Réseaux optiques à multiplexage en longueur d'onde : une étude sur le placement des convertisseurs

Rapport de stage
1ère année du Magistère Informatique et Modélisation
École Normale Supérieure de Lyon

Benjamin Wack

Encadrants :
Jean-Claude Bermond
David Coudert

5 juin - 15 juillet 2000

Table des matières

Introduction

Ce stage s'est déroulé au sein du projet MASCOTTE (Méthodes Algorithmiques, Simulation et Combinatoire pour l'OpTimisation des TÉlécommunications), à l'INRIA Sophia-Antipolis, sous la direction de David Coudert et Jean-Claude Bermond et avec les précieux conseils de Stéphane Pérennes. Les recherches menées au sein de ce projet concernent donc essentiellement la conception de divers réseaux : constellations de satellites, réseaux de fibres optiques, etc. Elles donnent généralement lieu à des partenariats avec des industries, c'est le cas du projet RNRT PORTO (Planification et Optimisation des Réseaux de Transport Optiques), en collaboration avec Alcatel et France Telecom R&D (ex CNET), dans lequel s'intègre ce stage.

L'utilisation des fibres optiques permet des communications rapides, même dans des réseaux longue distance depuis que l'on sait acheminer correctement les faisceaux lumineux sans perte de puissance. L'introduction du multiplexage en longueur d'onde (WDM pour Wavelength Division Multiplexing) dans le but d'augmenter les capacités de transmission mais également la rentabilité des systèmes conduit de plus en plus à une optimisation des systèmes existants et à une meilleure intégration des systèmes de nouvelle génération. L'objectif de ce stage est de contribuer au choix des routeurs à installer pour utiliser les fibres déjà présentes et mettre en place un réseau WDM, tout en respectant les contraintes établies. On s'intéressera plus particulièrement au cas des convertisseurs, qui autorisent une information à être véhiculée par plusieurs longueurs d'onde différentes le long de son parcours. Cet équipement optionnel des routeurs permet de supprimer beaucoup de contraintes et donc de minimiser l'utilisation des ressources, mais coûte très cher en installation et maintenance.

Le chapitre 1 décrit de manière globale l'état des connaissances actuelles sur les réseaux WDM, en mettant l'accent sur les convertisseurs à chaque fois que cela est possible. On commence donc par décrire les équipements physiques de ces réseaux. Ceci permet de se faire une bonne idée des performances susceptibles d'être atteintes, mais aussi de montrer la diversité et la complexité des problèmes qui se posent. Le principe général est d'affecter à chaque requête une longueur d'onde, sans que deux requêtes se partagent la même longueur d'onde le long d'une fibre. La conversion permet de changer la longueur d'onde d'une requête au milieu de son routage. On n'entre cependant pas trop dans les détails techniques car ils concernent plutôt la recherche dans le domaine de l'optique.

On formalise ensuite le problème de manière précise pour en donner une modélisation mathématique appropriée. Dans celle-ci, le réseau est un graphe orienté dont les arcs représentent les fibres, et les communications sont des chemins le long de ces arcs. Un indice important est la charge d'un réseau, c'est-à-dire le volume maximum de communications pris sur chacune de ses fibres. L'affectation des longueurs d'onde se traduit par un problème de coloriage des requêtes, d'autant plus naturellement que les longueurs d'onde employées sont dans le domaine du visible. On peut ensuite définir un graphe des conflits, dont les sommets représentent les requêtes et sont reliés par une arête si les requêtes correspondantes empruntent une même fibre. Cette modélisation n'est pas immédiatement adaptée à la conversion, on décrit donc rapidement le principe de formalisation qui sera posé rigoureusement dans le chapitre 2.

Pour finir cet état de l'art, on donne les problèmes les plus généraux et les résultats existants dans le domaine de la WDM, et plus particulièrement les résultats qui pourront être utiles dans le domaine de la conversion. On énonce ainsi des relations entre la charge d'un réseau et le nombre chromatique du graphe des conflits associé. Ensuite, des résultats plus précis sur des réseaux particuliers sont recensés, notamment des algorithmes polynomiaux de coloration donnant un nombre chromatique borné par un facteur de la charge : 1 pour les subdivisions d'étoile, 5/3 pour les arbres. On finit par donner un aperçu des travaux déjà menées dans le domaine spécifique de la conversion. Sont évoquées une approche probabiliste très différente de celle qu'on aura par la suite, ainsi qu'une étude d'une catégorie particulière de convertisseurs, plus limités mais moins coûteux. On cite également un article proche du cas que l'on va étudier, qui prend pour seule entrée un graphe physique et cherche un placement des convertisseurs valable pour une instance de communication quelconque. Ses résultats sont des bornes supérieures sur le nombre de convertisseurs nécessaires.

Dans le chapitre 2, le problème de placement des convertisseurs est abordé de manière théorique. Pour commencer, on formalise précisément la conversion dans un réseau, et on peut alors donner une formulation mathématique de la question posée. Le but de notre étude sera d'utiliser un minimum de convertisseurs pour trouver une coloration du graphe des conflits en un nombre de couleurs n'excédant pas la charge du réseau. Une solution approchée respectera cette condition chromatique mais se contentera d'un nombre non minimal de convertisseurs.

Pour mieux comprendre ce qui fait qu'une instance de communications demande plus de couleurs que la charge du réseau, on s'intéresse ensuite à la construction de cas pathologiques pour lesquels le nombre chromatique devient grand devant la charge. Une méthode consiste à inclure dans le graphe des conflits des configurations réputées pour faire augmenter le nombre chromatique : cycles de longueur impaire, clique maximum de taille plus ou moins arbitraire. On obtient ainsi un nombre chromatique au choix linéaire en la charge, quadratique, ou encore totalement indépendant. Il est intéressant de constater que les graphes sur lesquels on montre ces résultats ont des topologies bien connues : arbre, cycle, chemin ou grille.

On démontre enfin la $\mathcal{NP}$-complétude du problème, qu'on pouvait supposer assez naturellement étant données la complexité habituelle des problèmes de WDM et la présence d'une étape de coloration. Cela dit, la réduction s'effectue à partir du problème de l'ensemble indépendant maximum MAXISET ; la complexité montrée ici repose donc plus sur le placement des sites de conversion que sur le coloriage du graphe des conflits. En effet, on part d'une instance de MAXISET et on complique un peu le graphe d'entrée pour construire le graphe physique de notre problème, dans lequel on place de petits morceaux de routages dont on connaît bien la charge et le nombre chromatique.

Le chapitre 3 est consacré à la description et à l'évaluation d'une nouvelle heuristique. Son fonctionnement global est le suivant : on place des convertisseurs dans tous les nœuds du réseau, puis on "abandonne" ces conversions les unes après les autres en recoloriant le graphe des conflits au fur et à mesure. Un avantage certain est qu'on profite généralement de la coloration précédente pour effectuer la suivante. Le processus d'abandon se termine quand la recoloration demande d'augmenter le nombre chromatique.

Cet algorithme repose sur deux procédures critiques qui sont le choix des sites d'abandon et la recoloration locale du graphe des conflits modifié. On donne donc plusieurs versions de ces procédures et on compare leurs complexités et les performances qu'elles sont susceptibles d'atteindre. Le choix des sites peut se faire de manière exhaustive, avec une complexité moins importante si on emploie la technique du branch-and-bound, ou bien de manière gloutonne, plus rapide mais moins proche de l'optimal. On montre rapidement comment améliorer les performances de l'approche gloutonne en triant préalablement les sommets dans un certain ordre. La recoloration locale, quant à elle, doit largement privilégier l'efficacité car elle est effectuée un très grand nombre de fois. On donne donc une version dont la rapidité dépend de la taille du graphe, et une autre qui dépend plus de la charge et donc du volume de l'instance de communications.

Cette heuristique n'offrant aucune garantie mathématique, on la teste sur diverses instances du problème. D'une part on se place sur des graphes bien connus afin de comparer les performances avec les résultats théoriques connus, d'autre part on crée un modèle des réseaux réels et de leurs instances de communication. Ce modèle réutilise un algorithme de génération de graphes utilisés par d'autres laboratoires pour faire des tests de coloration ; il a fallu de plus implémenter un générateur de routages utilisant les plus courts chemins et donnant des volumes de communication assez réalistes. On teste plusieurs ordres de grandeur, pour la taille du graphe comme pour le volume des communications. Les temps de calcul sont assez bons, le facteur limitant étant surtout la mémoire, et les performances obtenues sont satisfaisantes.

Ainsi notre algorithme colorie totalement les modèles de réseaux réels tant que ceux-ci restent manipulables avec la mémoire disponible, c'est-à-dire que le nombre d'arêtes du graphe des conflits ne dépasse pas la dizaine de millions. Pour les graphes qui sont connus pour nécessiter peu ou pas de conversion, on parvient également à de bons résultats, puisque notre algorithme atteint toujours les bornes inférieures. Enfin, pour certains cas plus problématiques comme ceux montrés dans la section 2.2, on arrive à des résultats satisfaisants : le nombre de convertisseurs employé est en général du même ordre que les bornes infrieures connues sur l'optimal (ou du même ordre que l'optimal lui-même quand on le connaît).

Chapitre 1

La technique WDM

1.1 Technologies actuelles

Cette section passe en revue les équipements qui sont actuellement mis en vente par les constructeurs, afin de donner un aperçu des fondements du problème et de la diversité des facteurs à prendre en compte.

1.1.1 Généralités

L'utilisation des fibres optiques dans les réseaux de télécommunications est une idée relativement récente, qui a pris tout son essor dès qu'il a été possible de maintenir un signal lumineux sur une longue distance. Les techniques les plus récentes tendent à augmenter les capacités de transmission et d'acheminement des fibres en permettant de transmettre plusieurs signaux par le même système : elles sont regroupées sous le nom générique de **multiplexage**. La plus simple reste le multiplexage temporel, où chaque signal dispose à tour de rôle d'une courte plage de temps pour s'exprimer.

Elle est cependant limitée car les circuits existants ne sont pas capables d'assumer un traitement satisfaisant pour des débits d'information trop élevés. On a donc recours à une technologie plus récente : le multiplexage en longueur d'onde (souvent appelé WDM pour Wavelength Division Multiplexing). Il consiste à affecter une couleur différente à chaque signal et à former un signal lumineux pour toute la fibre en mixant les signaux sur tout le spectre de transmission. La seule restriction est donc que deux informations différentes ne doivent pas emprunter la même longueur d'onde dans la même fibre, pour éviter tous les problèmes d'interférence optique.

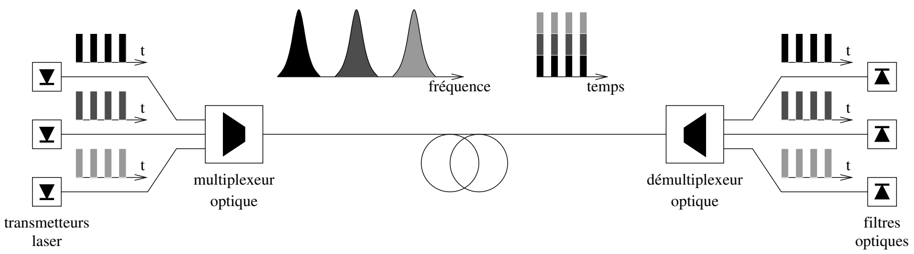

Fig. 1.1: Multiplexage en longueur d'onde (WDM)

Donnons un ordre de grandeur de ce que permettent les équipements actuels pour montrer leur efficacité. Pour repère, les réseaux de téléphone actuels fonctionnent à 64 kbit/s, alors que les fibres optiques les plus répandues montent à quelques centaines de Mbit/s *par longueur d'onde* ; les plus récentes enfin sont capables de performances de l'ordre des 10 Gbit/s, par longueur d'onde toujours.

L'intérêt du multiplexage est d'augmenter encore considérablement ces capacités. En effet les fibres des réseaux existants supportent généralement jusqu'à 32 longueurs d'ondes distinctes, et celles produites actuellement devraient en accepter plusieurs centaines. Avec les équipements les plus récents, le débit cumulé sur une fibre dépasse donc le Térabit/s [NST+99]. Le nombre de longueurs d'onde disponibles est critique car il constitue une borne du nombre de communications qui peuvent emprunter la même fibre.

Les réseaux WDM sont constitués de fibres optiques, et de nœuds servant d'une part au routage, d'autre part aux entrées et sorties de signaux dans les fibres. Toutes les études effectuées sur le sujet ont pour but de minimiser un coût total du réseau, qui pourra dépendre à la fois du prix de fabrication et d'entretien du matériel, du taux d'utilisation de ce matériel et du nombre de communications qui sont ou non correctement acheminées.

Le problème est rendu très vaste par la diversité des équipements que l'on peut trouver dans les nœuds :

- Multiplexeur à insertion/extraction : Ceux-ci sont utilisés pour permettre aux nœuds du réseau d'émettre ou de recevoir de l'information : leur utilisation est donc fixée par les demandes de communications. Cependant, sur certaines architectures, il est moins coûteux d'insérer une longueur d'onde sur un nœud si on doit déjà l'extraire, et réciproquement. Certaines études [GLS98, LLWF00] se sont donc intéressées à la minimisation du nombre de ces équipements.
- Répartiteur : Il permet uniquement de connecter deux à deux les fibres d'entrée et les fibres de sortie. Cet équipement est donc la base de tous les nœuds. Son coût est minime mais il oblige toutes les informations véhiculées par une même fibre d'entrée à ressortir par le même lien.
- Brasseur : Plus flexible mais aussi plus cher, il comporte des appareils de multiplexage et de démultiplexage. En pratique, cela permet de séparer les différentes longueurs d'onde arrivant par chaque fibre entrante et de les rediriger, indépendamment les unes des autres, sur d'autres fibres de sortie. Un modèle un peu moins coûteux se contente de séparer le signal entrant en *bandes* de longueurs d'onde, regroupant en général chacune un quart des informations transitant par la fibre.
- Convertisseurs : Cet équipement permet de faire passer un signal d'une longueur d'onde à une autre. Les plus classiques sont opto-électroniques, ce qui implique de retraduire le signal en impulsions électriques, mais on a développé récemment des convertisseurs tout-optiques [YZP+00], plus efficaces en termes de débit, de puissance consommée et de fiabilité du signal. C'est à ces derniers que nous allons nous intéresser plus particulièrement.

Enfin, un dernier critère est à prendre en compte pour évaluer le coût d'un réseau : on cherche au maximum à réutiliser des systèmes existants. En effet, si ceux-ci ne sont pas suffisants pour mettre en place le routage qu'on a choisi, il faudra en installer de nouveaux, ce qui revient bien plus cher. En plus d'être multicritère, le coût n'est donc pas toujours linéaire, ce qui fait de l'optimisation des résaux un problème très difficile.

1.1.2 Conversion dans les réseaux WDM

Notre étude consistera plus spécifiquement en une recherche du placement optimal des convertisseurs. Ces équipements permettent de changer plus ou moins arbitrairement la longueur d'onde utilisée par une requête. La longueur des chemins est alors diminuée, et donc leur coloration a des chances de se simplifier. Ils restent cependant un équipement très coûteux, qu'il faut donc utiliser au mieux pour ne pas devoir en installer plus que nécessaire.

Bien que le problème de la conversion ne soit pas le plus étudié parmi ceux que pose le multiplexage en longueur d'onde, il en existe plusieurs versions assez distinctes, dues notamment à la diversité des technologies.

On distingue en effet trois types de conversion dans un nœud :

- Conversion fixée : À chaque longueur d'onde entrante ne peut être associée qu'une unique longueur d'onde sortante, mais pas nécessairement la même.
- Conversion limitée : Ce modèle est plus souple, car tout ou partie des longueurs d'onde entrantes est susceptible d'être redirigée sur tout ou partie des sortantes. On les utilise en général de la manière suivante : seul un petit nombre de longueurs d'onde peuvent être redirigées, mais de manière totalement arbitraire.
- Conversion totale : C'est le cas le plus général et le plus puissant : toutes les redirections sont autorisées. C'est bien sûr également le plus coûteux.

Nous regarderons donc les diverses problématiques de la WDM et les résultats connus dans ce qu'ils ont d'intéressant pour l'étude de la conversion, puis nous nous focaliserons sur un problème de placement des convertisseurs. On explique maintenant comment ces problèmes peuvent être formalisés par la théorie des graphes et de leur coloration.

1.2 Modélisation mathématique

De manière assez naturelle, on représente le réseau par un graphe :

Définition 1 (Graphe physique)
À un réseau réel on associe un graphe $G = (V, E)$ orienté, dit graphe physique. *Les sommets V du graphe sont les nœuds du réseau, et E est constitué de la manière suivante : il existe un arc entre deux sommets de V s'il existe une fibre qui relie directement ces deux nœuds dans le réseau.*

On supposera par la suite que G est connexe sans perte de généralité puisque dans le cas contraire il suffirait d'étudier les différentes composantes connexes séparément. Bien qu'il y ait souvent plusieurs fibres entre deux mêmes nœuds du réseau, on ne considérera qu'un arc pour ces fibres car faire de G un multigraphe n'apporte pas grand-chose au modèle.

Souvent, comme c'est le cas dans la plupart des réseaux réels, le graphe est orienté symétrique, ce qui signifie mathématiquement que $\forall u, v \in V, \{(u, v) \in E \Rightarrow (v, u) \in E\}$. S'il n'est pas ambigu qu'on est dans ce cas, pour simplifier les figures, on dessinera toujours un unique arc non-orienté à la place des deux arcs symétriques.

On désire ensuite établir un flux d'informations sur ce réseau :

Définition 2 (Instance de communication)
Celle-ci définit le volume des informations échangées par les sommets de V. Formellement, une instance I est un ensemble de requêtes de connexion, qui sont représentées chacune par une paire (ordonnée) de sommets . Un codage simple de I est la matrice des demandes, *dont les coefficients sont positifs éventuellement nuls et donnent le nombre de requêtes pour chaque couple.*

Les réseaux WDM fonctionnent en mode connecté, ce qui signifie qu'une communication entre deux nœuds est établie une fois pour toutes. Les instances de communication sont donc fixées à l'avance et les problèmes peuvent être résolus de manière statique. Il existe cependant des études sur le routage dynamique, dans lequel le trafic est susceptible de changer à tout moment et de manière imprévisible, mais nous ne les considérerons pas ici.

De nombreuses études ont été menées sur des cas particuliers d'instances. On pourra se reporter à [Bea00], section “Communications structurées”, pour les définitions des instances les plus fréquentes.

Le graphe physique et l'instance de communication constituent l'entrée des problèmes les plus généraux posées par le multiplexage. Ceux-ci suivent toujours le même principe : pour chaque requête (u, v) exprimée dans I, il faut trouver d'une part un chemin de u à v dans G, d'autre part affecter une couleur à ce chemin. La contrainte de coloration est la suivante : deux chemins qui empruntent un même arc ne reçoivent pas la même couleur. Dans la plupart des cas, on ne traite qu'un sous-problème, en enrichissant les entrées ou bien en demandant une sortie moins complète.

Pour choisir entre les diverses solutions à ces problèmes, on cherche à minimiser une fonction de coût plus ou moins évoluée, allant du simple nombre de couleurs utilisées à un coût multicritère complexe à évaluer.

Pour faciliter la résolution, on définit divers outils. Commençons par deux constructions utiles :

Définition 3 (Routage)
Étant donnés un graphe G et une instance I, un routage R acceptable pour I consiste à associer à chaque requête de I un chemin empruntant les arcs de G et ayant pour extrémités les sommets de cette requête.

Par abus de langage, on désigne généralement par R l'ensemble des chemins considérés. Dans le cas où plusieurs requêtes de I ont les mêmes extrémités, on peut selon les cas trouver dans R des chemins différents, ou identiques mais multiples. R est donc en fait un multi-ensemble. Enfin, dans tous les problèmes que l'on considérera, il sera assez évident que l'on peut forcer les chemins de R à être élémentaires sans perte de généralité ni d'optimalité. On considérera donc que c'est toujours le cas.

Définition 4 (Graphe des conflits)
C'est un graphe non orienté qui dépend de G et de R (et par conséquent de I). Il n'existe pas de notation canonique, on choisit de l'appeler par la suite $H(G, R)$ ou plus simplement $H(R)$ (resp. H) s'il n'y a pas d'ambiguïté sur le graphe physique (resp. sur le graphe physique et sur le routage).

À chaque chemin de R est associé un sommet dans H, et deux sommets de H sont reliés si et seulement si les chemins correspondants empruntent un même arc dans G (ils sont alors dits en conflit*).*

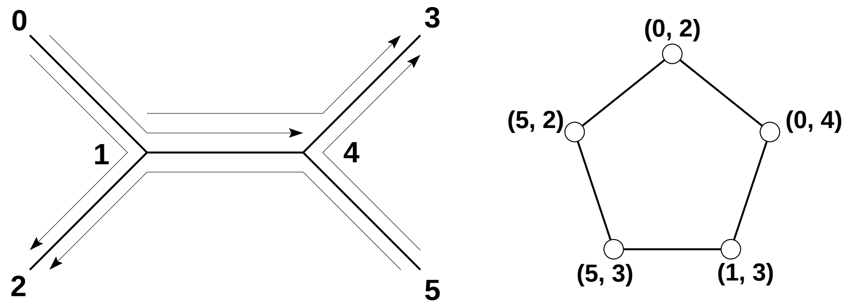

FIG. 1.2: Une instance simple dans un arbre et le graphe des conflits associé

Il est facile de voir que la phase de coloration du routage optique (voir problème 1 plus loin) revient exactement à effectuer une coloration classique des sommets de H : deux chemins de R doivent être coloriés de couleurs différentes si et seulement si ils sont en conflit, donc si et seulement si les sommets correspondants dans H sont voisins ; or c'est exactement le cas où ces deux sommets doivent être coloriés différemment.

Le graphe des conflits offre une façon de modéliser les convertisseurs : pour chaque chemin que l'on décide de convertir, on remplace le sommet correspondant dans H par deux sommets qui représentent les deux moitiés du chemin. Chacun de ces deux nouveaux sommets n'est alors plus adjacent qu'aux sommets qui correspondent aux chemins avec lesquels il reste en conflit (en particulier, les deux nouveaux sommets peuvent rester adjacents à un même troisième dans certains cas).

On définit enfin des notations pour les grandeurs le plus souvent manipulées :

Définition 5 (Charge d'un arc, d'un routage, d'une instance)
Étant donnés un graphe physique $G = (V, E)$, une instance de communication I et un routage acceptable R, pour tout arc (u, v) de G on définit sa charge $\pi(G, I, R, (u, v))$ comme le nombre de chemins de R qui empruntent cet arc.

On peut alors parler de la charge du routage : $\pi(G, I, R) = \max_{(u,v)\in E} \pi(G, I, R, (u, v))$ ainsi que de la charge optimale pour l'instance I : $\pi(G, I) = \min \pi(G, I, R)$ sur tous les routages R acceptables pour I.

Définition 6 (Nombre chromatique d'un routage, d'une instance)
On notera $w(G, I, R)$ le nombre minimum de couleurs nécessaires pour colorier de manière satisfaisante tous les chemins de R, et on l'appellera nombre chromatique. De même que pour la charge, on définit $w(G, I) = \min_R w(G, I, R)$ le nombre chromatique optimal pour l'instance I.

Voyons maintenant les questions qui se posent et les réponses, complètes ou partielles, qui y ont été apportées dans la littérature.

1.3 Principaux résultats existants

Les résultats énoncés ici sont le fruit d'un travail de bibliographie sur les réseaux WDM. On a retenu d'une part les problèmes généraux pour montrer les différentes pistes de recherche dans ce domaine, d'autre part les résultats connus pouvant servir à l'étude du problème de placement des convertisseurs.

1.3.1 Problèmes généraux de la WDM

On donne ici un aperçu des relations connues entre G, I, R, w et π.

Problème 1 (Problème du routage optique)
Pour un graphe physique G orienté ou non et une instance de communication I, on cherche un routage R qui minimise le nombre chromatique $w(G, I, R)$.

Le problème de décision associé a été montré $\mathcal{NP}$-complet par Erlebach et Jansen dans [EJ96], et ce même pour des graphes de topologies assez simples comme les arbres, les cycles ou les grilles. Ce problème est très difficile car il demande successivement de choisir un routage R parmi tous ceux existants, puis d'évaluer $w(G, I, R)$, ce qui demande une coloration des chemins. On le décompose donc en plusieurs sous-problèmes :

Problème 2 (Problème de la charge)
Pour un graphe physique G orienté ou non et une instance de communication I, on cherche un routage R qui minimise la charge $\pi(G, I, R)$.

Le problème de décision associé, consistant à déterminer si $\pi(G, I) \leq p$ pour un entier p, se ramène à un problème de multiflot entier. Il a donc largement été traité, et sa $\mathcal{NP}$-complétude découle des études de Even, Itai et Shamir sur les flots entiers (voir [Bea00]).

Problème 3 (Problème de la planification)
Pour un graphe physique G orienté ou non, une instance de communication I et un routage fixé *R, on cherche $w(G, I, R)$; le problème de décision associé consiste à déterminer si $w(G, I, R) \leq p$.*

Ce problème est intéressant car sa complexité diffère selon qu'on se place dans le cas orienté ou pas. En effet, le problème notoirement $\mathcal{NP}$-complet de la coloration des arêtes d'un multigraphe se réduit bien au modèle non-orienté, sauf dans des cas simples comme les arbres de degré borné par exemple.

Contrairement à ce qu'on pourrait penser, dans le cas orienté, le problème est en général moins difficile. La topologie assez simple de l'étoile (arbre dont tous les sommets sauf un sont de degré un) est un exemple parlant : le cas non-orienté est $\mathcal{NP}$-complet [GJ85] alors qu'en orienté il se réduit à un problème de couplage dans un biparti, soluble polynomialement d'après le théorème de König-Hall (voir [Ber83] par exemple).

Dans toute la suite du rapport, si rien n'est précisé, on considérera un graphe orienté symétrique.

Montrons maintenant comment les quantités w et π sont liées. Il est évident que, pour tous G, I et R, $\pi(G, I, R) \leq w(G, I, R)$ car on doit utiliser π couleurs distinctes sur l'arc de charge π. On en déduit facilement que $\pi(G, I) \leq w(G, I)$ car pour le routage R_w qui réalise le nombre chromatique minimal on a :

$$\pi(G, I) = \min_R \pi(G, I, R) \leq \pi(G, I, R_w) \leq w(G, I, R_w) = w(G, I)$$

Cependant, pour résoudre le problème du routage optique, il ne suffit pas de traiter successivement le problème de la charge puis le problème de la planification. En effet :

Proposition 1 [Bea00] *Pour un graphe G et une instance I, il n'existe pas toujours de routage R tel que $\pi(G, I, R) = \pi(G, I)$ et $w(G, I, R) = w(G, I)$.*

Pour s'abstraire de cette difficulté, on peut s'intéresser à un problème plus précis : comment caractériser les cas où $w = \pi$? En effet un routage optique optimal est alors forcément une solution du problème de la charge. On peut aussi essayer de trouver un routage optique dont le nombre chromatique est borné par un facteur de la charge.

Une approche souvent utilisée consiste à étudier des instances particulières. Il a par exemple été démontré [BHP98] que pour toute instance de *multicast*, c'est-à-dire où toutes les requêtes partent d'un même sommet, le problème du routage optique est polynomial et qu'alors $w = \pi$.

Dans le cas de l'instance d'échange total, *chaque* nœud du réseau envoie la même information à *tous* les autres. L'égalité $w = \pi$ a été prouvée pour cette instance dans de nombreux types de graphes [Bea99], et il est conjecturé qu'elle se vérifie pour tous les graphes orientés symétriques.

L'autre grande approche fixe la topologie du graphe physique mais pas l'instance de communication. On prouve ainsi par exemple que le problème de la charge est polynomial sur tout cycle orienté. Citons ici un résultat important sur l'égalité $w = \pi$.

Théorème 1 [Bea00] *Soit un arbre T orienté symétrique : l'égalité $w(T, I) = \pi(T, I)$ est vraie pour toute instance I si et seulement si T est une subdivision d'étoile (ce qui signifie que tous ses sommets sauf un sont de degré un ou deux). C'est encore vrai pour un graphe G qui n'est pas un arbre si on impose certaines conditions sur le routage (dans T, il n'existe qu'un unique chemin pour chaque paire de sommets donc R est fixé).*

On peut donner une esquisse de la preuve. Si G est une subdivision d'étoile, alors l'égalité s'obtient en combinant deux résultats plus simples : $w(G, I) = \pi(G, I)$ quelle que soit I si G est une étoile, et aussi si G est un chemin. La démonstration du premier a été citée à propos de la difficulté du problème de la planification (problème 3) ; le second s'obtient par une coloration gloutonne car c'est exactement le problème du *graphe d'intervalles*. Enfin la réunion des deux se fait en appliquant d'abord l'égalité pour l'étoile qui se trouve au centre de G, puis pour tous les chemins que sont les branches de G.

La réciproque se voit facilement : si deux sommets u et v sont de degré supérieur ou égal à trois, comme on est dans un graphe connexe, il existe un sous-graphe de G du même genre que celui de la figure ci-contre, sur laquelle u=1 et v=4. On pose alors l'instance I (et le routage correspondant si G n'est pas un arbre) représentée ici, et on constate que $\pi = 2$ mais $w = 3$.

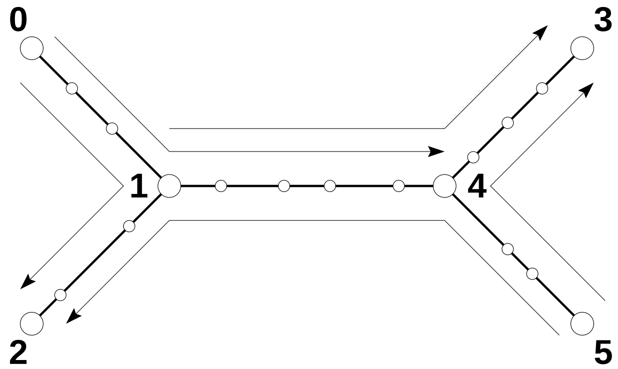

FIG. 1.3: Une instance demandant plus de couleurs que la charge

On a enfin des résultats d'approximation assez puissants pour des topologies un peu plus générales. L'un des plus utilisés est celui-ci :

Théorème 2 [EJK+99] *Le problème du routage optique dans un arbre peut être résolu polynomialement avec $\lceil 5\pi/3 \rceil$ couleurs seulement.*

Le lecteur intéressé par plus d'informations sur la technique WDM ou par une synthèse des résultats théoriques obtenus dans ce domaine pourra consulter la thèse de Beauquier [Bea00].

1.3.2 Résultats existants dans le domaine de la conversion

On s'intéresse ici aux travaux effectués plus spécifiquement à propos des convertisseurs. Comme on le verra, ils sont assez variés car il existe plusieurs modèles mathématiques mais aussi à cause de la diversité des équipements.

Une approche souvent employée est probabiliste. Elle consiste à décrire le trafic dans le réseau comme un ensemble de variables aléatoires bien choisies, et à calculer, pour une répartition donnée des convertisseurs, la probabilité de pouvoir acheminer toutes les requêtes sans utiliser plus de longueurs d'onde que nécessaire. Les travaux de Subramaniam, Azizoğlu et Somani sont les plus poussés dans ce domaine. Ils ont étudié essentiellement la conversion éparse : on s'autorise la conversion totale mais sur un nombre limité de sites. On leur doit notamment un modèle analytique pour évaluer les performances de cette conversion éparse [SAS96], qui ne s'intéresse pas aux endroits précis où se fait la conversion, mais à la densité globale des convertisseurs sur le réseau. Ils ont d'ailleurs repris ce modèle pour mener des études plus approfondies sur des réseaux simples comme le chemin, le bus ou l'anneau [SAS98].

Une des limites à leurs travaux est qu'ils considèrent généralement un nombre de longueurs d'onde de l'ordre de la dizaine, ce qui est relativement faible comparé aux performances des technologies actuelles. La performance évaluée n'est donc pas le nombre de longueurs d'onde nécessaires au routage, mais le nombre de requêtes correctement acheminées par rapport au volume total des demandes. D'autre part, leurs résultats sont nettement plus significatifs sur des réseaux très réguliers. Il en ressort tout de même une idée générale qui peut donner des perspectives de recherche : *l'efficacité des convertisseurs augmente rapidement avec la densité de ceux-ci dans le réseau, et ce jusqu'à un certain seuil*, au-delà duquel l'amélioration des performances devient négligeable. Pour beaucoup de réseaux, ce seuil correspond à une densité assez élevée (de l'ordre de $0,3$ pour des réseaux peu denses comme l'anneau et $0,6$ pour des réseaux denses comme la grille).

Toujours en s'appuyant sur le même modèle, Thiagarajan et Somani ont écrit un algorithme efficace de placement des convertisseurs [TS99]. Celui-ci repose notamment sur le principe suivant : l'instance de communication est partitionnée en instances de type multicast. On calcule donc l'efficacité des placements de convertisseurs en prenant successivement chacun des sommets comme "source" des requêtes, et on en déduit une mesure globale de l'efficacité. Une évaluation des performances de cet algorithme en fonction de certaines caractéristiques du graphe est donnée.

Une seconde catégorie de travaux a exploré le domaine de la conversion limitée. Ramaswami et Sasaki ont démontré dans [RS98] l'efficacité de la conversion très limitée pour les anneaux et les étoiles, puis les arbres et des réseaux plus arbitraires. Une de leurs principaux conclusions est que la restriction sur le nombre de conversions n'influe pas de manière significative sur les performances obtenues, à condition bien sûr d'autoriser cette conversion limitée sur un nombre de nœuds suffisant.

Cependant, si on examine les données techniques de certains équipements, on se rend compte que, pour un nœud donné, si on décide de faire une conversion, le surcoût nécessaire pour y faire d'autres conversions est relativement bas. La stratégie consistant à effectuer peu de conversions mais dans de nombreux nœuds n'est donc pas forcément optimale.

On aboutit donc à un troisième domaine d'étude, celui que nous allons développer dans la suite. Le but est d'utiliser le moins de convertisseurs possible et de les placer, pour obtenir une coloration des chemins respectant une borne dépendant de la charge du réseau.

Dans [Tog00], Togni s'est intéressé au cas du routage non fixé. Il propose donc, au moyen de la conversion, de partitionner le graphe physique en graphes sur lesquels on a des résultats de coloration. Entre autres, il utilise les théorèmes 1 et 2, et partitionne donc le graphe physique en subdivisions d'étoile (resp. en arbres) pour obtenir une coloration utilisant π (resp. $\lceil 5\pi/3 \rceil$) couleurs. Le placement des convertisseurs est $\mathcal{NP}$-complet dans les deux cas, mais Togni donne des bornes supérieures sur le nombre de sites où il faut faire de la conversion. Enfin il affine ces bornes pour des graphes particuliers (arbre, grille), et propose de chercher des résultats pour des facteurs de π plus grands que $5/3$.

Nous arrêterons ici cet état de l'art et nous considérons désormais le problème sur lequel porte notre étude : le placement optimal des convertisseurs à routage fixé.

Chapitre 2

Le placement optimal des convertisseurs

Ici commence la description des recherches effectuées pendant le stage. On a d'abord formalisé le problème, puis on a cherché à mettre en évidence les points problématiques par la construction de cas particuliers. Pour finir, on a étudié sa complexité et montré sa $\mathcal{NP}$-complétude.

2.1 Délimitation du problème étudié

On donne ici une formulation mathématique de notre problème. Il faut tout d'abord formaliser l'utilisation des convertisseurs : de manière plutôt naturelle, on coupe tous les chemins concernés.

Définition 7 (Routage avec conversion)
Étant donnés un graphe G, un ensemble R de chemins sur G et un sous-ensemble X de sommets de G, on notera R_X l'ensemble des chemins sur G obtenus de la manière suivante :

$$\forall \mu = (u_1, u_2 \dots u_k) \in R, \begin{cases} \text{Si } X \cap \mu = \emptyset, \text{ alors } \mu \in R_X \\ \text{Sinon, choisir } u \in \mu \cap X \\ \qquad \text{et appliquer le même traitement à } (u_1 \dots u) \text{ et } (u \dots u_k) \end{cases}$$

On notera assez naturellement H_X le graphe des conflits associé au problème de planification (G, R_X).

Problème 4 (Placement optimal des convertisseurs à routage fixé)
Étant donnés un graphe physique G, une instance de communication I et un routage R acceptable pour cette instance, déterminer le cardinal d'un plus petit sous-ensemble X de sommets de G tel que $w(G, I, R_X) = \pi(G, I, R_X)$. Le problème de décision associé est le suivant : pour un entier k, existe-t-il X tel que $|X| \leq k$ et $w(G, I, R_X) = \pi(G, I, R_X)$?

Ceci suppose ensuite qu'on détermine X et qu'on donne une coloration des chemins de R_X (donc des sommets de H_X) pour prouver la validité de cette solution.

Dans le cas où on ne cherche pas une solution exacte du problème, on conservera la clause $w(G, I, R_X) = \pi(G, I, R_X)$ mais on pourra se contenter d'un ensemble X non minimal. Remarquons au passage qu'une large majorité des travaux existants faisait le choix inverse, pour des raisons de complexité de la coloration. Les contraintes technologiques incitent cependant à limiter le nombre chromatique le plus possible, étant donnée la capacité limitée des fibres.

Pour mieux cerner le problème, on cherche à caractériser les configurations qui font augmenter le nombre chromatique. Pour cela, on montre ici des cas particuliers d'instances dont le nombre chromatique est supérieur à la charge et on explique quelles sont les raisons de cette écart entre les deux indices.

2.2 Bornes sur le nombre chromatique : cas pathologiques

On a vu rapidement sur l'exemple 1.2 que w et π n'étaient pas toujours égaux. On va essayer, pour généraliser ce résultat, de rendre w grand devant π, c'est-à-dire trouver des familles d'instances (G, I, R) dans lesquels on peut prendre π aussi grand qu'on veut, et pour lesquels il existe une fonction croissante $f : \mathbb{N} \to \mathbb{N}$ telle que $w(G, I, R) \geq f(\pi(G, I, R))$.

Dans le graphe des conflits H, on voit bien que π est un minorant de w, car H comporte pour chaque arc de G une clique de taille égale à la charge de cet arc, et donc H comporte une clique de taille au moins π. L'augmentation de w est due à d'autres configurations de H : on va s'intéresser à ces dernières.

2.2.1 Augmentation de w par création de cycles

Dans l'exemple 1.2, l'augmentation de w est due à la présence d'un cycle de longueur impaire. On peut remplacer chaque chemin de cette instance par k chemins pour faire augmenter l'écart entre w et π. En effet, on obtient alors un routage de charge $2k$, mais qui ne peut être colorié en moins de $\lceil 5k/2 \rceil$ couleurs.

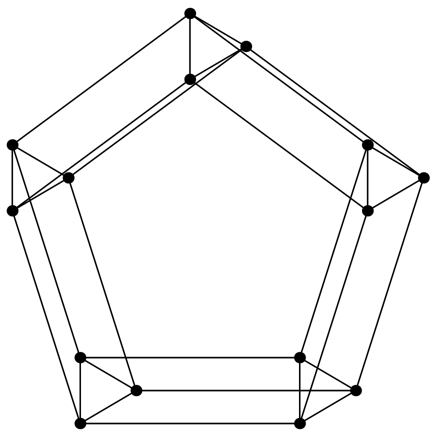

FIG. 2.1: Graphe des conflits obtenu pour $k = 3$; il y a en fait un biparti complet le long de chaque arête du cycle

On a ainsi $w \geq 5\pi/4$. La fonction de domination n'est ici que linéaire, mais ce cas offre un autre intérêt : la clique maximum du graphe des conflits reste de taille égale à la charge. Pour un facteur de domination plus élevé, remarquons qu'on connaît, pout tout π, des routages dans un cycle C_n tels que $w \geq 2\pi + 1$ [Tog00]. Nous allons maintenant vérifier qu'il est aussi possible d'augmenter arbitrairement la taille de la clique maximum par rapport à π.

2.2.2 Augmentation de w par création d'une clique dans le graphe des conflits

La construction utilisée ici est un peu plus complexe, mais elle permet d'obtenir une clique maximum de taille $k\pi$, pour π quelconque et k entier à choisir arbitrairement entre 1 et π. On peut donc s'en servir, au choix, pour obtenir une clique de taille linéaire en π (mais valable uniquement pour $\pi \geq k$) ou bien quadratique en π.

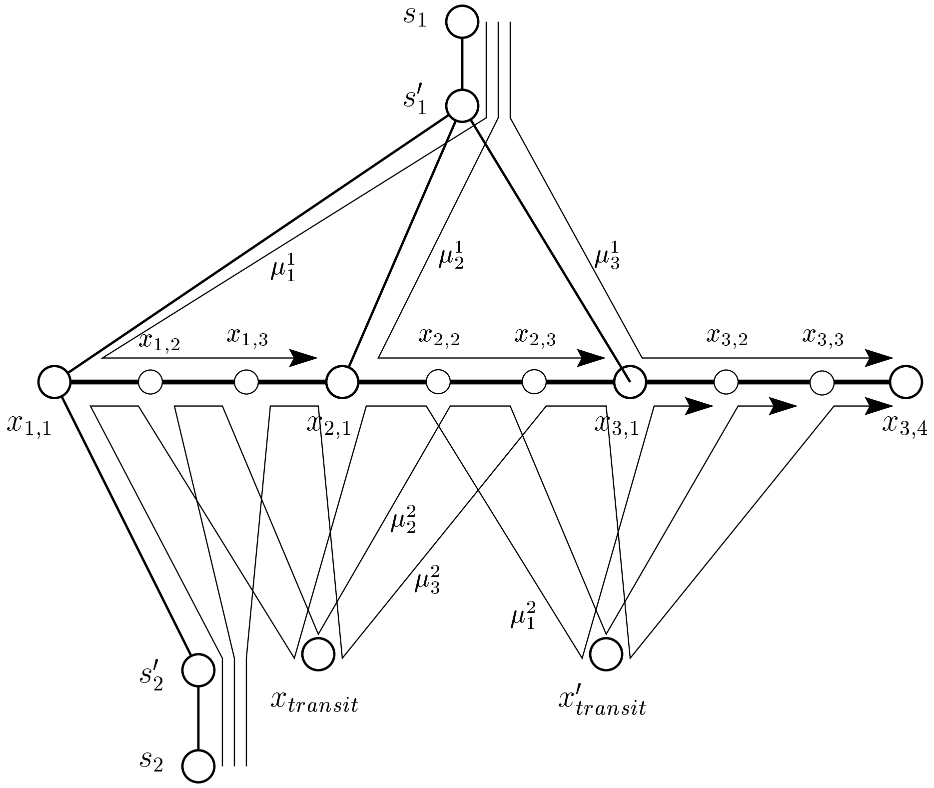

FIG. 2.2: Routage obtenu pour $\pi = 3$ et $k = 2$

- R est formé de k groupes de π chemins $R_1 = \left\{\mu_1^1, \mu_2^1 \dots \mu_\pi^1\right\}, \dots, R_k = \left\{\mu_1^k, \mu_2^k \dots \mu_\pi^k\right\}$.
- Chaque groupe R_i part d'une source s_i et passe par un même sommet s'_i, ce qui assure que tous ses chemins sont en conflit entre eux.
- Les autres conflits vont se faire le long d'un unique chemin X de longueur π^k, dont les sommets sont repérés par k indices et numérotés dans l'ordre lexicographique : $(x_{(1\dots1,1)} \dots x_{(1\dots1,\pi)}, x_{(1\dots2,1)} \dots x_{(1\dots2,\pi)} \quad \cdots \quad x_{(\pi,\pi\dots\pi)})$ plus un dernier sommet $x_{(\pi,\pi\dots\pi+1)}$. De plus, par convention, $x_{(\dots\alpha,\pi+1\dots)} \equiv x_{(\dots\alpha+1,1\dots)}$.
- Le chemin μ_i^j emprunte, dans X et dans le sens de parcours de X, les sous-chemins allant de $x_{(\alpha_1\dots\alpha_{j-1},i,1\dots1)}$ à $x_{(\alpha_1\dots\alpha_{j-1},i+1,1\dots1)}$, pour tous les $(j-1)$-uplets $(\alpha_1 \dots \alpha_{j-1}) \in [1..\pi]^{j-1}$.
- Enfin on crée autant de "courts-circuits" que nécessaire autour de X pour permettre à tous les chemins de R d'être d'un seul tenant. La seule contrainte est que jamais plus de π chemins n'empruntent le même court-circuit, afin que la charge reste de π. La façon la plus simple (mais pour laquelle R est peu réaliste) de le faire est d'ajouter $\pi^{k-1} - 1$ sommets de transit, chacun étant commun aux sommets d'un groupe R_j et leur permettant de passer du chemin $(\alpha_1 \dots \alpha_{j-1})$ au chemin $(\alpha_1 \dots \alpha_{j-1} + 1)$.

On peut par ailleurs faire en sorte que R soit un routage des plus courts chemins (c'est un choix assez fréquent quand on choisit pour le problème du routage optique de chercher successivement R puis une coloration minimale). Cependant, cela demande un nombre de courts-circuits significativement plus important.

Sur la figure 2.2, on a utilisé un minimum de sommets de transit ; pour des raisons de lisibilité, les arcs reliant ces sommets aux sommets du chemin X n'ont pas été représentés car ils correspondent chacun à l'unique chemin qui les emprunte.

Pour toutes valeurs de k et de π, on vérifie assez facilement les deux caractéristiques suivantes :

- La charge est bien de π (à condition que $k \leq \pi$). En effet, sur les arcs (s_i, s'_i), elle vaut π. Les arcs du chemin X sont tous de la forme $(x_{(\alpha_1\ldots\alpha_{k-1},i)}, x_{(\alpha_1\ldots\alpha_{k-1},i+1)})$ (modulo l'équivalence pour les indices dépassant π), et donc chacun n'est concerné que par les k chemins $\{\mu^1_{\alpha_1}, \mu^2_{\alpha_2}, \ldots, \mu^k_i\}$. Leur charge est donc k. Enfin les arcs de court-circuit ne sont empruntés que par des chemins qui ont parcouru l'arc précédent dans le chemin X, donc au maximum k.
- Tous les chemins de R sont en conflit deux-à-deux. Soient $(i, i', j, j') \in [1..\pi]^2 \times [1..k]^2$. Si $j' = j$ alors μ^j_i et $\mu^{j'}_{i'}$ sont tous deux dans R_j et donc sont en conflit le long de (s_j, s'_j). Sinon, supposons pour fixer les idées que $j < j'$: ils sont en conflit le long d'un chemin du type $x_{(\alpha_1\ldots\alpha_{j-1},i,\alpha_{j+1}\ldots\alpha_{j'-1},i',1\ldots1)} \quad \cdots \quad x_{(\alpha_1\ldots\alpha_{j-1},i,\alpha_{j+1}\ldots\alpha_{j'-1},i'+1,1\ldots1)}$.

On obtient donc ainsi, pour π quelconque, une instance de charge π mais ayant jusqu'à π^2 chemins tous en conflits. Le graphe des conflits associé est donc une clique de taille $|R|$ et ne peut être colorié en moins de $|R|$ couleurs.

2.2.3 Augmentation de w indépendamment de π

Après une augmentation linéaire puis quadratique, on montre ici comment créer une clique de taille arbitraire dans le graphe des conflits sans que la charge ne dépasse 2.

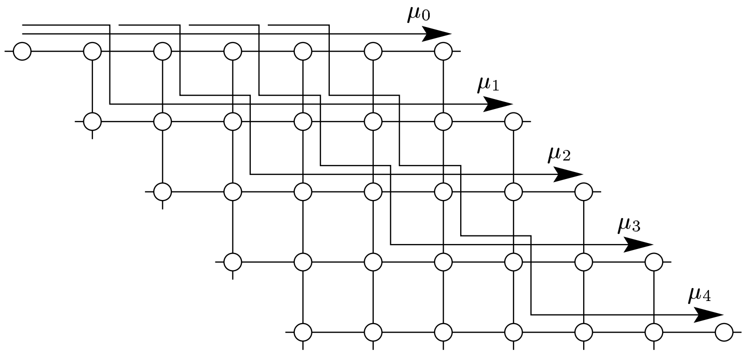

FIG. 2.3: Un routage de charge 2 et de nombre chromatique 5

Soit K la taille de la clique que l'on veut générer. On prend pour graphe physique $G = (V, E)$ une grille carrée torique de taille K, définie ainsi :

$$V = \{x_{i,j} | i \in [1..K], j \in [1..K] \text{ définis modulo } K\}$$
$$E = \{(x_{i,j}, x_{i+1,j}), (x_{i,j}, x_{i,j+1}) | i \in [1..K], j \in [1..K]\}$$

Le routage R est constitué des chemins $\{\mu_0, \mu_1 \ldots \mu_{K-1}\}$ définis comme suit :

$$\begin{cases} \mu_0 = (x_{1,1}, x_{1,2} \dots x_{1,K}) \\ \mu_i = (x_{1,i}, x_{1,i+1}, x_{2,i+1}, x_{2,i+2} \dots x_{i+1,2i}, x_{i+1,2i+1} \dots x_{i+1,i+K}) \forall i \in [1..K-1] \end{cases}$$

Sur la figure 2.3, tous les arcs du tore ne sont pas représentés mais on considère qu'il y a des arcs entre les bords respectivement droit/gauche et haut/bas du dessin.

Ce résultat est à priori plus puissant que ceux montrés précédemment. Par contre, il se base sur un routage inclus dans une grille, alors que les autres se servaient d'un arbre et d'un chemin et d'un cycle qui sont des topologies moins denses et surtout plus souvent présentes dans un graphe quelconque.

2.3 $\mathcal{NP}$-complétude

Comme c'est le cas pour la plupart des problèmes soulevés par le multiplexage en longueur d'onde, le placement des convertisseurs est un problème qui semble difficile à aborder. Il a donc paru naturel d'évaluer sa complexité, ce qui se fait assez simplement comme expliqué dans cette section.

2.3.1 Appartenance à $\mathcal{NP}$

Un certificat pour une solution consiste en un ensemble de sommets X où l'on fait de la conversion et une coloration des chemins de R_X en π couleurs.

La construction du graphe des conflits avec conversion H_X se fait en temps polynomial. Dans un premier temps on cherche R_X (ce qui par construction est au pire en temps $|R| * |X|$). Ensuite, on génère H_X. Les sommets sont directement crées à partir de R_X ; puis il suffit de parcourir tous les arcs de G, et le nombre d'arêtes construites pour chacun est de l'ordre du carré de sa charge. Le temps total (ainsi que la taille des données obtenues) est donc au pire de l'ordre de $|R_X|^2$.

Enfin, il faut vérifier la validité de la coloration proposée. Cette étape se fait en temps proportionnel au nombre d'arêtes de H_X puisqu'il suffit de vérifier qu'aucune d'entre elles ne relie deux sommets de même couleur.

2.3.2 Réduction polynomiale à partir d'un problème $\mathcal{NP}$-complet

Pour la réduction, on utilisera :

Problème 5 (Ensemble indépendant maximum (MAXISET))
Étant donnés un graphe $G = (V, E)$ et un entier k, déterminer s'il existe un ensemble S de sommets de G qui soit indépendant (c'est-à-dire tel qu'aucune arête de E ne relie entre eux deux sommets de S) et de taille au moins k.

Théorème 3 [GJ79] *Le problème de l'ensemble indépendant maximum est $\mathcal{NP}$-complet.*

Considérons une instance de MAXISET : un graphe $G = (V, E)$ et un entier k. L'instance correspondante du problème de placement optimal des convertisseurs à routage fixé $(G' = (V', E'), I, R, k')$ est construite de la façon suivante :

- Construction de G' :

$\forall v \in V$, $v \in V'$
$\forall e = (u, v) \in E$, e (orienté symétrique) $\in E'$

et on ajoute $\left\{ \begin{array}{l} u_e, u'_e, v_e, v'_e \text{ à } V' \\ (u_e, u), (u, u'_e), (v_e, v), (v, v'_e) \text{ à } E' \end{array} \right.$

- Construction de I et de R :

$\forall e = (u, v) \in E$, on ajoute dans I et R les cinq requêtes $\{(u_e, v), (u, v'_e), (v_e, v'_e), (v_e, u'_e), (u_e, u'_e)\}$ et le routage associé $\{(u_e uv), (uvv'_e), (v_e vv'_e), (v_e vuu'_e), (u_e uu'_e)\}$ comme décrit sur la figure 1.3

- On pose $k' = |V| - k$

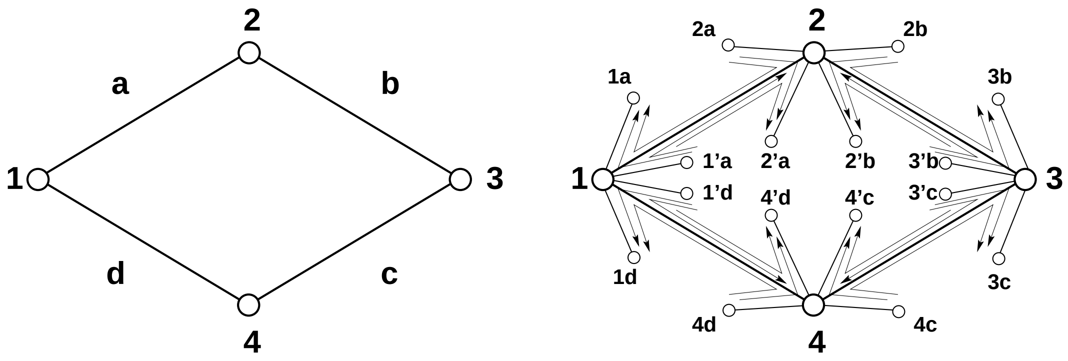

FIG. 2.4: Exemple : une instance de MAXISET et l'instance associée du problème des convertisseurs

La construction de cette instance se fait en temps linéaire du nombre d'arêtes du graphe G, par un parcours de ces dernières. En effet, G est recopié intégralement et on ajoute $4|E|$ sommets et arcs ; I et R sont de taille $5|E|$.

Constatons certaines caractéristiques de l'instance que nous venons de créer :

- Structure du graphe des conflits H : Chaque groupe de cinq chemins créé pour une arête de E se traduit par un cycle à cinq sommets (voir fig. 1.3). Ces cycles sont indépendants les uns des autres car aucun chemin ne passe par deux arcs de E, et les arcs de $E' \backslash E$ sont crées de telle sorte qu'aucun n'est partagé par deux distincts de ces groupes de cinq chemins.
- $\pi(G', I, R) = 2$
- $w(G', I, R) = 3$: Ces deux résultats s'obtiennent de la même manière : il suffit de s'intéresser à tous les sous-graphes $\{u, v, u_e, u'_e, v_e, v'_e\}$. Dans ceux-ci, ces valeurs se vérifient facilement : elles constituent donc des bornes inférieures des valeurs pour G tout entier. π n'augmente pas quand on considère le graphe dans son ensemble car aucun arc n'est partagé par deux sous-graphes ; w non plus car on ne crée aucun conflit supplémentaire.

Supposons maintenant qu'on a une réponse positive au problème de placement des convertisseurs $(G', I, R, |V| - k)$. On a donc un ensemble X de sommets de V' tel que $|X| \leq |V| - k$. A fortiori, on a donc $|X \cap V| \leq |V| - k$. Posons $S = V \backslash (X \cap V)$: c'est notre candidat pour la solution du problème MAXISET (G, k). On a bien $S \subset V$ et $|S| = |V| - |X \cap V| \geq |V| - (|V| - k) = k$.

Supposons donc que S ne soit pas stable dans G. Cela signifie qu'il existe $\{u, v\} \in S$ tels que $(u, v) \in E$. Or $u \in S \Rightarrow u \notin X$, donc on a une arête de E dont aucune extrémité n'est dans X. Or cela implique que le cycle correspondant à cette arête existe toujours dans le graphe des conflits, et donc que deux couleurs ne suffisent pas pour colorier H_X. On est alors en contradiction avec l'hypothèse selon laquelle X est une solution du problème $(G', I, R, |V| - k)$.

Réciproquement, une solution de MAXISET (G, k) donne une solution pour le placement des convertisseurs : $X = V \backslash S$ est bien de cardinal inférieur à $|V| - k$ et chaque arête de E dans G' est bien adjacente à l'un au moins de ses sommets sinon elle relie deux sommets de S. Tous les cycles de H sont donc éliminés dans H_X, ce qui nous donne la clause $w = \pi$.

On possède maintenant assez d'éléments pour élaborer une heuristique et caractériser les configurations pour lesquelles elle sera efficace ou non. Le chapitre suivant expose son fonctionnement et ses performances.

Chapitre 3

Une heuristique de placement et de coloration

Le problème considéré étant $\mathcal{NP}$-complet, il semble logique de n'en chercher qu'une solution approchée, calculable en un temps raisonnable. Pour cela, on se propose d'écrire un algorithme assez simple et dont les principes reposent sur quelques-uns des résultats précédemment cités, puis de tester son efficacité sur diverses instances bien choisies du problème.

3.1 Principes algorithmiques et choix d'implémentation

Comme on l'a déjà évoqué à propos de la complexité du problème, une méthode de résolution doit d'une part choisir des sites de conversion, d'autre part effectuer une coloration du graphe des conflits obtenu. Ces deux étapes sont susceptibles de demander la mise en œuvre d'un algorithme de complexité exponentielle. Nous allons voir par quels biais on espère revenir à un algorithme polynomial sans trop s'éloigner d'une solution correcte.

3.1.1 Fonctionnement global

Un premier choix important du paradigme retenu est le suivant : dans la configuration initiale, tous les nœuds sont capables de conversion. Le reste de la démarche consiste à "abandonner" des conversions, et donc à choisir des nœuds où on l'abandonne puis à recolorier le graphe des conflits ainsi obtenu en s'efforçant de ne pas augmenter son nombre chromatique.

Cette démarche trouve tout d'abord une justification dans les études de Subramaniam, Azizoğlu et Somani, qui ont montré que la conversion atteignait une efficacité maximale pour une densité de convertisseurs généralement assez élevée (voir section 1.3.2). Il est donc a priori plus économique de partir avec de la conversion partout puisque cette configuration a des chances d'être proche de la configuration optimale.

Elle trouve également des justifications au niveau algorithmique : le traitement du graphe des conflits est largement simplifié. D'une part, sa construction est rendue assez rapide puisqu'il est constitué de cliques disjointes dont les cardinaux sont les charges des différents arcs du graphe physique ; sa première coloration est par conséquent évidente. D'autre part, lorsqu'on abandonne la conversion en un site, le graphe des conflits modifié n'a pas à être entièrement recolorié : il suffit de régler les éventuels conflits crées autour du nœud d'abandon, ce qu'on peut faire localement dans la plupart des cas.

Le langage retenu pour l'implémentation est le `C`, notamment pour ses facilités d'utilisation des pointeurs. En effet, étant donné le nombre important de manipulations à effectuer sur la structure des graphes, il est plus souple de considérer que chaque sommet contient une référence sur chacun des arcs qui lui sont adjacents, et chaque arc une référence sur ses deux extrémités.

Par ailleurs, on a du faire certains choix pour permettre une traduction efficace de la conversion dans le graphe des conflits. Les chemins du routage sont donc décomposé en les arcs qui les forment, et ceux-ci sont affectés d'une étiquette repérant le chemin dont ils font partie. Dans le graphe des conflits, chaque sommet porte trois étiquettes : le numéro du chemin qu'il représente, le sommet de départ et celui d'arrivé dans le graphe physique. Ainsi deux sommets représentant les deux moitiés d'un chemin qui bénéficie de la conversion ont le même numéro, et le départ de l'un d'eux est l'arrivée de l'autre.

3.1.2 Placement des convertisseurs

Pour plus de souplesse, nous considérons que le choix des nœuds où l'on abandonne la conversion est effectué par une fonction indépendante du reste de l'algorithme, uniquement en fonction des réponses successives qu'elle reçoit à la question : "peut-on abandonner la conversion en un nœud donné sans augmenter le nombre chromatique ?" On envisagera donc plusieurs routines de choix et on comparera leurs performances respectives.

- Placement glouton : il a bien sûr l'avantage d'être linéaire en la taille du graphe physique. Son efficacité est par contre discutable : en effet, si le nombre minimal de convertisseurs nécessaire est de k, il peut exister des ensembles de convertisseurs de taille supérieure à k, qui permettent une coloration en π couleurs, mais qui ne contiennent pas forcément un placement optimal de taille k.

  Une manière d'améliorer la qualité de ce placement est de commencer par abandonner la conversion sur des nœuds formant un ensemble indépendant. Le graphe physique avec conversion est alors formé de subdivisions d'étoile, dont on sait qu'elles ne requièrent pas plus de π couleurs (voir théorème 1). La détermination d'un ensemble indépendant maximum est $\mathcal{NP}$-complète mais on peut se contenter d'un ensemble maximal (c'est-à-dire plus grand pour l'inclusion), qui se calcule trivialement par un algorithme glouton. Les résultats obtenus peuvent être encore un peu améliorés si on range les sommets par ordre de degré croissant.

- Placement par cardinal : le but de notre algorithme est de déterminer un ensemble de convertisseurs minimum. On peut donc opérer par diminution successive du nombre de convertisseurs : tant que le nombre chromatique n'augmente pas, on tente d'abandonner une conversion supplémentaire comme pour l'algorithme glouton. Si on échoue, on garde le même nombre de convertisseurs mais on essaye de les placer différemment, jusqu'à trouver une configuration satisfaisante ou bien avoir échoué sur toutes les configurations possibles avec ce nombre de convertisseurs. Cet algorithme va mieux fonctionner que l'algorithme glouton mais sa complexité au pire est exponentielle : il y a C_n^p façons de placer p convertisseurs sur n sites, et $C_n^p = O(2^n)$ pour des valeurs de p autour de $n/2$.

- Placement exhaustif : on essaye toutes les possibilités, c'est donc a priori le moins efficace en temps de calcul. Cependant, on peut l'améliorer jusqu'à le rendre meilleur que le placement par cardinal. La méthode à utiliser est celle du *Branch-and-bound* : elle consiste à mémoriser des résultats calculés pour certaines configurations et à les réinjecter dans les calculs sur d'autres configurations. On peut ainsi, pour des instances convenables, élaguer très largement l'arbre de recherche et espérer revenir à une complexité polynomiale. Par ailleurs, on constate que dans la plupart des réseaux réels, le nombre de nœuds est faible par rapport au volume des instances de communication, et donc que les efforts sur la complexité de l'algorithme doivent être faits plus au niveau de la coloration que du placement.

Pour le graphe des conflits H, un abandon de conversion dans le nœud u de G revient à chercher chacun des chemins de R qui traversent u puis à fusionner dans H les deux sommets qui représentent les deux moitiés de ce chemin.

Cela demande donc de concaténer les listes d'adjacence des deux sommets à fusionner en prenant soin d'éviter les doublons. Il apparaît d'ailleurs une difficulté si on se rend compte que l'abandon de la conversion n'est pas possible, car il faut revenir à la structure du graphe avant abandon, et donc recréer les arcs qui ont été supprimés et remettre en place ceux qui ont été déplacés. On garde donc en mémoire un historique des changements effectués, qu'on annule si besoin est ; sinon on vide l'historique afin de ne pas encombrer l'espace mémoire.

En pratique, l'implémentation n'est donc réalisable que pour les algorithmes gloutons : il faudrait prévoir une fonction de remise en place de la conversion, qui serait assez lourde en complexité. On verra cependant par la suite que les quantités de mémoire demandés par les algorithmes gloutons sont déjà plutôt prohibitifs.

3.1.3 Recoloration d'un graphe modifié

Quand on abandonne la conversion en u, on effectue en fait une recoloration locale de H pour chaque chemin μ qui traverse u. En effet, à μ sont associés deux sommets x et y dans H, que l'on va fusionner en un sommet z. Le graphe ainsi construit est noté ${}^{\mathrm{H}}/_{(\mathrm{x},\,\mathrm{y})}$. Le reste de H n'étant pas modifié, seul le nouveau sommet z n'est pas colorié correctement, et donc c'est à partir de lui qu'il va falloir parcourir ${}^{\mathrm{H}}/_{(\mathrm{x},\,\mathrm{y})}$ pour espérer garder un nombre chromatique égal à la charge.

De la même manière que pour le placement des sites de conversion, on va considérer plusieurs fonctions de recoloration distinctes.

- Coloration par échanges successifs : on s'appuie essentiellement sur le fait que le graphe H était colorié correctement avec π couleurs ; on appelle α et β les couleurs respectives de x et y dans H. On regarde donc d'abord s'il reste une couleur disponible une fois supprimées toutes celles employées par les sommets de z. Si c'est le cas, on l'utilise ; sinon, cela signifie notamment que $\alpha \neq \beta$ et que x et y avaient, dans H, un voisin de couleur respectivement β et α. On affecte donc une de ces deux couleurs (prenons α pour fixer les notations) à z et on parcourt le sous-graphe de ${}^{\mathrm{H}}/_{(\mathrm{x},\,\mathrm{y})}$ constitué des sommets de couleurs α et β, en partant des voisins de l'ancien sommet y.

  Cette opération s'effectue linéairement en le nombre d'arêtes de H. Pour chaque sommet rencontré, on cherche s'il existe une couleur non utilisée par ses voisins, et si c'est le cas on l'utilise et on ne continue pas le parcours à partir de ce sommet ; sinon on échange les couleurs α et β afin de régler les conflits avec les sommets déjà parcourus puis on continue le parcours (figure 3.1 à gauche). Si ce processus de recoloration n'aboutit pas, c'est que le parcours retombe sur un sommet déjà visité mais doit le recolorier. Comme le parcours ne se poursuit que dans le cas où on ne fait que des échanges, cela signifie qu'on rentre dans un cycle de

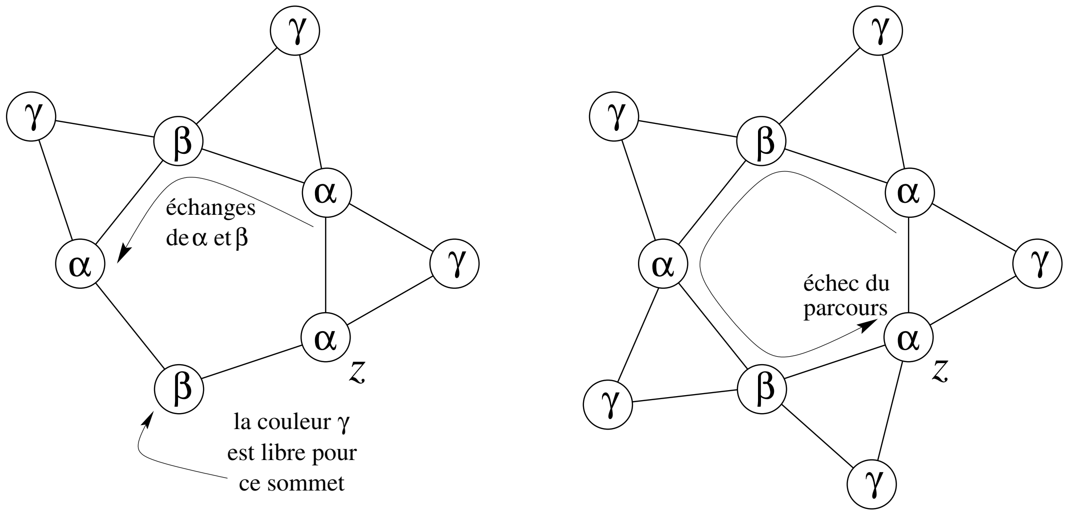

FIG. 3.1: Deux cas de recoloration en $\pi = 3$ couleurs : z est le sommet de départ ; les couleurs des sommets sont celles au moment de la fusion ; les flèches indiquent le travail de l'algorithme.

longueur impaire coloré uniquement avec α et β et dont chaque sommet a un voisin au moins dans chacune des $\pi - 2$ autres couleurs. De plus, comme H était colorié correctement, on sait que ce cycle a été créé par la fusion de x et y.

Cette procédure de recoloration détecte donc facilement les cas où le nombre chromatique augmente pour des raisons similaires à celle de l'exemple 1.2. Elle peut cependant échouer sur des graphes où le nombre chromatique n'augmente pas. Ainsi, sur la figure 3.1 à droite, une coloration acceptable en 3 couleurs est trouvable en coloriant d'abord le cycle de longueur 5 avec 3 couleurs puis en coloriant chacun des sommets extérieurs avec la couleur non utilisée par ses deux voisins.

– COLORATION LOCALE EXHAUSTIVE : dans le processus décrit ci-dessus, on s'autorisait un parcours de la totalité du graphe (ou presque), mais on ne demandait à l'algorithme aucun choix quant à la couleur à utiliser. À l'inverse, on teste ici toutes les combinaisons de couleurs possibles à condition qu'elles respectent les règles de coloration, mais on ne le fait que dans un voisinage du sommet z, c'est-à-dire pour les sommets de H qui se trouvent à une distance bornée d de z. Une coloration de ce voisinage est acceptée si elle est compatible avec le reste du graphe H. Le temps mis est donc de l'ordre de $\pi^{d\Delta}$, où Δ est le degré maximum des sommets de H, il est donc raisonnable pour les graphes de degré borné, ce qui est souvent le cas surtout dans les réseaux réels. Contrairement au précédent, ce processus ne détectera pas rapidement les cycles impairs crées par la fusion de x et y. Par contre, si une clique de taille supérieure à π est créée, ce sera autour de z, et donc elle sera détectée même si on ne s'autorise la recoloration qu'à distance 1.

Aucune de ces deux recolorations ne pose de problème particulier d'implémentation. La première demande un parcours efficace du graphe, ce qui se fait immédiatement grâce à l'utilisation systématique de pointeurs dans notre structure de données. La seconde n'est pas beaucoup plus difficile, il suffit de créer une fonction récursive dont un des paramètres est un entier représentant le nombre d'appels récursifs autorisés, de l'initialiser à d et de le faire décroître à chaque appel. On évitera d'utiliser cette procédure sur des graphes dont la charge devient trop grande.

3.2 Performances

L'heuristique donnée ici n'offre aucune garantie quant à un placement optimal des convertisseurs. Pour se donner une idée de sa valeur, on décide donc de la faire fonctionner sur une série de tests, puis de mesurer et d'interpréter les performances obtenues.

3.2.1 Création de batteries de tests adaptées

L'entrée du problème est constituée d'une part d'un graphe physique, d'autre part d'une instance de communication et de son routage. Pour effectuer des mesures, il a donc fallu générer des séries d'exemplaires de ces deux entrées.

Une partie des graphes testés sont des topologies régulières classiques : chemins, étoiles, cycles et grilles. Un intérêt est qu'on connaît déjà beaucoup de résultats théoriques sur ces graphes et donc que l'efficacité peut être évaluée plus précisément. Par ailleurs, les travaux précédents ont aussi fait beaucoup de leurs mesures de performances sur ces graphes, et ils constituent par conséquent de bonnes instances pour comparer les algorithmes entre eux.

Cependant, les réseaux réels ne correspondent que rarement à des topologies aussi particulières. Plusieurs exemples de réseaux européens et mondiaux., fournis par France Telecom, ont été étudiés et il apparaît qu'ils sont presque des graphes planaires, à quelques arêtes près. Pour les modéliser, on s'est donc servis d'un algorithme de génération de graphes planaires dont les sommets sont de degré élevé, trouvé dans les séries de tests du deuxième challenge DIMACS [JT93]. Le programme utilisé demande en entrée un entier qui représente la taille d'un cycle sur lequel est construit le graphe. Le graphe ainsi généré a une taille environ triple de celle du cycle, ce qui donne rapidement des graphes assez grands ; ceci pose d'ailleurs parfois des problèmes d'espace mémoire insuffisant.

Une fois le graphe physique fixé, il faut créer une instance de communication et un routage acceptable. Le problème du routage est étudié indépendamment par plusieurs équipes de recherche, y compris au sein du projet Mascotte. Les algorithmes proposés sont cependant assez lourds à mettre en œuvre car ils utilisent un solveur d'équations. Pour nos tests, on a retenu la solution assez rapide du routage des plus courts chemins, en affectant un certain poids à chaque arête du graphe physique. Son implémentation en $O(n^3)$ est très simple, mais il en existe des versions plus performantes [Sei92].

D'autre part, on affecte des volumes à ces routages correspondant aux différents volumes de l'instance de communication associée. Un premier modèle consistait à prendre des entiers totalement aléatoires, simplement bornés par un volume maximal. On se rend alors vite compte d'une part que la répartition des communications ne correspond pas bien aux réseaux réels, d'autre part que le volume total des communications est beaucoup trop important pour être traité (sur un graphe à soixante sommets, la charge atteignait huit cents, et le graphe des conflits comportait plus de dix millions d'arêtes).

Le modèle finalement retenu est celui qu'on appelle généralement modèle gravitationnel : chaque sommet reçoit un poids, et les communications entre deux sommets sont proportionnelles au produit des poids de ces deux sommets. On reproduit ainsi bien le fait que dans les réseaux réels se trouvent des nœuds de routage plus importants que les autres, qui communiquent beaucoup avec leurs voisins directs et entre eux, alors que le reste des nœuds ne communiquent que très peu entre eux. On peut enfin affiner ce modèle en diminuant le volume des communications entre deux sommets s'ils sont très éloignés l'un de l'autre.

3.2.2 Mesures effectuées

Pour les graphes modélisant des réseaux réels, étant limités par les capacités de mémoire nécéssaire, on a considéré trois ordres de grandeur de la taille du graphe :

- 15 sommets, 35 arêtes. Les sommets sont tous de degré 5 ou 6.
- 30 sommets, 85 arêtes. Les degrés 5 et 6 restent majoritaires mais quelques sommets momtent jusqu'à 10.
- 60 sommets, 190 arêtes. Le degré 7 commence à être plus utilisé mais le degré max reste de l'ordre de 10.

On a également considéré trois bornes distinctes sur les volumes de communications entre deux sommets : 10, 20 et 40. La combinaison d'un graphe d'environ 60 sommets et d'un volume de communication allant jusqu'à 40 entre deux sommets a d'ailleurs dépassé les capacités des machines que nous avions à disposition : le graphe des conflits obtenu compte en effet plus de dix millions d'arêtes. Or les tests ont été effectués sur un Pentium III (600 MHz) doté de 256 Mo de mémoire vive, et notre structure de données requiert environ 50 octets de mémoire par arête. On pourait chercher une implémentation moins gourmande, mais l'ordre de grandeur des graphes manipulables ne changerait pas beaucoup, et leur traitement en serait significativement compliqué. On a par contre pu tester un volume de communications encore plus important, allant jusqu'à un facteur de 100, sur les graphes d'une quinzaine de sommets.

Pour tous les calculs qui ont abouti, l'algorithme est parvenu à supprimer la conversion dans tous les nœuds sans pour autant augmenter le nombre chromatique. La solution trouvée est donc forcément optimale : pour ces graphes, il n'y avait pas besoin de conversion. On peut en tirer deux conclusions :

- Si notre modélisation gravitationnelle des réseaux réels est correcte, on peut supposer que la conversion n'apportera pas de bénéfices significatifs dans ceux-ci, même pour des ordres de grandeur plus importants. Des tests en quantité plus importante seraient donc susceptibles de montrer que la conversion ne doit être envisagée que si le réseau de fibres optiques possède une topologie un peu particulière, par exemple se rapprochant de celles que nous avons mises en évidence dans la section 2.2.
- Pour des instances de communications relativement régulières (contrairement aux cas pathologiques envisagés), notre heuristique de recoloration est efficace. On pourrait donc envisager de l'utiliser même hors du domaine de la conversion, car elle donne une approche polynomiale du problème de la planification si on décide plutôt de laisser le nombre chromatique augmenter mais de supprimer la conversion dans tous les nœuds.

Parmi les graphes particuliers, on a naturellement commencé par les graphes réputés pour ne nécessiter aucune conversion : les chemins et les étoiles. Les volumes de communications ont été générés par le même modèle que pour les réseaux "réels", en plus de quelques cas particuliers très réguliers.

Que ce soit sur un chemin court (une dizaine de sommets) ou long (quatre-vingt), on aboutit bien à un réseau sans conversion. Les volumes manipulables vont jusqu'à 8000 requêtes, soit une charge maximum de 700 pour le réseau et un graphe des conflits comportant 15 millions d'arêtes.

Pour les étoiles, on a retenu des tailles similaires : une dizaine puis une centaine de sommets. On parvient également à supprimer toute conversion, avec bien sûr un gros temps de calcul autour du nœud central.

On a également testé des réseaux cycliques. Dans ceux-ci, il suffit d'un convertisseur, car le graphe obtenu est alors un cycle. Notre algorithme parvient à ce résultat pour toutes les tailles de cycles manipulables. Il peut être intéressant de constater que le sommet sur lequel il refuse d'abandonner la conversion est systématiquement le dernier qu'il traite. Cela nous encourage à penser que l'ordre dans lequel on lui demande de les traiter n'influe pas sur ses performances, ce qui permet de gagner beaucoup de temps de calcul car il n'est pas nécessaire d'envisager toutes les combinaisons ordonnées de sommets mais seulement des k-uplets.

Ensuite, on a proposé à notre programme des cas pathologiques du type de ceux montrés dans la section 2.2. Cependant, les instances testées sont de taille limitée pour le graphe de la figure 2.2 ($\pi = 4$ et $k = 3$ au maximum) car le nombre de sommets nécessaire augmente quadratiquement et il n'est pas facile de générer automatiquement de telles instances. De même, on a testé une taille de grille de 10 au maximum pour celui de la figure 2.3 ; il est plus facile à générer, mais on voit de toutes façons assez vite comment l'algorithme se comporte.

Les tests effectués sur le graphe 1.2 avec chemins multiples, dans lequel w augmente par création de cycles sont concluants : pour tout volume de communication (d'une taille manipulable en mémoire, c'est-à-dire de l'ordre du millier de requêtes), la conversion n'est nécessaire que dans l'un des deux sommets de degré trois. On vérifie facilement que cette solution est optimale : la conversion dans un seul des deux sommets de degré trois du graphe physique (sommets 1 et 4 sur la figure 1.2) suffit à ne plus avoir, dans le graphe des conflits, que des cliques isolées de taille π ou $\pi/2$. Et bien entendu, sans conversion, des cycles impairs apparaissent entre des paires de sommets de ces cliques, donc il faut des couleurs supplémentaires.

Le graphe de la figure 2.2 se base sur un graphe qui est un chemin X de longueur π^k. On constate qu'il suffit généralement de peu de convertisseurs : notre algorithme donne des résultats de l'ordre de π. Cela s'explique facilement : la conversion dans les nœuds de "court-circuit" les plus utilisés est très intéressante car beaucoup de chemins à la fois peuvent y être coupés en deux ; or en utilisant les courts-circuits les plus simples, $\pi - 1$ d'entre eux sont traversés par les chemins les plus longs de R. Si on augmente le nombre de ces sommets de transit, la conversion a alors bien sûr tendance à se déporter sur le chemin X. Il suffit en général de faire beaucoup de conversion sur une unique portion de longueur π du chemin. En effet, on modifie ainsi un chemin de chaque groupe R_i, ce qui simplifie considérablement le coloriage. Les tests effectués ne suffisent cependant pas pour déterminer comment l'algorithme en arrive à faire ces choix, ni s'ils sont proches de l'optimalité. Il aurait fallu plus de graphes de tests, ce qui aurait demand d'crire un autre programme de gnration de graphes.

Pour la grille de la figure 2.3, on constate que le nombre de convertisseurs utilisés est de l'ordre du nombre de chemins de R. En effet, l'algorithme choisit d'effectuer une conversion sur chacun de ceux-ci (sauf pour $O(1)$ chemins), soit sur la première ligne de la grille, soit au début de la ligne que le chemin emprunte en dernier. Le résultat est donc ici peu satisfaisant par rapport au volume de l'instance de communication (puisqu'il faut environ autant de convertisseurs que de requêtes), un peu plus par rapport à la taille du graphe physique (celle-ci est de l'ordre du carré du nombre de requêtes).

Cependant, le nombre optimal de convertisseurs est d'au moins $|R|/9$. Montrons rapidement cette borne. Pour tout triplet de chemins de R, ces trois chemins sont en conflit mutuel : il faut donc qu'au moins un d'entre eux soit sujet à une conversion, sinon le nombre chromatique dépasse

$\pi(=2)$. Il faut donc convertir au moins $|R|/3$ chemins. Par ailleurs, jamais plus de trois chemins distincts ne passent dans un nœud donné de ce graphe (par construction). Pour convertir n chemins, il faut donc au moins $n/3$ sites de conversion. On a donc le résultat annoncé : il faut au minimum $|R|/9$ convertisseurs dans un tel graphe. Les performances de notre algorithme sur ces graphes sont donc du même ordre que l'optimal.

Conclusion et perspectives

Au cours de ce stage, nous avons étudié le multiplexage en longueur d'onde (WDM) et plus particulièrement le placement des convertisseurs.

Nous avons commencé par un travail de bibliographie assez important, afin de bien assimiler les diverses problématiques, les techniques de résolution et les résultats connus. Dans un premier temps, des connaissances avancées de la théorie des graphes ont donc été acquises. Ensuite, ceci nous a permis de dégager parmi les connaissances actuelles celles qui pouvaient conduire à une meilleure résolution du problème des convertisseurs. Nous avons enfin pu reprendre une modélisation mathématique largement répandue du problème et la développer pour qu'elle intègre l'existence des convertisseurs dans un réseau.

Dans une seconde partie, nous avons démontré quelques résultats théoriques sur le placement des convertisseurs.

Pour mettre en évidence les configurations des instances qui induisent un nombre chromatique supérieur à la charge π, on décrit certains cas pathologiques. On trouve ainsi pour k quelconque un routage simple dans un arbre pour lequel la charge est de $2k$ le nombre chromatique est de $\lceil 5k/2 \rceil$, et on cite un article décrivant un routage de charge k et de nombre chromatique $2k-1$ dans un cycle. La première instance nous donne une configuration du graphe des conflits qui fait augmenter le nombre chromatique : la présence de cycles de longueur impaire bien placés demande une couleur supplémentaire pour une partie d'entre eux.

On montre ensuite comment créer un graphe des conflits qui est une clique de taille supérieure à la charge, et qui donc demande autant de couleurs que sa taille. On utilise pour cela un chemin avec des "courts-circuits" qui donne une clique de taille π^2, puis une grille torique qui donne une clique de même taille que la grille avec une charge limitée à 2. On estime que les cycles et les cliques sont donc les principaux responsables de l'augmentation du nombre chromatique.

Enfin, une étude de complexité a débouché sur la preuve de la $\mathcal{NP}$-complétude du problème, par réduction à partir du problème de l'ensemble indépendant maximal.

Une troisième étape a consisté à construire une heuristique de placement des convertisseurs et de coloration qui s'exécute polynomialement en la taille des données, et qui s'appuie à la fois sur les résultats du domaine de la conversion trouvés dans la bibliographie et sur les cas étudiés à la suite de la $\mathcal{NP}$-complétude. On part de la configuration où la conversion se fait partout, qui est celle pour laquelle le graphe des conflits est le plus simple. On place ensuite en fait des sites sans conversion et on recolorie les chemins à la volée.

On a proposé plusieurs méthodes de choix des sites basées notamment sur les résultats existants, puis des méthodes de recoloration basées elles plus sur nos cas pathologiques de la seconde partie.et on a implémenté les plus efficaces en termes de temps de calcul. On a également implémenté des algorithmes de génération de graphes de test, sur lesquels on a évalué l'efficacité de notre algorithme. Une partie de ces graphes de test est construite pour ressembler aux réseaux optiques existants, suivant un modèle gravitationnel.

On constate dans un premier temps que les performances en temps d'exécution de notre programme sont assez bonnes ; le problème qui apparaît rapidement est la taille de la mémoire. En effet, les graphes des conflits manipulés comportent plusieurs centaines de millers d'arêtes, voire plusieurs millions, ce qui est l'ordre de grandeur de la mémoire vive exprimé en octets. Le programme est donc efficace jusqu'à saturation de la mémoire, ensuite il est obligé d'utiliser le swap, ce qui le ralentit considérablement à cause des accès répétés au disque dur.

Quant aux performances de coloration de l'algorithme, elles semblent satisfaisantes. Sur le modèle gravitationnel, on arrive de proche en proche à une coloration optimale de tout le graphe, sans aucune conversion. On obtient également ce résultat sur les graphes comme le chemin ou l'étoile, dont on sait qu'ils ne nécessitent pas de conversion. Pour un graphe qui en demande un nombre limité, comme le cycle, on parvient aux bornes inférieures donnés par la théorie, ce qui garantit l'optimalité. Enfin, sur les cas pathologiques de la section 2.2, on constate qu'il suffit en général d'un nombre de conversions de l'ordre de $|R|$, localisées sur certaines portions très précises du graphe physique. Si on n'atteint que rarement le nombre optimal, on trouve en général le même ordre de grandeur que certaines bornes inférieures sur ce dernier.

Les résultats théoriques montrés n'ouvrent pas de perspectives de recherche d'une solution exacte puisque notre problème est $\mathcal{NP}$-complet. On pourrait cependant s'intéresser à son approximabilité. Pour le problème très proche du routage fixé, l'étude de Togni [Tog00] donne quelques résultats en ce domaine, mais pour des graphes généraux la recherche d'une solution approchée à $5/3$ reste $\mathcal{NP}$-complète. On peut donc penser que, tout comme la coloration de graphe, notre problème est non-approximable, ce qui resterait à démontrer proprement. Une idée moins ambitieuse serait de mettre en évidence des cas pathologiques dans d'autres graphes bien connus que ceux envisagés dans la section 2.2.

Les performances prometteuses de notre heuristique nous incitent à penser qu'elle peut être reprise et améliorée, notamment en ce qui concerne les procédures de choix des sites de conversion et de recoloriage local. On pourrait par exemple utiliser des algorithmes efficaces de recherche d'ensembles indépendants pour trier les sommets dans un ordre approprié avant de commencer les abandons de conversion. Il semblerait également intéressant d'utiliser des algorithmes génétiques pour la recoloration, car ceux-ci sont bien adaptés au traitement dynamique de données. On pourrait ainsi fournir le graphe des conflits initial à l'algorithme génétique, puis lui communiquer au fur et à mesure les modifications qu'on souhaite y apporter, et enfin faire des mesures régulièrement pour contrôler que le nombre chromatique n'augmente pas.

Bibliographie

[Bea99] Beauquier (B.), All-to-all communication for some wavelength-routed all-optical networks , *Networks*, vol. 33, n^{o} 3, Mai 1999, p. 179–187.

[Bea00] Beauquier (B.), *Communications dans les réseaux optiques par multiplexage en longueur d'onde.* Thèse de doctorat, Université de Nice-Sophia Antipolis, 2000.

[Ber83] Berge (C.), *Graphes.* Gauthier-Villars, Paris, 1983.

[BHP98] Beauquier (B.), Hell (P.) et Pérennes (S.), Optimal wavelength-routed multicasting , *Discrete Applied Mathematics*, vol. 84, 1998, p. 15–20.

[EJ96] Erlebach (T.) et Jansen (K.), Scheduling of virtual connections in fast networks , dans *Proc. of 4th Workshop on Parallel Systems and Algorithms (PASA '96)*, p. 13–32. World Scientific, 1996.

[EJK^{+}99] Erlebach (T.), Jansen (K.), Kaklamanis (C.), Mihail (M.) et Persiano (P.), Optimal wavelength routing on directed fiber trees , *Theoretical Computer Science*, vol. 221, n^{o} 1-2, 1999, p. 119–137.

[GJ79] Garey (M. R.) et Johnson (D. S.), *Computer and Intractability : A Guide to the Theory of NP-Completeness.* New York, W. H. Freeman and Company, 1979.

[GJ85] Golumbic (M. C.) et Jamison (R. E.), The edge intersection graphs of paths in a tree , *Journal of Comb, Theory, Series B*, vol. 38, 1985, p. 8–22.

[GLS98] Gerstel (O.), Lin (P.) et Sasaki (G.), Wavelength assignement in a WDM ring to minimize system cost instead of number of wavelengths , dans *Proc. of IEEE INFOCOM*, vol. 1, p. 94–101, 1998.

[JT93] Johnson (D. S.) et Trick (M. A.), diteurs, *Cliques, Coloring and Satisfiability*, vol. 26 (coll. *Discrete Mathematics and Theoretical Computer Science*). American Mathematical Society, Octobre 1993.

[LLWF00] Liu (L.), Li (X.), Wan (P.-J.) et Frieder (O.), Wavelength assignment in WDM rings to minimize SONET ADMs , dans *Proc. of IEEE INFOCOM*, Tel Aviv, Israel, mars 2000.

[NST^{+}99] Naito (T.), Shimojoh (N.), Tanaka (T.), Nakamoto (H.), Doi (M.), Ueki (T.) et Suyama (M.), 1 Terabit/s WDM transmission over 10,000 km , dans *Proc. of the 25th European Conference on Optical Communication*, p. 24–25, Nice, Septembre 1999.

[RS98] Ramaswami (R.) et Sasaki (G.), Multiwavelength optical networks with limited wavelength conversion , dans *Proc. of IEEE INFOCOM*, vol. 6, p. 744–754, 1998.

[SAS96] Subramaniam (S.), Azizoğlu (M.) et Somani (A. K.), All-optical networks with sparse wavelength conversion , *IEEE/ACM Transactions on Networking*, vol. 4, n^{o} 4, Août 1996, p. 544–557.

[SAS98] Subramaniam (S.), Azizoğlu (M.) et Somani (A. K.), On the optimal placement of wavelength converters in wavelength-routed networks , dans *Proc. of IEEE INFOCOM*, p. 902–909, Avril 1998.

[Sei92] Seidel (R.), On the all-pairs-shortest-path problem , dans *Proc. of the Twenty-Fourth Annual ACM Symposium on the Theory of Computing*, p. 745–749, Victoria, British Columbia, Canada, Mai 1992. ACM.

[Tog00] Togni (O.), Placement de convertisseurs de longueur d'onde dans les réseaux optiques , dans *Proc. of Algotel'2000*, p. 35–40, 2000.

[TS99] Thiagarajan (S.) et Somani (A. K.), An efficient algorithm for optimal wavelength converter placement on wavelength-routed networks with arbitrary topologies , dans *Proc. of IEEE INFOCOM*, Mars 1999.

[YZP+00] Yu (J.), Zheng (X.), Peucheret (C.), Clausen (A. T.), Poulsen (H. N.) et Jeppesen (P.), 40-Gbit/s all-optical wavelength conversion based on a nonlinear optical loop mirror , *Journal of Lightwave Technology*, vol. 18, n° 7, Juillet 2000, p. 1001–1006.

Annexe : le code `C` de l'heuristique

Annexe : le code `C` de création des graphes de test