pour le problème du groupage de trafic quelconque sur l’anneau. Je relate également les principaux résultats dans le cas particulier où $g = 1$. Puis, dans la section 3.2, je traite du groupage de trafic sur le chemin. Enfin, je conclus ce chapitre dans la section 3.4 au travers d’une discussion sur la portée de nos contributions et les directions de recherches futures qui me semblent les plus intéressantes.

3.1 Echange total sur l’anneau unidirectionnel

3.1.1 Exemple

Sur l’anneau unidirectionnel, toutes les requêtes sont routées dans le même sens (sens des aiguilles d’une montre par exemple). Aussi, on peut router les requêtes (i, j) et (j, i) sur la même longueur d’onde en utilisant 2 ADMs et une fraction $\frac{1}{g}$ de la capacité de la longueur d’onde sur l’anneau. Nous avons montré que ce choix est optimal lorsque l’ensemble de requêtes est symétrique dans [Ci22]. De plus, dans ce cas, l’ensemble de requêtes est modélisé par un graphe non-orienté, chaque arête modélisant une paire de requêtes symétriques. Alors, un sous-graphe B à g arêtes du graphe modélisant l’ensemble de requêtes correspond à une affectation valide de g paires de requêtes à une longueur d’onde. Le nombre de sommets de B donne le nombre d’ADMs à utiliser sur cette longueur d’onde.

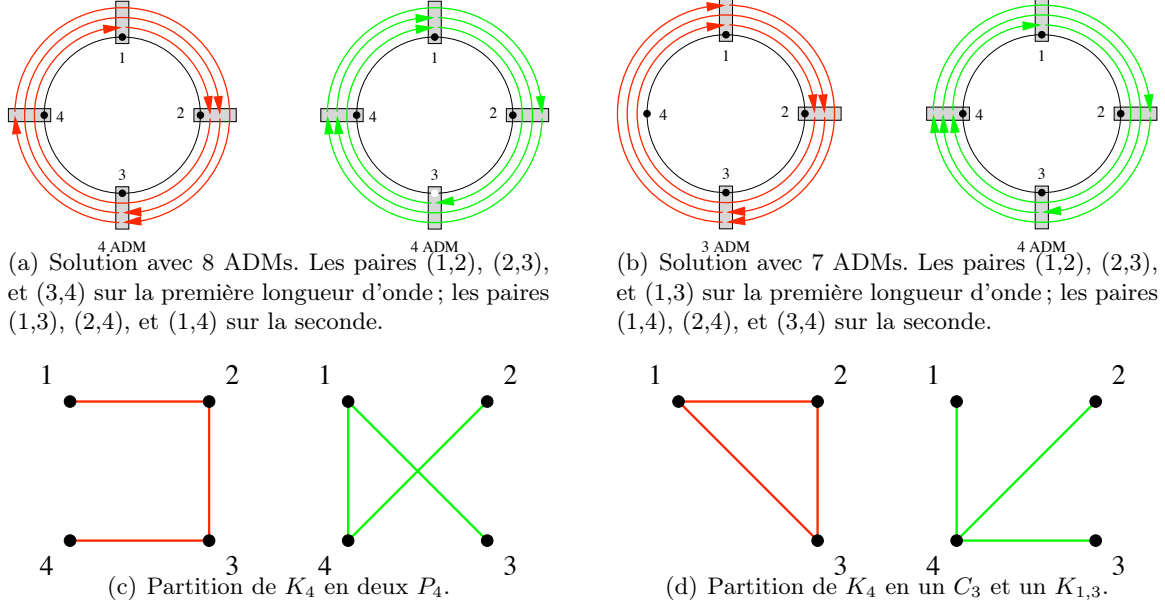


FIG. 3.2 – Groupage de trafic sur l’anneau unidirectionnel à 4 sommets, un facteur de groupage $g = 3$ et l’ensemble de requêtes $I = K_4^*$ (échange total). La solution 3.2(a) utilise 8 ADMs alors que la solution 3.2(b) n’utilise que 7 ADMs.

Considérons l’exemple de l’anneau unidirectionnel \vec{C}_4 à 4 sommets, $\{1, 2, 3, 4\}$, avec l’ensemble K_4^* de requêtes (une requête unitaire entre chaque paire de sommets). Comme nous devons placer un ADM à chaque extrémité d’une requête et que le routage est unique, nous pouvons placer la paire de requêtes (i, j) et (j, i) sur la même longueur d’onde. Nous utilisons alors $1/g$ de la capacité de cette longueur d’onde sur l’anneau. Il y a 6 paires de requêtes (i, j) et (j, i) , $1 \leq i < j \leq 4$. Sans groupage (i.e. $g = 1$) nous avons besoin de 6 longueurs d’ondes (une par paire) et un total de 12 ADMs. Avec un facteur de groupage $g = 2$, nous pouvons placer 2 paires sur une même longueur d’onde en utilisant 4 ADMs, ou seulement 3 si les 2 paires partagent un sommet. Par exemple, nous pouvons mettre les paires $(1, 2)$ et $(2, 3)$ sur une longueur d’onde, les paires $(1, 3)$ et $(3, 4)$ sur une deuxième, et les paires $(1, 4)$ et $(2, 4)$ sur une troisième, pour un total de 9 ADMs, ce qui est optimal.

Maintenant, si le facteur de groupage est $g = 3$, deux longueurs d’ondes sont suffisantes. Si nous plaçons sur une longueur d’onde les paires $(1, 2)$, $(2, 3)$, et $(3, 4)$ et sur une autre les paires $(1, 3)$, $(2, 4)$, et $(1, 4)$, nous avons besoin de 8 ADMs (solution *a*, figure 3.2(a)). Nous pouvons faire mieux en plaçant sur une longueur d’onde les paires $(1, 2)$, $(2, 3)$ et $(1, 3)$ et sur l’autre les paires $(1, 4)$, $(2, 4)$ et $(3, 4)$. Alors, nous utilisons seulement 7 ADMs (solution *b*, figure 3.2(b)).

Plus formellement, la solution *a* consiste à partitionner les arêtes de K_4 (toutes les paires) en deux chemins de 4 sommets : $[1, 2, 3, 4]$ et $[3, 1, 4, 2]$ (figure 3.2(c)), alors que la solution *b* donne une partition en un triangle $(1, 2, 3)$ et une étoile d’arêtes $(1, 4)$, $(2, 4)$, et $(3, 4)$ (figure 3.2(d)). Le nombre de sommets des graphes de la partition donne le nombre d’ADM à installer dans le réseaux.

3.1.2 Modèle

Pour des facteurs de groupage, ensemble de requêtes et topologies particuliers, il est possible de donner une construction optimale, c’est-à-dire une affectation valide des requêtes aux lon-

guez d'ondes qui minimise le nombre d'ADMs. C'est en particulier le cas pour l'échange total (une requête unitaire entre chaque paire de sommet) pour lequel des constructions optimales ont été obtenues pour des topologies simples (chemin, anneau uni et bi-directionnel) et des facteurs de groupage spécifiques.

Comme nous l'avons rappelé avec l'exemple de la figure 3.2 et prouvé dans [Ci22], sur l'anneau unidirectionnel avec un ensemble de requêtes symétriques, on peut affecter à la même longueur d'onde une paire de requêtes. Chaque paire utilise $1/g$ de la capacité de la longueur d'onde sur l'ensemble de l'anneau et 2 ADMs. L'ensemble des paires de requêtes symétriques est modélisé par le graphe non-orienté I . Un graphe $B \subseteq I$ à g arêtes correspond à une affectation valide (en terme de capacité induite) de g paires de requêtes à une longueur d'onde. Les sommets du graphe B correspondent aux ADMs à placer dans le réseau pour cette longueur d'onde.

Le problème du groupage de trafic avec un ensemble de requêtes symétriques sur l'anneau unidirectionnel se modélise alors de la façon suivante :

Problème 1 (Groupage de trafic symétrique sur l'anneau unidirectionnel)

Entrées : Un anneau unidirectionnel à N sommets, \vec{C}_N , le facteur de groupage g , et un ensemble de requêtes symétriques modélisées par un graphe I .

Sortie : Partition de I en sous-graphes B_w , $1 \leq w \leq W$, telle que $|E(B_w)| \leq g$.

Objectif : Minimiser $\sum_{w=1}^W |V(B_w)|$. L'optimal est noté $A(\vec{C}_N, I, g)$.

Comme nous le verrons dans la section 3.3.1, le problème 1 est NP-complet pour $g \geq 2$, et APX-complet (n'admet pas de schéma d'approximation polynomial de facteur d'approximation constant, sauf si $P=NP$), et il existe des algorithmes d'approximation de facteur $O(\log g)$.

Nous allons maintenant nous intéresser au cas de l'échange total, $I = K_N$. Il s'agit alors de partitionner le graphe complet en sous-graphes ayant au plus g arêtes.

3.1.3 Comment établir des bornes inférieures

Avec l'exemple de la figure 3.2 nous observons que la meilleure solution est obtenue en utilisant des graphes denses, c'est-à-dire maximisant le ratio nombre d'arêtes sur nombre de sommets. Les arbres ont une arête de moins que de sommets et donc un ratio $\frac{V(T)-1}{V(T)} < 1$, où T est un arbre, alors que les triangles (K_3) ont un ratio de 1 et plus généralement les cliques à k sommets ont un ratio de $\frac{k-1}{2}$. Nous avons donc intérêt à utiliser des graphes de grand ratio.

Notons $\rho(B)$ le ratio d'un graphe B , $\rho(B) = |E(B)|/|V(B)|$, $\rho(m)$ le ratio maximum d'un graphe à m arêtes, et $\rho_{\max}(g)$ le ratio maximum d'un graphe à au plus g arêtes, c'est-à-dire $\rho_{\max}(g) = \max\{\rho(B), |E(B)| \leq g\} = \max_{m \leq g} \rho(m)$. Un graphe de ratio $\rho_{\max}(g)$ optimise l'utilisation des ADMs. Aussi, si il existe une partition du graphe de requêtes en sous-graphes de ratio $\rho_{\max}(g)$, la solution obtenue sera optimale en nombre d'ADMs. Nous pouvons donc établir la borne inférieure suivante :

Théorème 2 ([Ci21]) $A(\vec{C}_N, K_N, g) \geq \frac{N(N-1)}{2\rho_{\max}(g)}$.

Nous avons calculé précisément la valeur de $\rho_{\max}(g)$, que nous rappelons ci-dessous, et nous donnons quelques exemples de valeurs dans la table 3.1.

Proposition 3 ([Ci21])

- Si $\frac{x(x-1)}{2} \leq g \leq \frac{(x-1)(x+1)}{2}$, alors $\rho_{\max}(g) = \frac{x-1}{2}$ et cette valeur est atteinte pour K_x .
- Si $\frac{(x-1)(x+1)}{2} < g < \frac{x(x+1)}{2}$, alors $\rho_{\max}(g) = \frac{g}{x+1}$ et cette valeur est atteinte pour tout graphe à g arêtes et $x+1$ sommets.

g	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	24	32	48	64
$\rho_{\max}(g)$	$\frac{1}{2}$	$\frac{2}{3}$	1	1	$\frac{5}{4}$	$\frac{3}{2}$	$\frac{3}{2}$	$\frac{8}{5}$	$\frac{9}{2}$	2	2	2	$\frac{13}{6}$	$\frac{14}{6}$	$\frac{5}{2}$	$\frac{5}{2}$	3	$\frac{32}{9}$	$\frac{9}{2}$	$\frac{64}{11}$

TAB. 3.1 – $\rho_{\max}(g)$ pour les petites valeurs de g .

Le problème 1 pour $I = K_N$ est très fortement relié aux problèmes combinatoires de la théorie des configurations (*design theory*) [104]. Une configuration classique est un $(N, k, 1)$ -design¹ qui est exactement une partition des arêtes de K_N en sous-graphes isomorphes à K_k (ce sont les *blocs* de la configuration). Par exemple, pour $g = 3$ on aura une solution optimale si on a un $(N, 3, 1)$ -design (auss appelé triplet de Steiner), c'est-à-dire une partition des arêtes de K_N en K_3 . Plus précisément : un $(7, 3, 1)$ -design est une partition des arêtes de K_7 en 7 K_3 , ce qui est optimal et donne $A(\vec{C}_7, K_7, 3) = 21$; un $(13, 3, 1)$ -design est une partition de K_{13} en 26 K_3 , d'où $A(\vec{C}_{13}, K_{13}, 3) = 78$; un $(13, 4, 1)$ -design est une partition de K_{13} en 13 K_4 et $A(\vec{C}_{13}, K_{13}, 6) = 52$.

La définition classique équivalente est : étant donné un ensemble de N éléments, trouver un ensemble de blocs tels que chaque bloc contienne k éléments et que chaque paire d'éléments apparaisse dans exactement un bloc (voir [104]). Plus généralement, un G -design d'ordre N (voir [105, section IV.22], [104, section IV.5], l'état de l'art [103] et aussi [101, 98]) est une partition des arêtes de K_N en sous-graphes isomorphes à un graphe donné G . L'intérêt des G -design est immédiat :

Proposition 4 ([A6]) *Si il existe un G -design d'ordre N , où G est un graphe à au plus g arêtes et ratio $\rho_{\max}(g)$, alors $A(\vec{C}_N, K_N, g) = \frac{N(N-1)}{2\rho_{\max}(g)}$.*

Pour qu'un G -design existe, il faut d'une part que le nombre d'arêtes de K_N soit un multiple du nombre d'arêtes de G , et d'autre part qu'il soit possible de répartir les arêtes adjacentes à un sommet dans différents graphes. Les conditions nécessaires d'existence d'un G -design sont donc :

Condition nécessaire 5 (existence d'un G -design) *Si il existe un G -design d'ordre N , alors il vérifie*

- (i) $\frac{N(N-1)}{2}$ est un multiple de $|E(G)|$,
- (ii) $N - 1$ est un multiple du plus grand diviseur commun des degrés des sommets de G .

Wilson a prouvé que ces conditions nécessaires sont aussi suffisantes pour N suffisamment grand [99, 102]. C'est un résultat très important de la théorie des configurations. De plus, il nous permet de dire que pour n'importe quelle valeur de g donnée et pour un nombre infini de valeurs de N , $A(\vec{C}_N, K_N, g) = \frac{N(N-1)}{2\rho_{\max}(g)}$. Ce résultat est asymptotique puisque les valeurs de N qui vérifient le théorème de Wilson sont très grandes. Toutefois, pour les petites valeurs de g , nous pouvons utiliser des résultats exacts de la théorie des configurations. Par exemple, de l'existence de G -designs pour $G = K_3$, $G = K_4$, $G = K_5$, etc, nous déduisons :

Théorème 6 ([Ci21])

- $A(\vec{C}_N, K_N, 3) = \frac{N(N-1)}{2}$ pour $N \equiv 1$ ou $3 \pmod{6}$;
- $A(\vec{C}_N, K_N, 4) = \frac{N(N-1)}{2}$ pour $N \equiv 0$ ou $1 \pmod{8}$;

¹Plus généralement, un (N, k, λ) -design est une partition des arêtes du multigraphe λK_N (chaque arête de K_N est répétée λ fois) en sous-graphes isomorphes à K_k . Voir [104] pour la liste de tous les (N, k, λ) -design connus.

- $A(\vec{C}_N, K_N, 5) = \frac{2N(N-1)}{5}$ pour $N \equiv 0$ ou $1 \pmod{5}$;
- $A(\vec{C}_N, K_N, 6) = A(\vec{C}_N, K_N, 7) = \frac{N(N-1)}{3}$ pour $N \equiv 1$ ou $4 \pmod{12}$;
- $A(\vec{C}_N, K_N, 8) = \frac{5N(N-1)}{16}$ pour $N \equiv 0$ ou $1 \pmod{16}$, et $N \neq 48$;
- $A(\vec{C}_N, K_N, 9) = \frac{5N(N-1)}{18}$ pour $N \equiv 0$ ou $1 \pmod{9}$, et $N \geq 235$;
- $A(\vec{C}_N, K_N, 10) = \frac{N(N-1)}{4}$ pour $N \equiv 1$ ou $5 \pmod{20}$;
- $A(\vec{C}_N, K_N, 15) = A(\vec{C}_N, K_N, 16) = \frac{N(N-1)}{5}$ pour $N \equiv 1$ ou $6 \pmod{15}$, et $N \geq 802$.

Pour certaines valeurs de g , par exemple 7, les partitions optimales sont obtenues en utilisant des graphes à moins de g arêtes, ici 6. Ceci s'explique simplement : tout graphe à 7 arêtes a au moins 5 sommets et donc ratio $\frac{7}{5}$. Une partition utilisant de tels graphes aura au moins $\frac{5N(N-1)}{14}$ ADMs. En remarquant que K_4 a 6 arêtes, 4 sommets et donc ratio $\frac{3}{2} = \rho_{\max}(7)$, nous obtenons une partition utilisant $\frac{N(N-1)}{3}$ ADMs, ce qui est bien optimal.

Nous avons utilisé cette remarque dans [Ci21] pour donner un contre-exemple à une conjecture de [119] disant : « le nombre minimum d'ADMs pour un trafic d'échange total sur l'anneau unidirectionnel est obtenu en utilisant le nombre minimum de longueurs d'ondes ». En effet, utiliser le nombre minimum de longueurs d'ondes impose d'affecter 7 paires de requêtes à chacune, donc de partitionner K_N en sous-graphes à 7 arêtes, ce qui n'est pas optimal lorsque $N \equiv 1$ ou $4 \pmod{12}$.

Le théorème 6 donne également un contre-exemple à une conjecture de [128] disant que $A(\vec{C}_N, K_N, 16) \geq \left\lceil \frac{7N(N-1)}{32} \right\rceil$. En effet, pour une infinité de valeurs de N , nous avons $A(\vec{C}_N, K_N, 16) = \frac{N(N-1)}{5}$, ce qui est inférieur.

La non-existence de G -design pour certaines valeurs de g et de N implique que K_N ne peut pas être partitionné optimalement par des graphes isomorphes. Ce manque de régularité complique la recherche de partitions optimales et donc la détermination de $A(\vec{C}_N, K_N, g)$. De plus, les solutions sont très différentes selon les valeurs de g et N , et les conditions 5 indiquent que les solutions dépendent de classes de congruences de N .

Par exemple, pour $g = 3$ nous avons $\rho_{\max}(3) = 1$ ce qui correspond à un K_3 . Cependant, les K_3 -designs n'existent que pour $N \equiv 1$ ou $3 \pmod{6}$. Pour les autres valeurs de N nous devons utiliser d'autres familles de graphes dans la partition. Comme K_3 est le seul graphe avec au plus 3 arêtes de ratio 1, les autres graphes qui peuvent être utilisés sont : un chemin à 4 sommets, P_4 , ou une étoile à 3 branches, $K_{1,3}$, tous les deux de ratio $\frac{3}{4}$, un chemin à 3 sommets, P_3 , de ratio $\frac{3}{2}$, ou encore une arête de ratio $\frac{1}{2}$. C'est ce qui a été fait dans [131], que nous rappelons dans le théorème 7, où les partitions optimales sont obtenues en utilisant un grand nombre de K_3 , et quelques P_3 , P_4 , $K_{1,3}$ et simples arêtes.

Théorème 7 (Théorème 4 de [131]) *Soit $N \geq 2$. Il existe une partition de K_N utilisant*

- Pour $N \equiv 1$ ou $3 \pmod{6}$: $\frac{N(N-1)}{6} K_3$;
- Pour $N \equiv 5 \pmod{6}$: $\frac{N(N-1)-8}{6} K_3$ et $2 P_3$;
- Pour $N \equiv 0$ ou $4 \pmod{12}$: $\frac{N(N-1)}{6} - \frac{N}{4} K_3$ et $\frac{N}{4} K_{1,3}$;
- Pour $N \equiv 2 \pmod{6}$: $\frac{N(N-1)-2}{6} - \left\lceil \frac{N-2}{4} \right\rceil K_3$, $\left\lceil \frac{N-2}{4} \right\rceil K_{1,3}$ et une arête ;
- Pour $N \equiv 6$ ou $10 \pmod{12}$: $\frac{N(N-1)}{6} - \frac{N+2}{4} K_3$, $\frac{N-2}{4} K_{1,3}$ et $1 P_4$,

où P_3 est un chemin à 3 sommets, P_4 un chemin à 4 sommets et $K_{1,3}$ une étoile à trois branches.

Lorsque plusieurs graphes doivent être utilisés dans la partition, la borne inférieure donnée par $\rho_{\max}(g)$ n'est plus suffisante. Pour affiner la valeur de la borne inférieure pour un g et un N donnés, nous utilisons le système d'équations suivant, où $\alpha_{i,j} \geq 0$ est le nombre de sous-graphes de K_N à i arêtes et j sommets :

$$R = \frac{N(N-1)}{2} = \sum_{i,j} i \cdot \alpha_{i,j} \quad (3.1)$$

$$A = \sum_{i,j} j \cdot \alpha_{i,j} \quad (3.2)$$

L'équation 3.1 signifie que toutes les arêtes sont couvertes une fois et l'équation 3.2 compte le nombre de sommets utilisés. Nous en déduisons l'équation 3.3.

$$\rho_{\max}(g) \cdot A = R + \sum_{i,j} (j \cdot \rho_{\max}(g) - i) \cdot \alpha_{i,j} \quad (3.3)$$

L'équation 3.3 nous confirme que si tous les graphes utilisés dans la partition sont de ratio $\rho_{\max}(g)$, alors $A = \frac{R}{\rho_{\max}(g)}$, c'est-à-dire la borne inférieure du théorème 2, mais cela ne suppose pas l'existence d'un G -design. En effet, la borne inférieure peut-être atteinte en utilisant une combinaison de graphes différents. Par exemple, pour $g = 4$ nous avons $\rho_{\max}(4) = 1$ et ce ratio est atteint à la fois pour K_3 , C_4 , et le graphe $K_3 + e$ (une arête accrochée à un unique sommet de K_3). Dans ce cas, nous cherchons des partitions en des graphes isomorphes à ceux-là. Un autre exemple est le cas où $g = 12$ et $N \equiv 1 \pmod{4}$. La borne inférieure est alors atteinte en utilisant une combinaison de K_5 et de $K_{2,2,2}$ (graphe triparti complet entre des groupes de 2 sommets), ces deux graphes ayant ratio 2, comme nous l'avons montré dans [Ci22].

Si des graphes de la partition n'ont pas le ratio $\rho_{\max}(g)$, alors $A > \frac{R}{\rho_{\max}(g)}$. Pour déterminer le nombre de tels graphes et ainsi affiner la borne inférieure, il est nécessaire d'examiner, pour un g donné, toutes les classes de congruences de N et les exceptions possibles (petites valeurs de N). Il faut de plus prendre en considération tous les graphes utilisables dans la partition, leur nombre d'arêtes, de sommets, et le degré de chacun de leurs sommets. Pour $g = 3$, il y a 5 graphes possibles (K_3 , P_4 , P_3 , $K_{1,3}$ et l'arête), mais pour $g = 6$ il y en a déjà 38. Ce travail étant long et technique, je ne le présenterai pas plus en détail ici, et je recommande la lecture de la preuve pour $g = 6$ [A6].

3.1.4 Résultats connus

Une fois les bornes inférieures établies, il reste à trouver des constructions (partitions) les atteignant, ou à montrer que cela n'est pas possible (ce qui est généralement le cas pour quelques valeurs de N , des exceptions). Là encore, je recommande la lecture de l'article [A6] où nous rappelons l'ensemble des définitions nécessaires. Je rappelle cependant la principale ci-dessous.

Définition 8 (Group Divisible Design, GDD) *Un group divisible design (GDD) est un triplet $(X, \mathcal{G}, \mathcal{B})$, où X est un ensemble de points, \mathcal{G} une partition de X en groupes, et \mathcal{B} une collection de sous-ensembles de X appelés blocs tel que toute paire de sommets distincts de X apparaisse soit dans un groupe soit dans exactement un bloc, mais pas dans les deux. Un k -GDD de type $g_1^{u_1} g_2^{u_2} \dots g_s^{u_s}$ est un GDD dans lequel tous les blocs ont tailles k et où il y a u_i groupes de tailles g_i , $i = 1, 2, \dots, s$.*

Cette définition n'est pas vraiment accessible au néophyte, il faut bien l'admettre. Cependant, il suffit de comprendre que le type $g_1^{u_1} g_2^{u_2} \dots g_s^{u_s}$ représente pour nous un graphe multi-parti complet avec u_i parties de tailles g_i , $i = 1, 2, \dots, s$. Ensuite, un k -GDD de ce type n'est autre qu'une partition des arêtes de ce graphe multi-parti en K_k . Par exemple, un 3-GDD de type $2^t 4^1$ est une partition du multi-parti à t parties de taille 2 et une partie de taille 4, $K_{2 \times t, 4}$,

en $\frac{2t(t-1)+8t}{3} K_3$. L'ensemble des k -GDD connus est disponible dans [105, section III.1.3, pages 189-190] et dans l'état de l'art [103].

A ce jour, les constructions optimales connues pour notre problème de groupage de trafic sur l'anneau unidirectionnel avec pour ensemble de requêtes l'échange total sont :

- $g = 3$ [131] ;
- $g = 4$ [128] avec une preuve plus simple dans [Ci22] ;
- $g = 5$ [134] ;
- $g = 6$ [A6] ;
- $g = 7$ [154] ;
- $g = 8$ [159] ;
- $g = 9$ [158] ;
- $g = 12$ pour $N \equiv 1 \pmod{4}$ [Ci22] ;
- $g \geq N(N-1)/6$ [Ci22]. Partition gloutonne de K_N en au plus 3 graphes.

Des valeurs au delà de $g = 10$ sont actuellement en cours d'étude. Par ailleurs, les solutions pour les cas pratiques des anneaux SONET/SDH, c'est-à-dire pour $N \leq 16$ et $g = 3, 4, 12, 16, 48, 64$, ont été données dans [Ci21].

Remarque sur la complexité du problème : dans le cas de l'échange total, $I = K_N$, la complexité du problème de groupage n'est pas connue. Cependant, les liens avec la théorie des configurations nous montrent bien que c'est un problème combinatoire difficile. Pour un facteur de groupage g fixé, il est possible de concevoir un algorithme polynomial donnant la construction optimale pour toute valeur de N (conséquence du théorème de Wilson [99]). C'est ce qui a déjà été fait pour $g \leq 9$. Mais, le travail à fournir pour une valeur donnée peut-être très important, ce qui a par exemple été le cas pour $g = 6$.

La question est maintenant de savoir si l'on peut utiliser les idées et les techniques génériques utilisées pour les cas déjà résolus afin de concevoir un algorithme pour toute valeur de g .

Ne sachant pas répondre à cette question, nous avons proposé des algorithmes d'approximation basé sur des partitions en graphes multi-partis complets. Ces algorithmes ont un facteur d'approximation $\sqrt{2}$ en utilisant des bipartis (entre 2 groupes de taille \sqrt{g}), $\sqrt{3/2}$ avec des tripartis, $\sqrt{4/3}$ avec des quadripartis, et plus généralement $\sqrt{1+1/k}$ avec des k -partis [A10]. Nous donnons également un schéma d'approximation lorsque N est suffisamment grand de facteur $1 + 4g/N + o(1/N)$.

3.2 Chemin orienté

Intéressons nous maintenant au cas, à priori simple, où la topologie physique est un chemin orienté. Toutes les requêtes sont orientées dans le même sens, de la gauche vers la droite.

Comme dans le cas de l'anneau unidirectionnel avec un ensemble quelconque de requêtes, modélisé par un graphe orienté \vec{I} , la contrainte induite par le facteur de groupage g est que la charge induite par un sous-graphe soit inférieure à g . En d'autres termes, soit $B \subseteq \vec{I}$ un sous-graphe et $(i, i+1)$ un arc quelconque de \vec{P}_N , alors au plus g arcs de B correspondent à des requêtes dont le routage dans \vec{P}_N empruntent l'arc $(i, i+1)$. Nous devons donc satisfaire la contrainte $\text{charge}(\vec{P}_N, B) \leq g$.

Le problème général du groupage de trafic sur le chemin s'exprime alors de la façon suivante :

Problème 9 (Groupage de trafic sur le chemin unidirectionnel)

Entrées : Un chemin unidirectionnel à N sommets, \vec{P}_N , le facteur de groupage g , et un ensemble de requêtes modélisées par un graphe orienté \vec{I} .

Sortie : Partition des arcs de \vec{I} en sous-graphes B_w , $1 \leq w \leq W$, telle que $\text{charge}(\vec{P}_N, B_w) \leq g$.

Objectif : Minimiser $\sum_{w=1}^W |V(B_w)|$. L'optimal est noté $A(\vec{P}_N, \vec{I}, g)$.

Dans le cas où $g = 1$, le problème est trivial et se résout avec un algorithme glouton. Il suffit d'associer, au niveau de chaque nœud, les requêtes terminant en ce nœud avec les requêtes issues de ce nœud. Le nombre d'ADMs à installer en ce nœud est le maximum entre le nombre de requêtes y terminant et celles qui en sont issues. Nous avons alors :

Proposition 10 ([A7, Ci25]) $A(\vec{P}_N, \vec{I}, 1) = \sum_{i=1}^N \max \left\{ d_{\vec{I}}^-(i), d_{\vec{I}}^+(i) \right\}$.

Dès que $g \geq 2$ et pour un ensemble quelconque de requêtes, le problème a été montré NP-complet et difficile à approcher [148]. Les résultats sur l'approximabilité du groupage de trafic obtenus dans le cas de l'anneau unidirectionnel s'appliquent (voir section 3.3), et nous avons un algorithme d'approximation de facteur $2 \ln g + o(\ln g)$ [138]. A ce jour, des constructions optimales ont été obtenues uniquement pour le cas où \vec{I} est l'ensemble complet des requêtes avec $g = 2$.

3.2.1 Ensemble complet de requêtes

Un tournoi transitif à N sommets, noté TT_N , est le graphe orienté dont les arcs sont : $\{(i, j) \mid 1 \leq i < j \leq N\}$. Il contient donc tous les arcs possibles orientés de la gauche vers la droite. Dans la suite, nous notons $\{a, b, c\}$ le TT_3 dont les arcs sont (a, b) , (b, c) et (a, c) .

En corollaire de la proposition 10 et pour l'ensemble de requêtes $I = TT_N$ (ensemble complet de requêtes), nous avons :

Corollaire 11 ([A7, Ci25]) $A(\vec{P}_N, TT_N, 1) = \frac{3N^2 - 2N - \varepsilon}{4}$, où $\varepsilon = 1$ si N est impair et 0 sinon.

Pour un facteur de groupage $g = 2$, le graphe TT_3 est un bon candidat comme brique de base de la partition : il a une charge induite de 2 et un ratio 1. Mais il n'est pas le seul graphe dans ce cas. Par exemple, le graphe dont les arcs sont $\{(u_1, u_2), (u_2, u_3), \dots, (u_{p-1}, u_p), (u_1, u_p)\}$, avec $1 \leq u_1 < u_2 < \dots < u_p$, a également un ratio 1 et induit une charge de 2. Le graphe construit à partir du $TT_3 \{a, b, c\}$ en ajoutant l'arc (c, d) , avec $c < d$, a lui aussi un ratio de 1.

Mais le ratio 1 n'est pas optimal sur le chemin avec $g = 2$. En effet, un même sous-graphe peut-être constitué de 2 TT_3 partageant un sommet. Le ratio sera alors de $\frac{6}{5}$. Plus généralement, si un graphe est constitué de l'union de k TT_3 , alors il aura ratio $\frac{3k}{2k+1}$. Nous avons étudié très précisément le nombre maximum d'arcs d'un sous-graphe de la partition pour un g donné dans [Ci26], ce dont nous parlerons dans la section 3.2.2.

Dans la figure 3.3, nous présentons plusieurs constructions optimales. En particulier, pour $N = 7$ on a $A(\vec{P}_7, TT_7, 2) = 20$ et la partition comprend 6 graphes : les 5 $TT_3 \{2, 3, 5\}$, $\{3, 4, 6\}$, $\{1, 5, 6\}$, $\{2, 6, 7\}$ et $\{1, 3, 7\}$, plus l'union de 2 $TT_3 \{1, 2, 4\} + \{4, 5, 7\}$.

Il est important de comprendre que c'est le nombre total d'unions de TT_3 qui est important. Dans l'exemple de la figure 3.3 pour TT_9 , le $TT_3 \{7, 8, 9\}$ pouvait être associé à n'importe quel $TT_3 \{x, y, 7\}$ sans changer l'optimalité de la solution.

Dans [A7, Ci25], nous avons montré comment déterminer le nombre d'unions nécessaires pour obtenir une partition optimale pour $g = 2$. Puis, nous avons montré comment construire des partitions optimales en utilisant des 3-GDD de type $u^1 v^1 1^u$ pour $N = 2u + v$, N impair, et de type $(2u)^1 (2v)^1 2^u$ pour $N = 4u + 2v$. La construction est récursive et utilise les partitions optimales pour u et v (resp. $2u$ et $2v$). L'argument principal est que chacun des sous-graphes de la partition pour u sera fusionné avec un des TT_3 du 3-GDD de type $u^1 v^1 1^u$ ou $(2u)^1 (2v)^1 2^u$, et que ce nombre de fusions est suffisant.

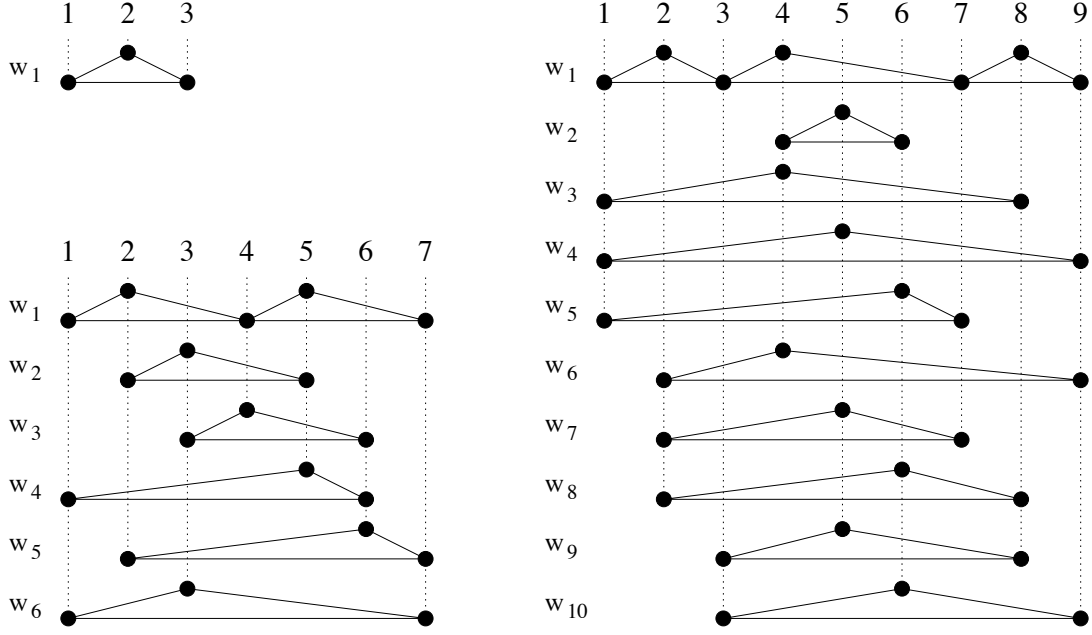


FIG. 3.3 – Constructions optimales pour TT_3 , TT_7 et TT_9 avec $g = 2$ utilisant respectivement 3, 20 et 34 ADMs.

Dans l'exemple de la figure 3.3, la partition de TT_7 utilise un 3-GDD de type $3^1 1^1 1^3$ et celle de TT_9 un 3-GDD de type $3^1 3^1 1^3$.

Pour aller plus loin, il faut utiliser d'autres types de graphes dans les partitions. Par exemple avec $g = 3$, il faut utiliser des $K_4 - e$ pour respecter la contrainte de capacité. Les lemmes 12 et 13 donnent les constructions optimales pour $N = 9$ et $N = 10$. Il faut également utiliser des G -designs plus complexes.

Nous travaillons à la preuve pour $g = 3$ qui semble raisonnable. Par contre, les autres valeurs de g semblent très difficiles à atteindre au vu des outils actuellement disponibles.

Lemme 12 $A(\vec{P}_9, TT_9, 3) = 29$.

Preuve. La construction utilise une partition de K_9 en 6 $K_4 - e$ et 2 K_3 donnée dans [134]. Les graphes de la partition sont :

- Le $K_3 + K_4 - e$: $\{1, 2, 3\} + \{3, 4, 7, 8\} - \{4, 8\}$
- Le K_3 : $\{0, 4, 8\}$
- Les 5 $K_4 - e$:

$\{0, 1, 5, 7\} - \{0, 5\}$	$\{0, 3, 5, 6\} - \{0, 6\}$	$\{0, 2, 6, 7\} - \{0, 7\}$
$\{1, 4, 6, 8\} - \{4, 8\}$	$\{2, 4, 5, 8\} - \{4, 8\}$	

Chaque sous-graphe respecte la contrainte de charge, et au total la construction utilise 29 ADMs, ce qui est optimal. \square

Lemme 13 $A(\vec{P}_{10}, TT_{10}, 3) = 36$.

Preuve. La décomposition est obtenue à partir d'une partition de K_{10} en 9 $K_4 - e$ donnée dans [100] (à une renumérotation des sommets prêt).

- | | | |
|-----------------------------|-----------------------------|-----------------------------|
| $\{1, 3, 5, 8\} - \{1, 8\}$ | $\{1, 4, 6, 8\} - \{1, 8\}$ | $\{2, 4, 5, 7\} - \{2, 7\}$ |
| $\{2, 3, 6, 7\} - \{2, 7\}$ | $\{1, 2, 7, 8\} - \{1, 8\}$ | $\{0, 1, 8, 9\} - \{0, 9\}$ |
| $\{0, 3, 4, 9\} - \{0, 9\}$ | $\{0, 5, 6, 9\} - \{0, 9\}$ | $\{0, 2, 7, 9\} - \{2, 7\}$ |

La construction utilise 36 ADMs, ce qui est la borne inférieure. \square

3.2.2 Ratio maximum

Pour établir des bornes inférieures sur le nombre d'ADM dans le cas où la topologie physique est un chemin orienté et pour un facteur de groupage g donné, nous nous sommes intéressés au nombre maximum de requêtes que peut contenir un sous-graphe de la partition, et donc au ratio maximum. Plus précisément, nous avons étudié le problème suivant, connu sous le nom de *maximum path packing* :

Problème 14 (Maximum path packing)

Entrées : Un chemin orienté à N sommets, \vec{P}_N , et le facteur de groupage g .

Sorties : Un graphe orienté D tel que $\text{charge}(\vec{P}_N, D) \leq g$.

Objectif : Maximiser le nombre d'arcs de D .

En d'autres termes, nous supposons que chaque nœud est équipé d'un ADM et nous cherchons le plus grand nombre de requêtes que peut contenir un sous-graphe de la partition. Par exemple, lorsque $g = 1$, il suffit de prendre toutes les requêtes de longueur 1 et D_1 est le graphe orienté à N sommets et les $N - 1$ arcs $\{(i, i + 1) \mid 1 \leq i < N\}$. Lorsque $g = 2$, le graphe contient toutes les requêtes de longueur 1 et quelques requêtes de longueur 2. Les arcs de D_2 sont donc $\{(i, i + 1) \mid 1 \leq i < N\} \cup \{(2i - 1, 2i + 1) \mid 1 \leq i < \lfloor N/2 \rfloor\}$. Lorsque $g = 3$, D_3 contient toutes les requêtes de longueurs 1 et 2.

Posons $g_s = \frac{s(s+1)}{2}$ et soit $T_s(N)$ l'ensemble de toutes les requêtes de longueurs au plus s . On a $|T_s(N)| = sN - g_s$. La conjecture disant que « pour $g = g_s$, le graphe D_g contient toutes les requêtes de longueurs inférieures à s » est fautive. En effet, pour $N = 11$ et $g = 10$ nous avons $s = 4$ et $T_4(11) = 34$. Or, il existe une meilleure solution obtenue en supprimant de la précédente la requête (4, 8) de longueur 4 et en ajoutant les requêtes (1, 6) et (6, 11) de longueur 5, et le graphe obtenu a 35 arcs. De la même façon avec $s = 6$, $g_6 = 21$ et $N = 16$, nous avons $T_6(16) = 75$ mais en supprimant les requêtes de longueur 6 (5, 11) et (6, 12) et en ajoutant les 4 requêtes de longueur 7 (1, 8), (8, 15), (2, 9) et (9, 16) nous obtenons un graphe à 77 arcs.

Nous avons étudié précisément le problème 14 dans [Ci26]. Nous avons en particulier établi le théorème suivant. Nous avons également fourni un algorithme glouton fournissant l'ensemble optimal d'arcs (requêtes).

Théorème 15 ([Ci26]) *Etant donnés N et g , posons : $g = g_s - d$ avec $0 \leq d \leq s - 1$; $N = qs + r$ avec $0 \leq r \leq s - 1$; $r = aq + \alpha$ avec $0 \leq \alpha \leq a - 1$; et $s - r = b(q + 1) + \beta$ avec $0 \leq \beta \leq b - 1$. Alors, le nombre maximum d'arcs d'un graphe D tel que $\text{charge}(\vec{P}_N, D) \leq g$ est :*

$$T(g, N) = T_s - dq + \min(A_s + d, B_s) \quad (3.4)$$

et le ratio maximum d'un graphe est :

$$\rho_{\max}(\vec{P}_N, g) = \frac{T(g, N)}{N} \quad (3.5)$$

avec $T_s = sN - g_s$, $g_s = \frac{s(s+1)}{2}$, $A_s = ar - \frac{a(a+1)}{2}q$ et $B_s = (b + 1)(s - r) - \frac{b(b+1)}{2}(q + 1)$.

Dans le problème 14, nous avons supposé que tous les arcs de TT_N étaient disponibles. Pour aller plus loin, il faudrait chercher le plus grand sous-graphe de charge induite au plus g dans un graphe donné. Ceci est une généralisation du problème du sous-graphe k -dense motivé par le groupage de trafic sur l'anneau unidirectionnel (voir section 3.3.1).

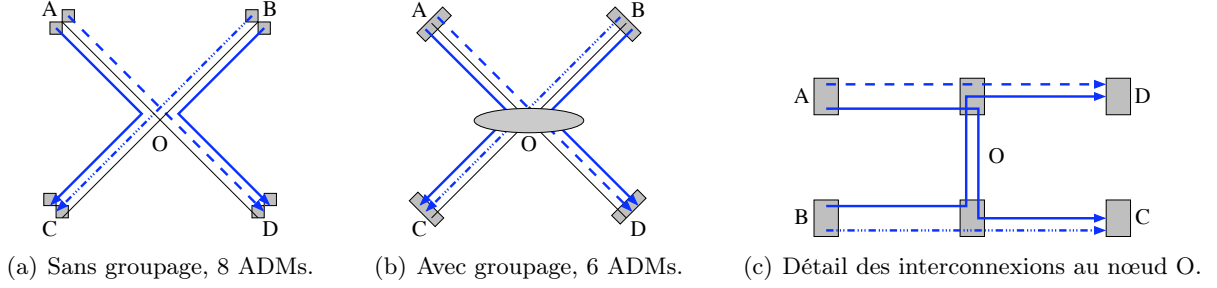


FIG. 3.4 – Groupage de trafic sur une étoile avec $g = 2$.

3.3 Trafic général

3.3.1 Complexité et (in)approximabilité

Dans le cas général d'un ensemble quelconque de requêtes, modélisé par un graphe orienté \vec{I} , sur le graphe G modélisant le réseau, la contrainte induite par le facteur de groupage g est que la charge induite par un sous-graphe soit inférieure à g . En d'autres termes, soit $B \subseteq \vec{I}$ un sous-graphe et $(i, i + 1)$ un arc quelconque de G , alors au plus g arcs de B correspondent à des requêtes dont le routage dans G empruntent l'arc $(i, i + 1)$. Nous devons donc satisfaire la contrainte $\text{charge}(G, B) \leq g$.

Comme le routage n'est à priori pas unique dans le graphe G , il faut résoudre conjointement le routage et le groupage de trafic. Ce problème, noté GRWA (*grooming, routing and wavelength assignment*), se formalise de la façon suivante :

Problème 16 (Groupage de trafic)

Entrées : Un graphe G à N sommets, le facteur de groupage g , un ensemble W de longueurs d'onde, et un ensemble de requêtes modélisées par un graphe orienté \vec{I} .

Sorties : Un routage R dans G pour les requêtes de \vec{I} et une partition des arcs de \vec{I} en sous-graphes B_w , $1 \leq w \leq W$, telle que $\text{charge}(G, R, B_w) \leq g$.

Objectif : Minimiser $\sum_{w=1}^W |V(B_w)|$. L'optimal est noté $A(G, \vec{I}, g)$.

Ce problème est très difficile à résoudre et je renvoie à [151, 165, 164] pour plus de détails. Notons que ce problème est souvent abordé en séparant le problème du routage et le problème du groupage (l'affectation de longueurs d'onde étant incluse dans le groupage). Aussi, dans la suite, nous supposons que le routage des requêtes dans le réseau est une entrée du problème et nous considérons le problème suivant :

Problème 17 (Groupage de trafic à routage fixé)

Entrées : Un graphe G à N sommets, le facteur de groupage g , un ensemble de requêtes modélisées par un graphe orienté \vec{I} , et un routage R dans G pour les requêtes de \vec{I} .

Sortie : Partition des arcs de \vec{I} en sous-graphes B_w , $1 \leq w \leq W$, telle que $\text{charge}(G, R, B_w) \leq g$.

Objectif : Minimiser $\sum_{w=1}^W |V(B_w)|$. L'optimal est noté $A(G, \vec{I}, g)$.

L'exemple de la figure 3.4 illustre une des difficultés du problème du groupage de trafic sur une topologie générale. En effet, il peut-être intéressant de placer des ADMs dans des nœuds du réseaux qui ne sont ni source ni destination de trafic. Dans cet exemple, sans groupage il faut 8 ADMs, alors qu'avec le groupage il suffit de 6 ADMs. La notion correspondante en terme

de groupage de trafic est la *bifurcation*, c'est-à-dire le fait d'autoriser une requête à changer de longueur d'onde en cours de route. Cette notion a été considérée dans [141] sur les chemins, étoiles et arbres, en autorisant les bifurcations uniquement dans les nœuds qui sont sources ou destinations de trafic. A ma connaissance, aucune étude n'a étudié le problème du groupage de trafic avec bifurcation en autorisant les bifurcations sur toutes les longueurs d'ondes et à tous les nœuds du réseau. Dans la suite, je considère uniquement le problème du groupage de trafic sans bifurcation.

Le problème 17 a été montré NP-complet sur l'anneau unidirectionnel dans [119] en utilisant une réduction du problème *bin packing*. D'autres preuves ont été proposées dans [122, 132]. Il a ensuite été montré NP-complet *au sens fort* dans [148] lorsque que le facteur de groupage $g \geq 2$ fait partie de l'entrée avec un nombre de longueurs d'ondes non bornée, ainsi qu'avec un facteur de groupage $g \geq 2$ fixé et un nombre borné de longueurs d'ondes. Ce problème est également NP-complet dans le cas où $g = 1$ (voir section 3.3.2).

Le premier résultat d'inapproximabilité pour un facteur de groupage fixé est donné dans [144], ce qui répond positivement à une conjecture de [135]. Plus précisément, il a été montré que pour un facteur de groupage fixé $g \geq 1$ sur l'anneau unidirectionnel le problème est APX-complet. C'est-à-dire qu'il n'existe pas d'algorithme polynomial d'approximation avec facteur d'approximation constant pour ce problème, sauf si $P=NP$. Ce résultat repose sur le fait que le problème de trouver le nombre maximum de triangles arêtes disjoints (et plus généralement de cycles de longueur $2g + 1$ dans un graphe de maille $2g + 1$) est APX-complet. En particulier cela implique que le problème du groupage est également NP-complet pour un facteur de groupage fixé $g \geq 1$ et un nombre non borné de longueur d'ondes, ce qui étend le résultat de [148].

Un premier algorithme d'approximation avec facteur d'approximation de l'ordre de \sqrt{g} a été proposé sur l'anneau unidirectionnel dans [132]. Cet algorithme glouton consiste à partitionner le graphe \vec{I} en arbre de largeur au plus g (charge induite au plus g). Cet algorithme a été amélioré dans [142, 157, 163]. Toutefois, ces travaux sont loin d'être satisfaisant dans la mesure ou l'existence d'un algorithme d'approximation de facteur $O(\log g)$ sur l'anneau unidirectionnel était mentionné dans [122], mais sans preuve. Cette question a été résolue dans [138] où un algorithme d'approximation de facteur $2 \ln g + o(\ln g)$ sur l'anneau unidirectionnel est présenté. De plus, cet algorithme peut-être utilisé sur d'autres topologies. Sa complexité en temps est en $O(N^g)$. Il est donc exponentiel en g . Cependant, dans les applications pratiques telles que les boucles SONET/SDH qui sont largement utilisées dans les réseaux dorsaux optique [126], le facteur de groupage est égal à 3 ou 4, avec par exemple le groupage de 4 flux à 655Mbit/s dans une longueur d'onde à 2.5Gbit/s.

Il était également important de proposer des algorithmes d'approximations dont la complexité en temps est polynomiale en N et en g . Il a donc été montré dans [144] qu'il existe un algorithme polynomial d'approximation du problème du groupage de trafic sur l'anneau unidirectionnel avec facteur d'approximation $O(N^{1/3} \log^2 N)$ pour tout facteur de groupage $g \geq 1$ (g faisant partie de l'entrée). De plus, cet algorithme donne un facteur d'approximation en $O(\log^2 N)$ lorsque le graphe de requêtes exclut un graphe fixé comme mineur (par exemple si le graphe est planaire ou de genre borné). Ceci a été obtenu en reliant le problème du groupage au problème du sous-graphe dense à k arêtes (*dense k -subgraph*) [76] et au problème du sous-graphe à degré contraint (*degree constrained subgraph*) [87].

3.3.2 Le cas $g = 1$

Lorsque $g = 1$, c'est-à-dire lorsqu'il n'y a pas de groupage, le problème est équivalent au problème de minimiser le nombre d'équipements terminaux, LTE (*light termination equipment*)², des chemins optiques dans le réseaux. Plus précisément, pour des raisons économiques, il est plus intéressant d'installer un seul équipement comprenant un émetteur et un récepteur optique opérant sur la même longueur d'onde plutôt que deux équipements distincts dans les nœuds du réseau. Il s'agit alors de minimiser le nombre de LTE à installer dans le réseau. C'est une variante du problème de l'affectation de longueurs d'onde où l'on ne s'intéresse pas au nombre de longueurs d'onde utilisées, mais seulement aux équipements.

Ce problème a été introduit dans [118] pour l'anneau unidirectionnel et montré NP-difficile indépendamment dans [127, 121, 144]. Ces preuves sont aussi valables pour l'anneau bidirectionnel lorsque le routage est donné.

Notons qu'en relâchant la contrainte imposant que l'émetteur et le récepteur d'un LTE opèrent sur la même longueur d'onde, c'est-à-dire en autorisant l'utilisation de longueur d'onde distinctes, le problème devient trivial. Il suffit d'associer de façon gloutonne une requête entrant dans un nœud avec une requête issue de ce nœud. Ceci donne la borne inférieure du problème. En fait, cette relaxation s'affranchit simplement du problème difficile d'affectation de longueur d'onde aux requêtes qui est ici le problème classique du « *circular arc coloring* » [62].

Un premier algorithme d'approximation de facteur $3/2$ a été proposé dans [125]. Par la suite des améliorations ont permis d'obtenir un facteur $10/7 + \epsilon$ [137] puis un facteur $10/7$ [136].

Bien que réalisés dans un tout autre contexte, les travaux que nous avons effectué pour la protection par cycles [A5, Cis36, Ci18, Cn40] répondent optimalement à la question de minimiser le nombre de LTEs sur l'anneau uni et bi-directionnel dans le cas de l'échange total. L'originalité de ces travaux est que les constructions optimales sont obtenues en utilisant des partitions de K_N en C_3 et C_4 uniquement.

Pour un état de l'art des travaux qui ont été réalisés sur ce problème, que ce soit sur l'anneau ou des topologies générales, pour du trafic statique ou dynamique, ainsi que sur les algorithmes centralisés et en lignes, voir le chapitre [B13].

3.3.3 Anneau bidirectionnel

Lorsque la topologie du réseau est un anneau bidirectionnel, le routage des requêtes n'est pas unique. Pour mieux comprendre les contraintes, plaçons nous dans le cas de l'échange total.

Dans un anneau bidirectionnel avec un nombre impair N de sommets, le routage des plus courts chemins implique un routage symétrique. Mais pour un nombre pair de sommets ce n'est pas nécessairement le cas. En effet, la requête $(i, i + \frac{N}{2})$ peut être routée sur un plus court chemin dans les deux directions. Prenons par exemple $N = 4$ et $g = 2$. Si nous n'imposons pas un routage symétrique, une solution consiste en les deux sous-graphes B_{λ_1} , avec les requêtes $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 0)$, $(0, 2)$, et $(2, 0)$ routées dans un sens, et B_{λ_2} avec les requêtes $(1, 0)$, $(0, 3)$, $(3, 2)$, $(2, 1)$, $(1, 3)$, et $(3, 1)$ routées dans le sens opposé. Cette solution utilise 8 ADMs. Imposons maintenant un routage symétrique et supposons que les requêtes $(0, 2)$ and $(1, 3)$ sont routées dans le même sens. Alors nous pouvons mettre dans un même sous-graphe à 4 sommets les requêtes $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 0)$, et soit la requête $(0, 2)$ soit la requête $(1, 3)$. L'autre requête nécessitera 2 ADMs supplémentaires. Nous utilisons donc au total 12 ADMs (6 dans chaque sens).

D'autre part, imposer un routage par des plus courts chemins peut augmenter le nombre d'ADMs d'une solution optimale. Prenons par exemple $N = 3$ et $g = 3$. Avec un routage des

²De nombreux auteurs parlent d'ADMs au lieu de LTEs alors qu'il n'y a pas de multiplexage ici. Ceci entraîne parfois des confusions dans les discussions.

plus courts chemins, nous utilisons les deux sous-graphes B_{λ_1} , avec les requêtes $(0, 1)$, $(1, 2)$, $(2, 0)$, et B_{λ_2} avec les requêtes $(1, 0)$, $(2, 1)$, $(0, 2)$, pour un total de 6 ADMs. Sans contrainte sur le routage, nous pouvons n'utiliser que 3 ADMs en routant toutes les requêtes dans le même sens.

Avec ces exemples, nous comprenons l'influence du routage sur le nombre total d'ADMs à utiliser. Comme il est très difficile de résoudre simultanément le problème du routage et celui du groupage de trafic, les études qui ont été menées jusqu'à présent ont considéré le cas du routage symétrique par des plus courts chemins [Ci27, 123, 145] et [161, chapitre 3].

Notons \vec{T}_N le graphe à N sommets ordonnés, $0, 1, \dots, N-1$, et contenant les arcs $\{(i, i+k), 0 \leq i \leq N-1, 1 \leq k \leq \frac{N-1}{2}\}$. \vec{T}_N est un tournoi. De plus, lorsque N est pair nous notons \vec{D}_N le graphe à N sommets ordonnés contenant les N arcs $\{(i, i + \frac{N}{2}), 0 \leq i \leq N-1\}$. Lorsque N est impair, alors nous posons $\vec{D}_N = \emptyset$.

Avec un routage des plus courts chemins sur l'anneau bidirectionnel, nous pouvons séparer le problème du groupage en deux sous-problèmes, chacun sur un anneau unidirectionnel. Si de plus nous considérons que le routage est symétrique, alors le problème du groupage d'un trafic d'échange total sur l'anneau bidirectionnel se formalise comme un problème de groupage de trafic sur l'anneau unidirectionnel avec un ensemble particulier de requêtes. Lorsque N est impair, cet ensemble de requêtes est exactement \vec{T}_N . Par contre, lorsque N est pair, l'ensemble \vec{I} des requêtes contient en plus de \vec{T}_N la moitié des arcs de \vec{D}_N , de sorte que \vec{I} contient soit l'arc $(i, i + \frac{N}{2})$, soit l'arc $(i + \frac{N}{2}, i)$. Le choix de ces arcs est bien sur à faire. Nous avons :

Problème 18 (Groupage de trafic symétrique sur l'anneau bidirectionnel)

Entrées : *Un anneau unidirectionnel à N sommets, \vec{C}_N , le facteur de groupage g , et un ensemble de requêtes modélisé par un graphe orienté \vec{I} et contenant \vec{T}_N plus la moitié des arcs de \vec{D}_N lorsque N est pair.*

Sortie : *Partition des arcs de \vec{I} en sous-graphes B_w , $1 \leq w \leq W$, telle que $\text{charge}(G, B_w) \leq g$.*

Objectif : *Minimiser $\sum_{w=1}^W |V(B_w)|$. L'optimal est noté $A(G, \vec{I}, g)$.*

Pour ce problème, les résultats connus sont une borne inférieure valable pour tout N et pour tout g , ainsi que des constructions optimales pour $g = 1, 2, 3$ et $g = \frac{k(k+1)}{2}$, $k \geq 1$, et pour certaines classes de congruences de N . Je renvoie à la thèse de Ignasi Sau [161, chapitre 3] et à l'article [145] pour plus de détails sur ces travaux.

Le fait que si peu de résultats soient connus pour le groupage de trafic sur l'anneau bidirectionnel montre bien la difficulté du problème dès que plusieurs routages sont possibles.

3.4 Discussion

D'un point de vue application réseaux, le groupage de trafic basé sur la technologie SONET/SDH a peu d'avenir, voir pas du tout. En effet, au cours des dernières années, les avancées technologiques et protocolaires ont permis d'exploiter la bande passante des longueurs d'ondes directement depuis la couche IP. C'est par exemple le cas avec MPLS et GMPLS qui permettent le contrôle de flux dans le réseau sans contrainte de granularité (taille des flux). La couche intermédiaire SONET/SDH est donc amenée à disparaître. D'autres travaux cherchent à exploiter la bande passante directement avec Ethernet et nous voyons aujourd'hui du 10Gbit-Ethernet, ainsi qu'un effort conséquent de recherche sur 100Gbit-Ethernet, par exemple avec le projet européen 100GET.

D'un point de vue théorique, nos travaux, en parallèle de ceux de Charles J. Colbourn, ont dégagé de nouvelles problématiques en théorie des configurations qui intéressent des chercheurs

de ce domaine. Notre travail est d'ailleurs présenté dans la deuxième édition de la « bible » de la théorie des configurations [104, chapitre VI.27]. Cela a également relancé des travaux de recherche sur la problématique de la recherche de sous-graphes denses, comme par exemple les travaux de Sau et al. [87, 88, 96, 161].

Comme nous l'avons vu dans la section 3.1.3, pour résoudre le problème du groupage sur l'anneau unidirectionnel, il faut savoir établir des bornes inférieures. En effet, les techniques que nous utilisons pour établir ces bornes nous donnent non seulement des valeurs, mais aussi des indications sur la structure d'une construction optimale (nombre de graphes avec g arêtes, séquences de degrés des graphes, ...). Toutefois, il est nécessaire d'adapter cette technique pour chaque valeur de g , ce qui est un travail long et peu intéressant. Aussi, il est nécessaire de développer des méthodes systématiques (programmes informatiques) pour établir les bornes inférieures et fournir les indications sur les structures possibles des constructions. Une première étape dans cette direction a été proposée par Colbourn et al. [153, 152] en utilisant une approche par programmation linéaire pour obtenir des bornes inférieures. Il serait intéressant de compléter ces travaux pour obtenir rapidement des bornes inférieures, et si possible de coupler ces travaux avec des méthodes pour construire automatiquement des décompositions optimales. C'est un travail ambitieux qui mélange les techniques de programmation linéaire et de la théorie des configurations.

Par ailleurs, une autre problématique intéressante est d'optimiser le nombre d'ADMs pour tout ensemble de requêtes vérifiant une propriété, ou pour du trafic en ligne (les requêtes sont routées dans l'ordre d'arrivée). Des premiers résultats dans cette direction ont été obtenus récemment. Tout d'abord, Sau et al. [156, 160] ont optimisé le nombre d'ADMs dans un anneau pour satisfaire tout ensemble de requêtes de degré borné. Ensuite, dans le cas particulier où $g = 1$, Shalom et al. [149, 162] ont proposé un algorithme de groupage pour du trafic en ligne de facteur de compétitivité $3/2$ pour l'anneau et $7/4$ pour des topologies générales, qui sont les meilleures possibles. Enfin, Crouser et al. [155] ont donné des bornes inférieures sur le nombre de sommets d'un chemin lorsque le nombre d'ADMs par sommet est donné, chaque sommet est source ou destination d'un nombre borné de requêtes, et pour un algorithme distribué glouton de routage en ligne. Nous poursuivons actuellement ces travaux avec Bermond et Peters [S54]. En particulier, nous avons obtenu des bornes supérieures valables pour n'importe quel algorithme de routage (localisé, distribué ou centralisé). Par la suite, nous comptons paramétrer ces bornes en fonction de l'algorithme choisi.

Pour conclure ce chapitre, je me dois de citer Jean-Claude Bermond lors d'une présentation à la communauté TAROT en mars 2003 :

« Ce n'est qu'un début, continuons le groupage! »

Chapitre 4

Groupe de ressources partageant un risque

Un groupe de ressources partageant un risque (*shared risk resource group*, SRRG) est un ensemble de ressources d'un réseau qui peuvent tomber en panne simultanément. Cette notion se décline en groupe de liens partageant un risque (*shared risk link group*, SRLG), et groupe de nœuds partageant un risque (*shared risk node group*, SRNG).

Un exemple de SRLG est donné dans la figure 4.1 qui représente un réseau multi-couches constitué d'un réseau physique (e.g., un réseau WDM), d'une couche intermédiaire (appelée ici réseau virtuel) où chaque arête correspond à un chemin dans le réseau physique, et un graphe de requêtes qui sont routées dans le réseau virtuel. Ici, la requête correspond à l'établissement de deux chemins disjoints de C à D . C'est le cas simple de protection 1 : 1 (un chemin de travail et un chemin de protection). L'un des chemins est routé dans le réseau virtuel par l'arête $\{C, D\}$, et l'autre par le chemin $\{C, A, B, D\}$. Aussi, les chemins sont parfaitement disjoints dans le réseau virtuel. Toutefois, les routes associées dans le réseau physique aux arêtes $\{C, D\}$ et $\{A, B\}$ du réseau virtuel empruntent le lien $\{E, B\}$ du réseau physique. Aussi, en cas de panne sur le lien $\{E, B\}$, les deux chemins seront affectés. Ces deux chemins partagent donc un même risque de panne.

Dans un réseau, un SRLG peut avoir une structure plus complexe que celle présentée dans la figure 4.1. Par exemple, deux liens disjoints du réseau physique peuvent, en pratique, être enterrés dans une même tranchée et donc partager un même risque de panne (cas classique du coup de pelleuse) [173]. Dans ce contexte, les problèmes d'optimisation combinatoire classique (chemin, arbre couvrant, coupe, chemins disjoints) doivent être redéfinis. Par exemple, nous ne cherchons pas un paire de chemins arêtes disjoints entre deux sommets du réseau, mais une paire de chemins SRLG disjoints. De même, nous sommes intéressés à trouver des chemins traversant le moins de SRLG possible, ceux-ci étant à priori les moins risqués.

La prise en compte des SRLG dans les problèmes d'optimisation a été largement considérée dans la littérature. Citons par exemple les travaux de l'IETF [169, 170, 172, 178], et quelques travaux dans les réseaux WDM [174, 183, 177, 176, 171, 188, 175, 180, 179]. De plus, il existe plusieurs dénominations pour la même notion. Tout d'abord, Pioro et al. [184, 187, 189] parlent de situation de panne pour décrire un ensemble de ressources subissant simultanément une panne, donc un SRRG, et Stefanakos [185] parle de panne de liens généralisée (*generalized link failure events*) pour décrire un SRLG. Ensuite, suivant les travaux de Jue et al. [177, 179] nous avons utilisé la terminologie de *graphe coloré* pour modéliser un réseau et ses SRLG, une couleur modélisant un SRLG. Ce modèle est en fait une autre appellation des *graphes étiquetés* qui ont été étudiés principalement pour leur propriétés combinatoires [106, 108, 109, 110, 111, 112, 113, 114, 115].

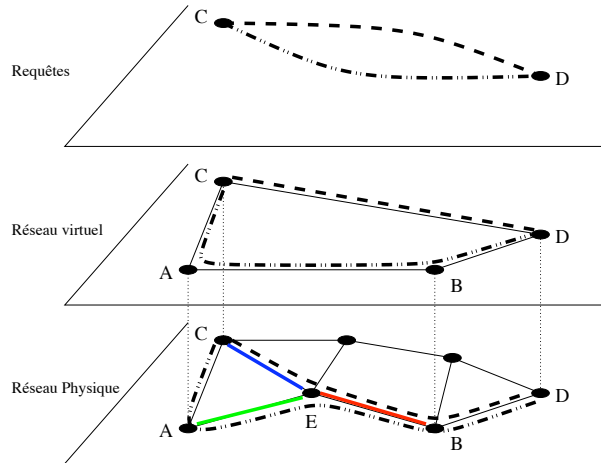


FIG. 4.1 – Exemple de SRLG

Dans ce chapitre, je présente nos principales contributions sur l'étude de la complexité et de l'(in)approximabilité des problèmes d'optimisation combinatoire classique (chemin, coupe,...) dans le contexte des SRLG [A8, S49, Ci29, 181, 182]. Comme nous le verrons, dans ce contexte, des problèmes tels que le calcul d'un plus court chemin deviennent NP-complets et difficiles à approcher. Aussi, je porte une attention particulière à l'ensemble des cas polynomiaux qui ont été identifiés à ce jour.

Je commence dans la section 4.1 par présenter la modélisation des réseaux avec SRLG par des graphes colorés. Ensuite, dans la section 4.2 je fais l'état-de-l'art des contributions pour les problèmes de plus courts chemins, coupes, chemins disjoints et arbres couvrants dans ce contexte, avant de conclure dans la section 4.3.

4.1 Modélisation

Un SRLG est usuellement modélisé par une couleur [A8, 175, 177, 179, 181, 183]. Ensuite, deux modélisations se distinguent dans la littérature pour modéliser le réseau avec les SRLG : les graphes colorés [A8, 176, 177, 181, 183] et les graphes multi-colorés [175, 179]. Dans les deux cas, les nœuds du réseau sont modélisés par les sommets du graphe et les liens du réseau par des arêtes.

Définition 19 (Graphe multi-coloré) *Un graphe multi-coloré est un triplet $G = (V, E, \mathcal{C})$ où $G = (V, E)$ est un graphe non-orienté et \mathcal{C} est une couverture (au sens de covering) de E .*

Définition 20 (Graphe coloré) *Un graphe coloré est un triplet $G = (V, E, \mathcal{P})$ où $G = (V, E)$ est un graphe non-orienté et \mathcal{P} est une partition de E .*

Dans un graphe multi-coloré, une arête peut appartenir à plusieurs couleurs, alors que dans un graphe coloré il n'y a qu'une couleur par arête. Par commodité, nous associons une couleur et l'ensemble des arêtes ayant cette couleur, et nous disons qu'une arête *appartient* à une couleur ou SRLG. Notons qu'un lien du réseau est affecté par une panne si l'un des SRLG le contenant l'est.

Remarquons que le modèle des graphes étiquetés (*labeled graphs*) [106, 108, 109, 110, 111, 112, 114, 115] correspond exactement aux graphes colorés, une étiquette correspondant à une couleur.

Il est possible de transformer un graphe coloré en graphe multi-coloré en contractant chaque paire d'arêtes adjacentes à un sommet de degré 2 en une seule arête (contraction). Celle-ci collecte alors les SRLG des arêtes contractées. Nous appelons *graphe multi-coloré minimal* un graphe multi-coloré dont tous les sommets ont degré au moins 3. Dans certains cas, il n'est pas opportun d'effectuer toutes les contractions possibles. C'est typiquement le cas si nous cherchons un chemin de s à t dans un graphe qui peut être contracté et que le degré de s ou t est 2. Nous appelons alors (s, t) -*graphe multi-coloré minimal* le graphe multi-coloré dont tous les sommets de degré 2, sauf s et t , ont été contractés.

Nous pouvons également transformer un graphe multi-coloré en graphe coloré en remplaçant chaque arête e par une chaîne de longueur égale au nombre de SRLG contenant e et en associant un SRLG distinct à chacune des arêtes de cette chaîne. Nous appelons *graphe coloré associé* un graphe coloré ainsi obtenu. Toutefois, cette transformation n'est pas unique et il existe un nombre exponentiel de graphes colorés associés à un même graphe multi-coloré. Dès lors, et comme cela a été montré par Voge [181], cette transformation n'est pas sans conséquence sur la complexité des problèmes d'optimisation.

Pour mesurer la dispersion d'une couleur dans le graphe, nous avons introduit la notion de *span* d'une couleur [A8], c'est-à-dire le nombre de composantes connexes du graphe (multi) coloré dans lequel toutes les arêtes qui n'appartiennent pas à cette couleur sont supprimées. Cette notion est au centre de l'étude de complexité des problèmes combinatoires dans les graphes (multi) colorés.

Définition 21 (Span) *Le span d'une couleur est le nombre de composantes connexes induites par les arêtes de cette couleur dans le graphe (multi) coloré.*

Voge [181] a proposé un algorithme polynomial constructif permettant de tester si il existe un graphe coloré associé à un graphe multi-coloré tel que toutes les couleurs aient span 1. Elle a de plus montré que le problème de minimiser le nombre de couleurs de span plus grand que 1 dans le graphe coloré associé est NP-difficile.

Une autre notion importante, introduite par Luo et Wang [175], est la propriété d'étoile d'une couleur.

Définition 22 (Propriété d'étoile, *star property*) *Une couleur vérifie la propriété d'étoile si toutes les arêtes appartenant à cette couleur sont adjacentes à un même sommet.*

4.2 Problèmes d'optimisation combinatoire colorés

Dans cette section, je résume nos travaux sur l'étude de la complexité et de l'(in)approximabilité des problèmes d'optimisation combinatoire classique dans le contexte des graphes (multi) colorés en montrant notre contribution vis-à-vis de l'état-de-l'art. Je ferai en particulier le point sur l'ensemble des cas polynomiaux qui ont été identifiés. J'occulte volontairement les très nombreux algorithmes heuristiques qui ont été proposés et je renvoie le lecteur à [176].

4.2.1 Chemin coloré

Dans le contexte des SRLG, nous sommes intéressés à calculer les chemins les plus fiables, ceux qui appartiennent au plus petit nombre possible de SRLG. Aussi, la mesure n'est plus le nombre d'arêtes empruntées par ce chemin, mais le nombre de SRLG auquel il appartient. Nous pouvons redéfinir ce problème d'optimisation combinatoire dans le contexte des SRLG, et donc des graphes (multi) colorés, comme suit :

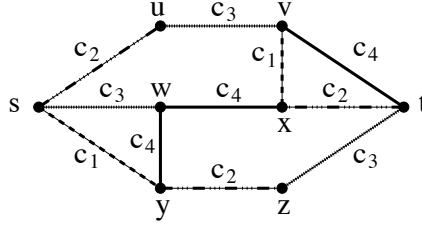


FIG. 4.2 – Un plus court chemin coloré n'est pas nécessairement constitué de plus courts chemins colorés.

Définition 23 (*(s, t) -chemin coloré, $colored (s, t)$ -path*) *Un (s, t) -chemin coloré est le plus petit ensemble de couleurs contenant un chemin de s à t dans le graphe coloré associé.*

Nous utilisons dans cette définition la notion de graphe coloré associé pour exprimer le fait que si un chemin emprunte une arête, alors il peut-être affecté par la panne de n'importe quel SRLG contenant l'arête.

Tout d'abord, remarquons qu'un plus court chemin coloré n'est pas nécessairement constitué de plus courts chemins colorés, comme c'est habituellement le cas dans les graphes. Dans l'exemple de la figure 4.2, le plus court chemin coloré de s à t est donné par les couleurs $\{c_1, c_4\}$. Cet ensemble de couleurs contient un chemin coloré de taille 2 de s à w , or le plus court chemin coloré de s à w est de taille 1 (couleur c_3). Aussi, l'algorithme de Dijkstra ne peut pas, en général, être utilisé dans ce contexte.

Le problème de déterminer le (s, t) -chemin coloré a été montré NP-difficile [176] et difficile à approcher en général à un facteur $\mathcal{O}(2^{\log^{1-\delta} |\mathcal{C}|^{\frac{1}{2}}})$ près, où \mathcal{C} est l'ensemble des couleurs et $\delta = (\log \log |\mathcal{C}|^{\frac{1}{2}})^{-\varepsilon}$ pour $\varepsilon < \frac{1}{2}$ [A8, 111].

Le premier cas polynomial a été identifié dans les graphes colorés par Datta et Somani [174, 183]. C'est le cas où toutes les arêtes d'une couleur sont incidentes à un même sommet (i.e., elles forment une étoile). Nous avons étendu ce résultat, toujours dans les graphes colorés, au cas où toutes les couleurs ont span 1 [A8]. Comme Voge [181] a donné un algorithme polynomial permettant de tester si il existe un graphe coloré associé à un graphe multi-coloré dont toutes les couleurs ont span 1, nous obtenons :

Théorème 24 *Le problème de déterminer un (s, t) -chemin coloré dans un graphe (multi) coloré est polynomial si il existe un graphe coloré associé dont toutes les couleurs ont span 1.*

Dans les graphes multi-colorés, Luo et Wang [175] ont montré que le problème du plus court chemin coloré est polynomial si le graphe vérifie la propriété d'étoile. Plus généralement, nous avons :

Théorème 25 *Le problème de déterminer un (s, t) -chemin coloré dans un graphe (multi) coloré est polynomial si son (s, t) -graphe multi-coloré minimal vérifie la propriété d'étoile.*

Les études de complexité et d'(in)approximabilité que nous avons réalisé dans [A8] sont basées sur le span maximal des couleurs. Plus précisément, nous avons montré qu'il existe un algorithme d'approximation k -approché si le span de toutes les couleurs est borné par k (voir table 4.1). Par la suite, nous avons montré que le paramètre important n'est pas le span maximal, mais le nombre de couleurs de span plus grand que 1 [S49]. Pour cela, nous avons montré que, dans les graphes colorés, le problème du plus court chemin coloré est *fixed parameter tractable*¹ (FPT) de paramètre le nombre de couleurs de span plus grand que 1.

¹Un problème est *Fixed Parameter Tractable* (FPT) lorsqu'il peut être résolu optimalement par un algorithme de complexité $\mathcal{O}(f(k)n^{O(1)})$, où k est le paramètre du problème et f est une fonction de ce paramètre [58]. Si k est borné, alors le problème se résout en temps polynomial.

Théorème 26 *Etant donné un graphe coloré $G = (V, E, \mathcal{P})$, le problème de déterminer un (s, t) -chemin coloré se résout en temps $O(2^k(m + n \log n))$, où k est le nombre de couleurs de span plus grand que 1.*

Théorème 27 *Etant donné un graphe multi-coloré $G = (V, E, \mathcal{C})$, le problème de déterminer un (s, t) -chemin coloré est dans FPT de paramètre le nombre de couleurs ne vérifiant pas la propriété d'étoile dans le (s, t) -graphe multi-coloré minimal.*

Il est intéressant de remarquer que, parmi les nombreux auteurs qui ont utilisé des algorithmes heuristiques pour calculer des plus courts chemins colorés dans la résolution de problèmes plus complexes (e.g., routage et affectation de longueur d'onde), certains ont utilisé des jeux de tests (réseaux + SRLG) où le calcul du plus court chemin se résout en temps polynomial. C'est par exemple le cas de Shen et al. [176] qui utilisent le réseau US-NET (24 nœuds et 86 liens) avec 11 SRLG dont 2 de span > 1 et un réseau italien (21 nœuds et 72 liens) avec 11 SRLG dont 3 de span > 1 .

Des résultats complémentaires ont été obtenus dans [180, Ci29] pour des variations du problème de déterminer un plus court chemin coloré avec des coûts linéaires sur les arêtes et des couleurs pondérées.

4.2.2 Coupe colorée

Une coupe entre deux sommets du réseau dans le contexte des SRLG est mesurée par le nombre de pannes qu'il faut pour déconnecter s de t . Rappelons qu'ici un SRLG suffit pour affecter une arête. Nous avons donc :

Définition 28 (*(s, t) -coupe colorée, **colored (s, t) -cut**) *Une (s, t) -coupe colorée est le plus petit ensemble de couleurs permettant de déconnecter s de t .**

Nous étendons cette définition à la notion de *multi-coupe colorée* qui est le plus petit ensemble de couleurs permettant de déconnecter s_i de t_i , pour $1 \leq i \leq l$. Nous définissons aussi la *coupe colorée minimum* comme le plus petit ensemble de couleurs déconnectant le graphe.

Nous avons montré que le problème de déterminer la (s, t) -coupe colorée est NP-difficile et difficile à approcher même lorsque que le span maximal est borné par une constante [A8] ou que toutes les couleurs de son graphe multi-coloré minimal vérifient la propriété d'étoile [S49]. Très récemment, Zhang et al. [117] ont proposé un algorithme d'approximation de facteur $O(\sqrt{m})$ pour ce problème. Par contre, le problème est polynomial lorsque toutes les couleurs ont span 1 ou que le degré coloré est borné [A8]. Il en est de même pour la multi-coupe colorée. En outre, nous avons :

Théorème 29 ([S49]) *Etant donné un graphe coloré $G = (V, E, \mathcal{P})$, le problème de déterminer une (s, t) -coupe colorée se résout en temps $O(2^k n^{O(1)})$, où k est le nombre de couleurs de span plus grand que 1.*

Le problème de la coupe colorée minimal a été montré NP-complet en général dans les graphes multi-colorés par Farago [179] et a été montré polynomial dans les graphes colorés dont toutes les couleurs ont span 1 [A8] (ce dernier résultat, bien qu'antérieur, est un corollaire du théorème 29). Tous les autres cas sont ouverts.

Par ailleurs, nous avons montré [A8] que dans les graphes colorés, le nombre maximum de (s, t) -chemins colorés disjoints est différent de la (s, t) -coupe colorée. En d'autres termes, la relation fondamentale "flot maximal = coupe minimale" n'est pas valable dans les graphes colorés.

4.2.3 Chemins colorés disjoints

A partir de la définition d'un (s, t) -chemin coloré, nous dérivons naturellement la notion de (s, t) -chemins couleurs disjoints. Le problème de décider si il existe deux (s, t) -chemins couleurs disjoints a été montré NP-complet par Hu [171]. Plus tard, une autre preuve a été proposée par Jue et al. [177]. Nous avons montré que ce problème de décision reste NP-complet lorsque le span des couleurs est borné par une constante [A8] ou qu'au moins une couleur a span plus grand que 1 [S49].

Toutefois, ce problème est polynomial si toutes les couleurs vérifient la propriété d'étoile dans le graphe multi-coloré minimal [175] ou qu'il existe un graphe coloré associé pour lequel toutes les couleurs ont span 1 [A8].

D'autre part, Jiang et al. [220] ont montré, pour une application en cryptographie, que le problème de déterminer le nombre maximum de (s, t) -chemins colorés disjoints est NP-difficile.

Comme il n'est pas toujours possible de trouver deux chemins couleurs disjoints, la notion de *chemins colorés les moins couplés* (*least coupled colored path*), c'est-à-dire des chemins qui partagent le moins de couleurs possibles, a été introduite. Hu [171] a montré que le problème de déterminer deux (s, t) -chemins colorés les moins couplés est NP-complet. Ce problème a également été montré NP-complet ultérieurement par Jue et al. [177]. Nous avons montré que ce problème reste difficile lorsque le span est borné par une constante [A8]. Ce problème est bien sur polynomial lorsque toutes les couleurs ont span 1 ou vérifient la propriété d'étoile.

4.2.4 Arbre couvrant

Transposons enfin la notion d'arbre couvrant dans les graphes colorés. Il est clair que la structure correspondante n'est pas un arbre en tant que tel, mais un ensemble de couleurs qui induit un graphe connexe atteignant tous les sommets du graphe coloré. Lorsque le graphe est multi-coloré, la structure doit rester connexe en cas de suppression des arêtes appartenant à une couleur qui n'est pas dans cet arbre. Aussi, nous avons :

Définition 30 (Arbre couvrant coloré, *colored spanning tree*) Soient $G = (V, E, \mathcal{C})$ un graphe (multi) coloré et $H = (V_H, E_H, \mathcal{P})$ son graphe coloré associé, avec $V \subseteq V_H$. Un arbre couvrant coloré est un ensemble de couleurs qui connecte tous les sommets de V dans le graphe H .

Le problème de déterminer un arbre couvrant coloré minimum a été particulièrement étudié dans les graphes étiquetés sous le nom de *minimum labeling spanning tree problem*. Ce problème a été montré NP-difficile et difficile à approcher en général, même pour des couleurs de span 1 [106, 107, 108]. Ce problème est donc considérablement plus difficile que dans les graphes classiques. De nombreux algorithmes d'approximations ont récemment été proposés [109, 111, 115, 116].

4.3 Conclusion

Dans ce chapitre, j'ai fait un état-de-l'art de la complexité et de l'(in)approximabilité des problèmes d'optimisation classique (chemin, coupe, arbre couvrant) dans le contexte des SRLG. Les résultats principaux sont rassemblés dans la table 4.1.

Nos travaux et ceux de Luo et Wang [175] ont déjà permis d'identifier de nombreux cas polynomiaux et il est important de continuer cet effort. En effet, les chemins colorés par exemple sont une brique de base pour de nombreux problèmes d'optimisation que nous rencontrons dans les réseaux (roulage avec protection, groupage de trafic,...). Pour aller plus loin, il serait intéressant de pouvoir utiliser d'autres propriétés structurelles des graphes colorés modélisant des problèmes de réseaux avec SRLG. Voge [182] a fait un premier pas dans cette direction en

Min. Color ...		(s, t) -Path	(s, t) } Multi }-Cut	Cut	2-Disjoint (s, t) - Paths	2-Min Overlap. (s, t) -Paths	Max. number of Disjoint (s, t) -Paths	Spanning Tree
general	complexity	NP [177]	NP	?	NP-Compleat [171]	NP [171]	NP [220]	NP [106]
	non approx	$2^{\log^{1-\delta} C ^{\frac{1}{2}}}$	$2^{\log^{1-\delta} C ^{\frac{1}{2}}}$?	—	?	$\forall \varepsilon > 0, V ^{\frac{1}{4}-\varepsilon}$	$o(\log(V))$ [108]
	approx	?	$O(\sqrt{m})$ [117]	?	—	?	?	$O(\log(V))$ [108]
span k	complexity	NP	NP	?	NP-Compleat	NP	NP	NP [106]
	non approx	$\exists \varepsilon > 0, k^\varepsilon$	$\exists \varepsilon > 0, k^\varepsilon$?	—	?	$\forall \varepsilon > 0, V ^{\frac{1}{4}-\varepsilon}$	$o(\log(V))$ [108]
	approx	k	k	k	—	?	?	$O(\log(V))$ [108]
span 1	complexity							NP [106]
	non approx	P [183]	P	P	P	P	P	$o(\log(V))$ [108]
	approx							$O(\log(V))$ [108]
Star property	P [175]	NP	?	P [175]	P	?	NP	
k col. span > 1	FPT	FPT	?	NP	NP	NP	NP	
bounded degree	—	P	P	—	—	—	—	

? = open question, $\delta = (\log \log |C|^{\frac{1}{2}})^{-\varepsilon}$ for $\varepsilon < \frac{1}{2}$, — = do not apply.

TAB. 4.1 – Complexité et (in)approximabilité des problèmes colorés.

montrant qu'il existe des graphes colorés qui ne peuvent pas modéliser des problèmes dans des réseaux multi-couches, où les SRLG correspondent aux arêtes de la topologie logique qui sont routées sur un même lien de la topologie physique. Toutefois, nous ne savons pas encore détecter de tels graphes et utiliser cette information. Il faut donc étudier les questions suivantes :

Question 1 *Quelles sont les caractéristiques des graphes (multi) colorés modélisant des problèmes de réseaux ?*

Question 2 *Comment utiliser les propriétés structurelles des graphes (multi) colorés modélisant des problèmes de réseaux pour résoudre plus efficacement les problèmes d'optimisation dans ce contexte ?*

Chapitre 5

Reconfiguration de routages

Dans ce chapitre, je présente mes principales contributions sur le problème de la reconfiguration du routage dans les réseaux orientés connexions, c'est-à-dire le problème de passer un ensemble de connexions d'un routage à un autre en limitant les interruptions de services.

Je commence dans la section 5.1 par motiver le problème et expliquer mon positionnement vis-à-vis de l'état-de-l'art. Ensuite, dans la section 5.2, je présente la modélisation générale qui est au centre de mes travaux. Cette modélisation exprime le problème de reconfiguration considéré comme un jeu sur le graphe orienté représentant les dépendances entre les deux routages. Puis, dans la section 5.3, je discute de plusieurs métriques (sans pondération) pour ce jeu. J'introduis un nouvel invariant, l'indice de traitement, et je montre les liens avec des invariants de graphe connus comme l'indice d'échappement dans les jeux de capture et l'indice de transmission. Dans la section 5.4, j'introduirai des modèles avec pénalités ou pondération. Dans la section 5.5, je présente nos contributions théoriques sur l'indice de traitement, ainsi que sur d'autres invariants. En particulier, je présenterai la caractérisation des graphes orientés et non-orientés d'indice de traitement au plus 2, un algorithme distribué permettant de calculer plusieurs invariants dont l'indice de traitement dans les arbres, et enfin une contribution sur la largeur de chemin des graphes planaires extérieurs. Je conclus ce chapitre dans la section 5.6 avec les pistes à explorer qui me semble les plus intéressantes.

Comme nous le verrons tout au long de ce chapitre, une des contributions les plus importantes de ces travaux a été la modélisation du problème de reconfiguration par un jeu semblable aux jeux de capture. Ce faisant, nous avons apporté un ensemble d'outils algorithmiques et de la théorie des graphes pour aborder ce problème.

5.1 Introduction

Dans les réseaux optiques à multiplexage en longueur d'onde (*Wavelength Division Multiplexing*, WDM), nous associons à chaque requête un chemin optique dans le réseau (un chemin et une longueur d'onde), sous la contrainte que deux chemins optiques utilisant la même fibre optique sur le même lien du réseau ont des longueurs d'ondes différentes. Le problème classique, qui a été très largement étudié dans la littérature, est celui du routage et de l'affectation de longueur d'onde (*routing and wavelength assignment*, RWA). Les méthodes qui me semblent aujourd'hui les plus performantes pour résoudre ce problème sont : les méthodes par programmation linéaire avec génération de colonnes telles que celles développées par Jaumard et al.[218] et les méthodes basées sur l'arrondi aléatoire du multiflot fractionnaire que nous avons proposé [Ci20, Ci23, Ci24, Cn42, Cn43].

Ces travaux se basent sur une topologie physique et un ensemble de requêtes de connexions donnés. Toutefois, les opérateurs de réseaux optiques sont amenés à changer régulièrement (e.g.,

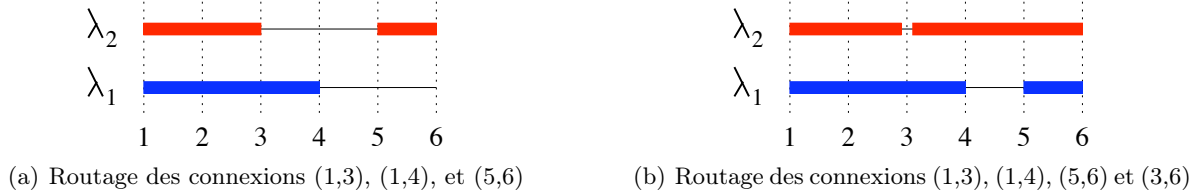


FIG. 5.1 – Le réseau est un chemin à 6 nœuds et 2 longueurs d’ondes. Avec le routage de la Fig. 5.1(a), il n’est pas possible d’accepter une nouvelle connexion (3,6). En changeant le routage de la connexion (5,6), nous obtenons le routage de la Fig. 5.1(b) avec la connexion (3,6).

tous les jours, toutes les heures) le routage de ces connexions, soit pour optimiser l’usage des ressources lors de changements de trafic (ajout ou suppression de connexions), soit pour détourner les connexions d’un lien devant subir une opération de maintenance. Par exemple, sur la figure 5.1, il n’est pas possible d’accepter un nouvelle connexion (3,6) sans modifier le routage de la connexion (5,6), alors que les 4 requêtes peuvent être satisfaites simultanément en changeant le routage. La figure 5.2 représente un autre exemple où le lien $\{5, 8\}$ doit subir une opération de maintenance. Avec le routage initial de la figure 5.2(a), il faut rerouter (changer le routage) de la connexion a . Mais, comme il n’y a pas de route disponible du nœud 4 au nœud 5 dans le routage courant, il est nécessaire de rerouter d’autres connexions pour obtenir un routage valide tel que celui de la figure 5.2(b). Donc, une opération de maintenance sur un lien du réseau peut impacter plus de connexions que celles dont les routes empruntent ce lien. Ceci soulève de nombreuses questions, dont :

- Quel nouveau routage choisir ? Faut il prendre en considération le routage courant ?
- Comment passer, en pratique, du routage courant au nouveau routage ?

Ces questions se posent pour toutes les technologies de réseaux orientés connexions, comme dans les années 70 pour les réseaux téléphoniques [190], ou plus récemment pour les réseaux WDM [191, 192, 193, 194, 196, 205, Ci33] et MPLS (Multi-Protocol Label Switching) [198, 201, 207]. De nombreuses approches ont été proposées (voir les état-de-l’art [195, 197, 203]).

Une approche classique est de se déplacer vers ce qui est libre (*Move-To-Vacant*, MTV) [193, 194, 205], dont le principe est de déterminer une séquence de déplacements de connexions pour atteindre un routage satisfaisant. Le principe est donc, en partant du routage courant, de choisir une connexion, lui calculer une nouvelle route en utilisant les ressources disponibles, la déplacer sur cette nouvelle route, puis répéter avec une autre connexion jusqu’à ce qu’un critère de qualité soit atteint (e.g., usage global des ressources, disponibilité d’une route donnée). Les principales difficultés de ces approches sont d’une part de garantir la convergence de l’algorithme et d’autre part de contrôler le nombre total de changements de routes (ou le temps de convergence). De plus, que faire si une telle séquence n’existe pas, comme c’est le cas dans l’exemple de la figure 5.2 ?

D’autres travaux ont proposé de calculer le nouveau routage R' , optimal selon une fonction de coût f , et qui soit le plus proche du routage courant R , la notion de proximité correspondant au nombre de connexions devant changer de route de R à R' . Les solutions proposées sont basées sur des programmes linéaires en nombres entiers [196, 199]. Toutefois, comme mentionné dans [196], ces travaux laissent ouvert le problème de passer du routage R au routage R' .

Par la suite, des travaux ont cherché à calculer à la fois le nouveau routage et la séquence de déplacements de connexions permettant de l’atteindre. Les solutions proposées utilisent des programmes linéaires en nombres entiers de très grandes tailles qui sont difficiles à résoudre, et des algorithmes heuristiques, dans le contexte des réseaux MPLS [198, 201, 202, 207] et WDM [204, 206]. Toutefois, ces modèles échouent si il n’existe pas de séquence de déplacements

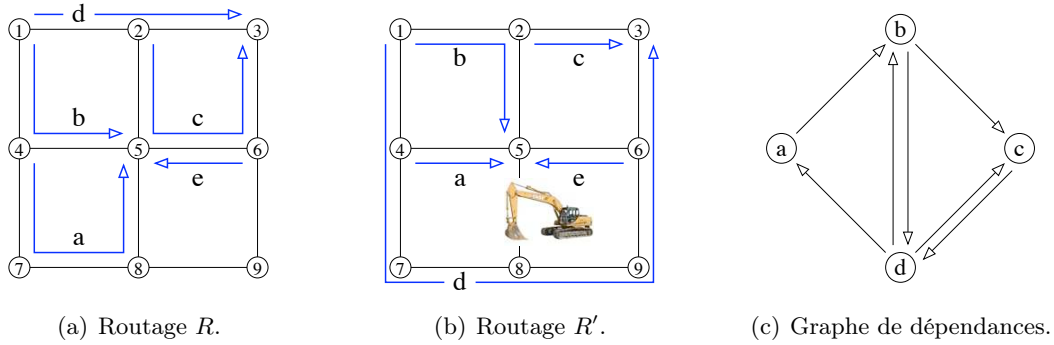


FIG. 5.2 – Le réseau est une grille 3×3 avec des liens symétriques de capacité 1 (i.e. une longueur d’onde). Les Figs. 5.2(a) et 5.2(b) donnent 2 routages, R et R' , pour l’ensemble de connexions $\{a, b, c, d, e\}$. La Fig. 5.2(c) représente les dépendances pour passer du routage R au routage R' . Il contient un sommet par requête ayant des routes différentes dans R et R' et un arc d’un sommet u vers un sommet v si le routage de la requête u dans R' intersecte le routage de v dans R .

comme dans le cas de la figure 5.2.

Notre approche sur le problème de la reconfiguration de routage a été de se concentrer sur le problème de la transition entre un routage et un autre. Aussi, nous considérons que les routages d’origine et de destination sont des données du problème et nous cherchons à déterminer la meilleure stratégie (séquence de déplacements de connexions) pour passer d’un routage à un autre. De plus, nous nous autorisons à interrompre certaines connexions au cours de la stratégie si cela est nécessaire. L’objectif est donc d’apporter une réponse au problème identifié mais laissé ouvert par de nombreux auteurs. Ce problème avait jusque là uniquement été abordé par Jose et Somani [200] qui ont proposé un algorithme heuristique cherchant à minimiser le nombre total de connexions interrompues, comme nous le verrons dans la section 5.3. Très récemment, Solano et Pióro [209, 210, 211] ont suivi notre approche en proposant un nouvel algorithme heuristique pour optimiser le nombre simultané de connexions interrompues (qu’ils n’ont curieusement pas comparé au notre...) et considéré le compromis entre nombre total et simultané de connexions interrompues.

5.2 Modélisation

Les concepts de *make-before-break* et *break-before-make* ont été standardisés pour les réseaux MPLS. Un *make-before-break* consiste à d’abord réserver les ressources nécessaires à la nouvelle route d’une connexion, puis à y établir le trafic avant de libérer les ressources utilisées par l’ancienne route. Un *break-before-make* consiste à d’abord interrompre la connexion et à libérer les ressources qui étaient utilisées, puis à réserver les ressources de la nouvelle route avant de rétablir le trafic. Les approches citées précédemment ont seulement considéré l’usage de *make-before-break*. Cependant, comme nous le voyons sur la figure 5.2, cette commande n’est pas suffisante pour passer d’un routage à un autre. En autorisant également l’usage de la commande *break-before-make*, il est possible d’effectuer la transition entre les deux routages, comme nous le verrons avec la figure 5.4.

5.2.1 Graphe de dépendances

Nous nous intéressons au problème de passer un ensemble I de requêtes du routage courant vers un nouveau routage prédéterminé, en déplaçant les requêtes une par une¹. Afin de déplacer une requête sur sa nouvelle route, il faut préalablement s'assurer que les ressources (capacité, longueur d'onde) à utiliser sur la nouvelle route sont disponibles. Si elles ne le sont pas, alors une ou plusieurs requêtes doivent être déplacées avant.

Pour modéliser ces dépendances, nous construisons un *graphe de dépendances* (*dependency digraph* [200]). Ce graphe orienté a un sommet par requête qui doit être déplacée et il y a un arc du sommet u vers le sommet v si la requête associée au sommet v doit être déplacée avant la requête associée au sommet u . En d'autres termes, il y a un arc (u, v) si des ressources nécessaires à la requête associée à u dans le nouveau routage sont utilisées par la requête associée au sommet v dans le routage initial.

Pour fixer les idées, étudions l'exemple de la figure 5.2. Le réseau physique est une grille 3×3 dont les liens sont symétriques et de capacité 1 (i.e. une unité de capacité ou une longueur d'onde dans chaque sens). Les figures 5.2(a) et 5.2(b) représentent deux routages, R et R' , pour l'ensemble de requêtes $\{a, b, c, d, e\}$, chaque requête nécessitant une unité de capacité ou une longueur d'onde. Le graphe de dépendances représenté par la figure 5.2(c) a un sommet par requête dont le routage change de R à R' . La requête e , ayant le même routage dans R et R' , n'intervient pas dans la construction du graphe de dépendances. Le routage de la requête a dans R' utilise le lien $(4, 5)$ qui est utilisé par la requête b dans R . Il faut donc changer le routage de la requête b avant que la requête a puisse utiliser le lien $(4, 5)$. Il y a une dépendance et donc un arc de a vers b dans le graphe de dépendances. De la même façon, la requête b doit attendre que le lien $(1, 2)$ ait été libéré par la requête d et le lien $(2, 3)$ par la requête c . Le graphe de dépendances contient donc les arcs (b, c) et (b, d) , et ainsi de suite.

En fait, un arc (u, v) dans le graphe de dépendances indique que les requêtes correspondantes changent toutes les deux de routage, l'une libérant une ressource que l'autre convoite. La figure 5.3 montre une modification simple du routage engendrant un arc dans le graphe de dépendances. En utilisant cette brique de base dans une grille suffisamment grande où n requêtes sont initialement routées sur les lignes, il est possible de construire n'importe quel graphe orienté, comme nous l'avons montré dans [S55]. Nous pouvons aussi remarquer que le degré entrant et sortant des sommets du graphe de dépendances est borné par la longueur du plus long chemin.

Théorème 31 ([S55]) *Pour tout graphe orienté D , il existe un réseau, un ensemble de requêtes et deux configurations C_1 et C_2 tels que D modélise les dépendances pour passer de la configuration C_1 à la configuration C_2 .*

Remarque : Ici, je me place de façon implicite dans le cas des réseaux WDM où nous associons à chaque connexion un chemin et une longueur d'onde. Aussi, il y a une relation binaire entre les connexions : il y a un arc de u vers v si et seulement si la connexion u doit utiliser dans le routage R' une longueur d'onde spécifique sur un lien donné qui est utilisée dans le routage R par la connexion v .

Dans le cas des réseaux MPLS, une connexion utilise une certaine quantité de capacité sur chacun des liens du réseau qu'elle emprunte. Elle peut être reroutée dès qu'il y a suffisamment de capacité disponible sur chacun des liens de sa nouvelle route. Pour un lien donné du réseau, la contrainte de dépendance doit prendre en compte l'ensemble des connexions qui doivent libérer

¹Le choix de déplacer les requêtes une par une est motivé par les contraintes physiques des réseaux WDM. En effet, l'établissement d'un chemin optique dans le réseau oblige à ajuster la longueur d'onde des longueurs d'ondes partageant les mêmes fibres optiques et ceci peut se propager à l'ensemble du réseau.

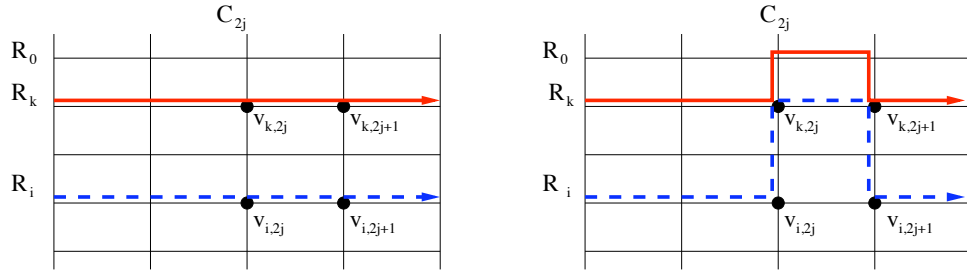


FIG. 5.3 – Construction élémentaire pour construire un arc dans le graphe de dépendances, utilisée pour prouver le théorème 31.

de la capacité sur ce lien, la capacité disponible, et l'ensemble des connexions qui utiliseront ce lien dans le nouveau routage. Voir [Ci34, 201, 207] pour plus de détails.

5.2.2 Stratégie de traitement

Lorsque le graphe de dépendances ainsi construit est un graphe orienté sans circuit (DAG), alors l'ordre dans lequel il faut rerouter les requêtes est immédiat. Pour cela, nous pouvons classer les sommets du DAG par rang : $\text{rang}(u) = 1 + \max_{v \in N^+(u)} \text{rang}(v)$, les feuilles ayant rang 1. La stratégie consiste alors à traiter les sommets par rangs croissants. Un sommet u de rang $\text{rang}(u)$ est donc traité après que tous ses successeurs aient été traités. Ici, traiter un sommet du graphe de dépendances signifie que la requête correspondante passe de sa route initiale à sa route finale. L'ordre induit par le rang assure donc la disponibilité des ressources lors du reroutage d'une requête.

Toutefois, tous les graphes de dépendances ne sont pas des DAGs, comme c'est le cas pour le graphe représenté sur la figure 5.2(c). Il faut alors "casser" des cycles de dépendances afin de déterminer un ordre de reroutage des requêtes. Ceci revient à interrompre temporairement des requêtes. Pour cela, nous utilisons une commande à *la break-before-make*, c'est-à-dire que la requête est interrompue pendant la durée nécessaire à la libération des ressources requises par sa nouvelle route². Pendant ce temps, les ressources qui étaient utilisées par la requête interrompue sont libérées et peuvent être directement utilisées par d'autres requêtes. Pour résumer, la séquence d'opérations effectuées pour passer du routage initial au routage final implique l'utilisation des deux types de commandes : *make-before-break* et *break-before-make*.

Pour modéliser l'interruption d'une requête dans le graphe de dépendances, et donc l'utilisation d'un *break-before-make*, nous utilisons la notion d'*agents*. Placer un agent sur un sommet du graphe de dépendances équivaut à interrompre la requête correspondante. Retirer un agent d'un sommet du graphe implique que la requête correspondante est maintenant routée sur sa route finale. L'ordre dans lequel sont effectuées ses opérations s'exprime alors par le jeu suivant :

Définition 32 (Stratégie de traitement, *process strategy*) *Etant donné un graphe orienté D (resp. non-orienté G) et un ensemble \mathcal{P} d'agents, une stratégie de traitement de D (resp. G) est une succession des opérations suivantes :*

- (R₁) *Placer un agent sur un sommet.*
- (R₂) *Retirer un agent d'un sommet si tous ses successeurs (resp. voisins) sont traités ou occupés par un agent. Le sommet est alors traité.*

²Il est également possible d'utiliser une route temporaire pour la requête si suffisamment de ressources sont disponibles.

(R_3) *Traiter un sommet (non couvert par un agent) si tous ses successeurs (resp. voisins) sont traités ou occupés par un agent.*

La figure 5.4 détaille la stratégie de traitement du graphe de dépendances de la figure 5.2(c) pour passer du routage de la figure 5.2(a) au routage de la figure 5.2(b). Comme le graphe de dépendances est fortement connexe, il est nécessaire d’interrompre au moins une requête, et donc d’utiliser un agent au cours de la stratégie. Le premier pas de la stratégie est donc de placer un agent sur le sommet d (figure 5.4(b), règle (R_1)). C’est le “break” de la commande break-before-make. Le seul successeur du sommet c est alors couvert par un agent et nous pouvons appliquer la règle (R_3) qui nous permet de traiter le sommet c et donc de rerouter la requête c (figure 5.4(c)). Ensuite, nous faisons de même pour les sommets b puis a (figures 5.4(d) et 5.4(e)). Enfin, comme tous les successeurs du sommet d ont été traités, nous appliquons la règle (R_2) pour traiter le sommet d et donc router la requête d sur sa route finale (figure 5.4(f)). Ce dernier pas est donc le “make” de la commande break-before-make. Tout au long de cette stratégie, nous vérifions bien qu’un sommet n’est traité dans le graphe de dépendances que lorsque les ressources nécessaires à l’établissement de la nouvelle route de la requête correspondante sont disponibles.

5.3 Métriques sans pénalité

Pour évaluer le coût d’une stratégie de traitement, plusieurs métriques sont possibles. Dans cette section, nous nous intéressons au cas sans pénalité, c’est-à-dire non pondéré, et nous nous intéressons au nombre de connexions interrompues au cours de la stratégie.

Tout d’abord, si nous cherchons à minimiser le nombre total de requêtes à interrompre au cours de la stratégie, il est alors facile de voir que ce nombre n’est autre que l’*indice de transmission* (taille d’un *minimum feedback vertex set*, MFVS) [59] du graphe de dépendances. En effet, l’indice de transmission est la taille du plus petit ensemble de sommets à supprimer d’un graphe orienté pour le rendre acircuitique. C’est cette approche qu’ont choisie Jose et Somani [200] qui, sans le savoir, ont proposé un algorithme heuristique pour MFVS.

Nous nous sommes plus particulièrement intéressés à optimiser le nombre d’agents utilisés simultanément au cours de la stratégie, c’est-à-dire le nombre de requêtes simultanément interrompues : l’*indice de traitement* du graphe [Cn44].

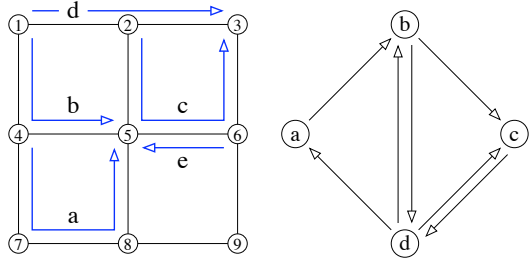
Définition 33 (*p -stratégie de traitement, p -process strategy*) *Une p -stratégie de traitement est une stratégie utilisant au plus p agents simultanément.*

Définition 34 (*Indice de traitement, process number, pn*) *L’indice de traitement d’un graphe est le plus petit entier p pour lequel il existe une p -stratégie de traitement.*

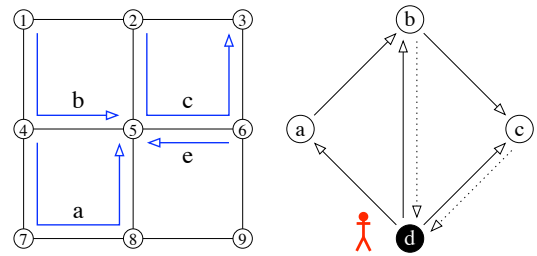
Clairement, un DAG a un indice de traitement nul, aucun agent n’étant utilisé, et le graphe de dépendances de la figure 5.2(c) a un indice de traitement égal à 1.

Une remarque importante est que l’indice de traitement d’un graphe orienté est égal au plus grand indice de traitement de ses composantes fortement connexes (CFCs). En effet, le traitement des CFCs du graphe s’effectue dans l’ordre induit par le DAG de ses CFCs (défini par un sommet u par CFC, C_u , et un arc (u, v) si le graphe contient des arcs de C_u vers C_v). La stratégie ne place des agents que dans une CFC à la fois. Aussi, nous ne nous intéressons qu’aux graphes orientés fortement connexes.

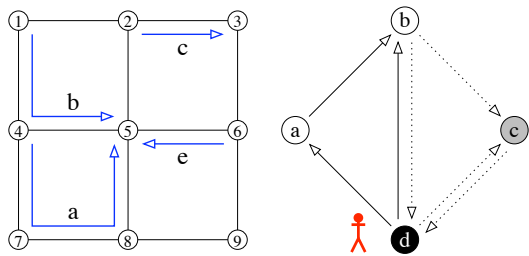
Lemme 35 *Etant donné un graphe orienté D et \mathcal{C} l’ensemble de ses composantes fortement connexes, nous avons $\text{pn}(D) = \max_{C \in \mathcal{C}} \text{pn}(C)$.*



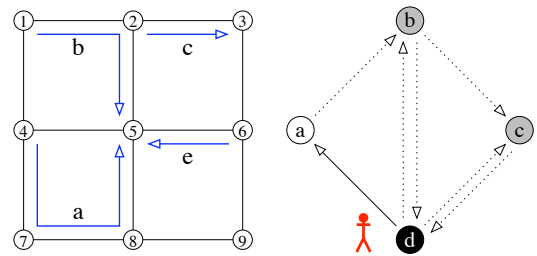
(a) Routage initial et graphe de dépendances du routage R vers le routage R' .



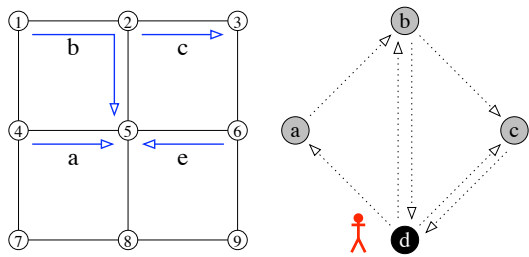
(b) Placer un agent sur le sommet d et interrompre la requête d .



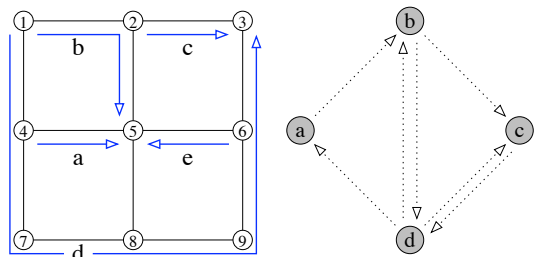
(c) Traiter le sommet c et rerouter la requête c .



(d) Traiter le sommet b et rerouter la requête b .



(e) Traiter le sommet a et rerouter la requête a .



(f) Retirer l'agent du sommet d , le traiter et rerouter la requête d .

FIG. 5.4 – Stratégie de traitement de l'exemple de la Fig. 5.2 : Les sommets traités sont en gris et ceux couverts par un agent en noir. Tous les sommets de la Fig. 5.4(b) ont au moins un successeur dans l'état initial et le graphe est fortement connexe. Il faut donc placer un agent sur un sommet. Nous choisissons de placer un agent sur le sommet d (Fig. 5.4(b)). Tous les successeurs du sommet c étant alors traités ou couverts par un agent, nous pouvons le traiter (Fig. 5.4(c)). Ensuite, nous pouvons traiter le sommet b (Fig. 5.4(d)), puis le sommet a (Fig. 5.4(e)). Finalement, nous pouvons traiter le sommet d (Fig. 5.4(f)) et libérer l'agent.

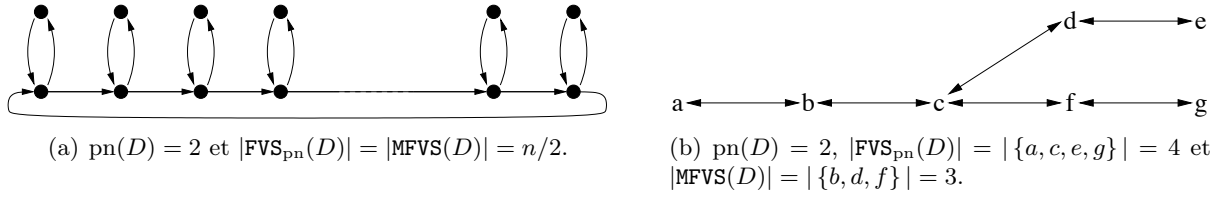


FIG. 5.5 – indice de traitement *vs* indice de transmission.

5.3.1 Indice de traitement *vs* indice de transmission

Il est clair que l'indice de transmission d'un graphe orienté D est une borne supérieure pour l'indice de traitement. Nous avons donc $\text{pn}(D) \leq |\text{MFVS}(D)|$. Dans les graphes non-orientés, *vs* comme des graphes orientés symétriques, la borne supérieure est donnée par *l'ensemble minimum couvrant les sommets* (*minimum vertex cover*, MVC) qui est le complémentaire de *l'ensemble indépendant maximum* (*maximum independent set*, MIS)³. Nous avons donc $\text{pn}(G) \leq |\text{MVC}(G)|$.

Ensuite, il existe de nombreux exemples de graphes tels que $|\text{MVC}(G)| = \text{pn}(G)$. Par exemple, pour le graphe biparti complet $K_{p,q}$, avec $1 \leq p \leq q$, nous avons $\text{pn}(K_{p,q}) = |\text{MVC}(K_{p,q})| = p$, et pour la clique d'ordre n , K_n , nous avons $\text{pn}(K_n) = |\text{MVC}(K_n)| = n - 1$.

Toutefois, l'écart entre ces deux indices peut-être arbitrairement grand. Par exemple, pour l'anneau d'ordre $n \geq 5$, C_n , nous avons $\text{pn}(C_n) = 3$ et $|\text{MVC}(C_n)| = \lceil \frac{n}{2} \rceil$, et pour le graphe D représenté sur la figure 5.5(a) (dont l'équivalent non orienté est un peigne), nous avons $\text{pn}(D) = 2$ et $|\text{MFVS}(D)| = n/2$. Plus généralement, nous avons montré dans [S50] qu'ajouter des boucles aux sommets d'un graphe orienté D augmente l'indice de traitement d'au plus 1, alors que cette opération force l'indice de transmission à n . L'écart entre ces deux indices dans les graphes orientés est donc non borné.

De plus, l'ensemble des sommets occupés par un agent au cours d'une $\text{pn}(D)$ -stratégie de traitement ne constitue pas nécessairement un ensemble de taille minimum intersectant tous les circuits du graphe. Par exemple, pour le graphe représenté sur la figure 5.5(b) ces ensembles ont des tailles différentes. En effet, nous avons $\text{MFVS}(D) = \{b, d, f\}$, $|\text{MFVS}(D)| = 3$, et le plus petit ensemble des sommets occupés par une $\text{pn}(D)$ -stratégie de traitement est $\text{FVS}_{\text{pn}}(D) = \{a, c, e, g\}$ (le sommet c et un sommet par branche) de taille 4.

Donc, la connaissance de l'indice de transmission n'est pas suffisante pour déterminer ou approcher l'indice de traitement d'un graphe.

5.3.2 Complexité et inapproximabilité

Pour établir la complexité du problème de déterminer l'indice de traitement d'un graphe, nous avons encadré l'indice de traitement par un invariant connu pour être difficile à approcher : la *sommet séparation* [78].

Définition 36 (Sommet séparation, *vertex separation*, *vs*) *Un arrangement linéaire (linear layout ou linear arrangement) d'un graphe orienté $D = (V, A)$ est une bijection $L : V \rightarrow \{1, 2, \dots, |V|\}$. La coupe maximum induite par les arcs retour de l'arrangement L est mesurée par $\max_{1 \leq i \leq |V|} |M_L(i)|$, avec*

$$M_L(i) = \{v \in V : L(v) > i \text{ et } \exists u \in N^+(v) : L(u) \leq i\}.$$

La *sommet séparation* de D , notée $\text{vs}(D)$, est le minimum pris sur tous les arrangements, $\text{vs}(D) = \min_L \max_{1 \leq i \leq |V|} |M_L(i)|$.

³L'ensemble indépendant maximum est le plus grand ensemble X de sommets d'un graphe G tel que pour toute paire de sommets $u, v \in X$, $N(u) \cap N(v) = \emptyset$.

Proposition 37 ([Cn44]) *Pour tout graphe orienté D , $vs(D) \leq pn(D) \leq vs(D) + 1$.*

Comme le problème de détermination la somme de séparation d'un graphe orienté n'admet pas d'algorithme d'approximation à un facteur constant [59], nous en déduisons :

Corollaire 38 ([Cn44]) *Le problème de déterminer l'indice de traitement d'un graphe orienté D donné est NP-complet et n'est pas dans APX (APX-hard).*

Nous avons formulé ce problème sous la forme d'un programme linéaire en nombres entiers. Cependant, cette formulation utilisant un nombre quadratique de variables binaires, il n'est possible d'obtenir des solutions que sur de petites instances. Aussi, nous avons proposé un algorithme heuristique pour l'indice de traitement (voir [S52] et [Ci33] pour plus de détails). Nous avons testé cet algorithme sur des familles particulières de graphes aléatoires dont nous connaissons l'indice de traitement par construction et les résultats obtenus sont encourageants. Solano [209] a depuis proposé un algorithme heuristique pour ce problème en prenant l'hypothèse que l'ensemble des connexions qui seront interrompues au cours de la stratégie est une entrée du problème.

La proposition 37 s'applique également aux graphes non-orientés. En effet, lorsque le graphe est orienté symétrique, nous pouvons travailler sur le graphe non-orienté support. Il est alors plus naturel d'utiliser la notion de largeur de chemin (*pathwidth*) [63] qui a été montrée égale à la somme de séparation dans les graphes non-orientés par Kinnersley [67].

Définition 39 (Largeur de chemin, *pathwidth*, pw) *La largeur de chemin d'un graphe G est la taille minimum de la clique maximale d'un graphe d'intervalle contenant G comme sous-graphe, moins un.*

Corollaire 40 ([Cn44]) *Pour tout graphe non-orienté G , $pw(G) \leq pn(G) \leq pw(G) + 1$.*

Il est important de remarquer que les bornes du corollaire 40 sont atteintes. En effet, l'anneau à 4 sommets, C_4 , a un indice de traitement et une largeur de chemin égaux à 2, $pn(C_4) = pw(C_4) = 2$. L'anneau à $n \geq 5$ sommets, C_n , quant à lui est tel que $pn(C_n) = 3$ et $pw(C_n) = 2$. En utilisant les graphes orientés symétriques associés, nous observons que les bornes de la proposition 37 sont également atteintes. Plus généralement, nous avons montré :

Théorème 41 ([Cn44]) *Pour tout $x \geq 1$ et $p \in \{x, x + 1\}$, il existe un graphe orienté D (resp. non-orienté G) tel que $vs(D) = x$ et $pn(D) = p$ (resp. $pw(G) = x$ et $pn(G) = p$).*

5.4 Modèles avec pénalités et interdictions

Dans cette section, nous introduisons des pénalités et des impossibilités dans le problème de la reconfiguration. En d'autres termes, nous introduisons différents contrats de services (*service level agreement*, SLA) passés entre le client et l'opérateur qui ont traités à la reconfiguration.

5.4.1 Interdictions – Requêtes qui ne peuvent pas être interrompues

Nous commençons par considérer le cas des requêtes qui ne peuvent pas être interrompues au cours d'une phase de reconfiguration. Ces requêtes appartiennent à des clients privilégiés (premium) qui ont inclus dans leur contrat de service avec l'opérateur une clause interdisant les *break-before-make*. Nous modélisons de telles requêtes par des *sommets interdits*, c'est-à-dire des sommets sur lesquels il est interdit de placer un agent. Pour les autres requêtes, nous appliquons les règles précédentes.

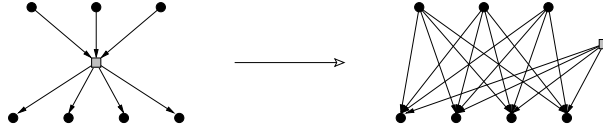


FIG. 5.6 – Principe de la transformation

Nous notons $\text{pn}^*(D, Q)$ l'indice de traitement d'un graphe orienté D dont les sommets $Q \subseteq V(D)$ sont interdits. Si nous reprenons l'exemple de la figure 5.2(c) page 41 en supposant qu'il est interdit de placer un agent sur le sommet d , alors une stratégie de traitement consiste à placer un agent sur les sommets b et c , ensuite à traiter le sommet a puis le d et enfin à traiter les sommets b et c . Nous avons donc $\text{pn}^*(D, \{d\}) = 2$ alors que $\text{pn}(D) = 1$.

Remarquons que la présence de sommets interdits peut augmenter l'indice de traitement de façon très importante. Par exemple, si le centre c de l'étoile d'ordre n , S_n , est interdit, alors il est nécessaire de placer un agent sur toutes les feuilles, soit $\text{pn}^*(S_n, \{c\}) = n - 1$ alors que $\text{pn}(S_n) = 1$. La présence de sommets interdits peut même empêcher l'existence d'une stratégie de traitement, comme nous l'avons montré dans le lemme 42, où $D[Q]$ est le sous-graphe de D ne contenant que les sommets de Q .

Lemme 42 ([Ci33]) *Soit D un graphe orienté et $Q \subseteq V(D)$ un ensemble de sommets interdits. $D[Q]$ contient un circuit orienté si et seulement si il n'existe aucune stratégie de traitement pour (D, Q) .*

En conséquence de ce lemme, nous observons qu'il existe toujours une stratégie lorsque $|Q| \leq 1$, et que la probabilité qu'il existe une impossibilité augmente avec $|Q|$. En particulier, si D est orienté symétrique, il ne faut pas que deux sommets voisins soient interdits. Notons de plus que si D est un DAG et quelque soit $Q \subseteq V(D)$, nous avons $\text{pn}^*(D, Q) = 0$.

Construisons maintenant le graphe $D^*(v)$ obtenu à partir de D pour le sommet interdit $v \in V(D)$ en supprimant les arcs des sommets de $N^-(v)$ vers v et en ajoutant des arcs de tous les sommets de $N^-(v)$ vers tous les sommets de $N^+(v)$, comme indiqué sur la figure 5.6. Dans le graphe $D^*(v)$, il n'est plus nécessaire de considérer le sommet v comme interdit. En effet, comme il n'a pas de prédécesseur, il sera simplement traité après ses successeurs. Aussi, une stratégie de traitement de $D^*(v)$ nous donne une stratégie de traitement valide de $(D, \{v\})$. De plus, cette stratégie est optimale comme nous l'avons montré dans la proposition 43.

Proposition 43 ([Ci33]) *Pour tout graphe orienté D et pour tout sommet interdit $v \in V(D)$, le graphe $D^*(v)$ est tel que $\text{pn}^*(D, \{v\}) = \text{pn}(D^*(v))$.*

En appliquant récursivement cette transformation pour tous les sommets interdits de Q d'un graphe D , nous obtenons un graphe D^* sans sommet interdit dont l'indice et la stratégie de traitement fournissent l'indice et la stratégie de traitement pour (D, Q) . De plus, l'ordre dans lequel sont effectuées les transformations ne change pas D^* .

Corollaire 44 ([Ci33]) *Etant donné un graphe orienté D et l'ensemble $Q = \{v_1, v_2, \dots, v_f\} \subseteq V(D)$ de sommets interdits, nous posons $D_1 = D^*(v_1)$ et, pour $i = 2, 3, \dots, f$, $D_i = D_{i-1}^*(v_i)$. Alors, $\text{pn}^*(D, Q) = \text{pn}(D_f)$.*

Nous avons aussi obtenu une borne serrée sur le nombre d'agents supplémentaires à utiliser dans le graphe lorsque des sommets sont interdits.

Proposition 45 ([Ci33]) *Pour tout graphe orienté D et pour tout ensemble de sommet interdits $Q \subseteq V(D)$ tel que Q n'induit pas de circuit, nous avons $\text{pn}^*(D, Q) \leq \text{pn}(D) + |N^+(Q)|$. De plus, cette borne est serrée.*

En résumé, l'introduction de sommets interdits dans le problème de la reconfiguration introduit des situations de blocage, mais ne change pas la structure combinatoire du problème. Il suffit d'appliquer un pré-traitement au graphe D pour construire D^* , puis un post-traitement pour déduire la stratégie de traitement, ces deux opérations s'effectuant en temps linéaire.

Notons enfin que le même raisonnement est valide pour l'indice de transmission.

5.4.2 Pénalité par interruption

Nous considérons maintenant que le contrat de service passé entre le client et l'opérateur inclut une pénalité à chaque fois qu'une connexion est interrompue. Cette pénalité peut-être la même pour tous ou différente pour chaque connexion. L'intérêt de l'opérateur est alors de minimiser le coût total d'une phase de reconfiguration et donc d'une stratégie de traitement. Nous associons donc à chaque requête, modélisée par un sommet v dans le graphe de dépendances, une pénalité (ou poids) $\omega(v)$.

Définition 46 (p -stratégie de traitement de coût au plus W) *Etant donné un graphe orienté D , une fonction de pénalité $\omega : V(D) \rightarrow \mathbb{N}$, p agents et un budget W , $W \geq 0$, une p -stratégie de traitement de coût au plus W est une p -stratégie de traitement de D de coût $\sum_{v \in V'} \omega(v) \leq W$, où $V' \subseteq V$ est l'ensemble des sommets couverts par un agent au cours de la stratégie.*

Nous observons que le problème de déterminer le plus petit entier W pour lequel il existe un $(\leq |V(D)|)$ -stratégie de traitement de coût au plus W n'est autre que le problème de déterminer l'indice de transmission de coût minimum (*minimum weighted feedback vertex set*, **MWFVS**). En effet, ce problème n'a pas de restriction sur le nombre d'agents employés et ne s'intéresse qu'au coût. Donc,

Corollaire 47 *Le problème de déterminer le plus petit entier W pour lequel il existe une $(\leq |V(D)|)$ -stratégie de traitement de coût au plus W est NP-complet et n'est pas dans APX.*

Question 3 *Quelle est la complexité du problème lorsque p fait parti de l'entrée ?*

Lorsque les pénalités sont uniformes ($\forall v, \omega(v) = 1$), il est intéressant de remarquer que W est le nombre total d'interruptions autorisées et p est le nombre maximum d'interruptions simultanées. De plus, comme nous l'avons vu dans la section 5.3, il existe des graphes D pour lesquels $p = \text{pn}(D)$ et $W > |\text{MFVS}(D)|$. Dans l'exemple de la figure 5.5(b), l'ensemble $\text{FVS}_{\text{pn}}(D)$ des sommets couverts par un agent au cours d'une $\text{pn}(D)$ -stratégie de traitement est de taille 4 alors que l'indice de transmission est 3. Maintenant, si seul $|\text{MFVS}(D)|$ sommets peuvent être couverts par un agent au cours de la stratégie, alors il est nécessaire d'utiliser 3 agents. En d'autres termes, le plus petit p pour lequel il existe une p -stratégie de traitement de coût 3 est 3.

Notons $\text{MFVS}_p(D)$ le plus petit ensemble de sommets couverts par un agent au cours d'une p -stratégie de traitement (qui a donc coût $|\text{MFVS}_p(D)|$). Nous avons

Lemme 48 *Etant donné un graphe orienté D et un entier $p \geq \text{pn}(D)$, nous avons*

$$|\text{MFVS}_p(D)| \geq |\text{MFVS}_{p+1}(D)| \geq |\text{MFVS}(D)|$$

En d'autres termes, augmenter le nombre d'agents permet de réduire le nombre de connexions à interrompre au cours de la stratégie, et réciproquement.

Question 4 *Quels sont les graphes tels que $|\text{MFVS}_{\text{pn}}(D)| = |\text{MFVS}(D)|$?*

Question 5 *Existe-t-il une constante c telle que $|\text{MFVS}_{\text{pn}}(D)| \leq c|\text{MFVS}(D)|$ pour tout D ?*

Dans [S55], nous avons répondu à cette question par la négative pour les graphes orientés (en utilisant une famille de graphes orientés d'indice de traitement 3). Pour les graphes non-orientés ou la question se transpose avec MVC, nous avons montré qu'il existe des graphes tels que $c \geq 3 - \varepsilon$, pour tout $\varepsilon > 0$, et que pour tout G , $|\text{MFVS}_{\text{pn}}(G)| \leq \text{pn}(G)|\text{MFVS}(G)|$.

5.4.3 Pénalité par étape

Nous considérons maintenant des contrats de services plus "agressifs", prenant en compte la durée d'une interruption. En effet, certaines stratégies peuvent avoir comme première étape de placer un agent sur un sommet v et comme dernière étape de retirer cet agent de v . C'est par exemple le cas pour l'anneau unidirectionnel d'ordre n , \vec{C}_n , où un agent est suffisant. La connexion en question peut donc être interrompue pendant un très longue durée.

Pour préciser notre propos, formalisons la notion d'étape.

Définition 49 (Etape dans une stratégie de traitement) *Une étape dans une stratégie de traitement consiste en*

1. le placement d'un ensemble d'agents parmi les agents libres ;
2. l'identification des sommets dont tous les successeurs sont déjà traités ou couverts par un agent ;
3. le traitement de tous les sommets identifiés. Si certains de ces sommets étaient couverts par un agent, ceux-ci sont libérés.

Le nombre d'agents présents dans le graphe au cours d'une étape est le nombre d'agents utilisés à l'issue du premier pas de l'étape.

Définition 50 (coût d'une étape) *Le coût d'une étape est la somme des pénalités des sommets qui sont couverts par un agent après la phase 1 de l'étape.*

A partir de ces définitions (étape et coût d'une étape), nous pouvons définir les jeux suivants :

Définition 51 (p -stratégie de traitement de coût par étape au plus W) *Etant donné un graphe orienté D , une fonction de pénalité $\omega : V(D) \rightarrow \mathbb{N}$, p agents et un budget par étape W , $W \geq 0$, une p -stratégie de traitement de coût par étape au plus W est une p -stratégie de traitement de D telle que chaque étape t a un coût $\sum_{v \in V_t} \omega(v) \leq W$, où $V_t \subseteq V$ est l'ensemble des sommets couverts par un agent après la phase 1 de l'étape t .*

Avec des coûts uniformes, le problème de déterminer le plus petit W tel qu'il existe un $(\leq |V(D)|)$ -stratégie de traitement de coût par étape au plus W n'est autre que le problème de déterminer l'indice de traitement du graphe. Les pénalités plus générales rejoignent la notion de largeur de chemin pondérée (*weighted pathwidth*) qui a été montrée NP-difficile par Mihai et Todinca [95].

Définition 52 (*p*-stratégie de traitement de coût total en étapes au plus W)

Etant donné un graphe orienté D , une fonction de pénalité $\omega : V(D) \rightarrow \mathbb{N}$, p agents et un budget total W , $W \geq 0$, une *p*-stratégie de traitement de coût total en étapes au plus W est une *p*-stratégie de traitement de D telle que $\sum_{t=0}^{|V(D)|} \sum_{v \in V_t} \omega(v) \leq W$, où $V_t \subseteq V$ est l'ensemble (potentiellement vide) des sommets couverts par un agent à l'étape t et $V_0 = \emptyset$.

Une stratégie telle que décrite dans la définition 52 est intéressante pour l'opérateur qui pourra minimiser sa pénalité. Toutefois, cette stratégie peut-être très mauvaise pour certains clients dont les connexions pourront être interrompues pendant un grand nombre d'étapes. Dans une telle stratégie, la contribution d'un sommet au coût total est égale au nombre d'étapes pendant lesquelles ce sommet a été couvert par un agent fois sa pénalité par étape. Si la pénalité est égale à 1, alors la contribution du sommet est exactement égale à sa persistance (*persistence*), notion introduite par Downey et McCartin [82] dans les graphes non-orienté pour compter le nombre de paquets (successifs) auxquels appartient un sommet dans une décomposition en chemin du graphe.

Pour l'anneau unidirectionnel, $\vec{C}_n = (v_1, v_2, \dots, v_n, v_1)$, avec pénalité égale à 1 par sommet couvert et par étape, le coût d'une stratégie avec 1 agent est n . Toutefois, le sommet v_1 est couvert pendant n étapes. Maintenant, si nous utilisons 2 agents que nous plaçons sur les sommets v_1 et $v_{n/2}$, le coût de la stratégie est toujours n , mais les sommets v_1 et $v_{n/2}$ ne sont couverts que pendant $n/2$ étapes. Plus généralement, si nous disposons de p agents que nous plaçons sur les sommets $v_{1+in/p}$, $0 \leq i < p$, alors chaque sommet n'est couvert que pendant n/p étapes pour un coût total de n . Il est donc important de contrôler également le budget par connexion.

Si nous cherchons uniquement à optimiser le budget maximal par clients, nous obtenons le jeu décrit dans la définition 53 et donc à optimiser la persistance pondérée maximal dans une stratégie de traitement. Si nous cherchons de plus à contrôler le budget total, nous obtenons le jeu de la définition 54.

Définition 53 (*p*-stratégie de traitement de coût individuel au plus W) Etant donné un graphe orienté D , une fonction de pénalité $\omega : V(D) \rightarrow \mathbb{N}$, p agents et un budget par sommet W , $p, W \geq 0$, une *p*-stratégie de traitement de coût individuel au plus W est une *p*-stratégie de traitement de D telle que $\max_{v \in V(D)} \sum_{t=0}^{|V(D)|} b_t(v) \omega(v) \leq W$, où $b_t(v) = 1$ si le sommet v est couvert par un agent à l'étape t et 0 sinon.

Définition 54 (*p*-stratégie de traitement de coûts individuel $\leq W^I$ et total $\leq W^T$)

Etant donné un graphe orienté D , une fonction de pénalité $\omega : V(D) \rightarrow \mathbb{N}$, p agents, un budget par sommet W^I et un budget total W^T , $p, W^I, W^T \geq 0$, une *p*-stratégie de traitement de coût individuel au plus W^I et de coût total au plus W^T est une *p*-stratégie de traitement de D telle que

1. $\max_{v \in V(D)} \sum_{t=0}^{|V(D)|} b_t(v) \omega(v) \leq W^I$, où $b_t(v) = 1$ si le sommet v est couvert par un agent à l'étape t et 0 sinon ;
2. $\sum_{t=0}^{|V(D)|} \sum_{v \in V_t} \omega(v) \leq W^T$, où $V_t \subseteq V$ est l'ensemble (potentiellement vide) des sommets couverts par un agent à l'étape t et $V_0 = \emptyset$.

Ce dernier jeu peut encore être raffiné en adaptant le budget individuel à chaque client, le sommet v ayant donc un budget maximal $W^I(v)$. Ceci correspond bien au résultat d'une négociation individuelle entre un client et un opérateur, le contrat de service étant individualisé.

5.4.4 Nombre d'étapes

Dans les jeux que nous avons précédemment défini, une question intéressante est d'étudier le nombre d'étapes (définition 49) nécessaires à une stratégie, à budget constant ou en autorisant une augmentation du budget. Nous avons commencé à aborder cette question avec Ronan Soares [208] en supprimant les pénalités. Nous avons proposé le jeu suivant :

Définition 55 ((p, ς)-stratégie de traitement) Une (p, ς)-stratégie de traitement est une stratégie permettant de traiter un graphe avec p agents en ς étapes.

Lorsque $\varsigma = 1$, il est facile de voir que $p = n$. En effet, si un sommet v n'est pas couvert par un agent, alors il existe un prédécesseur u (resp. voisin) de v qui ne vérifie pas "tous les successeurs (resp. voisins) sont déjà traités ou occupés par un agent" et qui ne peut donc pas être traité à cette étape. Réciproquement, si $p = n$ alors $\varsigma = 1$ et si $p < n$ alors $\varsigma > 1$.

Théorème 56 ([208]) Etant donné un graphe orienté D , le problème de déterminer si il existe une (p, ς)-stratégie de traitement pour D lorsque $p < n$ (resp. $\varsigma > 1$) est donné est un problème NP-complet.

Nous avons proposé un algorithme linéaire pour déterminer le nombre d'étapes lorsque le graphe est une union disjointe de chemins avec un nombre donné d'agents. Puis nous avons étudié le gain, en nombre d'étapes, de l'utilisation d'un agent dans un DAG.

5.5 Relations avec les jeux de capture

Au cours de nos travaux sur l'indice de traitement d'un graphe, nous avons été amené à étudier certaines questions portant sur la largeur de chemin et les jeux de capture. Je résume ici les principales contributions.

5.5.1 Indice de traitement vs indice d'échappement

Nous avons vu dans la section 5.3.2 que l'indice de traitement d'un graphe non-orienté est relié à sa largeur de chemin. Nous allons maintenant montrer le lien avec l'indice d'échappement d'un graphe et donc avec les stratégies de capture ou jeux des gendarmes et du voleur. Je recommande au lecteur de se reporter à la thèse de Nisse [85] et à l'état de l'art établi par Fomin et Thilikos [89] pour plus de détails sur les jeux de capture dans les graphes. Je ne reporte ici que les résultats nécessaires à mon propos.

Les jeux de capture sont des jeux où une équipe de chercheurs ou d'agents vise à la capture d'un fugitif dans le graphe. Le fugitif est capturé si il occupe le même sommet ou la même arête qu'un agent ou une arête dont les extrémités sont occupées par des agents. Il existe de très nombreuses variantes de ces jeux selon que le fugitif est visible ou invisible, la vitesse du fugitif et des agents, ou encore le mode de déplacement (sommet voisin, le long d'un chemin, par téléportation,...). Ici, nous nous intéressons aux jeux défini par Kirousis et Papadimitriou [64] comme suit :

Définition 57 (Stratégie (sommet) [64]) Etant donné un graphe G , une stratégie (sommet) est une séquence des opérations suivantes :

- Placer un agent sur un sommet du graphe ;
- Retirer un agent d'un sommet du graphe.

Dans ce jeu, les agents se déplacent d'un sommet à un autre sans contrainte (par hélicoptère ou téléportation). Le fugitif quant à lui se déplace simultanément aux agents et rapidement le long d'un chemin dont aucun sommet n'est occupé par un agent. Dans ce modèle, une arête est nettoyée si ces 2 extrémités sont occupées par un agent, et le fugitif est capturé si, à une étape, il occupe le même sommet qu'un agent ou une arête dans les 2 extrémités sont occupées par un agent et qu'il n'a aucun moyen de s'échapper. Une *stratégie de capture* est une stratégie qui assure la capture du fugitif. La stratégie est dite *monotone* si une fois nettoyé un sommet ou une arête ne peut plus être atteint par le fugitif. Le plus petit nombre d'agents nécessaires dans une stratégie de capture pour capturer un fugitif invisible et infiniment rapide est appelé *l'indice d'échappement* du graphe.

Définition 58 (Indice d'échappement, *node search number*, ns) *Etant donné un graphe G , l'indice d'échappement du graphe G est le plus petit entier k pour lequel il existe une stratégie de capture dans G impliquant au plus k agents.*

Ellis *et al.* [68] ont montré que l'indice d'échappement d'un graphe est égal à sa largeur de chemin plus 1, $ns(G) = pw(G) + 1$. En combinant ce résultat et le corollaire 40, nous déduisons :

Corollaire 59 *Pour tout graphe G , $ns(G) - 1 \leq pn(G) \leq ns(G)$.*

La différence principale entre stratégie de capture et stratégie de traitement dans les graphes non orientés est que, dans cette dernière, le fugitif est aussi capturé si il est encerclé par des agents, c'est-à-dire si tous ses voisins sont couverts par des agents. Cette différence est claire avec l'étoile d'ordre n , S_n , et la clique d'ordre n , K_n , où nous avons $ns(S_n) = 2$, $pn(S_n) = 1$, $ns(K_n) = n$ et $pn(K_n) = n - 1$. Pour d'autres graphes tels que l'anneau d'ordre $n \geq 5$, C_n , et la grille $n \times n$ avec $n \geq 3$, $M_{n \times n}$, nous avons : $ns(C_n) = pn(C_n) = 3$ et $ns(M_{n \times n}) = pn(M_{n \times n}) = n + 1$.

Un problème important est donc de caractériser les graphes pour lesquels l'indice d'échappement est égal à l'indice de traitement. C'est ce que nous avons fait avec la proposition 60 dans le contexte plus général des graphes orientés, et que nous transposons avec le corollaire 61 dans les graphes non-orientés.

Proposition 60 ([S50]) *Pour tout graphe orienté D , il existe une $pn(D)$ -stratégie de traitement telle que chaque sommet est couvert par un agent avant d'être traité si et seulement si $pn(D) = vs(D) + 1$.*

Corollaire 61 *Pour tout graphe non-orienté G , il existe une $pn(G)$ -stratégie de traitement telle que chaque sommet est couvert par un agent avant d'être traité si et seulement si $pn(G) = ns(G)$.*

Nous avons déjà identifié plusieurs familles de graphes qui satisfont ce corollaire. Par exemple, nous avons caractérisé les graphes d'indice de traitement 2 [S50], comme nous le verrons dans la section 5.5.2, ce qui nous permet d'établir :

- $pn(G) = 0$ et $ns(G) = 1 \Leftrightarrow G$ est un ensemble indépendant ;
- $pn(G) = 1$ et $ns(G) = 2 \Leftrightarrow G$ est une étoile ou une collection d'étoiles ;
- $pn(G) = 2$ et $ns(G) = 2 \Leftrightarrow G$ est une chenille (*caterpillar*) ;
- $pn(G) = 2$ et $ns(G) = 3 \Leftrightarrow G$ satisfait les conditions du théorème 64 et G n'est pas une chenille (ce qui est le cas de tous les homards (*lobsters*)).

Par ailleurs, nous avons caractérisé à l'aide du théorème 62 les graphes dont la connexité est égale à l'indice de traitement. Ces graphes sont tous les sous-graphes p -connexes du graphe $G_{p,q}$ construit en reliant la clique K_p à un ensemble indépendant de taille $q \geq p$ par un biparti complet. En fait, pour maintenir la connexité, seules les arêtes de K_p peuvent être supprimées. Comme il est facile de voir que $ns(G_{p,q}) = p + 1$, nous en déduisons le corollaire 63.



FIG. 5.7 – P_4 et C_3 sont les 2 obstructions pour les graphes d'indice de traitement 1.

Théorème 62 ([S50]) *Un graphe p -connexe G admet une p -stratégie de traitement si et seulement si il existe un sommet v de degré p tel que $G - N(v)$ est un ensemble indépendant.*

Corollaire 63 *Les graphes p -connexes d'indice de traitement p ont indice d'échappement $p + 1$.*

Les jeux de capture et l'indice d'échappement ont principalement été étudiés sur les graphes non-orientés, mais quelques travaux traitent de ces jeux dans les graphes orientés. Tout d'abord, Barát [83] a défini la notion de largeur de chemin orienté ainsi que le jeu de capture associé. Dans ce jeu, le fugitif se déplace uniquement dans le sens des arcs, donc il peut se déplacer d'un sommet u vers un sommet v si il existe un chemin orienté de u vers v . Bien sur, le fugitif ne peut pas traverser un sommet occupé par un agent. Par contre, les agents peuvent se déplacer sans contrainte (téléportation). Par la suite, Yang et Cao [86, 91, 92, 97] ont étudié de multiples variantes de jeux de capture dans les graphes orientés.

Les différences principales entre ces jeux et les p -stratégies de traitements sont d'une part que nous considérons les arcs dans un sens différent, et d'autre part que ces jeux impose de visiter chaque sommet et/ou arête. Ceci suggère que, comme dans le cas des graphes orientés, le nombre d'agents nécessaires diffère d'un petit intervalle. Toutefois, nous n'avons pas encore étudié précisément les relations et les différences entre ces jeux, que nous laissons donc en question ouverte.

Question 6 *Quelles sont les relations entre les stratégies de traitements et les jeux de capture orientés ?*

5.5.2 Graphes d'indice de traitement inférieur à 2

Nous avons caractérisé les graphes non-orientés d'indice de traitement inférieur ou égal à 2 [S50]. Nous avons en particulier proposé une caractérisation par mineurs exclus. Pour cela, nous avons exhibé le plus petit ensemble de graphes d'indice de traitement 2 (resp. 3), les obstructions, auxquels G ne doit pas pouvoir être réduit par une succession de contractions d'arêtes et de suppressions d'arêtes et de sommets. Dans le cas contraire, G a un indice de traitement supérieur à 2 (resp. 3). Les 2 obstructions pour les graphes d'indice de traitement 1 sont représentées sur la figure 5.7, et les 15 obstructions pour les graphes d'indice de traitement 2 sont représentés dans les figures 5.8 et 5.9.

Rappelons qu'il a été montré par Bodlaender que toute classe de graphes close par mineurs ne contenant pas l'ensemble des graphes planaires admet un algorithme de reconnaissance en temps linéaire [70]. Ce résultat découle d'un algorithme en temps linéaire décidant si un graphe a une largeur d'arborescence ou de chemin au plus k , et qui, si tel est le cas, fourni la décomposition. Toutefois, cet algorithme n'est pas utilisable en pratique [75]. Aussi, nous avons proposé une caractérisation structurelle des graphes d'indice de traitement 2 (théorème 64). La figure 5.10 représente un graphe d'indice de traitement 2. A partir de cette caractérisation, nous avons pu proposer un algorithme pratique de reconnaissance de ces graphes en temps linéaire (en le nombre de sommets et d'arêtes).

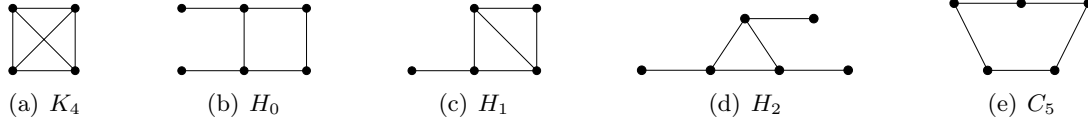


FIG. 5.8 – Ensemble M_1 d'obstructions pour les graphes d'indice de traitement 2.

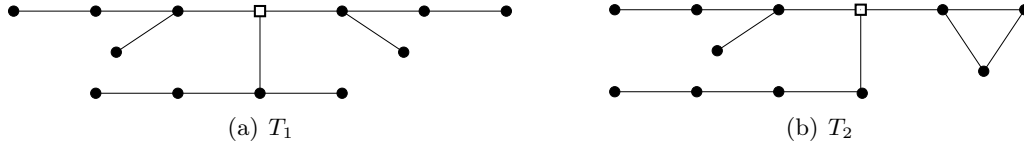


FIG. 5.9 – T_1 et T_2 sont 2 des 10 obstructions non isomorphes de l'ensemble M_2 pour les graphes d'indice de traitement 2. Ils sont construits en reliant 3 copies de graphes choisis dans $\{P_4, C_3\}$ par une arête avec un nouveau sommet \square . C'est la construction classique dite "en Y".

Nous avons donc montré que :

- $\text{pn}(G) = 0 \Leftrightarrow G$ est un ensemble indépendant ;
- $\text{pn}(G) = 1 \Leftrightarrow G$ est une étoile ou une collection d'étoiles ;
- $\text{pn}(G) = 2 \Leftrightarrow G$ satisfait les conditions du théorème 64, ce qui peut-être testé en temps linéaire.

Théorème 64 ([S50]) *Pour tout graphe connexe $G = (V, E)$, les assertions suivantes sont équivalentes.*

- (a) $\text{pn}(G) \leq 2$;
- (b) *Aucun mineur de G n'est dans $M_1 \cup M_2$ (voir figures 5.8 et 5.9) ;*
- (c) *G est tel que :*
 - $V = U \cup Z \cup T$ avec $U = \{u_1, \dots, u_r\}$ et $Z = \{z_i^j : 1 \leq i \leq r \text{ et } 1 \leq j \leq k_i\}$, où k_1, \dots, k_r sont des entiers positifs ;
 - *chaque composante connexe de $G[T]$ est une étoile S_i^ℓ , pour $\ell \in \mathbb{N}$ et $i \in \{1, 2, \dots, r\}$, qui est attachée dans G au sommet $u_i \in U$;*
 - *chaque sommet $z_i^j \in Z$ a degré 2 dans G , et ses 2 voisins sont u_i et u_{i+1} ; et*
 - $N(u_i) \cap U \subseteq \{u_{i-1}, u_{i+1}\}$, et lorsque $k_i = 0$ alors $u_{i+1} \in N(u_i)$.

Le théorème 64 est similaire à celui proposé par Megiddo *et al.* [65] pour caractériser les graphes d'indice d'échappement 3 et donc de largeur de chemin égale à 2. Notons que l'ensemble des mineurs exclus utilisé pour caractériser les graphes de largeur de chemin 2 ne contient que 2 graphes (le triangle et la pieuvre à 3 branches de longueurs 2) alors que nous en utilisons 15 dans le théorème 64. Ceci s'explique par le fait que des graphes de largeur de chemin égale à 3 peuvent avoir indice de traitement 2 (corollaire 40) et qu'il faut 110 mineurs exclus pour caractériser les graphes de largeur de chemin 3 [69].

Nous avons également commencé à identifier les mineurs exclus pour caractériser les graphes non-orientés d'indice de traitement 3 [S57]. Pour l'instant, nous avons identifié 185 266 mineurs exclus (dont 183 762 sont issus des constructions $\Delta+Y$) et il nous reste à démontrer que cet ensemble est clos. Cette valeur est sûrement la dernière atteignable par cette méthode, le nombre de mineurs exclus pour l'indice de traitement p étant supérieur au nombre de mineurs exclus pour la largeur de chemin p , celui si étant estimé à plus de 122 millions pour 4.

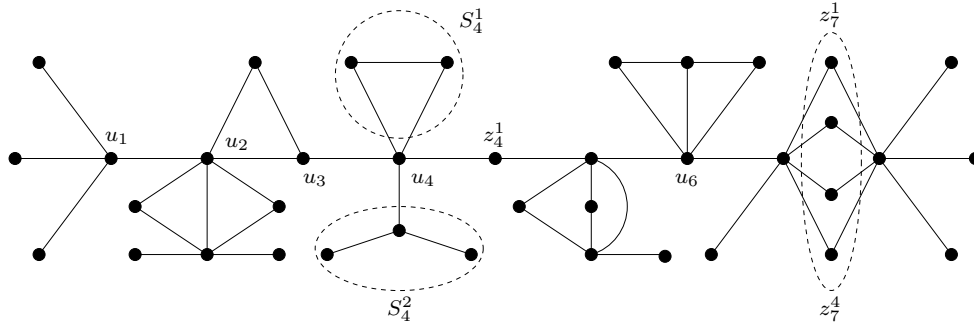


FIG. 5.10 – Exemple de graphe d'indice de traitement 2.

5.5.3 Graphes orientés d'indice de traitement inférieur à 2

Nous avons caractérisé les graphes orientés (fortement connexes) d'indice de traitement inférieur à 2 [S50]. Comme nous l'avons dit précédemment, seul les DAG ont un indice de traitement nul. Les graphes orientés d'indice de traitement 1 sont tels que la suppression d'un sommet particulier les rend acycliques. Ce sont donc des graphes d'indice de transmission 1.

Pour caractériser les graphes orientés d'indice de traitement 2, nous avons introduit la notion de $(2, w)$ -digraphes⁴, c'est-à-dire les graphes orientés pour lesquels il existe une stratégie de traitement utilisant 2 agents et commençant par placer un agent sur le sommet w . La caractérisation de ces graphes est donnée dans la proposition 65 qui utilise la notion de voisinage stricte, SN , d'un ensemble X de sommets du graphe, telle que $\text{SN}_D^+(X) = N_D^+(X) - X$. La figure 5.10 représente l'allure générale d'un $(2, w)$ -digraphe telle que décrite dans la proposition 65, et la figure 5.11 représente un exemple de tel graphe.

Proposition 65 ([S50]) *Soient D un graphe orienté et w l'un de ses sommets. D est un $(2, w)$ -digraphe si et seulement si $D - \{w\}$ peut être partitionné en 2 sous-graphes H et H' satisfaisant les conditions suivantes :*

- (i) *il existe un sommet w' de H' tel que H' est un $(2, w')$ -digraphe ;*
- (ii) $\text{SN}_D^+(H + \{w\}) \subseteq \{w'\}$
- (iii) *soit*
 - $\text{pn}(H) = 0$; *ou*
 - $\text{pn}(H) = 1$ *et il existe un ensemble (éventuellement vide) $Y \subset V(H)$ tel que $N_D^-(\{w'\}) \cap V(H) \subseteq Y$, $\text{pn}(D[Y]) = 0$, et $(Y, V(H) \setminus Y)$ est une coupe orientée de H depuis Y vers $V(H) \setminus Y$.*

Nous avons proposé un algorithme en temps $O(n(n + m))$ pour tester si un graphe orienté est un $(2, w)$ -digraphe. Donc, tester si un graphe orienté a un indice de traitement égal à 2 prend un temps $O(n^2(n + m))$, il suffit d'appliquer l'algorithme depuis chacun des sommets du graphe.

La question de caractériser les graphes orientés d'indice de traitement 3 et plus est ouverte.

5.5.4 Arbres

De nombreux algorithmes ont été proposés pour déterminer l'indice d'échappement ou la largeur de chemin d'un arbre en temps linéaire [66, 68, 81]. Ces algorithmes sont centralisés et

⁴Cette définition s'étend naturellement aux $(p, \{w_1, w_2, \dots, w_k\})$ -digraphes qui sont les graphes orientés pour lesquels il existe une stratégie de traitement utilisant p agents et commençant par placer un agent sur chacun des sommets $\{w_1, w_2, \dots, w_k\}$.

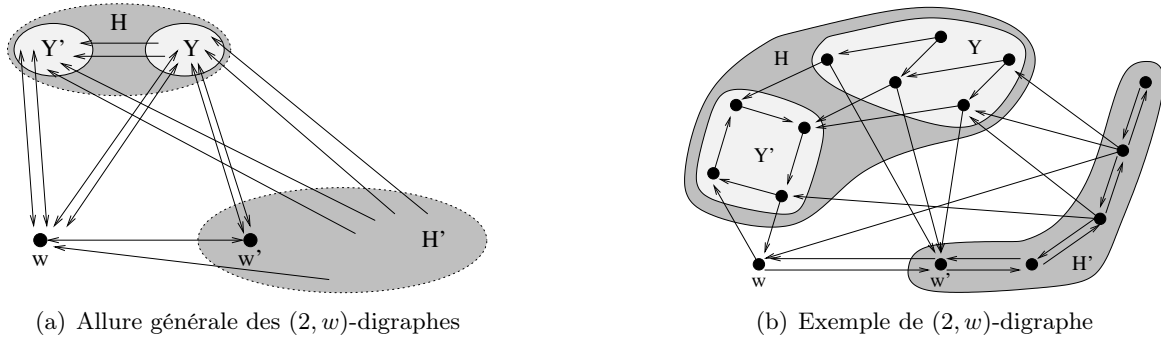


FIG. 5.11 – Allure générale des $(2, w)$ -digraphes et exemple de $(2, w)$ -digraphe.

difficiles à modifier pour calculer d'autres invariants comme l'indice de traitement. Aussi, avec Florian Huc et Dorian Mazauric, nous avons proposé un nouvel algorithme pour calculer ces invariants [S51]⁵. Notre algorithme présente les caractéristiques et avantages suivants :

- Il permet de calculer : l'indice d'échappement et donc la largeur de chemin, *l'indice d'échappement arête* (voir Parsons [61] ou Nisse [85]), et l'indice de traitement ;
- Un sommet n'a besoin de connaître que ses voisins ;
- L'algorithme est distribué ;
- Il peut-être exécuté dans un environnement asynchrone ;
- Il effectue au total $O(n \log n)$ opérations ;
- Il transmet au total n messages, chacun de taille $\log_3 n + 1$ bits ;
- En utilisant des messages de $\log_3 n + 3$ bits, il permet de maintenir ses invariants lors de l'ajout ou de la suppression d'une arête de l'arbre ;
- En utilisant des messages de au plus $2 \log_3 n + 3$ bits, il permet d'ignorer la taille de l'arbre.

D'autre part, il a été montré par Kirousis et Papadimitriou [64] que $ns(G) - 1 \leq es(G) \leq ns(G) + 1$, où $es(G)$ est l'indice d'échappement arête de G . Nous avons affiné ce résultat dans le cas des arbres en montrant :

Lemme 66 ([S51]) *Soit T un arbre, nous avons $ns(T) - 1 \leq es(T) \leq ns(T)$.*

Par ailleurs, avec Omid Amini et Nicolas Nisse, nous avons commencé l'étude des jeux de capture non-déterministes (*non-déterministique graph searching*) dans les arbres [S48]. Ces jeux, introduit par Fomin et al. [94], font le lien entre la largeur de chemin et la largeur d'arborescence. Il sera intéressant d'étendre notre algorithme distribué à ces jeux.

5.5.5 Graphes planaires extérieurs

Les graphes planaires extérieurs (*outerplanar graphs*) sont des graphes planaires dont tous les sommets sont adjacents à la face extérieure. Nous nous restreignons ici aux graphes planaires extérieurs bi-connexes, qui sont ceux qui ont été les plus étudiés [72, 73, 78]. Ces graphes ont une largeur d'arborescence (*treewidth*, tw) de 2. Aussi, il est théoriquement possible de calculer la largeur de chemin de ces graphes en temps polynomial [71], mais aucun algorithme pratique n'est connu à ce jour, le premier pas de l'algorithme de Boadlander et Kloks [71] (pour les graphes

⁵L'algorithme que nous avons proposé dans [Cn44] pour déterminer l'indice de traitement d'un arbre est malheureusement faux.

de largeur d'arborescence borné) étant en $O(n^{11})$. Ceci a motivé une série de travaux proposant des algorithmes approchés.

Govindan et al. [74] ont proposé un algorithme 3-approché en temps $O(n \log n)$. Ensuite, Bodlaender and Fomin [77] ont proposé un algorithme linéaire 2-approché en utilisant le dual géométrique des graphes planaires extérieurs. Le dual géométrique G^* d'un graphe planaire G a un sommet par face de G (dont la face extérieure) et une arête entre 2 sommets si les faces correspondantes dans G partagent une arête. Le dual géométrique d'un graphe planaire extérieur auquel on retire le sommet de la face extérieure (dual faible) est un arbre. Plus précisément, ils ont montré que $\text{pw}(G) \leq 2 \text{pw}(G^*) + 2$. Puis, avec Florian Huc et Jean-Sébastien Séréni, nous avons affiné ce résultat en montrant :

Théorème 67 ([A9]) *Pour tout graphe planaire extérieur bi-connexe G , nous avons :*

$$\text{pw}(G^*) \leq \text{pw}(G) \leq 2 \text{pw}(G^*) - 1$$

Théorème 68 ([A9]) *Pour tout $p \geq 1$ et pour tout $k \in \{1, 2, \dots, p + 1\}$, il existe un graphe planaire extérieur bi-connexe G tel que $\text{pw}(G) = p + k$ et $\text{pw}(G^*) = p + 1$.*

Nous avons de plus montré le théorème 68 qui implique que les bornes du théorème 67 sont serrées. Ce faisant, nous avons infirmé plusieurs conjectures de Bodlaender et Fomin [77, 80] (se reporter à [A9] pour plus de détails).

La question de proposer un algorithme performant pour calculer la largeur de chemin d'un graphe planaire extérieur est toujours ouverte. Les travaux les plus avancés sur cette question sont dus à Markov [90].

Une question plus générale est d'encadrer l'écart entre la largeur de chemin d'un graphe planaire et celle de son dual. Fomin et Thilikos [84] ont montré que pour tout graphe planaire 3-connexe G , $\text{pw}(G) \leq 6 \text{pw}(G^*)$. Puis, Amini, Huc et Pérennes [93] ont réduit ce facteur à 3 (et même à 2 si G est 4-connexe). Enfin, Fomin et Thilikos [84] ont montré que pour tout graphe planaire 2-connexe ce facteur est au moins 2.

Ces résultats plus le fait qu'il ne semble pas y avoir d'influence de la largeur d'arborescence sur ce facteur⁶ m'amènent à poser la conjecture suivante :

Conjecture 1 *Pour tout graphe planaire 2-connexe G , $\text{pw}(G) \leq 2 \text{pw}(G^*)$.*

5.6 Conclusion

Dans ce chapitre, j'ai essayé de donner un aperçu de l'ensemble des travaux que nous avons effectués à partir d'une simple question sur le problème de la reconfiguration du routage : comment passer concrètement d'un routage à un autre ?

Pour modéliser ce problème, nous avons introduit un nouveau jeu : la p -stratégie de traitement. Ceci nous a permis d'apporter des outils algorithmiques et de théorie des graphes pour aborder le problème de reconfiguration. En particulier, nous avons utilisé des résultats de la théorie des mineurs de graphes [63] et des techniques algorithmiques issues des jeux de capture [61, 85, 89].

A partir de cette modélisation nous avons exprimé plusieurs problèmes d'optimisation liés aux métriques présentées dans les sections 5.3 et 5.4. Entre autre, nous avons cherché à :

⁶Une simple modification des constructions utilisées dans [A9, 84] permet d'augmenter la largeur d'arborescence sans influencer sur l'écart entre la largeur de chemin du graphe et celle de son dual.

- Optimiser le nombre total (MFVS) ou simultané (pn) de connexions interrompues au cours de la stratégie, et les versions pondérées de ses métriques ;
- Mesurer l’impact du nombre d’interruptions simultanées autorisées sur le nombre total d’interruptions et réciproquement ;
- Comprendre l’impact de différents types de contrats de services :
 - Interdisant toute interruption de service ;
 - Avec pénalité fixe par interruption ;
 - Avec pénalité par durée d’interruption ;
- Contrôler divers paramètres dont le nombre d’étapes d’une stratégie.

Nos principaux résultats portent sur l’indice de traitement d’un graphe. Nous avons montré la complexité et l’inapproximabilité du problème de déterminer l’indice de traitement, proposé un algorithme heuristique, caractérisé les graphes orientés et non-orientés d’indice de traitement au plus 2, et étudier les similitudes et les différences avec l’indice d’échappement et donc les jeux de capture dans les graphes. Nous avons aussi proposé un algorithme générique permettant de calculer plusieurs invariants dans les arbres et d’affiner les relations connues entre ces invariants dans le cas particulier des arbres.

En parallèle avec ces travaux, nous avons abordé des questions portant sur d’autres invariants. Nous avons en particulier fait un pas significatif dans la compréhension de la relation entre la largeur de chemin d’un graphe planaire et celle de son dual géométrique.

Lors de discussions avec des collègues d’entreprises comme Alcatel-Lucent, British Telecom, Deutsche Telekom, France Telecom ou encore Nokia Siemens Networks sur le problème de la reconfiguration, j’ai réalisé que :

- Le problème de passer d’un routage à un autre intéresse les industriels qui n’ont pas à ce jour de solution pertinente ;
- Chacun d’eux propose d’optimiser une métrique différente des autres.

Ceci me conforte dans l’idée qu’il est important d’étudier de multiples métriques sur les p -stratégies de traitements, sans se soucier du “réalisme” de la métrique, afin de bien comprendre quels sont les paramètres les plus importants, ce qu’il est possible d’optimiser et quels sont les outils les plus adaptés. Aussi, je compte poursuivre mes travaux en étudiant des questions que j’ai laissées ouvertes dans ce chapitre. Je compte également aborder tout ou partie des questions suivantes :

- Q_1 Cas pratique d’une série d’opérations de maintenance. Il s’agit de déterminer l’ordre dans lequel effectuer les opérations de maintenance sur les liens du réseau en minimisant par exemple le nombre total d’interruptions ou de changements de routes (cas concret suggéré par Alcatel-Lucent) ;
- Q_2 Inclure les contraintes physiques de la fibre optique lors de chaque transition. Dans une fibre optique, la puissance d’émission de chaque longueur d’onde est ajustée en fonction des canaux utilisés dans cette fibre. Aussi, ajouter ou supprimer une connexion a un coût en terme d’énergie et surtout de temps nécessaire à l’ajustement des différents canaux. Concrètement, l’établissement d’une connexion dans le réseau peut prendre plusieurs minutes. De plus le coût du reroutage d’une connexion dépend des étapes précédentes. Nous étudions actuellement ses questions avec Dan Kilper (Bell labs, USA) ;
- Q_3 Stratégies mixtes entre utilisation des ressources disponibles (route temporaires) et interruptions. En autorisant l’utilisation des ressources disponibles dans le réseau, il est possible de réduire le nombre d’interruptions. Le graphe de dépendances évolue alors au cours du temps ;
- Q_4 Prise en compte de la protection dans les phases de reconfiguration et utilisation de ressources dédiées à la protection pour réduire le nombre d’interruptions. Il s’agit alors

d'étudier selon le modèle de protection considéré (dédiée, partagée, par segment, par p -cycle,...) l'impact sur le problème ;
 Q_5 Etude de solutions distribuées pour effectuer la reconfiguration.

Chapitre 6

Conclusion et perspectives

Les travaux que j'ai effectués au cours des dernières années ont principalement porté sur des problèmes d'optimisation qui se posent dans les réseaux optiques WDM avec des matrices de trafic statiques. Ce sont en particulier des travaux sur : le routage et de l'affectation de longueurs d'ondes ; le groupage de trafic (chapitre 3) ; et différents modèles de protection en cas de pannes simples ou multiples de nœuds ou de liens (chapitre 4). Ces travaux contribuent aux problèmes généraux du dimensionnement et de la planification de ces réseaux. Cependant, considérer des matrices de trafic statiques n'est plus suffisant pour les opérateurs des réseaux de cœur qui font face à une croissance régulière du trafic (augmentation du nombre d'utilisateurs, nouveaux usages) et doivent gérer au quotidien des variations de trafic, des pannes et des opérations de maintenance. Pour contourner les difficultés, les opérateurs de télécoms ont pendant longtemps choisi de surdimensionner les nœuds et les liens des réseaux. L'apparition d'applications très gourmandes en bande passante (réseaux pair-à-pair, vidéo à la demande) a rapidement saturé les réseaux, en particulier les capacités de traitement des routeurs, et forcé les opérateurs à augmenter plus vite que prévu l'ensemble des équipements. Aussi, les paradigmes centralisés utilisés jusqu'à présent dans la planification et la gestion des réseaux de cœur deviennent incompatible avec la taille et la dynamique des réseaux. La taille des réseaux rend difficile voire impossible le calcul de solutions optimales. Cette difficulté est renforcée par l'augmentation des événements (pannes, opérations de maintenance, variation du trafic) demandant des prises de décisions rapides.

Sur le long terme, je compte contribuer au développement d'outils algorithmiques, de théorie des graphes et de l'optimisation combinatoire qui sont indispensables pour résoudre des problèmes de télécoms. En particulier, je compte étudier des mécanismes centralisés et distribués pour résoudre efficacement les problèmes de plongement de graphes et les évolutions de ces plongements (déterminer les changements à effectuer dans le plongement et proposer des méthodes performantes pour effectuer la migration vers le nouveau plongement). Ces travaux permettront de gérer efficacement les topologies virtuelles dans les réseaux. Je compte également étudier des méthodes d'algorithmiques distribués avec informations partielles. Il s'agit alors de déterminer la quantité d'informations nécessaire à l'algorithme pour fournir des solutions à performances garanties et assurer la robustesse des solutions proposées. Ceci permettra d'accélérer les prises de décision dans les réseaux à gestion décentralisée. J'envisage aussi de contribuer à la proposition d'un modèle de graphes dynamiques et au développement des outils algorithmiques correspondant. Ceci permettrait de disposer de meilleurs outils pour aborder les problèmes de télécoms. Le modèle des graphes évolutifs [79] (aussi appelé réseaux tolérants les délais avec oracle) est une première proposition dans ce sens. Toutefois, ce modèle est déterministe (les évolutions de la topologie sont déterminées à l'avance) et donc ne permet pas de prendre en compte un

comportement aléatoire.

Par ailleurs, je n'exclue pas de considérer d'autres domaines d'applications comme par exemple les réseaux de distribution d'électricité (ou d'eau) ou les réseaux de transports.

Les activités de recherche que je vais mener dans les 4-5 prochaines années sont guidées par les collaborations et les thèses en cours, dont je cite trois exemples ci-dessous.

Algorithmes dynamiques pour le routage compact

J'ai récemment commencé à m'intéresser aux algorithmes de routage compact l'internet [219, 221]. Ces algorithmes construisent des tables de routage avec beaucoup moins d'entrées que celles utilisées par BGP (*border gate protocol*), mais au coût d'une augmentation de la longueur des chemins. Notre objectif est d'étudier de nouveaux paradigmes de routages afin de concevoir, développer et expérimenter un algorithme de routage dynamique et distribué satisfaisant pour l'internet du futur. Nous cherchons un algorithme de routage qui repousse les limitations des algorithmes actuels en termes de scalabilité des tables de routage, de dynamique de la topologie et des changements de politiques de routage. Nous souhaitons étudier des compromis entre la taille des tables de routage (scalabilité), la longueur des routes (qualité du routage) et le coût de la signalisation (réaction rapide aux pannes). Pour cela, nous envisageons d'utiliser des propriétés structurelles et statistiques de la topologie d'internet dans la conception de nouveaux algorithmes.

Ces travaux ont commencé en 2008 dans le cadre d'une collaboration avec Alcatel-Lucent Bell-labs et le LABRI. Pour le moment, nous construisons un simulateur permettant d'évaluer les performances des algorithmes de calcul de routes et de tables de routage sur des réseaux de grande taille (plus de 10^5 nœuds). Ce simulateur nous permettra de mieux comprendre l'influence de la topologie sur les algorithmes de routage.

Nous allons poursuivre ces travaux dans le cadre du STREP EULER (*Experimental Update-Less Evolutive Routing*, <http://euler-fire-project.eu>), porté par D. Papadimitriou (Alcatel-Lucent Bell labs, Antwerp, Belgique) et qui démarrera en 2010. Les partenaires sont Alcatel-Lucent Bell-labs, IBBT Ghent, UCL Louvain, UPC Barcelone, RACTI Patras, le LIP6 et l'INRIA. Ce projet propose des travaux sur les algorithmes de routage, l'identification de propriétés structurelles et statistiques de topologies, la génération de topologies vérifiant des propriétés données, l'analyse des algorithmes avec des outils de théorie des jeux, la validation par simulation puis par expérimentation des algorithmes proposés.

Dans ce projet, je suis *technical assistant*, responsable d'un workpackage et de 3 taches. Mes objectifs personnels dans ce projet sont de compléter mes connaissances en algorithmique distribuée, de travailler sur des méthodes de tests de propriétés et sur la génération de topologies.

Gestion décentralisée de réseaux multi-couches

Ces questions sont au centre de la thèse de P. Giabbanelli (depuis janv. 2010) avec qui je travaille dans le cadre de l'AdR HiMa du laboratoire commun INRIA/Bell-labs. Plus précisément, nous étudions des réseaux multi-couches de type IP sur WDM où la topologie du réseau IP est la topologie virtuelle que nous routons dans le réseau WDM sous-jacent. Notre objectif est de faire évoluer dynamiquement et de façon décentralisée la topologie virtuelle en fonction des variations de trafic. Nous devons décider quels liens ajouter ou supprimer de la topologie virtuelle afin de simplifier le routage des paquets et de limiter la quantité de traitement à effectuer aux nœuds intermédiaires. Tout ceci est sujet à des contraintes comme le degré maximum d'un nœud de la topologie virtuelle (nombre d'interfaces à installer dans un routeur), la capacité des liens, l'existence d'un routage de la topologie virtuelle dans la couche physique. Il faut de plus

prendre en compte des contraintes de QoS dont la tolérance aux pannes et différents modèles de groupage de trafic.

Nous avons de plus besoin de méthodes performantes pour faire évoluer le plongement de la topologie virtuelle dans la topologie physique et donc reconfigurer le routage. Les travaux que j'effectue en collaboration avec D. Kilper (Bell labs, USA) sur la prise en compte des contraintes physiques dans la reconfiguration seront donc utiles dans ce contexte. Nous travaillons pour le moment sur des approches centralisées (chapitre 5), mais dans le futur, nous devons fournir des solutions distribuées. Notons que le temps nécessaire pour établir un chemin optique dans le réseau va de quelques secondes à quelques minutes. En effet, lorsque l'on utilise une nouvelle longueur d'onde dans une fibre, il faut modifier légèrement le réglage (puissance d'émission, longueur d'onde) de toute celles qui sont déjà utilisées et ceci peut se reporter dans tout le réseau. Cette durée permet de déterminer la période de temps minimale entre deux modifications de la topologie virtuelle ou de son plongement, et aussi le temps imparti pour déterminer la meilleure stratégie.

Réseaux efficaces en énergie, *green networking*

Alcatel-Lucent vient de lancer l'initiative GreenTouch (<http://www.greentouch.org>) qui vise à fédérer les efforts de recherches autour d'un objectif : réduire d'un facteur 1000 la consommation électrique des réseaux. Pour atteindre cet objectif, il faut travailler à la fois sur la conception des composants du réseaux (technologie physique des routeurs) et sur leurs fonctionnalités (suspend/wake-on-line, down clocking, . . .), mais aussi sur la conception et la gestion des réseaux ainsi que sur les paradigmes de communications (transmission immédiate *vs* différée, push d'informations, . . .).

L'ANR DIMAGREEN rentre dans ce cadre. En effet, nous étudions l'impact du dimensionnement du réseau et des algorithmes de routage sur la consommation globale d'énergie du réseau. Des études récentes [222] montrent que la charge des routeurs actuels a peu d'influence sur leur consommation d'énergie, et que la consommation des réseaux est corrélée au nombre d'éléments actifs. L'objectif est donc de minimiser le nombre d'éléments actifs à chaque instant dans le réseau. A long terme, les équipementiers devront concevoir des routeurs dont la consommation est plus directement reliée à leur charge.

Les travaux que nous effectuons avec la PME 3Roam (<http://www.3roam.com>) rentrent également dans ce cadre. En effet, nous cherchons non seulement à dimensionner et gérer au mieux le réseau pour satisfaire des contraintes de QoS (délais, tolérance aux pannes, . . .), mais aussi à minimiser sa consommation globale. Nous avons déjà fait des propositions dans ce sens (gestion efficace en énergie des liens), mais nous comptons aller beaucoup plus loin. Par exemple, les routeurs actuellement conçus par 3Roam ne gèrent que deux liens bidirectionnels et il peut donc y avoir plusieurs routeurs dans un même nœud du réseau. Aussi, nous étudions la possibilité d'éteindre certains routeurs (et les liens correspondants) quand le trafic est faible (tout en respectant la QoS). Selon le résultat de nos études, la PME cherchera les solutions d'ingénierie les plus adaptées pour ajouter des fonctionnalités type suspend/wake-on-line.

Ces objectifs nous amènent à étudier (thèses de N. Nepomuceno et I. Tahiri) des problèmes de dimensionnement, et à concevoir des mécanismes permettant de faire évoluer le routage des connexions ou des paquets avec un minimum de perturbations pour les utilisateurs.

Merci de votre attention !

Bibliographie

Bibliographie personnelle

Articles disponibles sur: <http://www-sop.inria.fr/members/David.Coudert/Biblio/index.php>

— Thèse —

- [T1] D. Coudert. *Algorithmique et optimisation de réseaux de communications optiques*. PhD thesis, Université de Nice Sophia-Antipolis (UNSA), Décembre 2001.

— Revues internationales avec comité de lecture —

- [A2] D. Coudert, A. Ferreira, and X. Muñoz. A multihop-multi-ops optical interconnection network. *IEEE/OSA Journal of Lightwave Technology*, 18(12) :2076–2085, December 2000.
- [A3] D. Coudert, A. Ferreira, and X. Muñoz. Topologies for Optical Interconnection Networks Based on the Optical Transpose Interconnection System. *OSA Applied Optics – Information Processing*, 39(17) :2965–2974, June 2000.
- [A4] D. Coudert, A. Ferreira, and S. Pérennes. Isomorphisms of the de bruijn digraph and free-space optical networks. *Networks*, 40(3) :155–164, October 2002.
- [A5] J-C. Bermond, D. Coudert, and M-L. Yu. On DRC-Covering of K_n by cycles. *Journal of Combinatorial Designs*, 11(2) :100–112, 2003.
- [A6] J-C. Bermond, C. Colbourn, D. Coudert, G. Ge, A. Ling, and X. Muñoz. Traffic grooming in unidirectional WDM rings with grooming ratio $C=6$. *SIAM Journal on Discrete Mathematics*, 19(2) :523–542, 2005.
- [A7] J-C. Bermond, L. Braud, and D. Coudert. Traffic grooming on the path. *Theoretical Computer Science*, 384(2-3) :139–151, October 2007.
- [A8] D. Coudert, P. Datta, S. Pérennes, H. Rivano, and M-E. Voge. Shared risk resource group : Complexity and approximability issues. *Parallel Processing Letters*, 17(2) :169–184, June 2007.
- [A9] D. Coudert, F. Huc, and J.-S. Sereni. Pathwidth of outerplanar graphs. *Journal of Graph Theory*, 55(1) :27–41, May 2007.
- [A10] J-C. Bermond, D. Coudert, and B. Lévêque. Approximations for all-to-all uniform traffic grooming on unidirectional ring. *Journal of Interconnection Networks (JOIN)*, 9(4) :471–486, December 2008.
- [A11] D. Coudert, N. Nepomuceno, and H. Rivano. Power-efficient radio configuration in fixed broadband wireless networks. *Computer Communications*, to appear.

— Chapitres de livres —

- [B12] J-C. Bermond and D. Coudert. *Handbook of Combinatorial Designs (2nd edition)*, volume 42 of *Discrete mathematics and Applications*, chapter VI.27, Grooming, pages 494–496. Chapman & Hall- CRC Press, C.J. Colbourn and J.H. Dinitz editors, November 2006.
- [B13] T. Cinkler, D. Coudert, M. Flammini, G. Monaco, L. Moscardelli, X. Muñoz, I. Sau, M. Shalom, and S. Zaks. *Graphs and Algorithms in Communication Networks : Studies in Broadband, Optical, Wireless, and Ad Hoc Networks*, volume XXVII of *EATCS Texts in Theoretical Computer Science*, chapter 2, Traffic Grooming : Combinatorial Results and Practical Resolutions, pages 63–94. Springer, A. Koster and X. Muñoz edition, November 2009.

- [B14] D. Coudert and X. Muñoz. *Recent Research Developments in Optics*, volume 3, chapter 37, Graph Theory and Traffic Grooming in WDM Rings, pages 759–778. Research Signpost. Kerala, India, S.G. Pandalai editor, 2003. ISBN : 81-271-0028-5.

— Conférences internationales avec comité de lecture et actes —

- [Ci15] D. Coudert, A. Ferreira, and X. Muñoz. Multiprocessor Architectures Using Multi-hops Multi-OPS Lightwave Networks and Distributed Control. In *IEEE International Parallel Processing Symposium*, pages 151–155. IEEE Press, 1998.
- [Ci16] D. Coudert, A. Ferreira, and X. Muñoz. OTIS-Based Multi-Hop Multi-OPS Lightwave Networks. In J. Rolim et al., editor, *Proceedings of the Workshop on Optics and Computer Science 1999*, volume 1586 of *Lecture Notes in Computer Science*, pages 897–910. Springer Verlag, April 1999.
- [Ci17] D. Coudert, A. Ferreira, and S. Perennes. De Bruijn Isomorphisms and Free Space Optical Networks. In *IEEE International Parallel and Distributed Processing Symposium*, pages 769–774. IEEE Press, 2000.
- [Ci18] J-C. Bermond, L. Chacon, D. Coudert, and F. Tillerot. Cycle Covering. In *International Colloquium on Structural Information and Communication Complexity – SIROCCO*, Proceedings in Informatics 11, pages 21–34, Vall de Nuria, Spain, 27-29 June 2001. Carleton Scientific.
- [Ci19] D. Coudert and X. Muñoz. How Graph Theory can help Communications Engineering. In D.K.Gautam, editor, *Broad band optical fiber communications technology – BBOFCT*, pages 47–61, Jalgaon, India, December 2001. Nirtali Prakashan. Invited paper.
- [Ci20] D. Coudert and H. Rivano. Lightpath assignment for multifibers wdm networks with wavelength translators. In *IEEE Global Telecommunications Conference – GLOBECOM*, volume 3, pages 2686–2690, November 2002. OPNT-01-5.
- [Ci21] J-C. Bermond and D. Coudert. Traffic grooming in unidirectional WDM ring networks using design theory. In *IEEE ICC*, volume 2, pages 1402–1406, Anchorage, Alaska, May 2003. ON07-3.
- [Ci22] J-C. Bermond, D. Coudert, and X. Muñoz. Traffic grooming in unidirectional WDM ring networks : The all-to-all unitary case. In *The 7th IFIP Working Conference on Optical Network Design & Modelling – ONDM*, pages 1135–1153, Budapest, Hongrie, 2003.
- [Ci23] M. Bouklit, D. Coudert, J-F. Lalande, C. Paul, and H. Rivano. Approximate multicommodity flow for WDM networks design. In J. Sibeyn, editor, *10th International Colloquium on Structural Information and Communication Complexity – SIROCCO*, number 17 in Proceedings in Informatics, pages 43–56, Umea, Sweden, 2003. Carleton Scientific.
- [Ci24] D. Coudert, H. Rivano, and X. Roche. A combinatorial approximation algorithm for the multicommodity flow problem. In K. Jansen and R. Solis-Oba, editors, *WAOA 03*, number 2909 in *Lecture Notes in Computer Science*, pages 256–259, Budapest, Hungary, September 2003. Springer-Verlag.
- [Ci25] J-C. Bermond, L. Braud, and D. Coudert. Traffic grooming on the path. In A. Pelc and M. Raynal, editors, *12th International Colloquium on Structural Information and Communication Complexity – SIROCCO*, volume 3499 of *Lecture Notes in Computer Science*, pages 34–48, Le Mont Saint-Michel, France, May 2005. Springer.
- [Ci26] J-C. Bermond, M. Cosnard, D. Coudert, and S. Perennes. Optimal solution of the maximum all request path grooming problem. In *Advanced International Conference on Telecommunications (AICT)*. IEEE, 2006.
- [Ci27] J-C. Bermond, D. Coudert, X. Muñoz, and I. Sau Valls. Traffic grooming in bidirectional WDM ring networks. In *IEEE-LEOS ICTON / COST 293 GRAAL*, volume 3, pages 19–22, June 2006.
- [Ci28] D. Coudert, S. Perennes, H. Rivano, and M-E. Voge. Shared risk resource groups and survivability in multilayer networks. In *IEEE-LEOS ICTON / COST 293 GRAAL*, volume 3, pages 235–238, June 2006. Invited Paper.
- [Ci29] D. Coudert, F. Huc, F. Peix, and M-E. Voge. Reliability of connections in multilayer networks under shared risk groups and costs constraints. In *IEEE International Conference on Communication (ICC)*, number ON01-6, pages 5170 – 5174, Beijing, China, May 2008.

- [Ci30] J-C. Bermond, D. Coudert, J. Moulierac, S. Perennes, H. Rivano, I. Sau, and F. Solano Donado. MPLS label stacking on the line network. In L. Fratta et al., editor, *IFIP Networking*, volume 5550 of *Lecture Notes in Computer Science*, pages 809–820, Aachen, Germany, May 2009. Springer.
- [Ci31] J-C. Bermond, D. Coudert, J. Moulierac, S. Perennes, I. Sau, and F. Solano Donado. GMPLS routing strategies based on the design of hypergraph layouts. In *16th International Colloquium on Structural Information and Communication Complexity – SIROCCO*, volume 5869 of *Lecture Notes in Computer Science*, pages 57–71. Springer, May 2009.
- [Ci32] D. Coudert, F. Giroire, and I. Sau. Edge-simple circuits through 10 ordered vertices in square grids. In J. Kratochvíl and M. Miller, editors, *20th International Workshop on Combinatorial Algorithms – IWOCA*, volume 5874 of *Lecture Notes in Computer Science*, pages 134–145, Hradec nad Moravicí, Czech Republic, June 2009. Springer.
- [Ci33] D. Coudert, F. Huc, D. Mazauric, N. Nisse, and J-S. Sereni. Reconfiguration of the routing in WDM networks with two classes of services. In *13th Conference on Optical Network Design and Modeling (ONDM)*, Braunschweig, Germany, February 2009. IEEE.
- [Ci34] D. Coudert, D. Mazauric, and N. Nisse. On rerouting connection requests in networks with shared bandwidth. In A. Koster and V. Lozin, editors, *DIMAP workshop on Algorithmic Graph Theory (AGT09)*, volume 32 of *Electronic Notes in Discrete Mathematics*, pages 109–116, Warwick, UK, March 2009. Elsevier.
- [Ci35] D. Coudert, N. Nepomuceno, and H. Rivano. Minimizing energy consumption by Power-Efficient radio configuration in fixed broadband wireless networks. In *First IEEE WoWMoM Workshop on Hot Topics in Mesh Networking (HotMESH)*, Kos, Greece, June 2009. IEEE.

— **Conférences internationales avec comité de lecture et actes (articles courts)** —

- [Cis36] J-C. Bermond, L. Chacon, D. Coudert, and F. Tillerot. A Note on Cycle Covering. In *ACM Symposium on Parallel Algorithms and Architectures – SPAA*, pages 310–311, Crete island, Greece, 4-6 July 2001.
- [Cis37] D. Coudert, F. Huc, and D. Mazauric. Computing and updating the process number in trees (short paper). In T. Baker and S. Tixeuil, editors, *12th International Conference On Principles Of Distributed Systems (OPODIS)*, volume 5401 of *Lecture Notes in Computer Science*, pages 546–550, Luxor, Egypt, December 2008. Springer.
- [Cis38] D. Coudert, F. Huc, and D. Mazauric. A distributed algorithm for computing and updating the process number of a forest. In G. Taubenfeld, editor, *22nd International Symposium on Distributed Computing (DISC)*, volume 5218 of *Lecture Notes in Computer Science*, pages 500–501, Arcachon, France, September 2008. Springer. brief announcement.

— **Conférences nationales avec comité de lecture et actes** —

- [Cn39] D. Coudert, A. Ferreira, and S. Perennes. Theoretical Aspects of the Optical Transpose Interconnecting System Architecture. In *1ère Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'99)*, pages 101–106, Roscoff, France, May 1999.
- [Cn40] J-C. Bermond, L. Chacon, D. Coudert, and F. Tillerot. Conception d'un réseau WDM protégé par sous-réseaux. In *2ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'00)*, pages 83–88, La Rochelle, France, May 2000.
- [Cn41] D. Coudert. Chemins disjoints de poids minimum pour la sécurisation de réseaux de télécommunications. In *3ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'01)*, pages 47–53, St-Jean de Luz, France, May 2001.
- [Cn42] D. Coudert and H. Rivano. Routage optique dans les réseaux WDM multifibres avec conversion partielle. In *4ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'02)*, pages 17–24, Mèze, France, May 2002.
- [Cn43] M. Bouklit, D. Coudert, J-F. Lalande, and H. Rivano. Approximation combinatoire de multiflot fractionnaire : améliorations. In *5ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'03)*, Banyuls-sur-mer, France, May 2003.

- [Cn44] D. Coudert, S. Perennes, Q-C. Pham, and J-S. Sereni. Rerouting requests in wdm networks. In *7ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (Algo-Tel'05)*, pages 17–20, Presqu'île de Giens, France, May 2005. The algorithm for trees (section 3) proposed in this paper is false. A correct one is given in [CHM08].
- [Cn45] D. Coudert, F. Huc, and D. Mazauric. Algorithme générique pour les jeux de capture dans les arbres. In D. Simplot-Ryl and S. Tixeuil, editors, *10ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'08)*, pages 37–40, Saint-Malo, May 2008.
- [Cn46] D. Coudert, F. Huc, D. Mazauric, N. Nisse, and J-S. Sereni. Reconfiguration dans les réseaux optiques. In A. Chaintreau and C. Magnien, editors, *11ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'09)*, pages 25–28, Carry le Rouet, June 2009.
- [Cn47] D. Coudert, N. Nepomuceno, and H. Rivano. Joint optimization of routing and radio configuration in fixed wireless networks. In A. Chaintreau and C. Magnien, editors, *11ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'09)*, pages 49–52, Carry le Rouet, June 2009.

— Soumis ou en préparation —

- [S48] O. Amini, D. Coudert, and N. Nisse. Some results on non-deterministic graph searching in trees. Technical Report inria-00174965, HAL, September 2007. In preparation.
- [S49] D. Coudert, S. Perennes, H. Rivano, and M-E. Voge. Shared risk resource groups and colored graph : Polynomial cases and transformation issues. Technical Report inria-00175143, HAL, September 2007. In preparation.
- [S50] D. Coudert and J-S. Sereni. Characterization of graphs and digraphs with small process number. Research Report 6285, INRIA, September 2007. Submitted to a journal.
- [S51] D. Coudert, F. Huc, and D. Mazauric. A distributed algorithm for computing and updating the process number of a forest. Research Report 6560, INRIA, June 2008. Submitted to a journal.
- [S52] D. Coudert and D. Mazauric. Network reconfiguration using cops-and-robber games. Technical Report inria-00315568, INRIA, August 2008.
- [S53] J-C. Bermond, D. Coudert, J. Moulhierac, S. Pérennes, I. Sau, and F. Solano Donado. GMPLS Label Space Minimization through Hypergraph Layouts. Research Report RR-7071, INRIA, oct 2009. Submitted to a journal.
- [S54] J-C. Bermond, D. Coudert, and J. Peters. Online distributed traffic grooming on path networks. Technical Report RR-6833, INRIA, February 2009. In preparation.
- [S55] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse. Tradeoffs when optimizing Lightpaths Reconfiguration in WDM networks. Technical Report RR-7047, INRIA, September 2009. Submitted.
- [S56] D. Coudert, F. Giroire, and I. Sau. On the existence of circuits visiting k ordered vertices in graphs. In preparation, 2009.
- [S57] D. Coudert and J.-S. Sereni. Obstruction set for graphs with process number three. In preparation.

Bibliographie générale

— Algorithmique —

- [58] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, Heidelberg, 1998.
- [59] M.R. Garey and D.S. Johnson. *Computers and intractability : A guide to the theory of NP-completeness*. W.H. Freeman, San Francisco, 1979.
- [60] D.S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

- [61] T. D. Parsons. Pursuit-evasion in a graph. In *Theory and applications of graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. Springer, Berlin, 1978.
- [62] M.R. Garey, D.S. Johnson, G.L. Miller, and C.H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM J. Algebraic Discrete Methods*, 1(2) :216–227, 1980.
- [63] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *J. Combin. Theory Ser. B*, 35(1) :39–61, 1983.
- [64] M. Kirousis and C.H. Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(2) :205–218, 1986.
- [65] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. Assoc. Comput. Mach.*, 35(1) :18–44, 1988.
- [66] P. Scheffler. A linear algorithm for the pathwidth of trees. In R. Henn R. Bodendiek, editor, *Topics in Combinatorics and Graph Theory*, pages 613–620. Physica-Verlag Heidelberg, 1990.
- [67] N. G. Kinnersley. The vertex separation number of a graph equals its pathwidth. *Inform. Process. Lett.*, 42(6) :345–350, 1992.
- [68] J.A. Ellis, I.H. Sudborough, and J.S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1) :50–79, 1994.
- [69] N. G. Kinnersley and M. A. Langston. Obstruction set isolation for the gate matrix layout problem. *Discrete Applied Mathematics*, 54(2-3) :169–213, 1994.
- [70] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6) :1305–1317, 1996.
- [71] H. L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21(2) :358–402, 1996.
- [72] B. Reed. Treewidth and tangles : an new connectivity measure and some applications. In R. A. Bayley, editor, *Surveys in Combinatorics*, pages 87–162. Cambridge University Press, 1997.
- [73] H. L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2) :1–45, 1998.
- [74] R. Govindan, M. A. Langston, and X. Yan. Approximating the pathwidth of outerplanar graphs. *Inform. Process. Lett.*, 68(1) :17–23, 1998.
- [75] H. Röhrig. *Tree decomposition : A feasibility study*. Master’s thesis, Max-Planck-Institut für Informatik, Germany, 1998.
- [76] U. Feige, D. Peleg, and G. Kortsarz. The Dense k -Subgraph Problem. *Algorithmica*, 29(3) :410–421, 2001.
- [77] H. L. Bodlaender and F. V. Fomin. Approximation of pathwidth of outerplanar graphs. *J. Algorithms*, 43(2) :190–200, 2002.
- [78] J. Díaz, J. Petit, and M. Serna. A survey on graph layout problems. *ACM Computing Surveys*, 34(3) :313–356, 2002.
- [79] B. Bui-Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2) :267–285, April 2003.
- [80] F. V. Fomin. Pathwidth of planar and line graphs. *Graphs and Combinatorics*, 19(1) :91–99, 2003.
- [81] K. Skodinis. Construction of linear tree-layouts which are optimal with respect to vertex separation in linear time. *Journal of Algorithms*, 47(1) :40–59, 2003.
- [82] R. G. Downey and C. McCartin. Online problems, pathwidth, and persistence. In R. G. Downey, M. R. Fellows, and F. K. H. A. Dehne, editors, *International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 3162 of *Lecture Notes in Computer Science*, pages 13–24, Bergen, Norway, September 2004. Springer.
- [83] J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2) :161–172, 2006.

- [84] F. V. Fomin and D. M. Thilikos. On self duality of pathwidth in polyhedral graph embeddings. *Journal of Graph Theory*, 55(1) :42–54, 2007.
- [85] N. Nisse. *Les jeux des gendarmes et du voleur dans les graphes. Mineurs de graphes, stratégies connexes et approches distribuées*. PhD thesis, Université Paris-Sud 11, France, July 2007.
- [86] B. Yang and Y. Cao. Monotonicity of strong searching on digraphs. *Journal of Combinatorial Optimization*, 14(4) :411–425, 2007.
- [87] O. Amini, D. Peleg, S. Perennes, I. Sau, and S. Saurabh. Degree-constrained subgraph problems : Hardness and approximation results. In E. Bampis and M. Skutella, editors, *6th Workshop on Approximation and Online Algorithms (WAOA)*, volume 5426 of *Lecture Note in Computer Science*, pages 29–42, Karlsruhe, Germany, September 2008. Springer.
- [88] O. Amini, I. Sau, and S. Saurabh. Parameterized complexity of the smallest degree-constrained subgraph problem. In *International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 5008 of *Lecture Notes in Computer Science*, pages 13–29, Victoria, Canada, May 2008.
- [89] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3) :236–245, 2008.
- [90] M. Markov. On the vertex separation of maximal outerplanar graphs. *Serdica Journal of Computing*, 2(3) :207–238, 2008.
- [91] B. Yang and Y. Cao. Digraph searching, directed vertex separation and directed pathwidth. *Discrete Applied Mathematics*, 156(10) :1822–1837, 2008.
- [92] B. Yang and Y. Cao. Monotonicity in digraph search problems. *Theoretical Computer Science*, 407(1-3) :532–544, 2008.
- [93] O. Amini, F. Huc, and S. Pérennes. On the pathwidth of planar graphs. *SIAM Journal of Discrete Mathematics*, 23(3) :1311–1316, 2009.
- [94] F. V. Fomin, P. Fraigniaud, and N. Nisse. Nondeterministic graph searching : From pathwidth to treewidth. *Algorithmica*, 53(3) :358–373, 2009.
- [95] R. Mihai and I. Todinca. Pathwidth is np-hard for weighted trees. In X. Deng, J. E. Hopcroft, and J. Xue, editors, *3rd International Workshop on Frontiers in Algorithmics (FAW)*, volume 5598 of *Lecture Notes in Computer Science*, pages 181–195, Hefei, China, June 2009. Springer.
- [96] I. Sau and D. M. Thilikos. Subexponential parameterized algorithms for bounded-degree connected subgraph problems on planar graphs. In A. Koster and V. Lozin, editors, *DIMAP workshop on Algorithmic Graph Theory (AGT09)*, volume 32 of *Electronic Notes in Discrete Mathematics*, pages 59–66, Warwick, UK, March 2009. Elsevier.
- [97] B. Yang and Y. Cao. Standard directed search strategies and their applications. *Journal of Combinatorial Optimization*, 17(4) :378–399, 2009.

— **Théorie des configurations** —

- [98] J-C. Bermond and D. Sotteau. Graph decompositions and G -designs. In *5th British Combinatorial conference*, Congressus Numerantium 15 Utilitas math. Pub., pages 53–72, Aberdeen, 1975.
- [99] R.M. Wilson. Decomposition of complete graphs into subgraphs isomorphic to a given graph. *Congressus numerantium*, 15 :647–659, November 1976.
- [100] J-C. Bermond and J. Schonheim. G -decomposition of K_n where G has four vertices or less. *Discrete Mathematics*, 19 :113–120, 1977.
- [101] J-C. Bermond, C. Huang, A. Rosa, and D. Sotteau. Decomposition of complete graphs into isomorphic subgraphs with five vertices. *Ars Combinatoria*, pages 211–254, 1980.
- [102] E.R. Lamken and R.M. Wilson. Decompositions of edge-colored complete graphs. *J. Combin. Theory Ser. A*, 89 :149–200, 2000.
- [103] P. Adams, D. Bryant, and M. Buchanan. A survey on the existence of G -designs. *Journal of Combinatorial Designs*, 16(5) :373–410, 2008.
- [104] C.J. Colbourn and J.H. Dinitz, editors. *Handbook of Combinatorial Designs*. Chapman & Hall/CRC, 2 edition, 2006.

- [105] C.J. Colbourn and J.H. Dinitz, editors. *The CRC handbook of Combinatorial designs*. CRC Press, 1 edition, 1996.

— Graphes étiquetés —

- [106] R. Chang and S. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63(5) :277–282, 1997.
- [107] H.-C. Wirth. *Multicriteria Approximation of Network Design and Network Upgrade Problems*. PhD thesis, Bayerische Julius-Maximilians-Universität Würzburg, 2001.
- [108] Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84(2) :99–101, 2002.
- [109] R. Cerulli, P. Dell’Olmo, M. Gentili, and A. Raiconi. Heuristic approaches for the minimum labelling hamiltonian cycle problem. *Electronics Notes in Discrete Mathematics*, 25 :131–138, 2006.
- [110] R. Cerulli, A. Fink, M. Gentili, and S. Voß. Extensions of the minimum labelling spanning tree problem. *Journal of Telecommunications and Information Technology*, 4 :39–45, 2006.
- [111] R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *Journal of Combinatorial Optimization*, 14(4) :437–453, November 2007.
- [112] R. Hassin, J. Monnot, and D. Segev. The complexity of bottleneck labeled graph problems. In Springer Berlin / Heidelberg, editor, *33rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 07)*, volume 4769 of *Lecture Notes in Computer Science*, pages 328–340, Dornburg near Jena, Germany, June 2007.
- [113] B. Couëtoux, L. Gourvès, J. Monnot, and O. A. Telelis. On labeled traveling salesman problems. In *19th International Symposium on Algorithms and Computation (ISAAC)*, pages 776–787, Berlin, Heidelberg, 2008. Springer-Verlag.
- [114] F. Carrabs, R. Cerulli, and M. Gentili. The labeled maximum matching problem. *Comput. Oper. Res.*, 36(6) :1859–1871, 2009.
- [115] R. Cerulli, A. Fink, M. Gentili, and S. Voß. *The Next Wave on Computing, Optimization, and Decision Technologies*, chapter Metaheuristics comparison for the minimum labelling spanning tree problem, pages 93–106. Springer, New York, B.L. Golden, S. Raghavan and E.A. Wasil edition, 2005.
- [116] R. Cerulli, A. Fink, M. Gentili, and S. Voß. Extensions of the minimum labelling spanning tree problem. *Journal of Telecommunications and Information Technology*, to appear.
- [117] P. Zhang, J-Y. Cai, L-Q. Tang, and W-B. Zhao. Approximation and hardness results for label cut and related problems. *Journal of Combinatorial Optimization*, to appear.

— Groupage de Trafic —

- [118] O. Gerstel, P.J. Lin, and G. Sasaki. Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings. In *IEEE INFOCOM*, pages 94–101, 1998.
- [119] A. L. Chiu and E. H. Modiano. Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks. *IEEE/OSA Journal of Lightwave Technology*, 18(1) :2–12, January 2000.
- [120] R. Dutta and N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1(1) :73–89, January 2000.
- [121] L. Liu, X. Li, P-J. Wan, and O. Frieder. Wavelength assignment in a WDM rings to minimize SONET ADMs. In *INFOCOM’2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel-Aviv, Israel*, pages 1020–1025, 2000.
- [122] P-J. Wan, G. Calinescu, L. Liu, and O. Frieder. Grooming of arbitrary traffic in SONET/WDM BLSRs. *IEEE Journal of Selected Areas in Communications*, 18(10) :1995–2003, October 2000.
- [123] C.J. Colbourn and P-J. Wan. Minimizing drop cost for SONET/WDM networks with $\frac{1}{8}$ wavelength requirements. *Networks*, 37(2) :107–116, March 2001.
- [124] E. Modiano and P. Lin. Traffic grooming in WDM networks. *IEEE Communications Magazine*, 39(7) :124–129, July 2001.

- [125] G. Călinescu, O. Frieder, and P.-J. Wan. Minimizing electronic line terminals for automatic ring protection in general WDM optical networks. *IEEE Journal of Selected Area on Communications*, 20(1) :183–189, Jan 2002.
- [126] R. Dutta and N. Rouskas. Traffic grooming in WDM networks : Past and future. *IEEE Network*, 16(6) :46–56, November/December 2002.
- [127] T. Eilam, S. Moran, and S. Zaks. Lightpath arrangement in survivable rings to minimize the switching cost. *IEEE Journal of Selected Area on Communications*, 20(1) :172–182, Jan 2002.
- [128] J.Q. Hu. Optimal traffic grooming for wavelength-division-multiplexing rings with all-to-all uniform traffic. *OSA Journal of Optical Networks*, 1(1) :32–42, 2002.
- [129] G. Huiban, S. Pérennes, and M. Syska. Traffic grooming in WDM networks with multi-layer switches. In *IEEE International Conference on Communication (ICC)*, volume 5, pages 2896–2901, April 2002.
- [130] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks : Architectures and challenges. *SPIE Optical Networks Magazine*, pages 55–64, 2002.
- [131] J-C. Bermond and S. Ceroi. Minimizing SONET ADMs in unidirectional WDM ring with grooming ratio 3. *Networks*, 41(2) :83–86, 2003.
- [132] O. Goldschmidt, D. Hochbaum, A. Levin, and E. Olinick. The SONET edge-partition problem. *Networks*, 41(1) :13–23, 2003.
- [133] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks : Architectures and challenges. *Optical Networks Magazine*, 4(2) :55–64, March/April 2003.
- [134] J-C. Bermond, C.J. Colbourn, A. Ling, and M.-L. Yu. Grooming in unidirectional rings : $K_4 - e$ designs. *Discrete Mathematics, Lindner's Volume*, 284(1-3) :57–62, 2004.
- [135] T. Y. Chow and P. J. Lin. The Ring Grooming Problem. *Networks*, 44 :194–202, January 2004.
- [136] L. Epstein and A. Levin. Better bounds for minimizing SONET ADMs. In Springer Berlin / Heidelberg, editor, *2nd Workshop on Approximation and Online Algorithms (WAOA)*, volume 3351 of *Lecture Notes in Computer Science*, pages 281–294, Bergen, Norway, 2004.
- [137] M. Shalom and S. Zaks. A $10/7 + \epsilon$ approximation scheme for minimizing the number of ADMs in SONET rings. In *First Annual International Conference on Broadband Networks, San-José, California, USA*, pages 254–262, October 2004.
- [138] M. Flammini, L. Moscardelli, M. Shalom, and S. Zaks. Approximating the traffic grooming problem. In Springer Berlin / Heidelberg, editor, *16th Annual International Conference on Algorithms and Computation (ISAAC)*, volume 3827 of *Lecture Notes in Computer Science*, pages 915–924, 2005. to appear in *Journal of Discrete Algorithms*.
- [139] A. Gumaste and S. Q. Zheng. Next-generation optical storage area networks : The light-trails approach. *IEEE Communications Magazine*, 43(3) :72–78, March 2005.
- [140] L.-W. Chen, E. Modiano, and P. Saengudomlert. Uniform versus non-uniform band switching in WDM networks. *Computer Networks*, 50(2) :149–167, February 2006.
- [141] S. Huang, R. Dutta, and G.N. Rouskas. Traffic grooming in path, star, and tree networks : Complexity, bounds, and algorithms. *IEEE Journal on Selected Areas in Communications*, 24(4) :66–82, April 2006.
- [142] Y. Wang and Q. Gu. Grooming of symmetric traffic in unidirectional SONET/WDM rings. In *IEEE International Conference on Communications (ICC)*, volume 5, pages 2407–2414, Istanbul, June 2006.
- [143] Y. Ye, H. Woesner, and I. Chlamtac. Waveband switching in light trail optical networks with dynamic traffic. *OSA Journal of Optical Networking*, 2006.
- [144] O. Amini, S. Pérennes, and I. Sau Valls. Hardness and approximation of traffic grooming. In Takeshi Tokuyama, editor, *18th International Symposium on Algorithms and Computation (ISAAC)*, volume 4835 of *Lecture Notes in Computer Science*, pages 561–573, Sendai, Japan, December 2007. Springer.
- [145] J-C. Bermond, X. Muñoz, and I. Sau Valls. Traffic Grooming in Bidirectional WDM Ring Networks. Manuscript, 2007.

- [146] Y. Brehon, D. Kofman, M. Piore, and M. Diallo. Optimal virtual topology design using bus-label switched paths. *IEEE Journal on Selected Areas in Communications*, 25(5) :1001 – 1010, June 2007.
- [147] X. Cao, V. Anand, and C. Qiao. Waveband switching for dynamic traffic demands in multigranular optical networks. *IEEE/ACM Transactions on Networking*, 2007.
- [148] M. Shalom, W. Unger, and S. Zaks. On the complexity of the traffic grooming problem in optical networks. In Pierluigi Crescenzi, Giuseppe Prencipe, and Geppino Pucci, editors, *4th International Conference on Fun With Algorithms (FUN)*, volume 4475 of *Lecture Notes in Computer Science*, pages 262–271, Castiglioncello (LI), Tuscany, Italy, June 2007. Springer.
- [149] M. Shalom, P. W. H. Wong, and S. Zaks. Optimal on-line colorings for minimizing the number of ADMs in optical networks. In Andrzej Pelc, editor, *21st International Symposium on Distributed Computing (DISC)*, volume 4731 of *Lecture Notes in Computer Science*, pages 435–449, Lemesos, Cyprus, September 2007. Springer.
- [150] F. Solano, L.F. Caro, J. Cavalcante de Oliveira, R. Fabregat, and J-L. Marzo. G+ : Enhanced traffic grooming in WDM mesh networks using lighttours. *IEEE Journal on Selected Areas in Communications*, 25(5) :1034–1047, 2007.
- [151] J. Zheng and J.P. Jue. Waveband switching, routing, and grooming : introduction to the feature issue. *OSA Journal of Optical Networking*, 6(2) :217–218, January 2007.
- [152] C. J. Colbourn, G. Quattrocchi, and V. R. Syrotiuk. Grooming for two-period optical networks. *Networks*, 52(4) :307–324, December 2008.
- [153] C. J. Colbourn, G. Quattrocchi, and V. R. Syrotiuk. Lower bounds for two-period grooming via linear programming duality. *Networks*, 52(4) :299–306, December 2008.
- [154] C.J. Colbourn, H.-L. Fu, G. Ge, A.C.H. Ling, and H.-C. Lu. Minimizing SONET ADMs in unidirectional WDM rings with grooming ratio seven. *SIAM Journal on Discrete Mathematics*, 23(1) :109–122, 2008.
- [155] R.J. Crouser, B. Rice, A. Sampson, and R. Libeskind-Hadas. On-line distributed traffic grooming. In *IEEE International Conference on Communication (ICC)*, pages 5239 – 5246, Beijing, May 2008.
- [156] X. Muñoz and I. Sau. Traffic grooming in unidirectional WDM rings with bounded-degree request graph. In H. Broersma, T. Erlebach, T. Friedetzky, and D. Paulusma, editors, *34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 5344 of *Lecture Notes in Computer Science*, pages 300–311. Springer, June 2008.
- [157] Y. Wang and Q. Gu. On the complexity and algorithm of grooming regular traffic in WDM optical networks. *Journal of Parallel and Distributed Computing*, 68(6) :877–886, June 2008.
- [158] C.J. Colbourn, G. Ge, and A.C.H. Ling. Optical grooming with grooming ratio 9. preprint, 2009.
- [159] C.J. Colbourn, G. Ge, and A.C.H. Ling. Optical grooming with grooming ratio eight. *Discrete Applied Mathematics*, 157(13) :2763–2772, July 2009.
- [160] Z. Li and I. Sau. Graph Partitioning and Traffic Grooming with Bounded Degree Request Graph. In *35th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2009)*, volume 5911 of *Lecture Notes in Computer Science*, pages 250–261, June 2009. Best student paper award.
- [161] I. Sau. *Optimization in Graphs under Degree Constraints with Application to Telecommunication Networks*. PhD thesis, Université de Nice Sophia et Université Polytechnique de Catalogne, October 2009.
- [162] M. Shalom, P. Wong, and S. Zaks. Optimal on-line colorings for minimizing the number of ADMs in optical networks. *Journal of Discrete Algorithms*, In Press, Accepted Manuscript, 2009.
- [163] Y. Wang and Q. Gu. Minimizing SONET add-drop multiplexers in optical UPSR networks using the minimum number of wavelengths. *Networks*, 53(3) :276–286, May 2009.
- [164] R. Dutta, A.E Kamal, and G.N Rouskas, editors. *Traffic Grooming for Optical Networks : Foundations, Techniques and Frontiers*. Optical Networks. Springer, August 2008.

- [165] A. Somani. *Survivability & Traffic Grooming in WDM Optical Networks*. Cambridge Press, January 2006.
- [166] K. Zhu, H. Zhu, and B. Mukherjee. *Traffic Grooming in Optical WDM Mesh Networks*. Springer, 2005.
- **Protection, tolérance aux pannes** —
- [167] J. Suurballe and R. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2) :325–336, 1984.
- [168] W. D. Grover and D. Stamatelakis. Cycle-oriented distributed preconfiguration : Ring-like speed with mesh-like capacity for self-planning network restoration. In *IEEE ICC*, pages 537–543, 1998.
- [169] D. Papadimitriou, F. Poppe, J. Jones, S. Venkatachalam, S. Dharanikota, R. Jain, R. Hartani, D. Griffith, and Y. Xue. Inference of shared risk link groups. IETF Draft, Internet Working Group, November 2001.
- [170] D. Papadimitriou, F. Poppe, S. Dharanikota, R. Hartani, R. Jain, J. Jones, S. Venkatachalam, and Y. Xue. Shared risk link groups encoding and processing. IETF Draft, CCAMP Working Group, November 2002.
- [171] J.Q. Hu. Diverse routing in mesh optical networks. *IEEE Transactions on Communications*, 51(3) :489–494, 2003.
- [172] D. Papadimitriou, F. Poppe, S. Dharanikota, R. Hartani, R. Jain, J. Jones, S. Venkatachalam, and Y. Xue. Shared risk link groups inference and processing. IETF Draft, CCAMP Working Group, June 2003.
- [173] H. Zang, C. Ou, and B. Mukherjee. Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints. *IEEE/ACM Trans. Netw.*, 11(2) :248–258, 2003.
- [174] P. Datta and A.K. Somani. Diverse routing for shared risk resource groups (SRRG) failures in WDM optical networks. In *Proceedings of IEEE BroadNets*, pages 120– 129, October 2004.
- [175] X. Luo and B. Wang. Diverse routing in WDM optical networks with shared risk link group (srlg) failures. In *5th International Workshop on Design of Reliable Communication Networks – DRCN*, page 8pp. IEEE, 16-19 Oct. 2005.
- [176] L. Shen, X. Yang, and B. Ramamurthy. Shared risk link group (srlg)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks. *IEEE/ACM Transactions on Networking*, 13(4) :918– 931, 2005.
- [177] S. Yuan, S. Varma, and J.P. Jue. Minimum-color path problems for reliability in mesh networks. In *IEEE InfoCom*, volume 4, pages 2658–2669, 2005.
- [178] J. K. Choi and D. Guha. Fast end-to-end restoration mechanism with srlg using centralized control. IETF Draft, Network Working Group, August 2006.
- [179] A. Faragó. A graph theoretic model for complex network failure scenarios. In *Proceedings of the Eighth INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.
- [180] X. Luo and B. Wang. Multi-constrained routing in networks with shared risk link groups. In *IEEE BroadNets*, 2006.
- [181] M.-E. Voge. How to transform a multilayer network into a colored graph. In *IEEE-LEOS IC-TON / COST 293 annual conference on GRaphs and ALgorithms in communication networks*, volume 3, pages 116–119, Nottingham, June 2006.
- [182] M.-E. Voge. *Optimisation des réseaux de télécommunication : Réseaux multiniveaux, Tolérance aux pannes et Surveillance du trafic*. PhD thesis, Ecole doctorale STIC, Université de Nice-Sophia Antipolis, novembre 2006.
- [183] P. Datta and A.K. Somani. Graph transformation approaches for diverse routing in shared risk resource group (SRRG) failures. *Computer Networks*, 52(12) :2381–2394, August 2008.
- [184] S. Orłowski and M. Pióro. On the complexity of column generation in survivable network design with path-based survivability concepts. ZIB report ZR-08-51, Konrad-Zuse-Zentrum für Informationstechnik Berlin, December 2008.

- [185] S. Stefanakos. Reliable routings in networks with generalized link failure events. *IEEE/ACM Transactions on Networking*, 16(6) :1331–1339, December 2008.
- [186] M.S. Kiaei, C. Assi, and B. Jaumard. A survey on the p -cycle protection method. *IEEE Communications Surveys & Tutorials*, 11(3) :53–70, 2009.
- [187] S. Orlowski and M. Pióro. On the complexity of column generation in survivable network design with path-based survivability concepts. In *Proceedings of the 4th International Network Optimization Conference (INOC 2009), Pisa, Italy*, number 76, April 2009. Short version of ZIB report ZR-08-51 [184].
- [188] R. Bhandari. *Survivable Networks : Algorithms for Diverse Routing*. Kluwer Academic, 1999.
- [189] A. Tomaszewski, M. Pióro, and M. Zotkiewicz. On the complexity of resilient network design. *Networks*, to appear.
- Reroutage, reconfiguration —
- [190] M. Ackroyd. Call repacking in connecting networks. *IEEE Transactions on Communications*, 27(3) :589–591, March 1979.
- [191] J.-F.P. Labourdette, G.W. Hart, and A.S. Acampora. Branch-exchange sequences for reconfiguration of lightwave networks. *IEEE Transactions on Communications*, 42(10) :2822–2832, Oct 1994.
- [192] K. Bala, G. Ellinas, M. Post, Chien-Chung Shen, J. Wei, and N. Antoniadis. Towards hitless reconfiguration in WDM optical networks for ATM transport. In *IEEE Global Telecommunications Conference (Globecom)*, volume 1, pages 316–320 vol.1, Nov 1996.
- [193] K.-C. Lee and V.O.K. Li. A wavelength rerouting algorithm in wide-area all-optical networks. *IEEE/OSA Journal of Lightwave Technology*, 14(6) :1218–1229, June 1996.
- [194] G. Mohan and C.S.R. Murthy. A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks. *IEEE/OSA Journal of Lightwave Technology*, 17(3) :406–417, March 1999.
- [195] E.W.M. Wong, A.K.M. Chan, and T.-S.P. Yum. A taxonomy of rerouting in circuit-switched networks. *IEEE Communications Magazine*, 37(11) :116–122, November 1999.
- [196] D. Banerjee and B. Mukherjee. Wavelength-routed optical networks : Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transactions on Networking*, 8(5) :598–607, October 2000.
- [197] E.W.M. Wong, A.K.M. Chan, and T.-S.P. Yum. Analysis of rerouting in circuit-switched networks. *IEEE/ACM Transactions on Networking*, 8(3) :419–427, June 2000.
- [198] S. Beker, D. Kofman, and N. Puech. Off-line MPLS layout design and reconfiguration : Reducing complexity under dynamic traffic conditions. In *International Network Optimization Conference (INOC)*, pages 61–66, October 2003.
- [199] A. Gencata and B. Mukherjee. Virtual-topology adaptation for WDM mesh networks under dynamic traffic. *IEEE/ACM Transactions on Networking*, 11(2) :236–247, April 2003.
- [200] N. Jose and A.K. Somani. Connection rerouting/network reconfiguration. In *4th International Workshop on Design of Reliable Communication Networks (DRCN)*, pages 23–30. IEEE, October 2003.
- [201] B. G. Józsa and M. Makai. On the solution of reroute sequence planning problem in MPLS networks. *Computer Networks*, 42(2) :199 – 210, 2003.
- [202] S. Beker, N. Puech, and V. Friderikos. A tabu search heuristic for the off-line MPLS reduced complexity layout design problem. In *3rd International IFIP-TC6 Networking Conference (Networking)*, volume 3042 of *Lecture Notes in Computer Science*, pages 514–525, Athens, Greece, May 2004. Springer.
- [203] W. Golab and R. Boutaba. Policy-driven automated reconfiguration for performance management in WDM optical networks. *IEEE Communications Magazine*, 42(1) :44–51, January 2004.
- [204] M. Saad and Zhi-Quan Luo. Reconfiguration with no service disruption in multifiber WDM networks. *IEEE/OSA Journal of Lightwave Technology*, 23(10) :3092–3104, October 2005.

- [205] X. Chu, T. Bu, and X.-Y. Li. A study of lightpath rerouting schemes in wavelength-routed WDM networks. In *IEEE International Conference on Communications (ICC)*, pages 2400 – 2405, Hong-Kong, June 2007. IEEE.
- [206] J. Y. Zhang, O. W. W. Yang, J. Wu, and M. Savoie. Optimization of semi-dynamic lightpath rearrangements in a WDM network. *IEEE Journal on Selected Areas in Communications*, 25(9) :3–17, December 2007.
- [207] O. Klopfenstein. Rerouting tunnels for MPLS network resource optimization. *European Journal of Operational Research*, 188(1) :293 – 312, 2008.
- [208] R. Soares. Tradeoff between the number of agents and the number of rounds of a process strategy. Master’s thesis, Stage pré-doctoral, projet Mascotte et UFC (Brésil), 15 février - 15 mai 2009.
- [209] F. Solano. Analyzing two different objectives of the WDM network reconfiguration problem. In *IEEE Global Communications Conference (Globecom)*, Honolulu, Hawaii, USA, December 2009. to appear.
- [210] F. Solano and M. Pióro. A mixed-integer programming formulation for the lightpath reconfiguration problem. In *VIII Workshop on G/MPLS Networks (WGN8)*, Girona, Spain, October 2009.
- [211] F. Solano and M. Pióro. Optimizing maximum disruption during reconfiguration of lightpaths. In *16th Polish Teletraffic Symposium (PTS)*, Łódź, Poland, September 2009.

— Routage dans les réseaux optiques —

- [212] B. Mukherjee. WDM-based local lightwave networks – Part I : Single-hop systems. *IEEE Network*, 6(3) :12–27, May 1992.
- [213] B. Mukherjee. WDM-based local lightwave networks – Part II : Multi-hop systems. *IEEE Network*, 6(4) :20–32, July 1992.
- [214] B. Beauquier. *Communications dans les réseaux optiques par multiplexage en longueur d’onde*. PhD thesis, université de Nice-Sophia Antipolis – STIC, janvier 2000.
- [215] H. Rivano. *Algorithmique et télécommunications : coloration et multiflot approchés et applications aux réseaux d’infrastructure*. PhD thesis, Ecole doctorale STIC, Université de Nice-Sophia Antipolis, November 2003.
- [216] F. Ramos, E. Kehayas, J. M. Martinez, R. Clavero, J. Marti, L. Stampoulidis, D. Tsiokos, H. Avramopoulos, J. Zhang, P. V. Holm-Nielsen, N. Chi, P. Jeppesen, N. Yan, I. Tafur Monroy, A.M.J. Koonen, M.T. Hill, Y. Liu, H.J.S. Dorren, R. Van Caenegem, D. Colle, M. Pickavet, and B. Riposati. IST-LASAGNE : Towards all-optical label swapping employing optical logic gates and optical flip-flops. *IEEE Journal on Selected Areas in Communications*, 23(10) :2993–3011, 2005.
- [217] R. Van Caenegem, J. M. Martinez, D. Colle, M. Pickavet, P. Demeester, F. Ramos, and J. Marti. From IP over WDM to all-optical packet switching : Economical overview. *IEEE/OSA Journal of Lightwave Technology*, 24(4) :1638–1645, 2006.
- [218] B. Jaumard, C. Meyer, and B. Thiongane. Comparison of ILP formulations for the RWA problem. *Optical Switching and Networking*, 4(3-4) :157–172, 2007.

— Autre —

- [219] M. Thorup and U. Zwick. Compact routing schemes. In *SPAA 2001 : Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures, July 4-6, 2001, Heraklion, Crete Island, Greece*. ACM, pages 1–10, 2001.
- [220] Q. Jiang, D.S. Reeves, and P. Ning. Improving robustness of PGP keyrings by conflict detection. In *Proceedings of the RSA Conference Cryptographers’ Track (CT-RSA2004)*, pages 194–207. LNCS 2964, February 2004.
- [221] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. *ACM Transactions on Algorithms*, 4(3), 2008.
- [222] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *INFOCOM*, pages 457–465. IEEE, 2008.

CV complet

