Optimisation des réseaux de télécommunications : Réseaux multiniveaux, Tolérance aux pannes et Surveillance du trafic

Marie-Emilie Voge

UNIVERSITÉ de NICE-SOPHIA ANTIPOLIS – UFR SCIENCES

École Doctorale STIC

THÈSE

pour obtenir le titre de

Docteur en SCIENCES

de l'Université de Nice-Sophia Antipolis
Discipline : **INFORMATIQUE**

présentée et soutenue par

Marie-Emilie VOGE

Optimisation des réseaux de télécommunications : Réseaux multiniveaux, Tolérance aux pannes et Surveillance du trafic

Thèse dirigée par **Jean-Claude Bermond**

et préparée au sein du projet MASCOTTE (I3S(CNRS/UNSA)/INRIA)

soutenue le **17 novembre 2006**

Jury :

Examinateurs	Mme.	**Myriam**	**Preissmann**	Chargée de Recherche
	M.	**Michel**	**Cosnard**	Professeur
	M.	**David**	**Coudert**	Chargé de Recherche
Directeur	M.	**Jean-Claude**	**Bermond**	Directeur de Recherche
Rapporteurs	Mme.	**Claudia**	**Linhares-Sales**	Professeur
	M.	**Philippe**	**Mahey**	Professeur
	M.	**Jean-Claude**	**König**	Professeur

Table des matières

Table des figures

Chapitre 1

Introduction

Les problèmes étudiés dans cette thèse sont motivés par des questions issues de l'optimisation des réseaux de télécommunication. Ces problèmes d'optimisation sont indépendants des choix protocolaires et s'appuient uniquement sur les grands principes du routage et de l'acheminement des données. Par conséquent, les résultats obtenus pourraient s'appliquer à d'autres réseaux présentant des caractéristiques semblables pour l'acheminement, par exemple les réseaux de transport de voyageurs ou de marchandises.

Nous avons abordé ces problèmes sous deux angles principaux. D'une part nous avons étudié leurs propriétés de complexité et d'inapproximabilité. D'autre part nous avons dans certains cas proposé des algorithmes exacts ou d'approximation ou encore des méthodes heuristiques que nous avons pu comparer à des formulations en programme linéaires mixtes (MILP pour *Mixed Integer Linear Programming*) sur des instances particulières. Ces deux approches sont complémentaires : l'une s'intéresse aux limites théoriques des méthodes de résolution, l'autre consiste à trouver des moyens d'obtenir des solutions réalisables de bonne qualité. En particulier, connaître la classe de complexité d'un problème d'optimisation et comprendre le cœur de la difficulté permet parfois de déduire des méthodes de résolution efficaces.

Nous considérons les réseaux internet dans leur globalité c'est-à-dire aussi bien les *réseaux d'accès* que les *réseaux de cœur* . Un réseau d'accès est composé de plusieurs points de présence (POP pour *Point Of Presence*) gérés par des opérateurs concurrents. Les utilisateurs sont connectés entre eux à travers les POP de leurs opérateurs respectifs. Pour ce faire, les POP sont connectés entre eux par le réseau de cœur. Le réseau de cœur résulte de l'interconnexion de systèmes autonomes (AS pour *Autonomous System*) gérés par différentes autorités (universités, opérateurs, entreprises etc).

Pour des raisons historiques, les réseaux de télécommunication sont composés de plusieurs niveaux technologiques assurant chacun des fonctions spécifiques. L'intégration d'applications nouvelles et de services, comme la voix et les données, sur une même infrastructure de réseau a conduit à des empilements complexes comme IP/ATM/SDH/WDM. Les technologies récentes permettent aujourd'hui de simplifier la structure des réseaux et de converger vers un modèle IP/WDM dans lequel l'architecture MPLS,(*Multi Protocol Label Switching*) ou son extension GMPLS (*Generalized*-MPLS), est de plus en plus employée. Les réseaux IP/WDM sont constitués de plusieurs réseaux virtuels empilés sur un réseau physique de fibres optiques et d'équipements (nœuds) les interconnectant. Les liens du niveau virtuel, ou *Label Switched Paths* (LSP) dans le cadre de l'architecture MPLS, correspondent à des routes préétablies entre les nœuds du réseau. Les utilisateurs n'ont en général qu'une vision du réseau limitée au niveau virtuel le plus haut, sur lequel sont routées leurs requêtes.

Dans cette thèse nous considérons dans un premier temps les réseaux de cœur de type IP/WDM à deux niveaux utilisant l'architecture GMPLS. Ces réseaux comportent un niveau physique (niveau inférieur) et un seul niveau virtuel (niveau supérieur) constitué de LSP, sur lequel est routé un ensemble de requêtes provenant des utilisateurs du réseau.

Router les requêtes sur un niveau virtuel permet de simplifier l'acheminement des flux de paquets IP dans le réseau en regroupant des flux de faibles débits sur des routes communes au niveau physique. En effet, l'acheminement classique au niveau IP nécessite l'analyse des paquets IP en chaque nœud traversés avant d'arriver à destination. Par contre, l'acheminement sur un LSP ne nécessite qu'une seule analyse à l'entrée du LSP. Les LSP sont en quelque sorte des tubes, une fois qu'un flux est émis sur un LSP, il est acheminé sans avoir à être réexaminé à chaque nœud traversé jusqu'à l'extrémité du LSP. Grâce au routage sur le niveau virtuel, les paquets IP ne sont donc plus analysés à chaque nœud traversé ce qui induit une réduction du coût des nœuds aussi bien au niveau électronique (IP) qu'optique (WDM).

Réduire les coûts de maintenance et d'exploitation est crucial pour un opérateur. Pour être compétitif, il doit proposer des tarifs attrayants. Cependant il s'engage aussi auprès de ses clients à maintenir une certaine qualité de service dans le réseau (SLA pour *Service Level Agreement*), ce qui a un coût aussi bien en termes de ressources nécessaires que de gestion. Les performances d'un réseau, qui peuvent être évaluées par des mesures de trafic, dépendent non seulement de sa conception et de son dimensionnement, mais aussi de sa capacité à maintenir le service malgré les pannes qui surviennent régulièrement et à n'importe quel niveau.

Le rôle des mécanismes de *protection* et de *restauration* est d'éviter l'interruption des services en cas de panne. La restauration consiste à déterminer un nouveau chemin pour une requête seulement lorsqu'une panne se produit. C'est le mécanisme de survie utilisé au niveau IP : les chemins suivis par les paquets sont calculés au fur et à mesure de leur progression d'un nœud à l'autre en fonction de l'état du réseau. Lorsqu'un équipement est indisponible il n'est simplement pas pris en compte dans le calcul des routes. Dans cette thèse nous ne nous intéressons pas à la restauration mais uniquement à la protection plus adaptée aux niveaux virtuels et physique. L'idée essentielle des méthodes de protection est de prévoir, au moment d'établir une connexion entre deux nœuds, plusieurs chemins de sorte qu'il en reste toujours au moins un opérationnel en cas de panne. A partir de ce principe, de nombreuses variantes ont vu le jour afin d'atteindre le meilleur compromis entre le délai de rétablissement du service après une panne et les ressources requises. Notons que pour respecter le délai acceptable de 50 ms pour la téléphonie au niveau utilisateur (IP), le délai au niveau WDM doit être de l'ordre de la microseconde. Les méthodes de protection conçues pour un niveau de réseau seulement, sont toujours d'actualité dans les réseaux multiniveaux, d'autant plus que le mécanisme de tolérance aux pannes est actuellement optimisé indépendamment pour chaque niveaux.

Cependant, depuis l'introduction de MPLS, des mécanismes de protection unifiés où les différents niveaux coopèrent sont étudiés. Outre le type de mécanisme à adopter parmi les multiples possibilités pour chaque niveau virtuel, il est aussi nécessaire de déterminer les interactions entre les niveaux et le rôle de chacun en cas de panne. Ces questions sont d'ordre protocolaire et ne nous intéressent pas directement, mais cette nouvelle manière d'aborder la protection a mis en évidence les inconvénients générés par une optimisation indépendante pour chaque niveau. Ces inconvénients peuvent se limiter à une mauvaise utilisation des ressources ou aller jusqu'à la déconnexion complète d'une partie du niveau supérieur du réseau lorsqu'une unique panne survient au niveau physique. Ces problèmes ont été identifiés dans la littérature comme les conséquences de l'existence de *Shared Risk Resource Group* (SRRG), ou groupes de risque, dans les réseaux multiniveaux.

D'une manière générale un groupe de risque est un ensemble d'éléments dont la disponibilité

dépend de celle d'une même ressource. Dans un réseau multiniveaux, chaque ressource physique (nœud ou lien) est à l'origine d'un groupe de risque contenant tous les liens virtuels routés sur cette ressource. En effet, si cette ressource tombe en panne, tous les liens virtuels du groupe de risque associé sont coupés simultanément : ils sont soumis au même risque de panne.

Les groupes de risque sont à l'origine des deux premières problématiques d'optimisation que nous avons abordées dans cette thèse. Elles concernent la conception de réseaux virtuels (tolérants aux pannes ou non), et la connexité et la vulnérabilité aux pannes d'un réseau multiniveaux donné. La troisième problématique est liée à la mesure des performances d'un POP.

Conception de réseau virtuel Concevoir un réseau virtuel qui reste connexe quelle que soit la panne qui survienne au niveau physique permet de réduire l'influence des groupes de risque sur la disponibilité du réseau. Comme nous le verrons dans le chapitre 3, il s'agit d'un problème difficile contenant en particulier le problème du *groupage*. Le groupage consiste d'une manière générale à agréger des flux de faibles débit, par exemple les flux émis par un ensemble d'utilisateurs, en un flux de plus haut débit, par exemple le débit d'une longueur d'onde. Il est ensuite possible d'agréger plusieurs longueurs d'onde entre deux nœuds pour remplir une fibre optique. Nous avons étudié un problème particulier de groupage [PV05] issu du problème de conception de réseau virtuel tolérant aux pannes.

Tolérance aux pannes d'un réseau virtuel donné L'existence même d'un réseau virtuel implique l'existence de groupes de risque. Si une conception judicieuse permet de limiter l'influence des groupes de risque, elle est insuffisante pour garantir la disponibilité du réseau en cas de panne. Optimisé pour un ensemble de requêtes donné, le niveau virtuel n'est plus nécessairement adapté lorsque le trafic évolue. En outre, le problème de conception nécessite un temps de calcul important. Ainsi le réseau virtuel ne peut pas être optimisé dès la moindre variation de trafic. Par conséquent, dans un contexte dynamique, le routage de nouvelles requêtes doit s'effectuer dans un réseau virtuel non nécessairement optimal. C'est pourquoi nous avons abordé la question des groupes de risque sous un autre angle consistant simplement à étudier les problèmes de routage et de tolérance aux pannes dans un réseau virtuel en présence de groupes de risque. Pour cela nous avons modélisé les réseaux multiniveaux par des *graphes colorés*. Un graphe coloré est un graphe représentant le niveau virtuel d'un réseau multiniveaux. L'ensemble des arêtes est partitionné en couleurs représentant les groupes de risque. Deux arêtes appartenant à la même couleur correspondent à deux liens virtuels appartenant au même groupe de risque. Dans ces graphes nous étudions des problèmes fondamentaux liés à la connexité (existence d'ensembles de chemins particuliers, arbre couvrant) du réseau représenté ainsi qu'à sa vulnérabilité aux pannes (problèmes de coupes). Ces travaux ont fait l'objet de plusieurs publications [CDP$^+$06, CPRV06, Vog06a, Vog06b].

Surveillance du trafic dans un POP La surveillance du trafic est un outil important pour un opérateur qui lui permet d'approfondir sa connaissance du réseau. Il peut alors établir des SLA qu'il est en mesure de respecter, mais aussi vérifier qu'il les respecte effectivement. Mesurer le trafic posséde de nombreuses applications importantes, dont la détection des pannes qui est une étape incontournable à la mise en place de solutions de secours. Il existe deux méthodes de surveillance complémentaires, la surveillance passive et la surveillance active. La surveillance passive permet entre autres de mesurer les volumes des trafics de différents types, ce qui permet un dimensionnement correcte des ressources de secours à réserver. La surveillance active intervient directement dans la détection de panne et permet également de mesurer les performances du réseau du point de vue des utilisateurs. La surveillance du trafic, qu'elle soit active ou passive, nécessite l'installation

d'équipements spécifiques dans le réseau. Ces équipements étant assez coûteux, il n'est pas envisageable pour un opérateur d'en installer sur chaque lien physique pour la surveillance passive ou sur chaque nœud pour la surveillance active. Minimiser le nombre d'instruments de mesure à installer et déterminer leurs emplacements constituent donc des enjeux économiques importants pour un opérateur. Nous avons étudié les problèmes de placement des équipements pour la surveillance aussi bien active que passive du trafic, ce qui a donné lieu à deux publications [CFGL+05b, CFGL+05a].

Nous présentons nos travaux sur ces trois problématiques selon le plan suivant.

Le premier chapitre est consacré aux réseaux IP/WDM. Dans un premier temps nous présentons les principaux aspects de ces réseaux intervenant dans le routage et l'acheminement des flux en provenance des utilisateurs ainsi que l'architecture des points de présence des opérateurs. Ensuite nous proposons une modélisation des réseaux multiniveaux et des points de présences sous forme de graphes et nous précisons quelques propriétés des flux considérés dans ces réseaux. Les méthodes de protection classiques sont rappelées et étendus au cas multiniveaux. En particulier nous précisons la notion de groupe de risque.

Dans le second chapitre nous abordons la conception de réseau virtuel tolérant aux pannes permettant d'écouler un trafic donné et statique. Nous formulons en MILP ce problème dans le cadre de la protection par chemin, partagée et dépendante de la panne pour le niveau virtuel. La suite de ce chapitre traite d'un problème particulier de groupage extrait du problème de conception de réseau. Pour ce problème, le réseau physique que nous considérons est un chemin orienté de capacité infinie. Tous les liens virtuels, ou *tubes*, sont de même capacité, ou *facteur de groupage*. Tout aspect de tolérance aux pannes est nécessairement supprimé puisque le réseau physique est un chemin unique. Malgré ces hypothèses restrictives, minimiser le nombre de tubes nécessaire à l'acheminement d'un ensemble de requêtes unitaires quelconque reste un problème difficile. Nous proposons donc deux heuristiques dont nous comparons les performances aux solutions optimales fournies par une formulation en MILP du problème. Étant donné que pour des instances de taille moyenne les temps de calcul des solveurs sont déjà de l'ordre de plusieurs heures, nous étudions les propriétés d'ensembles de requêtes pour lesquels un groupage optimal est connu et permettraient de tester les heuristiques sur des instances de grande taille.

Dans le troisième chapitre nous proposons une modélisation des réseaux multiniveaux par des *graphes colorés* permettant de définir simplement les problèmes liés au routage et à la protection dans un réseau virtuel en présence de groupes de risque. Dans ces graphes nous définissons un ensemble de problèmes d'optimisation d'un intérêt majeur pour la tolérance aux pannes dont nous étudierons la complexité et les ressemblances avec les problèmes de théorie des graphes classique. Ceci nous conduira à définir un paramètre des graphes colorés qui donne une indication de l'influence d'une couleur dans le graphe et joue un rôle important dans la complexité de certains problèmes, le *span* des couleurs. Nous nous intéressons ensuite à la transformation d'un réseau multiniveaux en graphe coloré dont peut dépendre la complexité de certains problèmes.

Le quatrième et dernier chapitre a pour objet la surveillance du trafic circulant dans les points de présence d'un opérateur. Nous étudions les problèmes d'optimisation liés au placement de ces instruments de mesures pour la surveillance passive et pour la surveillance active. Nous présentons des méthodes de résolution et des formulations en MILP pour ces problèmes ainsi que quelques résultats de complexité. En particulier nous montrons que certains de ces problèmes sont équivalents à des

problèmes de couverture. Enfin nous présentons également les résultats de la comparaison entre nos formulations en MILP et d'autres méthodes de résolutions de la littérature pour les problèmes de placement d'instruments de mesure du trafic.

L'annexe A rappelle la définition de certains problèmes d'optimisation ou de décision ainsi que les principaux résultats de complexité et d'inapproximabilité les concernant. Dans cette thèse nous utilisons des réductions de ces problèmes difficiles pour montrer la complexité ou l'inapproximabilité des problèmes que nous avons étudiés.

Les annexes B et C se rapportent uniquement au chapitre 4. L'annexe B donne une version détaillée d'un algorithme polynomial évoqué pour un problème de transformation d'un réseau multiniveaux en graphe coloré. L'annexe C présente des exemples de réseaux utilisés dans la littérature pour effectuer des tests de méthodes de résolution pour les problèmes d'optimisation issus des groupes de risque.

Chapitre 2

Les réseaux IP/WDM

D'une manière générale, les réseaux de télécom (Internet, téléphone, câble etc) sont composés d'une partie *réseau de cœur* et d'une partie *réseau d'accès*.

C'est grâce au réseau d'accès que les utilisateurs peuvent se connecter au reste du réseau. Le réseau d'accès d'Internet est partagé entre plusieurs fournisseurs d'accès (ISP pour *Internet Service Provider*). Chacun possède ou loue à un autre opérateur un ensemble d'équipements, les routeurs d'accès, auxquels sont raccordés les utilisateurs par des câbles ou des liaisons radio. Pour des raisons technologiques d'atténuation des signaux, la longueur de ces câbles est limitée et un ISP doit posséder des installations d'accès dans chaque zone géographique, par exemple dans chaque ville, où il souhaite proposer ses services. Ces installations s'appellent des *points de présence* (POP pour *Point Of Presence*) et chaque ISP peut en posséder plusieurs suivant son importance. Un POP est constitué d'un ensemble de routeurs d'accès assurant la liaison avec les utilisateurs et de routeurs de cœur permettant l'ouverture du POP sur le cœur du réseau.

Comme le réseau d'accès, le cœur du réseau n'est pas construit et administré par une entité unique, mais résulte de l'interconnexion de *systèmes autonomes* (AS pour *Autonomous System*) hétérogènes plus ou moins étendus géographiquement. Un AS est un ensemble de réseaux sous le contrôle d'une seule et même entité, typiquement un fournisseur d'accès à Internet, une université, une entreprise, un opérateur etc. Les choix technologiques et protocolaires concernant le transport des données à l'intérieur d'un AS peuvent différer d'un AS à l'autre. Cependant le protocole IP (*Internet Protocol*) et des architectures comme MPLS (*Multi Protocol Label Switching*) indépendantes de la technologie assurent l'interopérabilité entre eux. L'architecture MPLS joue un autre rôle important dans les réseaux de cœur en facilitant l'agrégation du trafic. Dans le cœur du réseau les données transitent à très haut débit sur des fibres optiques connectant des routeurs optiques entre eux. La majorité des utilisateurs ne nécessitant pas toute la capacité de transmission d'une fibre optique, il est nécessaire d'agréger les flux de faible débit en provenance de plusieurs émetteurs en un flux de débit comparable à celui d'une longueur d'onde pour acheminer tous ces flux dans le cœur du réseau à très haut débit.

2.1 Description des Réseaux IP/WDM

Après avoir donner un aperçu de l'origine historique des réseaux IP/WDM, nous préciserons les modes d'interaction des différents niveaux au sein de ces réseaux. Nous présenterons ensuite les aspects important permettant de comprendre comment sont routées les données dans un tel réseau en décrivant chacun des trois composants principaux : la couche IP, la technologie WDM et enfin

l'architecture MPLS.

2.1.1 Évolution de l'architecture

Pour des raisons historiques, les réseaux de télécommunication ont une architecture en plusieurs couches technologiques. Chaque couche possède une fonction particulière et offre un service à la couche qui est au dessus en utilisant la couche du dessous. Le modèle de référence OSI (pour *Open Systems Interconnection*) de l'ISO (pour *International Standardization Organization*) est un modèle d'architecture en sept couches qui permet de délimiter toutes les fonctions assurant le fonctionnement d'un réseau et les grands principes de coopération entre les couches. Il reste toutefois un modèle théorique car dans la réalité une couche peut avoir plusieurs fonctions et il y en a en général moins de sept.

Les architectures les plus courantes sont composées d'une couche internet constitué d'un niveau IP et d'un niveau ATM. Le niveau IP offre un support au développement de services et d'applications pour les utilisateurs. Le contrôle des flux, la gestion de la qualité de service et plus généralement l'ingénierie de trafic sont assurés par le niveau ATM (*Asynchronous Transfert Mode*) sur laquelle repose le niveau IP. Ensuite la couche SDH (*Synchronous Digital Hierarchy*) gère le transport des flux ATM sur le réseau optique WDM (*Wavelength Division Multiplexing*). Cependant cette architecture est le fruit d'une évolution technologique progressive, par suite elle manque aujourd'hui de flexibilité et de dynamicité pour faire face à l'augmentation continuelle du trafic [Liu02]. D'autre part, pour l'acheminement des données, chaque couche leur ajoute des informations de contrôle lourdes (encapsulation), ce qui induit un sur-coût en bande passante et un traitement des données complexe dans les nœuds.

C'est pourquoi le besoin de simplifier cet empilement de couches est de plus en plus présent. La tendance est de supprimer les couches intermédiaires pour obtenir un réseau de type IP/WDM [SKS03, Liu02, RLA04].

L'intérêt du modèle IP/WDM vers lequel tendent les réseaux actuels se fonde sur plusieurs constats. D'une part, les réseaux optiques WDM peuvent suivre la croissance continue du trafic Internet en exploitant les infrastructures déjà existantes. L'utilisation de la technologie WDM permet d'améliorer significativement l'utilisation de la bande passante des fibres. Actuellement sur une fibre optique des données peuvent transiter à un débit de l'ordre de plusieurs terabits par seconde. D'autre part depuis que les opérateurs ont fait converger les différents types de trafics (voix, données, vidéo ou *triple play*) sur un même support physique, la majorité du trafic est de type IP. Enfin, ce modèle hérite de la flexibilité et de l'adaptabilité des protocoles de contrôle d'IP.

Cependant, les flux IP sont de débits très faibles par rapport au débit d'une fibre optique et les réseaux IP/WDM ne peuvent pas fonctionner efficacement sans un intermédiaire ayant pour rôle d'agréger les flux IP pour obtenir des flux de débits comparables à ceux des fibres optiques. L'architecture MPLS rempli admirablement cette fonction bien qu'elle n'ait pas été créée pour. Elle permet en effet l'agrégation des flux selon plusieurs niveaux de granularités : des flux de faibles débits sont agrégés en un flux de débit supérieur, puis de tels flux sont eux-mêmes agrégés en un flux de débit encore supérieur etc. Chaque niveau d'agrégation est un niveau de granulartié de flux. L'architecture MPLS gère également la qualité de service et l'ingénierie de trafic mais avec beaucoup plus de flexibilité que les solutions antérieures comme ATM. Les fonctionnalités de SDH comme les mécanismes de tolérance aux pannes sont transmises à la couche WDM grâce à des évolutions technologiques [Liu02, Wei02] en particulier au niveau de la commutation et de la reconfiguration dynamique. Cependant des fonctions comme le formatage des flux pour leur transmission physique ne peuvent pas être prises en charge par le niveau WDM, par conséquent une couche SDH réduite

à quelques fonctions doit subsister. Cette couche nommée *thin* SDH [Gro04] qui sera amenée à disparaître n'a pas d'incidence sur les problèmes d'optimisation que nous étudions dans cette thèse et ne sera pas plus évoquée ici. Dans la suite nous détaillons les principaux composants des réseaux IP/WDM, c'est-à-dire la couche IP, la technologie WDM et l'architecture de réseau MPLS.

2.1.2 Interactions entre les niveaux

Un réseau IP/WDM est composé de plusieurs niveaux qui jouent chacun un rôle spécifique dans l'acheminement des données. Chacun de ces niveaux comporte un *plan de données* et un *plan de contrôle*.

Le plan de données est responsable uniquement de la transmission des données. Le plan de contrôle est responsable d'une part de la découverte et de la connaissance de la topologie du réseau, de la disponibilité des équipements et des moyens existants d'atteindre les autres nœuds. D'autre part, à partir de ces informations, le plan de contrôle est chargé de calculer les routes par lesquelles doivent transiter les données au niveau où il opère. Toutes les informations sont collectées par l'échange de messages de contrôle spécifiques entre les nœuds. Les routes sont également établies, supprimées ou modifiées grâce à la signalisation gérée par le plan de contrôle.

Dans un réseau multiniveaux trois modèles d'interactions entre les plans de contrôle sont étudiés.

- *overlay* : les plans de contrôle des différents niveaux sont indépendants les uns des autres, aucune information n'est échangée. Les calculs de route, les optimisations, la signalisation etc, sont effectués séparément pour chaque niveau et sans tenir compte des autres.
- *augmented* : chaque niveau possède son propre plan de contrôle mais ils utilisent la même signalisation, en particulier les informations sur la disponibilité des connexions. Le routage est tout de même effectué séparément entre les niveaux, mais avec des informations communes.
- *peer* : il existe un unique plan de contrôle pour toutes les couches, cette collaboration améliore les performances globales du réseau puisque tous les niveaux peuvent être optimisés ensembles.

Le modèle overlay est utilisé actuellement, mais les réseaux devraient évoluer vers le modèle augmented puis peer qui permet une gestion plus efficace [Liu02].

2.1.3 Couche IP

Pour un utilisateur donné, Internet est un réseau mondial, transparent, qui interconnecte toutes les machines entre elles et permet l'échange de données. Cette vision correspond à un niveau virtuel du réseau qui en réalité est constitué d'AS hétérogènes utilisant des technologies et des modes de transmissions variés. Le rôle de la couche Internet est d'assurer l'interopérabilité et l'interconnexion de ces AS et de permettre aux données d'être acheminées à travers ces réseaux jusqu'à leur destination. Pour cela elle défini le protocole IP chargé de l'acheminement des données.

Pour être acheminées dans le réseau, les données sont découpées en *paquets* IP. Un paquet IP est composé de deux parties : une partie d'en-tête comportant diverses informations nécessaires à son acheminement, en particulier l'*adresse* IP de destination, et une partie contenant les données.

La transmission des données par le protocole IP est non fiable et se fait sans connexion. En effet il n'y a aucune garantie qu'un paquet arrive à destination, il peut être perdu, dupliqué et plusieurs paquets n'arrivent pas nécessairement à destination dans leur ordre d'émission. De plus l'émetteur envoie des paquets sans prendre contact préalablement avec le récepteur.

Des protocoles de niveau supérieur à IP comme TCP ou UDP [Tan01] mis en œuvre sur les machines des utilisateurs permettent alors de contrôler l'arrivée des paquets IP et éventuellement d'établir des connexions.

Le principe de l'acheminement des paquets IP est très simple. Il est basé sur l'adresse de destination contenue dans chaque paquet. Une fonction *acheminement* se charge, en chaque nœud traversé par un paquet, de déterminer le nœud vers lequel l'envoyer suivant son adresse de destination et les informations fournies par la fonction *routage* du nœud.

Adressage Chaque paquet IP contient une adresse IP de destination. Cette adresse est constituée de deux parties : un identificateur de réseau et un identificateur de la machine de destination dans ce réseau. Une adresse n'identifie pas simplement une machine mais une connexion à un réseau : si une machine est connectée à plusieurs réseaux, elle possède une adresse par réseau. D'autre part un organisme, le NIC (*Network Information Center*) est chargé de distribuer les adresses IP en sorte qu'une adresse ne soit affectée qu'à une unique machine.

fonction *acheminement* L'acheminement d'un paquet repose sur son adresse IP. A la réception d'un paquet, un routeur analyse son en-tête et en particulier son adresse IP de destination afin de déterminer dans quel réseau il doit être livré.

Les informations données par les adresses IP sont exploitées grâce aux *tables de routage* des routeurs IP. La table de routage d'un routeur contient toutes les adresses de réseaux distants existants et l'adresse du routeur auquel transmettre les paquets pour atteindre ces réseaux par un plus court chemin selon une métrique propre au réseau. La table de routage est donc consultée pour chaque paquet arrivant sur un routeur afin de déterminer le prochain routeur (*next hop* ou *prochain saut*) qui traitera à son tour le paquet, c'est la fonction d'*acheminement* du routeur.

fonction *routage* Les tables de routage sont maintenues dynamiquement par des protocoles distribués spécifiques, ou algorithmes de routage, mis en œuvre par la fonction *routage* de chaque routeur. Les protocoles de routage peuvent se diviser en deux classes, les protocoles IGP (*Interior Gateway Protocol*) et les protocoles EGP (*Exterior Gateway Protocol*). Les protocoles IGP sont à l'œuvre à l'intérieur d'un AS et permettent d'effectuer le routage d'un paquet jusqu'à sa destination une fois qu'il a atteint l'AS auquel elle appartient, RIP et OSPF sont deux exemples de protocoles IGP. Les protocoles EGP comme le protocole très répandu BGP, assurent l'interconnexion des AS entre eux, ils permettent la gestion du grand nombre de routes nécessaires pour prendre en compte les AS existants.

La mise à jour des tables de routage se fait suite à l'échange de messages de type *routing update* entre les routeurs qui leur permet de propager des informations sur l'indisponibilité de liens ou de routeurs, l'introduction de nouveaux réseaux et tous les changements topologiques du réseau. Un calcul à partir de ces informations permet de déterminer les nouveaux chemins les plus courts pour atteindre une destination. Les algorithmes de routage permettent donc de réagir dynamiquement aux modifications du réseau.

Plusieurs métriques peuvent être utilisées sur les liens de différents réseaux, il peut s'agir du nombre de routeurs traversé, d'un coût fixé par l'administrateur du réseau en fonction du délai observé sur chaque lien, de la bande passante disponible etc.

La dynamicité induite par les algorithmes de routage permet de tenir compte des changements topologiques du réseau qui peuvent intervenir comme l'indisponibilité d'un lien, une panne de routeur, l'engorgement d'un lien ou l'apparition d'un nouvel élément du réseau etc. Elle a pour autre conséquence que deux paquets provenant d'une même source vers une même destination peuvent ne pas emprunter la même route, c'est-à-dire passer par la même succession de routeurs.

Notons que le routage et l'acheminement du paquet se font simultanément puisque la décision du routeur suivant est prise indépendamment à chaque routeur lorsque le paquet est reçu et il est

réémis immédiatement suivant la décision prise.

Le protocole IP et ses applications sont détaillés dans plusieurs livres, en particulier dans [Tan01, Mél01, Liu02, Puj02].

2.1.4 Technologie WDM

L'opérateur exploitant un réseau de cœur possède un certain nombre de clients à qui il fournit des connexions d'un point à un autre du réseau (qui peuvent être d'autres opérateurs, de grandes entreprises, ...). Ces connexions sont l'agrégation des flux de données de faible débit, flux IP échangés par les utilisateurs d'un réseau WAN par exemple, dans le but de constituer des connexions d'une taille suffisante pour utiliser à bon escient des fibres optiques ayant un débit très élevé.

Nous présentons par la suite la terminologie associée aux réseaux à fibres optiques, ainsi que le principe général de leur fonctionnement. Une description détaillée de cette technologie peut être trouvée dans [GR00, LD02].

2.1.4.1 Multiplexage en longueurs d'onde

Les réseaux à fibres optiques sont constitués de nœuds reliés par des câbles. Un nœud interconnecte plusieurs câbles contenant chacun plusieurs fibres.

Grâce au multiplexage en longueur d'onde (*Wavelength Division Multiplexing* pour WDM), plusieurs longueurs d'onde distinctes peuvent emprunter la même fibre pour transporter à un très haut débit différents flux de données sans interférence. Les noeuds ont la capacité d'émettre sur une même fibre différentes longueurs d'onde réunies en un même signal lumineux mais aussi de séparer les longueurs d'onde composant un signal lumineux reçu. On parle alors de multiplexage et de démultiplexage de ces longueurs d'onde [BCJ+97].

2.1.4.2 Multiplexage temporel

Le multiplexage temporel (TDM pour *Time Division Multiplexing*) consiste à utiliser un même canal, par exemple une même longueur d'onde, pour transporter plusieurs flux indépendants. L'utilisation de ce canal est divisée en périodes de temps et chaque période est elle-même divisée en intervalles de temps ou *slot*. Chaque flux est associé à un slot, et son émission n'est autorisée que cycliquement pendant ce slot à chaque période.

2.1.4.3 Hiérarchie des conteneurs

Les réseaux de cœur permettent de véhiculer des flux agrégés selon une hiérarchie. Un conteneur désigne un flux résultant de l'agrégation ou de l'encapsulation de flux de débit inférieur.

Dans un réseau WDM, le conteneur de plus haut niveau est la fibre optique. Une fibre contient plusieurs bandes qui elles-mêmes contiennent plusieurs longueurs d'onde. La bande de longueurs d'onde a été introduite pour la première fois dans les réseaux en anneau [GRW00, SS99] et cette triple hiérarchie de conteneurs est décrite et utilisée dans [HPS02, LYK+02, YOM03, CAQ04].

Les équipements matériels des nœuds mettant en œuvre cette hiérarchie, notamment avec les opérations *add* et *drop* d'insertion et d'extraction de conteneurs agrégés dans un conteneur de niveau supérieur, sont des ADM pour *Add/Drop Multiplexer*.

Ces équipements peuvent par exemple extraire (*drop*) une longueur d'onde d'une fibre et convertir son signal lumineux en signal électronique pour qu'il soit traité par la couche électronique du nœud (par exemple, la couche IP). Inversement, des flux en provenance de la couche électronique

peuvent être insérés (*add*) dans une longueur d'onde par un ADM, et cette longueur d'onde insérée dans une fibre ou une bande.

Cependant, une longueur d'onde extraite d'une fibre optique n'est pas nécessairement dirigée vers la couche électronique mais peut être multiplexée dans une autre fibre optique, avec d'autres longueurs d'onde. Il s'agit alors d'une opération de *brassage* des fibres (2.1.4.4). Dans ce cas, l'équipement au niveau du nœud ne fait pas nécessairement l'interface avec la couche électronique : il est tout optique et permet d'extraire ou d'insérer une longueur d'onde dans une fibre ou une bande. Un tel équipement est un OADM pour *Optical Add/Drop Multiplexer*.

2.1.4.4 Brassage des conteneurs

Les nœuds d'un réseau WDM assurent une fonction de brassage permettant d'acheminer les conteneurs à travers le réseau jusqu'à destination.

Le brassage de conteneurs consiste dans un premier temps à démultiplexer le contenu de plusieurs conteneurs entrant dans le nœud, ou en d'autres termes à extraire de plusieurs conteneurs des conteneurs de niveau inférieur. Les conteneurs de niveau hiérarchique inférieur ainsi obtenus sont ensuite multiplexés suivant une nouvelle répartition de sortie. Par exemple, le brassage permet d'insérer dans une même fibre en sortie d'un nœud des longueurs d'onde arrivant au nœud par deux fibres différentes.

Le brassage s'appuie sur la *commutation* des conteneurs comparable au principe des aiguillages des lignes de chemin de fer. Les aiguillages sont configurés pour qu'un train passe sur la bonne voie sans avoir besoin de s'arrêter pour préciser sa destination. Il en est de même au niveau optique. Une longueur d'onde est commutée vers une fibre ou une autre grâce à un commutateur optique configurable constitué de lentilles mobiles.

L'OADM réalisant le brassage d'un niveau de conteneur est aussi appelé brasseur optique ou OXC, pour *Optical Crossconnect*. Plus précisément un F-OXC pour *Fiber Optical Crossconnect* permet de brasser des fibres à l'intérieur de câbles. Un F-OXC manipule toutes les longueurs d'onde contenues dans une fibre au sein d'un ensemble global non dissocié, le conteneur "fibre". De même un B-OXC permet de brasser des bandes contenues dans des fibres optiques. Enfin le brassage de longueurs d'onde est réalisé par un W-OXC pour *Wavelength Optical Crossconnect*.

Lorsque le brasseur possède les fonctionnalités de plusieurs de ces OXC spécifiques, c'est-à-dire qu'il permet le brassage de plusieurs niveaux de conteneur et non d'un seul, il est appelé *brasseur hiérarchique* (hierarchical crossconnect, HXC) [HSKO99, LYK+02].

2.1.4.5 Conversion de longueurs d'onde dans un réseau tout optique

Contrairement au brassage de longueurs d'onde qui ne fait que commuter une longueur d'onde vers une fibre de sortie ou une autre, la conversion de longueurs d'onde consiste à émettre le signal véhiculé par une longueur d'onde entrante donnée sur une longueur d'onde différente en sortie d'un nœud. En général, la conversion est assurée par un passage de l'optique vers l'électronique et vice versa. Qu'elle s'effectue entièrement au niveau optique ou qu'elle nécessite un passage au niveau électronique, la conversion de longueur d'onde nécessite des équipements coûteux (ADM), c'est pourquoi il faut éviter d'y avoir recours. De nombreux travaux s'intéressent donc à la réduction du coût des OADM nécessaires dans un réseau [KK99, BCM03a, BCC+05], tandis que d'autres font l'hypothèse de l'absence totale de conversion [ES03]. En effet certaines études montrent que dans la plupart des réseaux réels, la conversion de longueur d'onde n'est pas nécessaire à la bonne exploitation des ressources [JMY05].

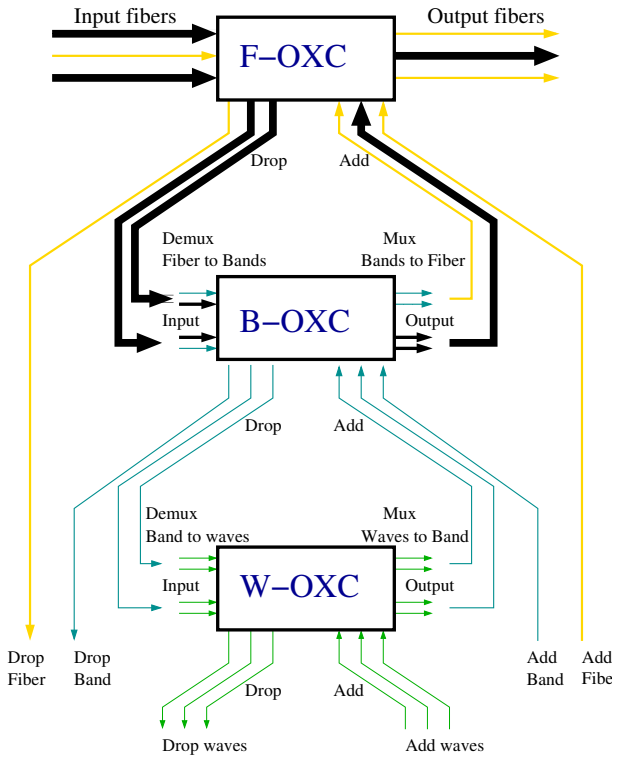


FIG. 2.1 – Schéma détaillé d'un exemple de brasseur à 3 niveaux.

Dans un réseau tout optique, l'absence de conversion de longueurs d'onde signifie que si un signal emprunte une longueur d'onde donnée dans une fibre du réseau, il devra utiliser cette même longueur d'onde sur tout le chemin emprunté. A chaque nœud cette longueur d'onde sera simplement commutée d'une fibre vers une autre sans jamais changer de fréquence optique.

2.1.4.6 Représentation des fonctionnalités des brasseurs

L'encapsulation dans différents niveaux hiérarchiques permet de regrouper des conteneurs d'un niveau donné dans des conteneurs de plus haut niveau. Par exemple au niveau d'un nœud, huit longueurs d'onde peuvent être encapsulées dans une bande et la bande dans une fibre. Pour cela le brasseur équipant le nœud doit posséder les deux composantes W-OXC et B-OXC à la fois. Un nœud peut aussi servir de point d'entrée ou de sortie à des données sur le réseau (*add*/*drop*), il doit alors être équipé d'un convertisseur optique/électronique.

La figure 2.1 présente le modèle détaillé du fonctionnement d'un brasseur adopté dans le cadre du projet RNRT PORTO [ABD+01] impliquant Alcatel, le projet Mascotte (I3S(CNRS/UNSA)/INRIA) et France Télécom. Les termes *fibers, bands, waves* correspondent à fibres, bandes et longueurs d'onde. La fonctionnalité *add* ou *drop* d'un brasseur lui permet d'insérer ou de retirer un signal du réseau. On peut directement insérer une fibre, une bande ou une longueur d'onde dans un des niveaux F-OXC, B-OXC ou W-OXC. Les capacités de multiplexage/démultiplexage sont illustrées par les connexions entre les niveaux deux par deux et par les termes "mux" et "demux". Notons aussi que ce schéma introduit la notion de nombre de ports de multiplexage, c'est-à-dire le nombre de conteneurs d'un niveau pouvant être envoyés au niveau inférieur (et vice versa). Les équipements

de brassage fournis par les équipementiers peuvent en effet varier suivant la taille, à fonctionnalités équivalentes. Il y a alors un lien étroit entre la capacité d'un équipement et son coût.

La figure 2.1 donne un exemple de brassage de deux fibres entrantes, représentées en noir. On extrait de ces deux fibres les bandes au niveau B-OXC, et si l'on suppose que le nombre de bandes extraites n'est pas supérieur à la capacité d'une fibre, on peut alors les regrouper dans une même fibre avant de remonter au niveau F-OXC. La fibre sortante (en noir) peut alors poursuivre son chemin vers un autre nœud du réseau.

Dans la suite, nous utiliserons le terme WDM de façon générique. Dans la pratique on distingue les technologies C-WDM, DWDM, UDWDM qui s'appliquent à différents types de réseau et ont des propriétés différentes, notamment en terme de capacité. Le nombre de longueurs d'onde par fibre peut varier de 8 à 1000 environ. La bande passante d'une longueur d'onde est actuellement de 10Go/s, elle devrait bientôt évoluer vers 40Go/s. En laboratoire elle atteint maintenant 160Go/s.

2.1.5 Architecture MPLS

A l'origine MPLS (*Multi Protocol Label Switching*) a été conçu pour améliorer l'efficacité des routeurs au niveau du traitement des paquets. Au lieu d'être analysés à chaque routeur traversé, les paquets sont analysés une seule fois à l'entrée du réseau et acheminés sur une route prédéfinie grâce à un système d'étiquettes (labels). Ces étiquettes sont de petite taille par rapport aux informations de contrôle ajoutées par chaque couche d'un réseau IP/ATM/SDH/WDM.

MPLS est une architecture de *commutation multiniveaux* qui contrairement à IP permet de séparer les fonctions de routage et d'acheminement des paquets, et ainsi profite de la rapidité d'acheminement des flux de la commutation et de la dynamicité du routage. MPLS s'inspire de technologies comme le *Tag Switching* de Cisco ou de ARIS pour *Aggregate Route-Based IP Switching* d'IBM, et d'ATM [Tan01] dont il généralise certains principes comme par exemple les notions de *circuits virtuels* et de *pile de label*.

Grâce aux évolutions technologiques, notamment l'unification des plans de contrôle entre tous les niveaux du réseau, MPLS comporte aujourd'hui une composante d'*ingénierie de trafic* (TE pour *Traffic Engineering*) qui permet entre autres le maintient de la *qualité de service* (QoS pour *Quality of Service*). MPLS permet également le déploiement facile de *réseaux privés virtuels* (VPN pour *Virtual Private Network*).

L'architecture MPLS est composée d'un certain nombre de protocoles qui peuvent varier et évoluer suivant son domaine d'application. Toutefois les protocoles utilisés n'influent ni sur le principe général de MPLS ni sur les problèmes étudiés dans cette thèse, c'est pourquoi il n'en sera pas fait mention.

Grâce à ses extensions, MPλS (*Muli Protocol Lamdba Switching*) et GMPLS (Generalized-MPLS), MPLS peut être utilisé sur plusieurs technologies de réseaux et en particulier sur les réseaux WDM pour mettre en oeuvre des réseaux de type IP/WDM. Une description détaillée sur les aspects de MPLS abordés dans la suite se trouve dans [RVC01, Liu02, Mél01].

2.1.5.1 Le principe de MPLS

Le principe de MPLS est basé sur le regroupement de paquets partageant des caractéristiques semblables pour leur acheminement au sein de FEC (*Forwarding Equivalence Class*). Ces classes peuvent être formées selon plusieurs critères : même préfixe d'adresse de destination comme pour le routage IP, paquets d'une même application, paquets issus d'un même préfixe d'adresses sources (utilisé pour la mise en oeuvre de réseaux privés virtuels ou VPN), qualité de service demandée, etc.

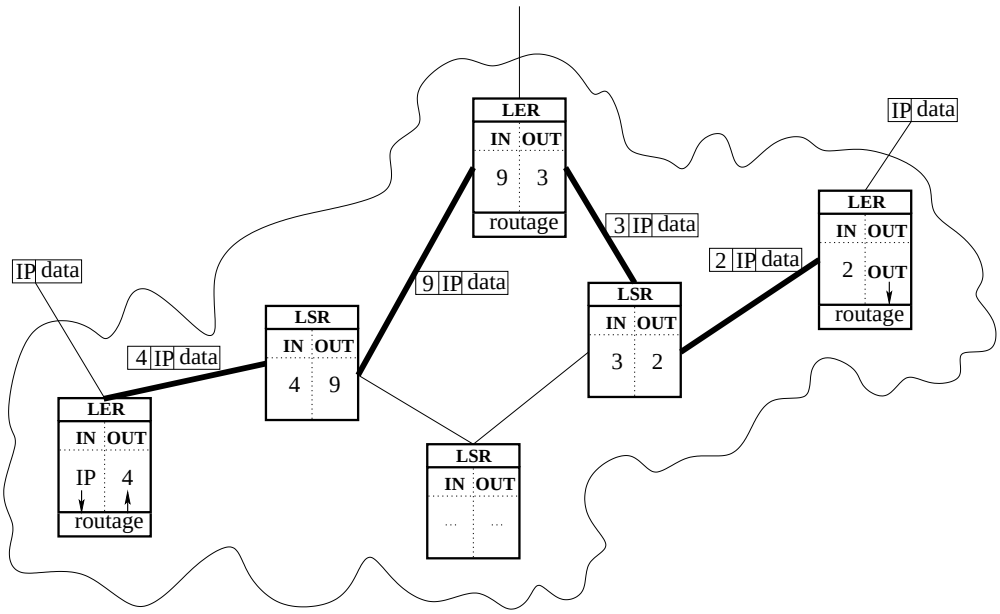


FIG. 2.2 – Acheminement d'un paquet dans un réseau MPLS.

D'autre part, contrairement au routage traditionnel décrit en section 2.1.3 où chaque paquet est analysé à chaque routeur qu'il traverse pour déterminer la prochaine étape de son parcours, avec MPLS le paquet est analysé une seule fois à son entrée dans le réseau MPLS et est immédiatement assigné à une FEC par le routeur d'entrée (LER pour *Label Edge Router* ou encore *Ingress Router*).

Une fois qu'un paquet est affecté à une FEC le LER lui ajoute une étiquette ou label et l'expédie au routeur MPLS (LSR pour *Label Switched Router*) suivant indiqué dans sa *table de transmission de labels* ou *table de forwarding de labels* pour cette FEC.

Le LSR suivant n'a plus qu'à lire l'étiquette des paquets qui lui arrivent et à consulter la table de transmission de labels pour connaître le LSR suivant. Avant de réexpédier le paquet il doit cependant changer son label d'après les informations fournies par la table de transmission de labels. En effet les labels sont locaux à chaque LSR et un LSR doit par conséquent traduire le label pour qu'il ait un sens pour le LSR suivant. Par exemple sur la figure 2.2 le paquet entre dans le domaine MPLS par un LER qui utilise un algorithme de routage pour décider vers quel LSR l'envoyer et avec quel label, ici le label 4. Le LSR suivant qui le reçoit échange le label 4 contre le label 9 après avoir consulté sa table de transmission de label et réexpédie le paquet au LSR suivant.

Notons que la table de transmission de labels possède beaucoup moins d'entrées qu'une table de routage IP habituelle puisqu'elle contient les routeurs voisins d'un LSR au lieu de contenir un nombre potentiellement grand d'adresses de sous-parties du réseau.

Lorsqu'un paquet arrive à un LER (ou *Egress Router*) pour ressortir du sous-réseau MPLS qu'il vient de traverser, le LER lit dans la table que le paquet doit sortir du réseau et lui enlève son étiquette sans la remplacer, ensuite le paquet est routé de manière classique.

La succession de labels reçus par un paquet entre les deux LER est un LSP ou *Label Switched Path*.

2.1.5.2 Empilement de labels

MPLS est une architecture de *commutation multiniveaux*, or jusque là un seul niveau a été évoqué. En fait MPLS met en œuvre la notion de *pile de labels* qui consiste à agréger plusieurs LSP de faibles débits en un seul LSP de débit supérieur en empilant un label supplémentaire commun en en-tête des paquets appartenant aux LSP de faibles débits. Ce nouveau LSP peut ensuite lui même être agrégé avec d'autres LSP sur une partie de sa route par l'empilement d'un autre label commun. Les LSR ne tiennent compte que du label de dessus de pile pour traiter les paquets qui sont acheminés comme décrit en 2.1.5.1. Le principe de la pile de label dans MPLS est en fait comparable à la hiérarchie de conteneurs de WDM. Les conteneurs ne sont plus des fibres, des bandes ou des longueurs d'onde mais tous des LSP distingués non par des caractéristiques physiques mais par des niveaux de labels. Ce principe était déjà utilisé dans ATM mais restreint à deux niveaux. Avec MPLS il n'y a pas de limite conceptuelle sur la taille de la pile de label.

L'empilement et le dépilement des labels en en-tête des paquets sont effectués par les LSR et LER du réseau MPLS grâce aux informations contenues dans leurs tables de transmission des labels. Elles contiennent, en plus du LSR suivant, l'opération à effectuer sur la pile de label d'un paquet d'un LSP donné, empiler ou dépiler, et aussi le label à empiler suivant l'opération.

La figure 2.3 illustre une pile de label avec deux niveaux. L'opérateur 1 dispose de deux domaines de réseau MPLS distants et pour les connecter il passe un accord avec l'opérateur 2 qui lui fournit un chemin entre les deux domaines. Tous les flux allant de l'un des sites vers l'autre sont agrégés dans divers LSP, mais la traversée du domaine de l'autre opérateur se fait par l'encapsulation de tous ces LSP en un LSP de niveau hiérarchique supérieur. Pour cela un label (label 1) est empilé à la sortie du site 1 et dépilé (label 3) à l'entrée du site 2.

La *granularité* d'un flux désigne le niveau hiérarchique du flux. Un flux de granularité fine est un flux de débit peu élevé, c'est-à-dire un LSP de niveau haut, alors qu'un flux de granularité grossière est un flux de débit élevé correspondant à un LSP de bas niveau.

2.1.5.3 Les composants de MPLS, terminologie

L'objectif de cette section est de préciser les définitions et les fonctions des éléments de MPLS évoqués jusque là.

Label Switched Router (LSR) : équipement de type routeur, ou commutateur, capable de commuter des paquets, en fonction des labels qu'ils contiennent. Dans le coeur du réseau, les LSR lisent uniquement les labels, et non les adresses IP.

Label Edge Router (LER) : routeur situé à la frontière du réseau MPLS, également appelé routeur d'extrémité (Ingress et Egress router). Les LER sont responsables de l'assignation et la suppression des labels au moment où les paquets entrent sur le réseau ou en sortent.

Label MPLS : petit en-tête (4 octets) ajouté aux paquets à leurs entrées dans le réseau. Il est utilisé par les LSR lors des décisions d'acheminement des paquets pour lire la table de transmission des labels. Les labels sont locaux entre deux LSR. Un LSR doit traduire les labels reçus en labels dont la signification est commune aux LSR suivants et à lui même. Le format ou la nature du label dépend de la nature du réseau sous-jacent. Par exemple avec MPλS, le label peut être la longueur d'onde sur laquelle arrivent les données 2.1.5.5.

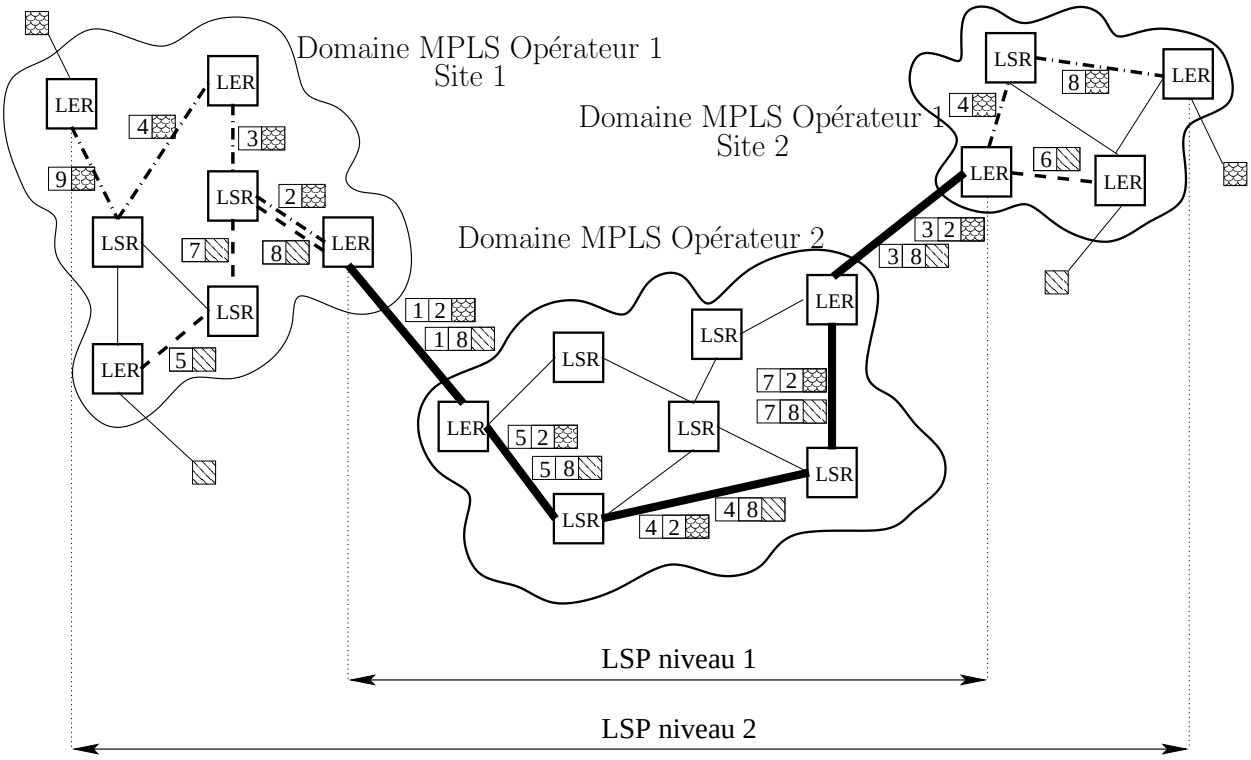


FIG. 2.3 – Empilement de LSP.

Label Switched Path (LSP) Un LSP est un chemin défini entre deux LER d'un réseau MPLS, il est défini par la succession de labels locaux assignés par les LSR à un flux transitant entre les deux LER. Dans l'exemple de la figure 2.2, le paquet représenté suit le LSP défini par la succession de labels 4-9-3-2. Les LSP correspondent aux circuits virtuels d'ATM.

Il existe deux sortes de LSP, les LSP statiques ou ER-LSP pour *Explicitly Routed*-LSP , établis explicitement par un opérateur par exemple pour un client lui-même opérateur, et les LSP dynamiques, établis automatiquement grâce à des protocoles de routage classiques (OSPF, RIP, BGP etc [Puj02]) et un protocole de distribution de labels (LDP).

Table de transmission de labels Une entrée de la table de transmission de labels correspond à un label et contient d'une part le routeur à qui transmettre les données arrivant avec ce label, d'autre part l'opération à effectuer sur la pile de labels. Cette opération peut être soit de remplacer le label en dessus de pile par un label spécifié comme illustré à la figure 2.2, soit de supprimer le label de dessus de pile, ou encore de remplacer le label de dessus de pile et d'empiler un label spécifié supplémentaire (Figure 2.3).

Label Distribution Protocol (LDP) Ce protocole distribue les labels et leurs significations entre les LSR et assure leur cohérence. Il assigne les labels dans les équipements situés aussi bien dans le cœur du domaine MPLS qu'à sa périphérie. Pour cela il s'appuie sur des protocoles de routage classiques comme le fait le protocole IP.

2.1.5.4 Les applications de MPLS

L'unification des plans de contrôle entre tous les niveaux d'un réseau MPLS permet de tenir compte pour le routage de toutes les informations disponibles sur le réseau, ce qui est particulièrement important pour une bonne gestion de la qualité de service et des pannes pouvant survenir, et plus généralement pour mettre en œuvre l'ingénierie de trafic. Les réseaux privés virtuels représentent une autre application importante de MPLS.

Ingénierie de Trafic - Traffic Engineering (TE) L'ingénierie de trafics correspond à l'assignation des flux de trafic sur une topologie physique, selon différents critères. Les applications les plus courantes concernent le routage des flux autour de points de congestion connus dans le réseau et le contrôle précis du reroutage de trafic affecté par un incident sur le réseau. D'une manière générale, le TE a pour objectif l'usage optimal de l'ensemble des liens physiques du réseau en évitant la surcharge de certains liens et la sous-utilisation d'autres. Pour ce faire les ER-LSP constituent un outil essentiel qui permet l'utilisation de routes peu intéressantes pour le protocole de routage à l'œuvre dans le réseau et donc peu utilisées.

Qualité de service - Quality of Service (QoS) Transmettre du son, des données ou des images sur un même réseau implique des caractéristiques différentes, voire opposées. Ainsi le transport du son peut s'accompagner de quelques erreurs de transmission, matérialisées par exemple par des grésillements ou une voix légèrement métallique. L'oreille humaine est en mesure de corriger ces erreurs, mais elle est en revanche sensible à des variations de débit de transmission. À l'inverse, les systèmes informatiques sont plus tolérants à des variations de débit, mais s'accommodent mal d'erreurs de transmission. Il est alors nécessaire que le réseau propose différents paramètres de transmission en fonction des besoins propres à chaque fonction.

La qualité de service d'un réseau désigne sa capacité à transporter dans de bonnes conditions les flux issus de différentes applications. Ceci se traduit par trois caractéristiques techniques essentielles. Le service d'acheminement du réseau doit être fiable et disponible (*reliability*) et doit proposer suffisamment de bande passante (*bandwidth*) pour absorber les trafics générés par les utilisateurs. De plus il doit permettre aux trafics utilisateur qui le désirent un service d'acheminement rapide (latence ou *delay*) et/ou régulier (gigue ou *jitter*) en particulier pour les applications voix. Enfin le service d'acheminement doit assurer aux trafics utilisateur qui le désirent un service sans perte (*loss ratio*).

MPLS propose deux mises en œuvre possibles de la QoS. Sur un même LSP les trafics peuvent être traités différemment par les LSR suivant la QoS qu'ils requièrent. Il est aussi possible de créer plusieurs LSP entre deux LER avec des critères d'acheminement différents, par exemple un LSP peut acheminer des trafics prioritaires avec une garantie de bande passante et de performance pendant qu'un autre LSP achemine des trafics moins prioritaires avec des garanties moins fortes.

Support des réseaux privés virtuels - Virtual Private Network (VPN) Un réseau privé virtuel simule le fonctionnement d'un réseau étendu (WAN pour *Wide Area Network*) privé sur un réseau public comme l'Internet. Afin d'offrir un service VPN fiable à ses clients, un opérateur doit alors résoudre deux problématiques essentielles, d'une part assurer la confidentialité des données transportées, d'autre part prendre en charge des plans d'adressage privés pouvant être identiques entre des réseaux privés distincts.

Grâce au principe des LSP, MPLS répond parfaitement aux problèmes de gestion des adresses. Un ensemble de LSP est établi pour chaque VPN et l'acheminement des flux ne se faisant pas en fonction

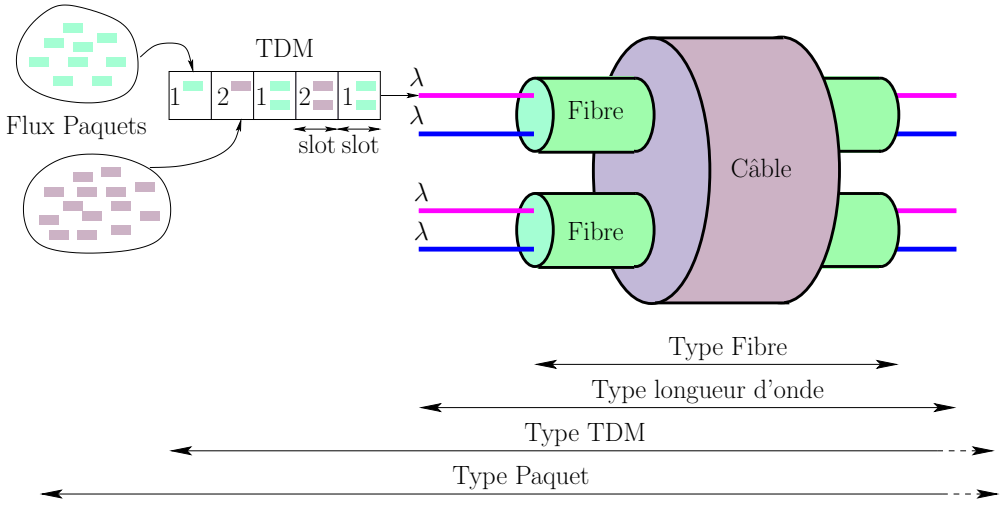


FIG. 2.4 – Hiérarchie de LSP avec GMPLS.

des adresses mais des labels, plusieurs VPN peuvent utiliser les mêmes adresses de sous-réseaux sans interférences.

2.1.5.5 Extension - GMPLS, MPλS

L'architecture MPLS a été conçue pour gérer des flux de type de paquets. Un label est représenté par un en-tête de paquet et doit donc être toujours analysé au niveau électronique pour être acheminé. Un objectif de la première extension, MPλS, de MPLS est justement de permettre la mise en place de LSP tout optique. Avec MPλS, un LSP peut être représenté par la longueur d'onde portant le flux. Un tel LSP n'est pas mis en place par la mise à jour d'une table de transmission de label mais par la configuration physique de commutateurs optiques au niveau des LSR.

Par exemple sur la figure 2.3, tout le trafic allant du site 1 de l'opérateur 1 vers le site 2 emprunte un même LSP symbolisé par la succession de labels 1-5-4-7-3 pour traverser le domaine de l'opérateur 2. En admettant que le débit utilisé sur ce LSP soit comparable à celui d'une longueur d'onde (pour éviter le gaspillage), il est possible grâce à MPλS de mettre en œuvre un LSP optique représenté par une longueur d'onde pour remplacer ce LSP traditionnel entre les deux LER de l'opérateur 2.

La deuxième extension, GMPLS pour *Generalized*-MPLS [Man04, KR05], va encore plus loin puisqu'elle autorise des LSP d'une autre nature fondés sur le multiplexage temporel (TDM pour *Time Division Multiplexing*, Avec le multiplexage temporel un LSP peut être représenté par un slot temporel sur une certaine longueur d'onde.

Ces ex-tentions nécessitent une évolution de tous les protocoles composant MPLS, en particulier les protocoles de signalisation et de distribution de label.

La figure 2.4 résume la hiérarchie des différents types de LSP prévus par GMPLS : type paquet, type TDM, type longueur d'onde et type fibre.

2.2 Architecture des POP d'un opérateur

Les réseaux des fournisseurs d'accès sont en général composés de plusieurs POP interconnectés par des liens à très haut débit, comme représenté par la figure 2.5. Chaque point de présence

est constitué de plusieurs routeurs proches géographiquement. Le cœur du réseau du founisseur relie les différents points de présence. Chaque point de présence fournit par le biais de routeurs d'accès une connexion à des utilisateurs divers tels que des fournisseurs locaux, des entreprises, des serveurs web, etc. Les points de présence de différents fournisseurs sont en général interconnectés par des liens de *peering* privé ou par des points d'accès (NAP pour *Network Access Point*). Les liens de peering privé sont des connexions entre deux fournisseurs exclusivement alors que les NAP interconnectent plusieurs opérateurs présents sur un même site géographique.

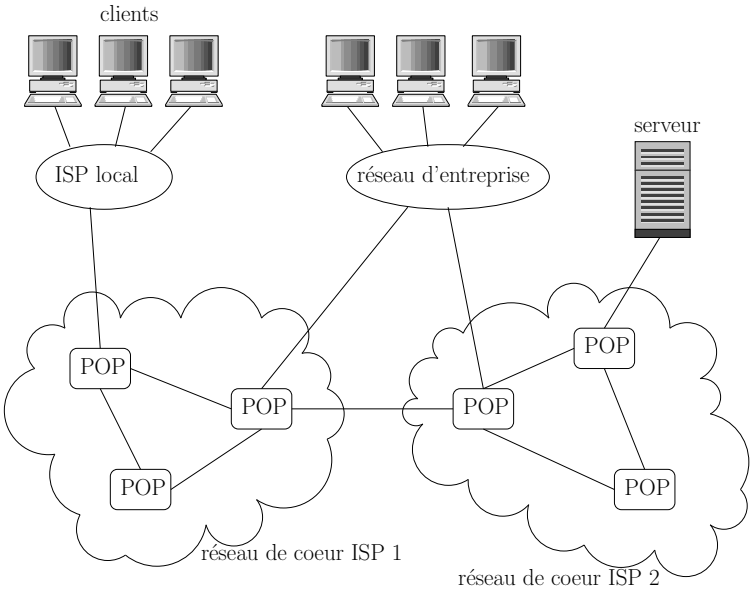


FIG. 2.5 – Architecture d'un réseau de fournisseur d'accès composé de plusieurs POP interconnectés par un réseau à très haut débit.

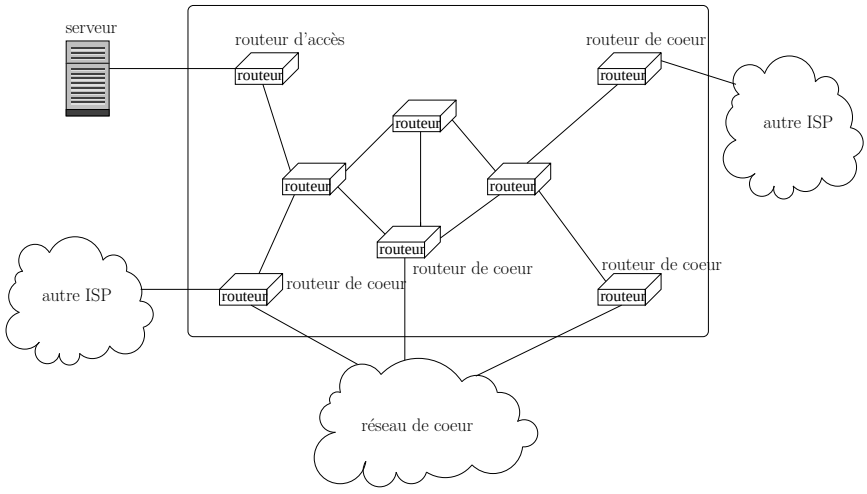


FIG. 2.6 – Architecture d'un point de présence composé de routeurs d'accès et de routeurs de cœur.

En résumé, l'architecture générale d'un point de présence peut être représentée par une structure

hiérarchique à deux niveaux comme indiqué sur la figure 2.6. Au niveau inférieur, les utilisateurs sont connectés au point de présence par des routeurs d'accès eux-mêmes connectés au cœur du réseau via des routeurs de cœur, le réseau de cœur permettant l'interconnexion avec les autres points de présence.

2.3 Modélisation des Réseaux IP/WDM

En général les réseaux sont modélisés par des graphes, les nœuds et les connexions, ou liens, du réseaux sont représentés par les sommets et les arêtes d'un graphe. Cependant les réseaux IP/WDM comportent des connexions de niveaux hiérarchiques différents du fait de l'empilement des labels et de l'encapsulation des longueurs d'onde dans des bandes ou des fibres. Un tel réseau ne sera donc pas modélisé par un seul graphe mais par l'empilement de plusieurs graphes chacun représentant un niveau du réseau.

2.3.1 Réseaux multiniveaux

Un réseau IP/WDM peut se décomposer en plusieurs réseaux empilés correspondant aux niveaux hiérarchiques des LSP. La figure 2.7 illustre un domaine MPLS traversé par cinq LSP de même niveau interconnectant quatre autres domaines. Les LSP sont représentés par des flèches parallèles aux fibres optiques[1] connectant les LSR et LER du domaine E sur lesquelles ils sont routés. Le sous réseau ne possède que deux niveaux, le niveau des fibres optiques et le niveau des LSP.

Chacun de ces niveaux peut être représenté par un graphe indépendamment de l'autre. Pour le niveau fibre la modélisation est immédiate, chaque LSR ou LER correspond à un sommet du graphe, et deux sommets du graphe sont adjacents si et seulement si une fibre connecte les deux LSR/LER associés. Le domaine E de la figure 2.7 donne alors le niveau fibre de la figure 2.8.

De la même façon le niveau LSP de la figure 2.8 est obtenu en associant à chaque LSR/LER du domaine E de la figure 2.7 un sommet et en ajoutant un arc d'un sommet vers un autre s'ils correspondent respectivement au départ et à l'arrivée d'un LSP. Les sommets des deux graphes représentent les mêmes LSR/LER ce qui permet de les superposer.

Cette représentation du réseau par plusieurs graphes empilés se généralise trivialement à des réseaux comportant un nombre quelconque de niveaux de LSP. Avec cette représentation du réseau les routes empruntées par les LSP d'un niveau sur le niveau immédiatement au dessous n'apparaissent pas. Cependant pour décrire certains problèmes d'optimisation qui se posent dans les réseaux multiniveaux la modélisation en graphes empilés est particulièrement adaptée.

2.3.2 Réseaux à deux niveaux

Les réseaux que nous considérons ne comportent que deux niveaux, c'est-à-dire un réseau que nous appellerons *physique* sur lequel est routé un autre réseau que nous appellerons *virtuel*.

Ces réseaux peuvent représenter différents types de liens, des câbles, des fibres, des longueurs d'onde ou plus généralement des LSP, tant que l'empilement des niveaux est cohérent par rapport à la hiérarchie prévue par GMPLS. Par exemple un réseau dont le niveau virtuel représente des fibres alors que le niveau physique représente des longueurs d'onde est à exclure car ce sont les longueurs d'ondes qui sont agrégées dans des fibres et non le contraire.

[1] Dans cet exemple nous parlons de fibres optiques et de LSP pour éviter toute confusion entre les niveaux, il serait cependant plus général de ne parler que de niveaux de LSP, étant donné qu'avec GMPLS les fibres peuvent elles-mêmes être considérées comme des LSP (section 2.1.5.5).

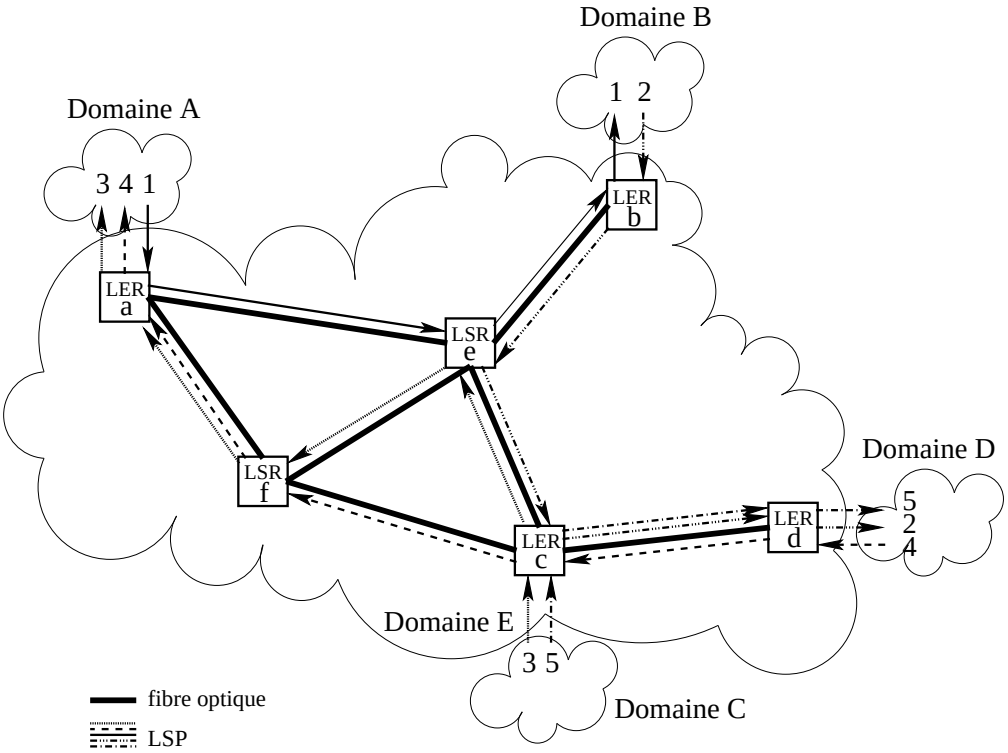


FIG. 2.7 – Réseau à deux niveaux.

Nous modélisons ce réseau par deux graphes représentant chacun un niveau. Suivant le problème étudié, ils pourront être orientés (Chapitre 3) ou non (Chapitre 4), être munis de capacités et/ou de coûts sur les arêtes (ou arcs). Bien que dans MPLS l'établissement d'un LSP ne s'accompagne pas de réservation physique de capacité, lorsque du trafic circule sur un LSP il faut qu'une capacité suffisante soit disponible pour son écoulement. C'est pour cela qu'on parlera dans cette thèse de capacité de LSP. On ne s'intéresse pas en effet à l'aspect protocolaire du routage mais au calcul de chemins sur lesquels la réservation de capacité sera possible.

Dans les problèmes qui sont étudiés dans ces réseaux, des requêtes de bande passante à réserver ou de trafic à écouler sur le niveau virtuel doivent être satisfaites. Un graphe, en général orienté, permet de représenter les requêtes comme un troisième niveau de réseau empilé sur les deux précédents. L'origine d'un arc de ce graphe correspond à un nœud par lequel une certaine quantité de trafic entre dans le réseau, ce trafic doit en ressortir au nœud correspondant à la pointe de l'arc. A chaque arc est associée une valeur exprimant la quantité de trafic, ou la *taille*, qui représente le *volume* de données à acheminer de la source à la destination de la requête correspondante. L'unité de la taille d'une requête est en général une unité de débit (Ko/s, Mo/s etc).

Les requêtes entre les nœuds du réseau peuvent être des flux de paquets IP à acheminer, des longueurs d'onde à réserver ou des LSP à router sur un niveau hiérarchique de LSP inférieur, suivant à quel niveau se place l'étude. Pour un opérateur louant de la bande passante à des opérateurs concurrents c'est en terme de longueur d'onde que s'exprime la demande, alors que pour un fournisseur d'accès à Internet, la demande de trafic à écouler provient directement de particuliers et pourra s'exprimer en flux de paquets de débit inférieur à la longueur d'onde.

Enfin, les requêtes sont toujours routées sur le niveau virtuel, c'est-à-dire le niveau le plus proche

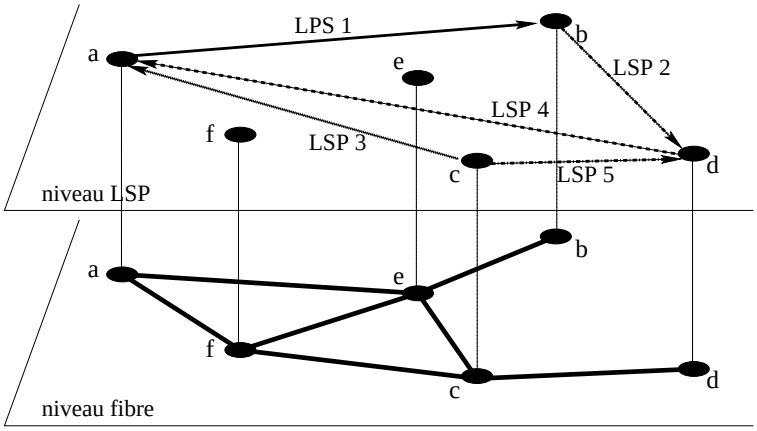


FIG. 2.8 – Représentation d'un réseau à deux niveaux par deux graphes superposés.

des utilisateurs, quels que soient ces utilisateurs (opérateurs, particuliers etc).

2.3.3 Modélisation des flux

Dans les réseaux IP/WDM, deux types de flux doivent être considérés : les flux IP constitués de paquets indépendants et les flux agrégés au sein de LSP, longueurs d'onde, fibre etc. Ces deux sortes de flux présentent des caractéristiques très différentes et ne peuvent pas se modéliser de la même façon.

Flux de paquets Deux propriétés du trafic IP permettent de le modéliser par un flot fractionnaire, c'est-à-dire un flot de taille requise pouvant être routé sur plusieurs chemins distincts chacun portant seulement une partie du flot. D'une part les paquets IP d'un même expéditeur vers un même destinataire peuvent être acheminés sur des routes différentes suivant l'algorithme de routage et l'état du réseau. D'autre part la taille d'un paquet IP étant négligeable par rapport au trafic total circulant sur le réseau, aux débits possibles des LSP, des longueurs d'onde ou des fibres, le trafic IP peut être considéré continu et non pas la somme de quantités discrètes.

Flux agrégés Les flux agrégés sont représentés par des chemins, les LSP, dans les réseaux IP/WDM. Tous les paquets circulant via un LSP reçoivent le même label et sont transportés sur la même route physique. Bien que dans MPLS l'établissement d'un LSP ne s'accompagne pas de réservation de capacité, lorsque du trafic arrive sur un LSP il faut qu'une capacité suffisante soit disponible sur le réseau physique pour son écoulement. Dans cette thèse étant donné que l'on s'intéressera à la planification de LSP et non pas aux aspects protocolaires, on s'autorisera à parler de taille de LSP ou de capacité de LSP suivant s'il représente une requête du problème ou un des niveaux du réseau.

Il apparaît donc naturel de modéliser un LSP ou un flux agrégé par un flot monorouté, c'est-à-dire un flot indivisible de taille ou capacité donnée circulant sur un unique chemin dans le réseau.

Suivant le modèle de flot les contraintes des problèmes étudiés sont différentes et ces problèmes peuvent être NP-difficiles ou polynomiaux.

2.3.4 Modélisation d'un POP

Pour les problèmes de surveillance du trafic evoqués en introduction, nous n'avons pas besoin de considérer l'aspect multiniveaux des POP puisque ces problèmes concernent l'installation d'équipements physiques sur les routeurs, seul le niveau physique est à prendre en compte. Un POP est donc modélisé par un graphe non orienté dont les sommets représentent les routeurs du POP tandis que les arêtes représentent les câbles physiques interconnectant les routeurs au sein du POP.

2.4 Tolérance aux pannes dans les réseaux multiniveaux

Certains services proposés par les opérateurs de télécommunication nécessitent une disponibilité permanente du réseau. Or en pratique des pannes surviennent régulièrement et sur n'importe quelle couche. C'est grâce aux mécanismes de protection et de restauration que les services ne sont pas interrompus.

Dans les réseaux multiniveaux actuels, chaque niveau dispose de son propre mécanisme de protection. Or lorsque la protection est planifiée indépendamment pour chaque couche, elle induit souvent une baisse des performances du réseau suite à une réservation de bande passante de protection désorganisée (redondances etc). Parfois même les mécanismes de protection sont dans l'incapacité de rétablir le trafic [LT02, CB00, DY01].

C'est pourquoi depuis quelques années des mécanismes de protection unifiés où les différents niveaux coopèrent sont étudiés [ZD02, DGA+99], notamment avec GMPLS.

Les stratégies de protection classiques des réseaux à un seul niveau sont toujours employées dans les réseaux multiniveaux pour trouver des routes de secours. Cependant la protection dans ces réseaux nécessite la définition précise du rôle de chaque niveau et leurs interactions dans la prise en charge d'une panne ainsi que la gestion des ressources de secours dont chacun peut disposer.

2.4.1 Les pannes

Une panne peut survenir sur tout élément ayant une existence matérielle dans le réseau. Il peut s'agir d'un câble contenant plusieurs fibres coupé par erreur lors de travaux sur une route, d'un émetteur d'une certaine longueur d'onde défaillant, d'un routeur dans un bâtiment incendié ou encore d'un composant électronique nécessaire à l'encapsulation des paquets IP dans un LSP. Certaines pannes sont dues à des opérations planifiées de maintenance du réseau. Des mesures [MIB+04, ICM+02] faites sur le réseau IP de Sprint ont montré que ces interruptions planifiées représentaient 20% des pannes et que parmi les pannes non planifiées 30% affectent des ressources (câble, routeur etc) qui provoquent l'indisponibilité de plusieurs liens simultanément. Les 70% de pannes restantes n'affectent qu'un seul lien à la fois.

Ainsi les pannes peuvent se produire à n'importe quel niveau du réseau, aussi bien au niveau des fibres optiques qu'à celui des longueurs d'ondes et des LSP, et aussi bien sur les liens que sur les nœuds. Avec la modélisation des réseaux par une superposition de graphes, une panne se traduit par la disparition d'un nœud ou d'une arête sur n'importe lequel de ces graphes.

2.4.2 Mécanismes de survie classiques

Deux façons opposées mais complémentaires d'aborder une panne coexistent dans les réseaux à un seul niveau : la restauration et la protection. Chacune peut être utilisée pour rétablir le trafic en cas de panne à un niveau donné dans un réseau multiniveaux. Les niveaux concernés par les

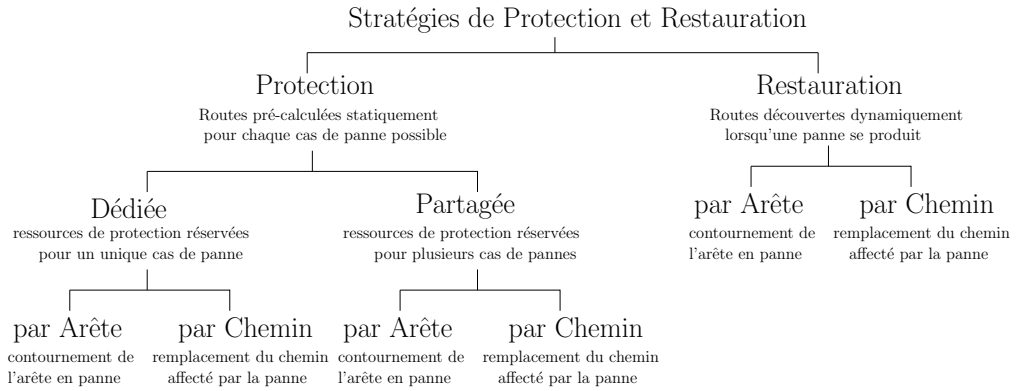


FIG. 2.9 – Une classification des modes de protection et restauration [RM99a].

mécanismes suivant sont les niveaux virtuels et le niveau physique, au niveau IP le routage évolue en fonction de la topologie du réseau et prend ainsi en compte implicitement les pannes.

2.4.2.1 Restauration

La restauration consiste à rerouter dynamiquement des connexions lorsqu'une panne survient sur le réseau. On doit alors calculer, au moment de la panne, un nouveau routage à partir des ressources disponibles. On parle d'algorithmes *online* puisqu'ils ne répondent pas à un problème statique ou connu à l'avance. Dans la suite de cette thèse nous ne traitons pas le problème de la restauration, mais celui de la protection. De nombreux articles traitent de la restauration comme [RM99b].

2.4.2.2 Protection

Le principe de la protection est de prévoir à l'avance tous les cas de pannes pouvant survenir sur le réseau afin de mettre en place des solutions de secours assurant la continuité du trafic. Il s'agit par exemple de proposer des *chemins de protection* sur lesquelles pourra être acheminée une requête lorsque son *chemin principal*, celui sur lequel elle est routée par défaut quand il fonctionne, est affecté par une panne.

Les premiers travaux portant sur la protection se plaçaient dans le cadre de réseaux à un seul niveau et faisaient en général l'hypothèse d'une unique panne de lien. Plusieurs modes de protection ont été envisagés, une classification en est donnée en particulier dans [RM99a] reproduite par la figure 2.9, une version plus détaillée est présentée dans [MM00].

Protection par reroutage global Le reroutage global consiste à prévoir un routage admissible pour chaque cas de panne possible. Pour chaque routage, une certaine capacité est nécessaire sur un câble du réseau. Pour assurer le routage de l'ensemble des requêtes quelle que soit la panne qui se produise, la capacité qui doit être disponible sur chaque lien du réseau est la capacité maximum utilisée sur ce lien parmi tous les cas de pannes possibles.

L'inconvénient direct d'une telle politique de protection vient du fait qu'entre l'état sans panne et un état de panne donné, aucune garantie n'est donnée quant à l'emplacement des routes principales et des changements à opérer. Dans le pire des cas, toutes les routes principales sont à modifier, provoquant un impact d'ordre technique dans la configuration des nœuds et un délai pouvant être

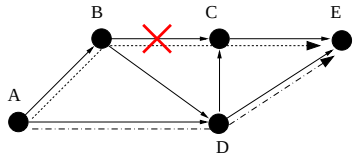


FIG. 2.10 – Protection 1 : 1 d'une requête *AE* pour la panne du câble AB.

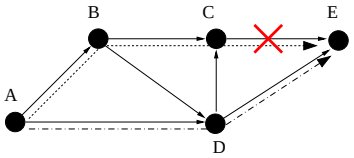


FIG. 2.11 – Protection 1 : 1 d'une requête *AE* pour la panne du câble CE.

important avant la mise en place des chemins de protection. Le passage d'un routage à un autre dans un réseau a été étudié dans [CPPS05] d'un point de vu algorithmique.

Protection par arête ou par chemin La protection par arête consiste à déterminer et réserver un chemin contournant chaque arête du graphe représentant le réseau, ainsi lorsque la panne se produit sur une arête donnée le trafic utilise le chemin qui la contourne. La protection par chemin consiste à prévoir pour chaque requête deux chemins, un chemin principal utilisé quand le réseau ne subit aucune panne, et un chemin de protection (*backup*) qui n'est utilisé que lorsque le chemin principal est affecté par une panne. En général la protection par chemin est préférée car elle utilise beaucoup moins de capacité que la protection par arêtes même si son délai de mise en place est plus long [IMG98]. En effet, une fois la panne détectée par un nœud, il diffuse cette information et les nœuds extrémités d'un chemin principal doivent la recevoir avant de pouvoir activer le chemin de secours. Plus le chemin est long plus il faut de temps pour que l'information leur arrive.

Il existe plusieurs variantes de la protection par chemin visant à réduire le délai de mise en place du chemin de secours. Avec la protection *local to egress* [SR06] en cas de panne, le chemin principal est utilisé jusqu'au nœud précédent la panne et le chemin de secours est calculé entre ce nœud et la destination du chemin. Le délai est réduit puisque c'est le même nœud qui détecte la panne et commute sur le chemin de protection. La protection par *segment* [XXQ03a] consiste à découper le chemin principal en plusieurs segments et à protéger chaque segment par un segment de secours, indépendamment les uns des autres. Lorsqu'une panne affecte un segment, l'information doit être transmise au nœud origine du segment qui commute alors sur le segment de protection, et réduit ainsi le délai de mise en place de la solution de secours.

Protections dédiées et partagées Les protections dédiées et partagées ne nécessitent pas un reroutage total en cas de panne. Il s'agit au contraire de ne rerouter que les chemins principaux touchés par la panne sur des chemins de secours. La protection dédiée nécessite d'allouer un chemin de secours qui ne peut être réutilisé dans un autre contexte. À l'inverse, la protection partagée permet d'utiliser une même ressource pour deux chemins de secours ou plus qui ne pourraient être activés en même temps. Notons que le reroutage global fait partie de la protection partagée, tout le réseau est partagé. D'autre part, les protections par chemin et par arêtes se déclinent en protection dédiée chemin et protection partagée.

Il existe deux façons de mettre en œuvre la protection dédiée. La protection $1 + 1$ consiste à envoyer la même information sur deux chemins disjoints, le chemin principal et le chemin de protection, en même temps. Au niveau du nœud de destination, le signal est reçu en double, garantissant la réception d'au moins un signal en cas de panne. La protection $1 : 1$ dédiée réserve un chemin de secours pour chaque chemin principal. En cas de panne, le chemin de secours est activé mais le signal n'est jamais reçu en double à destination.

Dans le cas de la protection partagée on parle aussi de protection $1 : 1$. Dans ce cas, les chemins

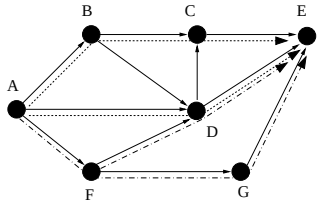


FIG. 2.12 – Protection 2 : 2 d'une requête AE de taille 2

de secours peuvent partager des ressources entre eux comme illustré par les figures 2.10 et 2.11 où l'on protège le chemin (A, B, C, E) pour deux cas de pannes possibles. Pour ces pannes, on utilise le même chemin de secours, (A, D, E), qui est dit *partagé*.

Plus généralement, pour plus de flexibilité, on utilise la protection $M : N$. Pour une même requête, M chemins principaux sont protégés par N chemins de secours. Les N chemins de secours peuvent partager des ressources avec d'autres chemins de secours, pour la même requête ou pour une requête différente, qui ne peuvent s'activer pour la même panne. La figure 2.12 montre le cas d'une protection du type 2 : 2. Les chemins principaux, en pointillés, sont protégés par les chemins en pointillés discontinus. On note alors que sur le câble AF, on peut partager la capacité de protection puisque les chemins principaux ne peuvent tomber en panne en même temps.

Notons enfin que pour les protections 1 : 1 et $M : N$, les ressources réservées pour les chemins de protection ne sont pas utilisées. En pratique, les opérateurs font circuler sur ces canaux des flux non prioritaires. Ces flux peuvent être interrompus et remplacés par des flux de protection, le temps que la situation revienne à la normale.

Les protections dédiées sont très coûteuses en terme de ressources puisque pour chaque connexion deux chemins sont réservés. La protection partagée permet de réduire nettement les ressources nécessaires, c'est pourquoi elle est très étudiée. Une variante de la protection partagée est la protection *Demand-wise Shared Protection* ou DSP qui consiste à diviser la bande passante requise par une connexion sur plusieurs chemins en sorte que quelle que soit la panne qui survienne, elle n'affecte qu'un certain pourcentage maximum de la bande passante utilisée par la connexion [HJK+06]. Cette méthode permet de réduire encore les besoins en ressources par rapport à la protection partagée classique.

Protection par chemin dépendante ou indépendante de la panne La protection indépendante de la panne ne prévoit qu'un seul chemin de secours pour chaque requête qui doit permettre le reroutage de la requête quelle que soit la panne se produisant sur le chemin. Les deux chemins, principal et protection, doivent donc être disjoints [LTS05].

Pour la protection dépendante de la panne, le chemin de protection sur lequel sera commutée la requête en cas de panne dépend de l'arête qui tombe en panne. Ainsi le chemin de protection pour la panne de l'arête i peut utiliser toutes les arêtes encore en fonctionnement du chemin principal, sauf l'arête i et il peut y avoir autant de chemins de protection pour une même requête que de liens dans son chemin principal.

Notons qu'en théorie les protections dépendante et indépendante de la panne peuvent être soit dédiées soit partagées bien que la protection dédiée dépendante de la panne induise une réservation de capacité potentiellement très supérieure à la protection dédiée indépendante de la panne, et ne présente donc que peu d'intérêt.

Protection différenciée La protection différenciée consiste à donner des priorités aux types de flux circulant dans le réseau. Plus un flux possède une priorité élevée plus il est important qu'il ne soit pas interrompu en cas de panne. Des chemins de secours doivent être prévus et des ressources suffisantes doivent être réservées pour continuer à acheminer les flux de priorité maximum quoi qu'il se produise dans le réseau. Par exemple les flux correspondant à des opérations chirurgicales à distance ou des visio-conférences ont des priorités maximum. Les flux de priorité minimum sont des flux qui ne nécessitent pas d'être protégés en cas de panne comme par exemple les flux correspondant au transfert de courriers électroniques.

Protection pour pannes multiples Comme le suggèrent les résultats de [MIB+04] il arrive que plusieurs pannes indépendantes se produisent en même temps. En général les travaux portant sur ce sujet se limitent à deux pannes simultanées. Des stratégies de protection ont étés proposées aussi bien pour le niveau WDM [CSC02, SP03, CSC04, GDK+06, PG06] que pour le niveau IP [CHGL05].

La stratégie de [CHGL05] consiste à décomposer d'une manière particulière le réseau en plusieurs réseaux connexes ouvrant chacun tous les nœuds. Les routes IP doivent être calculées dans ces réseaux, ainsi lorsque plusieurs pannes se produisent sur des éléments (nœuds ou liens) du même réseau il suffit d'utiliser les routes fournies par les autres réseaux pour assurer la connectivité.

Au niveau optique, un mode particulier de protection est la protection par des p-cycles. Le graphe est décomposé en cycles de manière à ce qu'une arête appartienne à un cycle ou soit une corde d'un cycle. Lorsqu'une arête tombe en panne, les longueurs d'onde affectées sont commutées sur la section en fonctionnement du cycle de protection de l'arête indisponible [GDK+06, PG06, KSG04, Mau03, SGA02, GS98].

Comparaison des modes de protection pour un seul niveau A partir des principes de protection précédents, de nombreux algorithmes ont étés proposés et comparés entre eux ou à des méthodes de résolution basées sur la programmation linéaire (en nombre entier) pour déterminer les plus efficaces [EBR+03]. De même, les différents modes de protection (et de restauration) ont été comparés [XM02] suivant des critères d'utilisation des ressources [DW94] ou de délai de mise en service des chemins de secours comme dans [RM99a] et [RM99b], ou encore du point de vu de critères de disponibilité du réseau [HJK+06]. Cependant cette thèse ne concerne pas ces aspects de la tolérance aux pannes mais certains problèmes spécifiques qui apparaissent dans les réseaux multiniveaux.

2.4.3 Particularités des réseaux multiniveaux

Plusieurs questions qui ne se posaient pas dans le cadre des réseaux à un seul niveau sont incontournables dans le cas des réseaux multiniveaux. Ces questions concernent le rôle de chacun des niveaux dans la prise en charge des pannes ainsi que dans la gestion des ressources mais également les propriétés de la topologie de chaque niveau du réseau.

2.4.3.1 Niveau de traitement d'une panne

Lorsque le réseau est composé de plusieurs niveaux et qu'une panne survient sur l'un d'eux il existe plusieurs façons d'envisager la protection indépendamment des stratégies classiques évoquées précédemment en section 2.4.2.

Il est généralement admit que chaque niveau du réseau doit être pourvu d'un mécanisme de protection qui peut être basé sur les stratégies classiques. Si seul un niveau possède un mécanisme de protection, le réseau ne sera que rarement en mesure de supporter efficacement des pannes

[DGA+99]. Cependant si chaque niveau possède un mécanisme de protection, il est nécessaire de déterminer à quel niveau et comment doit être prise en charge une panne donnée. Plusieurs stratégies sont résumées dans [DGA+99, TH01].

Recovery at the lowest layer La protection est assurée au niveau le plus proche de l'origine de la panne, si possible dans le niveau de la panne. Le routage est simple car le trafic est agrégé à un niveau de granularité proche de celui de la panne. Le nombre de connexions à rerouter est donc d'un ordre de grandeur gérable par le niveau qui met en œuvre la solution de secours. Cependant si la protection est déclenchée à un niveau trop inférieur au niveau de la panne, des connexions non affectées par la panne risquent d'être reroutées également. Ceci induit une mauvaise utilisation des ressources du réseau. Par exemple si une panne de longueur d'onde est protégée au niveau des fibres, la seule solution pour rerouter le trafic qui l'utilise est de rerouter toute la fibre optique concernée. Il faudra donc réserver beaucoup plus de capacité que ce qui est nécessaire, c'est-à-dire une fibre complète au lieu d'une seule longueur d'onde. D'autre part déclencher la protection au bon niveau nécessite un système de communication complexe entre les niveaux qui peut induire un certain temps d'adaptation du réseau à la survenue d'une panne. En effet lorsqu'une panne se produit sur le niveau WDM, les niveaux supérieurs détectent une panne sans savoir à quel niveau elle s'est produite et risquent donc de déclencher leurs propres mécanismes de protection.

Recovery at the highest layer La protection est assurée au niveau le plus proche de l'origine du trafic, c'est-à-dire au plus haut niveau. La communication entre les niveaux est réduite puisque c'est toujours le niveau supérieur qui prend les pannes en charge. De plus, il est beaucoup plus facile de mettre en œuvre une protection spécifique suivant les types de trafics et leurs degrés de priorité, puisqu'à ce niveau ils ne sont pas encore agrégés. Cependant protéger uniquement au plus haut niveau, le niveau IP, n'est pas vraiment envisageable puisqu'une coupure de fibre optique (plusieurs Tb/s) impliquerait que tout le trafic de cette fibre soit géré par la couche électronique des routeurs, or ils n'ont pas une capacité de calcul et de stockage suffisante pour traiter autant de données efficacement. D'autre part lors de la phase de conception des chemins de secours, il faut s'assurer que les chemins de protection n'utilisent pas les mêmes ressources que les chemins principaux, sinon ils seraient tous coupés par une unique panne de la ressource partagée.

Recovery at multiple layers La protection est distribuée sur plusieurs niveaux, pour combiner les avantages des deux solutions précédentes. Lorsqu'aucune coordination entre les niveaux n'est prévue, il est possible que plusieurs d'entre eux mettent en œuvre leur stratégie de protection simultanément ce qui peut conduire à une mauvaise utilisation des ressources, ou pire à un routage instable des connexions. Les mécanismes de protection des différents niveaux doivent donc être déclenchés séquentiellement. Il existe deux façons de procéder :

- *bottom-up* : le mécanisme de protection du niveau le plus bas est déclenché, s'il échoue à restaurer tout le trafic, le niveau supérieur prend la relève.
- *top-down* : le mécanisme de protection du niveau le plus haut est déclenché, s'il échoue à restaurer tout le trafic, le niveau inférieur prend la relève. Avec cette stratégie il est plus facile d'effectuer une protection différenciée suivant le type de trafic, mais comme les niveaux bas ne sont pas toujours en mesure de détecter si un niveau supérieur est capable de restaurer le trafic, une signalisation particulière entre les niveaux est nécessaire.

Un niveau détermine s'il doit prendre le relais soit à l'aide d'une minuterie déclenchée au moment de la panne, soit lorsqu'il en reçoit le signal directement du niveau qui a échoué, soit enfin lorsqu'un

mécanisme de contrôle unique pour tous les niveaux le lui indique, dans l'hypothèse où un tel mécanisme existe. Selon [SPD+06] cette stratégie est la plus prometteuse.

2.4.3.2 Utilisation des ressources

Comme dans le cas à un niveau, la prise en charge des pannes dans un réseau multiniveaux ne peut se faire que s'il reste dans le réseau des ressources disponibles pour mettre en œuvre les stratégies de protection choisies.

Lorsque chaque couche dispose d'un mécanisme de protection propre, l'utilisation de la capacité du réseau peut être mauvaise indépendamment du mode de décision du niveau de traitement des pannes et de la stratégie de protection. Supposons qu'un niveau supérieur A réserve une capacité c_A pour sa propre protection à un niveau inférieur B, ainsi qu'une capacité $c'_A = c_A$ pour le fonctionnement normal du réseau. Ce niveau B fait de même pour protéger l'ensemble de la capacité utilisée par les niveaux qui lui sont supérieurs. Il réserve donc sur un niveau encore inférieur C une capacité c_B pour sa protection et une capacité $c'_B = c_B$ pour le cas de fonctionnement normal du réseau. Ainsi au niveau B, la capacité c_A est réservée deux fois : une fois pour le fonctionnement normal du réseau, c'_A et une fois pour la protection du niveau A, c_A. Par conséquent au niveau C, la capacité utilisée au niveau A est réservée quatre fois : deux fois avec c'_B pour le fonctionnement normal du niveau B, et deux fois avec c_B pour le cas de panne.

Pour éliminer ce problème et diminuer les capacités réservées pour la protection par chaque couche, l'idée de *common pool of capacity* détaillée dans [DGA+99, TH01] est de traiter différemment au niveau inférieur les capacités demandées par le niveau supérieur selon leurs fonctions. Par exemple les capacités réservées pour la protection au niveau supérieur ne sont pas protégées au niveau inférieur. Plus généralement cette idée consiste à partager les capacités de protection entre tous les niveaux ce qui permet une meilleure utilisation des ressources.

2.4.3.3 Groupe de risque (SRRG)

Sur un niveau, le principe général de la protection peut se résumer à trouver entre deux nœuds deux chemins disjoints (ou plus), si l'un des deux est affecté par une panne, l'autre est utilisé. Il peut s'agir de deux chemins protégeant une connexion de bout en bout, ou bien d'un chemin protégeant une arête, ou d'un cycle etc. Lorsque le réseau comporte plusieurs niveaux, comme nous l'avons vu précédemment, chaque niveau peut disposer d'une stratégie de protection basées sur les méthodes classiques consistant donc à trouver des ensembles de chemins disjoints.

Au niveau virtuel du réseau de la figure 2.13, il existe deux chemins disjoints $\{A, F, E, I\}$ et $\{A, H, I\}$ qui permettent de router la connexion $\{A, I\}$ présente au niveau des requêtes. Cependant au niveau physique les liens $\{E, I\}$ et $\{A, H\}$ sont routés tous les deux sur le lien physique $\{F, G\}$, le routage du niveau virtuel est indiqué sur le niveau physique par les courbes reliant les extrémités des liens virtuels. Par conséquent en cas de coupure du lien physique $\{F, G\}$, les deux liens virtuels $\{E, I\}$ et $\{A, H\}$ sont indisponibles et la connexion $\{A, I\}$ est interrompue.

Par conséquent dans un réseau multiniveaux, lorsque deux liens d'un niveau virtuel semblent disjoints, il est possible qu'en réalité ils utilisent à un niveau inférieur une ressource commune. Dans le cas où cette ressource commune tombe en panne, les deux liens virtuels tombent en panne simultanément.

Dans un niveau virtuel un ensemble de liens qui utilisent au niveau physique une même ressource (nœud ou lien physique) appartiennent à un même groupe de risque ou SRRG pour *Shared Risk Resource Group*. Tous les liens du groupe correspondant à une ressource du niveau physique tombent en panne en même temps lorsque cette ressource tombe en panne [PPJ+01, DG02, YVJ05, DS04a].

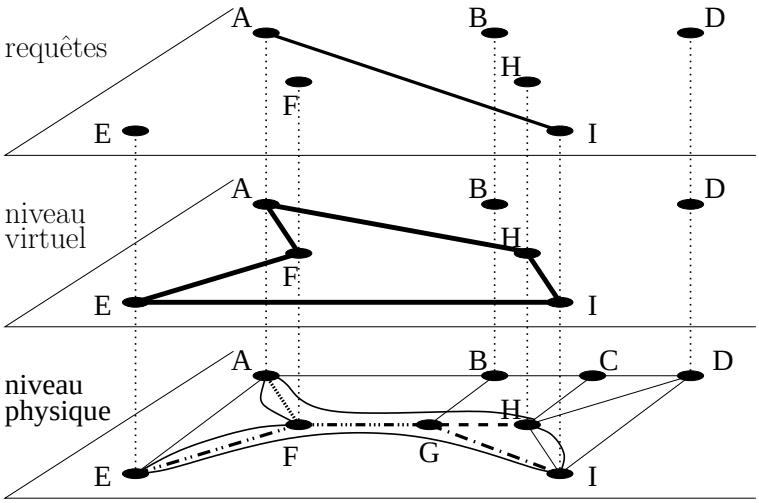


FIG. 2.13 – Exemple de réseau multiniveaux.

2.5 Conclusion

Ce chapitre était consacré à la description des réseaux de télécommunication d'où proviennent les problèmes d'optimisation que nous approfondissons dans cette thèse. Ces réseaux sont constitués d'une partie composée de réseaux d'accès auxquels sont connectés les utilisateurs, et d'une partie de réseaux de cœur interconnectant les réseaux d'accès. Nous considérons les réseaux multiniveaux de type IP/WDM utilisant une architecture MPLS pour lesquels nous proposons une modélisation par des graphes. La tolérance aux pannes est une qualité importante de ces réseaux. Elle peut être mise en œuvre par deux techniques complémentaires : la restauration et la protection dont nous rappelons les principes. Nous présentons également les techniques de protection proposées dans la littérature dans le cas des réseaux à un seul niveau ainsi que les nouvelles questions soulevées par une protection pensée pour plusieurs niveaux simultanément. Parmi ces questions, les groupes de risque induits par l'empilement des niveaux de réseau sont à l'origine de problèmes d'optimisation intéressants.

Chapitre 3

Conception de réseau virtuel et Groupage

Les problèmes de conception de réseaux virtuels englobent tout un ensemble de problèmes d'optimisation qui diffèrent par les contraintes imposées, et parfois par les données considérées. L'étude de ces problèmes est très importante pour les opérateurs. De la qualité du réseau virtuel dépendent le fonctionnement et les performances du réseau et, par suite, la satisfaction des clients, mais aussi le coût du réseau en termes d'équipements et d'exploitation. Pour un opérateur, il s'agit donc de mettre en place un réseau virtuel permettant de satisfaire toutes les requêtes de ses clients avec des exigences de qualité de service et de fiabilité, tout en maintenant les coûts faibles.

Dans la première section de ce chapitre, nous présentons un problème de conception de réseau virtuel tolérant aux pannes du niveau physique. Nous employons le mode de protection par chemin, dépendant de la panne et partagé. Chaque chemin de secours protège au niveau physique un lien virtuel pour un unique cas de panne. Plusieurs chemins protégeant un même lien virtuel peuvent partager des ressources physiques. Par contre, des chemins de secours protégeant deux liens virtuels distincts ne peuvent partager aucune ressource. Nous proposons une modélisation en programme linéaire mixte (MILP) de ce problème. Nous dégageons ensuite un sous-problème, le *groupage* auquel est consacré le reste du chapitre.

Le groupage de trafic (ou *grooming*) est le terme générique utilisé pour le problème qui consiste à agréger des flux de faible débit dans des flux de plus gros débit. Dans les réseaux IP/WDM il s'agit de grouper des LSP dans des LSP de niveau inférieur, ou de grouper des longueurs d'onde dans des bandes ou des fibres au niveau WDM. Le groupage permet à un opérateur de simplifier les équipements et la gestion du réseau, et par suite les coûts.

L'objectif que nous fixons dans le problème de groupage particulier que nous étudions est lié à la minimisation du coût d'exploitation du réseau. En considérant un chemin orienté comme réseau physique, nous poursuivons le travail de [BDPS03] qui considére l'anneau. Nous verrons que cette hypothèse place notre problème de groupage dans la classe des problèmes de conception de réseaux à un seul niveau. Nous proposons plusieurs approches, heuristiques, programmation linéaire mixte et en nombres entiers pour résoudre notre problème. Enfin, nous abordons la génération d'instances du problème de groupage de grande taille dont une solution optimale est connue et qui pourraient permettre d'évaluer la qualité des solutions de nos heuristiques même lorsque les solveurs (Cplex, lpsolve) ne sont pas en mesure d'aboutir rapidement à des solutions.

3.1 Conception d'un niveau virtuel fiable

Les problèmes de conception de réseaux virtuels consistent d'une manière générale à établir un ensemble de liens virtuels en donnant leurs capacités et leurs routages sur un réseau physique entièrement connu muni de coûts et de capacités sur ses liens. Le réseau virtuel construit doit permettre d'écouler un ensemble de requêtes donné dont le routage sur le niveau virtuel fait également partie de la solution [HV06]. Les requêtes peuvent par exemple appartenir au niveau IP ou bien représenter les liens d'un autre niveau virtuel de la hiérarchie GMPLS. Des contraintes variées peuvent s'ajouter à ce cadre basique, par exemple des contraintes de délai [BKP03a, BKP03b] ou de longueur (nombre de *sauts*) [GPS06] sur les routes des requêtes, ou encore des contraintes de connexité liées à la tolérance aux pannes, etc. Deux types d'objectifs classiques consistent à maximiser la quantité de requêtes satisfaites [KP06] ou à minimiser le coût lié à la capacité utilisée au niveau physique [BKO06].

Les problèmes de conception de réseaux virtuels ont été très étudiés dans le cadre de la technologie WDM. Dans ce contexte, il s'agit de trouver pour chaque requête un chemin dans le réseau WDM et de lui affecter une longueur d'onde. Suivant les hypothèses, un tel chemin doit utiliser une unique longueur d'onde d'un bout à l'autre, ou bien peut en changer au niveau des nœuds capables d'effectuer des conversions de longueur d'onde. L'objectif le plus répandu est la minimisation du nombre de longueurs d'onde utilisées. Ce problème connu sous le nom de *Routing and Wavelength Assignment* (RWA) a fait l'objet de nombreuses publications [DR00, SS02, NR06].

La thèse [Big06] aborde le problème de la conception d'un niveau virtuel tolérant aux pannes dans un réseau à deux niveaux de type IP/WDM utilisant une architecture MPLS. La conception est optimisée dans un premier temps séquentiellement, c'est-à-dire un niveau après l'autre, puis en intégrant tous les niveaux en même temps. Les outils utilisés sont basés principalement sur la programmation mixte et les techniques de branchement.

Dans [MNT01, LT02, TR04b] la topologie virtuelle fait partie des données et le problème est de la router sur le niveau physique de sorte qu'elle reste connexe quelle que soit la panne survenant au niveau physique. Ce problème est modélisé par des programmes linéaires mixtes dont les objectifs sont de minimiser la capacité utilisée au niveau physique. De plus, les problèmes consistant à réaliser la protection aux deux niveaux du réseau simultanément et à réaliser une protection dépendante de la panne sont aussi étudiés dans [LT02].

Enfin, citons le livre [PM04] qui traite en grande partie des problèmes de conception de réseaux à un seul niveau et des réseaux virtuels, ainsi que le livre [Som06] consacré à la tolérance aux pannes et au groupage dans les réseaux WDM. Notons également qu'en plus d'exposer les principes des réseaux IP/WDM, le livre [Liu02] présente des méthodes de résolution pour des problèmes de conception de réseaux.

Dans cette section, nous proposons une modélisation en programme mixte du problème de conception d'un niveau virtuel entièrement protégé au niveau physique. Quelle que soit la panne qui se produise au niveau physique, nous imposons l'existence d'un chemin de secours pour chacun des liens virtuels affectés. Par conséquent, la topologie virtuelle reste inchangée lorsqu'une panne survient au niveau physique.

Nous utilisons le mode de protection partagée par chemin dépendant de la panne. Plus précisément, un lien virtuel est protégé par plusieurs chemins de secours pouvant être routés sur des liens physiques communs. Le chemin de secours qui est activé en cas de panne dépend du lien physique qui est affecté. Lorsque plusieurs chemins protégeant le même lien virtuel utilisent un même lien physique, au lieu que chacun réserve la capacité nécessaire à son routage, cette capacité n'est réservée

qu'une seule fois. Comme un seul de ces chemins peut être actif à un instant donné, il n'y a pas de conflit s'opposant à une utilisation partagée de la capacité réservée. Par contre, dans notre modèle, deux chemins de secours pour deux liens virtuels distincts ne peuvent partager ainsi leurs ressources physiques.

L'objectif que nous fixons est de minimiser le coût en terme de capacité requise au niveau physique, et aussi en terme de nombre de liens virtuels mis en place, qui est lié au coût opérationnel du réseau.

3.1.1 Données et notations

Les données du problème que nous souhaitons formuler sont les suivantes :
- Un graphe orienté $G_\varphi = (V, E_\varphi)$ dont chaque arc $a \in E_\varphi$ est muni d'une capacité $c_a \geq 0$ et d'un coût d'utilisation unitaire de la capacité γ_a. Ce graphe représente le réseau physique,
- Un graphe orienté $G_R = (V, E_R)$ dont chaque arc (s, t) est muni d'une taille de requête d^{st} représentant les requêtes à écouler sur le réseau physique par l'intermédiaire du réseau virtuel,
- Un ensemble discret de capacités disponibles pour les liens du niveau virtuel $\mathcal{C}$.

Le graphe $G_V = (V, E_V)$ est le graphe orienté complet sur l'ensemble des sommets V de G_φ et de G_R. Les capacités sur ses arcs doivent être déterminées parmi un ensemble discret de capacités possibles $\mathcal{C}$.

$\Gamma_V^+(z)$ représente l'ensemble des arcs sortants du sommet z dans le graphe virtuel G_V et $\Gamma_V^-(z)$ l'ensemble des arcs entrants du sommet z dans ce graphe. Pour un sous ensemble de sommets $S \in V$ l'ensemble des arcs de G_φ sortant de S est noté $\Gamma_\varphi^+(S)$. Le sommet origine d'un arc virtuel de E_V est noté $\mathcal{O}_e$ tandis que son sommet destination est noté $\mathcal{D}_e$.

3.1.2 Variables

La formulation proposée est une formulation mixte, c'est à dire qu'elle contient des variables binaires ou entières et des variables réelles.

x_e^{st} : fraction de la requête entre s et $t \in V$ passant sur l'arc $e \in E_V$, $0 \leq x_e^{st} \leq 1$

$y_e^c = 1$ si le lien virtuel $e \in E_V$ a la capacité $c \in \mathcal{C}$, 0 sinon,

$z_{ea}^c = 1$ si le lien virtuel $e \in E_V$ a la capacité $c \in \mathcal{C}$ et passe sur l'arc $a \in E_\varphi$, 0 sinon,

$u_{ea}^{rc} = 1$ si le chemin de protection du lien virtuel $e \in E_V$ en cas de panne sur l'arc $r \in E_\varphi$ utilise l'arc $a \in E_\varphi$ avec la capacité $c \in \mathcal{C}$, 0 sinon,

$v_{ea}^c = 1$ si au moins un des chemins, principal et de protection du lien virtuel $e \in E_V$ pour une panne quelconque, passe sur l'arc $a \in E_\varphi$ avec la capacité $c \in \mathcal{C}$, 0 sinon,

3.1.3 Objectif

L'objectif est de minimiser la somme du coût d'utilisation de la capacité sur le réseau physique et du nombre de liens virtuels établis :

$$\sum_{c \in \mathcal{C}} \sum_{e \in E_V} \sum_{a \in E_\varphi} c\gamma_a z_{ea}^c \quad + \quad \sum_{c \in \mathcal{C}} \sum_{e \in E_V} y_e^c \tag{3.1}$$

Un inconvénient de cet objectif est qu'en minimisant le nombre total de liens virtuels, on risque d'induire, pour certaines requêtes, des chemins très longs en nombre de liens virtuels. En effet, lorsqu'un long chemin, qui permet le routage d'une requête donnée, existe dans le réseau virtuel,

au lieu de créer un nouveau lien virtuel, la requête est routée sur le long chemin. Or les opérateurs cherchent souvent à minimiser ce critère. Des études ont déjà étés réalisées sur ce sujet en particulier dans [Cha98], [Mar00] et [Cho02]. En général, pour remédier à cet inconvénient, des contraintes supplémentaires sont imposées sur la longueur des chemins. Un autre objectif qu'il pourrait être intéressant de prendre en compte est la minimisation de la longueur moyenne des chemins en nombre de liens virtuels traversés.

3.1.4 Contraintes

Le routage des requêtes sur le réseau virtuel d'une part et des liens virtuels sur le réseau physique d'autre part, est réalisé de manière très classique par des contraintes de multiflot. Des contraintes supplémentaires permettent d'imposer le partage des capacités ainsi que l'existence de chemins de protection dépendants de la panne physique.

$$\sum_{st \in E_R} d^{st} x_e^{st} \leq \sum_{c \in \mathcal{C}} c y_e^c, \qquad \forall e \in E_V \tag{3.2}$$

$$\sum_{e \in \Gamma_V^+(z)} x_e^{st} - \sum_{e \in \Gamma_V^-(z)} x_e^{st} = \begin{cases} 0 & si \quad z \neq s, z \neq t \\ 1 & si \quad z = s \\ -1 & si \quad z = t \end{cases} \qquad \forall (s,t) \in E_R,\ \forall z \in V \tag{3.3}$$

$$\sum_{c \in \mathcal{C}} y_e^c \leq 1, \qquad \forall e \in E_V \tag{3.4}$$

$$z_{ea}^c \leq y_e^c, \qquad \forall e \in E_V,\ \forall a \in E_\varphi,\ \forall c \in \mathcal{C} \tag{3.5}$$

$$\sum_{c \in \mathcal{C}} \sum_{a \in \Gamma^+(S)} z_{ea}^c \geq \sum_{c \in \mathcal{C}} y_e^c, \qquad \forall e \in E_V,\ \forall S \subseteq V,\ \mathcal{O}_e \in S,\ \mathcal{D}_e \notin S \tag{3.6}$$

$$u_{ea}^{rc} \leq z_{er}^c, \qquad \forall e \in E_V,\ \forall a \in E_\varphi,\ \forall r \neq a \in E_\varphi,\ \forall c \in \mathcal{C} \tag{3.7}$$

$$\sum_{c \in \mathcal{C}} \sum_{a \in \Gamma^+(S)} u_{ea}^{rc} \geq \sum_{c \in \mathcal{C}} z_{er}^c, \qquad \forall e \in E_V,\ \forall r \in E_\varphi,\ \forall S \subseteq V,\ \mathcal{O}_e \in S,\ \mathcal{D}_e \notin S \tag{3.8}$$

$$u_{ea}^{rc} + z_{ea}^c \leq 2 v_{ea}^c, \qquad \forall e \in E_V,\ \forall a \in E_\varphi,\ \forall r \neq a \in E_\varphi,\ \forall c \in \mathcal{C} \tag{3.9}$$

$$\sum_{c \in \mathcal{C}} c v_{ea}^c \leq c_a, \qquad \forall a \in E_\varphi,\ \forall r \neq a \in E_\varphi \tag{3.10}$$

La capacité du lien virtuel e est en partie déterminée par la contrainte (3.2). En effet, la capacité de e doit être suérieure à la somme de tous les trafics y circulant. La continuité du flot de la requête (s,t) en chaque sommet z est assurée par la contrainte (3.3). Ces deux contraintes sont classiques dans la modélisation sommet-arc d'un multiflot (fractionnaire) en programme linéaire.

Le rôle de la contrainte (3.4) est d'empêcher un LSP $e \in E_V$ de posséder plus d'une capacité c parmi les capacités disponible de $\mathcal{C}$. Il peut éventuellement n'en avoir aucune si ce lien virtuel n'est pas utilisé pour le routage des requêtes.

La contrainte (3.5) fait en sorte qu'un lien virtuel e ne puisse pas réserver une capacité c sur un lien physique a si la capacité de ce lien virtuel n'est pas fixée à c.

Inversement, la contrainte (3.6) force un lien virtuel sur lequel des requêtes sont routées à réserver de la capacité sur un chemin allant de son origine à sa destination. Cette contrainte peut être remplacée par la contrainte (3.11) plus courante de la formulation sommet-arc.

La contrainte (3.7) concerne le chemin de protection du lien virtuel e en cas de panne de l'arc physique r. Si le chemin principal sur lequel ce lien virtuel est routé n'utilise pas l'arc r avec une capacité c, il n'y a pas besoin de chemin de protection de capacité c pour ce cas de panne.

L'existence d'un chemin de protection entre l'origine et la destination du lien virtuel e pour chaque cas de panne d'un lien r utilisé par le chemin principal de e est forcée par la contrainte (3.8).

Lorsqu'un chemin principal ou un chemin de protection pour un même lien virtuel utilisent tous les deux un arc $a \in E_\varphi$, ces chemins qui ne seront pas actifs en même temps n'ont pas besoin de réserver chacun une capacité c sur a (contrainte (3.9)).

Enfin la contrainte (3.10) impose que la capacité réservée sur le niveau physique pour le routage des liens virtuels ne soit pas supérieure à la capacité disponible.

$$\sum_{c\in\mathcal{C}} \sum_{a\in\Gamma^+_\varphi(z)} z^c_{ea} - \sum_{c\in\mathcal{C}} \sum_{a\in\Gamma^-_\varphi(z)} z^c_{ea} = \begin{cases} 0 & si \quad z \neq \mathcal{O}_e, z \neq \mathcal{D}_e \\ 1 & si \quad z = \mathcal{O}_e \\ -1 & si \quad z = \mathcal{D}_e \end{cases} \qquad \forall e \in E_V \ \forall z \in V \qquad (3.11)$$

$$\sum_{c\in\mathcal{C}} \sum_{a\in\Gamma^+_\varphi(z)} u^{rc}_{ea} - \sum_{c\in\mathcal{C}} \sum_{a\in\Gamma^-_\varphi(z)} u^{rc}_{ea} = \begin{cases} 0 & si \quad z \neq \mathcal{O}_e, z \neq \mathcal{D}_e \\ 1 & si \quad z = \mathcal{O}_e \\ -1 & si \quad z = \mathcal{D}_e \end{cases} \qquad \forall e \in E_V, \ \forall r \in E_\varphi \ \forall z \in V \quad (3.12)$$

3.1.5 Remarques

Dans ce modèle seuls le chemin principal et les chemins de protection d'un même lien virtuel peuvent partager leur capacité sur le niveau physique. En ajoutant des variables binaires et des contraintes supplémentaires il serait possible de modéliser également le partage de capacité entre les chemins de protection de liens virtuels différents. De plus des contraintes de délai (non linéaires) comme celles utilisées dans [BKP03a, BKP03b] peuvent aussi être ajoutées, ou des contraintes limitant la longueur de la route d'une requête sur le réseau virtuel.

Cependant, le problème du groupage contenu dans ce problème de conception de réseau virtuel étant déjà un problème difficile, nous préférons l'étudier séparément avant de lui ajouter d'autres difficultés.

Le problème de groupage auquel nous nous intéressons nécessite d'une part de fixer différents éléments par rapport au cas général, comme le réseau physique, les requêtes, les tailles des liens virtuels qu'on appellera *tubes*, et d'autre part de supprimer les contraintes de protection qui n'ont plus de sens sur le chemin. En outre, nos hypothèses permettent de classer le groupage dans les problèmes de dimensionnement de réseau (Network Design) sur un seul niveau.

3.2 Groupage

Les problèmes de groupage font partie des problèmes de conception de réseaux virtuels. Du point de vue du groupage, router deux requêtes sur un même lien virtuel entre deux nœuds revient à grouper ces deux requêtes ensemble entre les deux nœuds, ou de manière équivalente à agréger ces deux requêtes en un flux de débit supérieur entre les deux nœuds. Au niveau physique, les deux requêtes sont alors routées sur le même chemin entre ces deux nœuds. Dans les problèmes de groupage, les liens virtuels sont appelés *tubes* [HPS02]. Cette terminologie souligne la notion de conteneur décrite au chapitre 2. Un LSP est bien en un sens un tube, avec deux extrémités bien définies correspondant aux LER chargés d'empiler puis de dépiler le label définissant ce LSP. Les données qui arrivent au LER de départ et reçoivent ce label sont ensuite acheminées un peu comme dans un tube, elles ne peuvent pas être extraites avant d'avoir atteint l'autre extrémité du LSP.

La hiérarchie GMPLS des réseaux IP/WDM prévoit deux types de LSP, ceux basés sur un label intégré aux paquets IP, et ceux basés sur une caractéristique physique comme la longueur d'onde utilisée ou le slot temporel. Le groupage s'opère également sur deux niveaux technologiques, le niveau des paquets correspondant aux premiers LSP avec label et le niveau WDM avec la hiérarchie de conteneurs décrite au chapitre 2.

L'intérêt principal du groupage, outre le fait d'utiliser au mieux les ressources du réseau (section 2.1.4 chapitre 2), est de simplifier considérablement le routage au niveau IP et de réduire la charge de travail pour les routeurs, et par conséquent leur coût (section 2.1.5 chapitre 2). La capacité de calcul nécessaire à l'acheminement de données sur un LSP est en effet inférieure à celle requise pour router les mêmes données sous forme de paquets IP indépendants, les équipements peuvent donc être moins complexes.

Au niveau optique le groupage permet également de réduire le coût des nœuds. Par exemple, router huit longueurs d'onde ou router une bande contenant huit longueurs d'onde dans un réseau n'est pas équivalent en terme de coût. Pour router les longueurs d'onde il est nécessaire que tous les ADM traversés sur la route comportent une composante W-OXC à huit entrées et sorties pour pouvoir traiter les huit longueurs d'onde entrantes. Par contre, pour router la bande de huit longueurs d'onde, les ADM traversés ne doivent comporter qu'une composante B-OXC à une entrée et une sortie. La complexité des ADM est donc beaucoup plus grande pour router les huit longueurs d'onde que la bande, et le coût des ADM dépend de leurs complexités tant du point de vue technologique que du point de vue du nombre d'entrées et sorties.

D'autre part le groupage facilite l'utilisation de la protection par segment : au lieu de protéger toutes les requêtes de bout en bout au niveau physique, ce sont les liens virtuels qui sont protégés. Les chemins de secours sont moins longs et par conséquent leur délai de mise en place est également moins long.

Il existe plusieurs façons d'aborder le problème du groupage. Parfois des contraintes de tolérance aux pannes sont prises en compte [YR05]. Dans certains travaux, les requêtes sont déjà routées sur le réseau physique et il faut trouver un ensemble de tubes permettant de les grouper au mieux et de respecter les routes imposées [HPS02]. D'autres travaux au contraire considèrent que le routage doit être effectué en même temps que le groupage, il faut alors trouver un routage permettant un groupage efficace.Un tour d'horizon du groupage est donné dans [ML01] et [ZM03].

Plusieurs critères d'optimisation ont été étudiés, comme la minimisation du nombre d'ADM, qui se justifie par le coût important de ces équipements indispensables aux extrémités d'un tube dans un réseau WDM [DR02, BCM03b]. Cet objectif est l'objet de nombreux travaux considérant des topologies physiques particulières [HDR06] et parfois des graphes de requêtes complets (communications *All to All*) [BCCP06]. Ces cas particuliers constituent d'intéressants problèmes de décomposition de graphes.

D'autres ont minimisé les capacités totales ou maximales utilisées sur le réseau support. L'objectif le plus étudié actuellement est la minimisation du nombre d'entrées et sorties utilisées par chaque niveau d'encapsulation. Dans [CAXQ03] par exemple, l'objectif est de minimiser le nombre total d'entrées et sorties sur tous les ADM du réseau.

Ici nous cherchons à minimiser le nombre de tubes nécessaires à l'acheminement de toutes les requêtes, dont l'une des applications pratiques est la minimisation du coût de gestion des LSP dans un réseau MPLS.

Une telle fonction de coût avait déjà été étudiée dans [BDPS03] pour un graphe support particulier, l'anneau. La solution proposée consistait à couvrir le graphe des requêtes par des briques élémentaires connues pour router un maximum de requêtes en un nombre minimum de tubes. L'objet des sections suivantes est l'étude du même problème de groupage, mais avec un chemin comme

graphe support.

3.3 Problème du groupage sur un chemin

Après avoir défini le problème de groupage particulier que nous étudions, nous expliquerons en quoi il peut être considéré comme un problème de dimensionnement de réseau à un seul niveau. Nous présenterons ensuite les résultats de la littérature s'appliquant à notre problème.

3.3.1 Définition du problème

Notre problème de groupage requiert la donnée de trois éléments, un graphe support, un ensemble de requêtes et un ensemble de tubes disponibles.

Graphe support $G = (V, A)$ Le graphe support considéré est un chemin orienté composé de n sommets numérotés de 1 à n, $V = (1, 2, \ldots, n)$ et des arcs $A = \{(i, i+1), 1 \leq i < n\}$.

La capacité des arcs est infinie : autant de tubes que nécessaire peuvent emprunter un arc donné.

Requêtes R Le graphe des requêtes $G_R = (V, R)$ est défini sur l'ensemble de sommets V du graphe support. L'ensemble des requêtes R est un ensemble de paires ordonnées de sommets de V de la forme (i, j), avec $i < j$ et $i, j \in V = \{1, \ldots, n\}$, qui constitue l'ensemble des arcs de G_R.

Les requêtes sont simples et unitaires, c'est-à-dire qu'il existe au plus une demande dans R entre deux sommets donnés et qu'elle n'utilise qu'une unité de capacité sur les arcs qu'elle emprunte dans le graphe support.

Notons que sur le chemin le routage des requêtes est unique et immédiat. La route d'une requête (s, t) est la portion du chemin support entre les sommets s et t. La *taille* d'une requête est la longueur du chemin support entre s et t.

Tubes disponibles T Tous les tubes de la forme $i - j$ avec $i < j$ et $i, j \in V = \{1, \ldots, n\}$ sont autorisés en autant d'exemplaires que nécessaire. Leur capacité C, ou *facteur de groupage*, est uniforme et fait partie des données. Au plus C requêtes peuvent donc emprunter le même tube.

Le coût d'utilisation d'un tube est unitaire et fixe, il ne dépend pas du nombre de requêtes qui l'utilisent. En d'autres termes le coût d'utilisation d'un ensemble de tubes est le cardinal de cet ensemble.

On peut constater que le graphe des tubes disponibles est un graphe multiple orienté acircuitique ayant une unique source, le sommet 1 et un unique puits, le sommet n.

Objectif Nous souhaitons installer d'un ensemble de tubes de cardinal minimum sur le graphe support constituant un groupage réalisable des demandes.

La figure 3.1 montre deux groupages différents pour un graphe des requêtes donné. Les tubes installés sont représentés par des rectangles dont les extrémités concordent avec les sommets origine et destination de chaque tube. Par exemple, le tube $1 - 3$ d'origine le sommet 1 et de destination le sommet 3 est représenté dans le premier groupage par un rectangle s'étendant du sommet 1 au sommet 3. Toutes les requêtes dont la représentation traverse l'un des rectangle est une requête empruntant le tube correspondant pour ce groupage. Selon notre critère d'optimisation, le second groupage qui n'utilise que cinq tubes est meilleur que le premier qui en utilise sept.

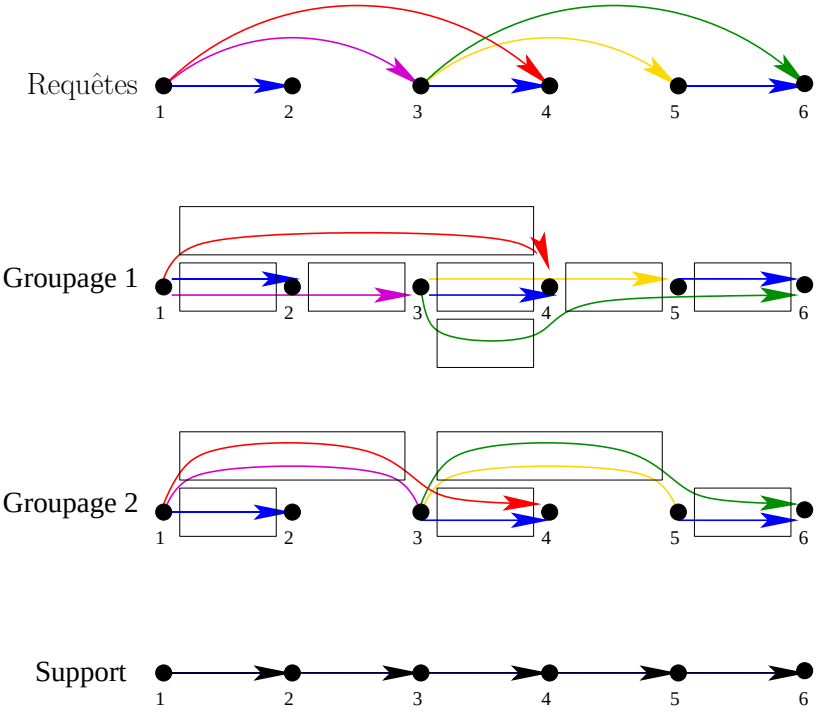


FIG. 3.1 – Exemple de groupage de requêtes sur un chemin orienté.

3.3.2 Le groupage sur le chemin en tant que dimensionnement de réseaux sur un niveau

La classe des problèmes de dimensionnement de réseaux sur un niveau englobe en particulier tous les problèmes dont l'objectif est l'installation d'un ensemble de liens de capacités données, afin de pouvoir écouler un ensemble de requêtes connu. En d'autres termes, l'objectif est de trouver un ensemble d'arcs de coût minimum entre des sommets connus permettant de router toutes les requêtes en respectant les capacités de ces arcs.

Une solution du groupage sur un chemin est d'une part un ensemble de tubes, d'autre part pour chaque requête la liste des tubes dans lesquels elle a été groupée. On peut donc interpréter cette solution comme un graphe virtuel se superposant au graphe support, dont les sommets sont ceux du graphe support et les arcs sont les tubes. La liste de tubes empruntés par une requête donne alors un routage de la demande dans ce graphe virtuel.

Ainsi la recherche d'un ensemble de tubes de cardinal minimum permettant le groupage de toutes les requêtes apparaît comme un problème de dimensionnement de réseaux. Le graphe de coût minimum à déterminer est le graphe virtuel des tubes.

Cependant, dans le cas général, des contraintes supplémentaires compliquent l'interprétation du groupage comme un dimensionnement de réseaux à un seul niveau. Par définition un tube est un chemin entre deux sommets du graphe support, donc s'il existe p chemins différant ne serait-ce que d'un arc, d'un sommet a vers un sommet b, il existe aussi p tubes possibles d'origine a et de destination b. Or, la route suivie par une requête dans le graphe support fait partie des données du problème de groupage. Ainsi, pour grouper une requête passant en a puis en b dans un tube entre ces sommets, il est indispensable que le chemin associé à ce tube soit aussi celui emprunté par la requête entre a et b. C'est à dire que parmi les p tubes possibles entre a et b, un seul est accessible

par cette demande, et bien sûr ce n'est pas nécessairement le même pour toutes les requêtes passant en a puis en b.

Sur le chemin il existe une unique route, et donc un unique tube, entre deux sommets. On s'affranchit ainsi des contraintes de respect des routes.

La littérature sur le dimensionnement de réseau à un seul niveau est abondante du fait du nombre de problèmes différents appartenant à cette classe. Comme dans le cas à deux niveaux, diverses contraintes de connexité et de tolérance aux pannes [BMN05] ou de qualité de service notamment sur les délais et la longueur des routes, ont étés étudiées. Les résolutions proposées sont souvent basées sur une approche polyèdrale [Bar96, Dah91, KM01]. D'autres méthodes, comme les approximations ou les heuristiques [GMS95], [BMN05], ont aussi été proposées.

3.3.3 Résultats antérieurs

Le groupage sur le chemin est très proche du groupage sur l'anneau puisque seul le graphe support diffère. Les résultats donnés dans [BDPS03] dont les démonstrations n'utilisent pas les propriétés du graphe support sont donc encore valables, en particulier la complexité et les bornes sur le nombre de tubes minimal nécessaire au groupage d'un ensemble de requêtes.

Complexité

Théorème 3.1 ([BDPS03]) *Décomposer un graphe en un nombre maximum de triangles se réduit au problème du groupage sur l'anneau ou sur le chemin pour un facteur de groupage $C = 2$.*

Corollaire 3.1 ([BDPS03]) *Le problème du groupage sur le chemin ou sur l'anneau est NP-Difficile pour un facteur de groupage C quelconque.*

Bornes La démonstration des propositions suivantes repose sur le fait que les requêtes sont simples et unitaires. Nous pouvons aussi donner une borne supérieure triviale mais atteinte.

Proposition 3.1 ([BDPS03]) *Soient R l'ensemble de requêtes, T l'ensemble de tubes d'une solution réalisable et C le facteur de groupage, alors $\frac{2|R|}{C+1} \leq |T| \leq |R|$.*

Preuve: Considérons un ensemble de tubes T solution au problème de groupage. Soit $|R_i|$ le nombre de requêtes dont le routage utilise exactement i tubes, alors $|R| = \sum_i |R_i|$ représente bien le nombre total de requêtes. Entre deux sommets s et t il existe au plus une requête (s,t), donc dans un tube d'extrémités s et t circule au plus une requête n'empruntant qu'un seul tube. Par conséquent $|T| \geq |R_1|$. De plus la somme sur toutes les requêtes des tubes que chacune utilise, $\sum_i i|R_i|$, vérifie $C|T| \geq \sum_i i|R_i|$ car il y a au plus C requêtes qui utilisent un tube donné. Nous pouvons réécrire ces égalités et inégalités de la façon suivante :

$$|R| = R_1 + R_2 + \sum_{i\geq 3} R_i \tag{3.13}$$

$$C|T| \geq R_1 + 2R_2 + \sum_{i\geq 3} iR_i \tag{3.14}$$

$$|T| \geq R_1 \tag{3.15}$$

L'équation (3.13) implique $|R_2| = |R| - |R_1| - \sum_{i\geq 3} |R_i|$ qui, une fois remplacé dans l'inéquation (3.14), donne : $C|T| \geq 2|R| - |R_1| + \sum_{i\geq 3}(i-2)|R_i|$. Cette inégalité peut se réécrire simplement

$C|T| + |R_1| \geq 2|R| + \sum_{i\geq 3}(i-2)|R_i|$. Comme $|T| \geq |R_1|$ (inéquation (3.15)), on en déduit que $(C+1)|T| \geq 2|R| + \sum_{i\geq 3}(i-2)|R_i| \geq 2|R|$, et par suite $|T| \geq \frac{2|R|}{C+1}$.

∎

Les propriétés que doit vérifier toute solution pour atteindre la borne inférieure peuvent se déduire de la preuve précédente.

Proposition 3.2 *Une solution du groupage de l'ensemble des requêtes R avec un facteur de groupage C utilise un nombre de tubes égal à la borne inférieure $\frac{2|R|}{C+1}$ si et seulement si les trois conditions suivantes sont vérifiées :*

1. *tous les tubes contiennent la requête n'utilisant que ce tube.*
2. *tous les tubes contiennent exactement C requêtes,*
3. *toute requête utilise moins de deux tubes.*

Preuve: La borne est atteinte si et seulement si les inégalités apparaissant dans la preuve de la proposition 3.1 deviennent des égalités. Ceci implique que $R_i = 0$ pour $i \geq 3$, $|T| = R_1$ et $C|T| = R_1 + 2R_2$, c'est à dire les trois conditions. ∎

3.4 Méthodes de résolution pour le groupage sur le chemin

Dans cette section nous présentons plusieurs méthodes de résolution pour le problème du groupage sur le chemin orienté. Parmi ces méthodes certaines sont basées sur la programmation linéaire, les autres sont des heuristiques. Nous comparons ensuite l'efficacité de ces méthodes.

3.4.1 Programmes linéaires

Comme nous l'avons dit précédemment, le problème de groupage peut être vu comme un cas particulier de network design. Nous allons maintenant donner la formulation de ce problème à l'aide d'un programme linéaire en nombres entiers.
Soit $T = (V, E_T)$ le graphe des tubes que l'on peut installer, c'est à dire un graphe complet défini sur les n sommets de G et orienté (comme G et D). Chercher le nombre minimal de tubes nécessaires pour grouper toutes les requêtes, c'est chercher un nombre minimal d'arcs de T qui permettent de faire passer le flot associé aux requêtes de D. En terme de formulations, deux approches sont possibles. On peut soit utiliser la formulation arc-chemin (3.16), soit la formulation sommet-arc (3.17). La première formulation comporte un nombre exponentiel de variables, mais elle est souvent employée en pratique car elle permet une résolution beaucoup plus rapide grâce aux technique de génération de colonnes [CCPS98]. Nous l'utiliserons pour obtenir (assez rapidement) une borne supérieure proche de la valeur optimale. Voici la formulation arc-chemin :

$$\begin{cases} Min & \sum_{a\in A_T} x_a & & \\ s.t. & \sum_{p\in\Pi_{st}} \Phi_p^{st} \geq 1 & & \forall (s,t) \in D \subseteq V^2, \\ & \sum_{(st)\in D} \sum_{p\in\Pi_{st}, p\ni a} \Phi_p^{st} \leq C * x_a & & \forall a \in A_T, \\ & \Phi_p^{st} \in \{0,1\} & & \forall (s,t) \in D, \forall p \in \Pi_{st} \\ & x_a \in \mathbb{N} & & \forall a \in A_T. \end{cases} \tag{3.16}$$

Dans cette modélisation, Φ_p^{st} représente la quantité de flot associée à la requête (s,t) qui transite sur le chemin p. Π_{st} est l'ensemble des chemins allant du sommet s au sommet t. x_a représente le nombre de fois où le tube a est choisi dans la solution. L'objectif est de minimiser le nombre de tubes. La première contrainte correspond au respect des requêtes et la seconde contrainte, au respect des capacités des tubes.
Voici maintenant la formulation sommet-arc :

$$\begin{cases} Min & \sum_{a \in A_T} x_a & & \\ s.t. & \sum_{a \in \delta^+(u)} f_a^{st} - \sum_{a \in \delta^-(u)} f_a^{st} = \begin{cases} 0 & si \quad u \neq v, u \neq t \\ 1 & si \quad u = s \\ -1 & si \quad u = t \end{cases} & \forall u \in V, \ \forall (s,t) \in D \subseteq V^2 \\ & \sum_{(st) \in D} f_a^{st} \leq C * x_a & \forall a \in A_T \\ & f_a^{st} \in \{0,1\} & \forall (s,t) \in D, \forall a \in A_T \\ & x_a \in \mathbb{N} & \forall a \in A_T. \end{cases} \tag{3.17}$$

La variable de décision x a la même signification que précédemment et la variable f_a^{st} représente le flot, associé à la demande (st), qui circule sur l'arc a. $\delta^+(s)$ (resp. $\delta^-(s)$) est l'ensemble des arcs sortants (resp. entrants) du sommet s.

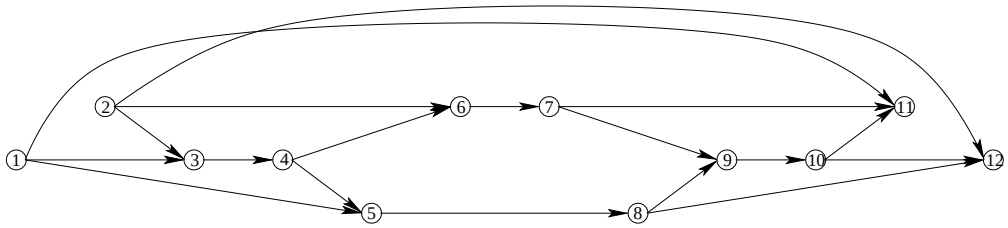
On peut noter, dans ces deux formulations, que le flot est entier. Mais cette contrainte est-elle vraiment nécessaire ? Nous avons relâché cette contrainte, pour remplacer le programme en nombres entiers par un programme mixte (MILP), afin de simplifier la résolution du problème. Nous nous sommes basés pour ce faire sur une conjecture que nous avons ainsi formulée :

Conjecture 3.1 *Dans le problème de groupage sur un chemin orienté, le nombre de tubes minimal est identique, que le flot soit fractionnaire ou entier.*

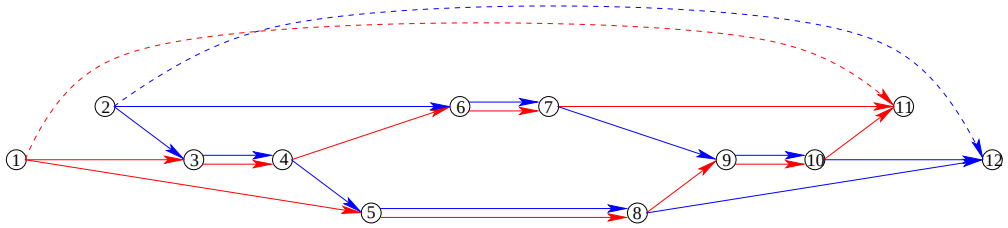
Cette conjecture signifie que le nombre de tubes minimal donné par la résolution des programmes linéaires en nombres entiers (3.17) et (3.16) est le même que celui fourni par ces mêmes programmes dans lesquels on a relâché la contrainte d'intégrité sur les flots. L'intérêt de cette relaxation est un fort gain de temps de calcul, comme nous le constaterons expérimentalement dans la section résultat (section 3.4.3).

Pour démontrer cette conjecture, nous avons essayé de montrer qu'à partir d'une solution fractionnaire, une permutation des flots sur des chemins ayant une même origine et destination permet de rendre ces flots entiers. Mais, comme le montre l'exemple suivant, il n'est pas toujours possible de permuter des flots réels pour les rendre entiers avec un ensemble de tubes minimum pour la formulation relâchée du problème.

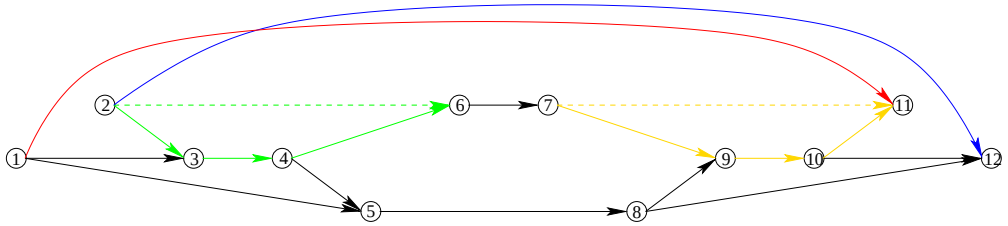
Considérons le graphe des requêtes représenté dans la figure 3.2(a) et un coefficient de groupage égal à 2. Le graphe des tubes constitué d'un tube par requête, excepté pour les requêtes $(1,11)$ et $(2,12)$, est une solution optimale du problème de groupage de ces requêtes pour la formulation relâchée. La figure 3.2(b) montre comment les requêtes $(1,11)$ et $(2,12)$ peuvent être routées dans les tubes de cette solution. Pour chacune de ces deux requêtes, le flot est séparé en deux quantités égales sur deux chemins distincts. En considérant ce graphe des tubes, le routage des requêtes ne peut pas être entier. En effet, si la requête $(1,11)$ est routée sur le chemin $(1,3,4,5,8,9,10,11)$ et la requête $(2,12)$ sur le chemin $(2,6,7,9,10,12)$, la quantité de flot passant sur le tube entre



(a) Graphe des requêtes



(b) Graphe des flots fractionnaires



(c) Graphe des flots entiers

FIG. 3.2 – Ensemble de tubes différents entre une solution entière et une solution réelle

les sommets 9 et 10 dépasse la capacité de ce tube. Dans d'autres configurations, cela pourra être la contrainte associée au tube entre 3 et 4 qui sera violée. En effet, quels que soient les chemins considérés pour router de manière entière les requêtes $(1, 11)$ et $(2, 12)$ (dans le graphe des tubes), ces chemins ont au moins un tube commun, or tous les tubes contiennent déjà une requête, il reste donc de la place pour une seule requête dans chaque tube. Pourtant, si l'on remplace les tubes $2-6$ et $7-11$ par les tubes $1-11$ et $2-12$, on ne modifie pas le nombre de tubes total, mais on a bien une solution avec des flots entiers (cf. fig. 3.2(c)).

Cet exemple ne prouve pas que notre conjecture est fausse. Il indique simplement que si les nombres de tubes optimaux fournis par les programmes en variables entières et par les programmes en variables mixtes sont égaux, les ensembles de tubes ne sont pas nécessairement identiques.

Notons que pour tester la qualité de méthodes de résolution, connaître le nombre de tubes optimal est suffisant et le programme en variables mixtes peut être utilisé. Par contre, pour trouver des ensembles de tubes optimaux, il est nécessaire d'utiliser l'un des programmes en variables entières.

3.4.2 Heuristiques

Nous proposons deux heuristiques gloutonnes pour le problème du groupage sur le chemin. Le principe de la première heuristique est d'installer toujours les tubes les plus courts possibles, l'idée sous-jacente étant que plus un tube est court, plus il y a de requêtes qui pourront éventuellement

l'utiliser. L'idée de la seconde heuristique est d'installer des tubes pleins les plus longs possibles. Pour cela les requêtes longues sont décomposées en une succession de requêtes plus courtes existant dans le graphe des requêtes, jusqu'à obtenir C requêtes identiques pour former un tube.

Heuristique 1 Dans un premier temps, les requêtes sont classées dans l'ordre croissant de leurs tailles, puis de leurs positions. Avec le graphe de requêtes de la figure 3.3, les requêtes de taille un, $(1,2)$, $(3,4)$ et $(5,6)$, sont classées avant les requêtes de taille deux, $(1,3)$ et $(3,5)$. Parmi les requêtes de taille un, la requête $(1,2)$ est classée avant la requête $(3,4)$ car son origine 1 est inférieure à 3.

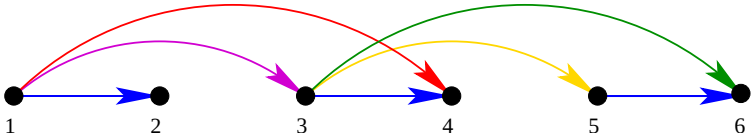


FIG. 3.3 – Un graphe de requêtes.

Les requêtes sont ensuite traitées une par une dans l'ordre. S'il existe une route pour une requête (i,j) donnée dans le graphe des tubes déjà choisis, cette route est fixée pour la requête (i,j), et les capacités restantes dans les tubes installés sont mises à jour. Lorsqu'il n'y a pas de route pour une requête, le tube le plus court possible permettant la création d'une route est installé et la requête peut alors être routée.

Le choix du tube à ajouter nécessite le calcul de deux chemins dans le graphe des tubes déjà installés G_T :

1. le plus long chemin p_1 dans G_T partant de i et dont l'autre extrémité est positionnée avant le sommet j : $p_1 = (i, ..., k)$.
2. le plus long chemin p_2 dans G_T partant d'un sommet k' compris entre i et j et dont l'autre extrémité est j : $p_2 = (k', ..., j)$.

Notons que dans un graphe orienté acircuitique comme le graphe des tubes choisis, trouver un plus long chemin est polynomial. Le tube à installer est déterminé suivant ces chemins de la manière suivante (figure 3.4) :

1. Si $k = i$ et $k' = j$, le tube $i - j$ est créé (figure 3.4(a)).
2. Si $k = i$ et $k' < j$, le tube $i - k'$ est créé (figure 3.4(b)).
3. Si $k > i$ et $k' = j$, le tube $k - j$ est créé (figure 3.4(c)).
4. Si $k > i$ et $k' < j$ et $k < k'$, le tube $k - k'$ est créé (figure 3.4(d)).
5. Si $k > i$ et $k' < j$ et $k > k'$, le plus petit (en termes de tailles) des deux tubes $i - k'$ et $k - j$ est créé (figure 3.4(e)).

Le tube créé, quel qu'il soit, a une capacité de C et permet de router la requête courante.

Exemple Avec l'heuristique 1, les requêtes de taille 1 du graphe de la figure 3.5 sont classées dans l'ordre $((1,2),(3,4),(5,6))$. Initialement aucun tube n'est installé, par conséquent un tube est créé pour chacune de ces requêtes de taille 1. Pour un facteur de groupage $C = 2$, il reste la place de router une autre requête dans chacun de ces tubes. Ensuite les requêtes de taille 2 sont traitées, la requête $(1,3)$ avant la requête $(3,5)$. Les tubes existants ne suffisent pas à router ces requêtes, il faut donc en installer d'autres. Pour la requête $(1,3)$ il y a deux possibilités, soit installer le tube $1-3$ soit installer le tube $2-3$. Comme $2-3$ est le plus court, c'est celui-ci qui est choisi par

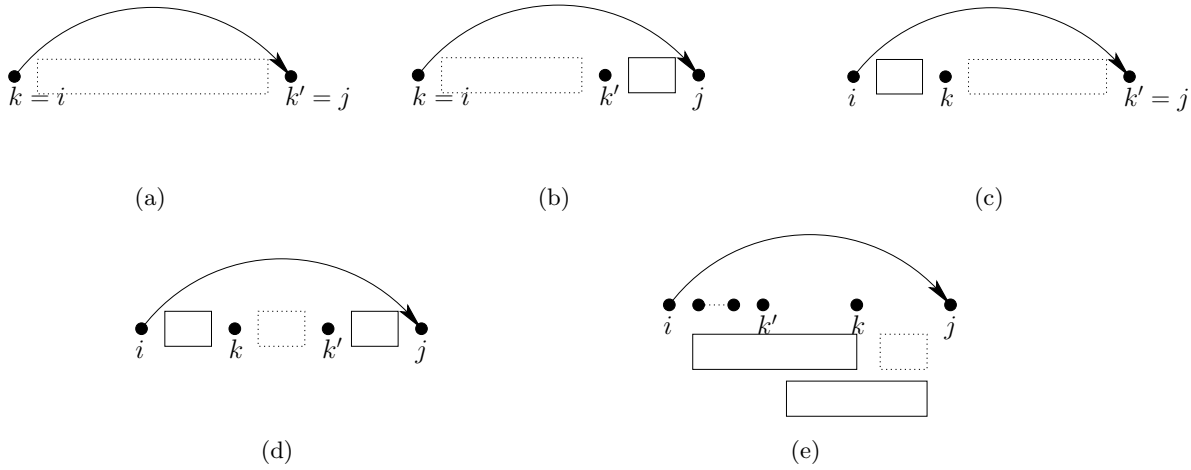


FIG. 3.4 – Illustration du choix du tube à ajouter pour créer un chemin

Heuristique 1

1: Classer les requêtes par ordre croissant (taille puis position) dans une liste L
2: Initialiser le graphe des tubes $G_T = (V, \emptyset)$
3: **tant que** $L \neq \emptyset$ **faire**
4: $(i, j) \leftarrow$ Premier(L)
5: $L \leftarrow L \backslash (i, j)$
6: **si** il existe un plus court chemin (nombre min. d'arcs) de i à j dans G_T **alors**
7: routage de la requête (i, j) sur ce chemin
8: mise à jour des capacités des tubes sur ce chemin
9: **sinon**
10: création dans G_T du plus petit tube nécessaire pour router (i, j)
11: routage de (i, j) sur le chemin créé
12: mise à jour des capacités des tubes sur ce chemin
13: **fin si**
14: **fin tant que**

l'heuristique 1. Ainsi la requête $(1, 3)$ est routée dans les tubes $1-2$ puis $2-3$. De la même façon, le tube $4-5$ est installé et la requête $(3, 5)$ utilise les tubes $3-4$ et $4-5$. Enfin l'heuristique 1 installe le tube $1-4$ puisque c'est l'unique tube dont l'ajout permet de router la requête $(1, 4)$. Pour router la requête $(3, 6)$, comme il reste une place dans les tubes $4-5$ et $5-6$, il suffit d'ajouter le tube $3-4$ ce qui donne la solution en sept tubes représentée par la figure 3.5.

En pratique cette heuristique donne de bons résultats, mais une requête peut emprunter beaucoup de petits tubes, comme la requête $(3, 6)$ dans l'exemple qui emprunte les trois tubes $3-4$, $4-5$ et $5-6$. Or dans un réseau de télécommunication réel, le respect de la qualité de service impose souvent un nombre de sauts maximum pour une route. Le principe de la deuxième heuristique permet de remédier à cet inconvénient.

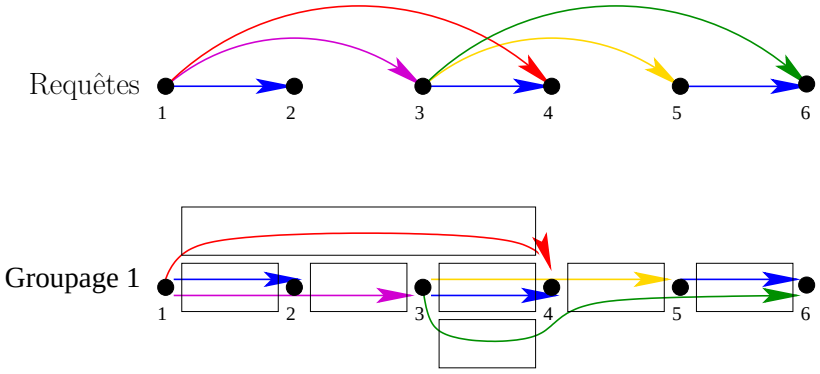


FIG. 3.5 – Exemple de groupage avec l'heuristique 1 pour $C = 2$

Heuristique 2 Dans la seconde heuristique, les requêtes sont classées par ordre décroissant de leurs tailles. L'idée de cette heuristique est de regrouper une longue requête (i, j) avec une succession de requêtes plus courtes formant un chemin entre i et j dans le graphe des requêtes et d'ajouter un tube entre deux sommets lorsque C requêtes ont pu être regroupées ou qu'aucun regroupement supplémentaire n'est possible. Pour trouver les requêtes avec lesquelles regrouper (i, j) il suffit de calculer un chemin entre i et j dans le graphe des requêtes privé des requêtes de taille supérieure ou égale à celle de (i, j). Afin de minimiser le nombre de sauts dans le groupage de chaque requête, le chemin recherché est un plus court chemin en nombre de tubes traversés. Dans l'exemple 3.6 avec $C = 2$, le chemin $((1, 3), (3, 4))$ est la seule succession de tubes entre les sommets 1 et 4 permettant de transporter la requête $(1, 4)$. Entre 1 et 3 les requêtes $(1, 3)$ et $(1, 4)$ sont donc regroupées : le tube $1 - 3$ est ajouté puisque $2 = C$ requêtes ont étés regroupées ; de même pour $3 - 4$. Si pour

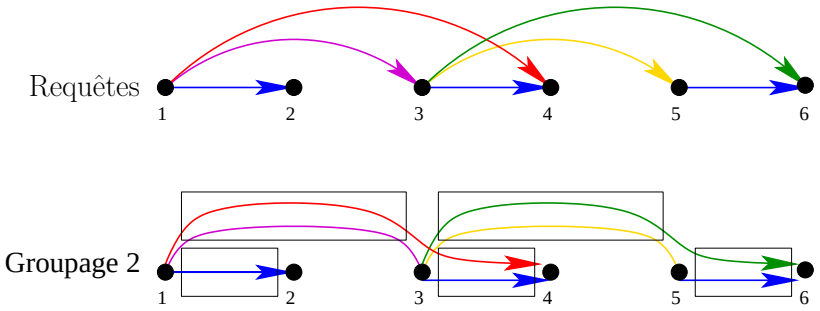


FIG. 3.6 – Exemple de groupage avec l'heuristique 2 pour $C = 2$

une requête il n'existe pas de chemin dans le graphe des requêtes plus courtes, l'heuristique 2 la décompose en deux requêtes telles qu'il existe un chemin, le plus long possible en nombre d'arcs du graphe support traversé, pour l'une des deux seulement.

Supposons que la requête $(2, 6)$ existe dans l'exemple 3.6. Il n'y a pas de chemin entre 2 et 6 mais il en existe au moins un entre 3 et 6 ($(3, 6)$ ou $((3, 5), (5, 6))$ et aussi entre 5 et 6. Deux décompositions sont possibles, soit $((2, 3), (3, 6))$ car il existe au moins un chemin de 3 à 6 et aucun de 2 à 3, soit $((2, 5), (5, 6))$. L'heuristique choisit la première car la requête $(3, 6)$ issue de cette décomposition est plus longue que $(5, 6)$.

Lorsqu'une requête est groupée avec d'autres plus courtes sur un chemin, elle est supprimée

Heuristique 2

1: classer les requêtes par taille décroissante dans une liste L
2: initialiser G_T
3: **tant que** $L \neq \emptyset$ **faire**
4: $(i, j) \leftarrow$ Premier(L)
5: $L \leftarrow L \backslash (i, j)$
6: $R \leftarrow R \backslash (i, j)$
7: calcul du plus court chemin (de capacité suffisante) p de i à j dans $G_R = (V, R)$
8: **si** il n'existe pas de chemin p **alors**
9: Soit r la plus petite requête nécessaire pour obtenir p
10: **si** $r = (i, j)$ **alors**
11: $T \leftarrow T \cup \{i - j\}$
12: aller à 2
13: **sinon**
14: $val(r) = 0$
15: $L \leftarrow L + r$
16: $R \leftarrow R + r$
17: **fin si**
18: **fin si**{Le chemin p existe dorénavant.}
19: **pour tout** $d \in p$ **faire**
20: $val(d) \leftarrow val(d) + val(i, j)$
21: **si** $val(d) = C$ **alors**
22: création du tube associé à d
23: $R \leftarrow R \backslash d$
24: $L \leftarrow L \backslash d$
25: **fin si**
26: **fin pour**
27: **fin tant que**

du graphe des requêtes et découpée pour donner des copies des requêtes avec lesquelles elle est groupée. Par exemple la requête $(1, 4)$ est découpée en une requête $(1, 3)$ et une requête $(3, 4)$. Ces deux nouvelles requêtes doivent alors être ajoutées au graphe des requêtes. Pour simplifier l'implémentation de cette heuristique nous avons choisi d'indiquer l'existence de plusieurs requêtes de mêmes extrémités i et j non pas par des arcs multiples, mais par une unique requête de poids $val(i, j)$ égal au nombre de requêtes groupées entre i et j (Algorithme 2).

L'heuristique 2 donne la solution en 5 tubes présentée à la figure 3.6. Cette heuristique est détaillée par l'algorithme 2 où $G_R = (V, R)$ représente le graphe des requêtes, $G_T = (V, T)$ le graphe des tubes.

Notons que pour un coefficient de groupage égal à deux, la deuxième heuristique revient à essayer de créer des triangles.

3.4.3 Résultats

Nous avons comparé les méthodes de résolution évoquées précédemment, les heuristiques (H1, H2), les programmes en nombres entiers sommet-arc (NAI, programme 3.17) et arc-chemin (API) et le programme mixte sommet-arc (NAF). Pour la formulation arc-chemin API nous n'avons généré qu'un ensemble restreint de chemins (variables du programme) par requête, le nombre de tubes fourni par le solveur Ilog Cplex est donc une borne supérieure.

Nous avons effectué des tests pour différents facteurs de groupage sur des graphes de requêtes complets, i.e. la demande (i, j) existe pour tous sommets i, j du graphe support tels que $i < j$.

Les graphiques suivants présentent le nombre de tubes obtenu par les différentes méthodes ainsi que les temps de calcul associés en millisecondes pour des facteurs de groupage $C = 2$ (Figure 3.7), $C = 4$ (Figure 3.8) et $C = 8$ (Figure 3.9).

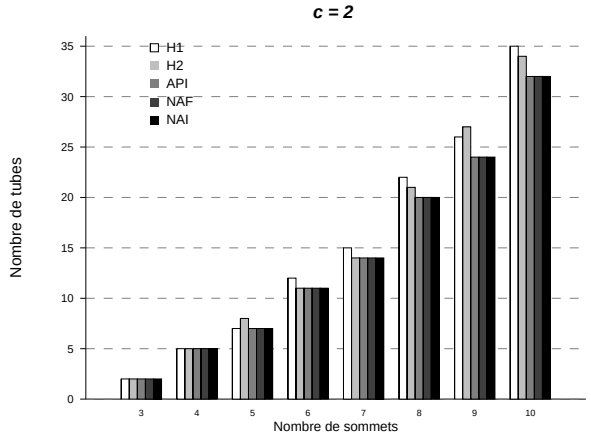
Notons que la conjecture 3.1 est vérifiée pour ces exemples, les valeurs obtenues par les programmes NAI et NAF sont identiques. Par contre, pour certains cas le calcul est plus rapide avec les variables de flots entières qu'avec les variables de flots réelles. Cependant la différence n'est pas significative et ne remet pas en cause l'intérêt de la relaxation NAF. Elle s'avère très intéressante en particulier pour un graphe à dix sommets et un facteur de groupage $C = 2$ puisque le temps de calcul passe de plus de 8h30 pour NAI à un peu plus de 55 minutes pour la relaxation NAF. Ces résultats montrent aussi que les heuristiques sont satisfaisantes car elles fournissent très rapidement des solutions proches de l'optimal.

3.5 Briques de recouvrement

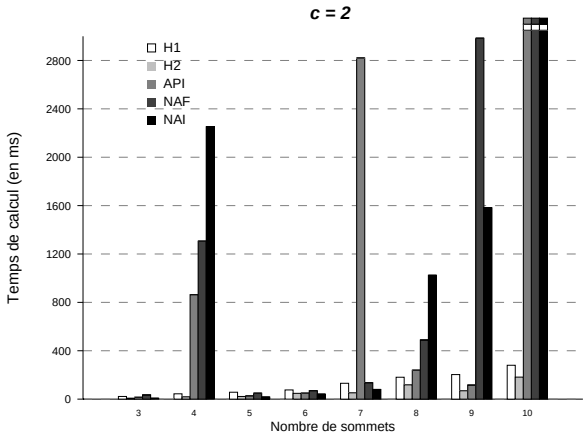
Les deux propriétés 3.1 et 3.2 ont permis aux auteurs de [BDPS03] d'envisager une approche du problème du groupage basée sur la théorie des designs [LR97]. Cette approche consiste à recouvrir le graphe des requêtes par des graphes de requêtes dont un groupage optimal atteignant la borne inférieure est connu, ou plus simplement des *briques*. Contrairement à [BDPS03] nous n'avons pas cherché à recouvrir les graphes de requêtes pour les grouper. Notre objectif est d'obtenir des graphes de requêtes dont un groupage optimal est connu afin de tester les heuristiques sur des instances du problème de grande taille pour lesquelles les solveurs ne sont plus capables de fournir de solutions optimales aux programmes linéaires en temps raisonnables.

3.5.1 Principe du recouvrement par des briques

Le principe du recouvrement consiste à déplacer le problème du groupage vers un problème de couverture du graphe des requêtes par un ou plusieurs sous graphes particuliers.

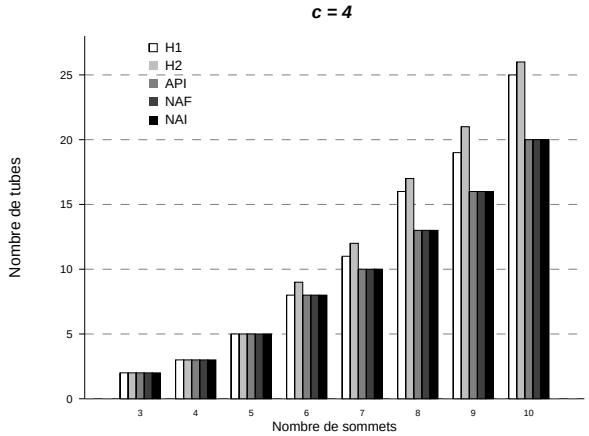


(a) Nombres de tubes obtenus pour $C = 2$.

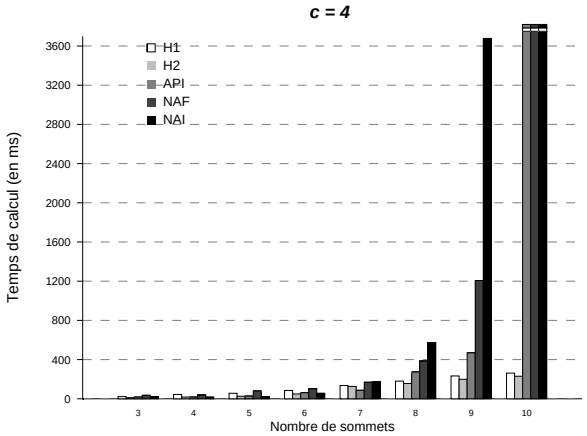


(b) Durées des calculs pour $C = 2$.

FIG. 3.7 – Résultats pour un facteur de groupage $C = 2$.

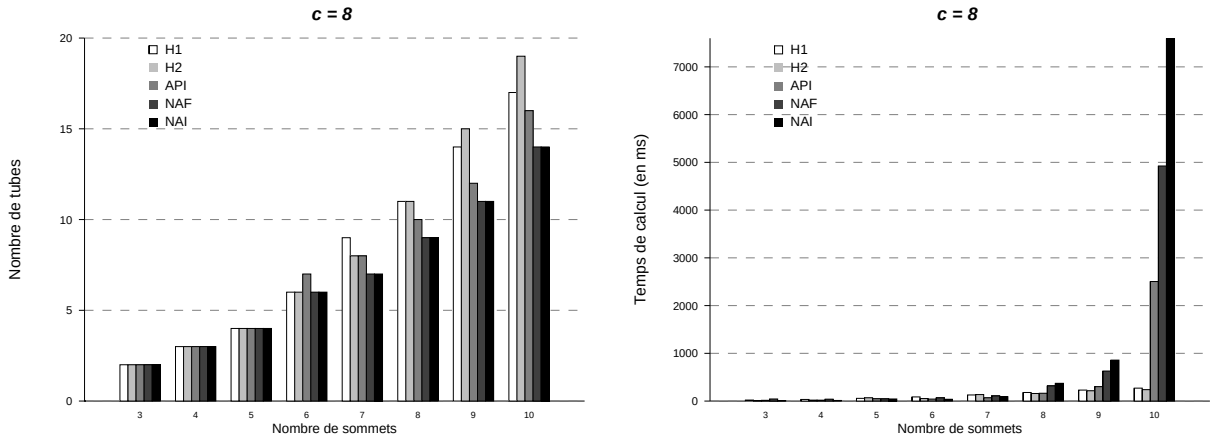


(a) Nombres de tubes obtenus pour $C = 4$.



(b) Durées des calculs pour $C = 4$.

FIG. 3.8 – Résultats pour un facteur de groupage $C = 4$.



(a) Nombres de tubes obtenus pour $C = 8$.

(b) Durées des calculs pour $C = 8$.

FIG. 3.9 – Résultats pour un facteur de groupage $C = 8$.

Supposons qu'un groupage soit connu pour un graphe de requête B^*. Alors s'il est possible de partitionner les arcs d'un graphe de requête G donné en sous graphes isomorphes à B^*, un groupage des requêtes de G peut être déduit du groupage connu pour B^*. Il suffit de grouper chaque sous ensemble de requêtes de la partition suivant le groupage connu pour B^* et de considérer l'union des ensembles de tubes ainsi choisis sur les sous ensembles de la partition.

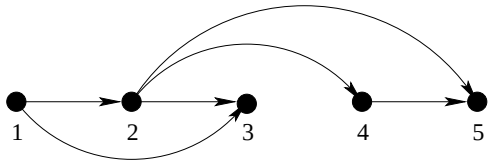
Pour un facteur de groupage égal à 2 et en faisant abstraction des orientations des requêtes, le triangle constitue un graphe de groupage optimal connu simple comme illustré à la figure 3.10(b) [BDPS03]. En effet le groupage représenté par cette figure répond à toutes les conditions de la proposition 3.2. Chacun des deux tubes contient exactement $C = 2$ requêtes dont une n'utilisant que ce tube et toutes les requêtes utilisent au plus deux tubes. Par conséquent ce groupage atteint la borne inférieure de $\frac{2|R|}{C+1} = \frac{2\times3}{3} = 2$ ce qui signifie qu'il est optimal pour cet ensemble de requêtes.

Cependant un groupage peut être optimal même si la borne inférieure n'est pas atteinte. Il faut donc faire la différence entre un groupage optimal, c'est à dire qui utilise un nombre minimum de tubes, et un groupage *parfait* qui est optimal *et* atteint la borne inférieure de la proposition 3.1. Par définition, le nombre de requêtes $|R|$ et le nombre de tubes $|T|$ doivent vérifier la relation $|T| = \frac{2|R|}{C+1}$, c'est-à-dire $|R| = \frac{|T|(C+1)}{2}$, pour qu'un groupage parfait de ces requêtes puisse exister. Dans la suite un graphe dont un groupage parfait est connu sera appelé une *brique de recouvrement*. Une brique de recouvrement *élémentaire* est alors une brique qui ne peut pas être décomposée en briques comportant moins de requêtes.

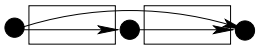
La figure 3.10(c) présente la décomposition du graphe de requête de la figure 3.10(a) en triangles qui sont donc des briques élémentaires pour un facteur de groupage $C = 2$. Le groupage obtenu par cette décomposition en deux triangles comporte 4 tubes comme le montre la figure 3.10(d), ce qui correspond à la borne inférieure pour ce graphe de requête $\frac{2|R|}{C+1} = \frac{2\times6}{3} = 4$.

D'une manière générale, le groupage obtenu par la décomposition exacte d'un graphe de requête en briques de recouvrement est aussi parfait puisqu'il vérifie les trois conditions de la proposition 3.2. En effet chaque tube d'un tel groupage est aussi un tube d'un groupage parfait. Par conséquent, il est emprunté par exactement C requêtes dont une qui n'emprunte que ce tube. De plus aucune requête n'emprunte plus de deux tubes sinon le groupage de la brique à laquelle elle appartient dans la décomposition ne serait pas parfait.

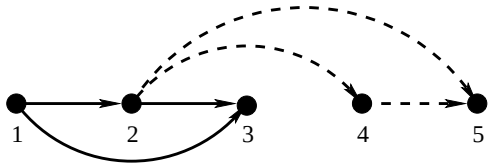
Tous les graphes de requêtes ne se décomposent pas en briques élémentaires. Une preuve en



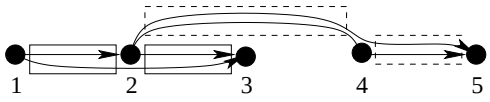
(a) Graphe de requête G



(b) Graphe de groupage optimal connu B^* : le triangle



(c) Décomposition du graphe G en deux triangles B^*



(d) Groupage optimal de G

FIG. 3.10 – Exemple de décomposition d'un graphe de requêtes en sous graphes de groupage optimal connu pour un facteur de groupage $C = 2$.

est que tous les graphes ne se décomposent pas en triangle qui est l'unique brique élémentaire existante pour un facteur de groupage égal à deux [BDPS03]. Toutefois, la décomposition en un nombre maximum de briques élémentaires permet d'obtenir d'excellents groupages, ou même des groupages optimaux.

Notre objectif n'était pas d'établir des heuristiques décomposant un graphe de requêtes en briques élémentaires comme l'avaient fait les auteurs de [BDPS03] pour un facteur de groupage $C = 2$ sur l'anneau. Il semble exister un très grand nombre de briques élémentaires pour les facteurs de groupage supérieurs à deux. Cependant la composition de ces briques entre elles représente une source infinie de graphes de requêtes de toutes tailles dont nous connaissons une solution optimale puisqu'elle atteint la borne inférieure.

Ces graphes de requêtes peuvent permettre de tester les performances des heuristiques détaillées à la section 3.4.2, ou d'autres méthodes sur des instances du problème pour lesquelles les techniques de programmation linéaire ne fournissent pas de solutions. Pour ces instances, trouver une solution optimale relève de l'impossible. En particulier nous pourrons expérimenter ces heuristiques pour des facteurs de groupage du même ordre de grandeur que dans les réseaux réels, qui nécessitent un très grand nombre de requêtes. Cependant, ces graphes ayant une propriété particulière, les tests risquent de ne pas refléter exactement le comportement des méthodes de résolution sur des graphes de requêtes vraiment quelconques. Pour limiter cet effet, il est possible d'ajouter un ensemble de requêtes dont le groupage optimal est connu mais n'est pas parfait à un ensemble de requêtes de grande taille et dont le groupage est parfait.

3.5.2 Construction de briques

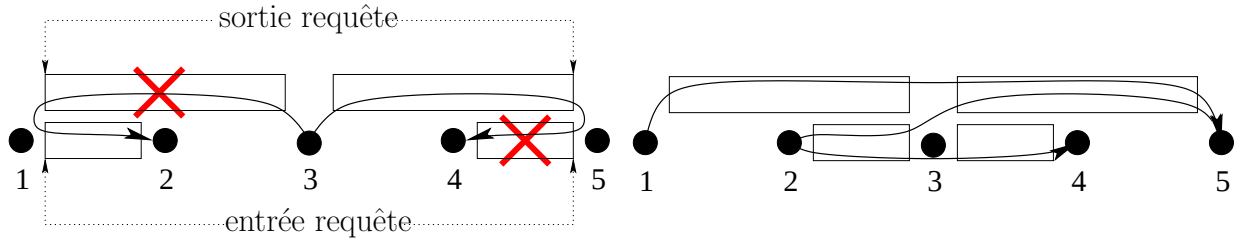
La recherche de briques repose entièrement sur la proposition 3.2. Étant donné que nous ne souhaitons pas utiliser ces briques comme dans [BDPS03] pour établir un algorithme de groupage par recouvrement par des briques élémentaires, nous ne cherchons pas spécialement à construire des briques élémentaires, mais au contraire des briques de plus grande taille possible.

3.5.2.1 Voisins

Chaque tube devant contenir C requêtes dont une n'empruntant que ce tube, un tube t doit donc partager une requête avec exactement $C-1$ autres tubes. Pour pouvoir faire ce partage, les extrémités du tube t doivent concorder avec les extrémités des $C-1$ tubes. De plus, pour que la requête puisse passer d'un tube à l'autre, elle doit utiliser l'une des extrémités concordantes des deux tubes comme sortie d'un tube et l'autre comme entrée dans le second tube. Donc deux tubes ne peuvent pas partager de requêtes si leur deux entrées ou leur deux sorties concordent, puisque les requêtes et les tubes sont orientés tous dans le même sens.

En effet si les entrées de deux tubes concordent comme pour les tubes 1-2 et 1-3 de la figure 3.11(a), pour partager une requête, cette requête doit nécessairement être orientée dans le sens inverse du chemin comme la requête (2,3). Or dans le problème de groupage étudié ici, les requêtes sont toutes orientées dans le sens du chemin. D'autre part si ce sont les deux sorties qui concordent, alors même si la requête (3,4) de la figure 3.11(a) est orientée convenablement, elle doit traverser le tube 4-5 dans le sens inverse du chemin, ce qui ne vérifie pas les contraintes du problème étudié.

Deux tubes dont la sortie de l'un et l'entrée de l'autre concordent seront dans la suite des tubes *voisins*. Un tube t doit donc avoir $C-1$ tubes voisins pour faire parti d'un groupage parfait. La figure 3.11(b) présente des exemples de tubes voisins, comme les tubes 1-3 et 3-5 ou 1-3 et 3-4 etc.



(a) Les extrémités des tubes 1-2 et 1-3 concordent, ils ne peuvent cependant pas partager de requête, de même pour les tubes 3-5 et 4-5.

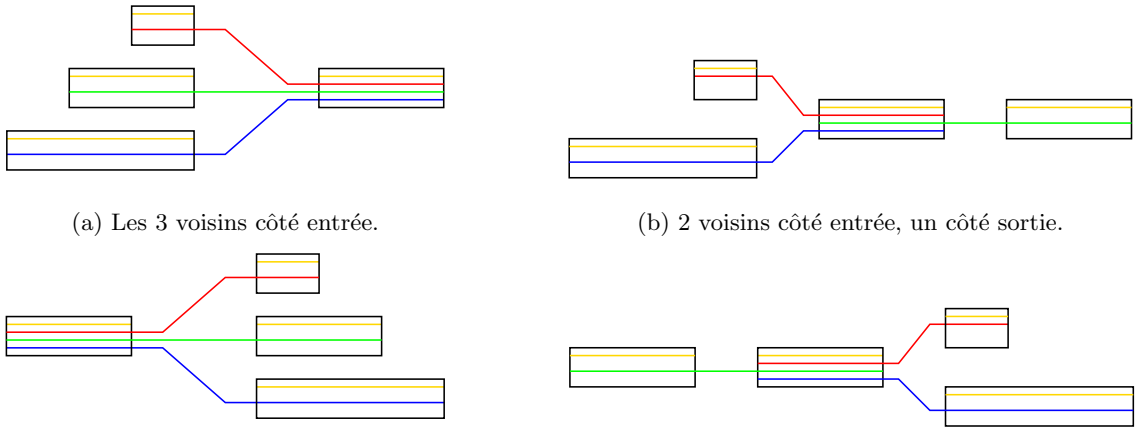
(b) Les tubes 2-3 et 3-4 sont voisins, mais pas 2-3 et 1-3 ni 3-4 et 3-5.

FIG. 3.11 – Deux tubes ne peuvent partager une requête que s'ils sont voisins.

Pour un tube il existe plusieurs façons de répartir ses voisins. Ils peuvent être voisins soit du côté de l'entrée du tube, soit de sa sortie. Il peut y avoir de 0 à $C-1$ tubes voisins du côté de l'entrée du tubes, ce qui fait C répartitions possibles pour des tubes ayant un facteur de groupage C. Notons que pour un facteur de groupage $C=2$ il existe deux répartitions qui induisent toutes les deux un triangle pour graphe de requêtes. C'est pour cela que le triangle est la seule brique élémentaire. La figure 3.12 présente les répartitions des voisins possibles pour un facteur de groupage $C=4$.

3.5.2.2 Requêtes simples

Dans la construction d'un graphe de requêtes admettant un groupage parfait, il faut faire attention à ne pas utiliser plusieurs fois la même requête car nous ne considérons que des requêtes simples (et unitaires). Sur la figure 3.13(a) tous les tubes possèdent bien $C-1=2$ voisins. Cependant la requête (1,3) est utilisée quatre fois et les requêtes (1,2) et (2,3) deux fois chacune. Le groupage ne peut donc pas être parfait avec cet agencement de tubes puisqu'il nécessiterait des requêtes multiples. Pour remédier à ce problème il suffit en fait de décaler les entrées et sorties des tubes en ajoutant des sommets afin de garantir que deux suites d'un ou deux tubes n'aient jamais leurs deux extrémités communes comme sur la figure 3.13(b). En particulier un groupage parfait



(a) Les 3 voisins côté entrée.

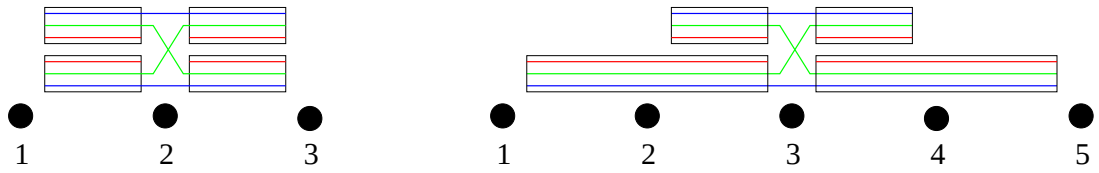
(b) 2 voisins côté entrée, un côté sortie.

(c) Les 3 voisins côté sortie.

(d) 2 voisins côté sortie, un côté entrée.

FIG. 3.12 – Les 4 répartitions des voisins d'un tube appartenant à un groupage parfait pour un facteur de groupage $C = 4$.

ne peut pas contenir deux tubes identiques, c'est à dire qui ont les deux mêmes extrémités.



(a) Pour $C = 3$ chaque tube possède suffisamment de voisins, mais le graphe de requête induit comporte nécessairement des requêtes multiples.

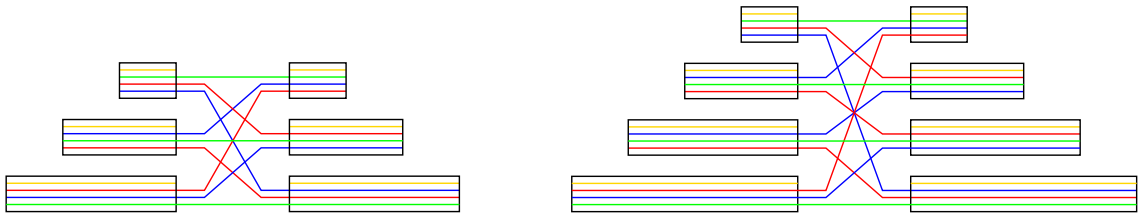
(b) En décalant les extrémités des tubes ont obtient une brique pour $C = 3$.

FIG. 3.13 – Problème de requêtes multiples et solution.

La figure 3.14 donne deux exemples de briques pour un facteur de groupage $C = 4$. Toute une famille de brique, les *étoiles* peuvent être obtenus itérativement à partir de de la brique de la figure 3.14(a). Il suffit d'ajouter des paires de tubes de plus en plus grands ainsi que les requêtes nécessaires à leurs remplissages, et de modifier le groupage des requêtes initiales comme pour la brique de la figure 3.14(b).

3.5.2.3 Combinaisons

A partir des remarques précédentes, pour construire un graphe de requêtes possédant un groupage parfait il suffit de combiner des sous graphes de requêtes soit possédant déjà un groupage parfait, soit nécessitant le partage d'un ou plusieurs tubes. Par exemple le sous graphe A encadré sur la figure 3.15 ne possède pas de groupage parfait, mais peut partager des tubes avec un autre sous graphe de requêtes. Ainsi à partir d'un graphe ayant un groupage parfait, en supprimant ou en coupant en deux un tube, et en le combinant avec un autre graphe, on peut facilement obtenir un graphe de taille supérieure.



(a) Brique pour $C = 4$.

(b) Brique pour $C = 4$.

FIG. 3.14 – Les étoiles constituent des briques pour $C = 4$. L'ensemble de requêtes de la figure 3.14(a) n'est pas inclus dans celui de la figure 3.14(b).

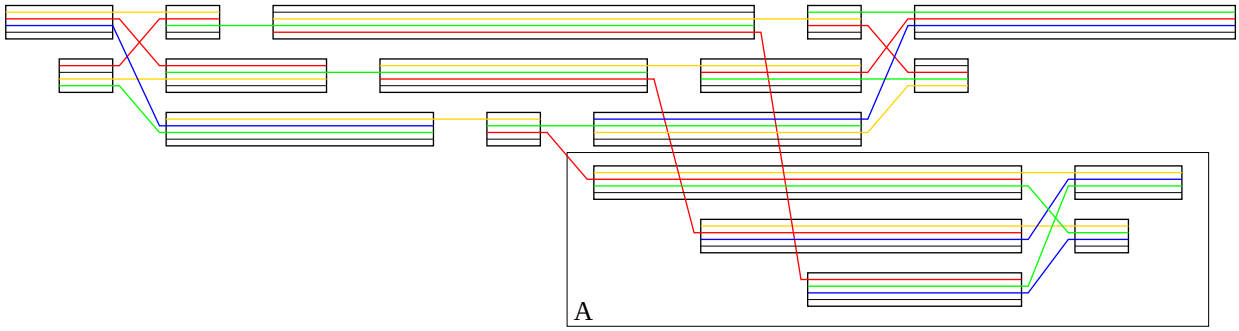


FIG. 3.15 – Combinaison de tubes pour $C = 4$

3.6 Conclusion

Les problèmes de conception de réseaux virtuels et de groupage sont d'une manière générale des problèmes difficiles et complexes aux formulations en programme linéaire lourdes. Malgré les hypothèses restrictives que nous avons considérées le groupage sur le chemin reste difficile. Nous avons toutefois proposé des heuristiques efficaces en temps de calcul et produisant des solutions d'assez bonne qualité par rapport aux performances des techniques basées sur la programmation linéaire. Cependant, les tests effectués ne concernent que des instances du problème de petites tailles et des tests supplémentaires sur des instances de tailles supérieures sont nécessaires pour confirmer ces premiers résultats. D'après nos observations, nous conjecturons qu'il est possible de relâcher certaines contraintes d'intégrité dans nos formulations en programmes linéaires. Grâce à ces formulations relâchées, nous pourrions accélérer le calcul du nombre de tubes optimal. Malgré cela, les temps de calcul pour les instances de grande taille restent relativement longs. L'automatisation de la construction de graphes de requêtes possédant un groupage parfait est par conséquent une étape à franchir avant la réalisation de tests supplémentaires.

Chapitre 4

Tolérance aux pannes et Graphes colorés

La conception de réseaux multiniveaux tolérants aux pannes n'est pas un problème facile à résoudre du fait de contraintes complexes. De plus, le trafic évolue sans cesse et il n'est pas réaliste de modifier la topologie virtuelle d'un réseau multiniveaux dès que de nouvelles requêtes arrivent et encore moins de réoptimiser le réseau à chaque fois. Un nouveau travail de conception et d'optimisation n'intervient que lorsque la situation devient critique et que le réseau n'est plus exploitable convenablement. Ainsi entre deux phases de réoptimisation, pour router et protéger de nouvelles connexions, un opérateur est amené à utiliser au mieux les ressources disponibles suivant un réseau virtuel non optimal fixé.

Certains problèmes de connexité liés aux groupes de risques (SRRG) doivent être évités pour garantir que les connexions supporteront les pannes survenant sur le réseau. Notamment, il est essentiel que les chemins de protection d'une connexion n'utilisent pas les mêmes ressources physiques que le chemin principal, c'est à dire qu'ils n'appartiennent pas aux mêmes groupes de risque. Une panne survenant sur une ressource partagée par tous ces chemins mettrait fin à la connexion.

Pour un opérateur l'enjeu économique est crucial, et les conséquences d'interruptions de connexions peuvent être dramatiques lorsqu'elles se traduisent par le non respect de contrats et des dédommagement financiers importants. Savoir utiliser un réseau non optimal est donc aussi important que de savoir concevoir un réseau optimal.

Ainsi l'objectif de ce chapitre est l'étude de l'utilisation d'un réseau multiniveaux soumis à des pannes à travers des problèmes d'optimisation relatifs à la connexité et à la vulnérabilité du réseau. Nous formulons ces problèmes grâce à une nouvelle modélisation des réseaux multiniveaux, les graphes colorés.

Nous présentons cette modélisation en section 4.1 où nous définirons les graphes colorés ainsi que les problèmes d'optimisation et de décision liés à la tolérance aux pannes que nous avons étudiés.

La section 4.2 est consacrée à la complexité de ces problèmes qui généralisent des problèmes classiques. Nous verrons que les problèmes colorés ont des propriétés très différentes de leurs équivalents de théorie des graphes classique et qui semblent dépendre d'un paramètre du graphe coloré, le span des couleurs. Suivant la valeur du span des couleurs d'un graphe, certains problèmes sont polynomiaux alors qu'ils deviennent NP-difficiles et non approximables même lorsque le span maximum est borné par une constante.

Nous proposerons à la section suivante des formulations en MILP pour certains problèmes colorés. Le temps de résolution de ces programme est lui aussi lié aux spans.

Enfin, à la lumière des résultats sur la complexité et l'inapproximabilité des problèmes colorés,

nous reviendrons en section 4.4 sur une étape importante de la modélisation d'un réseau multiniveaux en graphe coloré.

4.1 Modélisation des Réseaux et SRRG : Graphes Colorés

Le but de la modélisation en graphe coloré est de représenter un réseau multiniveaux de manière compacte, en ne gardant que les informations importantes : la topologie virtuelle et les groupes de risques auxquels appartiennent les connexions virtuelles. Dans une optique de tolérance aux pannes, la connaissance précise du routage de ces connexions sur le niveau physique n'apporte pas d'éléments utiles.

Cette modélisation permet donc de simplifier la représentation des réseaux multiniveaux, et par suite de définir simplement les problèmes d'optimisation dans ces réseaux.

4.1.1 Graphes Colorés

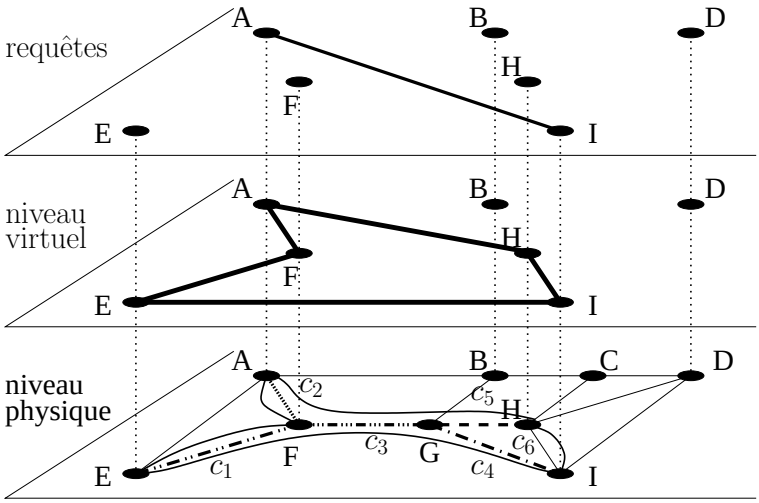
Dans un réseau multiniveaux tel que nous l'avons décrit à la section 2.3.2 du chapitre 2 (page 21), les requêtes sont routées et éventuellement protégées sur le niveau virtuel. Ce réseau virtuel se représente par un graphe que nous supposerons ici non orienté dont les sommets correspondent aux routeurs du niveau virtuel et les arêtes aux connexions virtuelles (LSP). Par hypothèse le routage des LSP sur le niveau physique est connu et donc l'ensemble des ressources physiques utilisées par chaque LSP est également connu. Pour représenter l'appartenance des connexions à un groupe de risque, chaque ressource du niveau physique est associée à une couleur[1]. L'ensemble de couleur correspondant aux risques auxquels une connexion virtuelle est soumise est affecté à l'arête représentant cette connexion dans le graphe.

La figure 4.1 illustre la représentation d'un réseau multiniveaux par un graphe muni de couleurs. Le routage du niveau virtuel sur le niveau physique du réseau de la figure 4.1(a) est indiqué au niveau physique par les courbes reliant les nœuds extrémités de chaque connexion virtuelle. La connaissance de ce routage permet de représenter ce réseau multiniveaux par le graphe de la figure 4.1(b). Il n'y a pas d'intérêt à garder dans le graphe les sommets qui ne sont pas à l'extrémité d'au moins une connexion virtuelle car ils ne sont pas atteignables, c'est pourquoi seuls les sommets A,E, F, H et I apparaissent dans le graphe de la figure 4.1(b). Au niveau physique du réseau 4.1(a), des couleurs $(c_1, c_2, \dots c_6)$ sont associées à chaque lien, par exemple le lien physique $\{F, G\}$ est associé à la couleur c_3. Chacune des connexions virtuelles correspond à une arête du graphe et est associée à un ensemble de couleurs. Par exemple, les connexions virtuelles $\{A, H\}$ et $\{E, I\}$ sont routées sur le lien $\{F, G\}$, dans le graphe elles portent donc la couleur c_3 qui symbolise le fait que ces deux connexions tombent en pannes toutes les deux lorsque le lien physique $\{F, G\}$ est coupé. La connexion virtuelle $\{A, H\}$ emprunte aussi les liens physiques $\{A, F\}$ et $\{G, H\}$, c'est pourquoi l'arête correspondante dans le graphe porte aussi les couleurs c_2 et c_5.

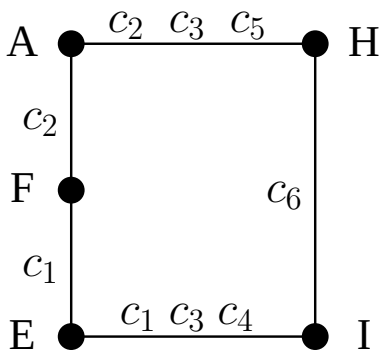
Le graphe obtenu ainsi à partir du réseau multiniveaux entre dans le cadre de la définition 4.1 d'un *réseau multicoloré* qui prend en compte la topologie des connexions du niveau virtuel, ainsi que les groupes de risques auxquels appartiennent ces connexions.

Définition 4.1 (Réseau multicoloré) *Un réseau multicoloré $\mathcal{R} = (V_\mathcal{R}, E_\mathcal{R}, \mathcal{C})$ est un graphe $(V_\mathcal{R}, E_\mathcal{R})$ non orienté, éventuellement multiple, muni d'un ensemble de couleurs $\mathcal{C}$ et dont chaque arête est associée à un sous-ensemble non vide de couleur.*

[1] Les couleurs ne sont au fond que des étiquettes.



(a) Exemple de réseau multiniveaux



(b) Graphe représentant le réseau 4.1(a)

FIG. 4.1 – Un réseau multiniveaux et sa représentation par un graphe muni de couleurs.

Grâce à cette modélisation Faragó [Far06] étudie la *vulnérabilité* des réseaux multiniveaux à travers les notions de coupe et de plus court chemin dans un réseau multicoloré. Cependant nous n'avons pas travaillé directement sur ce modèle qui de notre point de vue manque de précision.

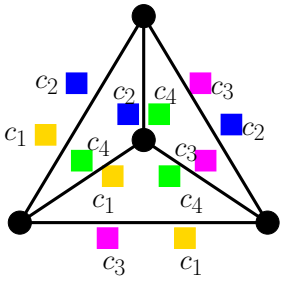
En effet cette définition permet plusieurs interprétations des couleurs, et donc des risques de panne. Un lien qui porte plusieurs couleurs est-il coupé lorsqu'une seule des couleurs est indisponible ou lorsque toutes les couleurs sont indisponibles simultanément ? En terme de graphe, doit on considérer que pour connecter les deux extrémités d'une arête donnée il faut utiliser son ensemble complet de couleurs associé ou bien utiliser une seule couleur suffit ? Cette double interprétation nous a conduit à la définition suivante des graphes colorés qui lève l'incertitude en imposant qu'une arête n'ait qu'une et une seule couleur.

Définition 4.2 (Graphe coloré) *Un graphe coloré est un triplet $G = (V, E, \mathcal{C})$ où (V, E) est un graphe non orienté, éventuellement multiple, et $\mathcal{C}$ est une partition de l'ensemble d'arêtes E.*

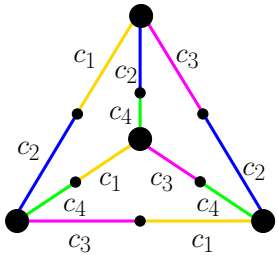
Un graphe coloré pondéré est un graphe coloré dont chaque couleur $c \in \mathcal{C}$ possède un poids $w_c \geq 0$.

Dans un graphe coloré, un sommet est adjacent à une couleur lorsqu'il est adjacent à au moins une arête de cette couleur. Le **degré coloré** d'un sommet est alors le nombre de couleurs qui lui sont adjacentes.

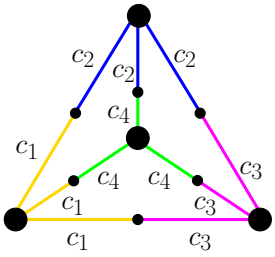
La restriction des graphes colorés aux couleurs partitionnant les arêtes n'est pas une restriction sur les réseaux modélisés par de tels graphes. Deux simples transformations permettent de transformer tout réseau multicoloré en graphe coloré (Figure 4.2). La transformation ET consiste à remplacer chaque arête multicolorée d'un réseau par un chemin contenant une arête par couleur de l'arête multicolorée (Figure 4.2(b)). Avec cette transformation, un chemin empruntant une arête du réseau utilise toutes les couleurs qu'elle porte. La transformation OU consiste à remplacer une arête du réseau par autant d'arêtes monocolorées parallèles que nécessaire pour que chacune porte une couleur de l'arête remplacée (Figure 4.2(d)). Un chemin dans le réseau empruntant cette arête n'utilise qu'une seule des couleurs qu'elle porte. Les graphes colorés modélisent donc toutes les applications où les liens d'un graphe appartiennent à un ou plusieurs groupes de risque.



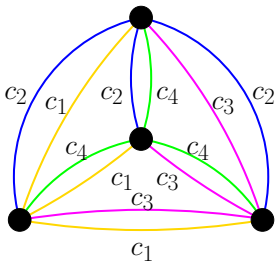
(a) un réseau multicoloré



(b) un graphe coloré issu du réseau par transformation ET



(c) un autre graphe coloré issu du même réseau par transformation ET



(d) le graphe coloré issu du réseau par transformation OU

FIG. 4.2 – Exemple de transformation ET et OU d'un réseau multicoloré en graphe coloré

Pour modéliser un réseau multiniveaux par un graphe coloré la transformation appropriée est la transformation ET. Reprenons l'exemple de la figure 4.1. Couper un seul des liens physiques $\{A, F\}$, $\{F, G\}$ ou $\{G, H\}$ suffit à couper la connexion virtuelle $\{A, H\}$, elle doit donc être modélisée dans le graphe coloré par un chemin de longueur trois comportant une arête de couleur c_2 (lien $\{A, F\}$), une arête de couleur c_3 (lien $\{F, G\}$) et une de couleur c_5 (lien $\{G, H\}$).

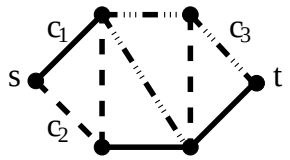
Comme l'illustrent les figures 4.2(b) et 4.2(c) plusieurs graphes colorés peuvent être obtenus par transformation ET d'un même réseau multicoloré. Suivant le graphe coloré choisi pour représenter le réseau les problèmes étudiés seront plus ou moins difficiles à résoudre ou à approximer. Le problème de la transformation ET sera traité en section 4.4 (page 86).

Notons que les graphes colorés sont un cas particulier de réseaux multicolorés et donc tous les résultats de NP-difficulté et d'inapproximabilité généraux obtenus sur les graphes colorés s'appliquent aussi aux réseaux multicolorés.

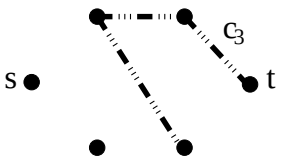
4.1.2 Problèmes Colorés

Dans les graphes classiques de nombreux problèmes consistent à trouver des sous-ensembles d'arêtes vérifiant diverses propriétés (chemin, coupe, etc). Or les arêtes ne suffisent plus à représenter la structure d'un graphe coloré, ce sont les couleurs qui jouent ce rôle. De plus une couleur étant par définition un ensemble d'arêtes, la connaissance d'un ensemble de couleurs vérifiant une propriété induit la connaissance d'un ensemble d'arêtes. C'est pourquoi les problèmes dans les graphes colorés consistent à trouver des ensembles de couleurs, et non des ensembles d'arêtes.

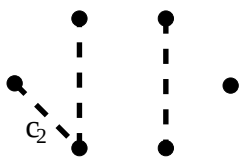
Les problèmes que nous avons étudiés peuvent se répartir en deux classes abordant la tolérance aux pannes des réseaux multiniveaux sous deux angles différents.



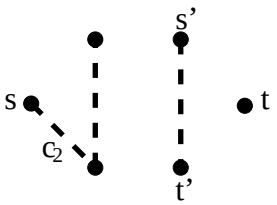
(a) un graphe coloré



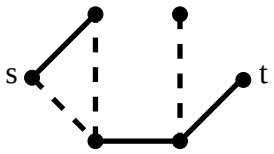
(b) la st-coupe colorée $\{c_1, c_2\}$ est supprimée



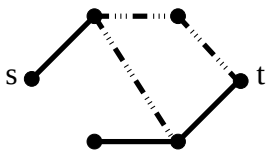
(c) la coupe colorée $\{c_1, c_3\}$ est supprimée



(d) la st-multicoupe colorée $\{c_1, c_3\}$ séparant les paires $\{s, t\}$ et $\{s', t'\}$ est supprimée



(e) un st-chemin coloré



(f) un arbre couvrant coloré

FIG. 4.3 – Exemples de chemins, coupes et arbres couvrants colorés

Les problèmes de *connexité* s'intéressent aux conditions de fonctionnement du réseau d'une manière générale, aux ressources qui doivent être disponibles pour que toutes les connexions soient assurées. Ils cherchent à connaître des ensembles de couleurs permettant de connecter des ensembles de sommets.

Les problèmes de *vulnérabilité* consistent au contraire à trouver des ensembles de couleurs dont la suppression déconnecte des ensembles de sommets et ainsi donnent une indication du nombre de pannes qui suffisent à mettre le réseau dans l'incapacité de rétablir les connexions interrompues. Ils peuvent avoir un rôle important dans la décision de réoptimiser le réseau.

4.1.2.1 Problèmes de connexité

Dans les graphes classiques le chemin est la base des problèmes de connexité, son équivalent dans les graphes colorés est le **chemin coloré**. Un st-chemin coloré est un ensemble de couleurs dont l'ensemble d'arêtes contient un chemin classique. Par exemple un graphe coloré connexe est un st-chemin coloré pour tous les sommets s et t. La figure 4.3(e) donne un autre exemple de st-chemin coloré. Vérifier qu'un ensemble de couleurs contient au moins un st-chemin au sens classique peut se faire en temps polynomial par un simple parcours des sommets. De plus, un chemin classique dans un graphe coloré induit un unique chemin coloré, il suffit de déterminer l'ensemble des couleurs portées par les arêtes du chemin classique. Dans la suite, par mesure de simplicité la désignation d'un chemin coloré pourra donc se faire dans certains cas par la désignation d'un chemin classique.

Le problème MINIMUM COLOR st-PATH (MC-st-PATH), ou st-chemin coloré minimum, consiste à trouver un chemin coloré entre les sommets s et t utilisant un nombre minimum de couleurs. Du point de vue d'un réseau un chemin coloré de nombre de couleur minimum offre une route entre deux nœuds soumise à un nombre minimum de risque de panne. En considérant la version pondérée du problème où le poids d'une couleur est la probabilité de panne du groupe de risque associé, il s'agit de trouver dans le réseau un chemin de probabilité de panne minimum.

Une autre stratégie de protection consiste à prévoir plusieurs chemins pour router une même connexion entre deux nœuds s et t, de sorte que quelle que soit la panne qui se produise au moins un des chemins fonctionne. Cette stratégie nécessite de savoir trouver dans un graphe coloré un ensemble de ***st*-chemins couleur-disjoints** (Color Disjoint Paths), c'est à dire un ensemble de chemins colorés deux à deux disjoints. Le problème consistant à trouver deux st-chemins couleur-disjoints est noté 2-Color Disjoint Paths (2-CDP) et le problème consistant à trouver un nombre maximum de st-chemins couleur-disjoints est noté Maximum Number of Color Disjoint Paths (Max-CDP).

Comme il n'est pas toujours possible de trouver des chemins couleur-disjoints entre deux sommets, un ensemble de ***st*-chemins partageant un nombre minimum de couleurs** (Minimum Overlapping Paths) permet par exemple de réaliser une protection dépendante de la panne. Le problème 2-Minimum Overlapping Paths (2-MOP) consiste à trouver deux st-chemins colorés dont l'intersection est de taille minimum.

Enfin on peut aussi s'intéresser au réseau dans sa globalité et à l'analogue coloré d'un arbre couvrant. Trouver un **arbre couvrant coloré minimum** (Minimum Color Spanning Tree ou MC-Spanning Tree) consiste à trouver un ensemble de couleurs de taille minimum dont les arêtes contiennent un arbre couvrant au sens classique (Figure 4.3(f)). Un arbre couvrant coloré ne garde de son équivalent classique que la notion de connexité, plusieurs chemins colorés peuvent exister entre deux sommets avec cette définition, par exemple un graphe coloré connexe est un arbre couvrant coloré pour lui-même. D'un point de vue pratique, un arbre couvrant coloré est un ensemble minimum de ressources nécessaires à la connexité du réseau et ainsi un ensemble stratégique de ressources à protéger. La taille d'un arbre couvrant coloré minimum donne également une borne supérieure atteinte du nombre maximum de pannes au delà duquel le réseau est déconnecté, c'est le nombre de couleurs n'appartenant pas à l'arbre.

4.1.2.2 Problèmes de vulnérabilité

Les problèmes de vulnérabilité sont tous les problèmes de coupe. Une **coupe colorée** est un ensemble de couleurs dont les arêtes contiennent une coupe au sens classique. C'est un ensemble de couleurs dont la suppression déconnecte le graphe en au moins deux parties comme le montre la figure 4.3(c).

De même une ***st*-coupe colorée** est un ensemble de couleurs dont les arêtes contiennent une st-coupe classique Figure 4.3(b). On peut aussi définir une **multicoupe colorée**, c'est un ensembles de couleur dont les arêtes contiennent une multicoupe classique, c'est à dire un ensemble de couleurs dont la suppression déconnecte plusieurs paires de sommets simultanément $\{s_1, t_1\}$, $\{s_2, t_2\}$, ..., $\{s_k, t_k\}$ (Figure 4.3(d)).

Une coupe colorée minimum (Minimum Color Cut ou MC-Cut) se traduit dans un réseau par un ensemble critique de ressources dont la panne provoque la coupure du réseau en au moins deux parties. Plus la coupe minimum colorée du graphe coloré représentant un réseau est petite, plus le réseau est vulnérable aux pannes puisque son fonctionnement dépend du fonctionnement de peu de ressources. La coupe minimum colorée donne donc une information précieuse sur la topologie virtuelle. Lorsqu'elle est petite la topologie virtuelle nécessite une réoptimisation. De même, une st-coupe colorée minimum (Minimum Color st-Cut ou MC-st-Cut) de petite taille indique que la connexion $\{s, t\}$ risque d'être interrompue si seulement un faible nombre de pannes survient dans le réseau. Une multicoupe colorée minimum (Minimum Color Multi-Cut ou MC-Multi-Cut) généralise cette remarque pour plusieurs connexions simultanément.

4.1.3 État de l'art

En 1997 est publié le premier article traitant du problème Minimum Color Spanning Tree alors nommé *minimum labeling spanning tree* dans le cadre de la conception de réseaux [CL97]. Les arêtes du graphe sont munies de labels représentant la nature du lien correspondant (optique, radio, téléphonique etc) et l'objectif est de construire un réseau utilisant le moins de technologies différentes. La complexité de ce problème est donnée par une réduction du problème Minimum Set Cover et deux heuristiques sont proposées. Certains des problèmes colorés que nous avons étudiés sont également introduit dans cet article.

Ensuite [KW98] et [Wir01] prouvent un premier facteur d'approximation de $2\ln|V|+1$ pour l'une des heuristiques de [CL97]. Ce facteur est amélioré à $\ln(|V|-1)+1$ dans [WCX02] où un facteur d'inapproximabilité à $(1-\varepsilon)\ln(|V|-1)$ est prouvé pour tout $\varepsilon > 0$ sauf si $NP \subseteq TIME(n^{\log\log n})$ (voir annexe A). Ces travaux ayant abouti aux mêmes facteurs d'inapproximabilité et d'approximabilité que pour le problème Minimum Set Cover, la littérature s'est tarie sur le problème Minimum Color Spanning Tree. Nous exposons l'algorithme d'approximation pour ce problème et les preuves de complexité et d'inapproximabilité à la section 4.2.5.2.

Dans [Wir01] un facteur d'inapproximabilité pour le problème *minimum label path* qui n'est autre que le problème Minimum Color *st*-Path est prouvé grâce à une réduction au problème Red Blue Set Cover. Cette réduction est également évoquée dans [CDKM00]. Nous proposons en section 4.2.5 une réduction différente qui améliore ce facteur.

Plus récemment, le problème Minimum Color *st*-Path à été étudié dans le contexte des SRRG par [DS04a] et [YVJ05]. Dans [YVJ05] une nouvelle preuve de complexité par réduction du problème Minimum Set Cover au Minimum Color *st*-Path est donnée ainsi que des heuristiques et une formulation en programme linéaire mixte. Un algorithme polynomial pour une classe particulière de graphes colorés est donné dans [DS04a]. Dans les graphes considérés, les arêtes d'une même couleur forment une étoile. Nous élargirons cette classe de graphes en section 4.2.3.1.

Trois preuves différentes de complexité du problème 2-Color Disjoint Paths ont été publiées dans [Hu03], [YJ04] et [EBR+03]. Nous en présenterons une quatrième pour un cas particulier en section 4.2.4.1. Ce problème 2-CDP à été étudié pour la première fois dans [Bha94] dans le cas de graphes orientés. De nombreuses heuristiques issues des chemins augmentant de [Suu74] ont été proposées. L'idée principale de ces méthodes consiste à calculer un premier chemin, supprimer du graphe les couleurs utilisées par ce chemin et calculer un second chemin. Bien que cette méthode et ses variantes [Bha97, BLE+02, LTS05, DGM94] semblent donner de bons résultats dans l'ensemble, son inconvénient majeur est que suivant la topologie du graphe il n'existe pas nécessairement de second chemin disjoint du premier même si une paire de chemins disjoints existe dans le graphe. Ce problème est connu sous le nom de *trap topology* [DGM94]. Dans [XXQ03b, LKD02, XXQL03, OMSY02, TR04a] des heuristiques sont proposées pour tenter d'éviter les *trap topologies*, ces méthodes sont résumées en particulier dans [XXQL03]. Dans ces travaux des formulations en programmes linéaires en nombres entiers ont aussi été étudiées. D'autres travaux proposent des formulations en programme linéaire mixte pour des problèmes très proches de 2-CDP, comme [SYR05] qui compare, en terme d'utilisation des ressources, les protections dédiées et partagées dans le contexte des SRRG. Notons qu'un algorithme polynomial pour le problème 2-CDP existe dans le cas particulier où les arêtes d'une même couleur forment une étoile [DS04a]. Comme pour le problème Minimum Color *st*-Path nous montrerons que ce problème est polynomial pour une classe de graphe plus large.

Le problème 2-Minimum Overlapping Paths a été défini dans [aKS01] et montré NP-difficile dans [Hu03], une formulation en programme linéaire en nombres entiers en est donnée ainsi que

pour le problème consistant à trouver deux chemins couleur-disjoints de coût total minimum.

Des variantes de ces problèmes sont présentées dans [YVJ05] consistant à trouver deux chemins arêtes disjoints partageant un nombre minimum de couleurs. Toutefois dès lors qu'une couleur appartient aux deux chemins, l'intérêt d'avoir deux chemins arête disjoints est limité dans le contexte de la tolérance aux pannes dans les réseaux multiniveaux. L'autre problème étudié dans [YVJ05] consiste à trouver deux chemins arête disjoints dont la somme des couleurs de ces deux chemins est minimum.

Le problème Max-CDP apparaît dans le contexte de l'authentification d'utilisateurs dans un réseau. Sa complexité est prouvée dans [JRN04] par une réduction du problème Maximum 3 Satisfiability et dans [GZLK01] le problème de décision associé est prouvé NP-Complet par une réduction du problème Independant Set. Nous verrons qu'une réduction proche permet de déduire un facteur d'inapproximabilité.

Les problèmes de vulnérabilité n'apparaissent pas dans la littérature excepté le problème de la coupe minimum colorée dans un réseau multicoloré qui est montré NP-difficile dans [Far06]. Cette preuve utilise le fait qu'une arête du réseau multicoloré porte plusieurs couleurs, elle n'implique donc pas la NP-difficulté du problème MC-Cut.

4.2 Complexité des Problèmes Colorés

D'une manière générale les problèmes colorés sont NP-difficiles alors que leurs équivalents classiques sont polynomiaux. Nous verrons en section 4.2.1 d'autres différences entre les problèmes classiques et colorés qui nous amènerons à définir un paramètre important de leur complexité, le span d'une couleur, en section 4.2.2. Les sections 4.2.3, 4.2.4 et 4.2.5 seront consacrées à l'étude de la complexité et de l'inapproximabilité des problèmes colorés. L'annexe A rappelle les définitions de tous les problèmes de la littérature auxquels nous faisons référence dans la suite.

4.2.1 Comparaison avec les problèmes classiques

Nous allons maintenant montrer à travers quelques exemples que les problèmes colorés diffèrent des problèmes de théorie des graphes classique non seulement par leurs complexités mais aussi par leurs relations mutuelles et certaines propriétés importantes.

4.2.1.1 Minimum Color *st*-Cut et Nombre de chemins couleur-disjoints, Minimum Color Cut et Nombre d'arbre couvrant couleur-disjoints

Depuis le contre exemple donné dans [aKS01], il est prouvé que le nombre maximum de chemins couleurs disjoints et la *st*-coupe colorée minimum ne sont pas égaux. La proposition 4.1 précise la relation entre ces deux éléments. Elle signifie en particulier que contrairement au cas des graphes classiques où la relation *max flow-min cut* existe, la valeur d'une *st*-coupe colorée minimum ne donne aucune garantie ou information réellement exploitable sur l'existence d'au moins deux *st*-chemins couleur-disjoints. Dans les graphes classiques cette relation est essentielle pour aborder la tolérance aux pannes. De plus il est prouvé qu'il existe au moins $C/2$ arbres couvrants disjoints dans un graphe classique de coupe minimum C [GGL95]. Cette relation n'est plus vraie dans les graphes colorés.

Proposition 4.1 *Pour tout $k \in \mathbb{N}$, il existe un graphe coloré et deux sommets s et t de ce graphe, tels qu'une st-coupe colorée minimum est de valeur k alors qu'il n'existe aucune paire de st-chemins*

couleur-disjoints. De plus dans ce graphe la coupe colorée minimum est de valeur k alors qu'il n'existe aucune paire d'arbres couvrants colorés couleur-disjoints.

Preuve: Soit $k \in \mathbb{N}$ une constante. Nous construisons un graphe coloré G à $\binom{2k}{k}+1$ sommets et $2k$ couleurs. G est un chemin composé de $\binom{2k}{k}$ arêtes multiples en sorte que chaque sous-ensemble de k couleurs corresponde à exactement une arête multiple, chacune des k couleurs d'un sous-ensemble appartient à l'une des k arêtes parallèles composant l'arête multiple. La figure 4.4 représente le graphe coloré obtenu pour $k = 2$. Soient s et t les extrémités du chemin. Un arbre couvrant coloré de G est simplement un st-chemin coloré, et une coupe colorée est en fait une st-coupe colorée.

Supposons qu'il existe un st-chemin coloré utilisant $l \leq k$ couleurs. Alors il existe un sous-ensemble d'au moins k couleurs qui n'est pas utilisé par ce chemin. Ceci implique que le chemin ne traverse aucune arête multiple composée de ces couleurs et mène à une contradiction.

Par conséquent, un chemin coloré de G, ou un arbre couvrant coloré, utilise au moins $k+1$ couleurs. Comme seulement $2k$ couleurs sont disponibles, la proposition 4.1 suit. ∎

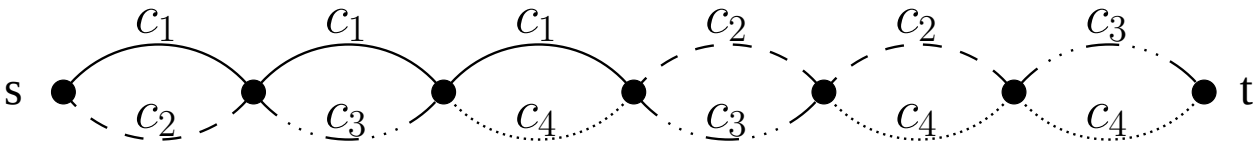


FIG. 4.4 – Aucune paire d'arbres couvrants colorés couleur-disjoints alors que la coupe vaut 2.

D'autres familles de graphes colorés existent qui permettent de prouver cette proposition pour la relation entre une st-coupe colorée et le nombre de st-chemins couleur-disjoints, par exemple la famille de graphes suivante.

Un graphe de cette famille contient k^2 couleurs et $2k^2$ arêtes. Soient s, u et t trois sommets. Dans un premier temps k chemins couleur-disjoints parallèls de k couleurs chacun sont créés entre s et u. Cela est possible car k^2 couleurs sont disponibles. Ainsi chaque couleur appartient à exactement une arête entre s et u. Ensuite, k chemins parallèles couleur-disjoints sont créés entre u et t de longueur

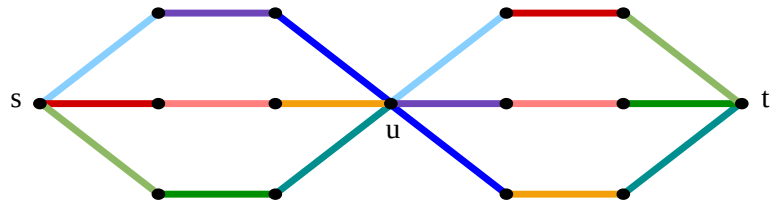


FIG. 4.5 – Aucune paire de st-chemins couleur-disjoints alors que la st-coupe vaut $k = 3$.

k également. Un chemin connectant u et t doit contenir exactement une couleur de chacun des k chemins connectant s et u comme illustré à la Figure 4.5. De cette manière un su-chemin et un ut-chemin partagent exactement une couleur.

En conséquence, un chemin de s à t utilise k couleurs dans sa section entre s et u, chacune appartenant à exactement un des ut-chemins, et ainsi il n'existe pas de paire de st-chemins couleurs disjoints dans ce graphe.

Une borne supérieure sur le ratio entre le nombre maximum de chemins couleurs disjoints et la st-coupe colorée minimum dépendant de la taille du graphe et du nombre de couleur existe peut-être. Notons que lorsque la st-coupe minimum utilise toutes les couleurs du graphe il existe alors un st-chemin de chaque couleur.

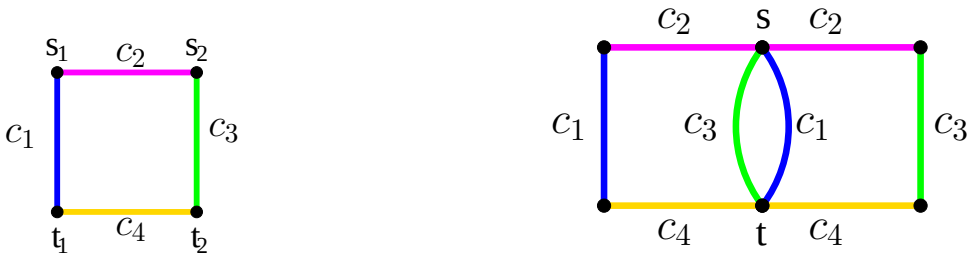
4.2.1.2 Minimum Color *st*-Cut et Minimum Color Multi-Cut

Une autre paire de problèmes, MC-*st*-Cut et MC-Multi-Cut, présente une différence majeure par rapport aux problèmes classiques.

Proposition 4.2 MC-Multi-Cut *est équivalent à* MC-*st*-Cut.

Preuve: Premièrement, MC-*st*-Cut est un cas particulier de MC-Multi-Cut.

Ensuite, soit une instance de MC-Multi-Cut, prenons k copies $G_1, \ldots, G_k$ du graphe G, une par paire s_i, t_i à déconnecter dans l'instance de MC-Multi-Cut. Fusionnons tous les sommets $s_i \in G_i$ (resp. t_i) en un seul sommet s (resp. t). De cette manière les k copies ne forment plus qu'un seul graphe (Figure 4.6). Une *st*-coupe colorée dans ce nouveau graphe est trivialement une multicoupe coloré dans le graphe d'origine G. ∎



(a) Une instance de MC-Multi-Cut... (b) ... une instance MC-*st*-Cut

Fig. 4.6 – Transformation d'une instance de MC-Multi-Cut en MC-*st*-Cut.

Dans la suite nous étudions le problème Minimum Color *st*-Cut, les résultats sont identiques pour le problème Minimum Color Multi-Cut.

4.2.1.3 Minimum Color *st*-Cut, Minimum Color *st*-Path et les graphes série-parallèles

Les problèmes MC-*st*-Cut et MC-*st*-Path sont équivalents dans un cas particulier des graphes série-parallèles. Soit G_1 un graphe série-parallèle. G_1 est construit à partir d'une arête dont les extrémités sont les sommets s_1 et t_1. La construction consiste en une succession de deux opérations : *diviser* une arête en deux arêtes consécutives par l'insertion d'un sommet et *ajouter* une arête en parallèle d'une autre.

Un autre graphe série-parallèle G_2 peut être construit en sorte que MC-*st*-Path dans G_1 est équivalent à MC-*st*-Cut dans G_2 et vice versa, de la manière suivante.

Soient s_2 et t_2 les extrémités d'une arête isolée. G_2 est construit à partir de cette arête en appliquant la succession d'opérations utilisée pour construire G_1 mais dans laquelle les opérations *diviser* et *ajouter* sont inversées. Ainsi les arêtes de G_1 et G_2 sont en bijection. Les couleurs des arêtes de G_1 peuvent donc être reportées sur les arêtes correspondantes dans G_2 d'après cette bijection.

Un s_1t_1-chemin dans G_1 est donc trivialement une s_2t_2-coupe dans G_2 puisque un chemin consiste à traverser des arêtes connectées en série alors qu'une coupe consiste à couper des arêtes "parallèles".

Proposition 4.3 *Dans les graphes série-parallèles les problèmes* Minimum Color *st*-Path *et* Minimum Color *st*-Cut *sont équivalents lorsque s et t représentent les extrémités du graphe.*

Dans la suite, tous les résultats d'inapproximabilité et de complexité sont basés sur des constructions de graphes série-parallèles particuliers dont les extrémités sont aussi les extrémités des chemins colorés cherchés. Grâce à l'équivalence de MC-*st*-Path et MC-*st*-Cut dans ce cas précis, toutes les preuves sont données seulement pour MC-*st*-Path.

4.2.1.4 Minimum Color *st*-Path

L'algorithme de Dijkstra permettant de calculer un plus court chemin dans un graphe classique repose sur une propriété des plus courts chemins qui n'est pas transférée aux chemins colorés. Contrairement à un plus court chemin classique, un chemin coloré minimum n'est pas nécessairement composé de chemins colorés minimum. Dans le graphe représenté à la figure 4.7

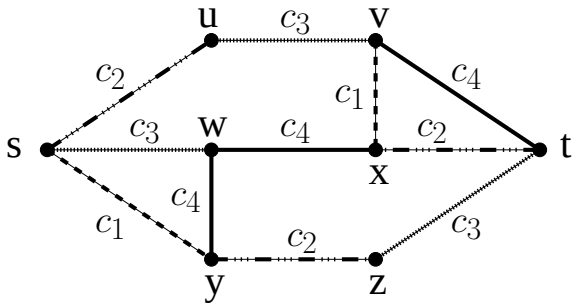


Fig. 4.7 – Un chemin coloré minimum n'est pas constitué de chemins colorés minimum.

l'ensemble de couleurs $\{c_1, c_4\}$ est un *st*-chemin coloré minimum. Il n'existe qu'un seul chemin classique dans le sous-graphe induit par ces couleurs, il passe par s, y, w, x, v et t. Or la couleur c_3 est un sw-chemin coloré minimum, contrairement à l'ensemble $\{c_1, c_4\}$ qui induit le chemin classique passant par s, y et w. Ainsi pour connecter s et w une seule couleur suffit mais pour connecter s et t, il vaut mieux utiliser deux couleurs entre s et w.

Proposition 4.4 *Un st-chemin coloré minimum n'est pas composé de chemins colorés minimum.*

Par conséquent il n'est pas possible d'adapter l'algorithme de Dijkstra au cas coloré. Il ne suffit pas de savoir quel ensemble minimum de couleurs permet de connecter deux sommets, il faut savoir si ce sont les meilleures pour atteindre la deuxième extrémité du chemin.

4.2.1.5 Minimum Color Cut

Dans les graphes classiques et même dans les hypergraphes, la coupe possède deux propriétés importantes : la *sous-modularité* et la *symétrie*. Une fonction réelle f sur 2^V l'ensemble des parties d'un ensemble V est sous-modulaire ssi $\forall A, B \subseteq V,\ f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$. Une fonction f sur un ensemble V est symétrique ssi $\forall A \subseteq V,\ f(A) = f(V - A)$. L'algorithme de Nagamochi et Ibaraki [NI92] permet de calculer une coupe minimum en temps polynomial grâce à cette propriété. R. Rizzi [Riz99] a montré que l'algorithme reste efficace pour une classe de fonctions encore plus large : les fonctions *symétriques*, *monotones* et *cohérentes*.

Une fonction réelle f sur l'ensemble $2^V \times 2^V$ est monotone si $g(S, T') \leq g(S, T)$ pour tout ensembles $S, T \subseteq V$ disjoints et tout $T' \subseteq T$. Elle est cohérente si $g(A, W \cup B) \geq g(B, W \cup A)$ pour tout $A, B, W \subseteq V$ disjoints tels que $g(A, W) \geq g(B, W)$.

Par définition, une coupe colorée est un ensemble de couleurs dont les arêtes contiennent une coupe au sens classique. Or la coupe classique peut être définie dans un graphe $G = (V, E)$ par la

fonction $\Gamma(S,T) = |\{st \in E | s \in S, t \in T\}|$ pour tout $S, T \subseteq V$ [Riz99]. Le problème de la coupe minimum est alors de trouver un ensemble $S \subseteq V$ tel que $\Gamma(S, V-S)$ est minimum.

A partir de cette définition de la coupe classique nous pouvons définir la coupe colorée dans un graphe coloré $G = (V, E, \mathcal{C})$ par la fonction $\Gamma_c(S,T) = |\{c \in \mathcal{C} | st \in c \subseteq E \text{ et s} \in \text{S}, \text{t} \in \text{T}\}|$.

La coupe colorée Γ_c est bien une fonction symétrique, elle est également monotone car si $T' \subseteq T$ et S disjoint de T, il ne peut pas y avoir plus de couleurs entre S et T' qu'entre S et T. Par contre, la coupe colorée n'est pas cohérente comme le prouve la figure 4.8. En effet, dans ce graphe $\Gamma_c(A,W) = 2$ et $\Gamma_c(B,W) = 1$ mais $\Gamma_c(A, W \cup B) = 2$ est inférieur à $\Gamma_c(B, W \cup A) = 3$.

Par conséquent la coupe colorée ne possède pas les propriétés nécessaires pour que l'algorithme de Rizzi adapté au cas coloré permette de calculer une coupe minimum colorée en temps polynomial.

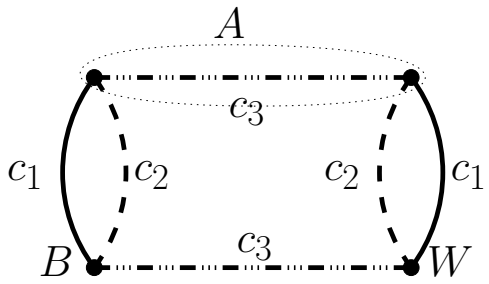


FIG. 4.8 – La coupe colorée n'est pas cohérente.

4.2.2 Span d'une couleur

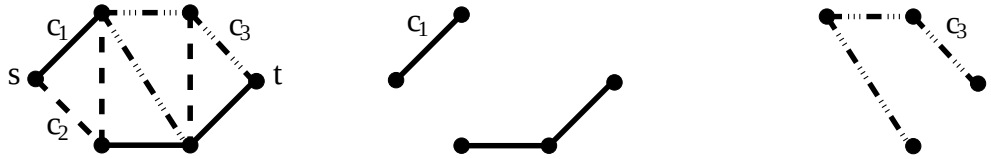
Intuitivement les différences entre les problèmes colorés et leurs équivalents classiques doivent provenir du fait qu'une arête dans un graphe classique a une influence locale sur le graphe, alors qu'une arête dans un graphe coloré représente toute une couleur, et par conséquent tout un ensemble d'arêtes. Dans un graphe coloré une arête a donc une incidence plus globale à travers sa couleur (une partie plus importante du graphe est touchée). Pour essayer de quantifier l'influence d'une couleur on définit donc son **span**.

Définition 4.3 (Composantes d'une couleur) *Les composantes d'une couleur sont les composantes connexes du sous-graphe induit par les arêtes de cette couleur.*

La figure 4.9 présente un graphe coloré (4.9(a)) ainsi que le sous-graphe induit par les arêtes de la couleur c_1 (4.9(b)) et celui induit par les arêtes de la couleur c_3 (4.9(c)). Dans ces graphes seuls les arêtes de la couleur choisie et les sommets qui leur sont incidents apparaissent.

Définition 4.4 (Span d'une couleur) *Le span d'une couleur est le nombre de composantes de cette couleur.*

Comme la suite le montrera, une couleur de span 1 est plus simple à manipuler qu'une couleur de span élevé. De plus lorsque toutes les couleurs sont de span 1, la plupart des problèmes d'optimisation sont polynomiaux. Si le span maximum est borné, les problèmes sont difficiles mais approximables, alors que lorsque le span des couleurs est quelconque, les problèmes sont difficiles et mal approximables.



(a) un graphe coloré (b) couleur de span 2 (c) couleur de span 1

FIG. 4.9 – Span d'une couleur

4.2.3 Cas polynomiaux

Bien que les problèmes colorés soient très différents des problèmes classiques et en général NP-difficiles, il existe des classes de graphes colorés pour lesquels de nombreux problèmes sont polynomiaux.

4.2.3.1 Graphe coloré de span maximum 1 et Hypergraphe

Lorsque toutes les couleurs d'un graphe coloré sont de span 1 il correspond à un hypergraphe dont l'ensemble de sommets est identique à celui du graphe coloré. L'hypergraphe comporte une hyperarête pour chaque couleur, il s'agit de l'ensemble des sommets adjacents à la couleur qui lui est associée (Figure 4.10).

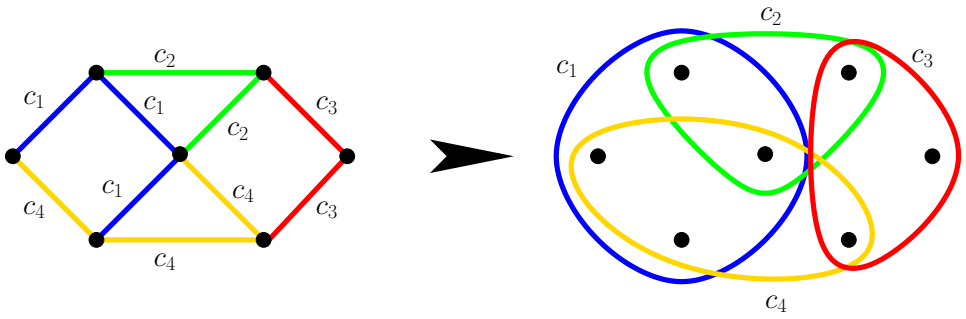


FIG. 4.10 – Transformation d'un graphe coloré en hypergraphe

Dans les hypergraphes, trouver un plus court st-chemin, une st-coupe minimum ou une coupe minimum est un problème polynomial [Riz99].

La construction suivante met en évidence que les problèmes 2-COLOR DISJOINT PATHS et 2-MINIMUM OVERLAPPING PATHS sont également polynomiaux (Figure 4.11).

Soit $G = (V, E, \mathcal{C})$ un graphe coloré dont toutes les couleurs sont de span 1, un graphe $H = (V_H, E_H)$ est construit dont chaque sommet représente une couleur de $\mathcal{C}$. Une arête connecte deux sommets de H si les couleurs qu'ils représentent sont adjacentes dans G. Deux sommets s et t sont ajoutés ainsi que des arêtes entre s (resp. t) et chaque sommet de H représentant une couleur adjacente à s (resp. t) dans G.

Deux st-chemins couleur-disjoints dans G sont simplement deux chemins sommet-disjoints entre s et t dans H. Inversement deux st-chemins sommet-disjoints dans H correspondent à deux ensembles de couleurs disjoints dans G. Comme chaque sommet de H représente une composante

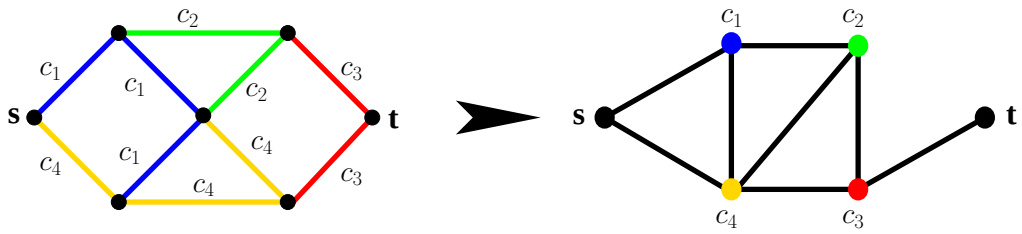


FIG. 4.11 – Transformation d'un graphe coloré de span maximum 1, les problèmes MC-*st*-PATH, MC-*st*-CUT, 2-CDP et 2-MOP se réduisent à leurs équivalents en nombre de sommets dans un graphe classique.

connexe d'une couleur et que deux sommets sont adjacents dans H s'ils sont associés à des couleurs adjacentes dans G, il est possible à partir des deux ensembles de couleurs disjoints de reconstituer en temps polynomial deux chemins couleur-disjoints entre s et t. Il suffit pour cela de calculer un st-chemin dans chacun des deux sous-graphes induits par les deux ensembles de couleurs disjoints.

Pour le problème 2-MOP, notons que deux st-chemins passent nécessairement tous les deux par chaque sommet constituant une st-coupe de taille 1 en nombre de sommets.

Proposition 4.5 *Lorsque toutes les couleurs sont de span 1,* MC-*st*-PATH, MC-*st*-CUT, MC-CUT, *2*-CDP *et 2*-MOP *sont polynomiaux.*

Notons que cette proposition étend un résultat de [DS04a] pour MC-*st*-PATH et 2-CDP, puisque le cas de couleurs en étoile, i.e. toutes les arêtes d'une même couleur ont un même sommet en commun, est un cas particulier de couleur de span 1.

4.2.3.2 Coupe colorée

Outre le cas où toutes les couleurs sont de span 1, le problème MINIMUM COLOR CUT est polynomial lorsque le nombre d'arêtes par couleur est borné par une constante k.

Proposition 4.6 *Lorsque chaque couleur contient au plus k arêtes, pour une constante $k \in \mathbb{N}$ donnée,* MINIMUM COLOR CUT *peut être résolu en temps polynomial.*

Preuve: Soient $G = (V, E, \mathcal{C})$ un graphe coloré et S la valeur d'une coupe minimum *au sens du nombre d'arête classique* dans G. Alors la valeur d'une coupe colorée minimum appartient à $[S/k, S]$ et la taille de l'ensemble d'arêtes associé à cette coupe colorée appartient à $[S, kS]$. Dans [Kar93], Karger montre qu'il existe au plus $|V|^{2k}$ coupes de taille comprise dans $[S, kS]$ et précise de plus qu'elles peuvent être générées en temps polynomial.

Par conséquent trouver une coupe minimum colorée peut se faire en temps polynomial en énumérant et évaluant chacune des coupes de nombre d'arête compris entre S et kS. ∎

Notons que lorsque le degré coloré maximum du graphe est borné, les problèmes de coupe MC-CUT, MC-*st*-CUT et MC-MULTI-CUT sont également polynomiaux, puisque tous les sous-ensembles de couleurs de taille inférieure à la borne peuvent être énumérés en temps polynomial.

4.2.3.3 Nombre de couleur de span > 1 borné

Il est possible d'énumérer en temps polynomial tous les sous ensembles de couleurs de span strictement supérieur à 1 lorsque leur nombre est borné. Une méthode polynomiale basée sur cette

énumération permet de résoudre les problèmes Minimum Color *st*-Path, Minimum Color Cut et Minimum Color *st*-Cut. Pour cela construisons à partir d'un graphe coloré G un graphe $H = (V_H, E_H)$ inspiré de la construction utilisée en section 4.2.3.1 (page 69).

Chaque sommet de H représente une composante d'une couleur de $\mathcal{C}$. Deux sommets de V_H sont reliés par une arête de E_H si les deux composantes associées possèdent un sommet en commun dans G. Pour les problèmes MC-*st*-Path et MC-*st*-Cut deux sommets s et t sont ajoutés à V_H et une arête de E_H connecte s (resp. t) à un sommet v de V_H si s (resp. t) appartient à la composante représentée par v (Figure 4.12).

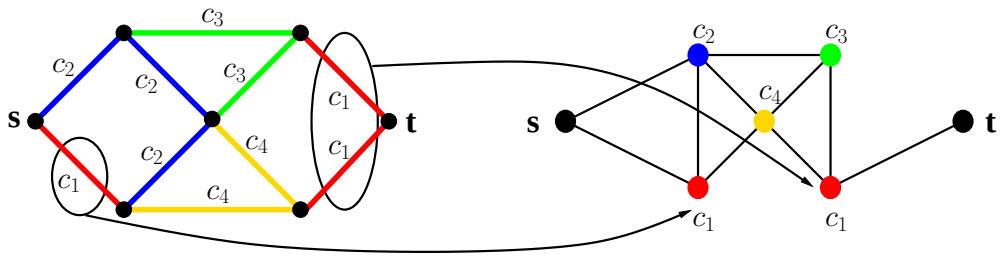


Fig. 4.12 – Construction du graphe dont les sommets sont les composantes des couleurs d'un graphe coloré.

Pour les mêmes raisons qu'en section 4.2.3.1, c'est à dire que chaque sommet de H représente un ensemble connexe de sommets et que deux sommets de H sont adjacents s'ils représentent des composantes partageant au moins un sommet, il y a bien correspondance entre les *st*-coupes colorées de G et les *st*-coupes en nombre de sommets de H, de même pour les deux autres problèmes.

Nous pouvons maintenant décrire la méthode de résolution. Nous effectuons un calcul dans H pour chaque sous-ensemble $\mathcal{C}'$ de couleurs de span strictement supérieur à 1. Pour cela nous attribuons un coût à chaque sommet, il est nul pour les sommets associés aux couleurs de $\mathcal{C}'$ et vaut 1 pour les autres sommets. Dans le graphe H muni de ces coûts, l'analogue classique du problème coloré étudié est résolu, le coût de la solution ne dépend pas des arêtes utilisées mais des sommets traversés. Au coût de la solution trouvée dans H, il faut ajouter $|\mathcal{C}'|$ pour obtenir une borne supérieure du coût en nombre de couleur de cette solution et de la solution optimale. C'est ce coût que nous utilisons dans la suite. Plusieurs cas sont à étudier.

Supposons qu'il existe une couleur c de span strictement supérieur à 1 qui n'appartient pas à $|\mathcal{C}'|$ et dont au moins deux sommets correspondant sont utilisés par la solution calculée pour $\mathcal{C}'$. Alors la solution trouvée pour l'ensemble $\mathcal{C}' \cup \{c\}$ est nécessairement de coût inférieur ou égal au coût de la solution obtenue pour $\mathcal{C}'$. La solution associée à $\mathcal{C}'$ est en effet moins coûteuse avec les coûts correspondant à l'ensemble de couleurs $\mathcal{C}' \cup \{c\}$, ce qui fait au moins une solution moins coûteuse pour cet ensemble.

Supposons maintenant qu'un seul des sommets correspondant à une couleur de span strictement supérieur à 1 qui n'appartient pas à $\mathcal{C}'$ soit utilisé dans la solution associée à $\mathcal{C}'$. Il est encore possible de trouver un ensemble de couleur, $\mathcal{C}' \cup \{c\}$, dont la solution associée est de coût inférieur. Sur la figure 4.13 avec $\mathcal{C}' = \{c_5\}$ le *st*-chemin utilisant les couleurs c_2, c_3 et c_4 est de longueur 5 si seuls les sommets de couleur c_5 ont un coût nul, alors que celui utilisant les couleurs c_1, c_6 et c_7 est de coût 6. Par contre en considérant l'ensemble de couleurs $\mathcal{C}' = \{c_5, c_1\}$, le chemin utilisant c_1, c_6 et c_7 est moins coûteux que l'autre.

Enfin, supposons qu'une couleur c appartienne à $\mathcal{C}'$ mais ne soit pas utilisée, alors la solution associée à l'ensemble $\mathcal{C}' - \{c\}$ est de coût inférieur ou égal à celle obtenue pour $\mathcal{C}'$.

Soit $\mathcal{C}^*$ l'ensemble de couleurs pour lequel la solution de coût minimum est obtenue. Alors

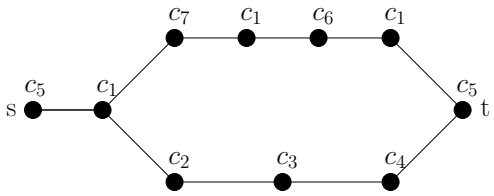


FIG. 4.13 – Le st-chemin coloré minimum est obtenu pour une répartition des coûts particuliére

d'après ce qui précède, on peut supposer que toutes les couleurs de $\mathcal{C}^*$ sont utilisées et qu'aucune couleur de span strictement supérieur à 1 n'appartenant pas à $\mathcal{C}^*$ n'est utilisée dans la solution. Ainsi toutes les couleurs utilisées ne sont comptées qu'une seule fois dans le coût de la solution, les couleurs de span strictement supérieur à 1 peuvent être utilisées autant de fois que nécessaire sans être comptées plusieurs fois et aucune couleur n'est comptée sans être utilisée. Cette solution obtenue dans H correspond à l'ensemble de couleur $\mathcal{C}_0^*$ qui est une solution au problème coloré dans G par construction de H.

Soit $\mathcal{C}_0^{opt}$ une solution optimale du problème coloré dans G dont l'ensemble de couleur de span strictement supérieur à 1 est $\mathcal{C}^{opt}$. La résolution du problème classique dans H dont les sommets sont de coût nul s'ils correspondent à une couleur de $\mathcal{C}^{opt}$ et de coût 1 sinon, donne une solution de coût $|\mathcal{C}_0^{opt}|$. En effet on sait qu'il existe une solution réalisable qui n'utilise que les sommets associés aux couleurs de $|\mathcal{C}_0^{opt}|$, et s'il en existe une de coût inférieur, alors on pourrait en déduire une solution colorée dans G qui serait aussi de coût inférieur à l'optimal, ce qui est impossible.

Ainsi on dispose d'une solution optimale dans G qui induit une solution de coût $|\mathcal{C}_0^{opt}|$ dans H et d'une solution de coût minimum dans H égale à $|\mathcal{C}_0^*|$. Si $|\mathcal{C}_0^{opt}| > |\mathcal{C}_0^*|$ l'optimalité de $\mathcal{C}_0^{opt}$ dans G est contredite et si $|\mathcal{C}_0^{opt}| < |\mathcal{C}_0^*|$ c'est l'optimalité de $\mathcal{C}_0^*$ dans H qui est contredite. Par conséquent $\mathcal{C}_0^*$ est une solution optimale dans G, et elle a été obtenue en temps polynomial.

Proposition 4.7 *Lorsque le nombre de couleur de span strictement supérieur à 1 dans un graphe coloré est borné par une constante $k \in \mathbb{N}$, les problèmes* MC-st-PATH, MC-st-CUT *et* MC-CUT *sont polynomiaux.*

Cette proposition peut s'étendre au cas où le nombre de couleur de span supérieur à 1 est borné par $p \log |V|$ pour une constante $p > 0$ puisque qu'alors le nombre de sous-ensembles de ces couleurs est de l'ordre de $|V|^p = 2^{p \log |V|}$. Le nombre de calculs à effectuer dans H est alors polynomial en $|V|$.

L'hypothèse que le nombre de couleur de span strictement supérieur à 1 dans un réseau est borné est fondée. En effet, dans les articles traitant de SRRG les simulations sont toujours effectuées sur les mêmes réseaux connus [DS04a], dont la figure 4.14 donne un exemple [TR04b]. Les SRRG sont représentés par des couleurs, les arêtes $\{1, 6\}$ et $\{2, 6\}$ par exemple appartiennent au même groupe de risque. Les groupes ne contenant qu'une seule arête ne sont pas indiqués. On constate que dans ce réseau multicoloré très peu de SRRG (deux en fait) donnerons des couleurs de span supérieur à 1 dans le graphe coloré. Il en est de même pour les autres réseaux couramment utilisés (COST239, NJLATA [CSC02, DS04a], NSFNET [OSYZ95, Jau, YDA00]) dont certains sont présentés en annexe C.

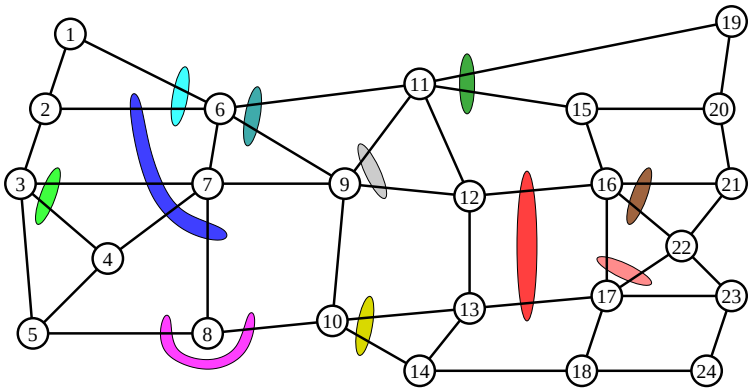


FIG. 4.14 – Exemple de réseau utilisé dans la littérature [TR04b, SYR05]

4.2.4 Span Borné

Après avoir étudié les cas particuliers pour lesquels certains problèmes colorés sont polynomiaux, nous nous intéressons à la complexité de ces problèmes lorsque le span maximum des couleurs du graphe coloré est borné par une constante.

4.2.4.1 Trouver deux chemins couleur-disjoints

Trois preuves de complexité ont été présentées dans [EBR+03], [Hu03] et [YJ04] pour le cas général. Une simple variation de ces preuves permet de montrer que trouver deux chemins couleur-disjoints dans un graphe coloré de span maximum 2 est également NP-Difficile. Nous adaptons la réduction de [Hu03] du problème SET SPLITTING NP-Difficile au problème de trouver deux chemins couleur disjoints dans ce cas particulier.

Théorème 4.1 *Le problème* SET SPLITTING *se réduit à trouver deux chemins couleur-disjoints entre deux sommets d'un graphe coloré de span maximum 2.*

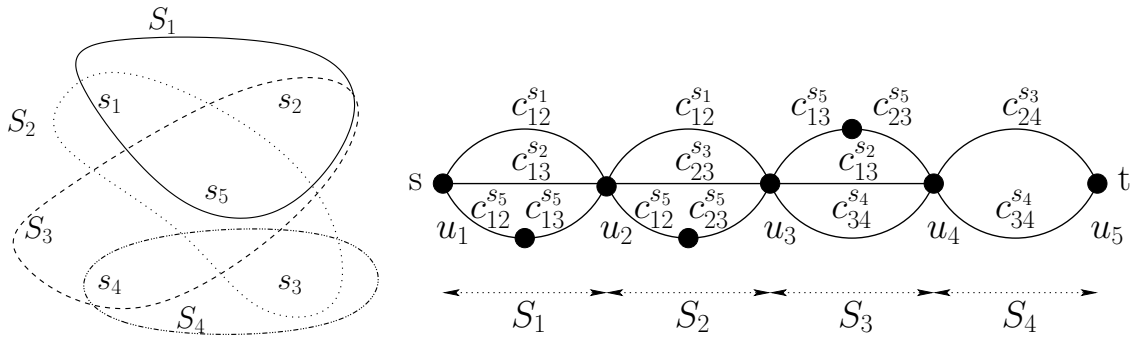
Preuve: Soit $\{S_i | i \in 1, \ldots, N\}$ une collection de sous-ensembles d'un ensemble fini S. Le problème SET SPLITTING consiste à trouver deux sous-ensembles de S disjoints, tels que chacun ait une intersection non vide avec tous les sous-ensembles de la collection $\{S_i | i \in 1, \ldots, N\}$ (figure 4.15(a)).

Un graphe coloré $G = (V, E)$ est construit de la manière suivante à partir d'une instance de SET SPLITTING ainsi définie. V comporte $N + 1$ sommets $u_1, \ldots u_{N+1}$. A chaque triplet (s, S_i, S_j) tel que $s \in S_i \cap S_j$ nous associons une couleur c_{ij}^s. Pour tout $i \leq N$ les sommets u_i et u_{i+1} sont connectés par $|S_i|$ chemins parallèles, chacun correspond à un élément $s \in S_i$. Le chemin associé à s entre u_i et u_{i+1} comporte une arête de chaque couleur c_{ij}^s pour chaque j tel que $s \in S_i \cap S_j$.

Dans ce graphe les couleurs sont de span au plus deux, et deux chemins couleur-disjoints entre u_1 et u_{N+1} induisent deux sous ensembles de S chacun contenant au moins un élément de $S_i, \forall i \in 1, \ldots, N$ et vice versa.

∎

Corollaire 4.1 *Trouver deux chemins couleur-disjoints ou deux chemins partageant un nombre minimum de couleurs dans un graphe coloré de span maximum 2 est NP-Complet.*



(a) Set Splitting (b) 2 Chemins Couleur-Disjoints

Fig. 4.15 – Le problème de Set Splitting se réduit à trouver deux chemins couleur-disjoints dans un graphe coloré de span maximum 2.

Le problème de décision associé à 2-MOP contient le problème 2-CDP, c'est ce qui nous permet de conclure à sa NP-Complétude comme dans [Hu03].

4.2.4.2 Nombre maximum de chemins couleur-disjoints

Ce problème a déjà été montré NP-difficile dans [JRN04]. L'équivalent du problème classique de trouver p chemins arête disjoints consiste à trouver p chemins couleur-disjoints. Dans le cas de chemins connectant deux sommets (st-chemins), le problème classique peut être aisément résolu (Théorème de Menger). Cependant la version colorée du problème est beaucoup plus difficile.

Théorème 4.2 *Trouver un nombre maximum de chemins couleur-disjoints dans un graphe dont les couleurs sont de span maximum 2 n'est pas approximable à un facteur* $o(|V|^{\frac{1}{4}-\varepsilon})$ *quel que soit* $\varepsilon > 0$ *sauf si* $P = NP$.

Preuve: Cette preuve repose sur l'inapproximabilité du problème Maximum Independant Set (annexe A page 134). A partir d'un graphe $H = (V_H, E_H)$, nous construisons un graphe coloré $G = (V, E, \mathcal{C})$ dans lequel chaque arête de H est associée à une couleur. Entre deux sommets s et t un chemin par sommet v de G est créé dont la longueur est le degré de v dans H comme illustré à la Figure 4.16. Le chemin associé à v dans le graphe coloré est donc assez long pour contenir une arête de chacune des couleurs représentant les arêtes incidentes à v dans H. Chaque arête de H n'est incidente qu'à deux sommets, par conséquent dans G les couleurs sont toutes de span au plus deux.

Trivialement, deux st-chemins sont couleur disjoints dans le graphe coloré G si et seulement si les deux sommets correspondant ne sont pas adjacents dans H. Le nombre maximum de chemins couleur-disjoints est donc égal à la taille maximum d'un ensemble indépendant dans H. De plus dans le graphe construit il y a au plus $|V| = |V_H|^2$ sommets. ∎

4.2.4.3 MC-st-Cut et MC-st-Path

Lorsque le span maximum du graphe coloré est borné par 2, la réduction de Minimum Set Cover à MC-st-Path présentée dans [YVJ05] est inadéquate pour montrer la NP-difficulté de MC-st-Path. En effet la restriction sur le span maximum imposerait que la taille des sous-ensembles disponibles pour couvrir les éléments de l'instance de Minimum Set Cover soit également bornée

(a) Maximum Independant Set (b) Nombre maximum de chemins couleur-disjoints

Fig. 4.16 – Le problème Maximum Independant Set se réduit à trouver un nombre maximum de chemins couleur-disjoints.

par 2. Or dans ce cas, Minimum Set Cover peut être résolu en temps polynomial par des techniques de couplage.

Nous donnons donc une réduction de Maximum 3 Satisfiability qui prouve non seulement la NP-difficulté de MC-*st*-Path et MC-*st*-Cut lorsque le span maximum est borné par 2, mais aussi un facteur d'inapproximabilité.

Théorème 4.3 *Il existe une réduction du problème* Maximum 3 Satisfiability *aux problèmes* Minimum Color *st*-Path *et* Minimum Color *st*-Cut *préservant l'inapproximabilité lorsque les couleurs sont de span au plus deux et contiennent au plus deux arêtes.*

Preuve: Considérons une instance de Maximum 3 Satisfiability comportant n variables x_i $1 \leq i \leq n$ et m clauses C_j $1 \leq j \leq m$ chacune étant une disjonction de trois littéraux. Le problème Maximum 3 Satisfiability consiste à affecter des valeurs binaires aux variables en sorte qu'un nombre maximum de clauses soient satisfaites par cette affectation.

A une telle instance nous associons un graphe coloré G de la manière suivante. G contient $n+m+1$ sommets u_i $1 \leq i \leq n+m+1$. Deux couleurs $V_{i,j}$ (V comme vrai) et $F_{i,j}$ (F comme faux) sont associées à chaque couple (i,j) tel que x_i ou $\overline{x_i}$ apparaît dans la clause C_j. Soit n_i le nombre de clauses contenant la variable x_i, c'est à dire le littéral x_i ou $\overline{x_i}$. Il y a donc $6m = 2\sum_{i=1,\dots n} n_i$ couleurs.

Pour tout $i \in \{1,\dots,n\}$, deux chemins de longueur n_i connectent u_i et u_{i+1}, le premier contient une arête par couleur $T_{i,j}$ et le second une arête par couleur $F_{i,j}$, avec j tel que x_i ou $\overline{x_i} \in C_j$.

Pour tout $j \in \{1,\dots,m\}$, u_{n+j} et u_{n+j+1} sont connectés par trois arêtes parallèles, chacune correspondant à un littéral y présent dans la clause C_j. La couleur de l'arête associée à y est soit $F_{i,j}$ ou $T_{i,j}$ suivant si $y = \overline{x_i}$ ou $y = x_i$ pour un certain $i \in \{1 \dots n\}$ (Figure 4.17). Par conséquent, G contient au plus deux arêtes de chaque couleur, et les couleurs sont donc de span au plus deux.

Dans le graphe construit ainsi, l'instance de Minimum Color *st*-Path intéressante est de trouver un chemin entre u_1 et u_{n+m+1} utilisant un nombre minimum de couleurs. Premièrement, nous montrons que chaque affectation des variables de l'instance de Maximum 3 Satisfiability satisfaisant $\mu \leq m$ clauses correspond à un chemin coloré utilisant $4m - \mu$ couleurs entre u_1 et u_{n+m+1} dans le graphe G.

Considérons une affectation des variables et déduisons un chemin coloré entre u_1 et u_{n+m+1} de la façon suivante. Si la variable x_i a la valeur *vrai* (resp. *faux*), le u_1, u_{n+m+1}-chemin est composé entre u_i et u_{i+1}, $i \leq n$, du sous chemin utilisant les couleurs $V_{i,j}$ (resp. $F_{i,j}$).

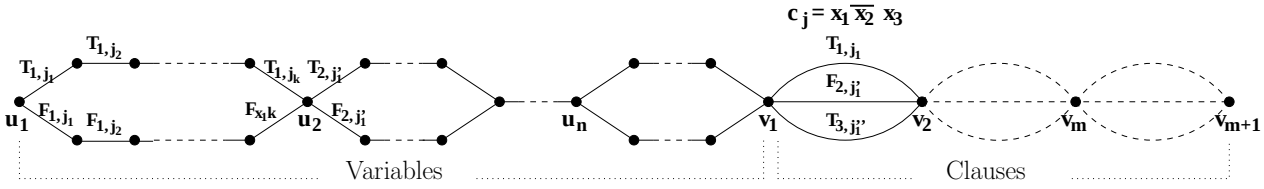


FIG. 4.17 – Réduction de MAXIMUM 3 SATISFIABILITY à MC-*st*-PATH dans le cas de couleurs de span au plus 2.

Si la clause C_j est satisfaite par l'affectation, le u_1, u_{n+m+1}-chemin emprunte entre u_{n+j} et u_{n+j+1} l'arête associée à l'une des variables permettant de satisfaire la clause. C'est-à-dire que si le littéral x_i est présent dans C_j et que la valeur *vrai* est affectée à la variable x_i, le chemin utilise l'arête de couleur $V_{i,j}$, et si la clause C_j contient le littéral $\overline{x_i}$ et que la valeur *faux* est affectée à la variable x_i, le chemin utilise l'arête de couleur $F_{i,j}$.

Si la clause C_j n'est pas satisfaite par l'affectation, alors le u_1, u_{n+m+1}-chemin peut utiliser n'importe laquelle des trois arêtes connectant u_{n+j} et u_{n+j+1}.

Le chemin ainsi décrit utilise $3m+(m-\mu)$ couleurs. Plus précisément, $\sum_{i=1,\dots n} n_i = 3m$ couleurs sont utilisées entre u_1 et u_{n+1} et pour chacune des μ clauses satisfaites la couleur utilisée fait partie des $3m$ couleurs précédentes. En revanche pour chacune des $m-\mu$ clauses non satisfaites une couleur supplémentaire est nécessaire.

Nous montrons maintenant qu'un chemin utilisant $3m+(m-\mu)$ couleurs dans G représente une affectation des variables de l'instance de MAXIMUM 3 SATISFIABILITY telle que μ clauses sont satisfaites.

Considérons un u_1, u_{n+m+1}-chemin. Entre u_i et u_{i+1} il utilise soit les arêtes de couleur $V_{i,j}$, et dans ce cas la valeur *vrai* est affectée à la variable x_i, soit les arêtes de couleur $F_{i,j}$, et la valeur affectée à x_i est alors *faux*.

Supposons qu'entre u_{n+j} et u_{n+j+1} le chemin utilise l'une des $3m$ couleurs déjà utilisées entre u_1 et u_{n+1}, sans perte de généralité cette couleur est $V_{i,j}$, ce qui implique que la variable x_i a la valeur *vrai*. Alors par construction du graphe coloré G, le littéral x_i appartient à la clause C_j, sinon la couleur $V_{i,j}$ ne serait pas disponible entre u_{n+j} et u_{n+j+1}. Comme la variable x_i a la valeur *vrai*, la clause C_j est satisfaite.

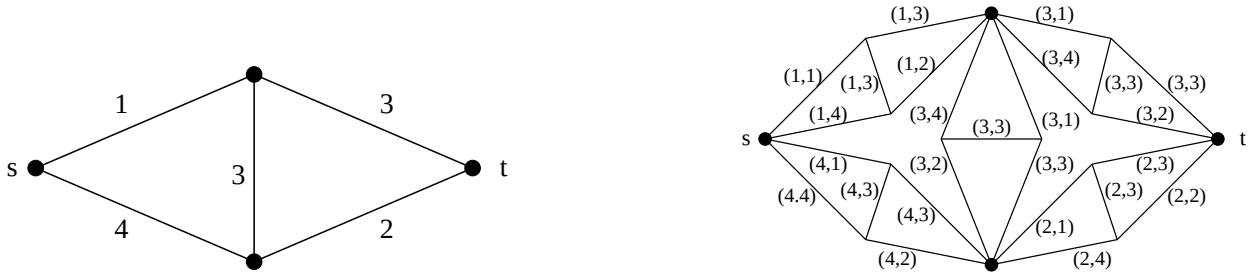
Étant donné que le chemin utilise $3m+(m-\mu)$ couleurs, μ couleurs parmi les $3m$ utilisées entre u_1 et u_{n+1} sont utilisées entre u_{n+1} et u_{n+m+1} et ainsi μ clauses sont satisfaites.

Pour le problème MINIMUM COLOR *st*-CUT, il suffit de remarquer que le graphe coloré construit est bien un graphe série-parallèle comme indiqué en section 4.2.1.3 (page 66). ∎

Corollaire 4.2 *Les problèmes* MINIMUM COLOR *st*-PATH *et* MINIMUM COLOR *st*-CUT *sont NP-difficiles lorsque les couleurs sont de span au plus deux et contiennent au plus deux arêtes, et il existe $\varepsilon > 0$ tel qu'il est NP-difficile d'approximer ces problèmes à un facteur $1+\varepsilon$ près.*

Une construction semblable à celle utilisée dans une preuve d'inapproximabilité du problème Maximum Clique [Hoc97] permet d'amplifier le facteur d'inapproximabilité donné dans le théorème 4.2 pour MC-*st*-PATH et MC-*st*-CUT jusqu'à la valeur énoncée dans le théorème suivant.

Théorème 4.4 *Soit une constante $k \in \mathbb{N}$ donnée. Il existe une constante $\varepsilon > 0$ indépendante de k telle que* MC-*st*-PATH *et* MC-*st*-CUT *ne sont pas approximables à un facteur k^{ε} près lorsque le span des couleurs est borné par k, sauf si $P = NP$.*



(a) G (b) $G^2_{a,b}$

FIG. 4.18 – Un graphe coloré G et son carré G^2_{st}.

Etant donné un graphe coloré $G = (V, E, \mathcal{C})$ et deux sommets $s, t \in V$, la construction susmentionnée consiste en un *produit de graphes colorés* G^2_{st} (Fig. 4.18). Tout d'abord, nous remplaçons chaque arête (u, v) de G par une copie G_{uv} du graphe G en identifiant les sommets u et v aux sommets s et t de la copie. G^2_{st} possède donc $|E|^2$ arêtes. Ensuite nous définissons la couleur d'une arête (x, y) appartenant à une copie G_{uv} par le couple (c_{uv}, c_{xy}) où c_{xy} est la couleur de l'arête (x, y) dans G. Ainsi l'ensemble couleurs de G^2_{st} est $\mathcal{C} \times \mathcal{C}$

Remarquons que deux arêtes de deux copies différentes remplaçant des arêtes de couleurs distinctes dans G sont de couleurs différentes dans G^2_{st} même si elles sont de même couleur dans G.

Le graphe G^p_{st} est construit récursivement pour tout $p \in \mathbb{N}$ en remplaçant comme ci-dessus chaque arête de G^{p-1}_{st} par une copie de G.

Deux relations clés entre les solutions de MC-*st*-PATH ou MC-*st*-CUT dans G^p_{st} et G sont données par les lemmes 4.1 et 4.2. Ces lemmes sont basés sur la décomposition d'un chemin de G^p_{st} en chemins de G.

Lemme 4.1 *Soit $\mathcal{A}$ un algorithme calculant des chemins colorés et $\mathcal{A}(G^p_{st})$ un chemin coloré de G^p_{st} retourné par $\mathcal{A}$, alors $\forall p \in \mathbb{N}$ un chemin coloré P_G de G de coût inférieur ou égal à $|\mathcal{A}(G^p_{st})|^{\frac{1}{p}}$ peut être déduit de $\mathcal{A}(G^p_{st})$.*

Preuve: La preuve est détaillée pour le cas $p = 2$ mais la décomposition de chemin utilisée se généralise à toute constante $p \in \mathbb{N}$. Il y a alors p niveaux de chemins au lieu des deux seuls suivants. Considérons le chemin $\mathcal{A}(G^2_{st})$ retourné par l'algorithme $\mathcal{A}$ dans le graphe G^2_{st}. Ce chemin peut se décomposer en un chemin *externe* P et plusieurs chemins *internes* P_i.

Le chemin externe P est la projection dans G du chemin $\mathcal{A}(G^2_{st})$, c'est à dire qu'il faut considérer les copies de G de la construction de G^2_{st} comme les arêtes qu'elles représentent dans G. P est alors le chemin de G utilisant les couleurs de ces arêtes. Soit $|C_P|$ le nombre de couleurs utilisées par P dans G.

Un chemin interne P_i est le chemin induit par le chemin $\mathcal{A}(G^2_{st})$ sur une copie de G remplaçant une arête i de G au cours de la construction de G^2_{st}. Soit $|C_{P_0}|$ le nombre minimum sur tous les chemins internes de couleurs utilisées par un chemin interne.

Pour conclure la preuve, remarquons que $|\mathcal{A}(G^2_{st})| \geq |C_P||C_{P_0}| \geq [\min\{|C_P|, |C_{P_0}|\}]^2$. ∎

Lemme 4.2 *Soit G un graphe coloré, s et t deux sommets de G et $p \in \mathbb{N}$ une constante. Les solutions optimales au problème* MINIMUM COLOR *st*-PATH *entre s et t dans G et dans G^p_{st} vérifient* $|\mathrm{OPT}(\mathrm{G^p_{st}})| = |\mathrm{OPT}(\mathrm{G})|^{\mathrm{p}}$.

Preuve: Comme pour le lemme 4.1, la preuve est détaillée pour $p = 2$ mais se généralise facilement pour tout $p \in \mathbb{N}$.

Considérons une solution optimale P au problème MC-*st*-Path dans G utilisant l'ensemble de couleurs C_{opt} et remarquons que le produit cartésien $C_{opt} \times C_{opt}$ est chemin coloré dans G^2_{st}. En effet si le chemin P connecte s et t uniquement grâce aux couleurs de C_{opt} dans G, le chemin obtenu dans G^2_{st} en remplaçant chaque arête de P dans G par le chemin P lui-même est bien un chemin connectant s et t dans G^2_{st} qui n'utilise que des couleurs de $C_{opt} \times C_{opt}$. Par conséquent $\mathrm{OPT}(\mathrm{G}^2_{\mathrm{st}}) \leq \mathrm{OPT}(\mathrm{G})^2$.

D'après le lemme 4.1, si une solution optimale dans G^2_{st} utilise $\mathrm{OPT}(\mathrm{G}^2_{\mathrm{st}})$ couleurs, alors une solution dans G peut en être déduite qui utilise moins de $[\mathrm{OPT}(\mathrm{G}^2_{\mathrm{st}})]^{\frac{1}{2}}$ couleurs. Ainsi $\mathrm{OPT}(\mathrm{G}) \leq [\mathrm{OPT}(\mathrm{G}^2_{\mathrm{st}})]^{\frac{1}{2}}$. ∎

Pour prouver le théorème 4.4, supposons que $\mathcal{A}$ est un algorithme de facteur d'approximation $(1+\varepsilon)^p$ et utilisons le pour trouver un chemin dans G^{p+1}_{ab}. Les lemmes 4.1 et 4.2 mènent à une contradiction avec le théorème 4.2.

Preuve: [Théorème 4.4] A partir d'une instance de MC-*st*-Path dans un graphe G tel que les couleurs sont de span au plus 2 et contiennent chacune au plus deux arêtes, nous construisons l'instance associée dans G^p_{st} pour une constante $p \in \mathbb{N}$. Dans G^p_{st}, les couleurs sont au plus de span 2^p car au plus deux arêtes sont de la même couleur dans G, et par conséquent le nombre d'arêtes portant une même couleur est multiplié par deux en passant de G^{p-1}_{st} à G^p_{st}.

D'après les lemmes 4.2 et 4.1, il existe une constante ε_1 telle qu'aucun algorithme ne peut approcher les solutions optimales de MC-*st*-Path dans G^p_{st} à un facteur $(1+\varepsilon_1)^p$ près en temps polynomial sans contredire le théorème 4.2.

Comme ce facteur d'inapproximabilité est prouvé pour les graphes de span maximum 2^p pour $p \in \mathbb{N}$, il reste vrai pour l'ensemble des graphes colorés de span bornés par une constante $k \in \mathbb{N}$

Nous pouvons écrire $k = 2^{\log_2 k}$. Ainsi pour $p = \lceil \log_2 k \rceil$, MC-*st*-Path n'est pas approximable à un facteur $(1+\varepsilon_1)^{\log_2 k} \leq (1+\varepsilon_1)^{\lceil \log_2 k \rceil}$. Soit $\varepsilon = (\log_2 k)(\ln(1+\varepsilon_1))/\ln k = \log_2(1+\varepsilon_1)$, qui est indépendant de k. Ce qui précède implique $k^\varepsilon = (1+\varepsilon_1)^{\log_2 k}$. ∎

Bien que l'écart entre les facteurs d'inapproximabilité et d'approximabilité pour MC-*st*-Path et MC-*st*-Cut reste important, l'algorithme d'approximation trivial suivant prouve que Minimum Color *st*-Path, Minimum Color *st*-Cut et Minimum Color Cut ne sont peut-être pas aussi difficiles à approximer que le problème de trouver un nombre maximum de chemins couleur-disjoints entre deux sommets d'un graphe coloré.

Proposition 4.8 *Lorsque le span maximum du graphe est borné par une constante $k \in \mathbb{N}$,* MC-*st*-Path, MC-*st*-Cut *et* MC-Cut *sont approximables à un facteur k près.*

Preuve: Nous rappelons la transformation utilisée en section 4.2.3.3 (page 70) qui permet d'obtenir en temps polynomial des solutions optimales pour les problèmes MC-Cut, MC-*st*-Path et MC-*st*-Cut.

Soit $G = (V, E, \mathcal{C})$ un graphe coloré, alors nous construisons le graphe $H = (V_H, E_H)$ comme suit. Chaque sommet de H représente une composante d'une couleur de $\mathcal{C}$. Deux sommets de V_H sont reliés par une arête de E_H si les deux composantes associées possèdent un sommet en commun dans G. Deux sommets s et t sont ajoutés à V_H et une arête de E_H connecte s (resp. t) à un sommet v de V_H si s (resp. t) appartient à la composante représentée par v.

Maintenant supposons que Γ est la valeur d'une *st*-coupe minimum en nombre de sommets dans H. Comme il n'y a pas plus de k sommets dans H correspondant à une même couleur de G, et comme au moins Γ composantes sont coupées par une *st*-coupe colorée minimum dans G, le

nombre de couleurs d'une *st*-coupe colorée minimum appartient à $[\Gamma, k\Gamma]$. Le même raisonnement s'applique au problème MINIMUM COLOR *st*-PATH.

Ainsi calculer une *st*-coupe minimum en nombre de sommets dans H ou un *st*-chemin de nombre de sommets minimum fournit une k-approximation des problèmes MINIMUM COLOR *st*-CUT et MINIMUM COLOR *st*-PATH.

Pour le problème MINIMUM COLOR CUT, nous construisons un hypergraphe dont l'ensemble de sommets est V et qui contient une hyperarête par composante de chaque couleur comprenant les sommets appartenant à cette composante. Chaque couleur produit donc au plus k hyperarêtes dans l'hypergraphe et le même raisonnement que pour MC-*st*-CUT et MC-*st*-PATH permet de conclure qu'une coupe minimum dans l'hypergraphe offre une k-approximation du problème MINIMUM COLOR CUT. ∎

4.2.5 Span quelconque

Lorsque le span des couleurs est quelconque les informations exploitables sur le graphe coloré sont réduites et les méthodes que nous avons utilisées dans les sections précédentes (énumération, bornes etc) ne sont plus applicables. Il s'en suit que des problèmes qui étaient polynomiaux ou approximables à des facteurs raisonnables pour des cas particuliers comme les problèmes MINIMUM COLOR *st*-PATH et MINIMUM COLOR *st*-CUT deviennent inapproximables à des facteurs très élevés. Par contre nous verrons que le niveau de complexité du problème MINIMUM COLOR SPANNING TREE est constant quelles que soient les hypothèses faites sur le graphe.

4.2.5.1 MC-*st*-PATH et MC-*st*-CUT

Dans le cas des graphes colorés de span quelconque, un facteur d'inapproximabilité égal à $2^{\log^{1-\delta}|C|^{\frac{1}{4}}}$ avec $\delta = (\log\log|C|^{\frac{1}{4}})^{-\varepsilon}$ pour tout $\varepsilon < 1/2$ peut être démontré par une réduction du problème RED BLUE SET COVER au problème MINIMUM COLOR *st*-PATH préservant l'inapproximabilité[CDKM00, Wir01]. Dans la suite, nous améliorons ce résultat grâce à une réduction au problème MINIMUM LABEL COVER (annexe A page 135). Cette réduction permet de contrôler plus finement les dimensions des instances construites et ainsi d'augmenter l'ordre de grandeur du facteur d'inapproximabilité.

Théorème 4.5 *Les problèmes* MINIMUM COLOR *st*-PATH *et* MINIMUM COLOR *st*-CUT *sont difficiles à approximer à un facteur* $2^{\log^{1-\delta}|C|^{\frac{1}{2}}}$ *avec* $\delta = (\log\log|C|^{\frac{1}{2}})^{-\varepsilon}$ *pour tout* $\varepsilon < 1/2$ *sauf si* $P = NP$.

Preuve: Nous considérons une instance de MINIMUM LABEL COVER dans le graphe biparti $B = (U, V, E)$ avec les ensembles de labels L_U et L_V et la relation Π_{uv} pour chaque arête $\{u, v\} \in E$, avec $u \in U$ et $v \in V$. A partir de cette instance nous construisons un graphe coloré $G = (V_\mathcal{C}, E_\mathcal{C}, \mathcal{C})$ et une instance de MINIMUM COLOR *st*-PATH dans ce graphe de la manière suivante :

1. A chaque couple $(l, u) \in L_U \times U$ nous associons une couleur c_{lu}, ainsi $|\mathcal{C}| = |L_U||U|$.
2. Pour chaque sommet $v_i \in V$ un sommet x_i est créé dans $V_\mathcal{C}$.
3. Un sommet supplémentaire $x_{|V|+1}$ est ajouté à $V_\mathcal{C}$.
4. A chaque label $l_v \in L_V$ éligible pour le sommet $v_i \in V$ est associé un chemin connectant x_i et x_{i+1}. Ce chemin comporte une arête par voisin u de v_i. Cette arête est éventuellement multiple car elle doit comporter une arête de couleur c_{lu} par label $l \in L_U$ formant une paire admissible $(l, l_v) \in \Pi_{uv}$ pour l'arête $\{u, v\}$. Le nombre d'arêtes en parallèle est égal au nombre

de labels $l \in L_U$ formant une paire admissible $(l, l_v) \in \Pi_{uv}$ pour l'arête $\{u, v\}$, c'est à dire au moins un, mais éventuellement $|L_U|$.

Dans ce graphe, un $x_1x_{|V|+1}$-chemin coloré utilisant μ couleurs correspond à un labeling couvrant les arêtes de E de coût μ. En effet, entre x_i et x_{i+1} le chemin coloré utilise un sous chemin, il est associé à un label $l_v \in L_V$ éligible pour le sommet v_i. Ce label est affecté à v_i, $f_V(v_i) = \{l_v\}$. De plus, ce sous chemin comporte nécessairement une arête par voisin u de v_i de couleur $c_{l_u u}$ pour un label $l \in L_U$ formant avec l_v une paire admissible $(l_u, l_v) \in \Pi_{uv_i}$. Affecter à un voisin u de v_i le label l_u permet donc de couvrir l'arête $\{u, v\} \in E$. En affectant ainsi les labels on obtient un labeling couvrant toutes les arêtes de E. Son coût est μ car pour chaque couleur du chemin correspondant à un sommet $u \in U$, exactement un label à été affecté au sommet u. En effet, même si une couleur contient plusieurs arêtes dans plusieurs sous chemins différents, elle représente toujours le même couple de $L_U \times U$ et le label correspondant n'est affecté qu'une seule fois au sommet u.

Inversement un labeling couvrant toutes les arêtes de E de coût μ induit un $x_1x_{|V|+1}$-chemin coloré utilisant μ couleurs. Soit l'ensemble de couleurs $C = \{c_{l_u} | l \in f_U(u) \subseteq L_U, u \in U\}$ contenant les couleurs associées aux paires label-sommet (l_u, u) tel que le label l_u a été affecté au sommet u par le labeling. Par définition, cet ensemble de couleur est de taille μ. De plus pour un sommet $v_i \in V$, toutes les arêtes de E incidentes à v_i sont couvertes par le labeling. $f_V(v_i)$ contient donc au moins un label l_v pour lequel il existe un label $l_u \in L_U$ appartenant à $f_U(u)$ pour tout sommet $u \in U$ voisin de v_i et tel que $(l_u, l_v) \in \Pi_{uv}$ est admissible pour l'arête $\{u, v\}$. Par définition de l'ensemble de couleurs C, la couleur associée à chacun des couples (l_u, u) précédents appartient à C. Par construction du graphe coloré G, il existe entre x_i et x_{i+1} un chemin utilisant exactement les couleurs c_{l_uu} associées aux couples précédents. L'ensemble de couleurs C permet donc de connecter x_1 et $x_{|V|+1}$, il s'agit donc d'un chemin coloré de taille μ.

D'après la preuve d'inapproximabilité pour ce problème donnée dans [DS04b], nous pouvons considérer uniquement des instances de Minimum Label Cover telles que $|L_U| = \mathcal{O}(2^{\log^{1-\delta}|V|}) \leq |V|^{\frac{1}{2}}$ (pour $|V|$ assez grand) et $|U| \leq |V|(\log\log|V|)^{\frac{1}{2}} \leq |V|^{\frac{3}{2}}$ pour lesquelles le facteur d'inapproximabilité en $2^{\log^{1-\delta}|V|}$ pour $\delta = (\log\log|V|)^{-\varepsilon}$ et $\varepsilon < \frac{1}{2}$ est prouvé. Comme le nombre de couleurs de l'instance de Minimum Color *st*-Path construite est égal à $|U||L_U|$ et que par hypothèse $|U||L_U| \leq |V|^2$, nous pouvons en déduire le théorème.

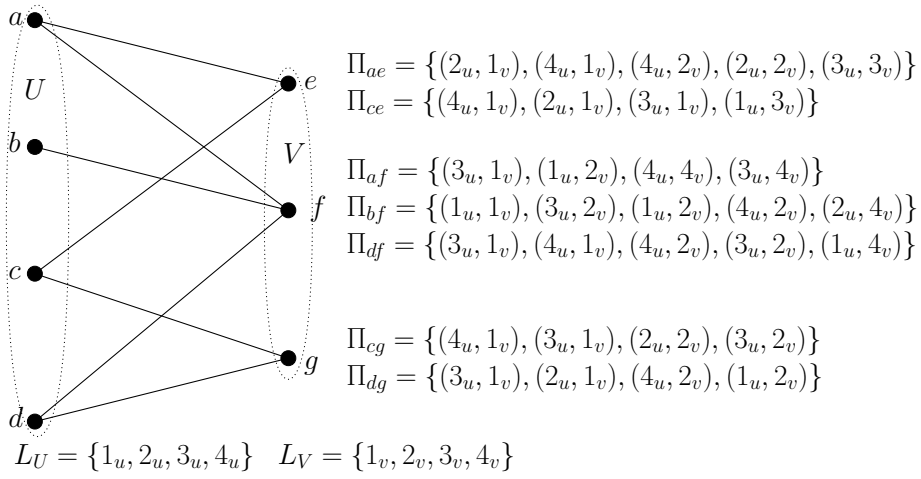
Pour le problème Minimum Color *st*-Cut, il suffit de remarquer que le graphe coloré construit est bien un graphe série-parallèle comme indiqué en section 4.2.1.3 (page 66) et que le facteur d'inapproximabilité est exprimé en fonction du nombre de couleurs et non du nombre de sommets. ∎

La figure 4.19(b) donne l'instance de Minimum Color *st*-Path construite à partir de l'instance de Minimum Label Cover de la figure 4.19(a) selon la réduction précédente. Le label 2 n'est pas éligible pour le sommet e, il appartient à la relation Π_{ae} mais pas à la relation Π_{ce}. Affecter le label 2 au sommet e empêcherait donc de couvrir l'arête $\{c, e\}$.

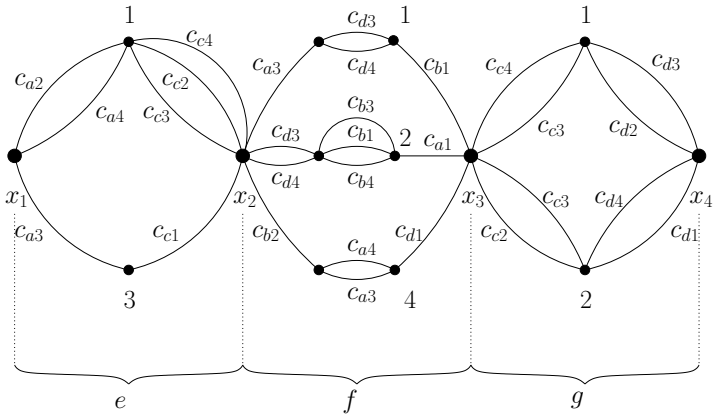
Enfin, notons que le problème MC-*st*-Path est un cas particulier du problème coloré consistant à trouver un arbre de Steiner (annexe A page 138) utilisant un nombre minimum de couleurs. Ce dernier est donc aussi difficile à approximer que le problème MC-*st*-Path.

4.2.5.2 Minimum Color Spanning Tree

Ce problème a été étudié dans [CL97, KW98, WCX02] sous le nom de *Minimum label spanning tree*. Sa complexité ne dépend pas du span des couleurs et est identique à celle du problème Minimum Set Cover avec lequel sa complexité et un résultat d'inapproximabilité sont montrés.



(a) Instance de Minimum Label Cover



(b) Instance de Minimum Color *st*-Path

Fig. 4.19 – Illustration de la réduction de Minimum Label Cover à Minimum Color *st*-Path.

Théorème 4.6 ([WCX02]) *Le problème* Minimum Color Spanning Tree *est NP-difficile même dans les graphes colorés de span maximum 1. De plus il n'est pas approximable à un facteur* $(1-\varepsilon)\ln(|V|-1)$ *pour tout* $\varepsilon > 0$ *à moins que* $NP \subseteq TIME(n^{\log\log n})$.

Preuve: Nous construisons un graphe coloré G à partir d'une instance de Minimum Set Cover. Chaque sous-ensemble s_i de l'instance de Minimum Set Cover est identifié à une couleur c_i du graphe coloré (figure 4.20).

Les sommets du graphe G sont l'union des éléments à couvrir $u_1, \ldots, u_n$ de l'instance de couverture et d'un sommet supplémentaire u_0.

Ensuite pour chaque sous-ensemble s_i, tous les sommets de $s_i \cup \{u_0\}$ forment une clique de couleur c_i dans G. Pour l'instance de Minimum Set Cover initiale, cela revient à dire qu'il existe un élément $u_0 \in U$ qui appartient à chaque $s_i \in S$. Cet ajout n'a aucune influence sur les solutions du problème de couverture, mais garantit que toute couverture de U donne un sous-ensemble de couleurs *connexe* dans le graphe G.

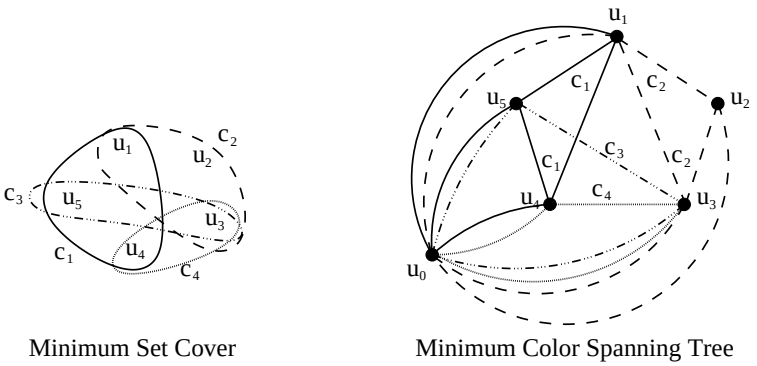


FIG. 4.20 – Exemple d'instance de MINIMUM COLOR SPANNING TREE construit à partir d'une instance de MINIMUM SET COVER.

Notons de plus que pour chaque couleur c_i le sous-graphe induit par ses arêtes est connexe puisque c'est une clique sur les sommets $s_i \cup \{u_0\}$.

Un arbre couvrant coloré $\mathcal{C}' = \{c_{i_1}, \ldots, c_{i_{|\mathcal{C}'|}}\} \subseteq \mathcal{C}$ du graphe ainsi construit induit une couverture des éléments de U, $S' = \{s_{i_1}, \ldots, s_{i_{|\mathcal{C}'|}}\}$, et vice versa grâce à la présence du sommet u_0.

Enfin, le nombre de sommets de l'instance d'arbre couvrant coloré est égal au nombre d'éléments de l'instance de MINIMUM SET COVER initiale plus un élément supplémentaire qui appartient à tous les ensembles s_i. Ainsi approximer à un facteur $(1-\varepsilon)\ln(|V|-1)$ près la taille minimum d'un arbre couvrant coloré dans le graphe construit revient à approximer à un facteur $(1-\varepsilon)\ln(|U|-1)$ près la solution optimale de l'instance de MINIMUM SET COVER, ce qui est difficile à moins que $NP \subseteq TIME(n^{\log\log n})$ [Fei98]. ∎

La ressemblance avec la MINIMUM SET COVER ne s'arrête pas là, l'algorithme glouton étudié dans [CL97, KW98, WCX02] est semblable à celui de [Sla96] et permet d'obtenir un facteur d'approximation similaire en $H(|V|-1)$ pour le problème MINIMUM COLOR SPANNING TREE dans le cas général pondéré.

L'algorithme part du graphe dépourvu de ses arêtes, i.e. comportant $|V|$ composantes connexes réduites à un sommet, et ajoute à chaque itération les arêtes de la couleur la "plus avantageuse", fusionnant ainsi plusieurs composantes connexes deux par deux. L'algorithme termine lorsque le graphe est connexe après au plus $\min\{|V|, |\mathcal{C}|\}$ itérations. Par construction, les couleurs ainsi choisies forment un arbre couvrant coloré.

On note w_c le poids de la couleur c et n_c^i le nombre de fusions entre deux composantes réalisées si la couleur c est choisie à l'itération i, i.e. la différence entre le nombre de composantes avant et après l'ajout des arêtes de couleur c. Le *coût réparti* de la couleur c est alors le poids de la couleur c réparti sur l'ensemble des fusions effectuées au cours de l'itération i si c est choisie, c'est à dire w_c/n_c^i. A une itération donnée, choisir la couleur de coût réparti minimum est donc avantageux, puisque c'est celle qui permet la fusion de composantes connexes à moindre frais.

Plus formellement l'algorithme glouton est le suivant :

Théorème 4.7 ([WCX02]) *L'algorithme glouton donne une approximation à un facteur $H(|V|-1)$ près du coût d'un arbre couvrant coloré de coût minimum, où $H(|V|-1) = 1 + \frac{1}{2} + \cdots + \frac{1}{|V|-1}$.*

Preuve: Au cours de l'algorithme, $|V|-1$ fusions doivent être effectuées. Sans perte de généralité, nous supposons que la couleur c_i est choisie à l'itération i, et qu'il y a t itérations en tout.

Algorithme 1 Approximation du problème MINIMUM COLOR SPANNING TREE

1: **Initialisation :** $E \leftarrow \emptyset$, $G \leftarrow (V, E)$, $\mathcal{C}' \leftarrow \emptyset$, $i = 0$
2: **tant que** G n'est pas connexe **faire**
3: **Calculer** n^i_c, et **Choisir la couleur** c_i de coût réparti $w_{c_i}/n^i_{c_i}$ minimum
4: $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{c_i\}$, $E \leftarrow E \cup c_i$, $i \leftarrow i + 1$
5: **fin tant que**
6: **retourner** $\mathcal{C}'$

Plaçons nous au début de l'itération i lors de laquelle la k^{eme} fusion entre composantes connexes a lieu et supposons que n_i fusions restent encore à effectuer. Alors $n_i \geq |V| - k$ puisqu'il reste exactement $(|V| - 1) - (k - 1) = |V| - k$ fusions juste avant d'effectuer la k^{eme} pour un coût $\alpha_k = w_{c_i}/n^i_{c_i}$.

L'ajout d'une partie des couleurs d'une solution optimale permettrait de réaliser les n_i fusions restantes pour un coût total inférieur à l'optimal OPT. Pour ne pas entrer en contradiction avec ce fait, le coût réparti de la couleur choisie c_i, i.e. le minimum des coûts répartis qui est aussi le coût de la k^{eme} fusion, doit vérifier $\frac{w_{c_i}}{n^i_{c_i}} \leq \frac{OPT}{n_i}$ et ainsi $\alpha_k = \frac{w_{c_i}}{n^i_{c_i}} \leq \frac{OPT}{|V|-k}$.

Enfin, puisque le coût de chaque couleur choisie par l'algorithme a été réparti sur les $|V| - 1$ fusions entre composantes connexes, le coût total de l'arbre couvrant trouvé est la somme des coûts des fusions : $\alpha_1 + \alpha_2 + \cdots + \alpha_{t-1} + \alpha_t \leq OPT(\frac{1}{|V|-1} + \frac{1}{|V|-2} + \cdots + \frac{1}{2} + 1) = H(|V| - 1)OPT$. ∎

4.2.6 Synthèse des complexités

La table 4.1 présente les résultats principaux sur la complexité et l'approximabilité des problèmes colorés.

4.3 Formulations en MILP

Dans cette section nous proposons des formulations en programmes linéaires en variables mixtes pour les problèmes MC-*st*-PATH, MC-*st*-CUT et MC-CUT.

4.3.0.1 Formulation en MILP pour MC-CUT et MC-*st*-CUT

La formulation suivante en MILP exprime MC-*st*-CUT comme l'affectation de *potentiels* aux sommets du graphe à travers les variables non négatives $x_u,\ u \in V$.

Le graphe est partitionné en sous ensembles de sommets disjoints selon la valeur des potentiels des sommets. Deux sommets u, v appartiennent au même sous ensemble si et seulement si leurs potentiels sont égaux $x_u = x_v$.

Si deux sommets adjacents u et v ont des potentiels distincts, $x_u \neq x_v$, alors la couleur c de l'arête les connectant doit appartenir à la coupe colorée. Pour ce faire la variable binaire y_c associée à la couleur c est forcée de prendre la valeur 1 (ligne 4.2).

Les lignes 4.3 et 4.5 assurent que les sommets s et t sont bien déconnectés par la coupe colorée puisque des potentiels différents leur sont affectés.

Min. Color $\cdots$		st-Path	$\left.\begin{matrix} st \\ \text{Multi} \end{matrix}\right\}$-Cut	Cut	2-Disjoint st-Paths	2-Min Overlap. st-Paths	Max. number of Disjoint st-Paths	Spanning Tree
général	complexité	NP-Difficile [YVJ05]	NP-Difficile	?	NP-Complet [Hu03]	NP-Difficile [YVJ05]	NP-Difficile [JRN04]	NP-Difficile [CL97]
	non approx	$2^{\log^{1-\delta}\|\mathcal{C}\|^{\frac{1}{2}}}$	$2^{\log^{1-\delta}\|\mathcal{C}\|^{\frac{1}{2}}}$	?	———	?	$\forall\varepsilon>0,\ \|V\|^{\frac{1}{4}-\varepsilon}$	$o(\log(\|V\|))$ [WCX02]
	approx	?	?	?	———	?	?	$O(\log(\|V\|))$ [WCX02]
span k	complexité	NP-Difficile	NP-Difficile	?	NP-Complet	NP-Difficile	NP-Difficile	NP-Difficile [CL97]
	non approx	$\exists\varepsilon>0,\ k^{\varepsilon}$	$\exists\varepsilon>0,\ k^{\varepsilon}$	?	———	?	$\forall\varepsilon>0,\ \|V\|^{\frac{1}{4}-\varepsilon}$	$o(\log(\|V\|))$ [WCX02]
	approx	k	k	k	———	?	?	$O(\log(\|V\|))$ [WCX02]
span 1	complexité	P [DS04a]	P	P	P	P	P	NP-Difficile [CL97]
	non approx							$o(\log(\|V\|))$ [WCX02]
	approx							$O(\log(\|V\|))$ [WCX02]
degré coloré borné		———	P	P	———	———	———	———

? = problème ouvert, $\delta = (\log\log|\mathcal{C}|^{\frac{1}{2}})^{-\varepsilon}$ pour $\varepsilon < \frac{1}{2}$, ——— = ne s'applique pas.

TAB. 4.1 – Complexité et propriétés d'approximabilité des problèmes colorés

$$\begin{aligned}
\text{Minimiser} \quad & \textstyle\sum_{c\in\mathcal{C}} y_c & & & (4.1)\\
\text{t.q.} \quad & y_c \geq |x_u - x_v| & \forall c \in \mathcal{C},\ \forall (u,v) \in c & & (4.2)\\
& x_u \geq 0 & \forall u \in V & & (4.3)\\
& y_c \in \{0,1\} & \forall c \in \mathcal{C} & & (4.4)\\
& x_s = 0 \text{ et } x_t = 1 & & & (4.5)
\end{aligned}$$

La fonction objectif peut être remplacée par $\sum_{c\in\mathcal{C}} w_c y_c$ pour prendre en compte la version pondérée du problème pour laquelle choisir une couleur c dans la coupe ne coûte plus 1 comme dans le programme linéaire ci-dessus, mais $w_c \geq 0$.

Ce MILP contient assez peu de variables binaires et n'importe quel solveur peut lui trouver une solution optimale en un temps *raisonnable* tant que le nombre de couleurs $|\mathcal{C}|$ reste *faible*.

Une formulation pour le problème MINIMUM COLOR CUT peut être déduite de la formulation donnée pour MINIMUM COLOR *st*-CUT simplement en remplaçant la ligne 4.5 : $\{x_s = 0,\ x_t = 1\}$ par $\{x_u = 0,\ \sum_{v\in V} x_v \geq 1\}$ pour un u quelconque. En d'autres termes, pour assurer l'existence de potentiels différents, et donc d'une partition des sommets du graphe, on impose d'une part qu'un sommet quelconque ait un potentiel nul et d'autre part que la somme des potentiels soit non nulle, c'est à dire qu'au moins un sommet du graphe ait un potentiel non nul.

4.3.0.2 Formulation MILP pour MC-*st*-PATH

La formulation suivante est adaptée de [YVJ05]. Elle ne concerne qu'un seul chemin contrairement à celle de [YVJ05] qui modélise la recherche de plusieurs chemins colorés et cherche à minimiser le nombre moyen de couleurs utilisées. Il s'agit d'une formulation sommet-arc classique légèrement modifiée. Pour trouver un chemin entre deux sommets s et t nous cherchons en fait un flot de valeur 1 entre ces sommets, la variable réelle $z_e \geq 0$ représente la valeur du flot circulant sur l'arête e. Il suffit ensuite d'ajouter une variable binaire y_c par couleur dont la valeur est fixée par la dernière contrainte pour que le problème modélisé soit le problème MC-*st*-PATH. La dernière contrainte implique que si au moins une arête e de couleur c porte un flot non nul $z_e > 0$, alors la couleur c doit être choisie et la variable $y_c > 0$. Dans le programme 4.6, $\delta^-(u)$ est l'ensemble des arêtes portant du flot entrant dans le sommet u, alors que $\delta^+(u)$ est l'ensemble des arêtes portant du flot sortant de ce sommet.

Programme Linéaire 1 (MC-*st*-PATH**)**

$$\begin{aligned}
& \textit{Minimiser} \quad \sum_{c\in\mathcal{C}} y_c \\
\textit{t.q.} \quad & \sum_{e\in\delta^+(u)} z_e - \sum_{e\in\delta^-(u)} z_e = \begin{cases} 0 & si \quad u \neq s, u \neq t \\ 1 & si \quad u = s \\ -1 & si \quad u = t \end{cases} \qquad \forall u \in V \\
& y_c \geq z_e \quad \forall c \in \mathcal{C},\ \forall e \in c \\
& z_e \geq 0 \quad \forall e \in E \\
& y_c \in \{0,1\} \quad \forall c \in \mathcal{C}
\end{aligned}$$

Notons que dans [YVJ05] la variable z_e est binaire alors que dans notre programme elle est réelle. Nous avons relâché la contrainte d'intégrité de cette variable car si un flot de valeur 1 utilise plus d'un chemin alors en particulier il en utilise 1 ce qui suffit à connecter s et t. L'ensemble de couleurs trouvé grâce à ce MILP est donc nécessairement de taille minimum et connecte s et t, il s'agit bien d'un chemin coloré de taille minimum.

Nous pouvons déduire un second programme linéaire mixte de la transformation d'un graphe coloré utilisée en section 4.2.3.3 pour montrer que si le nombre de couleurs de span supérieur à 1 est borné, le problème MINIMUM COLOR st-PATH est polynomial.

Programme Linéaire 2 (MC-st-PATH)

$$\textit{Minimiser} \quad \sum_{c \in \mathcal{C}} y_c$$

$$t.q. \quad \sum_{e \in \delta^+(u)} w_e - \sum_{e \in \delta^-(u)} w_e = \begin{cases} 0 & si \quad u \neq s, u \neq t \\ 1 & si \quad u = s \\ -1 & si \quad u = t \end{cases} \qquad \forall u \in V$$

$$y_c \geq w_e \quad \forall c \in \mathcal{C},\ \forall e = \{u, v\} \text{ tq u} \in \text{c ou v} \in \text{c}$$

$$w_e \geq 0 \quad \forall e \in E$$

$$y_c \in \{0, 1\} \quad \forall c \in \mathcal{C}$$

Il s'agit encore de faire circuler une unité de flot du sommet s au sommet t mais cette fois dans le graphe transformé. La variable $w_e \geq 0$ représente la valeur du flot sur l'arête e, la variable binaire y_c associée à la couleur c vaut 1 si du flot traverse au moins un sommet u correspondant à cette couleur dans le graphe transformé (noté $u \in c$ dans le programme 4.6), et 0 sinon. La première contrainte impose qu'un flot circule entre s et t et la seconde implique que si du flot circule sur une arête incidente à un sommet u de couleur c, alors le sommet u est traversé par du flot et la couleur c appartient au chemin.

Dans le programme 4.6 nous pouvons relâcher les variables y_c associées aux couleurs c de span égal à 1, c'est à dire $y_c \in [0, 1]$ au lieu de $y_c \in \{0, 1\}$. En effet dans une solution il existe une variable y_c dont la valeur n'est pas entière pour une couleur c de span 1, alors le flot est fractionné entre plusieurs chemins. Tous ces chemins sont de coût égaux car sinon il serait possible de déplacer tout le flot sur le chemin de coût minimum et ainsi d'obtenir une meilleure solution. Par conséquent nous pouvons déplacer tout le flot sur un unique chemin sans augmenter le coût de la solution. A partir d'une solution fournie par le programme linéaire mixte 4.6 dans lequel les variables y_c associées aux couleurs de span 1 sont relâchées, nous pouvons donc trouver un chemin utilisant un nombre minimum de couleurs, pour cela il suffit de retenir un seul chemin de cette solution.

Il est donc possible de réduire le temps de calcul d'un chemin coloré minimum en diminuant le nombre de variables binaires du programme 4.6. Lorsque le nombre de couleurs de span supérieur à 1 est borné, il y a donc un nombre borné de variables binaires. Ainsi le nombre de combinaisons de valeurs pour ces variables est également borné et le programme peut être résolu en temps polynomial. Ceci rejoint le fait que le problème MC-st-PATH est polynomial dans ce cas particulier déjà étudié en section 4.2.3.3.

4.4 Transformation

Les problèmes d'optimisation dans les graphes colorés sont polynomiaux (pour la plupart) lorsque le span maximum des couleurs est 1 mais sont très mal approximables dans le cas général.

Par conséquent il serait très arrangeant pour traiter les instances issues de réseaux réels de travailler avec des graphes dont les couleurs sont au plus de span 1 après la transformation ET présentée à la section 4.1.1 (page 58) d'un réseau réel vers un graphe coloré.

Bien entendu tous les réseaux n'induisent pas des graphes colorés de span maximum 1 mais plusieurs problèmes de décision ou d'optimisation peuvent être définis pour formuler la question de la transformation d'un réseau en un graphe aux spans aussi petits que possible. Nous nous intéresserons à la minimisation du nombre de couleurs de span supérieur à 1 après avoir étudié le problème consistant à décider si un réseau multicoloré peut être transformé en un graphe coloré de span maximum 1.

4.4.1 Décision : span 1 ?

Décider s'il existe une transformation d'un réseau en graphe coloré de span maximum 1 est polynomial. Outre un premier traitement du réseau, décider si une couleur peut être de span 1 nécessite deux vérifications, l'une concernant uniquement la couleur elle-même et les propriétés du graphe qu'elle induit, l'autre s'intéressant au partage des arêtes entre les couleurs.

4.4.1.1 Traitement initial du réseau

Le réseau peut contenir des arêtes multicolorées d'un nombre quelconque de couleurs, cependant seules deux d'entre elles peuvent être placées aux extrémités du chemin qui remplacera cette arête après transformation. Ainsi si toutes les couleurs de l'arête sauf au plus deux ne sont présentes dans le réseau que sur cette arête, il est inutile d'en tenir compte. Quelles que soient leurs positions, elles seront nécessairement de span 1 et elles pourront être placées à l'intérieur du chemin comme les couleurs c_1, c_3 et c_5 de la figure 4.21(a). Les extrémités seront alors libres pour placer les deux couleurs, c_2 et c_4, qui nécessitent un regroupement avec une autre occurrence de leur couleur autour de l'un des sommets extrémité du chemin. Le placement illustré par la figure 4.21(b) permet alors à toutes les couleurs d'être de span 1.

(a) un réseau multicoloré

(b) une transformation ET du réseau multicoloré en graphe coloré

FIG. 4.21 – Au plus deux couleurs peuvent être placées aux extrémités du chemin remplaçant une arête d'un réseau multicoloré.

En revanche pour l'arête $\{u, v\}$ du réseau 4.22, les couleurs c_1, c_2 et c_4 sont toutes les trois présentes sur d'autres arêtes multicolorées. Il est donc certain qu'au moins l'une d'elle ne sera pas placée à une extrémité du chemin et ne sera pas de span 1.

Lemme 4.3 *Un réseau multicoloré dont une arête comporte au moins trois couleurs qui appartiennent aussi à d'autres arêtes ne peut pas donner un graphe coloré de span maximum 1 par transformation* ET.

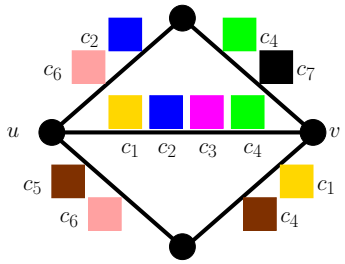


FIG. 4.22 – Les couleurs de l'arête $\{u, v\}$ ne peuvent pas être de span 1 simultanément.

Une première exploration du réseau multicoloré permet ainsi dans certains cas de décider qu'il n'est pas transformable en graphe coloré de span maximum 1. Pour les autres cas, cette exploration fourni un réseau simplifié où n'apparaissent que les couleurs dont la position sur les chemins du graphe coloré a réellement une importance.

Dans la suite l'étude de la transformation est donc restreinte au cas où les arêtes du réseau comportent toutes au plus deux couleurs distinctes.

4.4.1.2 Propriétés des couleurs de span 1

La première vérification consiste à décider indépendamment des autres couleurs si une couleur c peut être de span 1 après transformation ET par l'étude de l'agencement des arêtes du réseau portant la couleur c.

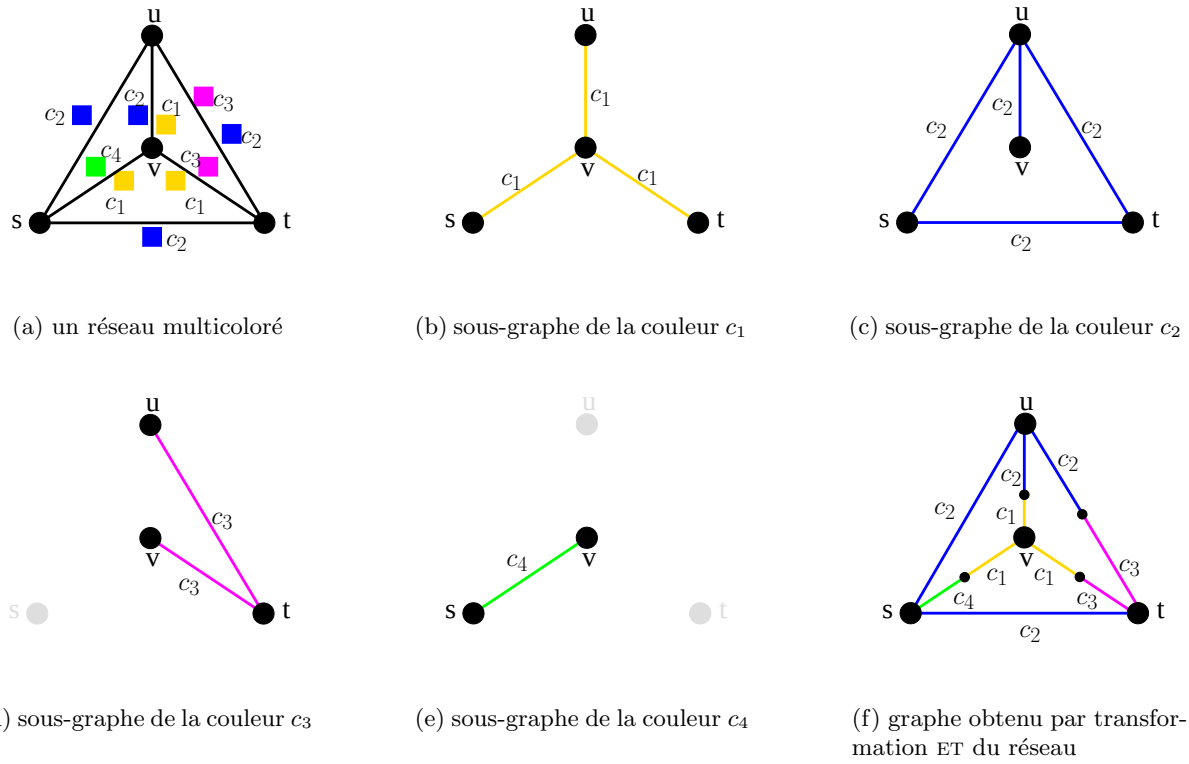
Connexité Considérons le sous-graphe d'une couleur c. Il s'agit du sous graphe du réseau induit par les arêtes multicolorées portant la couleur c. La figure 4.23(a) présente un réseau et les figures 4.23(b), 4.23(c), 4.23(d) et 4.23(e) les sous-graphes induits par les couleurs c_1, c_2, c_3 et c_4. Si ce sous-graphe n'est pas connexe la couleur ne peut évidemment pas être de span 1. Ceci peut se vérifier en temps polynomial par un simple parcours des sommets.

Dans la suite nous considérons donc que le sous-graphe d'une couleur c traitée est connexe.

Étoiles Dans le graphe d'une couleur c les arêtes peuvent être divisées en deux classes, les arêtes *fixes* et les arêtes *positionnables*.

Les arêtes fixes sont des arêtes du réseau qui ne portent en fait qu'une seule couleur au lieu de deux, comme les arêtes $\{s, u\}$ et $\{u, t\}$ du réseau 4.23(a) qui ne portent que la couleur c_2. Ces arêtes ne nécessitent pas de transformation puisqu'elles sont déjà monocolorées (Figure 4.23(f)). Outre ceci, leur particularité est que quoi qu'il advienne du reste du réseau, les deux extrémités d'une arête fixe pour la couleur c appartiendrons à la même composante connexe vis à vis de cette couleur.

Les arêtes positionnables correspondent à toutes les autres arêtes multicolorées du réseau qui portent deux couleurs. Il faut donc positionner chaque couleur à l'une ou l'autre des extrémités du chemin de longueur deux qui remplacera l'arête positionnable après transformation du réseau. Deux sommets du réseau ne peuvent pas appartenir à la même composante connexe vis à vis d'une couleur c s'ils ne sont reliés que par des arêtes positionnables portant cette couleur. En effet une fois la transformation effectuée, la couleur c sera adjacente soit à l'un de ces sommets soit à l'autre mais pas aux deux en même temps. Sur la figure 4.23(f) le chemin issu de l'arête $\{s, v\}$ illustre cette propriété des arêtes positionnables, la couleur c_4 est adjacente à s mais pas à v.



(a) un réseau multicoloré

(b) sous-graphe de la couleur c_1

(c) sous-graphe de la couleur c_2

(d) sous-graphe de la couleur c_3

(e) sous-graphe de la couleur c_4

(f) graphe obtenu par transformation ET du réseau

FIG. 4.23 – Exemple de réseau multicoloré et des sous-graphes des couleurs.

Par conséquent pour qu'une couleur soit de span 1, il faut que toutes les arêtes positionnables soient adjacentes à au moins une arête fixe s'il en existe dans le graphe de la couleur, mais aussi que les arêtes fixes forment un sous-graphe connexe du graphe de la couleur.

Lemme 4.4 *Dans un réseau multicoloré, pour qu'une couleur comportant au moins une arête fixe puisse être de span 1 après une transformation* ET*, il est nécessaire que :*

1. *toutes les arêtes fixes forment un sous-graphe connexe,*
2. *toutes les arêtes positionnables soient adjacentes à au moins une des arêtes fixes.*

En effet si deux arêtes fixes comme les arêtes $\{x, y\}$ et $\{z, t\}$ pour la couleur c_1 de la figure 4.24 ne forment pas une composante connexe mais sont reliées par une arête positionnable, alors quelle que soit la position choisie pour la couleur sur cette arête positionnable, les deux arêtes fixes ne feront pas partie de la même composante.

D'autre part, s'il n'y a pas d'arête fixe dans le sous-graphe de la couleur c, deux sommets de ce graphe ne pourrons pas être connexes après transformation puisque les arêtes positionnables ne permettent pas de connecter deux sommets. Ainsi, un seul sommet du graphe de la couleur peut faire partie de la composante de la couleur c après transformation. Pour les besoins de la transformation des sommets sont créés qui seront au milieu des chemins et qui eux pourront appartenir à cette composante, mais seul un sommet du réseau initial ne subsiste dans la composante connexe de la couleur après transformation, comme la couleur c_1 des figures 4.23(a), 4.23(b) et 4.23(c) l'illustre. Pour trouver si un sommet convient il suffit de vérifier si l'un d'entre eux est adjacent à toutes les arêtes multicolorées portant la couleur traitée c, c'est à dire de vérifier si l'intersection des extrémités de toutes les arêtes n'est pas vide. En d'autres termes il faut vérifier l'existence d'un

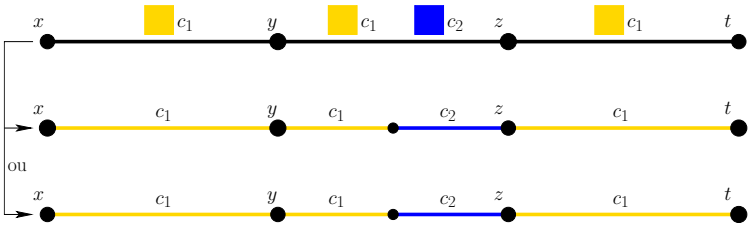


FIG. 4.24 – Deux arêtes fixes pour une couleur doivent appartenir à la même composante connexe sinon la couleur ne peut pas être de span 1.

sommet au centre d'une étoile rassemblant toutes les arêtes de couleur c dans le graphe de cette couleur. On parlera de sommets centraux également pour les extrémités des arêtes fixes lorsqu'il en existe.

Lemme 4.5 *Dans un réseau multicoloré, une couleur c ne comportant aucune arête fixe ne peut être de span 1 après transformation* ET *que si l'intersection de ses arêtes est non vide, c'est à dire que toutes ses arêtes doivent être incidentes à un même sommet.*

La recherche de groupes d'arêtes fixes connexes ou d'étoiles dans le graphe d'une couleur ne permet pas seulement de décider si toutes les couleurs du réseau pourront être de span 1 indépendamment les unes des autres après transformation, mais permet aussi de positionner les arêtes des couleurs qui le peuvent. Si les arêtes de couleur c forment une étoile de centre le sommet u, alors elles devront toutes être transformées en un chemin dont l'arête adjacente à u est de couleur c, et c'est la seule solution pour garantir le span. De même, il n'y a pas d'alternatives pour les arêtes adjacentes à une arête fixe, elles doivent nécessairement être transformées en un chemin dont l'extrémité adjacente à une arête fixe est de couleur c.

Lemme 4.6 *Soit c une couleur pour laquelle un ensemble non vide K_c de sommets centraux à été trouvé. Alors c ne peut être de span 1 que si toutes les arêtes de couleur c sont incidentes aux sommets de K_c après transformation* ET.

Une seule sous classe d'arêtes positionnables échappe à ce placement forcé, il s'agit des arêtes *libres*. Les arêtes libres pour une couleur sont les arêtes sur lesquelles le placement de cette couleur est sans incidence sur son span. Les arêtes portant une couleur qui n'est présente sur aucune autre arête du réseau, comme l'arête $\{s, v\}$ pour la couleur c_4 dans le réseau 4.23(a), sont des arêtes libres. D'autres arêtes libres pour une couleur c sont les arêtes adjacentes à leurs deux extrémités à des arêtes fixes appartenant à la même composante connexe vis à vis de la couleur considérée c. La position de la couleur c après transformation sur ces arêtes n'a pas d'importance puisque les deux sommets auxquels c peut être adjacente après transformation font déjà partie de la même composante connexe vis à vis de c. Dans le réseau 4.23(a) l'arête $\{u, t\}$ est libre pour la couleur c_2 car elle est adjacente à $\{s, u\}$ et $\{s, t\}$ qui sont fixes toutes les deux. C'est grâce à la confrontation avec les impératifs des autres couleurs qui partagent une arête du réseau avec c que la position de c sur les arêtes libres pourra être déterminée.

Lemme 4.7 *La position après transformation* ET *d'une couleur c sur une arête e n'a aucune influence sur le span de c dans les cas suivants :*

- *la couleur c n'appartient qu'à une seule arête, l'arête e,*
- *l'arête e est adjacente par ses deux extrémités à deux arêtes fixes pour la couleur c.*

4.4.1.3 Cohabitation des couleurs

La confrontation avec les autres couleurs ne permet pas seulement de positionner les couleurs sur les arêtes libres mais aussi de décider finalement si le graphe pourra être de span 1. Après les deux traitements effectués, sur chaque arête du réseau puis sur chaque couleur, les conflits de position entre deux couleurs constituent la dernière raison pour laquelle un réseau ne pourrait pas être transformé en graphe de span maximum 1.

Deux couleurs sont en conflit lorsque des étoiles ou des arêtes fixes imposent qu'elles soient toutes les deux positionnées à la même extrémité d'une arête du réseau qu'elles partagent. Détecter un conflit peut se faire en temps polynomial, puisqu'il suffit d'énumérer toutes les arêtes du réseau et de comparer les positions imposées pour chacune des deux couleurs partageant une arête. Notons qu'il ne peut pas y avoir de conflit sur une arête qui est libre pour au moins une des deux couleurs, ni pour les arêtes fixes car elles ne portent qu'une couleur.

Lemme 4.8 *Un réseau multicoloré ne peut être transformé en graphe coloré de span 1 que si deux couleurs partageant une arête peuvent ne pas avoir la même position.*

4.4.1.4 Arêtes multiples

Les réseaux peuvent contenir des arêtes multiples qui requièrent un traitement spécial du point de vue de la transformation.

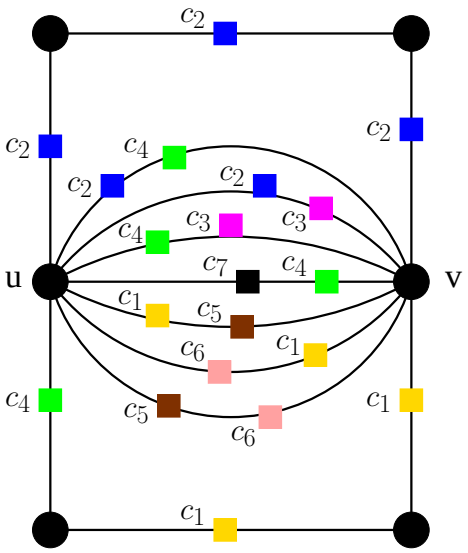


FIG. 4.25 – Réseau avec arête multiple.

La première remarque a propos des arêtes multiples est qu'elles peuvent parfois se décomposer en plusieurs arêtes multiples indépendantes comme dans le réseau 4.25. Les deux sous-ensembles de couleurs $\mathcal{C}_1 = \{c_2, c_3, c_4, c_7\}$ et $\mathcal{C}_2 = \{c_1, c_5, c_6\}$ induisent une partition des arêtes composant l'arête multiple $\{u, v\}$. Aucune arête entre u et v ne comporte à la fois une couleur de $\mathcal{C}_1$ et une de $\mathcal{C}_2$. Ces deux sous-ensembles d'arêtes peuvent donc être traités indépendamment l'un de l'autre.

Ensuite les couleurs peuvent être classées en trois catégories. La première catégorie regroupe les couleurs *libres* sur une arête. Les couleurs c_2 et c_7 du réseau 4.25 sont libres puisque leur position

est indifférente. La couleur c_2 est libre sur l'arête multiple $\{u, v\}$ car u et v font tous les deux partie d'une même composante connexe vis à vis de la couleur c_2 alors que c_7 est libre car cette couleur n'est présente dans tout le réseau que sur l'arête uv, son span sera 1 quoi qu'il arrive.

La seconde catégorie englobe toutes les couleurs qui doivent impérativement être adjacentes à un sommet précis. Dans le réseau 4.25 la couleur c_1 doit impérativement être adjacente à v, alors que c_4 doit être adjacente à u. La position de ces couleurs est fixée comme sur n'importe quelle arête positionnable.

La troisième catégorie comprend les couleurs *semi-libres* n'ayant pas d'impératif concernant le sommet auquel elles seront adjacentes, mais devant cependant être adjacentes à un unique sommet. C'est le cas des couleurs c_3, c_5 et c_6 du réseau 4.25. En effet la couleur c_3 peut être indifféremment adjacente à u ou v, mais si l'une des occurrences de c_3 dans l'arête multiple est adjacente à u, par exemple, alors toutes les occurrences de c_3 devront aussi être adjacentes à u. Ces arêtes ne sont pas libres mais ne peuvent être positionnées que par confrontation avec les autres couleurs. Notons que les couleurs c_5 et c_6 sont en conflit dès lors que la couleur c_1 est positionnée puisqu'alors elles doivent toutes les deux être adjacentes au sommet u, mais comme elles partagent une arête cela est impossible.

Lemme 4.9 *Une couleur c semi-libre ne peut être de span 1 après transformation* ET *que si les positions laissées libres pas les autres couleurs sur les arêtes portant la couleur c, sont toutes incidentes à un même sommet.*

4.4.1.5 Algorithme exact et polynomial

L'existence d'un algorithme polynomial décidant si un réseau multicoloré peut être transformé en graphe coloré de span maximum 1 repose sur les points évoqués précédemment et résumés ici :

Si plus de trois couleurs partagent une arête et sont aussi présentes sur au moins une autre arête du réseau, elles ne peuvent pas être de span 1 simultanément, ce qui permet de se limiter au cas ou chaque arête porte au plus deux couleurs (lemme 4.3 page 87).

Pour qu'une couleur puisse être de span 1, soit il existe des arêtes fixes, et alors elles doivent former un sous-graphe connexe et les autres arêtes doivent être adjacentes à au moins une arête fixe, soit il n'en existe pas et toutes les arêtes doivent avoir un unique sommet en commun (lemmes 4.4 et 4.5 pages 89 et 90).

Lorsque toutes les arêtes d'une couleur sont incidentes à un ensemble de sommets centraux, la position de cette couleur sur ces arêtes est imposée, excepté pour les arêtes libres (lemmes 4.6 et 4.7 page 90).

Deux couleurs partageant une arête ne peuvent être de span 1 que si les positions qui leur sont imposées ne sont pas en conflit (lemme 4.8 page 91).

Enfin, les couleurs semi-libres peuvent être de span 1 à condition que les positions laissées par les couleurs avec lesquelles elles partagent une arête soient toutes adjacentes à un même sommet (lemme 4.9 page 92).

Chacun de ces points permet de détecter des conflits qui empêchent un réseau d'être transformé en un graphe coloré de span 1. Si aucun de ces conflits n'est trouvé, alors on obtient pour chaque couleur un positionnement qui n'est en conflit avec aucune autre couleur, et qui est tel que la couleur est de span 1 dans le graphe coloré issu du réseau. Les couleurs semi-libres peuvent être positionnées en sorte d'avoir span 1 sans conflit, les couleurs qui comportent des arêtes fixes et celles qui n'en comportent pas également, ce qui couvre l'ensemble des couleurs.

D'autre part toutes ces vérifications peuvent se faire en temps polynomial. Nous proposons en annexe B un algorithme de complexité $\mathcal{O}(|\mathcal{C}||E_{\mathcal{R}}| + |\mathcal{C}|^2|E_{\mathcal{R}}| + |E_{\mathcal{R}}|)$ qui ne prétend pas être

optimisé.

Le premier traitement nécessite un parcours des couleurs pour chaque arête ($\mathcal{O}(|\mathcal{C}||\mathcal{E}_\mathcal{R}|)$). Déterminer les couleurs semi-libres peut se faire par un parcours des arêtes pour chaque couleur ($\mathcal{O}(|\mathcal{C}||\mathcal{E}_\mathcal{R}|)$). Pour chaque couleur un parcours des arêtes permet de déterminer les sommets centraux, un second de vérifier la connexité des arêtes fixes lorsqu'il en existe, et un troisième permet de fixer les positions des couleurs sur les arêtes incidentes aux sommets centraux ($\mathcal{O}(|\mathcal{C}||\mathcal{E}_\mathcal{R}|)$). Pour positionner chaque couleur semi-libre il faut dans un premier temps trouver une arête partagée avec une couleur déjà positionnée ($\mathcal{O}(|\mathcal{C}||E_\mathcal{R}|)$) puis parcourir toutes les arêtes comportant cette couleur semi-libre ($\mathcal{O}(|\mathcal{C}||E_\mathcal{R}|)$). Ensuite s'il n'existe plus de couleur semi-libre partageant une arête avec une couleur déjà positionnée, il faut positionner l'une des couleurs semi-libres restantes aléatoirement ($\mathcal{O}(|E_\mathcal{R}|)$) et recommencer depuis le début du positionnement des couleurs semi-libres jusqu'à ce qu'il n'en reste plus. La complexité en temps de l'algorithme que nous proposons pour le positionnement des couleurs semi-libres est donc $\mathcal{O}(|\mathcal{C}||E_\mathcal{R}| + |\mathcal{C}|^2|E_\mathcal{R}|)$. Enfin la confrontation des couleurs sur chaque arête nécessite un simple parcours des arêtes ($\mathcal{O}(|E_\mathcal{R}|)$).

4.4.2 Maximiser le nombre de couleurs de span 1

Comme il n'est pas toujours possible d'obtenir un graphe coloré de span maximum 1 après transformation d'un réseau, le problème de maximiser le nombre de couleurs de span 1 prend tout son intérêt. En effet, nous avons vu dans la section 4.2.3.3 (page 70) que la résolution de plusieurs problèmes peut se faire en temps polynomial si le nombre de couleur de span plus grand que 1 est borné. Aussi, étant donné un réseau multicoloré, il serait très intéressant de savoir le transformer en un graphe coloré dont le nombre de couleurs de span 1 est maximum. Ceci permettrait entre autres de savoir identifier des cas polynomiaux. Cependant, nous allons voir que le problème de transformer un réseau multicoloré en un graphe coloré tout en maximisant le nombre de couleurs de span > 1 est NP-difficile et difficile à approximer.

Nous pouvons formaliser ce problème de la façon suivante.

Problème 4.8 (Transformation**)**

Entrée :	*Un réseau multicoloré* $\mathcal{R} = (V_\mathcal{R}, E_\mathcal{R}, \mathcal{C})$.
Sortie :	*Un graphe coloré* $G = (V_G, E_G, \mathcal{C})$ *obtenu à partir de* $\mathcal{R}$ *par la transformation* ET*, c'est à dire en remplaçant chaque arête multicolorée de* $E_\mathcal{R}$ *par un chemin d'arêtes monocolorées dans* G.
Objectif :	*Maximiser le nombre de couleurs de span 1 dans* G.

En d'autres termes, les solutions du problème sont des placements des couleurs sur les chemins d'arêtes monocolorées remplaçant les arêtes multicolorées de $\mathcal{R}$.

4.4.2.1 Complexité et approximabilité

La complexité de ce problème découle d'une réduction du problème Maximum Set Packing.

Théorème 4.9 *Il existe une réduction préservant l'approximabilité du problème* Maximum Set Packing *vers le problème* Transformation.

Preuve: Soit $\mathcal{R}$ un réseau construit à partir d'une instance quelconque de Maximum Set Packing. Ce réseau multicoloré contient un sommet u_i pour chaque élément $u_i \in U$, chaque sommet est connecté à un même sommet supplémentaire v de $\mathcal{R}$ par une arête multicolorée formant ainsi une étoile (Figure 4.26).

Chaque sous-ensemble $c_j \in \mathcal{C}$ de MAXIMUM SET PACKING est identifié à une couleur du réseau $\mathcal{R}$ et l'ensemble de couleurs de l'arête multicolorée vu_i représente la sous collection d'ensembles $c_j \in \mathcal{C}$ contenant u_i dans l'instance de MAXIMUM SET PACKING.

Si un ensemble c_k est réduit à un seul élément u_k, un sommet $u_{k\text{bis}}$ est ajouté au réseau $\mathcal{R}$ et connecté à v par une arête dont l'ensemble de couleurs est réduit à la couleur c_k.

Un ensemble $c_j \in \mathcal{C}$ est ainsi représenté par sa couleur sur les arêtes multicolorées connectant v à chacun des sommets correspondant à un de ses éléments.

Puisque chaque ensemble $c_j \in \mathcal{C}$ est représenté sur au moins deux arêtes multicolorées, pour être de span 1 chaque arête d'une couleur donnée doit être adjacente à v après transformation, et une seule couleur par arête multicolorée ne peut être adjacente à v. Ainsi deux couleurs portées par une même arête ne peuvent pas être de span 1 simultanément et donc deux couleurs de span 1 correspondent à des ensembles disjoints puisqu'elles ne doivent pas partager d'arêtes.

Par conséquent un ensemble de couleurs de span 1 dans le graphe coloré obtenu après la transformation du réseau $\mathcal{R}$ correspond à une sous collection de même taille d'ensembles nécessairement disjoints de $\mathcal{C}$ dans l'instance de MAXIMUM SET PACKING initiale.

D'autre part, une sous collection d'ensembles disjoints pour l'instance de MAXIMUM SET PACKING induit un ensemble de couleurs pouvant être de span 1 simultanément après transformation du réseau $\mathcal{R}$ de même taille que la sous collection. Puisque les ensembles de la sous collection sont disjoints, les couleurs qui les représentent dans le réseau ne partagent deux à deux aucune arête et peuvent donc toutes être adjacentes à v en même temps. ∎

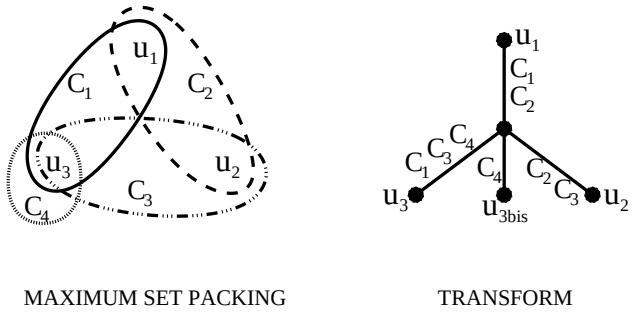


FIG. 4.26 – Exemple de réseau construit à partir d'une instance de MAXIMUM SET PACKING.

Corollaire 4.3 *Le problème* TRANSFORMATION *est NP-Difficile et non approximable à un facteur* $|\mathcal{C}|^{\frac{1}{2}-\varepsilon}$ *pour tout* $\varepsilon > 0$ *sauf si* $P = NP$. *De plus, il est non approximable à un facteur* $|\mathcal{C}|^{1-\varepsilon}$ *pour tout* $\varepsilon > 0$ *sauf si* $NP = ZPP$[2].

Le corollaire 4.3 anéanti tout espoir de trouver de bonnes approximations du nombre maximum de couleurs de span 1 obtenu par transformation d'un réseau, mais laisse toutefois la possibilité d'utiliser des méthodes heuristiques pour les instances réelles à résoudre.

4.4.2.2 Une piste pour des méthodes heuristiques

A chaque fois que l'algorithme de décision décrit en section 4.4.1 (page 87) produit une preuve qu'une couleur ne peut pas être de span 1, il élimine en fait une couleur sur laquelle une heuristique qui cherche à maximiser le nombre de couleurs de span 1 n'a pas besoin de travailler.

[2]Voir annexe A

Par contre tous les choix à faire par une telle heuristique résident dans le fait que certaines couleurs ne peuvent pas être de span 1 simultanément mais le pourraient indépendamment les unes des autres. Par exemple lorsque trois couleurs portées par une même arête sont présentes sur d'autres arêtes, ou lorsque deux couleurs doivent être positionnées sur la même extrémité d'une arête pour être de span 1.

Le point de départ pour une heuristique pourrait être de débarrasser chaque arête du réseau de toutes les couleurs libres sur cette arête, puis d'éliminer toutes les couleurs qui ne peuvent pas être de span 1 quelles que soient les positions fixées. Enfin il s'agirait de choisir lorsque deux couleurs sont en conflit pour une position laquelle des deux n'aura pas span 1 après transformation. Comme plusieurs stratégies sont toujours possibles en présence de choix, ce cadre général peut donner lieu à plusieurs heuristiques différentes faisant intervenir plus ou moins de décisions aléatoires.

A défaut de donner des solutions optimales cette procédure donne une borne supérieure du nombre maximum de couleurs de span 1 dans le graphe obtenu après transformation, c'est le nombre de couleurs qui pourraient avoir span 1 indépendamment des autres.

4.5 Conclusion

Les graphes colorés offrent une modélisation simple des réseaux multiniveaux, adaptée à l'étude des problèmes d'optimisation fondamentaux relatifs au routage et à la tolérance aux pannes en présence de groupes de risque. Ces problèmes, dérivés de la théorie des graphes classique, se répartissent en problèmes de connexité et problèmes de vulnérabilité. Les problèmes de connexité consistent à trouver des chemins entre deux sommets du graphe. Il s'agit par exemple de trouver des chemins couleurs disjoints entre deux sommets : ils représentent dans un réseau un ensemble de chemins ne tombant pas en panne simultanément. Un chemin utilisant un minimum de couleurs est en pratique un chemin de probabilité de panne minimum. Les problèmes de vulnérabilité sont d'un point de vue de la théorie des graphes des problèmes de coupe. En effet un graphe coloré dont la coupe minimum en nombre de couleur est faible représente un réseau qui peut être déconnecté par un faible nombre de pannes.

Contrairement à leurs équivalents classiques polynomiaux, ces problèmes sont difficiles et mal approximables. De plus, nous avons montré que les relations essentielles qui constituent les éléments de base des méthodes de résolution de nombreuses questions de tolérance aux pannes dans les réseaux à un niveau, ne sont plus vraies dans les graphes colorés. C'est le cas de la relation flot max-coupe min.

Nous avons contribué à l'approfondissement des connaissances sur les graphes colorés en étudiant la complexité et l'inapproximabilité des problèmes colorés. Nous avons amélioré certains résultats et nous en avons proposé de nouveaux, mais surtout, nous avons mis en évidence un paramètre incontournable de la complexité des problèmes Minimum Color st-Path et Minimum Color st-Cut, le span des couleurs. En effet, lorsque les couleurs sont de span borné par une constante k, ces deux problèmes sont approximables à un facteur k près et non approximables à un facteur k^{ε} pour une constante $\varepsilon > 0$ particulière. Pour $k = 1$ ces problèmes sont polynomiaux alors qu'en l'absence d'hypothèses sur le span ils sont difficiles à approximer à un facteur $2^{\log^{1-\delta}|C|^{\frac{1}{2}}}$ avec $\delta = (\log\log|C|^{\frac{1}{2}})^{-\varepsilon}$ pour tout $\varepsilon < 1/2$ sauf si $P = NP$. De plus bien que la NP-difficulté du problème Minimum Color Cut reste une conjecture dans le cas général ainsi que lorsque le span maximum du graphe est borné par une constante, les quelques cas polynomiaux que nous avons identifiés laissent supposer que la complexité et les propriétés d'approximabilité de Minimum Color Cut dépendent également du span.

En revanche, le span ne semble avoir aucune influence (ou peu) sur les problèmes de chemins couleur-disjoints et Minimum Color Spanning Tree. Si le problème Minimum Color Spanning Tree possède les mêmes propriétés que Minimum Set Cover, nous avons montré que trouver un nombre maximum de chemins couleur-disjoints n'est pas approximable à un facteur $O(n^{\frac{1}{4}-\varepsilon})$ pour tout $\varepsilon > 0$ sauf si $P = NP$.

Suite au constat de l'importance du span pour certains problèmes, la question de la transformation d'un réseau multiniveaux en graphe coloré prend tout son intérêt. En effet, de cette transformation dépend dans une certaine mesure le span des couleurs, et donc la complexité des problèmes précédents. Nous avons montré que décider si un réseau peut être transformé en un graphe coloré de span maximum 1 est polynomial mais qu'il est NP-difficile de maximiser le nombre de couleurs de span 1.

Nous envisageons la poursuite de l'étude des graphes colorés sous différents angles. Dans un premier temps nous devons traiter la question de la transformation d'un réseau multicoloré en graphe coloré de span maximum deux ou tout autre constante, comme nous l'avons déjà fait dans le cas de span 1.

Ensuite, il s'agit de trancher les questions ouvertes comme la complexité de Minimum Color Cut dans le cas général, les propriétés exactes d'approximabilité des problèmes colorés, c'est à dire de trouver des algorithmes d'approximation et des facteurs d'inapproximabilité serrés pour tous les problèmes colorés.

Toutefois, les facteurs d'inapproximabilité que nous avons déjà obtenus étant élevés, connaître des algorithmes d'approximation a un intérêt essentiellement théorique. Dans la pratique, obtenir une solution optimale même au bout de plusieurs heures (jours ?) de calcul peut être préférable à obtenir une solution rapidement mais pouvant être relativement éloignée de l'optimale en conséquence d'un facteur d'approximabilité élevé. Ce peut être le cas par exemple pour les problèmes de vulnérabilité pouvant intervenir dans le processus de planification et de réoptimisation d'un réseau. La question qui se pose est donc de trouver des méthodes de résolution exactes, exponentielles mais cependant aussi efficaces que possible. Nous souhaitons en particulier approfondir l'influence du span sur les problèmes colorés notamment par le biais de la *complexité paramétrique* [DF99, Nie06].

Enfin, la recherche approfondie d'éventuelles propriétés des spans des couleurs dans les graphes colorés issus de réseaux réels pourrait mettre en évidence de nouveaux cas polynomiaux. L'objectif serait alors de proposer des méthodes de résolution polynomiales efficaces, et si possible exactes, des problèmes d'optimisation dans les réseaux réels. Nous pourrions ainsi étendre les résultats que nous avons obtenus dans les graphes dont le nombre de couleurs de span supérieur à 1 est borné.

Chapitre 5

Surveillance du trafic

Mesurer différents paramètres d'un réseau et du trafic qui y circule est essentiel pour estimer les performances qu'il peut atteindre, pour identifier et localiser certains problèmes, en particulier les pannes. Deux stratégies courantes, l'une passive l'autre active, sont utilisées. L'approche passive consiste à équiper les liens du réseau d'appareils spécifiques capables d'enregistrer le trafic qui transite par ce lien. Pour l'approche active des paquets de contrôle sont émis par les nœuds, leur trajet et leur date d'arrivée en d'autres nœuds fournissent de précieuses informations. Les objectifs principaux dans le domaine de la surveillance du trafic sont la minimisation des coûts en terme d'équipements, de logiciels, de maintenance et de trafic supplémentaire.

Dans ce chapitre nous étudions le problème de l'affectation au liens du réseau d'équipements d'enregistrement pour la surveillance passive et de l'affectation aux nœuds d'appareils d'émission de paquets de contrôle pour la surveillance active. Minimiser le nombre d'équipements et trouver leur localisation optimale sont essentiels pour le déploiement de la surveillance à l'échelle d'un réseau. Nous présentons une modélisation du problème grâce à laquelle nous obtenons des résultats de complexité et d'approximabilité mais aussi des formulations en programme linéaire mixte efficaces.

5.1 Motivations

Internet et les réseaux IP sont au centre d'activités toujours plus nombreuses, les services proposés se multiplient. Tous ces services requièrent des niveaux d'exigence divers en termes de qualité, de sécurité, de fiabilité etc. Afin d'établir avec leurs clients des accords de service (SLA pour *Service Level Agreement*) qu'ils sont en mesure de respecter, les fournisseurs d'accès à Internet, ou ISP pour *Internet Service Provider*, doivent maintenant connaître avec précision les performances de leurs points de présence (POP) et la nature des flux qui y circulent. La métrologie est donc devenue une activité essentielle pour les opérateurs qui leur permet de concevoir une ingénierie efficace pour leurs offres de services. Pour la recherche sur les réseaux de télécom, la métrologie est aussi importante car elle permet la validation et l'amélioration des modèles dont le but est de représenter le plus fidèlement possible la réalité.

Jusqu'à maintenant les travaux sur la surveillance et l'analyse du trafic ont porté sur plusieurs thèmes [OASC03] comme la mesure de la QoS (délais, pertes, débits etc) [JID+04], la tomographie du trafic [HLO03], la détection des points de congestion et la découverte de la topologie du réseau. Des études ont également été menées dans le but de déterminer les matrices de trafic et les évolutions de la nature du trafic avec comme applications le dimensionnement de réseau et la détection d'intrusions [MVS01, KL03, BP01].

Plusieurs méthodes de mesure différentes ont été envisagées, chacune permettant de mieux comprendre un aspect spécifique des réseaux IP. On distingue les approches passives, divisées en mesures passives en ligne et hors ligne, et les approches actives.

La surveillance active est basée sur l'émission de trafic supplémentaire dans le réseau et l'étude de son acheminement. C'est pour le moment la seule méthode qui permet à un utilisateur de connaître les paramètres du service dont il pourra bénéficier dans le réseau. Cependant ces mesures sont biaisées par l'introduction du trafic de mesure qui perturbe l'état du réseau. Les approches actives sont tout de même utilisées couramment ne serait-ce qu'avec les outils classiques comme *ping*, qui permet en particulier la vérification de l'existence d'un chemin entre deux machines et la mesure du taux de perte, ou *traceroute*, qui donne la liste des routeurs traversés par les paquets jusqu'à leur destination et une indication de leurs temps d'acheminement. La surveillance active permet également au niveau IP la détection de panne dans le réseau.

Contrairement aux mesures actives, les mesures passives n'ont pour ainsi dire aucune incidence sur le fonctionnement du réseau. La surveillance passive consiste à placer sur les liens du réseau des appareils de mesure collectant le trafic circulant sur le lien et enregistrant les informations importantes issues des paquets capturés comme leur date et heure d'arrivée. Les analyses peuvent se faire soit en ligne, soit hors ligne. Pour les mesures en ligne, toute l'analyse doit être effectuée pendant la courte durée correspondant au passage d'un paquet dans l'appareil de mesure. Ce type d'analyse permet d'effectuer des mesures sur de très longues périodes et ainsi d'obtenir des statistiques significatives. Par contre le faible temps de calcul disponible pour chaque paquet limite la complexité des calculs réalisés. Dans le cas des analyses hors ligne, une trace du trafic doit être sauvegardée ce qui limite cette fois la durée des observations. Les calculs peuvent en revanche être beaucoup plus approfondis et permettre l'étude de propriétés non triviales du trafic. L'échantillonnage du trafic (*sampling*) consiste à ne garder la trace que d'un certain pourcentage du trafic, ce qui permet d'effectuer des mesures sur de plus longues périodes. Il existe plusieurs techniques d'échantillonnage qui seront décrites plus tard. Dans tous les cas les mesures passives ne permettent pas de connaître les performances qu'un utilisateur peut attendre du réseau.

5.2 État de l'art

Plusieurs projets se sont intéressés aux mesures de performances des réseaux. La métrologie et la surveillance font l'objet d'études partout dans le monde. Le groupe de travail IPPM, IP *Performance Metrics*, de l'IETF relatif aux métriques de performances d'IP [PAMM98] développe un ensemble de métriques standards qui peuvent être appliquées aussi bien à l'évaluation des performances qu'à la qualité et la fiabilité des services d'acheminement de données de l'Internet. Le groupe de travail IPFIX, pour IP *Flow Information eXport*, [QZCZ04] travaille sur la définition du protocole IPFIX et des recommandations pour son implémentation. IPFIX est inspiré du protocole NetFlow de Cisco et a pour but l'unification des méthodes de mesure des flux IP, de collecte et d'échange de l'information mesurée entre les équipements du réseau. L'objectif du groupe de travail BMWG, *Benchmarking Methodology Working Group*, est d'établir des recommandations concernant la pertinence des indicateurs de performance caractéristiques pour diverses technologies de l'Internet englobant à la fois les équipements, les systèmes et les services. La spécification des méthodes d'échantillonnage des paquets parmi les flux IP ainsi que des informations à collecter est un des objectifs principaux du groupe de travail PSAMP (*Packet Sampling*). La fonction du groupe de recherche IMRG (*Internet Measurement Research Group*) de l'IRTF est d'offrir un espace d'échange entre tous les groupes travaillant sur le thème général des mesures de l'Internet.

D'ambitieux projets ont été lancés pour effectuer des mesures réelles dans l'Internet grâce à

des plateformes de grande échelle, comme NIMI [1] (*National Internet Measurement Infrastructure*) [PAM00]. Le groupe MNA (*Measurement and Network Analysis*) du NLANR (*National Laboratory for Applied Network Research*) s'intéresse spécialement aux réseaux HPC (*High Performance Connection*), c'est à dire constitués de connections à hautes performances alors que le projet IPMON (IP *MONitoring*)[2] de Sprint a pour objectif la mise en place d'un système général de mesure des réseaux IP capable de collecter à la fois des statistiques détaillées du comportement du trafic au niveau paquet, et des statistiques sur les délais, les pertes et autres indicateurs de performance du réseau.

Les mesures sont nécessaires pour estimer les performances, ainsi qu'identifier et localiser les problèmes dans les réseaux. Les observations du trafic représentent des données essentielles pour la gestion d'un réseau et la recherche. La stratégie déployée pour obtenir ces informations est par conséquent d'une grande importance pour la communauté de chercheurs travaillant sur ce thème [GT00, JJJ+00, SMW02] connu sous le nom de tomographie de l'Internet. La majorité des contributions concernent soit la découverte de topologies soit la surveillance des délais sur les liens. Des travaux comme [BDJ01] étudient les demandes de trafic dans un réseau IP, l'identification des routes qu'elles utilisent et l'évaluation des granularité de trafic dont l'utilisation permettrait une meilleure répartition de la charge dans le réseau. Dans [JID+04], les auteurs proposent une méthodologie de surveillance passive permettant de collecter des informations spécifiques sur le trafic en provenance d'un émetteur afin d'estimer les performances d'un réseau du point de vue d'un utilisateur.

D'autres travaux montrent que la surveillance active permet la détection de pannes dans les réseaux IP [HLO03, NT04, BR03]. En effet dans les réseaux IP aucune information sur l'état du réseau n'est fournie et la surveillance active prend tout son intérêt dans la mise en œuvre d'une ingénierie de trafic. La surveillance active nécessite l'installation de plusieurs points de mesures, appelés *beacon*, dont la fonction est l'émission de paquets IP, ou *sondes*, à destination de tous les autres nœuds du réseau. Une panne est détectée lorsque plusieurs sondes successives n'utilisent pas le même chemin entre leur source et leur destination [NT04].

Tous ces projets et études utilisent largement la surveillance pour détecter et signaler des problèmes ou des anomalies, mais aussi pour la gestion et les problèmes de configuration, de disponibilité des ressources, et de dimensionnement des réseaux. La surveillance passive joue un rôle important dans la détection d'intrusions et de menaces. Cependant collecter et analyser les données du trafic circulant dans un réseau n'est pas chose aisée. Déployer les instruments de mesure et les beacons dans un réseau opérationnel est coûteux et prend du temps.

Dans tous les projets et études listés ci-dessus, l'objectif principal est la minimisation des coûts de gestion et de déploiement en terme de nombre d'instruments de mesure pour la surveillance passive ou de nombre de beacons actifs et de volume de trafic supplémentaire pour la surveillance active. Ainsi minimiser le nombre d'appareils et leur trouver des localisations stratégiques est une question essentielle, nécessaire au déploiement de plateformes de mesure à grande échelle. Ce type de problème d'optimisation a déjà été largement étudié [JJJ+00, BCG+01, BR03, HLO03, LTYP03, NT04, KK04, BDG04, CFL05, SGKT05].

Dans [SGKT05] des méthodes heuristiques pour le positionnement d'instruments de surveillance passive dans un réseau et le contrôle de leurs taux d'échantillonnage sont présentées. L'échantillonnage consiste pour un appareil à capturer seulement un pourcentage du trafic circulant sur le lien qu'il surveille, ce pourcentage ou *taux d'échantillonnage* peut être réglé indépendamment pour chaque appareil du réseau. Trois problèmes principaux sont considérés. Le premier consiste à maximi-

[1] http ://www.psc.edu/networking/papers/nimi.html
[2] `http://ipmon.sprintlabs.com/`

ser l'*utilité* du trafic capturé avec un budget d'installation d'équipement fixé, l'utilité du trafic capturé est une fonction définie dans [SGKT05]. L'objectif du second problème est la minimisation du coût d'installation permettant d'atteindre un niveau d'utilité fixé, alors que pour le troisième problème, on souhaite simplement minimiser la somme des coûts d'installation et d'exploitation des équipement sans fixer de niveau d'utilité. Le problème de la surveillance passive avec échantillonnage est également étudié. Une preuve de la NP-Complétude des problèmes est donnée en plus de méthodes heuristiques donnant des solutions pour chacun. Les performances de ces méthodes sont évaluées par des simulations pour plusieurs matrices de trafic et topologies de réseau fournies par le générateur Rocketfuel [SMW02].

Le problème de surveillance passive qui fait l'objet de [CDD+05] est celui que nous présentons en section 5.3. Sa complexité et ses propriétés d'approximabilité sont étudiées pour des topologies de réseaux particulières. Sur les chaines, les anneaux et les étoiles orientés, il est polynomial alors que sur les arbres il est NP-difficile et approximable à un facteur constant près. Nous verrons que dans le cas général il est équivalent au problème Minimum Partial Cover.

Le problème de la surveillance active est étudié en particulier dans [KK04] et [NT04]. L'objectif de [NT04] est la détection de pannes multiples. Le principe des solutions proposées est de déterminer dans un premier temps un ensemble de sondes permettant de surveiller chaque lien puis un ensemble de nœuds où des beacons pourront être placés afin d'émettre les sondes choisies. Ce travail fait suite à [BR03] qui s'intéressait à la détection d'une seule panne avec des méthodes similaires. Contrairement au cas de [BR03] où les routes dans le réseau sont figées, dans [KK04] les beacons placés doivent permettre la surveillance des liens quelles que soient les modifications des routes qui surviennent. Pour cela la phase de détermination des sondes est modifiée. Pour chaque nœud du réseau l'ensemble des liens qui pourront être surveillés par une sonde quelles que soient les routes ip est calculé, l'ensemble des beacons est choisi ensuite. L'objectif est de minimiser le nombre de beacons placés comme dans [HLO03].

La majorité des problèmes de placement de beacons ou d'appareils pour les mesures passives sont très proches des problèmes Minimum Partial Cover et Minimum Set Cover. Des preuves de complexité et des algorithmes d'approximation basés sur ceux connus pour Minimum Set Cover sont proposés dans la majorité des articles et dans ce chapitre. Ces travaux ont cependant été menés indépendamment les uns des autres.

5.3 Surveillance passive

Dans cette section nous considérons la surveillance passive. Comme mentionné précédemment, la surveillance passive n'introduit pas de trafic supplémentaire dans le réseau. Par contre les équipements de surveillance peuvent être fort coûteux en raison des capacités de traitement et de stockage de données requises. Il est donc très important de minimiser dans le réseau le nombre de ces équipements à installer, que nous appellerons *sondes* dans la suite. D'autre part des études [CFL04, DLT05] ont montré qu'il n'est pas nécessaire de surveiller la totalité du trafic mais seulement un certain pourcentage pour obtenir des informations exploitables.

Dans la suite nous présentons une modélisation du problème dont nous pourrons déduire des résultats de complexité et d'approximabilité ainsi qu'une formulation en programme linéaire mixte.

5.3.1 Modèle de réseau

Avant de formaliser le problème, nous décrivons le modèle de réseau utilisé. Le réseau est représenté par un graphe $G = (V, E)$ où V est l'ensemble des nœuds du réseau et E l'ensemble des

liens interconnectant les nœuds.

Un trafic t dans ce réseau est un chemin simple p_t entre deux nœuds, ou sommets de V, dont le poids est la bande passante utilisée par les données transférées sur ce chemin depuis sa source jusqu'à sa destination. Nous appelons *charge* d'un lien la somme des poids des trafics empruntant ce lien.

Nous considérons dans un premier temps qu'une sonde installée sur un lien e permet de surveiller la totalité du trafic empruntant le lien e. Ainsi surveiller une proportion k ($0 < k \leq 1$) du trafic circulant dans le réseau consiste à sélectionner un sous ensemble des liens où une sonde doit être installée en sorte que suffisamment de trafic soit surveillé.

Le problème de la surveillance passive partielle du trafic (*Partial Passive Monitoring* ou $PPM(k)$) est de trouver un tel ensemble de liens de taille minimum.

Problème 5.1 ($PPM(k)$)

Entrée : *Un graphe $G = (V, E)$, un ensemble $D = \{p_i, v_i\}$ de chemins pondérés dans G et une constante $k \in]0, 1]$. On note $\mathcal{V} = \sum_i v_i$ le volume de données total circulant dans le réseau.*

Sortie : *Un sous ensemble $E' \subseteq E$ d'arêtes tel que $\sum_{i|\exists e \in E', e \in p_i} v_i \geq k\mathcal{V}$, c'est à dire tel que la somme des poids des trafics circulant sur ces liens est supérieure à k pour cent du trafic total circulant dans le réseau.*

Objectif : *Cardinalité de E'.*

Notons que $PPM(1)$ consiste à surveiller la totalité du trafic du réseau, ce problème est appelé *Passive Monitoring*.

5.3.2 $PPM(k)$ et Minimum Partial Cover

Dans cette section nous montrons que le problème $PPM(k)$ est équivalent au problème Minimum Partial Cover (annexe A) pour tout $k \in]0, 1]$. Cette équivalence implique la NP-difficulté et des propriétés d'approximabilité et d'inapproximabilité pour $PPM(k)$.

Équivalence et complexité Nous considérons une instance de Minimum Partial Cover sur un ensemble d'éléments U à couvrir par des sous ensembles appartenant à la collection S de sous ensembles de U.

Intuitivement, les éléments de l'ensemble U représentent les trafics et les sous ensembles de S correspondent aux liens du réseau. L'objectif pour $PPM(k)$ est de minimiser le nombre de liens permettant de couvrir k pour cent des trafics, ce qui correspond bien à l'objectif de Minimum Partial Cover consistant à minimiser le nombre de sous ensembles permettant de couvrir au moins k pour cent des éléments.

Théorème 5.2 *Le problème $PPM(k)$ est équivalent au problème* Minimum Partial Cover.

Preuve: Dans un premier temps nous construisons une instance de $PPM(k)$ à partir d'une instance quelconque de Minimum Partial Cover pour laquelle $k\%$ des éléments de U doivent être couverts, comme illustré par la figure 5.1. Soit G un graphe dont l'ensemble d'arêtes E est défini comme suit :

- E contient une arête e_i pour chaque $s_i \in S$.
- si $s_i \cap s_j \neq \emptyset$, E contient une arête e_{ij} et une arête e_{ji} qui sont à la fois adjacentes à e_i et e_j en sorte que ces quatre arêtes forment un cycle.

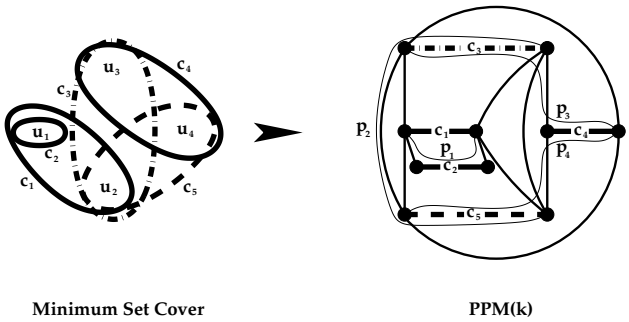


FIG. 5.1 – Exemple d'instances de MINIMUM PARTIAL COVER et PPM(k) équivalentes

Notons que seulement $2|S|$ sommets sont nécessaires pour définir E et ainsi G.

Ensuite l'ensemble de trafics, D, contient un trafic t_i pour chaque élément u_i de S. Le chemin p_i associé à t_i traverse l'arête e_j si et seulement si u_i appartient à s_j. De plus p_i peut utiliser toute arête e_{jk} tant qu'il utilise aussi e_j et e_k. De tels chemins peuvent toujours être trouvés[3] en temps polynomial par construction de G. Enfin chaque trafic t_i est de même poids que l'élément u_i associé, c'est à dire 1.

Supposons maintenant que E' est une solution optimale de l'instance de $PPM(k)$ ainsi construite. Une solution optimale S' pour l'instance de MINIMUM PARTIAL COVER peut en être déduite :

- si $e_i \in E'$ alors $s_i \in S'$.
- si $e_{ij} \in E'$ alors ni e_i ni e_j n'appartiennent à E'. En effet tous les trafics empruntant e_{ij} empruntent nécessairement e_i et e_j par construction, donc placer une sonde sur e_{ij} alors qu'une est placée sur e_i (ou e_j) est redondant et contredit l'optimalité de E'. Ainsi e_{ij} peut être remplacée arbitrairement soit par e_i soit par e_j, c'est à dire $c_i \in C'$ ou $c_j \in C'$.

E' permet de couvrir au moins $k\%$ des trafics, par construction S' aussi, puisque chaque élément correspond à un trafic et vice versa. De plus, E' est de cardinalité minimum ce qui implique la même propriété pour S' qui est par conséquent optimal pour cette instance de MINIMUM PARTIAL COVER.

D'autre part, une solution optimale S' de l'instance de MINIMUM PARTIAL COVER induit une solution optimale E' de l'instance de $PPM(k)$ construite. Pour chaque sous ensemble s_j appartenant à S', l'arête e_j correspondante à s_j appartient à E'. Puisque $k\%$ des éléments de U sont couverts par la sous collection S', et que chaque élément $u_i \in s_j$ correspond à un trafic t_i circulant sur l'arête e_j, $k\%$ des trafics sont surveillés grâce aux sondes placées sur les arêtes de E'. E' est nécessairement optimale pour $PPM(k)$ sinon une solution meilleure que S' pourrait être construite à partir de E', contredisant l'optimalité de S'.

Construisons maintenant une instance de MINIMUM PARTIAL COVER à partir d'une instance de $PPM(k)$ pour $k \in]0, 1]$ dans un graphe $G = (V, E)$ avec un ensemble D de trafics. Chaque arête e de G appartient à un ensemble π_e de chemins associés aux trafics de D. Installer une sonde sur e signifie que chaque trafic t_i de chemin $p_i \in \pi_e \subseteq D$ est surveillé. L'instance de MINIMUM PARTIAL COVER est alors définie par $U = D$ et $S = \{\pi_e,\ e \in E\}$. Soit E' une solution de l'instance de $PPM(k)$, une solution S' de MINIMUM PARTIAL COVER telle que $|S'| = |E'|$ est obtenue simplement par $S' = \{\pi_e \in D | e \in E'\}$. De même si S' est une solution de MINIMUM PARTIAL COVER, une solution

[3]Si u_i appartient à s_j et s_k, p_i doit emprunter e_j et e_k. Ordonnons arbitrairement toutes les arêtes que doit emprunter le chemin p_i, par exemple e_k succède à e_j. Par construction il existe une arête e_{jk} connectant les deux arêtes consécutives e_j et e_k. Il sera donc toujours possible de trouver un chemin associé à chaque u_i qui n'utilise que les arêtes associées aux ensembles contenant u_i.

E' de $PPM(k)$ vérifiant $|E'| = |S'|$ est définie trivialement par $E' = \{e \in E | \pi_e \in S'\}$.

Les deux problèmes $PPM(k)$ et MINIMUM PARTIAL COVER sont donc équivalents. ∎

Étant donnée l'équivalence des problèmes, les résultats de complexité et d'approximabilité connus pour MINIMUM PARTIAL COVER restent vrais pour $PPM(k)$.

Corollaire 5.1 *Le problème $PPM(k)$ pour une constante $k \in]0,1]$ est NP-Difficile et approximable à un facteur $\ln k|D| - \ln\ln k|D| + o(1)$. Le problème $PPM(1)$ n'est pas approximable à un facteur $(1-\varepsilon)\ln|D|$ pour tout $\varepsilon > 0$ sauf si* NP $\subset$ TIME$(n^{\log\log n})$.

5.3.3 $PPM(k)$ et MINIMUM EDGE COST FLOW

Dans cette section nous modélisons le problème $PPM(k)$ en un problème de MINIMUM EDGE COST FLOW (annexe A) dans un graphe particulier. Cette modélisation classique des problèmes de couverture nous a permis de voir le problème sous un angle légèrement différent et de le formuler en un programme plus efficace que ceux précédemment utilisés [CFL04, SGKT05].

Le problème MINIMUM EDGE COST FLOW est un problème de flot classique entre deux sommets dans un graphe orienté à l'exception des coûts des arêtes qui sont binaires, le coût d'utiliser un arc est constant quelle que soit la quantité de flot qui y circule tant qu'elle est strictement positive, par contre le coût est 0 si le flot sur cet arc est nul.

Problème 5.3 (MINIMUM EDGE COST FLOW**)**

Entrée : *$G' = (W, A)$ un graphe orienté dont chaque arc a est de capacité u_a et possède un coût d'utilisation fixe c_a, une requête de taille d entre une source $S \in W$ et un puits $T \in W$.*

Sortie : *un st-flot f satisfaisant la requête.*

Objectif : *le coût du flot f : la somme des coûts des arc de flot non nul, $\sum_{a\in A|f_a>0} c_a$.*

A partir d'un instance quelconque de $PPM(k)$, pour $0 < k \leq 1$, nous pouvons maintenant construire une instance de MINIMUM EDGE COST FLOW associée. Cette construction est illustrée à la figure 5.3.3.

Commençons par définir le graphe $G' = (W, A)$

1. W contient un sommet w_e par arête $e \in E$,
2. W contient un sommet w_t pour chaque trafic $t \in D$,
3. W contient deux sommets supplémentaires S et T,
4. il existe un arc de capacité non bornée et de coût 1 dans A de S vers chaque w_e, ainsi chaque arc Sw_e correspond à une arête e de l'instance de surveillance,
5. il existe un arc dans A de w_e vers w_t si et seulement si le chemin p_t associé au trafic t utilise l'arête e dans G. La capacité de tels arcs n'est pas bornée et son coût est nul,
6. il existe un arc de capacité v_t, le volume du trafic t, et de coût nul dans A de chaque w_t vers T.

L'objectif est de faire circuler de S vers T une quantité de flot égale au volume de trafic devant être surveillé soit $k\sum_{t\in D} v_t$.

Dans le graphe G', les arcs w_e qui portent un flot non nul correspondent pour l'instance de $PPM(k)$ aux arêtes sur lesquelles il faut installer un appareil de mesure.

Proposition 5.1 *Une solution optimale du problème* MINIMUM EDGE COST FLOW *dans G' induit une solution optimale pour $PPM(k)$ dans G.*

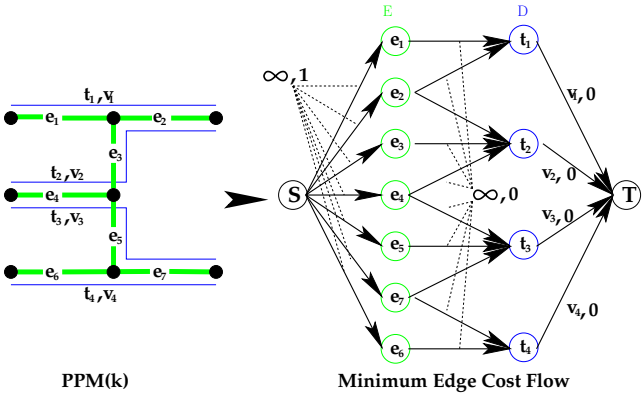


FIG. 5.2 – Instance de MINIMUM EDGE COST FLOW construite à partir d'une instance de $PPM(k)$

Preuve: Considérons un flot f solution de l'instance de MINIMUM EDGE COST FLOW construite à partir d'une instance de $PPM(k)$. Les seuls arc de G' de coût non nul sont les arcs (S, w_e), par conséquent le coût d'une solution est égal au nombre d'arcs (S, w_e) portant un flot non nul. L'ensemble des arêtes correspondant à ces arcs dans l'instance de $PPM(k)$ sera noté E' dans la suite.

Dans une solution au problème MINIMUM EDGE COST FLOW, le flot sur l'arc (w_t, T) peut provenir de plusieurs arcs (w_e, w_t), c'est à dire que le trafic t peut être partagé et donc surveillé partiellement par plusieurs appareils sur des arêtes différentes. Or ce procédé n'est pas prévu dans le problème de surveillance passive que nous étudions ici, puisqu'un appareil de mesure surveille la totalité du trafic qui circule sur le lien où il est installé. Cependant nous pouvons supposer que le partage du flot correspondant à un trafic ne se produit jamais dans G', il suffit en effet de déporter tout ce flot sur un même chemin, ce qui est toujours possible car les capacités des arcs concernés ne sont pas bornées.

De plus, le volume correspondant au trafic t pris en compte ne peut pas excéder v_t dans l'instance de MINIMUM EDGE COST FLOW étant donné que la capacité de l'arc (w_t, T) est égale à v_t.

Enfin, le volume total du flot traversant les sommets $w_e \quad \forall e \in E'$ est supérieur à la demande $k \cdot \sum_{t \in D} v_t$ et doit traverser les arcs (w_t, T) atteignables depuis ces sommets w_e, c'est à dire les arcs (w_t, T) correspondant aux trafics utilisant les arêtes $e \in E'$. E' est par conséquent une solution du problème de surveillance et le volume de flot circulant à la fois à travers w_e et w_t représente le volume du trafic t que l'appareil de surveillance installé sur e doit surveiller.

De plus si E^* est l'ensemble d'arêtes de G correspondant à une solution optimale du problème MINIMUM EDGE COST FLOW, il s'agit aussi d'une solution optimale de l'instance de $PPM(k)$. Dans le cas contraire, soit E'' une solution optimale du problème $PPM(k)$, alors $|E''| < |E^*|$ car toute solution de l'instance de MINIMUM EDGE COST FLOW représente une solution de $PPM(k)$, mais E^* n'est pas optimale pour $PPM(k)$.

D'autre part, une solution de MINIMUM EDGE COST FLOW peut être construite à partir de E" de la façon suivante. Premièrement notons qu'un unique chemin p_t^e traverse à la fois w_e et w_t. Pour chaque arête $e \in E''$ nous ajoutons un flot de volume v_t sur le chemin p_t^e si le chemin p_t utilise l'arête e dans G et si t n'a pas déjà été traité avec une autre arête de E''. Comme un trafic t n'est traité qu'une seule fois la contrainte de capacité sur l'arc (w_t, T) est respectée dans l'instance de MINIMUM EDGE COST FLOW et la valeur du flot est au moins $k \sum_{t \in D} v_t$ puisque le volume du trafic surveillé est supérieur à ce volume. Ce flot est donc une solution du problème MINIMUM EDGE COST FLOW de coût $|E''| < |E^*|$ ce qui contredit l'optimalité de E^*. ∎

Formulation en programme linéaire mixte Il existe deux formulations classiques des problèmes de flot en programme linéaire, la formulation arc-chemin et la formulation sommet-arc. Le programme 5.1 est la formulation arc-chemin à laquelle sont ajoutées des variables binaires (x_e) indiquant si le flot sur un arc (S, w_e) est nul ou non. Les contraintes permettant de fixer la valeur de ces variables sont également ajoutées.

Programme Linéaire 3 (PPM(k))

$$
\begin{array}{rrcl}
\textit{Minimiser} & \sum_{e\in E} x_e & & \\
t.q. & \sum_{t\in\pi_e} {f_t}^e & \leq & x_e \sum_{t\in\pi_e} v_t \quad \forall e \in E \\
& \sum_{e\in p_t} {f_t}^e & \leq & v_t \quad t \in D \\
& \sum_{t\in D}\sum_{e\in p_t} {f_t}^e & \geq & k\sum_{t\in D} v_t \\
& {f_t}^e & \geq & 0 \quad \forall e \in E \ \ \forall t \in \pi_e \\
& x_e & \in & \{0,1\} \quad \forall e \in E
\end{array}
$$

– ${f_t}^e$: quantité de flot circulant à la fois à travers w_e et w_t $\forall e \in E \, \forall t \in \pi_e$,
– x_e : 0 si le flot est nul sur l'arc (S, w_e), 1 sinon,

La première contrainte implique que le flot total circulant sur les chemins traversant un sommet w_e est nul si l'utilisation de l'arc (S, w_e) n'a pas été payée. La seconde contrainte assure que la capacité v_t de chaque arc (w_t, T) n'est pas violée. La troisième contrainte impose qu'une quantité de flot au moins supérieure à la requête $k\sum_{t\in D} v_t$ circule entre S et T. La fonction de coût est simplement le nombre d'arcs (S, w_e) portant un flot non nul.

Après avoir effectué des tests, nous avons constaté que cette formulation n'est pas plus rapide à résoudre que la formulation de [CFL04]. Elle comporte beaucoup plus de variables et de contraintes. De plus la transformation de l'instance initiale vers l'instance de flot correspondante prend un peu de temps.

Cependant, cette formulation nous a suggéré que les contraintes d'intégrité de certaines variables de la formulation de [CFL04] pouvaient être relâchées. Le programme 4 est la relaxation que nous avons obtenue à partir de cette formulation.

Programme Linéaire 4 (PPM(k))

$$
\begin{array}{rrcl}
\textit{Minimiser} & \sum_{e\in E} x_e & & \\
t.q. & \sum_{e\in p_t} x_e & \geq & \delta_t \quad \forall t \in D \\
& \sum_{t\in D} \delta_t \cdot v_t & \geq & k\sum_{t\in D} v_t \\
& \delta_t & \in & [0,1] \quad \forall t \in D \\
& x_e & \in & \{0,1\} \quad \forall e \in E
\end{array}
$$

– x_e est égale à 1 si un appareil de mesure doit être installé sur e, 0 sinon,
– δ_t est le pourcentage du volume de trafic t surveillé.

Dans cette formulation le volume surveillé d'un trafic n'est pas tout ou rien du fait de la relaxation des variables δ_t. Ceci n'est pas en contradiction avec la définition du problème qui précise que chaque trafic est associé à un *unique chemin* et qu'un instrument de mesure placé sur un lien surveille la *totalité* du trafic qui y circule. En effet une fois qu'un instrument est installé sur un lien déterminé grâce aux programmes précédents, si la solution du programme prévoit que seul un pourcentage du trafic t sera surveillé c'est que le trafic t emprunte un des liens surveillés, et donc le trafic t peut être surveillé intégralement. Il n'y a aucune contrainte de capacité, ce qui compte est de savoir où doivent être placés les instruments pour pouvoir surveiller au moins un certain pourcentage du trafic. Si plus de trafic peut être surveillé par les appareils installés, ce n'est pas un problème.

En plus de contenir un nombre réduit de variables binaires, ces deux formulations permettent de calculer des solutions pour un problème un peu différent. Supposons qu'un ensemble d'instruments de mesure soit déjà installé dans un réseau et qu'un certain nombre d'instruments supplémentaires doivent être installés. Le nouveau problème est de placer les appareils supplémentaires afin de maximiser le volume de trafic surveillé sans déplacer les instruments déjà en place. Les variables x_e associées aux liens comportant déjà un appareil de mesure sont alors fixées à 1 et traitées comme des constantes et l'objectif devient de maximiser $\sum_{t \in D} \sum_{e \in p_t} {f_t}^e$, ce qui permet de modéliser le nouveau problème. On peut aussi, pour un trafic donné, chercher à ajouter un nombre minimum de nouveaux appareils à un ensemble déjà installé. Il suffit de fixer les variables correspondant aux instruments placés à 1, et de résoudre le programme 4.

D'autre part, par l'ajout d'une contrainte, le problème d'installer un nombre limité d'appareils peut également être modélisé.

5.3.4 Simulation et résultats

Nous avons voulu comparer sur plusieurs topologies de POP les solutions obtenues par l'algorithme d'approximation du problème MINIMUM PARTIAL COVER et celles obtenues par le programme linéaire 4.

Nous avons utilisé pour cela des topologies fournies par l'outil Rocketfuel [SMW02] comme dans [CFL04, SGKT05].

Nous supposons comme dans [NT04] que le trafic à l'intérieur d'un POP est routé suivant le plus court chemin, du routeur par lequel il entre dans le POP au routeur par lequel il en sort. Contrairement à [BR03] nous ne faisons pas l'hypothèse qu'entre deux routeurs le routage est symétrique, c'est à dire que le chemin p_{uv} de u vers v n'utilise pas nécessairement les arcs inverses de ceux utilisés pour le chemin $p_v u$ de v vers u.

Comme nous ne disposons pas de matrices de trafic réelles associées aux topologies testées, nous avons généré aléatoirement plusieurs matrices. L'analyse présentée dans [BDJ01] indique que la répartition géographique du trafic à travers les POP est loin d'être uniforme. En particulier, ce comportement non uniforme provient de la manière dont est conçu Internet (certains POP reçoivent beaucoup plus de trafic que d'autres à cause de leur localisation géographique).

Pour se rapprocher des cas réels et ne pas générer de trafic uniforme entre tous les routeurs, nous choisissons au hasard des paires de routeurs entre lesquels le trafic sera plus important. La figure 5.3 présente un POP et la charge de trafic générée aléatoirement.

Tous les résultats présentés sont en fait une moyenne sur 20 simulations. Nous avons utilisé CPLEX pour résoudre les programmes linéaires mixtes.

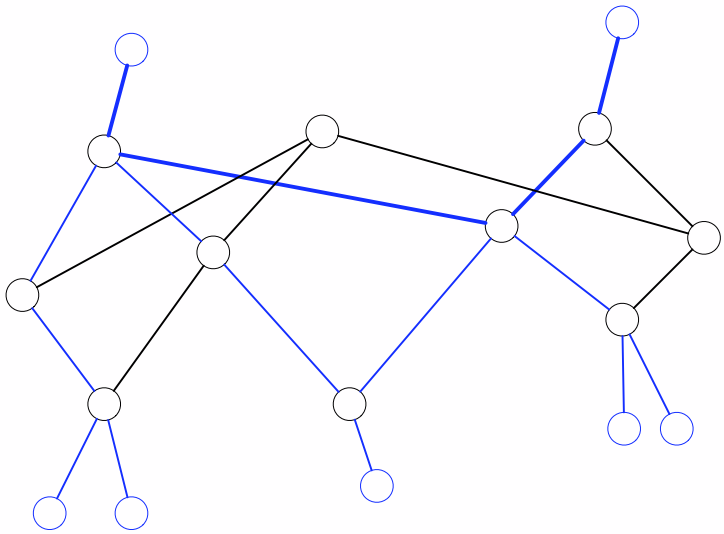


FIG. 5.3 – Charge des liens d'un POP. L'épaisseur d'une arête représente le pourcentage de trafic empruntant cette arête. Le trafic n'est pas uniforme.

La figure 5.4 synthétise les résultats du placement d'instruments de mesure dans un POP à 10 routeurs, 27 liens et 132 trafics pour l'algorithme d'approximation de MINIMUM PARTIAL COVER et le programme mixte. L'axe des abscisses indique le pourcentage de trafic qui est surveillé et démarre à 75%. L'axe des ordonnées donne le nombre d'appareils de mesure placés par les deux méthodes de calcul.

Premièrement nous constatons que jusqu'à 95% de trafic surveillé la courbe correspondant au programme linéaire mixte est presque linéaire. Une cassure apparaît à 95% de trafic surveillé, le nombre d'appareils requis double entre 95% et 100%. Ceci indique qu'il est plus économique de ne surveiller que 95% du trafic total, le coût de surveiller les 5% restant est très élevé par rapport au gain d'information supplémentaire. De plus l'algorithme d'approximation place en moyenne deux fois plus de trafic que la solution optimale fournie par notre formulation en programme mixte, ce qui est bien inférieur à $\ln k132 - \ln\ln k132$ pour $k > 0.1$.

La figure 5.5 montre les résultats obtenus sur un POP à 15 routeurs, 71 liens et 1980 trafics. On observe ici trois paliers. De 75% à 85% de trafic surveillé la croissance du nombre d'appareils est linéaire en fonction du pourcentage de trafic. De 85% à 95% la croissance est toujours linéaire mais plus rapide et finalement la courbe présente un accroissement important de 95% à 100%, passant de 16 appareils à 95% à 41 pour 100% de trafic surveillé. La conclusion est la même que pour le POP à 10 routeurs, il est beaucoup plus économique de ne surveiller que 95% du trafic.

Nous constatons également que l'algorithme d'approximation semble ne jamais trouver la solution optimale mais que pour 15 routeurs les solutions trouvées se rapprochent beaucoup plus de l'optimale que pour 10 routeurs.

5.4 Surveillance passive et échantillonnage

Dans les sections précédentes, les équipements de surveillance étaient considérés idéaux et capables d'enregistrer l'intégralité du trafic circulant sur un lien. En réalité ces équipements ne sont pas conçus pour analyser chaque paquet transitant sur un lien mais seulement un certain pourcen-

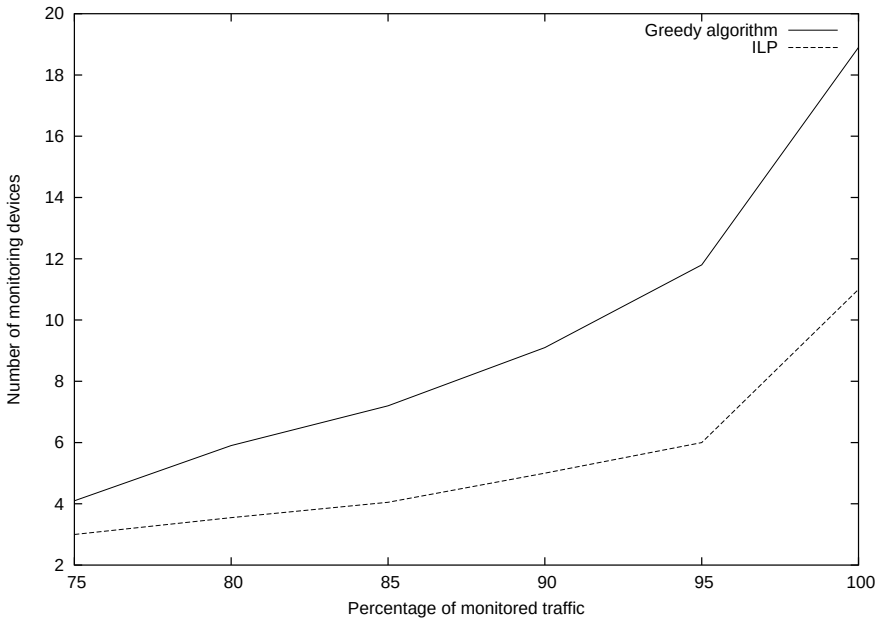


FIG. 5.4 – Surveillance passive : placement d'instruments sur un POP à 10 routeurs.

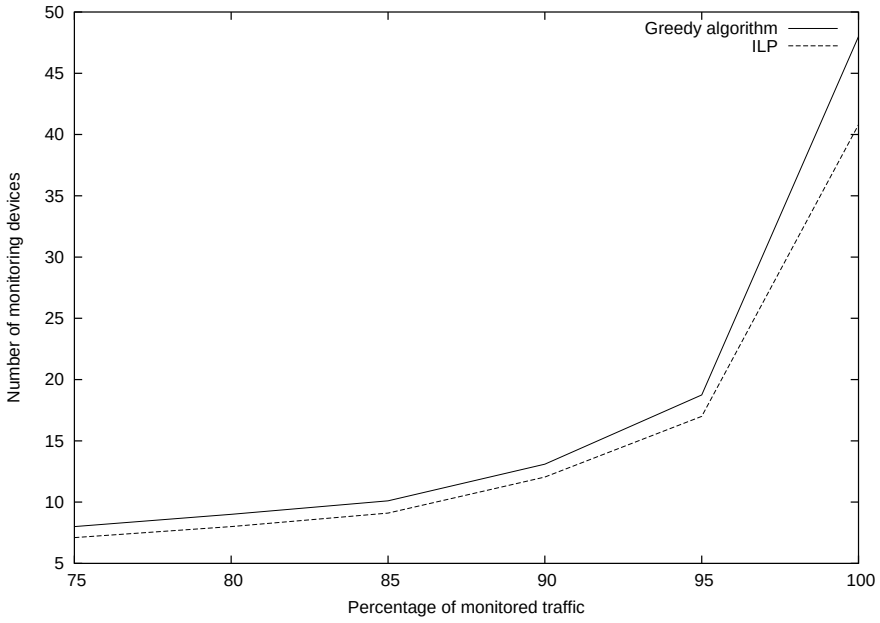


FIG. 5.5 – Surveillance passive : placement d'instruments sur un POP à 15 routeurs.

tage appelé *taux d'échantillonnage*. Étant donné le nombre vertigineux de paquets transitant sur un lien haut débit, comme une longueur d'onde ou une fibre, la nécessité de réduire le volume de données surveillées est parfaitement justifiée.

Réduire la quantité de paquets traitée et stockée contribue à réduire le *coût d'exploitation* des appareils de surveillance déployés dans le réseau. Le coût d'exploitation dépend du coût de traitement d'un paquet et du taux d'échantillonnage qui peuvent varier suivant les appareils. Lorsque l'échantillonnage est possible, la surveillance passive consiste à placer des appareils en sorte de surveiller au moins $k\,\%$ du trafic total, tout en minimisant le coût d'installation de l'ensemble des appareils ainsi que le coût d'exploitation induit par les taux d'échantillonnages affectés à chaque appareil.

Dans la suite nous considérons qu'un trafic t est l'agrégation de tous les trafics entrant dans le réseau en un nœud u et en sortant en un nœud v. Le trafic t suit entre les deux nœuds u et v le routage déterminé par la stratégie en œuvre dans le réseau, c'est à dire qu'il ne circule pas sur une route unique entre u et v comme dans les sections précédentes, mais peut emprunter tout un ensemble de chemins simultanément entre u et v suivant le modèle des flux de paquets défini en section 2.3.3. L'ensemble des chemins associés au trafic t entre u et v est noté $\mathcal{P}_{u,v}$ ou de manière équivalente $\mathcal{P}_t$, on note aussi $\mathcal{P} = \cup_t \mathcal{P}_t$.

Dans le cas où l'administrateur du réseau souhaite une vue de tous les trafics circulant sans pour autant surveiller chaque chemin, nous pouvons introduire le paramètre h_t, le pourcentage minimum à surveiller d'un trafic t. Notons que $h_t \leq k$ puisque h_t est relatif au seul trafic t alors que k concerne le trafic total du réseau.

5.4.1 Réduire la quantité de données

Les techniques permettant de diminuer la quantité de données traitée et stockée se divisent en trois classes principales.

- **Filtrage :** il consiste à capturer seulement un sous ensemble des flux suivant un critère particulier, comme le type de protocole, le numéro de port etc [ZMD+05].
- **Classification :** les paquets aussi peuvent être classés, par exemple suivant leurs préfixes d'adresse, et seules certaines classes sont surveillées.
- **Échantillonnage** les paquets sont capturés (pseudo) aléatoirement. Plusieurs méthodes d'échantillonnage ont été étudiées [DLT05, DLT02, ZMD+05].

L'échantillonnage présente de nombreux avantages. Premièrement il ne nécessite que peu de calcul en comparaison avec les deux autres techniques, le filtrage et la classification. Ensuite, il ne nécessite aucune configuration et ainsi il est plus facilement adaptable aux évolutions du trafic et donc plus adapté à la détection de trafics malveillants.

5.4.2 Techniques d'échantillonnage

L'échantillonnage et d'une manière générale la réduction du volume de données traitées, soulève de nombreux problèmes. L'utilisation d'un sous ensemble des paquets pour le calcul des statistiques biaise les estimations et il n'est pas toujours facile ou même possible de déduire les caractéristiques du trafic original à partir des données échantillonnées. La manière d'effectuer l'échantillonnage a une grande influence sur les conclusions qu'il est possible de tirer des données réduites. Dans [Duf04], Duffield présente différentes méthodes d'échantillonnage et leurs avantages et inconvénients relatifs.

- **Échantillonnage temporel** l'appareil de surveillance capture des paquets à intervalles de temps réguliers. Cette technique est problématique avec des applications ayant des contraintes de temps et qui émettent des paquets régulièrement eux aussi. Sur les liens bas débit en

particulier, un risque existe de ne considérer qu'un sous ensemble des flux et de manquer d'importantes informations.
- **Échantillonnage régulier** l'appareil de surveillance capture exactement un paquet tout les N paquets. Cette technique présente de meilleurs résultats que la précédente car elle a plus de chance de capturer des paquets appartenant à un flux très bref (*burst*). Cependant les résultats sont aussi influencés par les trafics périodiques.
- **Échantillonnage probabiliste** l'appareil capture les paquets avec une probabilité $1/N$.
- **Échantillonnage probabiliste basé sur une distribution** l'appareil capture un paquet tout les X, X étant une variable aléatoire suivant une loi donnée (géométrique, exponentielle) d'espérance N.

Le projet français Metropolis[4] a étudié l'influence de l'échantillonnage sur la perception des flux dans un réseau. En considérant un paquet sur mille, l'utilisation du modèle classique de *souris* et d'*éléphants* pour la classification des flux, désignant respectivement des flux courts et longs, a mis en évidence certaines erreurs d'identification des flux par l'échantillonnage. Avec seulement un paquet sur mille il est en effet délicat de décider à quelle classe appartient un flux étant donné que la probabilité de capturer plus de deux paquets de chaque flux éléphant est faible. Quant aux flux les plus courants, les flux souris, la plupart ne seront pas échantillonné du tout, et les statistiques tirés des traces d'échantillonnage ont tendance à surestimer leur volume.

D'autres contributions [DLT03, MUK+04] étudient le problème d'améliorer l'estimation des caractéristiques du trafic à partir de traces échantillonnées.[MUK+04] étudie plus particulièrement le problème de l'identification des flux éléphants avec un échantillonnage périodique. Il utilise le théorème de Bayes pour estimer la probabilité qu'un flux représenté par plus de y paquets dans une trace échantillonnée soit en réalité composé de plus de x paquets dans la trace complète. [DLT03] propose de compter les paquets SYN identifiant le début de la majorité des connexions TCP dans le but d'estimer plus précisément le nombre de flux. A partir de cette estimation, il est plus facile de déduire des statistiques réelles de traces échantillonnées.

[SGKT05] étudie le problème de déterminer le positionnement optimal de sondes dans un réseau sous des contraintes de coût qui limitent en fait le nombre de sondes. Ils considère que si un même flux circule sur plusieurs liens chacun surveillé par une sonde, alors ce flux ne sera échantillonné qu'une seule fois.

On peut s'attendre à ce qu'échantillonner plusieurs fois un même flux par plusieurs sondes différentes permette d'obtenir des informations supplémentaires et des statistiques plus détaillées qu'avec une seule sonde.

5.4.3 Modèle pour la surveillance avec échantillonnage

Dans cette section, nous désignons le coût d'installation d'une sonde sur un lien e par $cost_i(e)$ et le coût d'exploitation de cette même sonde par $cost_e(e)$. Ces deux fonctions de coût peuvent être générales, sans impact sur la formulation en programme linéaire 5. Cependant le coût d'exploitation est en général une fonction croissante concave [SGKT05] qui permet de prendre en compte le facteur d'échelle. Notons également que le modèle de [SGKT05] est un programme non linéaire mixte alors que le suivant est un MILP qui peut être résolu beaucoup plus rapidement même si le problème qu'il modélise est NP-Difficile.

[4] `http://www.laas.fr/~owe/METROPOLIS/metropolis_eng.html`

Programme Linéaire 5 (PPME(h,k))

$$
\begin{array}{lrcll}
\textit{Minimiser} & \sum_{e\in E} & (cost_i(e)\cdot x_e + cost_e(e)\cdot r_e) & & \\
 & & \textit{coût d'installation et d'exploitation} & & \\
s.c. & \sum_{e\in p} r_e & \geq & \delta_p & \forall p \in \mathcal{P} \\
 & x_e & \geq & r_e & \forall e \in E \\
 & \sum_{p\in\mathcal{P}_t} \delta_p \cdot v_p & \geq & h \cdot \sum_{p\in\mathcal{P}_t} v_p & \text{pour tout trafic } t \\
 & \sum_{p\in\mathcal{P}} \delta_p \cdot v_p & \geq & k \cdot \sum_{p\in\mathcal{P}} v_p & \\
 & \delta_p, r_e & \in & [0,1] & \forall p \in \mathcal{P}, \forall e \in E \\
 & x_e & \in & \{0,1\} & \forall e \in E
\end{array}
$$

Dans le programme 5, la variable binaire x_e indique si une sonde est placée sur le lien e. La variable δ_p représente ici le volume de trafic échantillonné circulant sur le chemin p. Nous introduisons la variable r_e qui représente le taux d'échantillonnage de la sonde placée sur le lien e.

La première contrainte modélise simplement le fait qu'il est nécessaire d'installer une sonde sur un lien si du trafic doit être capturé sur ce lien. Les contraintes suivantes imposent qu'un pourcentage minimum h_t de chaque trafic t soit capturé et qu'au moins k pour cent du trafic total du réseau soit surveillé.

5.4.4 Trafic dynamique

Le programme linéaire mixte précédent offre une méthode pour minimiser les coûts d'installation et d'exploitation des sondes. Cependant le trafic qui circule dans un réseau peut évoluer. Une modification importante du trafic peut anéantir tous les efforts d'optimisation et sérieusement dégrader la qualité des informations collectées par les opérateurs. Or d'un point de vue matériel il n'est pas concevable de déplacer une sonde d'un lien vers un autre à chaque fluctuation du trafic. En revanche, il est envisageable de modifier les taux d'échantillonnage des sondes pour les adapter aux variations du trafic. Il suffit alors de trouver une solution au problème $PPME(h,k)$ lorsque tous les x_e sont connus puisque les sondes sont déjà installées. Ce problème est désigné par $PPME^*(x,h,k)$.

Le problème $PPME^*(x,h,k)$ peut se formuler par le programme linéaire 5 dans lequel tous les x_e sont des constantes. Toutes les variables binaires ont disparu et il est possible de résoudre en temps polynomial ce problème puisqu'il se formule en programme linéaire de nombre de contraintes et de variables polynomiaux. D'autre part, notons que ce problème peut s'exprimer comme un flot de coût minimum pour lequel des algorithmes polynomiaux efficaces n'utilisant pas la programmation linéaire sont connus.

Dans le cas où un opérateur souhaite maintenir un pourcentage minimum d'échantillonnage h_t pour chaque trafic t et un pourcentage global k sur le volume total de trafic surveillé, s'il est en mesure de définir un seuil de tolérance $T < k$ en dessous duquel la dégradation de la surveillance devient critique pour ses applications, une stratégie simple permet de maintenir les contraintes d'échantillonnage dans un réseau.

1. Tant que $\sum_{p\in\mathcal{P}} \delta_p \cdot v_p \geq T \cdot \sum_{p\in\mathcal{P}} v_p$, attendre ;

2. Dès que $\sum_{p\in\mathcal{P}} \delta_p \cdot v_p < T \cdot \sum_{p\in\mathcal{P}} v_p$, résoudre $PPME^*(x,h,k)$, mettre à jour les taux d'échantillonnage des sondes ;
3. Aller à 1.

La résolution du problème $PPME$ peut être considérée comme l'étape initiale lors de l'installation de la surveillance dans un réseau. Pour une phase initiale, le temps de calcul nécessaire pour obtenir une solution optimale n'est en général pas crucial. Cependant, une fois les sondes mises en place, le temps d'adaptation du dispositif aux fluctuations du trafic devient un facteur clef et savoir résoudre $PPME*$ rapidement permet de répondre à cette exigence.

5.5 Surveillance active

La surveillance active a reçu beaucoup plus d'attention que la surveillance passive dans la littérature. Si cette approche implique un supplément de trafic, elle permet des mesures différentes et importantes. En général l'objectif est de trouver le nombre minimum de beacons dont les paquets sondes permettent de couvrir tous les liens du réseau [BR03, HLO03]. Lorsque les beacons sont choisis, un ensemble minimum de paquets sonde à émettre doit être déterminé. Dans [NT04] une approche différente est proposée. Elle consiste à commencer avec un ensemble de beacons possibles, ensuite à calculer un ensemble optimal de paquets sonde et enfin à positionner les beacons en fonction des paquets à émettre. Ils montrent que le placement des beacons est NP-Difficile et utilisent un algorithme glouton pour cette phase, ils sélectionnent un beacon et suppriment tous les paquets qu'il émet et ainsi de suite.

5.5.1 Le problème

Pour étudier ce problème nous utilisons le modèle de réseau de [NT04], *i.e.* un graphe non orienté $G = (V, E)$ avec V correspondant à l'ensemble des nœuds du réseau et E représentant l'ensemble des liens connectant les nœuds. Un sous ensemble des nœuds $V_B \subseteq V$ du réseau peut accueillir un beacon. A partir de cet ensemble V_B les auteurs de [NT04] donnent un algorithme polynomial qui calcule le nombre optimal de sondes à émettre. Ensuite à partir de cet ensemble optimal de sondes, les beacons utiles sont sélectionnés. Dans cette section, nous proposons d'améliorer cette phase de sélection. Notons bien que pour la surveillance active les appareils de mesure sont placés sur les nœuds du réseau et non pas sur les liens comme pour la surveillance passive.

5.5.2 Méthodes de résolution

Le problème de placement des beacons peut se traduire en un programme linéaire en variables binaires (ILP pour *Integer Linear Programming*). Supposons que Φ soit l'ensemble de paquets sondes optimal à émettre obtenu avec l'algorithme de [NT04]. Chaque sonde $\varphi \in \Phi$ est identifiée par ses deux extrémités φ_u et φ_v, sachant qu'une sonde allant de φ_u à φ_v équivaut à une sonde allant de φ_v à φ_u. Le programme linéaire est le suivant :

$$\min \sum_{i=1}^{n} y_i$$

$$\text{s.c. } \forall i \in V \backslash V_B \; y_i = 0$$

$$et \; \forall \varphi \in \Phi, y_{\varphi_u} + y_{\varphi_v} \geq 1$$

$$\forall i \in V, y_i \in \{0,1\}$$

où $n = |V|$ est le nombre de nœuds du réseau et $y = (y_i)_{i \in V}$ est la variable représentant le placement des beacons, i.e. $y_i = 1$ si un beacon doit être placé au nœud i et $y_i = 0$ sinon.

La première contrainte assure qu'aucun beacon ne sera placé sur un nœud non autorisé, c'est à dire les nœuds qui ne sont pas dans V_B. La seconde contrainte impose que chaque sonde φ sera émise par un beacon placé. L'objectif est de minimiser le nombre de beacon à placer.

Algorithmes d'approximation Dans [NT04] un algorithme de facteur d'approximation 2 est présenté. Il consiste à choisir aléatoirement un par un des nœuds où placer des beacons jusqu'à ce que toutes les sondes soient couvertes par les beacons choisis.

Nous proposons une variante sur le choix des beacons, au lieu de choisir aléatoirement les nœuds, nous choisissons toujours en premier le nœud qui permet de couvrir le plus de sondes à la fois. Cette modification n'influe pas sur la preuve du facteur d'approximation, par conséquent il s'agit encore d'une 2-approximation.

Comme nous le verrons dans la section suivante, notre algorithme donne de meilleurs résultats que celui de [NT04] sur les instances testées.

5.5.3 Simulations et résultats

La topologie du réseau utilisée pour ces simulations est générée de la même façon que dans la section 5.3. Nous avons implémenté l'algorithme de [NT04] pour calculer les ensembles de sondes optimaux. A partir de cet ensemble Φ, nous calculons le placement des beacons grâce à l'algorithme de [NT04], à notre algorithme glouton et grâce au programme linéaire en variables binaires. Pour résoudre le programme linéaire nous avons utilisé CPLEX. Tous les résultats représentent une moyenne sur 20 simulations.

La figure 5.6 présente les résultats du placement de beacons sur un réseau à 15 nœuds. Nous comparons l'algorithme de [NT04] noté *Thiran* dans la figure, notre algorithme noté *Greedy* et la solution basée sur la formulation en ILP.

L'axe des abscisses représente la taille de V_B et l'axe des ordonnées donne le nombre de beacons placés. Nous constatons que notre solution gloutonne place toujours moins de beacons que l'algorithme de [NT04] et que l'écart entre les deux augmente avec le nombre de beacons possibles ($|V_B|$). Ceci peut s'expliquer facilement par le fait que lorsque $|V_B|$ est faible, il y a seulement peu de marge de placement, alors que pour une grande valeur de $|V_B|$ il y a plus de façons d'optimiser le placement, et dans ce cas de toutes manières le programme linéaire est très efficace.

Pour $|V_B| = 15$ notre algorithme donne des solutions dont le nombre de beacons placés est seulement la moitié de celui donné par l'algorithme de [NT04], et il est très proche de celui donné par l'ILP. Par exemple pour 8 beacons possibles, ils ne diffèrent que d'un beacon.

La figure 5.7 donne les résultats pour le placement de beacons dans un réseau à 29 nœuds. Ils sont semblables aux résultats obtenus avec 15 nœuds. La solution obtenue par l'ILP est proche des solutions gloutonnes jusqu'à 15 beacons possibles environ. Le plus grand écart entre l'ILP et l'algorithme de [NT04] est de 33% et est obtenu pour $|V_B| = 29$. Notre algorithme est très proche de l'ILP : ils diffèrent d'au plus 2 beacons placés pour $|V_B| = 15$.

La figure 5.8 présente les résultats dans un réseau à 80 nœuds. Une fois encore la même conclusion peut être déduite. Le nombre de beacons est également réduit de 33% lorsque notre algorithme est utilisé plutôt que celui de [NT04]. Notons que dans ce cas, la différence entre notre solution gloutonne et le programme linéaire est plus marquée que pour les autres réseaux. Avec 80 beacons possibles, la solution gloutonne place 7 beacons de plus.

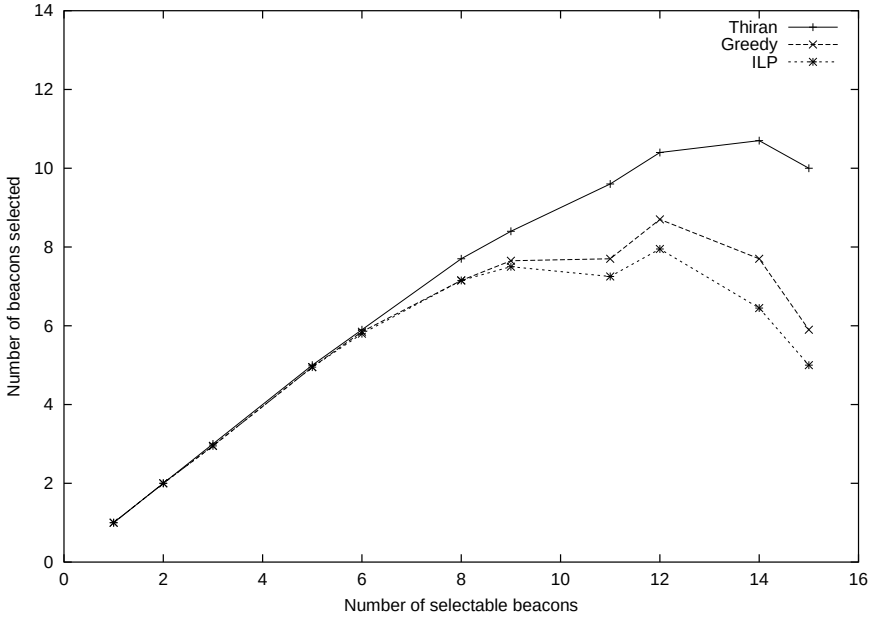


FIG. 5.6 – Surveillance active : placement de beacons dans un réseau de 15 nœuds

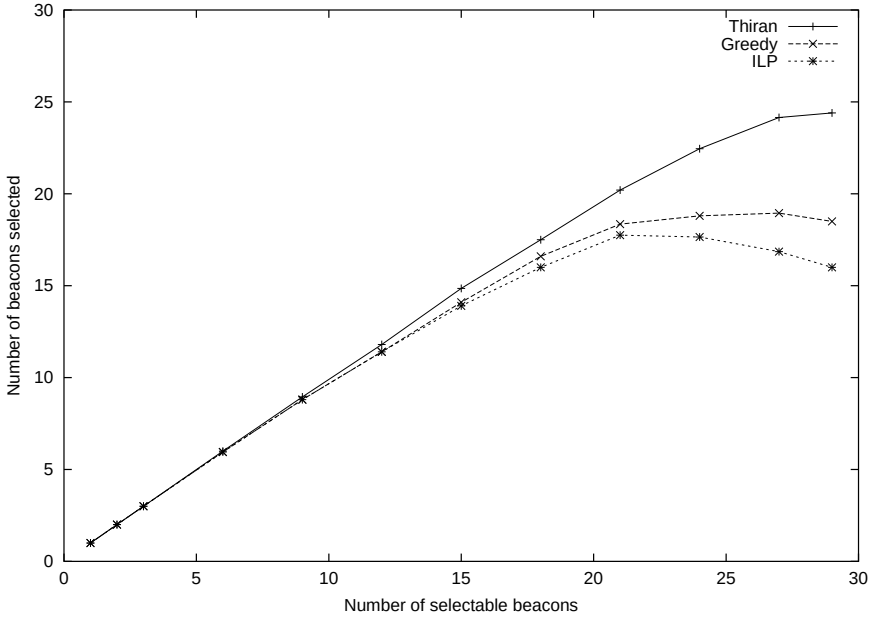


FIG. 5.7 – Surveillance active : placement des beacons dans un réseau à 29 nœuds.

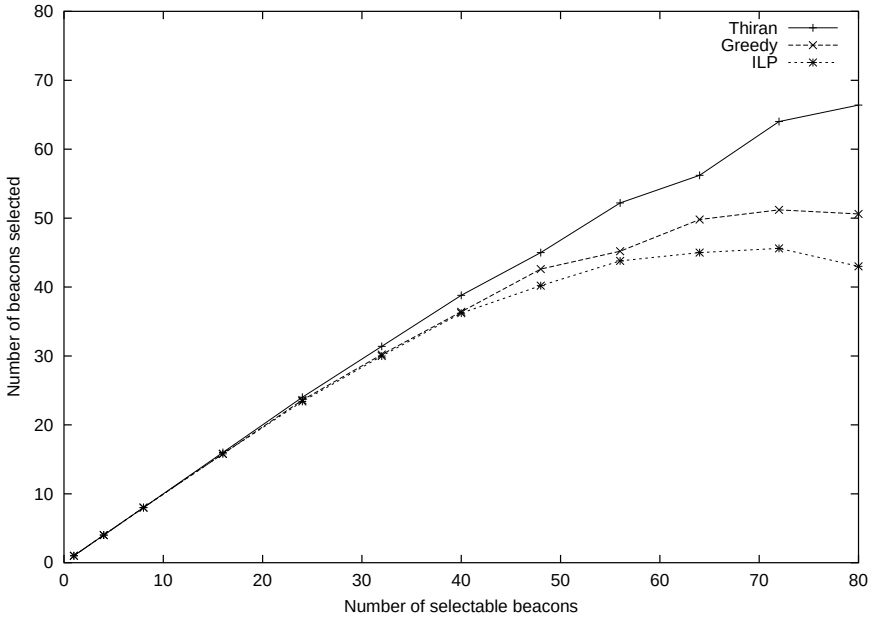


FIG. 5.8 – Surveillance active : placement des beacons dans un réseau à 80 nœuds.

Toutes ces courbes montrent que le nombre de beacons placés décroit à partir d'un certain seuil sur $|V_B|$ pour la solution obtenue avec l'ILP (ce qui est aussi le cas pour les autres solutions mais pas avec toutes les topologies). Avoir plus de choix pour placer les beacons permet de trouver de meilleures solutions. Par conséquent, il vaut mieux offrir le plus grand ensemble de nœuds possibles pour le placement des beacons.

5.6 Conclusion

Dans ce chapitre, nous nous sommes intéressé à divers problèmes de placement d'instruments de mesure pour la surveillance passive et active du trafic. Nous avons proposé une modélisation du problème de surveillance passive partielle en terme de flot et montré son équivalence avec le problème classique MINIMUM PARTIAL COVER. Cette modélisation nous a permis de mieux comprendre le cœur de la difficulté des problèmes de placement, mais aussi de donner une formulation en MILP améliorant celles de la littérature. Grâce à cette formulation, nous avons pu proposer également une méthode de résolution polynomiale et efficace pour gérer les évolutions de trafic. De plus, par l'ajout de contraintes simples à notre formulation en MILP, des problèmes légèrement différents peuvent être modélisés. Il s'agit de problèmes comme celui de trouver le meilleur placement pour des instruments supplémentaires dans un réseau déjà équipé, d'estimer le gain induit par l'ajout d'un ou plusieurs instruments ou encore de trouver le meilleur placement pour un ensemble de taille fixée d'appareils. Nous avons également proposé une formulation en MILP pour le placement d'instruments de mesure dans le cadre de la surveillance active, ainsi qu'un algorithme glouton améliorant celui de [NT04].

Trois pistes principales s'ouvrent pour poursuivre le travail sur les problèmes de placement d'appareils de mesures. Premièrement, nous devons affiner notre modèle d'instruments de mesure pour la surveillance avec échantillonnage pour améliorer les taux d'échantillonnage obtenus par plusieurs appareils observant un même flux en divers points du réseau. Ensuite nous devons modifier

nos formulations en MILP pour prendre en compte des trafics routés sur plusieurs chemins afin de réduire si possible le nombre de variables et de contraintes, et par suite le temps de calcul. En effet nous considérons dans ce chapitre qu'un trafic correspond à un unique chemin, or le routage avec équilibrage de charge utilisé dans certains réseaux peut conduire à diviser le trafic d'une même source vers une même destination sur plusieurs chemins. Enfin, lorsqu'un ensemble d'instruments de mesure est déjà installé dans un réseau, l'opérateur peut souhaiter modifier la stratégie de routage plutôt que l'emplacement des points de mesure afin de maximiser le trafic surveillé. Ceci peut donner lieu à plusieurs problèmes d'optimisation intéressants proches des problèmes de flots classiques.

Chapitre 6

Conclusion

Dans cette thèse nous avons étudié des problèmes d'optimisation et de décision issus des réseaux de télécommunication du point de vue de leur complexité et de leurs propriétés d'approximabilité, mais aussi de leur résolution pratique. Nous avons considéré aussi bien les réseaux d'accès que les réseaux de cœur multiniveaux de type IP/WDM utilisant une architecture MPLS. Nous avons abordé trois problématiques différentes : la conception de réseaux virtuels, les propriétés de connexité et de vulnérabilité aux pannes d'un réseau multiniveaux donné, et enfin le placement d'instruments de mesure du trafic dans un réseau d'accès.

Le premier chapitre présente les réseaux que nous avons considérés, leurs modélisations par des graphes, ainsi que les principes de tolérance aux pannes existant dans la littérature. Dans le second chapitre, nous avons formulé un problème de conception de réseau virtuel tolérant aux pannes et nous avons proposé des méthodes de résolution pour le problème du groupage sur un chemin orienté qui en est dérivé. Le troisième chapitre est consacré à l'étude de la complexité et des propriétés d'approximabilité des problèmes d'optimisation qui se posent dans le contexte de la tolérance aux pannes des réseaux multiniveaux. Ces problèmes se divisent en problèmes de connexité et de vulnérabilité. Les problèmes de connexité consistent à trouver des chemins ayant différentes propriétés (chemins disjoints, plus court chemins etc) entre des paires de sommets. Pour les problèmes de vulnérabilité, on recherche des ensembles de ressources dont la suppression déconnecte des ensembles de sommets (coupes, st-coupe). Pour étudier ces problèmes, nous avons modélisé les réseaux multiniveaux par des graphes colorés. Les problèmes de placement d'instruments de mesure du trafic dans les réseaux d'accès font l'objet du quatrième et dernier chapitre. Nous abordons la surveillance passive avec et sans échantillonnage, mais aussi les problèmes liés à la surveillance active. Nous montrons que ces problèmes de placement sont en fait des problèmes de couverture.

Si la tolérance aux pannes n'est qu'un domaine d'application des mesures de trafic, elle est au cœur de l'étude que nous avons menée sur les graphes colorés qui représentent les réseaux multiniveaux. Dans cette thèse nous nous sommes concentrés sur les problèmes d'optimisation les plus fondamentaux dans ces graphes (chemin, coupe, arbre couvrant etc) qui sont les briques de base de la plupart des méthodes de protection. Il faut maintenant étendre ces travaux à des notions plus complexes et plus globales.

Les opérateurs ne peuvent en effet se contenter de calculer des chemins risque disjoints dans leurs réseaux, ils doivent aussi tenir compte de contraintes de capacité, de qualité de service etc. C'est pourquoi le problème du multiflot dans les graphes colorés, avec ses diverses variantes (type des requêtes, contraintes de chemins disjoints, contraintes de longueur des chemins etc), constitue l'un des problèmes à étudier en priorité. Dans le cadre des réseaux à un seul niveau, comme les réseaux WDM très largement étudiés, les problèmes liés au routage et à la protection sont effectivement

traités en majorité par des variantes du multiflot.

En outre, le multiflot coloré pourrait être un outil d'une importance significative pour traiter les problèmes de groupage et de conception de réseaux virtuels tolérants aux pannes. Les problèmes que nous avons déjà abordés sont des outils qui pourraient être utilisés pour améliorer les algorithmes de groupage en mettant en évidence les points faibles des solutions vis à vis de la tolérance aux pannes. Cependant, les informations que pourra apporter la résolution d'un multiflot coloré seront plus complètes et en particulier tiendront compte des questions de capacité.

Le travail sur le multiflot coloré devra commencer par la définition précise de ce problème, des objectifs et des contraintes à prendre en compte. L'exploration des techniques à mettre en œuvre et des modélisations à adopter pour traiter cette question pourra ensuite débuter.

Bibliographie

[ABD+01] O. Audouin, C. Blaizot, E. Dotaro, M. Vigoureux, B. Beauquier, J.-C. Bermond, B. Bongiovanni, S. Pérennes, M. Syska, S. Bibas, L. Chacon, B. Decocq, E. Didelet, A. Laugier, A. Lisser, A. Ouorou, and F. Tillerot. Planification et optimisation des réseaux de transport optiques. Rapport final RNRT PORTO, Alcatel Research & Innovation, Projet MASCOTTE (CNRS/INRIA/UNSA) et France Télécom R&D, Sophia Antipolis, December 2001.

[AdC03] F. Alvelos and J.M. Valério de Carvalho. comparing branch-and-price algorithms for the unsplittable multicommodity flow problem. In Walid Ben-Ameur and Alain Petrowski, editors, *International Network Optimization Conference*, pages 7–12. Institut National des Télécommunications, October 2003.

[ADP80] G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1) :136–153, 1980.

[aKS01] K. Lee ang K. Siu. An algorithmic framework for protection switching in WDM networks. In *NFOEC'01*, pages 402–410, Baltimore, July 2001.

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In 33^{rd} *Annual IEEE Symposium on Foundations of Computer Science*, pages 14–23, 1992.

[AR01] Y. Azar and O. Regev. Strongly polynomial algorithms for the unsplittable flow problem. In *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 15–29. Springer-Verlag, 2001.

[Asa00] Y. Asano. Experimental evaluation of approximation algorithms for the minimum cost multiple-source unsplittable flow problem. In *ICALP workshop*, pages 111–121, 2000.

[Bar96] F. Barahona. Network design using cut inequalities. *SIAM Journal on optimization*, 6 :823–837, 1996.

[BCC+05] 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.

[BCCP06] 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.

[BCG+01] Y. Breitbart, C.-Y. Chan, M.N. Garofalakis, R. Rastogi, and A. Silberschatz. Efficiently monitoring bandwidth and latency in IP networks. In *INFOCOM*, pages 933–942, 2001.

[BCJ+97] M. Berger, M. Chbat, A. Jourdan, M. Sotom, P. Demeester, B. Van Caenegem, P. Gødsvang, B. Hein, M. Huber, R. März, A. Leclert, T. Olsen, G. Tobolka, and T. Van den Broeck. Pan-european optical networking using wavelength division multiplexing. *IEEE Communications Magazine*, 35(4) :82–88, April 1997.

[BCLR04] C. Bentz, M.-C. Costa, L. Létocart, and F. Roupin. A bibliography on multicut and integer multiflow problems. Technical Report 654, CeDRIC Centre de Recherche en Informatique du Cnam, http ://cedric.cnam.fr/AfficheMembre.php ?id=30, 2004.

[BCM03a] 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.

[BCM03b] J.-C. Bermond, D. Coudert, and X. Munoz. Traffic grooming in unidirectional WDM ring networks : the all-to-all unitary case. In *ONDM*, pages 1135–1153, 2003.

[BDG04] Y. Breitbart, F. Dragan, and H. Gobjuka. Effective network monitoring. In *13th International Conference on Computer Communications and Networks (ICCCN'04)*, pages 394–399, Chicago, Illinois, October 2004.

[BDJ01] S. Bhattacharyya, C. Diot, and J. Jetcheva. POP-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IMW)*, San Francisco, November 2001.

[BDPS03] J.-C. Bermond, O. DeRivoyre, S. Pérennes, and M. Syska. Groupage par tubes. In *Conference ALGOTEL2003, Banyuls, May 2003*, pages 169–174, 2003.

[Bha94] R. Bhandari. Optimal diverse routing in telecommunication fiber networks. In *IEEE INFOCOM '94*, volume 3, pages 1498–1508, Toronto, Ont., Canada, June 1994.

[Bha97] R. Bhandari. Optimal physical diversity algorithms and survivable networks. In *ISCC '97 : Proceedings of the 2nd IEEE Symposium on Computers and Communications (ISCC '97)*, page 433, Washington, DC, USA, 1997. IEEE Computer Society.

[Big06] W. C. Bigos. *Optimized Modeling and Design of Multilayer* IP *over Optical Transport Netwrok Architectures*. PhD thesis, Université de Rennes I, March 2006.

[BKO06] P. Belotti, A. Koster, and S. Orlowski. A cut-and-branch-and-price approach to two layer network design. In *INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[BKP03a] 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*, pages 61–66, Oct 2003.

[BKP03b] S. Beker, D. Kofman, and N. Puech. Off line reduced capacity layout design for MPLS networks. In *IEEE workshop on* IP *Operations and Management (IPOM)*, pages 99–105, Kansas City (USA), Oct 2003.

[BLE+02] E. Bouillet, J.F. Labourdette, G. Ellinas, R. Ramamurthy, and S. Chaudhuri. Stochastic approaches to compute shared mesh restored lightpaths in optical network architectures. In *IEEE INFOCOM*, volume 2, pages 801–807, 2002.

[BMN05] A. Balakrishnan, P. Mirchandani, and H.P. Natarajan. Connectivity upgrade models for survivable network design. In *McCombs Research Paper Series*, number IROM-02-06. `http://ssrn.com/abstract=876488`, June 2005.

[BP01] P. Barford and D. Plonka. Characteristics of network traffic flow anomalies. In *ACM SIGCOMM Internet Measurement Workshop*, 2001.

[BR03] Y. Bejerano and R. Rastogi. Robust Monitoring of Link Delays and Faults in IP Networks. In *Proceedings of IEEE Infocom*, 2003.

[CAQ04] X. Cao, V. Anand, and C. Qiao. Multi-layer versus single-layer optical cross-connect architectures for waveband switching. In *IEEE Infocom*, Hong Kong, China, March 2004.

[CAXQ03] X. Cao, V. Anand, Y. Xiong, and C. Qiao. Performance evaluation of wavelength band switching in multi-fiber all optical networks. In *IEEE Infocom*, San Francisco, California, USA, April 2003.

[CB00] O. Crochat and J.-Y. Le Boudec. Protection interoperability for WDM optical networks. *IEEE/ACM Transactions on Networking*, 2000.

[CCPS98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*, chapter Maximum Flow Problems (Multicommodity Flows p. 85). John Wiley, 1998.

[CDD+05] O. Cogis, B. Darties, S. Durand, J.-C. König, and J. Palaysi. Contrôle de routes par des appareils de surveillance (cras). In *7èmes Rencontres Francophones sur les Aspects ALGOrithmiques des TELécommunications (AlgoTel'05)*, Presqu'île de Giens, May 2005.

[CDKM00] R.D. Carr, S. Doddi, G. Konjevod, and M. Marathe. On the red-blue set cover problem. In *SODA '00 : Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 345–353, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[CDP+06] D. Coudert, P. Datta, S. Pérennes, H. Rivano, and M-E. Voge. Shared risk resource group : Complexity and approximability issues. *Parallel Processing Letters*, 2006. To appear.

[CFGL+05a] C. Chaudet, E. Fleury, I. Guérin-Lassous, H. Rivano, and M.-E. Voge. Optimal positioning of active and passive monitoring devices. In *CoNEXT 2005*, Toulouse, France, October 2005.

[CFGL+05b] C. Chaudet, E. Fleury, I. Guérin-Lassous, H. Rivano, and M.-E. Voge. Surveillance passive dans l'internet. In *Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05)*, pages 121–124, Presqu'île de Giens, May 2005.

[CFL04] C. Chaudet, E. Fleury, and I. Guérin Lassous. Optimal positioning of active and passive monitoring devices. Research Report 5273, INRIA, July 2004.

[CFL05] C. Chaudet, E. Fleury, and I. Guérin Lassous. Positionnement optimal de sondes pour la surveillance active et passive de réseaux. In *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP)*, Bordeaux, France, April 2005.

[Cha98] P. Chanas. *Réseaux ATM : Conception et optimisation.* PhD thesis, France Télécom CNET Sophia Antipolis, 1998.

[CHGL05] T. Cicic, A. F. Hansen, S. Gjessing, and O. Lysne. Applicability of resilient routing layers for k-fault network recovery. In *Proceedings of International Conference on Networking (ICN), Reunion, France April 17-21*, pages 173 – 183. Springer-Verlag GmbH, 2005. ISSN 0302-9743, ISBN 3-540-25339-4,.

[Cho02] S. Choplin. *Dimensionnement de réseaux virtuels de télécommunications*. PhD thesis, Université de Nice-Sophia Antipolis, 2002.

[CL97] R. Chang and S. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63 :277–282, 1997.

[CLR03] M.-C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximal integer multiflow : a survey. *EJOR Eur. J. on Oper. Res. To appear*, 2003.

[CPPS05] D. Coudert, S. Pérennes, Q.-C. Pham, and J.-S. Sereni. Rerouting requests in WDM networks. In *Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05)*, pages 17–20, Presqu'île de Giens, May 2005.

[CPRV06] D. Coudert, S. Pérennes, H. Rivano, and M.-E. Voge. Shared risk resource groups and survivability in multilayer networks. In *IEEE/COST 293 annual conference on GRAphs and ALgorithms in communication networks*, volume 3, pages 235–238, June 2006. Invited Paper.

[CSC02] H. Choi, S. Subramaniam, and H. Choi. On double-link failure recovery in WDM optical networks. In *IEEE Infocom*, pages 808–816, New-York, USA, June 2002.

[CSC04] H. Choi, S. Subramaniam, and H. Choi. Loopback recovery from double-link failures in optical mesh networks. *IEEE/ACM Trans. Netw.*, 12(6) :1119–1130, 2004.

[Dah91] G. Dahl. Contributions to the design of survivable directed networks. *Thesis presented to the university of OSLO*, 1991.

[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. ISBN : 0-387-94883-X.

[DG02] J. Doucette and W. D. Grover. Capacity design studies of span-restorable mesh transport networks with shared-risk link group (SRLG) effects. In *SPIE Opticomm*, 2002.

[DGA+99] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemente, M. Ravera, A. Jajszczyk, D. Janukowicz, K. Doorselaere, and Y. Harada. Resilience in multilayer networks. *IEEE Communications Magazine*, 37 :70–76, August 1999.

[DGM94] A. D. Dunn, W. D. Grover, and M. H. MacGregor. Comparison of k-shortest paths and maximum flow routing for network facility restoration. *IEEE Journal on Selected Areas of Communications*, 2(1) :88–99, January 1994.

[DLT02] N. Duffield, C. Lund, and M. Thorup. Properties and prediction of flow statistics from sampled packet streams. In *IMW '02 : Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 159–171, New York, NY, USA, 2002. ACM Press.

[DLT03] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. In *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Karlsruhe, Germany, October 2003.

[DLT05] N. Duffield, C. Lund, and M. Thorup. Learn more, sample less : Control of volume and variance in network measurement. *IEEE Transactions on Information Theory*, 51(5) :1756– 1775, May 2005.

[DR00] R. Dutta and G. N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1(1) :73–89, January 2000.

[DR02] R. Dutta and G. N. Rouskas. Traffic grooming in WDM networks : past and future. *IEEE Networks*, 16(6) :46–56, november/december 2002.

[DS04a] P. Datta and A.K. Somani. Diverse routing for shared risk resource groups (SRRG) failures in WDM optical networks. In *IEEE BroadNets*, 2004.

[DS04b] I. Dinur and S. Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5) :247–254, 2004.

[Duf04] N. Duffield. Sampling for passive internet measurement : a review. *Statistical Science*, 19(3), 2004.

[DW94] R. Doverspike and B. Wilson. Comparison of capacity efficiency of DCS network restoration routing techniques. *Journal of Network and System Management*, 2(2) :95–123, 1994.

[DY01] R. Doverspike and J. Yates. Challenges for MPLS in optical network restoration. *IEEE Communications Magazine*, feb :89–96, 2001.

[EBR+03] G. Ellinas, E. Bouillet, R. Ramamurthy, J.-F. Labourdette, S. Chaudhuri, and K. Bala. Routing and restoration architectures in mesh optical networks. *Optical Networks Magazine*, January 2003.

[EH02] T. Erlebach and A. Hall. Np-hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow. In *SODA '02 : Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 194–202, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.

[ES03] T. Erlebach and S.K. Stefanakos. On shortest-path all-optical networks without wavelength conversion requirements. In *STACS '03 : Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 133–144, London, UK, 2003. Springer-Verlag.

[Far06] A. Faragó. A graph theoretic model for complex network failure scenarios. In *INFORMS*, 2006.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4) :634–652, July 1998.

[GDK+06] W. D. Grover, J. Doucette, A. Kodian, D. Leung, A. Sack, M. Clouqueur, and G. Shen. *Handbook of Optimization in Telecommunications*, chapter Design of Survivable Networks Based on p-Cycles. Springer, 2006.

[Gef01] J. Geffard. A solving method for singly routed traffic in telecommunication networks. *Annales des Télécommunications*, 56(3-4) :140–149, 2001.

[GGL95] R. L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of combinatorics*, volume 1 ch.2 A. Frank. MIT Press, Cambridge, MA, USA, 1995.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability : A Guide to the theory of NP-completeness*. Freeman NY, 1979.

[GMS95] M. Grötschel, C.L. Monma, and M. Stoer. *Handbooks in Operations Research and Management Science*, volume 7 :Network Models. Elsevier publisher, 1995.

[GPS06] L. Gouveia, P. Patrício, and A. De Sousa. Hop-constrained node survivable network design : and application to MPLS over WDM. In *INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[GR00] J. Gruber and R. Ramaswami. Moving toward all-optical networks. *Lightwave Magazine*, pages 60–68, December 2000.

[Gro04] W.D. Grover. *Mesh Based Survivable Transport Networks : Options and Stratégies for optical,* MPLS, SONET *and* ATM *networking.* Prentice Hall PTR, 2004.

[GRW00] O. Gerstel, R. Ramaswami, and W. Wang. Making use of a two-stage multiplexing scheme in a WDM network. In *OSA/SPIE Optical Fiber Communication Conference and Exposition*, volume 3, pages 44–46, 2000.

[GS98] W. Grover and D. Stamatelakis. Cycle-oriented distributed preconfiguration : ring-like speed with mesh-like capacity for self-planning network restoration. In *IEEE Internation Conference on Communications*, volume 1, pages 537–543, 1998.

[GT00] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of IEEE Infocom.* IEEE, 2000.

[GZLK01] B. Doverspike G .Z. Li and C. Kalmanek. Fiber span failure protection in mesh optical networks. In *SPIE Opticomm*, volume 4559, pages 130–142, 2001.

[Has99] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182 :105–142, 1999.

[HDR06] 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.

[HJK+06] R. Hülsermann, M. Jäger, A. Koster, S. Orlowski, R. Wessäly, and A. Zymolka. Availability and cost based evaluation of demand-wise shared protection. In *7th ITG-Workshop on Photonic Networks*, pages 161–168, Leipzig, Germany, 2006.

[HLO03] J. D. Horton and A. Lopez-Ortiz. On the Number of Distributed Measurement Points for Network Tomography. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC)*, Miami Beach, USA, October 2003.

[Hoc97] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, 1997.

[HPS02] G. Huiban, S. Pérennes, and M. Syska. Traffic grooming in WDM networks with multi-layer switches. In *IEEE International Conference on Communications*, pages 2896–2901, New-York. USA, April 2002. Cdrom.

[HSKO99] K. Harada, K. Shimizu, T. Kudou, and T. Ozeki. Hierarchical optical path cross-connect systems for large scale WDM networks. In *IEEE Optical Fiber Communication*, pages 356–358, San Diego, USA, 1999.

[Hu03] J.Q. Hu. Diverse routing in mesh optical networks. *IEEE Transactions on Communications*, 51(3) :489–494, 2003.

[HV06] H. Höller and S. Voß. Heuristics for the multi-layer design of MPLS/SDH/WDM networks. In *INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[ICM+02] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an IP backbone. In *IMW '02 : Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 237–242, New York, NY, USA, 2002. ACM Press.

[IMG98] R.R. Iraschko, M.H. MacGregor, and W.D. Grover. Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks. *IEEE / ACM Transactions on Networking*, 6(3) :325–336, June 1998.

[Jau] B. Jaumard. Exemples de réseaux. `http://www.algorithmic-solutions.com/enleda.htm`.

[JID+04] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, Hong Kong, March 2004.

[JJJ+00] S. Jamin, C. Jin, Y. Jin, D. Raz, and L. Zhang. On the placement of internet instrumentation. In *Proceedings of IEEE Infocom*, Tel Aviv, Israel, March 2000.

[JMY05] B. Jaumard, C. Meyer, and Xiao Yu. When is wavelength conversion contributing to reducing the blocking rate ? In *IEEE Global Telecommunications Conference (GLOBECOM'05)*, volume 4, pages 2078 – 2083, 2005.

[JRN04] Q. Jiang, D.S. Reeves, and P. Ning. Improving robustness of PGP keyrings by conflict detection. In *RSA Conference Cryptographers' Track (CT-RSA2004)*, pages 194–207. LNCS 2964, February 2004.

[Kar72] R.M. Karp. *Complexity of Computer Computations*, chapter Reducibility Among Combinatorial Problems, pages 85–103. Plenum Press, 1972.

[Kar93] D.R. Karger. Global min-cuts in RNC and other ramifications of a simple mincut algorithm. In 4^{th} *ACM-SIAM Symposium on Discrete Algorithms*, 1993.

[KK99] J. Kleinberg and A. Kumar. Wavelength conversion in optical networks. In *SODA '99 : Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 566–575, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[KK04] R. Kumar and J. Kaur. Efficient beacon placement for network tomography. In *IMC '04 : Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 181–186, New York, NY, USA, 2004. ACM Press.

[KL03] M. Kodialam and T.V. Lakshman. Detecting Network Intrusions via Sampling : A Game Theoretic Approach. In *Proceedings of IEEE Infocom*, San Francisco, USA, March 2003. IEEE.

[Kle96] J. M. Kleinberg. Single-source unsplittable flow. *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–77, 1996.

[KM01] H. Kérivin and A.R. Mahjoub. On survivable network polyhedra. *submitted to Discrete Mathematics*, 2001.

[KP06] E. Kubilinskas and M. Pióro. Iterative design of two layer networks to achieve throughput maximization. In *INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[KR05] K. Kompella and Y. Rekhter. Label switched paths (LSP) hierarchy with generalized multi-protocol label switching (GMPLS) traffic engineering (TE). RFC 4206, IETF, October 2005.

[KS98] S. G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proceedings of the 6th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 153–168. Springer-Verlag, 1998.

[KS02a] S. G. Kolliopoulos and C. Stein. Approximation algorithms for single-source unsplittable flow. *SIAM Journal on Computing*, 31(3) :919–946, 2002.

[KS02b] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 184–193. Society for Industrial and Applied Mathematics, 2002.

[KSG04] A. Kodian, A. Sack, and W.D. Grover. p-cycle network design with hop limits and circumference limits. In *First International Conference on Broadband Networks (BROADNETS'04)*, pages 244–253, 2004.

[KW98] S.O. Krumke and H.-C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66 :81–85, 1998.

[LD02] B. Liau and B. Decocq. Réseaux optiques du futur : optimisation des réseaux. In *Actes des Conférences France Télécom Recherche*, number 19, Juin 2002.

[Liu02] K.H. Liu. IP *over* WDM. J.Wiley & sons, 2002.

[LKD02] G. Li, C. Kalmanek, and R. Doverspike. Fiber span failure protection in mesh optical networks. *Optical Networks Magazine*, 3(3) :21–31, May 2002.

[LR97] C.C. Lindner and C.A. Rodger. *Design Theory*. Chapman & Hall, 1997.

[LT02] Y. Liu and D. Tipper. Multilayer network survivability models and their application on fault tolerant VPN design. In *INFORMS*, 2002.

[LTS05] Y. Liu, D. Tipper, and P. Siripongwutikorn. Approximating optimal spare capacity allocation by successive survivable routing. *IEEE/ACM Trans. Netw.*, 13(1) :198–211, 2005.

[LTYP03] L. Li, M. Thottan, B. Yao, and S. Paul. Distributed network monitoring with bounded link utilization in IP networks. In *INFOCOM 2003*, volume 2, pages 1189–1198, March 2003.

[LYK+02] M. Lee, J. Yu, Y. Kim, C-H. Kang, and J. Park. Design of hierarchical crossconnect WDM networks employing a two-stage multiplexing scheme of waveband and wavelength. *IEEE Journal on Selected Areas in Communications*, 20(1) :166–171, January 2002.

[Man04] E. Mannie. Generalized multi-protocol label switching (GMPLS) architecture. RFC 3945, IETF, October 2004.

[Mar00] N. Marlin. *Communications structurées dans les réseaux*. PhD thesis, Université de Nice-Sophia Antipolis, 2000.

[Mau03] C. Mauz. p-cycle protection in wavelength routed networks. In *Proceedings of the Seventh Working Conference on Optical Network Design and Modelling (ONDM'03)*, february 2003.

[Mél01] J.-L. Mélin. *Qualité de Service sur* IP. Eyrolles, 2001.

[MIB+04] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *IEEE Infocom*, Hong Kong, China, March 2004.

[ML01] E. Modiano and P. Lin. Traffic grooming in WDM networks. *IEEE Communications Magazine*, 39(7) :124–129, July 2001.

[MM00] G. Mohan and C. Siva Ram Murthy. Lightpath restoration in WDM optical networks. *IEEE Network*, nov/dec, 2000.

[MNT01] E. Modiano and A. Narula-Tam. Survivable routing of logical topologies in WDM networks. In *IEEE INFOCOM*, pages 348–357, 2001.

[MUK+04] T. Mori, M. Uchida, R. Kawahara, J. Pan, and S. Goto. Identifying elephant flows through periodically sampled packets. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, Taormina, Italy, October 2004.

[MVS01] D. Moore, G. M. Vœlker, and S. Savage. Inferring Internet Denial of Service Activity. In *Proceedings of the 10th Security Symposium (USENIX Security '01)*, Washington D.C., USA, August 2001.

[NI92] H. Nagamochi and T. Ibaraki. Computing edge connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5 :54–66, 1992.

[Nie06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Number 31 in Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

[NR06] T. Noronha and C. Ribeiro. Routing and wavelength assignment by partition coloring. *European Journal of Operational Research*, 171(3) :797–810, 2006.

[NT04] H. X. Nguyen and P. Thiran. Active Measurement for Multiple Link Failures Diagnosis in IP Networks. In *5th International Workshop on Passive and Active Network Measurement (PAM 2004)*, number 3015 in LNCS, pages 185–194, Antibes Juan-les-Pins, France, April 2004. Springer.

[OASC03] P. Owezarski, F.-X. Andreu, K. Salamatian, and C. Chekroun. Rapport d'état de l'art sur la métrologie. Technical report, projet METROPOLIS, 2003.

[OMSY02] E. Oki, N. Matsuura, K. Shiomoto, and N. Yamanaka. A disjoint path selection scheme with shared risk link groups in GMPLS networks. *IEEE Communications Letters*, 6(9) :406– 408, September 2002.

[OSYZ95] M. O'Mahony, D. Simeonidu, A. Yu, and J. Zhou. The design of the european optical network. *Journal of Lightwave Technology*, 13(5) :817–828, 1995.

[PAM00] V. Paxson, A.K. Adams, and M. Mathis. Experiences with NIMI. In *Passive & Active Measurement Workshop (PAM 2000)*, Hamilton, New Zealand, April 2000.

[PAMM98] V. Paxson, G. Almes, J. Mahdavi, and Mathis M. Framework for IP performance metrics. RFC 2330, IETF, May 1998.

[PG06] J. Doucette P. Giese, W. D. Grover. Physical-layer p-cycles adapted for router-level node protection : A multi-layer design and operation strategy. *IEEE Journal on Selected Areas in Communications (JSAC), in review*, April 2006.

[PM04] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier-Morgan Kaufmann, 2004.

[PPJ+01] D. Papadimitriou, F. Poppe, J. Jones, S. Venkatachalam, S. Dharanikota, R. Jain, R. Hartani, and D. Griffith. Inference of shared risk link groups. IETF Draft, OIF Contribution, OIF 2001-066, 2001.

[Puj02] G. Pujolle. *Les Réseaux*. Eyrolles, 3^{eme} edition, 2002.

[PV05] S. Petat and M.-E. Voge. Groupage sur un chemin orienté. In *Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05)*, pages 21–24, Presqu'île de Giens, May 2005.

[QZCZ04] J. Quittek, T. Zseby, B. Claise, and S. Zander. Requirements for IP Flow Information Export. RFC 3917, IETF, October 2004.

[Riz99] R. Rizzi. On minimizing symmetric set functions. Technical report, University of Trento, 1999.

[RLA04] B. Rajagopalan, J. Luciani, and D. Awduche. IP over optical networks : A framework. RFC 3717, IETF, March 2004.

[RM99a] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part 1 : Protection. In *IEEE INFOCOM*, volume 2, pages 744–751, New York, March 1999.

[RM99b] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part 2 : Restoration. In *IEEE ICC*, volume 3, pages 2023–2030, Vancouver, Canada, June 1999.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. RFC 3031 multiprotocol label switching architecture. *www.ietf.org*, January 2001.

[SGA02] D. Schupke, C. Gruber, and A. Autenrieth. Optimal configuration of p-cycles in WDM networks. In *IEEE International Conference on Communications (ICC'02)*, volume 5, pages 2761–2765, 2002.

[SGKT05] K. Suh, Y. Guo, J. Kurose, and D. Towsley. Locating network monitors : complexity, heuristics, and coverage. In *Proceedings of IEEE Infocom*, Miami, USA, March 2005.

[SKS03] S. Sengupta, V. Kumar, and D. Saha. Switched optical backbone for cost-effective scalable core IP networks. *IEEE Communications Magazine*, 41(6) :60–70, June 2003.

[Sku02] M. Skutella. Approximating the single source unsplittable min-cost flow problem. *Mathematical Programming*, Ser.B(91) :493–514, 2002.

[Sla95] P. Slavík. Improved performance of the greedy algorithm for Minimum Set Cover and Minimum Partial Cover problems. Technical Report 95-45, Departement of Computer Science, SUNY at Buffalo, 1995.

[Sla96] P. Slavík. A tight analysis of the greedy algorithm for set cover. In *STOC '96 : Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 435–441, New York, NY, USA, 1996. ACM Press.

[SMW02] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with rocketfuel. In *SIGCOMM*. ACM, 2002.

[Som06] A. K. Somani. *Survivability and Traffic Grooming in* WDM *Optical Networks*. Cambridge University Press, 2006.

[SP03] D. A. Schupke and R. G. Prinz. Capacity efficiency and restorability of path protection and rerouting in WDM networks subject to dual failures. *Photonic Network Communications*, September 2003.

[SPD+06] D. Staessens, B. Puype, L. Depré, I. Lievens, D. Colle, M. Pickavet, and P. Demeester. Multilayer recovery mechanisms in backbone networks. In *INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[SR06] T. Stidsen and S. Ruepp. Shortcut span protection. In *INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[SS99] A. A. M. Saleh and J. M. Simmons. Architectural principles of optical regional and metropolitan access networks. *IEEE/OSA Journal of Lightwave Technology*, 17(12) :2431–2448, December 1999.

[SS02] M.D. Swaminathan and K.N. Sivarajan. Practical routing and wavelength assignment algorithms for all optical networks with limited wavelength conversion. In *IEEE International Conference on Communications*, volume 25, pages 2750–2755, NewYork, April 2002.

[Suu74] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4 :125–145, 1974.

[SYR05] 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, August 2005.

[Tan01] A. Tanenbaum. *Réseaux, cours et exercices*. Dunod, Prentice Hall, 2001.

[TH01] F. Touvet and D. Harle. Network resilience in multilayer networks : A critical review and open issues. In *ICN '01 : Proceedings of the First International Conference on Networking-Part 1*, pages 829–838, London, UK, 2001. Springer-Verlag.

[TR04a] A. Todimala and B. Ramamurthy. Imsh : An iterative heuristic for srlg diverse routing in WDM mesh networks. In *IEEE Thirteenth International Conference on Computer Communications and Networks (ICCCN '04)*, pages 199–204, Chicago, October 2004.

[TR04b] A. Todimala and B. Ramamurthy. Survivable virtual topology routing under shared risk link groups in WDM networks. In *First Annual International Conference on Broadband Networking (BroadNets '04)*, pages 130–139, San Jose, CA, Oct. 2004.

[Vog06a] M.-E. Voge. Graphes colorés - arbre couvrant coloré. In *Huitièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'06)*, pages 41–44, Trégastel, May 2006.

[Vog06b] M.-E. Voge. How to transform a multilayer network into a colored graph. In *IEEE ICTON/COST 293 annual conference on GRAphs and ALgorithms in communication networks*, Nottingham, June 2006.

[WCX02] Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84 :99–101, 2002.

[Wei02] J.Y. Wei. Advances in the management and control of optical internet. *IEEE Journal on Selected Areas in Communications*, 20(4) :768–784, May 2002.

[Wir01] H.-C. Wirth. *Multicriteria Approximation of Network Design and Network Upgrade Problems*. PhD thesis, Bayerische Julius-Maximilians-Universität Würzburg, 2001.

[XM02] Y. Xiong and L. Mason. Comparison of two path restoration schemes in self-healing networks. *Comput. Networks*, 38(5) :663–674, 2002.

[XXQ03a] D. Xu, Y. Xiong, and C. Qiao. Novel algorithms for shared segment protection. *IEEE Journal on Selected Areas in Communications*, 21(8) :1320–1331, October 2003.

[XXQ03b] D. Xu, Y. Xiong, and C. Qiao. Protection with multi-segments (PROMISE) in networks with shared risk link groups (SRLG). *IEEE/ACM Transactions on Networking*, 11(2) :248–258, 2003.

[XXQL03] D. Xu, Y. Xiong, C. Qiao, and G. Li. Trap avoidance and protection schemes in networks with shared risk link groups. *Journal of Lightwave Technology*, 21(11) :2683–2693, November 2003.

[YDA00] Y. Ye, S. Dixit, and M. Ali. On joint protection/restoration in IP-centric DWDM-based optical transport networks. *IEEE Communications Magazine*, pages 174–183, June 2000.

[YJ04] S. Yuan and J.P. Jue. Dynamic lightpath protection in WDM mesh networks under risk disjoint constraints. In *IEEE Globecom*, 2004.

[YOM03] S. Yao, C. Ou, and B. Mukherjee. Design of hybrid optical networks with waveband and electrical TDM switching. In *IEEE Globecom*, pages 2803–2808, San Francisco, CA, December 2003.

[YR05] W. Yao and B. Ramamurthy. Survivable traffic grooming in wdm mesh networks under SRLG constraints. In *IEEE International Conference on Communications, ICC'05*, volume 3, pages 1751–1755, May 2005.

[YVJ05] S. Yuan, S. Varma, and J.P. Jue. Minimum-color path problems for reliability in mesh networks. In *IEEE InfoCom*, 2005. 59-04.

[ZD02] H. Zhang and A. Durresi. Differentiated multi-layer survivability in IP/WDM networks. *IEEE/IFIP Network Operations and Management Symposium*, 8 :681–696, 2002.

[ZM03] 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.

[ZMD+05] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall. Techniques for IP packet selection, July 2005. Draft IETF : `http://www.ietf.org/internet-drafts/draft-ietf-psamp-sample-tech-07.txt`.

Annexe A

Problèmes de référence

A.1 Classe de complexité

A.1.1 RP, $coRP$ et ZPP

Randomized Polynomial Time (RP) RP est la classe des problèmes de décision solubles par une machine de Turing non déterministe tels que :

- Si la réponse doit être OUI au moins la moitié des chemins de calcul aboutissent à la réponse OUI,
- Si la réponse doit être NON, tous les chemins de calcul aboutissent à la réponse NON.

coRP La classe $coRP$ est le complémentaire de la classe RP. Il s'agit des problèmes de décision solubles par une machine de Turing non déterministe tels que :

- Si la réponse doit être NON au moins la moitié des chemins de calcul aboutissent à la réponse NON,
- Si la réponse doit être OUI, tous les chemins de calcul aboutissent à la réponse OUI.

Zero Probability of error (ZPP) Par définition $ZPP = RP \cap coRP$. Pour tous les problèmes de cette classe il existe donc une machine de Turing non déterministe qui ne se trompe jamais lorsqu'elle répond OUI, et une seconde qui ne se trompe jamais lorsqu'elle répond NON. Pour résoudre les problèmes de cette classe il suffit d'utiliser les deux machines en parallèle jusqu'à ce que l'une d'elle donne une réponse certaine. Les problèmes de cette classe peuvent donc toujours être résolus, mais le temps nécessaire avant d'arriver à une réponse définitive n'est pas connu a priori. Notons que la probabilité de n'avoir pas obtenu de réponse certaine après k essais de chacune des deux machines est égale à 2^{-k}.

A.1.2 $TIME(f(n))$

La classe $TIME(f(n))$ comprend tous les problèmes dont les instances de taille $n \in \mathbb{N}$ pouvent être résolus en temps $\mathcal{O}(f(n))$ par une machine de Turing déterministe, pour une fonction non décroissante de $\mathbb{N}$ dans $\mathbb{N}$. Par exemple la classe P peut être définie par $P = TIME(n^k) = \cup_{j>0} TIME(n^j)$.

Il existe également la classe $NTIME(f(n))$ des problèmes dont les instances de taille $n \in \mathbb{N}$ pouvent être résolus en temps $\mathcal{O}(f(n))$ par une machine de Turing déterministe, pour une fonction non décroissante de $\mathbb{N}$ dans $\mathbb{N}$. La classe NP est alors donnée par $NP = NTIME(n^k)$.

A.2 Quelques problèmes difficiles et non approximables

A.2.1 Maximum 3 Satisfiability

Problème A.1 (Maximum 3 Satisfiability**)**

Entrée : *Un ensemble $U = \{x_1, \ldots, x_n\}$ de n variables booléennes, une collection $C = \{C_1, \ldots, C_m\}$ de m clauses disjonctives d'au plus trois littéraux où un littéral est une variable $x_i \in U$ ou sa négation $\overline{x_i}$.*

Sortie : *Une affectation des valeurs* vrai *ou* faux *aux variables de U.*

Objectif : *Maximiser le nombre de clauses satisfaites par l'affectation des valeurs aux variables.*

Exemple Considérons une instance de Maximum 3 Satisfiability avec 4 variables $U = \{x_1, x_2, x_3, x_4\}$ et 4 clauses $C = \{C_1 = (x_1 \vee x_2 \vee \overline{x_3}), C_2 = (\overline{x_2} \vee x_3 \vee \overline{x_4}), C_3 = (\overline{x_1} \vee \overline{x_3} \vee x_4), C_4 = (\overline{x_1} \vee x_3 \vee \overline{x_4})\}$.

Pour satisfaire la clause C_1 il faut que la valeur vrai soi affectée à au moins un des littéraux x_1, x_2 ou $\overline{x_3}$, c'est à dire que l'une des variables x_1 ou x_2 reçoive la valeur vrai ou que la variable x_3 reçoive la valeur faux. Une solution réalisable de cette instance consiste en l'affectation suivante : $x_1 \leftarrow$ vrai, $x_2 \leftarrow$ faux, $x_3 \leftarrow$ vrai, $x_4 \leftarrow$ vrai. Toutes les clauses sont satisfaites par cette affectation.

Théorème A.2 ([ALM+92]) *Le problème* Maximum 3 Satisfiability *est NP-difficile et il existe $\varepsilon > 0$ tel que* Maximum 3 Satisfiability *n'est pas approximable à un facteur $1 + \varepsilon$ sauf si $P = NP$.*

A.2.2 Minimum Set Cover et Minimum Partial Cover

Problème A.3 (Minimum Set Cover**)**

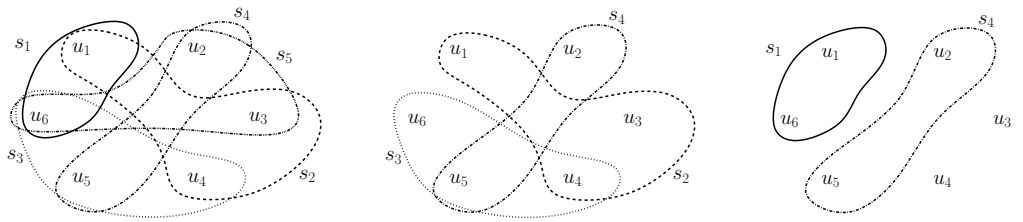
Entrée : *Un ensemble $U = \{u_1, \ldots, u_n\}$ et une collection de sous ensembles $S = \{s_1, \ldots s_m\}$ de U.*

Sortie : *Une couverture de U : une sous collection S' de sous ensembles appartenant à S telle que chaque élément de U appartienne à au moins un sous ensemble de S'.*

Objectif : *Minimiser $|S'|$.*

Exemple La figure A.1(a) donne un exemple d'instance de Minimum Set Cover pour lequel $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ et $S = \{s_1, s_2, s_3, s_4, s_5\}$ avec $s_1 = \{u_1, u_6\}$, $s_2 = \{u_1, u_3, u_4\}$, $s_3 = \{u_4, u_5, u_6\}$, $s_4 = \{u_2, u_5\}$ et $s_5 = \{u_2, u_3, u_6\}$. Une solution réalisable de cette instance de Minimum Set Cover est représentée par la figure A.1(b). Cette solution est composée des ensembles s_2, s_3 et s_4 qui couvrent bien tous les éléments de U. De plus il s'agit d'une solution optimale puisque deux sous ensembles ne suffisent pas à couvrir tous les éléments.

La complexité du problème Minimum Set Cover a fait l'objet de nombreuse publications. Elle est assez précisément déterminée grâce aux travaux de [Fei98] et [Sla96] qui ont apporté les dernières améliorations concernant le facteur d'inapproximabilité d'une part et le facteur d'approximation de l'algorithme le plus performant connu d'autre part. Ces deux facteurs sont du même ordre de grandeur comme le montrent les deux théorèmes A.4 et A.5 suivants.



(a) Une instance de Minimum Set Cover

(b) Une couverture

(c) Une couverture partielle, pour $k = \frac{2}{3}$

Fig. A.1 – Minimum Set Cover et Minimum Partial Cover

Théorème A.4 ([Fei98]) *Le problème* Minimum Color *st*-Path *n'est pas approximable à un facteur* $(1-\varepsilon)\ln|U|$ *pour tout* $\varepsilon > 0$ *sauf si* $NP \subseteq TIME(n^{\log\log n})$.

Le principe de l'algorithme 2 est simple, à chaque itération l'ensemble s_i choisi est celui qui permet de couvrir le plus d'éléments de U non encore couverts, jusqu'à ce que tous les éléments de U soient couverts.

Théorème A.5 ([Sla96]) *L'algorithme 2 donne une approximation du problème* Minimum Set Cover *à un facteur* $\ln|U| - \ln\ln|U| + 3 + \ln\ln 32 - \ln 32$*, c'est à dire de l'ordre de* $\ln|U| - \ln\ln|U| + o(1)$.

Algorithme 2 Algorithme d'approximation pour les problème Minimum Set Cover et Minimum Partial Cover

Entrées: Un ensemble $U = \{u_1, \ldots, u_n\}$ et une collection de sous ensembles $S = \{s_1, \ldots s_m\}$ de U.

Sorties: Une couverture S' de U de taille minimum.

1: $S' \leftarrow \emptyset$, $U' \leftarrow U$.
2: **tant que** $U' \neq \emptyset$ **faire**
3: **Choisir** s_i tel que $|U' - s_i|$ est minimum
4: $U' \leftarrow U' \cap s_i$, $S' \leftarrow S' \cup s_i$
5: **fin tant que**
6: **retourner** S'

Le théorème A.6 signifie que le facteur d'approximation de l'algorithme 2 ne pourra plus être amélioré.

Théorème A.6 ([Sla96]) *Pour tout ensemble* U *tel que* $|U| > 2$ *il existe une collection* S *de sous ensembles de* U *couvrant* U *telle que l'algorithme 2 donne une solution de valeur strictement supérieure à* $(\ln|U| - \ln\ln|U| - 1 + \ln 2)$ *fois l'optimale.*

Les résultats de [Sla96] s'étendent au problème plus général Minimum Partial Cover. L'algorithme 2 s'adapte au problème Minimum Partial Cover, il suffit de remplacer la condition de la boucle à la ligne 2 par $|U'| > |U| - \lceil k|U| \rceil$ pour que l'algorithme termine dès que $\lceil k|U| \rceil$ éléments de U sont couverts.

Problème A.7 (Minimum Partial Cover)
Entrée : *Un ensemble* $U = \{u_1, \ldots, u_n\}$*, une collection de sous ensembles* $S = \{s_1, \ldots s_m\}$ *de* U *et une constante* $k \in]0, 1]$.
Sortie : *Une sous collection* $S' \in S$ *permettant de couvrir* $\lceil k|U| \rceil$ *éléments de* U.
Objectif : *Minimiser* $|S'|$.

Exemple Pour obtenir une instance de Minimum Partial Cover il suffit d'ajouter une constante $k \in]0, 1]$ à l'instance de Minimum Set Cover décrite par la figure A.1. Prenons $k = \frac{2}{3}$, $\frac{2}{3} \times 6 = 4$ éléments de U doivent être couverts dans une solution de cette instance de Minimum Partial Cover. La figure A.1(c) illustre une telle solution, elle est composée des sous ensembles s_1 et s_4 et est optimale car aucun sous ensemble ne contient quatre éléments de U.

Théorème A.8 ([Sla96, Sla95]) *L'algorithme 2 donne une approximation du problème* Minimum Partial Cover *à un facteur* $\ln\lceil k|U| \rceil - \ln\ln\lceil k|U| \rceil + 3 + \ln\ln 32 - \ln 32$*, c'est à dire de l'ordre de* $\ln\lceil k|U| \rceil - \ln\ln\lceil k|U| \rceil + o(1)$.

Théorème A.9 ([Sla96, Sla95]) *Pour tout ensemble* U *et toute constante* $k \in]0, 1]$ *tels que* $\lceil k|U| \rceil > 2$ *il existe une collection* S *de sous ensembles de* U *couvrant* $\lceil k|U| \rceil$ *élément telle que l'algorithme 2 donne une solution de valeur strictement supérieure à* $(\ln\lceil k|U| \rceil - \ln\ln\lceil k|U| \rceil - 1 + \ln 2)$ *fois l'optimale.*

A.2.3 Maximum Independant Set et Maximum Clique

Problème A.10 (Maximum Clique)
Entrée : *Un graphe* $G = (V, E)$.
Sortie : *Une clique : un sous ensemble de sommets* $V' \subseteq V$ *tel que* $\forall u, v \in V \ \{u, v\} \in E$.
Objectif : *Maximiser* $|V'|$.

Problème A.11 (Maximum Independant Set)
Entrée : *Un graphe* $G = (V, E)$.
Sortie : *Un ensemble indépendant : un sous ensemble de sommets* $V' \subseteq V$ *tel que* $\forall u, v \in V \ \{u, v\} \notin E$.
Objectif : *Maximiser* $|V'|$.

Théorème A.12 ([Has99]) *Les problèmes* Maximum Clique *et* Maximum Independant Set *ne sont pas approximables à un facteur* $|C|^{1-\varepsilon}$ *pour tout* $\varepsilon > 0$ *sauf si* $NP = ZPP$*, et ne sont pas approximables à un facteur* $|C|^{\frac{1}{2}-\varepsilon}$ *pour tout* $\varepsilon > 0$ *sauf si* $P = NP$.

A.2.4 Set Splitting

Problème de décision A.13 (Set Splitting)
Données : *Une collection* $\{S_i | i \in 1, \ldots, N\}$ *de sous ensembles d'un ensemble fini* S.
Question : *Existe-t-il deux sous ensembles* $S', S'' \subseteq S$ *disjoints, tels que chacun intersecte tous les sous ensembles de la collection* $\{S_i | i \in 1, \ldots, N\}$ *?*

Théorème A.14 ([GJ79]) *Le problème* Set Splitting *est NP-Complet.*

A.2.5 Red Blue Set Cover

Problème A.15 (Red Blue Set Cover)
Entrée : *Deux ensembles* R *et* B*, une collection* S *de sous ensembles de* $R \cup B$.
Sortie : *Une sous collection* $S' \in S$ *couvrant tous les éléments de* B.
Objectif : *Minimiser le nombre d'éléments de* R *couverts par* S'.

Théorème A.16 ([CDKM00]) *Le problème* Red Blue Set Cover *n'est pas approximable à un facteur* $\mathcal{O}(2^{\log^{1-\varepsilon}|S|^4})$ *pour tout* $\varepsilon > 0$ *sauf si* $NP \subseteq DTIME(n^{polylog(n)})$.

A.2.6 Maximum Set Packing

Problème A.17 (Maximum Set Packing)
Entrée : *Une collection* C *de sous ensembles d'un ensemble* U.
Sortie : *Une sous collection* $C' \subseteq C$ *telle que* $\forall c_i, c_j \in C'\ c_i \cap c_j = \emptyset$.
Objectif : *Maximiser* $|C'|$.

Le résultat suivant est obtenu grâce à une réduction du problème Maximum Clique au problème Maximum Set Packing.

Théorème A.18 ([Kar72, ADP80]) *Le problème* Maximum Set Packing *n'est pas approximable à un facteur* $|C|^{1-\varepsilon}$ *pour tout* $\varepsilon > 0$ *sauf si* $NP = ZPP$*, et n'est pas approximable à un facteur* $|C|^{\frac{1}{2}-\varepsilon}$ *pour tout* $\varepsilon > 0$ *sauf si* $P = NP$.

A.2.7 Minimum Label Cover

Soit $B = (U, V, E)$ un graphe biparti et deux ensembles de labels L_U et L_V qui peuvent être affectés aux sommets de U et de V respectivement. Pour chaque arête $\{u, v\} \in E$, $u \in U$ et $v \in V$, une relation $\Pi_{uv} \subseteq L_U \times L_V$ consistant de paires de labels *admissibles* pour l'arête $\{u, v\}$ est donnée.

Un labeling est une paire de fonctions $f_U : U \to 2^{L_U}$[1] et $f_V : V \to 2^{L_V} - \{\emptyset\}$ affectant à chaque sommet de B un sous ensemble de labels. Le coût d'un labeling est donné par $\sum_{u_i \in U} |f_U(u_i)|$. Un labeling *couvre* une arête $\{u, v\}$, $u \in U$ et $v \in V$, si pour *chaque* label $l_v \in f_V(v)$ il existe un label $l_u \in f_U(u)$ tel que $(l_u, l_v) \in \Pi_{uv}$. Le problème est de trouver un labeling couvrant toutes les arêtes de B de coût minimum.

Pour assurer l'existence d'un labeling couvrant toutes les arêtes, on impose que le label $1_v \in L_V$ appartienne à une paire admissible de Π_{uv} pour toute arête $\{u, v\} \in E$. Un label *éligible* pour un sommet $v \in V$ est un label qui appartient à au moins une paire admissible de Π_{uv} pour *toute* arête $\{u, v\} \in E$. Le label 1_v est donc un label éligible pour tous les sommets de V.

Problème A.19 (Minimum Label Cover)

Entrée :
- $B = (U, V, E)$ *un graphe biparti,*
- *deux ensembles de labels* L_U *pour* U *et* L_V *pour* V*,*
- *une relation* $\Pi_{uv} \subseteq L_U \times L_V$ *consistant de paires de labels admissibles pour chaque arête* $\{u, v\} \in E$*.*

Sortie : *Un labeling couvrant toutes les arêtes : deux fonctions* $f_U : U \to 2^{L_U}$ *et* $f_V : V \to 2^{L_V} - \{\emptyset\}$ *telles que pour chaque label* $l_v \in f_V(v)$ *il existe un label* $l_u \in f_U(u)$ *tel que* $(l_u, l_v) \in \Pi_{uv}$ *pour toute arête* $\{u, v\} \in E$*.*

Objectif : *Minimiser* $\sum_{u_i \in U} |f_U(u_i)|$*.*

Exemple La figure A.2 représente une instance de Minimum Label Cover. La relation Π_{ce} donnée pour l'arête $\{c, e\}$ indique que lorsque le label 1_v est affecté au sommet $e \in V$, l'un des labels 4_u, 2_u ou 3_u doit être affecté au sommet $c \in U$ pour que l'arête $\{c, e\}$ puisse être couverte. De même si le label 3_v est affecté au sommet e, le label 1_u doit être affecté au sommet c. La relation Π_{ae} contient les paires admissibles $(4_u, 2_v)$ et $(2_u, 2_v)$, donc si le label 2_v est affecté à e, l'un des labels 4_u ou 2_u doit être affecté au sommet $a \in U$ pour couvrir l'arête $\{a, e\}$. Cependant le label 2_v n'est pas éligible pour le sommet e car il n'appartient à aucune paire admissible dans la relation Π_{ce}. Affecter ce label au sommet e empêcherait de couvrir l'arête $\{c, e\}$ puisqu'aucun label de L_U formant une paire admissible avec 2_v ne peut être affecté au sommet c.

Les fonctions suivantes constituent une solution réalisable pour cette instance de Minimum Label Cover de coût $|f_U(a)| + |f_U(b)| + |f_U(c)| + |f_U(d)| = 5$:
- $f_U(a) = \{3_u\}$, $f_U(b) = \{1_u\}$, $f_U(c) = \{1_u, 2_u\}$, $f_U(d) = \{4_u\}$,
- $f_V(e) = \{3_v\}$, $f_V(f) = \{1_v\}$, $f_V(g) = \{2_v\}$,

Le plus grand facteur d'inapproximabilité pour le problème Minimum Label Cover a été prouvé dans [DS04b].

Théorème A.20 ([DS04b]) *Le problème* Minimum Label Cover *n'est pas approximable à un facteur* $2^{\log^{\delta} |V|}$ *avec* $\delta = (\log \log |V|)^{-\varepsilon}$ *pour tout* $\varepsilon < \frac{1}{2}$ *sauf si* $P = NP$*.*

On peut constater sur la figure A.3 que la fonction $2^{\log^{1-(\log \log x)^{-\frac{1}{3}}} x}$ démarre plus lentement que la fonction $\log x$ mais la dépasse et croit beaucoup plus rapidement.

[1] 2^L désigne l'ensemble des parties d'un ensemble L donné.

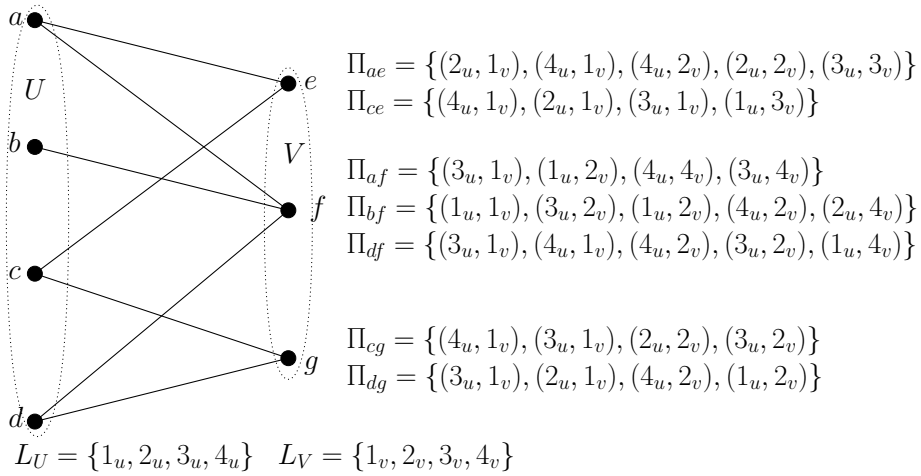


FIG. A.2 – Exemple d'instance de MINIMUM LABEL COVER.

A.2.8 UNSPLITTABLE FLOW

Le problème UNSPLITTABLE FLOW a été introduit en 1996 par Kleinberg [Kle96] et a depuis été très étudié [EH02, Sku02, KS02a, Asa00, CLR03, BCLR04, KS02b, Gef01, KS98, AR01, AdC03].

Problème A.21 (UNSPLITTABLE FLOW)

Entrée : *Un graphe $G = (V, E)$ orienté ou non, une capacité $c_e \geq 0$ par arête $e \in E$, k paires de sommets (s_i, t_i) et une demande $d_i \geq 0$ entre la source s_i et la destination t_i de chaque commodité.*

Sortie : *Un unique chemin pour chaque commodité (s_i, t_i) de capacité d_i tel que le flot total sur chaque arête respecte sa capacité.*

Objectif : *Plusieurs objectifs ont été étudiés :*

- **Congestion minimum** *Minimiser la valeur $\alpha \geq 1$ telle qu'il existe un flot monorouté violant la capacité d'une arête d'un facteur au plus α.*
- **Partition minimum** *trouver une partition des commodités en un nombre minimum de sous-ensembles tels qu'il existe un flot monorouté pour chaque sous-ensemble.*
- **Demande routable maximum** *trouver un flot monorouté pour un sous-ensemble de commodités maximisant la somme des demandes routées.*

Certains articles ([EH02, Sku02, KS02a]) tiennent aussi compte de la version avec coûts sur les arêtes, l'objectif est de minimiser la congestion tout en maintenant le coût de la solution inférieur à un budget B donné.

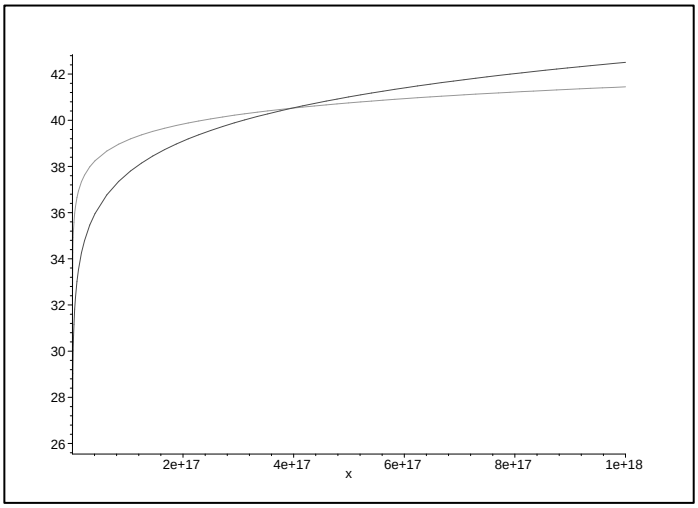


FIG. A.3 – Fonctions $2^{\log^{1-(\log\log x)^{-\frac{1}{3}}} x}$ et $\log x$

A.2.9 MINIMUM STEINER TREE [GJ79]

Problème A.22 (MINIMUM STEINER TREE)

Entrée : *Un graphe non orienté $G = (V, E)$, un sous-ensemble de sommets $S \subseteq V$, un poids $w_e \geq 0$ sur chaque arête $e \in E$.*

Sortie : *Un arbre T couvrant tous les sommets de S.*

Objectif : *Minimiser la somme des poids des arêtes de T : $\sum_{e \in T} w_e$.*

Annexe B

Transformation d'un réseau multicoloré : Algorithme de décision

Nous présentons une suggestion d'implémentation de l'algorithme exposé à la section 4.4.1 du chapitre 4 permettant de décider si un réseau multicoloré peut être transformé en un graphe coloré de span maximum 1. Notre objectif n'étant pas d'obtenir les meilleurs performances mais simplement d'évaluer la complexité en temps de l'algorithme, le pseudo-code suivant n'est pas optimisé pour une exécution efficace.

Organisation des parties, algorithme global (Algorithme 3) L'algorithme peut être découpé en cinq parties qui doivent être exécutées successivement. Elles s'organisent très simplement comme indiqué par l'algorithme 3. Chaque partie opère des modifications sur les ensembles et les variables décrits ci-après afin que les parties suivantes disposent des éléments nécessaires à leur bon déroulement. Ces éléments sont connus et accessibles par chaque partie comme des variables globales de l'algorithme 3 et sont définis comme suit.

e.statut(c) : statut de la couleur $c \in \mathcal{C}$ sur l'arête $e \in E_{\mathcal{R}}$, POSITIONNABLE, LIBRE, FIXE ou SEMI-LIBRE. Par défaut le statut est POSITIONNABLE.

e.position(c) : position que doit avoir la couleur $c \in \mathcal{C}$ sur l'arête $e \in \mathcal{R}$ pour être de span 1, il s'agit du sommet extrémité de e auquel c doit être adjacente.

e.position.oppose(c) : désigne l'extrémité de e qui n'est pas la position de la couleur c, cette position peut donc être prise par la deuxième couleur portée par e.

e.autre.col(c) : pour une couleur c portée par e, désigne l'autre couleur portée par cette arête, à utiliser uniquement pour une arête e portant exactement deux couleurs.

c.fixe : variable booléenne, VRAI si la couleur c contient au moins une arête fixe, FAUX par défaut.

c.semi − libre : variable booléenne, VRAI si la couleur c est semi-libre, FAUX par défaut.

NbrSemiLibre : nombre de couleurs semi-libres, initialement nul.

Continu : variable booléenne.

c.arete : ensemble des arêtes portant la couleur c dans $\mathcal{R}$.

c.sommet : ensemble des sommets adjacents à la couleur c dans $\mathcal{R}$.

c.centre : ensemble de sommets pouvant être au centre d'une étoile pour la couleur c, ou au sens large, sommets adjacents à une arête fixe pour la couleur c. Cet ensemble est initialement vide.

e.couleur : ensemble des couleurs portées par l'arête e.

e.extremite : ensemble des deux sommets extrémité de l'arête e.

Avec une librairie comme MASCOPT [1] les ensembles c.arete, c.sommet, e.couleur et e.extremite peuvent être maintenus et sont accessibles en temps constant.

Algorithme 3 Décision concernant la possibilité pour un réseau d'une transformation en graphe coloré de span maximum 1.

Entrées: un réseau $\mathcal{R} = (V_\mathcal{R}, E_\mathcal{R}, \mathcal{C})$.

Sorties: retourne VRAI et un graphe coloré $G = (V, E, \mathcal{C})$ de span maximum 1 issu de la transformation de $\mathcal{R}$ si cette transformation est possible, sinon retourne FAUX et une preuve que la transformation en graphe de span 1 n'est pas possible.

1: exécuter l'Algorithme 4
2: exécuter l'Algorithme 5
3: exécuter l'Algorithme 6
4: exécuter l'Algorithme 7
5: exécuter l'Algorithme 8
6: **retourner** le graphe obtenu à partir des positions des couleurs déterminées au cours des 5 algorithmes précédents.

Réduction du nombre de couleurs par arête (Algorithme 4) L'objectif de la première partie de l'algorithme est l'élimination des couleurs qui ne sont présentes que sur une seule arête du réseau afin d'obtenir un réseau dont les arêtes portent au plus deux couleurs. C'est aussi dans cette partie que sont repérées toutes les arêtes fixes.

Le rôle des lignes 1 à 3 est l'identification des couleurs qui ne sont présentes que sur une seule arête et dont la position est libre par conséquent. Ensuite les arêtes ne portant qu'une seule couleur sont repérées (lignes 5-7). Avec les lignes 8 à 13, les couleurs présentes sur une seule arête sont supprimées du réseau jusqu'à laisser au moins deux couleurs par arête non fixe. S'il reste plus de deux couleurs sur une arête ces couleurs ne peuvent pas être de span 1 simultanément et cette preuve est retournée par l'algorithme (lignes 14-16). Notons qu'à l'issue de cette partie des arêtes portant deux couleurs libres peuvent se trouver dans le réseau, les positions peuvent être décidées arbitrairement dès maintenant mais pour respecter les fonctions particulières de chaque partie ce ne sera fait que plus tard.

Couleurs semi-libres (Algorithme 5) L'objectif de cette partie de l'algorithme est l'identification du statut des couleurs n'apparaissant que sur une arête multiple (couleurs c telles que $|c.\text{sommet}| = 2$). Les lignes 2 à 7 permettent de détecter les couleurs qui sont libres sur certaines arêtes constituant l'arête multiple car au moins une des arêtes auxquelles ces couleurs appartiennent sont fixes. Si une couleur ne comporte aucune arête fixe il s'agit d'une couleur semi-libre (lignes 8-14).

Étoiles (Algorithme 6) Cette partie est consacrée à la recherche des sommets constituant le centre des étoiles pour les couleurs qui ne contiennent pas seulement une arête multiple (les couleurs c telles que $|c.\text{sommet}| > 2$). Lorsqu'une couleur contient au moins une arête fixe, toutes les arêtes doivent être adjacentes à au moins une extrémité d'arête fixe (lignes 2-5). De plus les arêtes fixes doivent être connexes (lignes 6-9). Par contre si une couleur ne contient aucune arête fixe un unique sommet pourra être incident à toutes les arêtes de la couleur, c'est pourquoi on recherche

[1] `http://www-sop.inria.fr/mascotte/mascopt/`

Algorithme 4 Réduire le nombre de couleurs par arête à deux dans le réseau ou fournir un ensemble de couleurs ne pouvant être de span 1 simultanément et l'arête en cause.

pour tout $c \in \mathcal{C}$ telle que $|c.\text{arete}| = 1$ **faire**
  soit $\{e\} = c.\text{arete}$, $e.\text{statut}(c) \leftarrow$ LIBRE
fin pour
pour tout $e \in E_{\mathcal{R}}$ **faire**
  si $|e.\text{couleur}| = 1$ **alors**
    soit $e.\text{couleur} = \{c\}$, $e.\text{statut}(c) \leftarrow$ FIXE
    $c.\text{fixe} \leftarrow$ VRAI
  sinon si $|e.\text{couleur}| > 2$ **alors**
    pour tout $c \in e.\text{couleur}$ **faire**
      si $e.\text{statut}(c) =$ LIBRE et $|e.\text{couleur}| > 2$ **alors**
        $e.\text{couleur} \leftarrow e.\text{couleur} - \{c\}$
      fin si
    fin pour
    si $|e.\text{couleur}| > 2$ **alors**
      retourner FAUX et $e.\text{couleur}$
    fin si
  fin si
fin pour

Algorithme 5 Identification des couleurs semi-libres

pour tout $c \in \mathcal{C}$ tel que $|c.\text{sommet}| = 2$ **faire**
  si $c.\text{fixe} =$ VRAI **alors**
    pour tout $e \in c.\text{arete}$ **faire**
      si $e.\text{statut}(c) \neq$ FIXE **alors**
        $e.\text{statut}(c) \leftarrow$ LIBRE
      fin si
    fin pour
  sinon
    $c.\text{semi} - \text{libre} \leftarrow$ VRAI
    NbrSemiLibre $\leftarrow$ NbrSemiLibre $+ 1$
    pour tout $e \in c.\text{arete}$ **faire**
      $e.\text{statut}(c) \leftarrow$ SEMI-LIBRE
    fin pour
  fin si
fin pour

un sommet à l'intersection de plusieurs arêtes aux lignes 9 à 18. Quand aucun sommet n'est incident à toutes les arêtes d'un sous ensemble (lignes 14-16) la couleur ne pourra pas être de span 1 après transformation et les arêtes le prouvant sont retournées.

Une fois qu'ont été déterminés les sommets auxquels toutes les arêtes d'une couleur doivent être incidentes pour qu'elle puisse être de span 1, il faut vérifier que toutes les arêtes sont bien incidentes à ces sommets (lignes 21-23). Lorsqu'une arête est adjacente à un des sommets du centre sa position est imposée (lignes 24-27). Enfin, au cas où une arête serait incidente à deux sommets du centre à la fois, c'est que sa position est libre.

A la fin de cette partie, toutes les positions qui sont imposées à cause des étoiles sont déterminées et si une couleur ne peut pas être de span 1, elle est identifiée et la preuve en est retournée par l'algorithme.

Algorithme 6 Recherche d'étoiles, positionnement des couleurs dans les étoiles.

1: **pour tout** $c \in \mathcal{C}$ tel que $|c.\text{sommet}| > 2$ **faire**
2: **si** c.fixe = VRAI **alors**
3: **pour tout** $e \in c$.arete telle que $e.\text{position}(c) =$ FIXE **faire**
4: $c.\text{centre} \leftarrow c.\text{centre} \cup e.\text{extremite}$
5: **fin pour**
6: **si** c.centre non connexe **alors**
7: **retourner** FAUX et c.centre
8: **fin si**
9: **sinon**
10: soit $\{e_1, e_2, \ldots, e_{|c.\text{arete}|}\} = c.\text{arete}$
11: $c.\text{centre} \leftarrow e_1$, $i = 1$
12: **répéter**
13: $i \leftarrow i + 1$
14: **si** $|e_i.\text{extremite} \cap c.\text{centre}| = 0$ **alors**
15: **retourner** FAUX c, e_i et c.centre
16: **fin si**
17: $c.\text{centre} \leftarrow e_i.\text{extremite} \cap c.\text{centre}$
18: **jusqu'à** $|c.\text{centre}| \leq 1$
19: **fin si**
20: **pour tout** $e \in c$.arete **faire**
21: **si** $e.\text{statut}(c) =$ POSITIONNABLE **alors**
22: **si** $|e.\text{extremite} \cap c.\text{centre}| = 0$ **alors**
23: **retourner** FAUX et c et c.centre et e
24: **sinon si** $|e.\text{extremite} \cap c.\text{centre}| = 1$ **alors**
25: soit $\{v\} = e.\text{extremite} \cap c.\text{centre}$,
26: $e.\text{position}(c) \leftarrow v$
27: **sinon si** $|e.\text{extremite} \cap c.\text{centre}| = 2$ **alors**
28: $e.\text{statut}(c) \leftarrow$ LIBRE
29: **fin si**
30: **fin si**
31: **fin pour**
32: **fin pour**

Positionnement des couleurs semi-libres (Algorithme 7) Une fois que les positions imposées par les étoiles ont été déterminées, il est possibles de déterminer les positions des couleurs semi-libres.

Les lignes 3 à 17 positionnent les couleurs semi-libres en fonction des positions imposées des couleurs positionnables avec lesquelles elles partagent une arête. Lorsque la position d'une couleur semi-libre est déterminée elle devient elle-même positionnable et peut permettre de décider de la position d'une autre couleur semi-libre, c'est pourquoi il faut répéter ce processus de propagation de position (ligne 15). Ainsi en arrivant à la ligne 18 les couleurs semi-libres restant dans le réseau partagent nécessairement toutes leurs arêtes avec des couleurs soit semi-libres, soit libres. Leurs positions peuvent donc être déterminées arbitrairement, elles deviennent alors des couleurs positionnables et le processus de propagation de position doit reprendre (ligne 27).

Ce processus termine uniquement lorsqu'il ne reste plus aucune couleur semi-libre dans le réseau. En revanche il peut rester des arêtes libres pour certaines couleurs et après cette phase ce sont les seules dont les positions ne sont pas fixées.

Algorithme 7 Positionnement des couleurs semi-libres

1: **répéter**
2: Continu $\leftarrow$ FAUX
3: **pour tout** $c \in \mathcal{C}$ tel que c.semi $-$ libre $=$ VRAI **faire**
4: soit $\{e_1, \ldots, e_{|c.\text{arete}|}\} = c$.arete, $i \leftarrow 0$
5: **répéter**
6: $i \leftarrow i + 1$
7: **jusqu'à** e_i.statut(e_i.autre.col(c)) $\neq$ POSITIONNABLE ou $i = |c.\text{arete}|$
8: **si** e_i.statut(e_i.autre.col(c)) $=$ POSITIONNABLE **alors**
9: **pour tout** $e \in c$.arete **faire**
10: e.position(c) $\leftarrow$ e_i.position.oppose(e.autre.col(c))
11: e.statut(c) $\leftarrow$ POSITIONNABLE
12: **fin pour**
13: c.semi $-$ libre $\leftarrow$ FAUX
14: NbrSemiLibre $\leftarrow$ NbrSemiLibre $- 1$
15: Continu $\leftarrow$ VRAI
16: **fin si**
17: **fin pour**
18: **si** Continu $=$FAUX et NbrSemiLibre > 0 **alors**
19: soit $c \in \mathcal{C}$ telle que c.semi $-$ libre $=$ VRAI
20: soit $v \in c$.sommet arbitraire
21: **pour tout** $e \in c$.arete **faire**
22: e.position(c) $\leftarrow v$
23: e.statut(c) $\leftarrow$ POSITIONNABLE
24: **fin pour**
25: c.semi $-$ libre $\leftarrow$ FAUX
26: NbrSemiLibre $\leftarrow$ NbrSemiLibre $- 1$
27: Continu $\leftarrow$ VRAI
28: **fin si**
29: **jusqu'à** Continu $=$ FAUX

Confrontation des positions fixées (Algorithme 8) Lorsque l'algorithme arrive à cette partie, toutes les couleurs peuvent indépendamment être de span 1, il faut donc vérifier que les positions établies pour chaque couleur et pour chaque arête ne sont pas en conflit.

Un conflit peut survenir si deux couleurs doivent avoir la même position sur une arête (lignes 3-6). Si l'arête contient une couleur libre et une positionnable, alors la position de la couleur libre est imposée par celle de l'autre couleur (lignes 7-8). Lorsque les deux couleurs partageant une arête sont libres, leurs positions sont décidées arbitrairement (lignes 9-13).

Algorithme 8 Confrontation des positions fixées.

```
1:  pour tout e ∈ E_R telle que |e.couleur| = 2 faire
2:    soit {c_1, c_2} = e.couleur
3:    si e.statut(c_1) = POSITIONNABLE et e.statut(c_2) = POSITIONNABLE alors
4:      si e.position(c_1) = e.position(c_2) alors
5:        retourner FAUX et e.couleur et e
6:      fin si
7:    sinon si e.statut(c_1) = POSITIONNABLE et e.statut(c_2) = LIBRE alors
8:      e.position(c_2) ← e.position.oppose(c_1)
9:    sinon si e.statut(c_1) = LIBRE et e.statut(c_2) = LIBRE alors
10:     soit v ∈ e.extremite arbitraire
11:     e.position(c_2) ← v
12:     e.position(c_1) ← e.position.oppose(c_2)
13:   fin si
14: fin pour
```

Annexe C

Exemples de réseaux utilisés dans la littérature

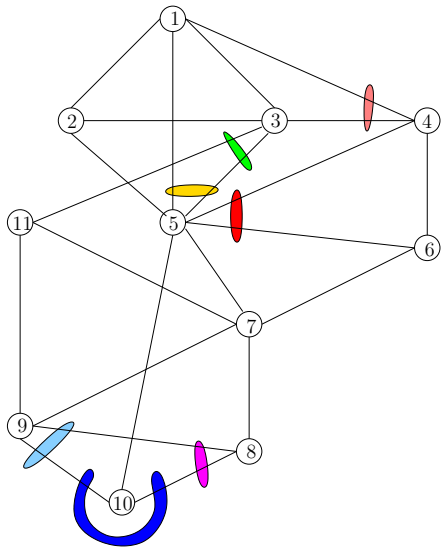


Fig. C.1 – NJLATA [CSC02, DS04a]

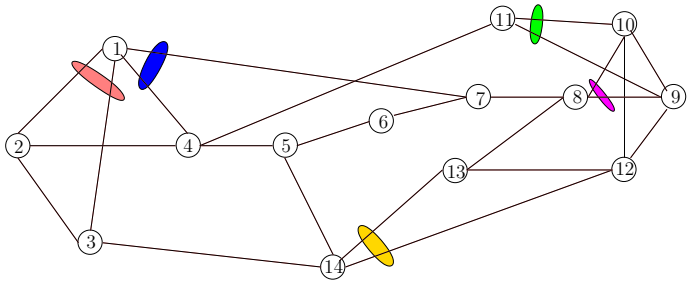


Fig. C.2 – NSFNET [OSYZ95, Jau, YDA00]

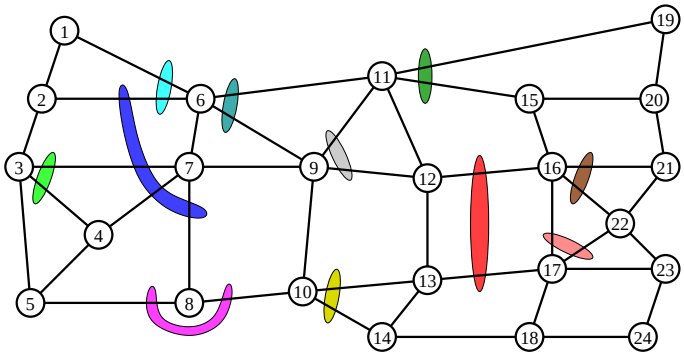


FIG. C.3 – Réseau Américain [SYR05, TR04b]

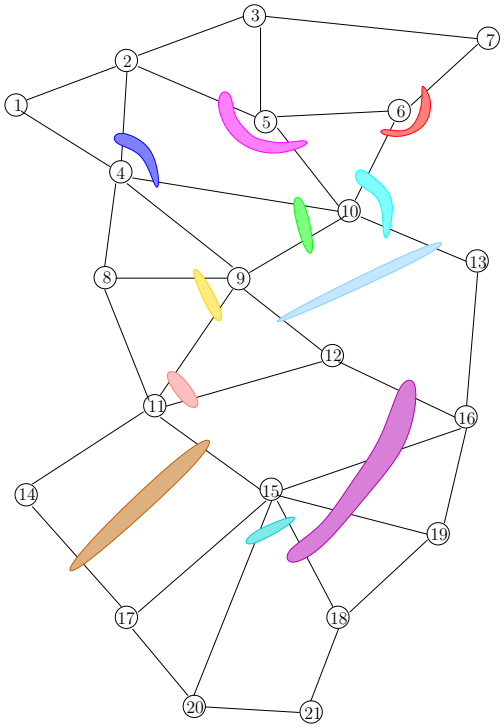


FIG. C.4 – Réseau Italien [SYR05]

Résumé

Les problèmes étudiés dans cette thèse sont motivés par des questions issues de l'optimisation des réseaux de télécommunication. Nous avons abordé ces problèmes sous deux angles principaux. D'une part nous avons étudié leurs propriétés de complexité et d'inapproximabilité. D'autre part nous avons dans certains cas proposé des algorithmes exacts ou d'approximation ou encore des méthodes heuristiques que nous avons pu comparer à des formulations en programme linéaires mixtes sur des instances particulières.

Nous nous intéressons aussi bien aux réseaux de cœur qu'aux réseaux d'accès. Dans le **premier chapitre**, nous présentons brièvement les réseaux d'accès ainsi que les réseaux multiniveaux de type IP/WDM et l'architecture MPLS que nous considérons pour les réseaux de cœur. Ces réseaux sont composés d'un niveau physique sur lequel est routé un niveau virtuel. A leur tour les requêtes des utilisateurs sont routées sur le niveau virtuel. Nous abordons également la tolérance aux pannes dans les réseaux multiniveaux qui motive deux problèmes que nous avons étudiés.

Le **second chapitre** est consacré à la conception de réseaux virtuels. Dans un premier temps nous modélisons un problème prenant en compte la tolérance aux pannes, puis nous en étudions un sous-problème, le groupage. Notre objectif est de minimiser le nombre de liens virtuels, ou tubes, à installer pour router un ensemble de requêtes quelconque lorsque le niveau physique est un chemin orienté. Le **troisième chapitre** traite des groupes de risque (SRRG) induits par l'empilement de niveaux au sein d'un réseau multiniveaux. Grâce à une modélisation par des graphes colorés, nous étudions la connexité et la vulnérabilité aux pannes de ces réseaux. L'objet du **quatrième chapitre** est le problème du placement d'instruments de mesure du trafic dans le réseau d'accès d'un opérateur. Nous considérons aussi bien les mesures passives qu'actives. La surveillance du trafic possède de nombreuses applications, en particulier la détection de pannes et l'évaluation des performances d'un réseau.

Abstract

This thesis is devoted to optimization problems arising in telecommunication networks. We tackle these problems from two main points of view. On the one hand we study their complexity and approximability properties. On the second hand, we propose heuristic methods, approximation algorithms or even exact algorithms that we compare with mixed integer linear programming formulations on specific instances.

We are interested in backbone networks as well as access networks. In the **first chapter**, we briefly present access networks and IP/WDM multilayer backbone networks using the MPLS architecture. These networks are composed of a physical layer on which is routed a virtual layer. In turn, the users' requests are routed on the virtual layer. We also present multilayer network survivability issues motivating two of the questions we have studied.

The **second chapter** is dedicated to the design of virtual networks. First we propose a mixed integer linear programming formulation with network survivability constraints. Then we study a sub-problem, the grooming problem. Our objective is to minimize the number of virtual links, needed to route a given set of requests when the physical layer is a directed path. The **third chapter** deals with Shared Risk Resource Groups (SRRG) induced by stacking up network layers in multilayer networks. Thanks to the colored graphs model, we study connexity and failure vulnerability of these networks. The positioning of active and passive traffic measurement points in the access network of an internet service provider is the subject of the **fourth chapter**.