



Titre: Routage et affectation de longueurs d'onde dans les réseaux optiques WDM
Title:

Auteur: Khyda D. ésiré Yannick Oulaï
Author:

Date: 2002

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Oulaï, K. D. Y. (2002). Routage et affectation de longueurs d'onde dans les réseaux optiques WDM [Mémoire de maîtrise, École Polytechnique de Montréal].
Citation: PolyPublie. <https://publications.polymtl.ca/7146/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7146/>
PolyPublie URL:

Directeurs de recherche: Samuel Pierre, & Steven Chamberland
Advisors:

Programme: Génie électrique
Program:

In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

UNIVERSITÉ DE MONTRÉAL

ROUTAGE ET AFFECTATION DE LONGUEURS
D'ONDE DANS LES RÉSEAUX OPTIQUES WDM

KHYDA DÉSIÉ YANNICK OULAI
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

DÉCEMBRE 2002

© Khyda Désiré Yannick Oulai, 2002



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-86422-7

Our file Notre référence

ISBN: 0-612-86422-7

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

AFFECTATION DE LONGUEURS
D'ONDE DANS LES RÉSEAUX WDM

présenté par : OULAÏ Khyda Désiré Yannick

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. PESANT Gilles, Ph.D., président

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. CHAMBERLAND Steven, Ph.D., membre et codirecteur de recherche

M. GALINIER Philippe, Ph.D., membre

REMERCIEMENTS

Je tiens tout d'abord à élever le nom du Seigneur Jésus-Christ, soutien fidèle sans qui je ne serais pas là.

J'aimerais aussi remercier MM Samuel Pierre et Steven Chamberland, mes directeur et codirecteur de recherche dont l'encadrement et les conseils m'ont guidé tout au long de ma recherche.

Ma reconnaissance va aussi naturellement à mon père et à ma mère dont l'éducation, la confiance, l'appui moral et financier m'ont permis de m'épanouir. À mes frères Jean Marie, Stéphane et Grégory, mon oncle Marcellin, à toute ma famille, je dis merci pour ces moments où vous m'avez fait ressentir la chaleur fraternelle.

Je ne saurais continuer ces remerciements sans mentionner celle qui a fait énormément pour moi pendant cette recherche, celle qui a supporté mes humeurs changeantes en fonction de l'évolution du projet, ma Lau, compagne dévouée et attentionnée. Merci pour tout.

À mes amis, Stéphane, Yao, Alia, Zeinab, Guibs, Brice, Jean Phillippe, Martial, Isabelle, je vous suis reconnaissant pour votre appui et vos relectures. À Younès, Ali, Fabien, Rolland et Christiane et à tous mes collègues du LARIM, je dis merci pour l'ambiance chaleureuse et enrichissante du laboratoire.

À tous ceux qui m'ont gardé dans leurs prières, Pasteur Eugène, Mathurin, Valentine, Karen, Diane, Welsia, JB et bien d'autres, je vous remercie de ce temps précieux investi pour moi.

À mon pays d'origine, la Côte d'Ivoire, qui subit une des pires crises de son histoire, je souhaite un prompt retour de la paix.

RÉSUMÉ

Durant le dernier quart de siècle, le monde de l'information a connu des développements formidables. Cela a entraîné un accroissement de la demande en bande passante des réseaux. Pour faire face à cela, les réseaux optiques avec plusieurs longueurs d'ondes apparaissent comme une alternative intéressante non seulement à cause de leur importante largeur de bande, mais aussi à cause de la faible atténuation du signal. Ces réseaux se retrouvent essentiellement au niveau dorsal. La planification des réseaux optiques représente un processus long et complexe dont les étapes principales sont la caractérisation du trafic, la conception topologique, le routage et l'affectation de longueurs d'ondes, ainsi que le dimensionnement des liens. En plus de cela, il faut tenir compte des critères de fiabilité du réseau. Dans ce vaste éventail de problèmes complexes, ce mémoire se situe au niveau du routage et de l'affectation de longueurs d'ondes pour une topologie, un dimensionnement et une demande statique donnés. Le routage consiste à trouver un chemin satisfaisant certaines conditions et l'affectation de longueurs d'ondes vise à déterminer une longueur d'onde disponible pour l'établissement de la connexion. Nous avons considéré la continuité de longueur d'onde sur tout le réseau pour une connexion donnée. En plus de cela, le routage choisi se base sur une politique de plus court chemin et l'affectation de longueurs d'ondes est modélisée par un problème de coloriage dans un graphe.

L'objectif de ce mémoire est de proposer une heuristique permettant de trouver une bonne solution au problème de routage et d'affectation de longueurs d'ondes qui a été répertorié comme NP-difficile. Pour y parvenir, nous avons d'abord proposé une formulation mathématique du problème représentant la fonction objectif à minimiser, les contraintes à satisfaire et les variables dont dépendent la fonction et les contraintes. Ensuite, nous avons présenté un algorithme simple de résolution basé sur l'algorithme de Dijkstra pour trouver les chemins et sur une heuristique pour l'affectation de longueurs d'ondes. Comme contribution principale, nous avons ensuite proposé une

ABSTRACT

During the last quarter of the century, tremendous developments occurred in the world of information, increasing the demand for broadband applications and networks.

In order to face this expansion, multi-wavelength optical networks appear as an interesting alternative thanks not only to their significant bandwidth, but also to the low attenuation of their signal. These networks are found primarily at the backbone level. The planning of the multi-wavelength optical networks represents a long and complex process. In fact, its main stages are the characterization of the traffic, the topological design, the routing and wavelength assignments and the link dimensioning. Furthermore, it is necessary to take into account the network's criteria of reliability. In this wide environment of complex problems, we tackle the routing and the wavelength assignment problem for a given topology, link dimensioning and static request of connections.

The routing consists of finding a path satisfying certain conditions, and the wavelength assignment aims at determining a wavelength available for the establishment of the connection. The wavelength continuity was monitored over the entire network for a given connection. In addition, the routing was established upon the shortest path policy, while the wavelengths assignment was carried out by transforming it into an equivalent graph coloring problem.

The objective is to propose heuristics to find a good solution for the routing and wavelength assignment problem that was indexed as NP-hard. For that purpose, we initially proposed a mathematical formulation of the problem.

Then, we presented a simple algorithm. Its resolution was based on the algorithm of Dijkstra for the routing, while the wavelength assignment part of the problem was carried out by transforming the networks into an equivalent graph coloring problem. Our main contribution was to propose a meta-heuristic allowing to improve the solutions of the basic algorithm. The latter can contribute to improve the results of

various wavelength assignment algorithms.

Finally, failing to know the optimum of our solutions, we tried to estimate lower bounds by using the linear optimizer CPLEX. That enabled us to assess the performance of our solution approach.

The two main criteria we estimated are the execution time and the gap between our results and the optimal solution or, if not possible, between our results and the lower bounds of the optimal solution. The routing and wavelength assignment problem is known as NP-hard. That is why the execution times of our algorithms are much lower than those obtained with CPLEX. For the solutions, we were able to obtain some lower bounds. For a small number of available wavelengths on the network, the gap between our solution and the lower bound is small. For the same network and the same demand, the gap increases up to a threshold with the increase of the number of available wavelengths. Once this threshold is reached, the gap starts to decrease with the increasing of the number of available wavelengths. This gap is null when there are enough wavelengths to support the demand. Despite the variation of the gap, our results are interesting.

heuristique permettant d'améliorer les solutions de l'algorithme de base. Cette dernière peut contribuer à améliorer les résultats de différents algorithmes. Enfin, à défaut de connaître l'optimum de nos solutions, nous avons tenté d'en estimer les bornes en utilisant le logiciel d'optimisation CPLEX. Cela nous a permis de statuer sur la qualité de nos solutions.

Les deux critères principaux de performance que nous avons évalués sont le temps et l'écart de notre solution vis-à-vis de la solution optimale ou, à défaut, vis-à-vis d'une borne inférieure de cette solution optimale. Le problème de routage et d'affectation de longueurs d'ondes étant NP-difficile, il fallait s'attendre à ce que les temps d'exécution fournis par nos algorithmes surclassent ceux obtenus à partir du logiciel CPLEX. Au niveau des solutions obtenues, nous avons pu estimer quelques bornes inférieures et calculer les écarts. On note que, pour un faible nombre de longueurs d'ondes disponible sur le réseau, l'écart entre notre solution et la borne est faible. Pour un même réseau et une même charge, l'écart augmente jusqu'à un certain seuil en fonction de l'augmentation du nombre de longueurs d'ondes. Une fois ce seuil atteint, l'écart se met à décroître avec l'augmentation du nombre de longueurs d'ondes pour devenir nul dans le cas où le réseau recèle assez de ressources pour supporter les demandes. Malgré la variation de la distance entre la borne et la solution obtenue, on peut dire que les résultats obtenus sont intéressants.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES FIGURES	xii
LISTE DES TABLEAUX	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
CHAPITRE I INTRODUCTION	1
1.1 Définitions et concepts de base	1
1.2 Éléments de la problématique	5
1.3 Objectifs de recherche	7
1.4 Esquisse méthodologique	8
1.5 Plan du mémoire	9
CHAPITRE II ROUTAGE ET AFFECTATION DE LONGUEURS D'ONDE	10
2.1 La technologie WDM	11
2.1.1 Description	11
2.1.2 Quelques équipements optiques	12
2.2 Aspects fondamentaux du problème	13
2.2.1 Angle d'approche du problème	13
2.2.2 Contrôle du réseau	14
2.2.3 Survivabilité des réseaux optiques	15
2.3 Formulation du problème	17
2.4 Algorithmes classiques de résolution	19
2.4.1 Coloriage	19
2.4.2 Méthodes de protection	20
2.4.3 Algorithme génétique et recuit simulé	21

2.4.4	Algorithme de perte de capacité relative	25
2.4.5	Algorithme de chemins disjoints d'arcs.....	26
2.4.6	Approche de conversion de graphe.....	27
2.4.7	Algorithme basé sur le délai	29
2.4.8	Autres algorithmes simples.....	30
CHAPITRE III ALGORITHMES DE ROUTAGE ET D'AFFECTATION DE LONGUEURS D'ONDE PROPOSÉS		33
3.1	Motivations et formulation mathématique du problème.....	33
3.2	Méthodes de résolution	40
3.2.1	Description de l'algorithme H	40
3.2.2	Amélioration de l'algorithme.....	43
3.2.3	Algorithme d'affectation de longueur d'onde	52
3.2.4	Complexité des algorithmes.....	54
CHAPITRE IV IMPLÉMENTATION ET ANALYSE DES RÉSULTATS		58
4.1	Détails d'implémentation.....	58
4.2	Jeux de données pour les tests	61
4.2.1	Topologie des réseaux	61
4.2.2	Demandes de trafic	63
4.2.3	Nombre de longueurs d'onde.....	64
4.3	Analyse des résultats des algorithmes.....	64
4.3.1	Algorithme H	64
4.3.2	Algorithme H+	68
4.3.3	Utilisation d'une méta-heuristique : la recherche taboue	70
4.3.4	Temps d'exécution.....	73
4.3.5	Effet de la proportion de demandes robustes.....	74
4.3.6	Effet de la connexité du réseau	76
4.4	Solutions optimales	77
4.4.1	Borne pour le scénario sans panne.....	77
4.4.2	Estimations des bornes de la solution optimale	80

CHAPITRE V CONCLUSION	86
5.1 Synthèse des travaux.....	86
5.2 Limitations des travaux.....	87
5.3 Indications des travaux futurs	88
BIBLIOGRAPHIE	89

LISTE DES FIGURES

Figure 1.1 Une fibre optique	2
Figure 1.2 Illustration d'un chemin optique	3
Figure 1.3 Conversion et continuité de longueur d'onde.....	5
Figure 2.1 Multiplexage de longueur d'onde.....	11
Figure 2.2 Conversion de graphe	28
Figure 3.1 Exemples pour les valeurs de N_k	46
Figure 3.2 Gain en nombre de connexions pour 80 nœuds et $\lambda = 8$	47
Figure 3.3 Gain en nombre de connexions pour 80 nœuds et $\lambda = 16$	48
Figure 3.4 Gain en nombre de connexions pour 80 nœuds et $\lambda = 32$	48
Figure 3.5 Gain en nombre de connexions pour 80 nœuds et $\lambda = 64$	48
Figure 3.6 Exemple pour la première itération	50
Figure 3.7 H+ vu comme une méta-heuristique	51
Figure 4.1 Réseau de 20 nœuds	62
Figure 4.2 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.02$)	64
Figure 4.3 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.02$)	65
Figure 4.4 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.05$)	65
Figure 4.5 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.05$)	65
Figure 4.6 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.1$)	66
Figure 4.7 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.1$)	66
Figure 4.8 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.2$)	66
Figure 4.9 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.2$)	67
Figure 4.10 Gain en pourcentage de demandes bloquées ($n = 20$ nœuds).....	68
Figure 4.11 Gain en pourcentage de demandes bloquées ($n = 40$ nœuds).....	68
Figure 4.12 Gain en pourcentage de demandes bloquées ($n = 80$ nœuds).....	69
Figure 4.13 Blocage en pourcentage ($n = 20$ nœuds)	75
Figure 4.14 Blocage en pourcentage ($n = 80$ nœuds)	76

LISTE DES TABLEAUX

Tableau 4.1 Réseaux utilisés pour les tests	62
Tableau 4.2 Charge en nombre de demandes	63
Tableau 4.3 Pourcentage d'amélioration apportée par la recherche taboue	72
Tableau 4.4 Temps d'exécution en secondes pour $p = 0.2$	73
Tableau 4.5 Nombre de demandes bloquées ($n = 10, p = 0.10$)	79
Tableau 4.6 Nombre de demandes bloquées ($n = 10, p = 0.20$)	79
Tableau 4.7 Nombre de demandes bloquées ($n = 20, p = 0.10$)	79
Tableau 4.8 Nombre de demandes bloquées ($n = 20, p = 0.20$)	80
Tableau 4.9 Nombre de demandes bloquées ($n = 30, p = 0.10$)	80
Tableau 4.10 Estimation des bornes ($n = 10, p = 0.10$)	84
Tableau 4.11 Estimation des bornes ($n = 10, p = 0.20$)	84
Tableau 4.12 Estimation des bornes ($n = 20, p = 0.10$)	84
Tableau 4.13 Estimation des bornes ($n = 20, p = 0.20$)	84
Tableau 4.14 Estimation des bornes ($n = 30, p = 0.10$)	85

LISTE DES SIGLES ET ABRÉVIATIONS

<u>Sigle ou abréviation</u>	
WDM	Wavelength Division Multiplexing (Multiplexage en longueurs d'onde)
DWDM	Dense Wavelength Division Multiplexing (Multiplexage dense en longueurs d'onde)
RWA	Routing and Wavelength Assignment (Routage et affectation de longueurs d'onde)
WA	Wavelength Assignment (Affectation de longueurs d'onde)
ILP	Integer Linear Program (Programme entier linéaire)
LP	Linear Program (Programme linéaire)
<u>Signification</u>	

CHAPITRE I

INTRODUCTION

La fin du vingtième siècle a été marquée par un important développement des télécommunications. Les réseaux actuels transportent des quantités phénoménales d'informations qui peuvent être de la voix, des données ou de la vidéo. Dans ce contexte, on a vu l'explosion de l'offre des services par les opérateurs de télécommunications dont l'objectif principal est d'intégrer différents types de service sur le même réseau, tout en garantissant une certaine qualité de service. Pour ce faire, il est nécessaire de pouvoir transmettre les informations à des vitesses élevées, d'où la naissance des réseaux à haute vitesse. C'est dans ce contexte que sont apparus les réseaux optiques qui, en plus de fournir une grande largeur de bande, sont moins sensibles à l'atténuation du signal à cause de sa nature optique. Actuellement, plusieurs institutions ont installé des réseaux fibrés à l'échelle planétaire. Toutefois, la conception, l'opération et la gestion des réseaux optiques sont des processus longs et complexes. Ce mémoire traite du problème de routage et d'affectation de longueur d'onde (RWA) dans les réseaux optiques, un sous-problème majeur du problème de conception de ces réseaux. Dans ce chapitre d'introduction, nous présentons quelques concepts et principes de base qui nous permettront par la suite d'énoncer les éléments de la problématique. Ensuite, nous précisons nos objectifs de recherche, les résultats escomptés et le plan du mémoire.

1.1 Définitions et concepts de base

Un *réseau optique* est composé d'éléments interconnectés par des fibres optiques. On utilise le terme *optique* parce que les informations sont transportées sous forme lumineuse en lieu et place d'impulsions électriques. Durant un laps de temps, la

présence d'un signal lumineux représente le bit 1 et l'absence de signal, le bit 0. Une *fibres optique* est une fibre de silice qui joue le rôle de médium d'interconnexion permettant de transporter l'information lumineuse dans les réseaux optiques. Elle est composée d'un revêtement protecteur, d'une gaine optique et d'un cœur, comme l'illustre la Figure 1.1. En fait, il s'agit d'un guide d'onde cylindrique. La gaine optique et le cœur sont faits de silice dont les indices sont différents. Cette différence permet la réfraction et le guidage de l'onde sur de grandes distances.

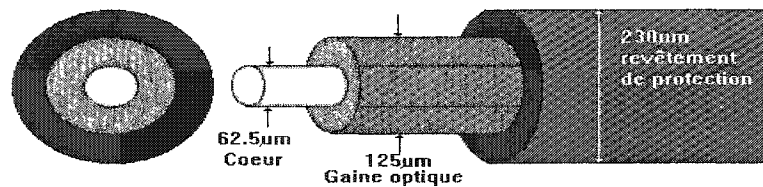


Figure 1.1 Une fibre optique

En jouant sur le diamètre du cœur de la fibre, on peut forcer une fibre à être monomode (moins de dispersion et plus de bande passante) ou multimode. Les fibres sont aussi caractérisées par :

- l'*atténuation* qui représente l'affaiblissement du signal sur une distance donnée et qui rend nécessaire la présence de régénérateurs, elle peut atteindre les 0.2 dB/Km dans la zone des 1550 nm ;
- la *dispersion* qui représente l'étalement du spectre du signal à l'arrivée.

Chaque fibre peut supporter plusieurs longueurs d'ondes différentes. Toutefois, elles doivent être suffisamment séparées (100 ou 200 GHz) pour limiter les interférences entre signaux optiques et permettre aux nœuds actuels de faire un routage adéquat. Actuellement, on retrouve des fibres pouvant supporter jusqu'à 80 canaux optiques multiplexés.

Dans le problème de RWA associé au réseau optique, un *chemin optique* représente un ensemble de liens constituant une chaîne de l'origine à la destination. Comme dans tous les réseaux, il existe deux types de chemins : les chemins *permanents* et les chemins *robustes*. Un chemin robuste est un chemin contenant un trafic qui, s'il a été affecté lors d'une panne, peut être rerouté en utilisant un autre chemin optique. Les chemins permanents, eux, ne bénéficient pas de cette possibilité. En cas de panne les affectant, ils sont simplement bloqués. Dans la Figure 1.2, les liens en gras représentent un chemin optique entre les nœuds 1 et 6. Comme on le voit, il peut y avoir plusieurs chemins optiques possibles pour une connexion, le rôle des algorithmes étant de trouver la combinaison optimale de chemins pour tous les trafics du réseau.

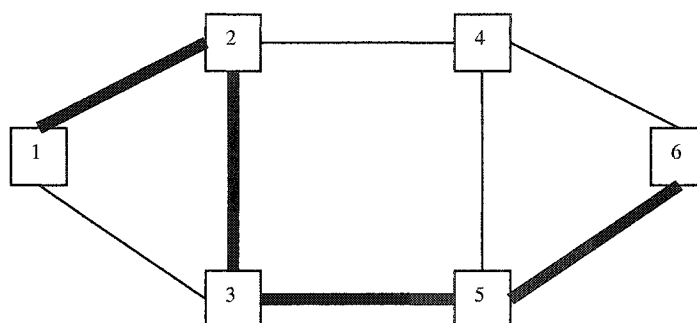


Figure 1.2 Illustration d'un chemin optique

Qu'il soit optique ou d'une autre nature, il peut arriver qu'un réseau refuse l'établissement d'une connexion pour diverses raisons qui peuvent être : la surcharge du réseau, une erreur matérielle ou logicielle, un requis en bande passante impossible à satisfaire, etc. Dans ce cas, on parle de *taux de blocage* correspondant au rapport du nombre de connexions qui ont dû être refusées au nombre total de demandes de connexions. L'objectif de nombreux travaux de recherche est de minimiser le taux de blocage.

Dans les communications optiques, les informations circulent selon une longueur d'onde donnée qui représente la distance, dans la direction de la propagation d'une onde périodique, entre deux points successifs où l'oscillation a la même phase. Chaque fibre

peut supporter plusieurs longueurs d'ondes différentes. Le blocage peut intervenir lorsqu'on ne peut trouver de longueur d'onde disponible sur le chemin optique considéré. Pour acheminer un message d'une source à une destination, il est possible de n'utiliser qu'une seule longueur d'onde ou de se servir d'un ensemble de longueur d'onde. Une fois le chemin déterminé, on peut décider d'appliquer la contrainte de *continuité de longueur d'onde*, ce qui implique qu'il n'y ait pas de conversion de longueur d'onde (Gangxiang, 1999 ; Peng et Wei, 2000 ; Harder *et al.*, 1997). D'autre part, on peut décider de ne pas imposer cette contrainte et utiliser des *convertisseurs de longueurs d'onde*. En fait, si on se rend compte que la longueur d'onde d'entrée du routeur n'est pas disponible à la sortie, on continue d'acheminer le trafic en utilisant une longueur d'onde disponible à la sortie.

La Figure 1.3 représente un réseau composé de six nœuds. On a deux demandes de trafic dont les paires origine/destination sont (1,6) et (3,6). Le chemin sélectionné pour (1,6) est une illustration du routage sans conversion de longueur d'onde. Par ailleurs, le chemin du trafic (3,6) utilise la longueur d'onde λ_1 sur le lien (3,2) mais il y a une conversion au niveau du nœud 2.

La conversion de longueur d'onde a l'avantage de diminuer le taux de blocage car toutes les longueurs d'ondes disponibles sont mises à contribution. Toutefois, les équipements nécessaires devront être installés à chaque nœud du réseau et restent dispendieux. La continuité de longueur d'onde a le mérite d'être moins coûteuse mais elle présente aussi plusieurs inconvénients par rapport à la conversion. Premièrement, le temps d'établissement de la connexion est plus long car il faut trouver une longueur d'onde disponible sur tous les liens qui composent le chemin. En plus de cela, la largeur de bande est sous-utilisée.

Comme bien d'autres équipements dans le domaine des télécommunications, les équipements optiques ne sont pas à l'abri de pannes pouvant aller du bris physique au dysfonctionnement matériel ou logiciel. Dans ces cas, il est important pour les opérateurs de pouvoir continuer à offrir leurs services. La *survivabilité* d'un réseau représente sa capacité à rester en opération lorsque certains de ses éléments sont en

panne, autrement dit la capacité à supporter les demandes malgré les pannes. C'est un enjeu majeur des concepteurs de réseaux.

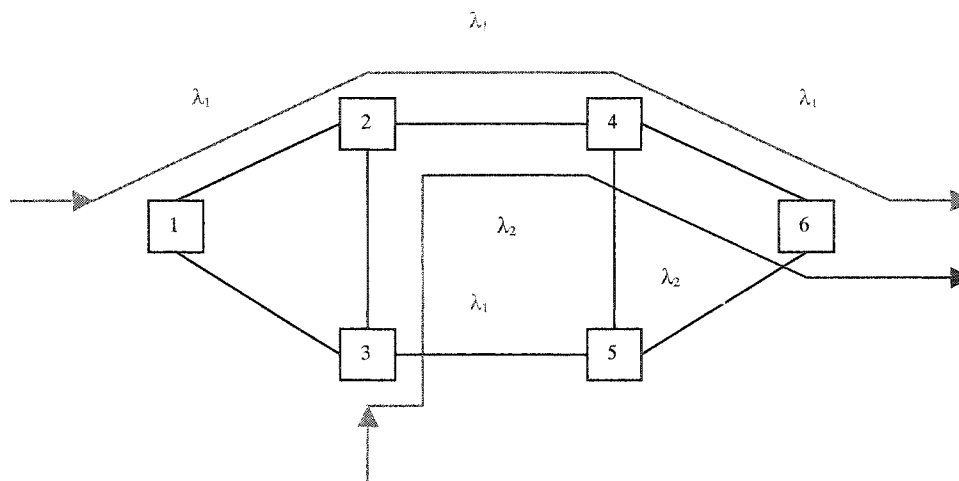


Figure 1.3 Conversion et continuité de longueur d'onde

Enfin, pour un réseau donné, il existe une *topologie physique* et une *topologie logique*. La topologie physique représente l'ensemble des nœuds, leur situation géographique et l'ensemble des liens physiques entre ces nœuds. La topologie logique d'un réseau de communication est représentée par un graphe. Dans ce graphe, les nœuds représentent les nœuds physiques du réseau et les liens sont établis entre deux nœuds seulement s'il y a une demande de trafic entre ces éléments. Il est important de faire correspondre la topologie logique et la topologie physique du réseau en s'assurant que la topologie logique reste connectée en cas de bris de lien simple.

1.2 Éléments de la problématique

Le problème général de planification de réseau optique est complexe et se décompose en plusieurs étapes d'optimisation dont les principales sont :

- caractérisation du trafic : il faut prévoir la demande en trafic ;

- conception topologique : il faut déterminer l'emplacement des nœuds et la topologie d'interconnexion optimale pour le réseau ;
- routage et affectation de longueur d'onde : il faut trouver le chemin et la longueur d'onde ;
- dimensionnement des liens.

En plus de tout cela, il faut aussi tenir compte de la survivabilité du réseau. En effet, il est nécessaire de mettre en place des stratégies efficaces pour assurer le moins de pertes possibles.

Le problème de routage et d'affectation de longueur d'onde peut être scindé en deux sous-problèmes :

- Le *routage* qui consiste à trouver le chemin que doit emprunter un trafic entre la source et la destination. En effet, il est important de déterminer ce chemin eu égard à certaines considérations comme le coût d'exploitation des chemins.
- L'*affectation de longueurs d'onde* qui consiste, une fois le chemin déterminé, à trouver un ensemble de longueurs d'onde disponibles sur les liaisons et qui permettront d'acheminer le trafic de manière optimale. La condition fondamentale dans un réseau optique est que deux chemins qui traversent le même lien physique ne soient pas sur le même canal optique.

Les deux problèmes susmentionnés sont complémentaires. En effet, il ne sert à rien de déterminer un chemin sans affecter de longueur d'onde et on ne peut affecter de longueur d'onde si on n'a pas déterminé de chemin.

Il existe des méthodes de résolution exactes pour trouver une solution optimale au problème de routage et d'affectation de longueur d'onde qu'on désigne par RWA (*Routing and Wavelength Assignment*). Toutefois, elles ne peuvent s'appliquer que sur des problèmes de petite taille. En effet, les problèmes de routage et d'affectation ont chacun été démontrés NP-difficile (Harder *et al.*, 1997), ce qui veut dire que la mise en œuvre de ces méthodes exactes nécessiteraient des efforts de calcul considérables et des temps énormes pour résoudre des problèmes de taille moyenne. Cet état de fait amène à

recourir à des heuristiques permettant de trouver un routage et une affectation dans des temps de calcul raisonnables et en respectant toutes les contraintes imposées.

Selon qu'on se trouve dans une situation de continuité ou de conversion de longueurs d'onde, le problème de RWA est plus ou moins difficile à résoudre. Dans ce mémoire, on considère la continuité de longueur d'onde. Si, en plus de cela, on doit considérer les défaillances possibles des éléments du réseau, on se rend compte que le problème devient encore plus complexe, car il faut trouver des chemins secondaires pour les chemins optiques affectés, tout en faisant attention de ne pas congestionner le réseau. Face à cela, nous constatons le caractère inadéquat des méthodes exactes pour résoudre les problèmes de moyenne ou grande taille.

Dans ce mémoire, on traitera principalement du problème d'affectation de longueur d'onde. On utilisera un algorithme de plus court chemin pour le routage, algorithme qui se présente comme le plus convenable pour l'implémentation et la simulation lors de la recherche.

1.3 Objectifs de recherche

Comme nous l'avons mentionné plus haut, nous nous intéressons au routage et à l'affectation de longueur d'onde dans un réseau optique. Il existe deux types principaux de routage : le *routage statique* qui se base sur une demande statique, et le *routage adaptatif* qui tient compte du niveau de trafic dans le réseau. Dans ce mémoire, nous nous concentrons sur le routage statique. Le premier objectif que nous nous fixons est de proposer un modèle mathématique permettant de représenter le problème de RWA dans un réseau optique en considérant les chemins robustes et permanents. Ce modèle devra aussi considérer un ensemble de valeurs possibles pour les coûts des liens. Ensuite, nous tiendrons compte des caractéristiques du problème pour développer une heuristique dont l'objectif principal sera d'affecter les longueurs d'onde au trafic tout en minimisant le taux de blocage dans le cas d'un réseau dont le coût des liens est fixe. Nous essaierons d'obtenir des résultats proches de l'optimum. Cette heuristique devra

tenir compte de l'affectation initiale du trafic et des cas de pannes de liaisons simples qui sont les plus fréquentes. La dernière étape consistera à appliquer une méta-heuristique sur notre heuristique, ce qui nous permettra de déterminer les coûts des différents liens qui nous permettent d'obtenir la meilleure solution pour notre heuristique. Nous utiliserons la méthode de recherche taboue comme méta-heuristique.

Plusieurs travaux ont déjà été effectués sur la question (Banerjee et Mukherjee, 1996; Inkret *et al.*, 1998; Mohan *et al.*, 2001). Toutefois, ils aboutissent à des résultats plus ou moins satisfaisants. En plus de cela, les travaux entrepris précédemment se sont concentrés sur des scénarios d'erreurs avec chemins robustes seulement. Dans ce mémoire, nous allons tenter de présenter un algorithme d'affectation qui tient compte de chemins robustes et de chemins permanents et qui sera rapide à exécuter.

1.4 Esquisse méthodologique

Pour ce travail, nous débuterons par le développement du modèle mathématique qui s'appuiera sur une publication relative au problème de routage et d'affectation de longueur d'onde (Chamberland, Oulaï et Pierre, 2002). Suite à cela, nous implémenterons l'heuristique proposée dans l'article susmentionné. Nous avons obtenu de bons résultats en appliquant cet algorithme sur un exemple illustratif avec un réseau de sept nœuds. Avec cette implémentation, nous effectuerons une batterie de tests pour donner des indications plus précises sur le comportement de l'algorithme. Nous testerons cet algorithme sur des réseaux de différentes tailles.

Pour juger des performances de l'algorithme, nous le comparerons à d'autres algorithmes connus dans la littérature. Il sera aussi important de définir correctement la topologie des réseaux testés afin de les utiliser pour obtenir les résultats des différents algorithmes et avoir donc une base solide de comparaison. Lors de cette étape d'analyse de performance, l'idéal pour nous serait de pouvoir comparer les résultats de nos algorithmes aux valeurs de la solution exacte. Nous essaierons donc, lorsque c'est possible, d'obtenir la solution optimale du problème. Si la taille du problème ne nous

permet pas de trouver une solution par méthode exacte, nous accentuerons nos efforts sur la recherche d'une borne inférieure pour le taux de blocage. Ces deux indicateurs seront très importants dans l'évaluation de la performance de notre heuristique par rapport au taux de blocage.

Une fois ces performances répertoriées, nous identifierons les lacunes de l'heuristique et tenterons de proposer des améliorations à ces points faibles. Enfin, puisque nous nous intéressons principalement à l'affectation de longueurs d'onde, nous proposerons aussi d'autres algorithmes pour réaliser l'affectation des longueurs d'onde et combler les lacunes de l'algorithme de base que nous avons utilisé. En plus du taux de blocage, ces algorithmes devront respecter des contraintes de temps d'exécution, de robustesse et, dans la mesure du possible, tenir compte d'une certaine équité entre les demandes de connexions.

1.5 Plan du mémoire

Ce mémoire est composé de cinq chapitres. Le deuxième chapitre comporte une présentation générale des réseaux optiques et de la technologie de multiplexage de longueur d'onde WDM ainsi qu'un approfondissement sur les aspects de routage et d'affectation de longueur d'onde. Le chapitre 3 présente le modèle et les algorithmes que nous proposons ainsi qu'une analyse de performance de nos algorithmes. Le chapitre 4 traite de l'implémentation et des résultats ainsi que de leurs comparaisons à des estimations de bornes pour la solution optimale. Finalement, le chapitre 5 conclut notre mémoire avec un rappel des principaux résultats, une présentation des limites de nos algorithmes et des indications de travaux futurs.

CHAPITRE II

ROUTAGE ET AFFECTATION DE LONGUEURS D'ONDE

Depuis la deuxième révolution industrielle, les réseaux de télécommunications participent activement à la vie des sociétés humaines. En effet, c'est à partir de cette époque et des avancées technologiques qu'elle a permise que se sont réellement développés les réseaux télégraphiques et téléphoniques. Les années 1970 voient aussi le développement fulgurant de l'informatique née de la deuxième guerre mondiale. L'objectif des réseaux de télécommunications n'est plus simplement de transporter de la voix, mais aussi d'échanger de l'information, partager des ressources et offrir des services. Avec le développement d'Internet dans les années 80-90, on assiste à une explosion de services et d'utilisateurs. Les réseaux ne sont plus le seul fait des gouvernements, des grosses multinationales et des grandes universités, mais ils sont aussi utilisés par de petites entreprises et des particuliers. Il apparaît donc nécessaire d'avoir des architectures capables de traiter ces débits grandissants, mais aussi des supports physiques avec une grande disponibilité en largeur de bande. Dans ce contexte, la fibre optique présente une solution quasi-parfaite. En effet, les fibres optiques permettent actuellement d'atteindre des taux de transmissions de l'ordre du téraoctet par seconde grâce à la technologie WDM. Il faut aussi mentionner que les fibres optiques sont essentiellement présentes dans les réseaux dorsaux et que l'utilisation actuelle de la largeur de bande n'est pas optimale, dû entre autres aux limitations des équipements électroniques. Dans ce chapitre, nous présentons d'abord un survol de la technologie WDM. Par la suite, nous exposons quelques aspects fondamentaux du problème suivis d'une brève caractérisation de la survivabilité des réseaux optiques. Vient ensuite une

formulation du problème qui précède une synthèse des algorithmes classiques de résolution.

2.1 La technologie WDM

WDM est une technologie de multiplexage de longueurs d'onde qui permet de regrouper plusieurs communications sur une même fibre. Grâce à cette technologie, il est possible d'augmenter considérablement la capacité des réseaux optiques.

2.1.1 Description

La technologie WDM permet d'accroître la largeur de bande des fibres optiques par le multiplexage. Typiquement, une longueur d'onde permet de transporter des informations à un taux avoisinant les 2.5 Gb/s. En multiplexant 40 longueurs d'ondes, on peut transporter jusqu'à 100 Gb/s sur une seule fibre et ainsi de suite. La Figure 2.1 en est une illustration.

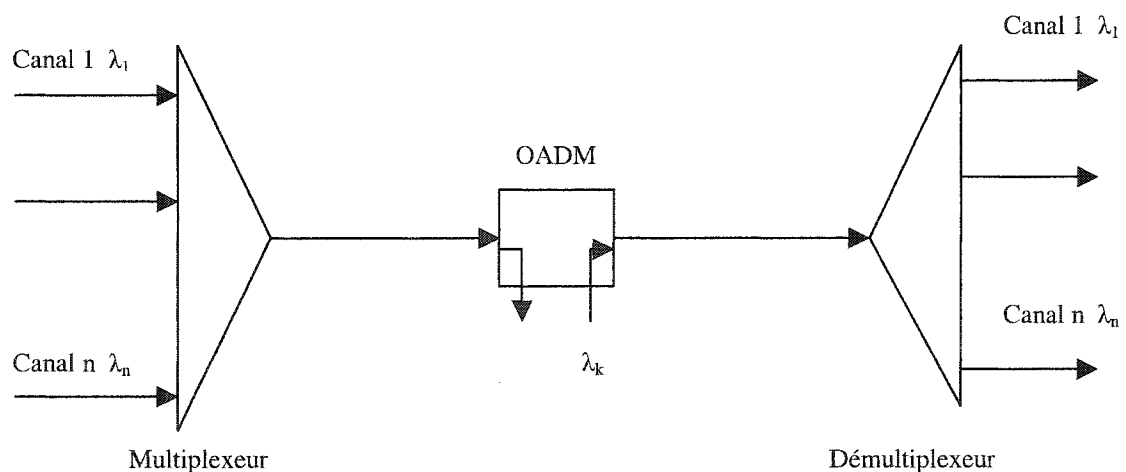


Figure 2.1 Multiplexage de longueur d'onde

Par ailleurs, les liens des réseaux optiques actuels comprennent des dizaines de fibres optiques par lien physique, ce qui permet la réutilisation des longueurs d'onde.

2.1.2 Quelques équipements optiques

Avec l'émergence de WDM, on a aussi vu le développement d'une panoplie d'équipements : OADM, régénérateurs, étoile passive, routeur passif, commutateur passif. Nous allons les décrire brièvement.

OADM

L'OADM (*Optical Add and Drop Multiplexer*) ou multiplexeur optique d'insertion-extraction est un composant qui permet essentiellement de retirer ou d'ajouter les informations contenues sur une seule longueur d'onde parmi un flux de chemins optiques. Il ne requiert pas de démodulation pour opérer.

Régénérateurs

Du fait de l'atténuation du signal optique, il est nécessaire de le régénérer après une certaine distance, typiquement 50 ou 100 km. Il y a quelques années, cette régénération se faisait par un processus relativement long et coûteux : transformation optique-électronique, amplification électronique, transformation électronique-optique. De nos jours, il existe des régénérateurs optiques qui ne nécessitent plus la conversion optique-électronique-optique.

Le routeur passif

Cet équipement permet de diriger chacune des longueurs d'onde de l'entrée vers une sortie prédéfinie. Il permet la réutilisation des longueurs d'ondes. Toutefois, sa matrice de routage est statique.

Le commutateur actif

Il peut effectuer les mêmes actions que le routeur passif, à la différence que sa table de routage est modifiable sur commande. De telles modifications nécessitent un plus grand suivi. En plus de cela, le commutateur actif peut être doté d'une capacité de

conversion de longueur d'onde, c'est à dire qu'un trafic à l'entrée peut être acheminé sur une longueur d'onde différente à la sortie.

L'étoile passive

C'est un dispositif de diffusion. Si un signal se présente à l'une des entrées, le signal est divisé et réparti sur toutes les sorties de l'étoile. Il y a des risques de collision si deux signaux de même longueur d'ondes se présentent en même temps aux entrées de l'étoile.

2.2 Aspects fondamentaux du problème

Dans les années 80, la majorité des réseaux optiques étaient des réseaux de diffusion (Broadcast-and-select). Dans ce type de réseau, la station émettrice envoyait son message à toutes les autres stations qui devaient vérifier si le message leur était adressé. Très vite, les limites de ces réseaux se sont fait sentir avec l'augmentation du nombre de machines dans le réseau (Stern et Krishna, 2000). À la même période émergeaient les réseaux à routage de longueur d'onde qui contribuaient à réduire la charge du réseau. Bien que les réseaux optiques soient dotés d'énormes largeurs de bande, il apparaît nécessaire aux fournisseurs de service d'optimiser l'utilisation de cette capacité. L'objectif principal des opérateurs est de maximiser le nombre de connexions établies sur le réseau, autrement dit minimiser la probabilité de blocage.

2.2.1 Angle d'approche du problème

Comme nous l'avons mentionné précédemment, le problème de RWA se subdivise en deux sous-problèmes : le routage et l'affectation de longueur d'onde. Il est donc possible de résoudre le problème de manière conjointe ou séparée. Toutefois, la résolution du problème global est beaucoup plus complexe (Mokhtar et Azizoglu, 1998 ; Banerjee et Chen, 1996). La résolution séparée (Chamberland, Oulaï et Pierre, 2002) semble plus simple à effectuer. En effet, pour résoudre le problème d'affectation,

on peut partir d'une méthode de routage simple et connue comme le plus court chemin et ensuite développer un algorithme pour l'affectation de longueur d'onde. De même, pour étudier le routage, on peut se baser sur un algorithme d'affectation comme le *Premier Trouvé* qui affecte les longueurs d'onde dans un ordre relatif à un indice. Cependant, le fait de séparer ces deux problèmes ne conduit pas toujours à une solution globale qui soit optimale à cause de l'interdépendance de ces deux problèmes. Précisons aussi que ces deux problèmes ont été classifiés NP-difficiles, c'est à dire qu'on ne peut les résoudre de manière exacte dans un temps polynomial (Harder *et al.*, 1997). Par conséquent, la recherche sur les réseaux optiques s'attelle à proposer des heuristiques qui fournissent des solutions sous-optimales mais au prix d'un effort raisonnable de calcul. Dans ce mémoire, nous considérerons l'approche séparée en fixant comme méthode de routage un algorithme de plus court chemin basé sur une métrique que nous définirons plus loin.

2.2.2 Contrôle du réseau

Dans un contexte de réseaux optiques, les opérations de contrôle peuvent être réparties sur plusieurs nœuds du réseau ou centralisées sur un serveur donné. En mode centralisé, un processeur central qui a une connaissance globale du réseau se charge de toutes les opérations d'affectation de trafic et de reroutage du trafic interrompu. Périodiquement, chaque nœud transmet des informations au nœud central sur son état. Les informations sont ensuite renvoyées aux différents nœuds. L'avantage de ce processus est que le nœud central possède toutes les informations pour faire des calculs efficaces. Cependant, une panne au niveau du seul nœud central rend le réseau entier inopérant. En moyenne, ce type de contrôle nécessite quelques minutes pour le reroutage en cas de pannes. D'autre part, le mode de contrôle distribué a été introduit pour pallier les lacunes du mode centralisé. Bien que la gestion et la coordination soient plus complexes à effectuer, les méthodes réparties assurent des temps de restauration avoisinants les quelques secondes (Stern et Krishna, 2000 ; Peng et Wei, 2000).

Dans un contexte de contrôle réparti, il existe principalement deux approches (Zyane, 2001):

- *L'état de lien* : chaque nœud met à jour la topologie complète du réseau avec l'information sur les longueurs d'ondes en service sur chaque lien. Les informations sont directement échangées entre nœuds adjacents. Lors d'une demande de connexion, le nœud émetteur utilise ces informations pour déterminer un chemin et une longueur d'onde disponible. Il lance une demande de réservation sur le réseau. S'il y a blocage sur un des nœuds empruntés, la connexion est rejetée. S'il y a établissement de la connexion, chaque nœud du chemin envoie des informations aux autres nœuds du réseau pour les mises à jour.
- *Le routage réparti* : chaque nœud met à jour ses tables de routage et reçoit les informations des tables de routage de ses voisins. En connaissant le coût des liaisons pour rejoindre ses voisins, le nœud peut ainsi bâtir sa propre table de routage. Il garde en mémoire les informations sur le chemin le moins coûteux pour rejoindre les différentes destinations ainsi que les longueurs d'ondes associées. Lors d'un appel, si la connexion est établie, chaque nœud envoie des informations à ses voisins pour une mise à jour. En cas d'échec, il peut y avoir une tentative sur une longueur d'onde différente.

2.2.3 Survivabilité des réseaux optiques

Une des préoccupations majeures des concepteurs de réseaux optiques, et de réseaux en général, est de pouvoir faire face aux pannes pouvant survenir dans les réseaux. Cela est d'autant plus important pour les réseaux optiques que leurs débits sont énormes. En effet, puisque ces réseaux se retrouvent assez souvent au niveau dorsal, une panne prolongée peut entraîner la perte d'une grande quantité d'information. L'objectif est donc de pouvoir reconfigurer le réseau très rapidement en utilisant les ressources encore fonctionnelles.

Il existe deux types de pannes : panne de liens et panne de nœuds. Les pannes de liens sont généralement dues à des bris de liens tandis que les pannes de nœuds peuvent avoir diverses causes (Chamberland, 2001). D'autre part, les pannes de nœuds sont plus complexes à traiter car elles impliquent la désactivation de tous les liens liés au nœud, tandis qu'une panne de lien reste circonscrite à ce lien. Aussi, il apparaît plus judicieux de tenir compte des scénarios d'intérêt pour les administrateurs de réseaux. Comme la probabilité d'avoir deux éléments ou plus en panne simultanément est faible, on s'intéresse principalement aux pannes simples. Dans ce mémoire, nous considérerons les pannes simples de liens car elles sont les plus fréquentes.

Il faut aussi mentionner que lors d'une panne, il est possible de faire une reconfiguration totale de toutes les connexions ou simplement de rerouter les connexions touchées par la panne. Bien que la reconfiguration totale puisse permettre de trouver une meilleure solution, il est préférable de garder telles quelles les connexions qui sont encore fonctionnelles et d'essayer d'acheminer les autres. Cela permet de ne pas perturber les connexions qui ne sont pas affectées.

Pour pallier les défaillances des réseaux optiques, deux types de méthodes sont utilisés : la protection et la restauration. La protection consiste à réserver des ressources du réseau qui seront utilisées comme *chemins de secours* (backups) en cas de panne. Cette approche est dite *proactive*. Généralement, les chemins de secours sont disjoints d'arcs des chemins principaux pour garantir la restauration en cas de bris de lien simple (Gangxiang, 1999 ; Alanyali et Ayanoglu, 1998). Voici quelques exemples de protection (Stern et Krishna, 2000) :

- *Protection optique (1+1)* : les données sont envoyées simultanément sur le chemin principal et le chemin de secours. En cas de panne, le récepteur se syntonise sur le chemin de secours.
- *Protection optique (1:1)* : Dans ce cas, un chemin de secours disjoint d'arc est réservé pour chaque chemin principal. Contrairement au (1+1), les informations ne sont pas envoyées simultanément sur les deux canaux. On n'émet sur le chemin de secours que lorsqu'une panne est déclarée sur le chemin principal.

- *Protection optique (1:N)* : N chemins principaux partagent le même canal de protection. En cas de défaillances touchant plusieurs chemins principaux, seul le trafic de plus haute priorité est rerouté.

Comme on peut le constater, la protection est un moyen efficace et rapide de réponse face aux pannes de réseaux. En effet, puisque les chemins de secours sont réservés, le réacheminement se fait directement. Cependant, cette stratégie n'optimise pas l'utilisation de la largeur de bande à cause des réservations inutilisées. Certains algorithmes, que nous présenterons plus loin, se proposent d'accroître le taux d'utilisation de la largeur de bande.

La *restauration*, quant à elle, a pour principe fondamental de calculer le nouveau chemin et d'y affecter les longueurs d'onde seulement après la panne. Cette approche est dite réactive et dynamique. Il existe deux types de restauration (Lima et Stubkjaer, 1999) :

- *Restauration de lien* : dans cette méthode, on se propose de contourner la liaison défectueuse. On considère les deux extrémités de la liaison comme source et destination et on cherche un chemin optimal qui formerait une boucle avec la liaison défectueuse.
- *Restauration de chemin* : c'est la méthode la plus étudiée. Elle consiste simplement à déterminer un nouveau chemin pour le trafic considéré.

Bien que la restauration permette une meilleure utilisation de la largeur de bande, elle a pour principal inconvénient le temps de recherche des éventuels chemins de secours. En plus de cela, il n'y a pas de garantie d'acheminement car les chemins de secours possibles peuvent tous être en conflit de canal, ce qui entraînerait le blocage de la connexion.

2.3 Formulation du problème

Pour pouvoir résoudre ce problème, il est souvent nécessaire de le formuler mathématiquement. Généralement, il y a deux approches qui sont privilégiées :

minimiser le nombre de longueurs d'onde utilisées sur le réseau (Stern et Krishna, 2000 ; Harder *et al.*, 1997) ou minimiser la probabilité de blocage des appels (Mokhtar et Azizoglu, 1996 ; Mokhtar et Azizoglu, 1998 ; Banerjee et Chen, 1996 ; Fumagalli et Valcarengh, 2001) tout en minimisant le coût total d'opération du réseau. Rappelons que dans un contexte de réseau dont les ressources sont connues, minimiser le taux de blocage équivaut à maximiser le nombre de connexions établies. Comme il s'agit de problèmes de minimisation sujets à des contraintes, le modèle de programmes linéaires est très souvent utilisé.

Dans un modèle de programmation linéaire, on cherche les valeurs non négatives des variables qui optimisent une fonction linéaire dite *économique* et qui satisfont un système d'équations ou d'inéquations linéaires représentant les contraintes. Généralement, on représente l'utilisation d'une longueur d'onde par un chemin par une variable binaire. Cela nous ramène donc à une formulation de programme linéaire entier (ILP) dont certains exemples sont présentés dans Modiano et Narula-Tam (2001), Mokhtar et Azizoglu (1996), Fumagalli et Valcarengh (2001). Lorsqu'on abandonne la contrainte entière, on effectue une relaxation linéaire, ce qui ramène à une formulation classique de programme linéaire (LP relaxation). Cette méthode permet d'avoir des indications sur les bornes de la fonction objectif. Ramaswami et Sivarajan (1995) proposent de maximiser le trafic supporté par le réseau. Pour ce faire, ils proposent un ILP qu'ils relâchent en effectuant une LP relaxation. Cela leur permet d'avoir une estimation du nombre maximal de connexions que peut supporter le réseau.

Il existe deux types de ILP très connus. Le premier, appelé *RCA1*, a pour objectif de minimiser le nombre de longueurs d'onde utilisées pour une charge fixe du réseau. C'est une formulation intéressante mais qui nécessite d'ajouter des contraintes pour réduire la complexité car le nombre de variables devient énorme même pour un réseau de taille moyenne. De plus, la solution optimale n'est pas unique. D'autre part, on retrouve un modèle, *RS version*, dont le but est de maximiser la charge du réseau à partir d'un ensemble fixe de longueurs d'onde disponibles. C'est le dual du *RCA1*. Toutefois, *RS version* favorise les trafics de plus courte distance.

Il existe plusieurs types de contraintes qui complexifient le problème de RWA. En voici les principales :

- nombre de longueurs d'onde disponible par fibre ;
- nombre de fibres par lien ;
- conservation de flot ;
- valeurs entières ;
- nombre de demande de connexions ;
- délais de propagation ;
- nombre de sauts.

Comme ce problème est NP-difficile, différentes heuristiques ont été présentées pour le résoudre. La section suivante présente quelques unes d'entre elles.

2.4 Algorithmes classiques de résolution

Dans cette section, nous faisons un tour d'horizon des algorithmes existants. Comme les méthodes de résolution proposées ne sont généralement pas optimales, il est souvent nécessaire de calculer la borne (inférieure ou supérieure) de la variable d'intérêt pour juger de la qualité des solutions obtenues. Il existe une méthode basée sur le théorème de flot maximal-coupe minimale de la théorie des graphes, pour donner une estimation du nombre minimal de longueurs d'onde nécessaire pour supporter un ensemble de trafic donné. Ces bornes servent de base de comparaison pour les résultats obtenus.

2.4.1 Coloriage

Le coloriage est une méthode de résolution qui est applicable à l'affectation de longueur d'onde. On détermine d'abord un ensemble de routes susceptibles de satisfaire la demande. Chaque chemin est représenté par un nœud, et deux nœuds sont reliés lorsque les chemins correspondants partagent au moins une fibre. L'objectif devient d'assigner des couleurs à chaque nœud de telle sorte que deux nœuds adjacents n'aient

pas la même couleur, ce qui résultera en l'absence de conflits entre deux appels. Le *nombre chromatique* représente le nombre minimal de couleur, donc de longueurs d'onde, nécessaires pour satisfaire la demande. Ce problème est aussi NP-difficile. L'algorithme de Dsatur (Stern et Krishna, 2000) permet de trouver une solution de manière simple. Il se base sur le degré de saturation d'un nœud qui représente le nombre de nœuds de couleurs différentes auxquels il est adjacent. En voici les principales étapes :

1. Ranger les nœuds par ordre décroissant de degré.
2. Assigner la première couleur au nœud de degré maximal.
3. Choisir un nœud non coloré de degré de saturation maximal.
4. Colorier le nœud avec la couleur d'indice le plus faible.
5. Si tous les nœuds sont colorés, arrêtez; sinon allez à l'étape 3.

Banerjee et Chen (1996) proposent de réduire la complexité du problème en modifiant le routage au fur et à mesure qu'on assigne des couleurs. En fait, ils proposent de diminuer le degré du nœud courant en reroutant le trafic de telle sorte qu'il y ait moins de nœuds adjacents qui soient déjà colorés. Cette méthode donne de meilleurs résultats que le routage simple par plus court chemin.

2.4.2 Méthodes de protection

Dans cette catégorie, on réserve des ressources pour répondre en cas de panne simple sur le réseau. Gangxiang *et al.* (2000) étudie la possibilité de chemins de secours partagé ou non. Dans le cas de chemins de secours non partagés, chaque connexion a un chemin de rechange qui lui est réservé tandis que, dans le cas de partage, deux chemins qui sont disjoints d'arcs peuvent partager le même lien. Son algorithme se résume ainsi :

- classer arbitrairement les trafics ;
- trouver des chemins par l'algorithme de Dijkstra ;
- prendre une longueur d'onde et chercher tous les chemins principaux ainsi que leur chemins de secours auxquels elle pourrait être affectée ;

- répéter l'étape précédente pour toutes les longueurs d'onde.

Cet algorithme permet de réduire sensiblement le nombre de longueurs d'onde utilisées dans le cadre de backups dédiés.

Mohan *et al.* (2001) proposent une approche très intéressante appelée *primary-backup-multiplexing*. Ils suggèrent de multiplexer sur un même canal, un chemin primaire et des chemins de secours qui ne sont pas relatifs au chemin primaire considéré. Ils proposent deux algorithmes pour le cas où le chemin primaire et son chemin de secours utiliseraient la même longueur d'onde (*PDBWA : Primary Dependent Backup Wavelength Assignment*) et le cas où la longueur d'onde du chemin de secours est indépendante du chemin primaire (*PIBWA : Primary Independent Backup Wavelength Assignment*). Bien que cette technique favorise une meilleure utilisation de la largeur de bande et une baisse de la probabilité de blocage, elle n'assure pas un taux de restauration de 100 %. Elle est efficace principalement dans le cas de connexions de courte durée permettant de minimiser le risque qu'un chemin de secours soit utilisé par un chemin primaire. Les résultats montrent que le PDBWA et le PIBWA réduisent considérablement la probabilité de blocage comparativement à un cas sans multiplexage. Par ailleurs, le PIBWA fournit de meilleurs résultats du fait de l'absence de la contrainte sur la longueur d'onde.

2.4.3 Algorithme génétique et recuit simulé

Mikac *et al.* (1998) appliquent ces deux méthodes au problème d'affectation de longueur d'onde et comparent les résultats obtenus. Définissons brièvement ces techniques qui peuvent être qualifiées d'algorithmes probabilistes de recherche.

Les algorithmes génétiques (AG) sont fondés sur la théorie de la survie des espèces. L'application d'opérateurs génétiques sur des candidats a pour but de produire des candidats plus performants pour la génération suivante. On doit d'abord initialiser la population de solutions qui doit être diversifiée pour avoir plus de chance d'arriver à la solution optimale. Les transitions entre solutions sont probabilistes et une fonction

objectif est utilisée pour évaluer les solutions. Il peut aussi y avoir des actions de croisement ou de mutation de solutions.

D'autre part, le recuit simulé (RS) est un processus physique qui consiste à chauffer un solide jusqu'à ce qu'il fonde avant de le refroidir jusqu'à ce qu'il cristallise. Le refroidissement doit se faire de manière méticuleuse pour éviter d'être pris dans un minimum local de température. Par analogie, RS est une heuristique d'optimisation qui utilise les perturbations pour échapper au piège du minimum local en acceptant de temps à autre des solutions qui détériorent la fonction objectif. La probabilité d'accepter une détérioration est fixée à une valeur élevée au départ, mais elle décroît avec l'évolution de l'algorithme.

L'expérimentation est menée sur un réseau où chaque connexion est associée à un chemin principal et un chemin de secours qui sont disjoints d'arcs. Selon le cas, il sera décidé que ces deux chemins utilisent obligatoirement la même longueur d'onde ou non. La continuité de longueur d'onde est considérée. L'objectif de ce travail est de minimiser la longueur totale des fibres dans le réseau. Pour ce faire, Mikac *et al.* (1998) essaient de maximiser l'utilisation des fibres.

Voici les variables utilisées :

- N_λ , le nombre de longueurs d'onde sur une fibre, typiquement 4 ou 8 ;
- w , le nombre de connexions à établir ;
- C_k , le coût du lien k ;
- E , le nombre de liens dans le réseau ;
- F_k , le nombre de fibres sur le lien k ;
- l_k , la longueur du lien k ;
- $UF_k(i)$, le facteur d'utilisation du lien k pour la solution i ;
- UF , le facteur d'utilisation du réseau.

Les solutions possibles sont encodées de la même manière pour AG et RS. On utilise un vecteur de bits. Dans le cas où on aurait 8 longueurs d'onde par lien, chaque chemin serait encodé avec trois bits. La longueur du vecteur solution est calculée en multipliant le nombre de bits nécessaire pour l'encodage par le nombre de connexions à

établir. Dans le cas où le chemin principal et le chemin secondaire n'utilisent pas obligatoirement la même longueur d'onde, cette longueur est doublée car il faudra aussi représenté les chemins de secours. Concernant AG, le vecteur solution est appelé *chromosome* et les bits sont appelés *gènes*. D'autre part, AG et RS considèrent un ensemble de fonctions communes pour évaluer la solution i dont la fonction objectif $f(i)$:

$$f(i) = \sum_{k=1}^E C_k = \sum_{k=1}^E \left[F_k(i) l_k + a \frac{1}{UF_k(i)} \right] \quad (2.1)$$

$$UF_k(i) = \frac{w_k(i)}{F_k(i) N_\lambda} \quad (2.2)$$

$$UF = \frac{1}{E} \sum_{k=1}^E UF_k(i) \quad (2.3)$$

$w_k(i)$ représente le nombre total de longueurs d'onde utilisées sur le lien k . a est une constante réelle utilisée pour influencer la contribution du facteur d'utilisation. Cette fonction objectif détermine le coût total du réseau. La première partie de la fonction dépend du nombre de fibres utilisé et de la longueur des liens, ce qui influence la minimisation de la longueur totale des fibres. La seconde partie de la fonction accorde une préférence aux solutions avec un grand facteur d'utilisation. Il faut aussi mentionner que, si on ne trouve pas de longueurs d'onde disponibles sur les fibres existantes, on rajoute une nouvelle fibre.

Voici une description des paramètres considérés pour AG :

- On utilise un type de croisement à un lieu de croisement. On choisit aléatoirement un caractère et on échange la partie comprise entre ce caractère et la fin du vecteur solution avec la partie située à la même position dans un autre chromosome. Ce type de croisement s'est montré efficace pour les chromosomes de grande taille. Dans ce cas, la taille des chromosomes varie entre 150 et 450 bits.

- La technique de sélection est celle du tournoi. Un certain nombre de chromosomes sont choisis parmi la population et le meilleur de ce groupe est sélectionné pour participer à la production de la prochaine génération.
- La taille de la population est de 100. Elle doit être assez grande pour augmenter les chances d'arriver à une bonne solution. Des expériences ont montré qu'il est intéressant de choisir la taille de la population entre la taille des chromosomes en bits et le double de cette valeur.
- La probabilité de croisement est de 0.6. Plus cette probabilité est grande, plus l'échange de *qualité* entre chromosome est grand. En revanche, la probabilité de destruction de bons chromosomes augmente aussi.
- La probabilité de mutation est de 0.01. Une grande valeur de cette probabilité peut permettre à un chromosome de retrouver des qualités perdues mais cela peut transformer AG en une recherche aléatoire.
- L'espace de génération, qui représente la fraction de la population qui prend part à la *reproduction* en vue de créer une nouvelle génération, est fixée à 1.

Pour RS, le *calendrier de refroidissement* est défini par la variation du paramètre de contrôle et du nombre de transitions effectuées à travers les itérations. On définit un paramètre de contrôle c_k qui joue le rôle de la température dans le processus physique. D'autre part, L_k représente le nombre de transitions effectuées à la $k^{\text{ième}}$ itération.

$$c_k = \alpha c_{k-1} \quad (2.4)$$

$$L_k = \frac{L_{k-1}}{\beta^{0.1}} \quad (2.5)$$

α est une constante réelle comprise entre 0 et 1. β est aussi une constante réelle strictement supérieure à 1. Dans la pratique, si le nombre de transitions possibles

dépasse 500, on se limite à 500 transitions effectuées. Les valeurs suivantes ont été choisies : $\alpha = 0.9$, $\beta = 1.01$, $c_0 = 10\,000$, $L_0 = 2000$.

Comme conclusion, il est apparu que RS est un algorithme beaucoup plus rapide que AG. Mais les performances de RS par rapport à AG diminuent au niveau du temps d'exécution quand la taille du problème augmente. Toutefois, la qualité des solutions de AG est meilleure comparativement à celles générées par RS.

2.4.4 Algorithme de perte de capacité relative

Zhang et Qiao (1998) proposent un algorithme basé sur la perte de capacité relative (PCR) d'un chemin par rapport à une longueur d'onde. Cet algorithme est indépendant de la topologie du réseau et de la nature du trafic. La capacité d'un lien sur une longueur d'onde λ est définie comme le nombre de fibres sur lesquelles λ est disponible sur le lien. Par conséquent, la capacité de chemin sur λ (WPC : *Wavelength Path Capacity*) représente le minimum des capacités des liens composant le chemin. Après une affectation de chemin, le WPC peut être décrémenté au maximum de 1.

Toutefois, cette mesure peut paraître inappropriée si on considère deux chemins dont les WPC avant établissement de la connexion sont de 1 et 3 par exemple. Il paraît plus indiqué d'utiliser le chemin avec un WPC de 3. Pour l'algorithme MS (*MaxSum*) qui tend à maximiser la somme des WPC, le fait de choisir l'un ou l'autre des chemins n'a aucune influence car la somme sera décrémentée de 1. Pour pallier cela, Zhang et Qiao (1998) suggèrent l'introduction d'une mesure relative, le PCR, qui représente le taux de réduction par rapport à l'état actuel.

La comparaison des résultats se fait avec ceux de l'algorithme MS. PCR permet de réduire la probabilité de blocage. Il faut aussi mentionner que, pour une largeur de bande fixe, la différence de résultats est plus sensible dans le cas avec moins de fibres par lien mais plus de longueurs d'onde par fibre. Par ailleurs, les auteurs ont aussi montré qu'il était possible de réduire la différence de probabilité de blocage liée à l'utilisation de convertisseurs de longueurs d'onde, en combinant une solution multifibres, le PCR et l'utilisation de chemins de secours.

2.4.5 Algorithme de chemins disjoints d'arcs

L'algorithme de chemins disjoints d'arcs (CDA) est basé sur le fait que les chemins utilisant une longueur d'onde commune sont disjoints d'arcs (Manohar *et al.*, 2002). Dans la théorie des graphes, trouver l'ensemble maximal de chemins disjoints d'arcs pour un graphe donné et un ensemble de paires origine/destination est un problème bien connu. Soit $G = (V, E)$ le graphe de la topologie physique. Soit (s_i, t_i) l'ensemble des paires origine/destination, et τ un ensemble de connexions qui doivent être disjoints d'arcs. τ est dit réalisable dans G s'il existe des chemins P_i mutuellement disjoints d'arcs pour chaque paire origine/destination. Le problème de chemins disjoints d'arcs maximum (maximum CDA) revient à trouver le sous-ensemble de τ réalisable de taille maximale. Ce problème est bien connu comme étant NP-difficile.

Ce problème peut être appliqué au RWA. Étant donné un graphe G et un ensemble τ , appelons τ_1, \dots, τ_i une partition de τ telle que chaque τ_i est une solution du maximum CDA. Supposons que pour la première itération, on obtienne τ_1 comme solution. Puisque toutes les connexions de τ_1 sont disjoints d'arcs, on peut leur assigner λ_1 comme longueur d'onde. Par la suite, les auteurs proposent de construire un ensemble τ'_1 tel que $\tau'_1 = \tau - \tau_1$, contenant les connexions non assignées. Une autre itération du maximum CDA sur G et τ'_1 donnera une solution τ'_2 à qui on assignera λ_2 comme longueur d'onde. On pourra répéter ce processus jusqu'à ce que l'ensemble τ'_i soit vide pour un certain i . Mentionnons que cette valeur de i représente le nombre minimum de longueurs d'onde utilisé dans le réseau. Si la solution du maximum CDA est optimale, i représentera le nombre optimal de longueurs d'onde pour le réseau. Le problème revient donc à trouver un moyen de résoudre le maximum CDA. Pour ce faire, il est proposé de fixer une borne supérieure d du nombre de liens dans un chemin. On pose :

$$d = \max (\text{diamètre}(G), |E|^{1/2})$$

Voici le déroulement de l'algorithme :

1. Choisir aléatoirement une connexion.
2. Trouver le plus court chemin.
3. Si la longueur du plus court chemin ne dépasse pas d , retenir la connexion et le chemin et effacer tous les liens utilisés par le chemin. Si la longueur dépasse d , la connexion n'est pas routée.
4. Répéter le processus pour chaque connexion ou jusqu'à ce qu'il ne reste plus de liens dans le réseau.

À la fin de ce processus, on obtient les connexions pouvant utiliser la même longueur d'onde.

En termes de résultats, l'algorithme est comparé à une méthode basée sur la coloration de graphe et la programmation linéaire. L'algorithme étudié produit les mêmes résultats en termes de nombre de longueurs d'onde mais s'exécute beaucoup plus rapidement que l'algorithme de référence, cette différence augmentant avec le nombre de nœuds du réseau.

2.4.6 Approche de conversion de graphe

Ohta et Greca (2001) ont proposé un algorithme de RWA qui minimise le nombre de longueurs d'onde pour une demande statique. Ils considèrent la continuité de longueur d'onde. Leur objectif est de résoudre le problème de routage et celui d'affectation simultanément en convertissant le graphe original et en réduisant les problèmes en un problème de flot à multicommodité.

Soit un réseau représenté par un graphe $G = (V, E)$ permettant d'offrir k longueurs d'onde pour l'établissement des chemins optiques. Si une affectation a déjà été réalisée, la situation revient à considérer k réseaux séparés. Chacun de ces k réseaux a la même topologie que le réseau initial mais n'offre qu'une seule longueur d'onde. Cette décomposition est possible à cause de la continuité de longueur d'onde. Appelons G_1, \dots, G_k les k graphes résultant de la décomposition. Soit $v^{(i)}$ le nœud dans G_i correspondant au nœud v dans le graphe original. Il est possible de combiner ces k graphes de la manière suivante. Premièrement, on ajoute $|V|$ nœuds s_1, \dots, s_v et on crée

des liens de s_v vers $v^{(i)}$ pour tous les $v \in V$ et les $1 \leq i \leq k$. De la même façon, on rajoute $|V|$ nœuds t_1, \dots, t_v et on crée des liens de $v^{(i)}$ vers t_v pour tous les $v \in V$ et les $1 \leq i \leq k$. Ce nouveau graphe est appelé G^* . La Figure 2.2 illustre la transformation avec $k = 2$.

Pour la suite, le flot de chaque commodité et la capacité de chaque lien de G^* valent 1. En outre, le flot sur un lien ne peut prendre que les valeurs 0 et 1. Il en découle les observations suivantes :

- le flot de chaque commodité passe sur une seule route ;
- le flot de chaque commodité ne peut traverser qu'un seul sous-graphe ;
- à travers chaque lien, il ne passe au maximum qu'une commodité.

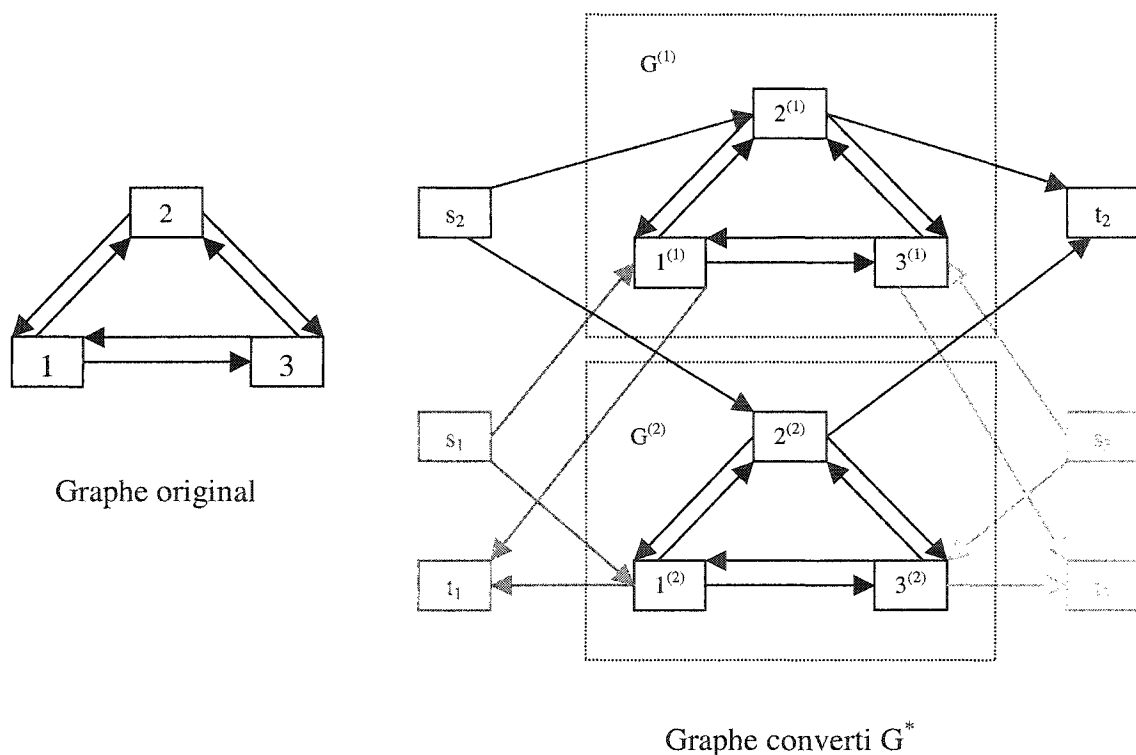


Figure 2.2 Conversion de graphe

On s'aperçoit donc que le chemin emprunté par la commodité représentera le chemin emprunté par le trafic dans le réseau et la longueur d'onde sera représentée par

l'indice du sous-graphe que traversera la commodité. Ohta et Greca (2001) propose des algorithmes permettant de faire la transformation de graphe, trouver la solution initiale et résoudre le problème de multicommodité. Mentionnons aussi qu'ils utilisent un algorithme de plus court chemin pour les multicommodités.

Pour évaluer la performance de l'algorithme proposé, il a été comparé à deux autres algorithmes de la littérature appliqués à un réseau de cinquante nœuds. Le premier algorithme (Nagatsu *et al.*, 1996) est une approche itérative d'amélioration où le routage et l'affectation sont exécutés séparément à chaque itération. Le second algorithme est une combinaison de relaxation linéaire d'un problème de programmation en nombres entiers, de routage aléatoire et de coloriage de graphe (Banerjee et Mukherjee, 1996). Pour un nombre de demandes ne dépassant pas 200, on note une amélioration de près de 27 % par rapport aux autres études. Pour les fortes demandes, l'algorithme se comporte similairement à ceux auxquels il a été comparé.

2.4.7 Algorithme basé sur le délai

Gangxiang *et al.* (1999) ont proposé une série d'algorithmes qui visent à minimiser le nombre total de longueurs d'onde nécessaire au fonctionnement du réseau pour une demande donnée. Ils suggèrent de minimiser le nombre maximal de chemins passant par un lien du réseau.

L'algorithme qui nous intéresse est celui relatif au réseau avec continuité de longueur d'onde. Premièrement, il est suggéré de classer les chemins du plus grand au plus petit en se basant sur les plus courts chemins des demandes. Après affectation des longueurs d'onde, on détermine w , le nombre de longueurs d'onde nécessaire au fonctionnement du réseau dans cette configuration. On sélectionne ensuite aléatoirement un des trafics utilisant la $w^{\text{ième}}$ longueur d'onde et on efface tous les liens où la 1^{ère} longueur d'onde est utilisée. L'étape suivante consiste à essayer de trouver un chemin utilisant la 1^{ère} longueur d'onde pour le trafic sélectionné. Si ce n'est pas possible, on restaure tous les liens où la 1^{ère} longueur d'onde est utilisée. On refait le même processus avec la 2^{ème} longueur d'onde. Si on ne trouve pas de chemins, on

recommence jusqu'à la $(w-1)^{ème}$ longueur d'onde. Si il n'y a aucun chemin trouvé, l'algorithme s'arrête. Dans le cas contraire, la même procédure est répétée pour les autres trafics utilisant la $w^{ème}$ longueur d'onde. Après avoir examiné tous les chemins utilisant la $w^{ème}$ longueur d'onde, on fixe $w = w-1$ et le processus entier est repris.

Pour juger de la performance du réseau, Gangxiang *et al.* (1999) introduisent une mesure de performance appelée *coût de la baisse d'une longueur d'onde*. Il existe une relation entre la baisse du nombre de longueurs d'onde nécessaire au fonctionnement du réseau et l'augmentation du délai. Voici la formule où les indices A et D réfèrent respectivement à l'algorithme étudié et à l'algorithme de Dijkstra :

$$Coût_A = \frac{\left(\frac{Delay_A - Delay_D}{Delay_D} \right) / \left(\frac{W_{max D} - W_{max A}}{W_{max D}} \right)}{W_{max D} - W_{max A}} \quad (2.6)$$

$Delay_A$ et $Delay_D$ représentent le délai mesuré en nombre de sauts. $W_{max A}$ et $W_{max D}$ sont utilisés pour le nombre maximum de longueurs d'onde requises. Mentionnons que l'algorithme de Dijkstra donne une indication sur la valeur maximale de W et la valeur minimale du délai. Pour un bon algorithme, la valeur de ce coût devrait être faible. L'algorithme a été testé sur le réseau ARPA-2 de 21 nœuds et 26 liens. Les résultats ont montré que l'augmentation du nombre de trafics à établir permettait une réduction de ce paramètre.

2.4.8 Autres algorithmes simples

Dans cette section, nous présentons d'abord cinq approches faciles d'utilisation pour l'affectation de longueurs d'onde (Mokhtar et Azizoglu, 1998):

1. PACK : Les longueurs d'onde sont rangées par ordre décroissant d'utilisation. L'algorithme essaie d'acheminer le trafic sur la longueur d'onde disponible la plus utilisée, dans le but de maximiser l'utilisation des longueurs d'onde.

2. SPREAD : L'objectif est d'affecter la longueur d'onde la moins utilisée au trafic pour essayer de distribuer uniformément la charge sur l'ensemble des longueurs d'onde. Cette fois-ci, les longueurs d'onde sont classées dans l'ordre croissant d'utilisation.
3. RANDOM : La recherche de longueurs d'onde se fait aléatoirement. On les essaie l'une après l'autre sans ordre de recherche.
4. EXHAUSTIVE : Le plus court chemin est retenu après l'examen de toutes les longueurs d'ondes. Cette méthode optimise la recherche du plus court chemin mais peut s'avérer fastidieuse.
5. FIXED : L'ordre de recherche est fixe et on affecte le trafic à la première longueur d'onde disponible. Dans ce cas, la méthode est rapide mais on peut ne pas considérer des chemins de coûts inférieurs.

Deux autres approches ont retenu notre attention. Premièrement, Alanyali et Ayanoglu (1998) ont proposé de répondre aux pannes de réseau en étudiant au départ toutes les défaillances possibles. Cela permet de prévoir le comportement du réseau en cas de panne. En effet, pour un trafic fixe, ils proposent de garder en mémoire les reconfigurations possibles pour tous les bris de liens simples. La réponse à un bris de lien simple serait alors immédiate. Pour ce faire, ils préconisent de fournir un surplus de bande passante au réseau. Bien que cette stratégie soit rapide, il est facile de voir qu'elle requiert beaucoup de redondance de fibres.

Par ailleurs, Zhang et Acampora (1995) ont privilégié l'affectation des demandes importantes en premier lieu. Pour eux, il vaut mieux bloquer les petites demandes que les demandes importantes. Cette approche contribue à maximiser la charge du réseau mais n'apparaît pas équitable vis à vis des connexions moins chargées.

Enfin, Rossier-Ramuz *et al.* (2002) ont traité des futurs réseaux optiques intelligents (IWS : *Intelligent Wavelength Services*). Pour ce faire, ils préconisent l'introduction d'un système multi-agents mobiles sur les réseaux optiques. Un agent mobile est une entité logicielle capable de se déplacer d'une machine à une autre et d'y continuer son

exécution. Dans le cas d'un réseau optique, les agents mobiles peuvent être utilisés pour recueillir des informations stratégiques pour le contrôle, la gestion de la charge du réseau et l'affectation de longueurs d'onde. Les premiers résultats de cette approche semblent intéressants, mais il faut veiller à ne pas surcharger le réseau par ces agents mobiles.

CHAPITRE III

ALGORITHMES DE ROUTAGE ET D’AFFECTATION DE LONGUEURS D’ONDE PROPOSÉS

Comme nous l’avons mentionné précédemment, le problème de RWA est un problème NP-difficile. Il est donc important de proposer des heuristiques afin de pouvoir trouver, en des temps de calcul raisonnables, des solutions réalisables de bonne qualité. Dans ce chapitre, nous présentons un modèle mathématique du problème de RWA. Ce modèle a pour objectif de minimiser le taux de blocage et tiendra compte de certaines contraintes comme la continuité de longueur d’onde. Suite à cela, nous proposons un algorithme de RWA qui tient compte des chemins robustes et des chemins permanents. L’évaluation des performances de cet algorithme nous servira de base pour proposer éventuellement de nouvelles heuristiques pour l’affectation des longueurs d’onde.

3.1 Motivations et formulation mathématique du problème

Dans cette section, nous nous proposons de minimiser la quantité de trafic bloqué sur l’ensemble des scénarios possibles. Il est plutôt rare, voire inusité, que des demandes permanentes et des demandes robustes soient considérées simultanément. Cela permet aux clients de choisir l’une ou l’autre option selon leur budget. En plus de cela, nous étudions l’ensemble des scénarios. Pour un opérateur de réseau, il est important de savoir comment se comportera son réseau dans les différents cas de pannes pour une charge donnée. Cela lui permet d’avoir une réponse prête en cas de

bris simple de lien. C'est donc une analyse intéressante aussi bien pour le client que pour l'opérateur de réseau.

Nous proposons un modèle de programmation mixte, c'est-à-dire qui intègre aussi bien des variables entières que des variables réelles. Les hypothèses considérées sont les suivantes :

- continuité de longueurs d'onde ;
- routage selon une méthode de plus court chemin ;
- réseau bi-connexe ;
- liens bidirectionnels.

Nous allons maintenant définir les ensembles utilisés pour ce modèle :

- N , l'ensemble des nœuds du réseau ;
- M , l'ensemble des liens unidirectionnels du réseau ;
- K , l'ensemble des chemins tels que $K = K^P U K^R$ où K^P représente l'ensemble des chemins permanents et K^R , l'ensemble des chemins robustes ;
- $O(k)$ et $D(k)$, l'origine et la destination du chemin $k \in K$;
- Ω , l'ensemble des longueurs d'onde disponibles sur tout le réseau ;
- Ω_{ij} , l'ensemble des longueurs d'onde disponibles sur le lien $(i,j) \in M$, avec $\Omega_{ij} \in \Omega$;

- T , l'ensemble des états du réseau considéré. Le scénario sans faute est noté t_0 . Chaque scénario de panne est caractérisé par une panne simple de liaisons. Il y a donc $m+1$ scénarios possibles dans le réseau avec $m = |M|/2$.

D'autre part, nous utilisons aussi des variables de décision :

- $x_{ij}^{k\lambda}$, une variable binaire telle que $x_{ij}^{k\lambda} = 1$ si et seulement si le chemin $k \in K$ utilise la longueur d'onde $\lambda \in \Omega_{ij}$ et passe à travers le lien $(i,j) \in M$ dans le scénario $t \in T$;
- b^{kt} , une variable binaire telle que $b^{kt} = 1$ si et seulement si le chemin $k \in K$ est bloqué dans le scénario $t \in T$;

- c_{λ}^{kt} , une variable binaire telle que $c_{\lambda}^{kt} = 1$ si et seulement si le chemin $k \in K$ utilise la longueur d'onde $\lambda \in \Omega$ dans le scénario $t \in T$;
- d_j^{kt} , une variable entière positive représentant, dans le scénario $t \in T$, la longueur du chemin entre l'origine du trafic $k \in K$ et le nœud j . Cette longueur représente le coût de ce chemin.

Enfin, la variable a_{ij} désigne la métrique de routage sur le lien $(i,j) \in M$. Le modèle que nous proposons est appelé *RWAP* pour *Routing and Wavelength Assignment Problem*. L'objectif est de minimiser le nombre total de chemins bloqués sur l'ensemble des scénarios possibles.

$$\text{RWAP :} \quad \min_{x,b,c} \sum_{t \in T} \sum_{k \in K} b^{kt} \quad (3.1)$$

Sujet à...

$$\begin{aligned} \text{Des contraintes} \quad & d_i^{kt} = 0 \quad \text{pour } i = O(k) \\ \text{de plus courts} \quad & d_j^{kt} \leq d_i^{kt} + a_{ij} \sum_{\lambda \in \Omega_{ij}} x_{ij}^{kt\lambda} \\ \text{chemins} \quad & \text{pour } k \in K^P, t = t_0, (i,j) \in M \end{aligned} \quad (3.2 \text{ a})$$

$$\begin{aligned} & d_i^{kt} = 0 \quad \text{pour } i = O(k) \\ & d_j^{kt} \leq d_i^{kt} + a_{ij} \sum_{\lambda \in \Omega_{ij}} x_{ij}^{kt\lambda} \\ & \text{pour } k \in K^R, t \in T, (i,j) \in M \end{aligned} \quad (3.2 \text{ b})$$

$$\begin{aligned} \text{Une contrainte} \quad & \sum_{(i,j) \in M} \sum_{\lambda \in \Omega_{ij}} x_{ij}^{kt\lambda} - \sum_{(j,i) \in M} \sum_{\lambda \in \Omega_{ji}} x_{ji}^{kt\lambda} = \begin{cases} 1 - b^{kt} & \text{si } j \in D(k) \\ b^{kt} - 1 & \text{si } j \in O(k) \\ 0 & \text{sinon} \end{cases} \\ \text{de conservation} \quad & (j \in N, k \in K, t \in T) \\ \text{de flot} \quad & \end{aligned} \quad (3.3)$$

La relation (3.1) minimise la fonction objectif qui exprime le taux de blocage. La relation (3.2) implique l'utilisation du plus court chemin par le trafic, ce qui permet de minimiser les coûts d'opérations du réseau. Par ailleurs, la relation (3.3) représente la contrainte de conservation de flot. Si dans le scénario $t \in T$, le chemin $k \in K$ est bloqué, la variable b^{kt} vaut 1 et le membre de droite de la relation (3.3) vaut 0. Dans le cas contraire, on retrouve des contraintes classiques de conservation de flot. D'autres contraintes viennent s'ajouter à notre problème : ce sont les contraintes de continuité de longueur d'onde, les contraintes d'intégrité sur certaines variables et les contraintes de pannes de réseaux que nous allons expliciter.

Contraintes de continuité de longueur d'onde

$$\sum_{k \in K} x_{ij}^{kt\lambda} \leq 1 \quad ((i,j) \in M, \lambda \in \Omega_{ij}, t \in T) \quad (3.4)$$

$$\sum_{\lambda \in \Omega} c_{\lambda}^{kt} \leq 1 \quad (k \in K^P, t = t_0) \text{ et } (k \in K^R, t \in T) \quad (3.5)$$

$$x_{ij}^{kt\lambda} \leq c_{\lambda}^{kt_0} \quad ((i,j) \in M, \lambda \in \Omega_{ij}, k \in K^P, t \in T) \quad (3.6)$$

$$x_{ij}^{kt\lambda} \leq c_{\lambda}^{kt} \quad ((i,j) \in M, \lambda \in \Omega_{ij}, k \in K^R, t \in T) \quad (3.7)$$

Les variables c_{λ}^{kt} et $x_{ij}^{kt\lambda}$ jouent respectivement le rôle de paramètre de contrôle de la continuité de longueur d'onde et de variables de chemin. La relation (3.4) implique que chaque longueur d'onde sur un lien ne soit utilisée que par un seul trafic, condition importante dans les réseaux optiques actuels. Au cas où on utiliserait un multiplexage temporel, cette condition pourrait être relâchée. La relation (3.5) assure la continuité de longueur d'onde sur tout le réseau. D'autre part, les relations (3.6) et (3.7) assurent que la variable $x_{ij}^{kt\lambda}$ soit toujours inférieure ou égale à la variable c_{λ}^{kt} . En effet, la valeur de c_{λ}^{kt} influence la valeur de la variable $x_{ij}^{kt\lambda}$. Si un trafic a pu être affecté dans un scénario

t , c_λ^{kt} vaut 1 et $x_{ij}^{kt\lambda}$ pourra prendre les valeurs 0 ou 1, ce qui indique le chemin suivi par ce trafic. Par contre, si un trafic n'a pas pu être affecté dans un scénario t , c_λ^{kt} vaut 0 et $x_{ij}^{kt\lambda}$ vaudra toujours 0 car le trafic n'emprunte aucun lien. Plus spécifiquement, la relation (3.6) concerne les chemins permanents. Si on ne trouve pas d'affectation pour un trafic permanent donné dans le cas sans panne, la variable $c_\lambda^{kt_0}$ vaut 0. Dans ce cas, $x_{ij}^{kt\lambda}$ vaudra toujours 0 quel que soit le scénario considéré, car les chemins permanents ne bénéficient pas du reroutage. Ces deux contraintes, ajoutées à la contrainte (3.5), induisent implicitement que chaque trafic n'utilise qu'une et une seule longueur d'onde par fibre, condition qui découle de la continuité de longueur d'onde car :

$$\begin{aligned} x_{ij}^{kt\lambda} \leq c_\lambda^{kt} &\Rightarrow \sum_{\lambda \in \Omega_{ij}} x_{ij}^{kt\lambda} \leq \sum_{\lambda \in \Omega} x_{ij}^{kt\lambda} \leq \sum_{\lambda \in \Omega} c_\lambda^{kt} \leq 1 \\ &\Rightarrow \sum_{\lambda \in \Omega_{ij}} x_{ij}^{kt\lambda} \leq 1 \end{aligned} \quad (3.8)$$

$$((i,j) \in M, k \in K, t \in T)$$

Contraintes d'intégralité :

$$x \in \{0,1\}^{|M||K||T||\Omega|}, b \in \{0,1\}^{|K||T|}, c \in \{0,1\}^{|K||T||\Omega|} \quad (3.9)$$

Comme nous l'avons déjà mentionné, les variables x , b et c sont des variables binaires entières. Cependant, on peut noter que la contrainte d'intégralité sur les variables b est inutile. En effet, ces variables interviennent dans la contrainte (3.3). Si les variables x sont contraintes à l'intégralité, la différence du membre de gauche de la contrainte (3.3) est entière et le membre de droite est aussi entier car c'est une égalité. Par conséquent, les variables b sont aussi entières.

Contraintes des scénarios de pannes :

Comme nous l'avons dit, nous considérons les scénarios d'intérêt pour les planificateurs de réseaux. Dans notre cas, nous nous concentrons sur les pannes de liens simples qui sont les plus probables et nous notons:

$$\{(i,j) \in M : i < j\} = \{(i_1, j_1), (i_2, j_2), \dots, (i_{|M|/2}, j_{|M|/2})\}$$

En supposant que le lien (i_l, j_l) soit bi-directionnellement défectueux dans le scénario t_l pour l allant de 1 à $|M|/2$, on peut écrire :

$$\begin{aligned} x_{i_1 j_1}^{k t_1 \lambda} &= 0, \quad x_{j_1 i_1}^{k t_1 \lambda} = 0 & (\lambda \in \Omega_{i_1 j_1}) \\ x_{i_2 j_2}^{k t_2 \lambda} &= 0, \quad x_{j_2 i_2}^{k t_2 \lambda} = 0 & (\lambda \in \Omega_{i_2 j_2}) \\ &\vdots \\ x_{i_{|M|/2} j_{|M|/2}}^{k t_{|M|/2} \lambda} &= 0, \quad x_{j_{|M|/2} i_{|M|/2}}^{k t_{|M|/2} \lambda} = 0 & (\lambda \in \Omega_{i_{|M|/2} j_{|M|/2}}) \end{aligned} \quad (3.10)$$

pour $k \in K^R, (i_l, j_l) \in M$

$$\begin{aligned} x_{ij}^{k t_1 \lambda} &\leq 1 - \sum_{\lambda' \in \Omega_{i_1 j_1}} (x_{i_1 j_1}^{k t_0 \lambda'} + x_{j_1 i_1}^{k t_0 \lambda'}) \\ x_{ij}^{k t_2 \lambda} &\leq 1 - \sum_{\lambda' \in \Omega_{i_2 j_2}} (x_{i_2 j_2}^{k t_0 \lambda'} + x_{j_2 i_2}^{k t_0 \lambda'}) \\ &\vdots \\ x_{ij}^{k t_{|M|/2} \lambda} &\leq 1 - \sum_{\lambda' \in \Omega_{i_{|M|/2} j_{|M|/2}}} (x_{i_{|M|/2} j_{|M|/2}}^{k t_0 \lambda'} + x_{j_{|M|/2} i_{|M|/2}}^{k t_0 \lambda'}) \end{aligned} \quad (3.11)$$

pour $k \in K^P, (i_l, j_l) \in M, \lambda \in \Omega_{ij}$

$$\begin{aligned}
x_{ij}^{kt_1\lambda} &\geq x_{ij}^{kt_0\lambda} \left[1 - \sum_{\lambda' \in \Omega_{i_1j_1}} (x_{i_1j_1}^{kt_0\lambda'} + x_{j_1i_1}^{kt_0\lambda'}) \right] \\
x_{ij}^{kt_2\lambda} &\geq x_{ij}^{kt_0\lambda} \left[1 - \sum_{\lambda' \in \Omega_{i_2j_2}} (x_{i_2j_2}^{kt_0\lambda'} + x_{j_2i_2}^{kt_0\lambda'}) \right] \\
&\vdots \\
x_{ij}^{kt_{|M|/2}\lambda} &\geq x_{ij}^{kt_0\lambda} \left[1 - \sum_{\lambda' \in \Omega_{i_{|M|/2}j_{|M|/2}}} (x_{i_{|M|/2}j_{|M|/2}}^{kt_0\lambda'} + x_{j_{|M|/2}i_{|M|/2}}^{kt_0\lambda'}) \right]
\end{aligned} \tag{3.12}$$

pour $k \in K^P, (i_l, j_l) \in M, \lambda \in \Omega_{ij}$

Les contraintes (3.10) imposent à chaque chemin robuste de contourner le lien brisé, quel que soit le scénario de panne considéré. Les contraintes (3.11) s'appliquent dans le cas de chemins permanents. Elles permettent de désactiver totalement un chemin permanent qui aurait été affecté par une panne. Dans ce cas, la valeur de la sommation est 1 et la variable $x_{ij}^{kt_1\lambda}$ prend la valeur 0 quel que soit le lien considéré. Par ailleurs, les contraintes (3.12) permettent aux chemins permanents qui n'ont pas été affectés par la panne de rester actifs. Dans ce cas, la sommation vaut 0 et si le chemin était actif dans le scénario sans panne, la variable $x_{ij}^{kt_0\lambda}$ vaut 1. Alors, la contrainte (3.12) impose à la variable $x_{ij}^{kt_1\lambda}$ d'être supérieure ou égale à $x_{ij}^{kt_0\lambda} = 1$. Comme $x_{ij}^{kt_1\lambda}$ est binaire, elle prendra la valeur 1. Il faut remarquer que les contraintes (3.12) ont été écrites sous forme d'inégalité. Toutefois, on peut écrire les contraintes (3.12) avec des égalités. Dans ce cas, si un trafic permanent n'est pas touché, $x_{ij}^{kt_1\lambda}$ vaudra 1 sur les liens empruntés par le trafic et 0 sur tous les autres liens. L'avantage de l'inégalité, c'est qu'on admet des solutions différentes de la solution initiale, ce qui sera profitable lorsqu'on utilisera le logiciel CPLEX pour calculer les bornes en relâchant certaines contraintes. Moyennant quelques modifications, ce genre de contraintes pourrait être

appliquées dans le cas de pannes de nœuds simples, mais cela n'est pas l'objet de ce mémoire. Enfin, on peut constater que le modèle RWAP est non linéaire à cause des contraintes de pannes de liens.

Comme RWAP est NP-difficile (Gary et Johnson, 1979), nous proposons, dans la section suivante, une heuristique dont l'objectif est de trouver de bonnes solutions réalisables en un temps d'exécution raisonnable.

3.2 Méthodes de résolution

L'algorithme principal que nous proposons est une heuristique dénotée H . Cette heuristique considère le problème en deux étapes. La première étape trouve le routage et l'affectation de longueur d'onde pour le scénario sans panne, tandis que la deuxième étape se charge des cas de pannes.

3.2.1 Description de l'algorithme H

Voici les principales étapes de l'algorithme :

Étape 1 : Scénario sans panne

Étape 1.1 : Trouver les plus courts chemins

Pour chaque paire $i, j \in N$,

Trouver les plus courts chemins entre i et j en considérant la métrique de routage a_{ij} pour tous les liens $(i, j) \in M$.

Étape 1.2 : Router les trafics

Pour chaque trafic $k \in K$,

Router k selon le plus court chemin trouvé à l'Étape 1.1.

Étape 1.3 : Affecter les longueurs d'onde aux trafics

Pour chaque trafic $k \in K$,

Affecter une longueur d'onde notée $\lambda(k)$ au chemin considéré en tenant compte de la continuité de longueur d'onde.

Si une longueur d'onde est trouvée, faire

$$b^{kt_0} = 0;$$

$$c_{\lambda(k)}^{kt_0} = 1;$$

$$c_{\lambda}^{kt_0} = 0 \quad \text{pour } \lambda \in \Omega \setminus \{\lambda(k)\}$$

Sinon

S'il existe un autre plus court chemin pour k ,

Essayer ce chemin

Sinon pour tout $t \in T$, faire

$$b^{kt} = 1;$$

$$c_{\lambda}^{kt} = 0 \quad \text{pour } \lambda \in \Omega$$

Étape 1.4 : Traiter les variables $x_{ij}^{kt_0\lambda}$

Pour chaque trafic $k \in K$ et $(i,j) \in M$

Étape 1.4.1 :

Si le trafic k n'est pas affecté dans le scénario sans pannes,

$$x_{ij}^{kt\lambda} = 0 \quad \text{pour } (i,j) \in M, t \in T, \lambda \in \Omega$$

Si le trafic k passe par le lien (i,j) et utilise λ , faire

$$x_{ij}^{kt_0\lambda} = 1$$

Sinon

$$x_{ij}^{kt_0\lambda} = 0$$

Étape 2 : Les scénarios de panne

Pour tous les scénarios allant de t_1 à $t_{|T|-1}$ et les chemins affectés dans le scénario sans panne,

Étape 2.1 : Trouver le routage des chemins permanents

Pour chaque trafic $k \in K^P$,

Si le chemin k utilise un élément défectueux, faire

$$b^{kt} = 1;$$

$$c_{\lambda}^{kt} = 0 \quad \text{pour } \lambda \in \Omega$$

$$x_{ij}^{kt\lambda} = 0 \quad \text{pour } \lambda \in \Omega_{ij}, (i, j) \in M$$

Sinon

$$x_{ij}^{kt\lambda} = x_{ij}^{kt_0\lambda} \quad \text{pour } \lambda \in \Omega_{ij}, (i, j) \in M$$

Étape 2.2 : Trouver le routage des chemins robustes

Étape 2.2.1 :

Pour chaque trafic $k \in K^R$,

Si k n'est pas affecté par la panne,

$$b^{kt} = 0;$$

$$c_{\lambda}^{kt} = c_{\lambda}^{kt_0} \quad \text{pour } \lambda \in \Omega$$

$$x_{ij}^{kt\lambda} = x_{ij}^{kt_0\lambda} \quad \text{pour } \lambda \in \Omega_{ij}, (i, j) \in M$$

Étape 2.2.2 :

Pour chaque trafic $k \in K^R$,

Si k est affecté par la panne,

Trouver un plus court chemin de l'origine à la destination.

Étape 2.2.3 :

Pour tous les trafics reroutés à l'Étape 2.2.2,

Affecter une longueur d'onde notée $\lambda(k)$ au chemin considéré en tenant compte de la continuité de longueur d'onde et des longueurs d'onde déjà affectées.

Si une longueur d'onde est trouvée, faire

$$b^{kt} = 0;$$

$$c_{\lambda(k)}^{kt} = 1;$$

$$c_{\lambda}^{kt} = 0 \quad \text{pour } \lambda \in \Omega \setminus \{\lambda(k)\}$$

Sinon

S'il existe un autre plus court chemin k ,

Essayer ce chemin

Sinon, faire

$$\begin{aligned} b^{kt} &= 1; \\ c_{\lambda}^{kt} &= 0 \quad \text{pour } \lambda \in \Omega \end{aligned}$$

Étape 3 : Fin de l'algorithme

Retourner les vecteurs b , c et x . La valeur de la solution est donnée par la fonction objectif (3.1).

Dans cet algorithme, les plus courts chemins sont calculés en utilisant l'algorithme de Dijkstra (Stern et Bala, 2000). L'affectation de longueurs d'onde, qui est un problème NP-difficile, peut être résolu approximativement en utilisant des heuristiques. Dans notre cas, nous avons transformé le WA en un problème de coloration de graphe associé dont l'algorithme est présenté à la section 3.2.3. Comme on peut le constater, la partie importante du travail se fait à la première étape, donc à l'état sans panne du réseau. C'est à cette étape que le calcul des chemins et l'affectation initiale se font. À l'étape 2 et pour les scénarios de pannes, on se base sur la configuration initiale du réseau pour effectuer les opérations. Lors d'une panne, on essaie de réaffecter les trafics actifs dans le scénario sans panne et qui empruntent le lien défectueux. Si le trafic est permanent, il est bloqué et, dans le cas contraire, on recalcule les plus courts chemins avec la nouvelle configuration du réseau. Par conséquent, on remarque que la réalisation de l'étape 1 est primordiale à la suite de l'algorithme. La quantité de trafic bloquée dans les scénarios de pannes sera toujours supérieure ou égale à la quantité de trafic bloquée dans le scénario sans panne.

3.2.2 Amélioration de l'algorithme

Comme nous savons qu'il nous sera difficile d'obtenir la solution optimale pour les problèmes d'une certaine taille, nous avons décidé de proposer une méthode pour améliorer les résultats de l'algorithme H.

Principe de la méthode

C'est une méthode empirique qui pourrait se résumer par la phrase suivante : « *En en considérant moins, vous en satisferez plus !* ». Sortons de la sphère philosophique pour expliquer notre idée. Premièrement, rappelons que l'objectif de ce mémoire est de minimiser le nombre de connexions bloquées à travers tout le réseau. Par conséquent, on ne fait aucune différence entre les connexions car c'est le nombre qui compte. Prenons l'exemple d'un étudiant qui a plusieurs dépenses à effectuer: voyage 900\$ (V-900), livre de réseau 100\$ (L-100), scolarité 100\$ (S-100) et quatre autres dépenses de 100\$ chacune (D1-100, D2-100, D3-100, D4-100). Bien évidemment, notre jeune étudiant ne dispose que d'un budget limité. Ses parents, qui représentent sa seule source possible de revenu sur le moment, mettent à sa disposition une somme de 1000\$ pour toutes ces dépenses qui coûtent 1500\$ au total. L'instinct de jeunesse le pousse à voyager en premier (V-900), mais il se rend compte qu'il ne lui restera que 100\$ pour effectuer les six autres dépenses de 600\$. Alors il décide d'effectuer le plus de dépenses possibles. Pour cela, il surseoit à son voyage et effectue les six autres dépenses. Comme on peut le constater, le voyage V-900 aurait posé un problème car l'étudiant n'aurait pu effectuer au maximum que deux dépenses alors qu'il a pu en faire six. Le fait de réaliser V-900 aurait donc été pénalisant pour son objectif de réaliser le plus grand nombre de dépenses.

Partant du même principe, nous pouvons dire que l'affectation d'une longueur d'onde λ à un trafic k pénalise d'une manière ou d'une autre le nombre de connexions affectées car aucune autre connexion ne pourra utiliser la longueur d'onde λ sur le chemin emprunté par k . L'affectation de longueur d'onde se fait de manière séquentielle sur les trafics. On peut donc penser qu'il y aurait un ordonnancement des trafics qui donnerait une solution optimale. Comme cet ordonnancement est difficile à trouver, nous allons procéder autrement. Nous considérons le scénario sans panne. Les chemins sont calculés en utilisant le même algorithme de Dijkstra. Supposons qu'on ait $k = |K|$ trafics à affecter et que l'algorithme H donne comme résultat n_a trafics affectés. Dans le même ensemble K , nous allons considérer un sous-ensemble de connexions K'

appartenant à K et contenant $k - \alpha$ connexions. L'hypothèse est qu'il existe des α tels que le résultat n_a' , obtenu en appliquant l'algorithme H sur l'ensemble K' , soit supérieur à n_a . Autrement dit, si un ensemble de 500 connexions entraîne une affectation de 300 connexions, il serait possible qu'en considérant 450 demandes, ce qui revient à supprimer 50 demandes au départ, on puisse en affecter 350. Il y a donc toute une plage de α à explorer concernant le nombre de connexions à supprimer, allant de 0 à $n_b = k - n_a$, le nombre de connexions bloquées lors de l'affectation initiale. En effet, si $\alpha > n_b$, on est certain d'obtenir une solution moins bonne car on désactive déjà plus de connexions que le nombre bloqué dans l'affectation initiale. Une fois le principe acquis, il nous faut trouver un indice qui permettra de déterminer si une connexion doit être désactivée ou non.

Indice proposé

L'indice proposé est un estimateur du nombre de longueurs d'onde nécessaires à l'établissement d'une connexion k , dénoté n_k . Nous allons partir du principe que, l'affectation d'une connexion dont l'établissement ainsi que l'établissement de toutes les connexions adjacentes nécessite un grand nombre de longueurs d'onde, est pénalisante. Pour trouver ce nombre, on considère le trafic k et tous les trafics avec lesquels k partagent au moins un lien. Ce problème étant lui-même NP-difficile, nous nous sommes référés à une borne inférieure (Stern et Bala, 2000). La figure (3.1) illustre ces valeurs. Prenons l'exemple de la connexion noire. Elle est adjacente aux connexions rouge et grise qui sont elles-mêmes adjacentes entre elles. Le nombre minimal de longueurs d'onde nécessaires à l'établissement de la connexion noire est de 3. Pour la connexion verte, elle n'est adjacente qu'à la rouge d'où la nécessité d'avoir au moins deux longueurs d'ondes disponibles par fibres. Enfin, la connexion bleue n'a aucune adjacence ce qui entraîne une valeur de n_k égale à 1.

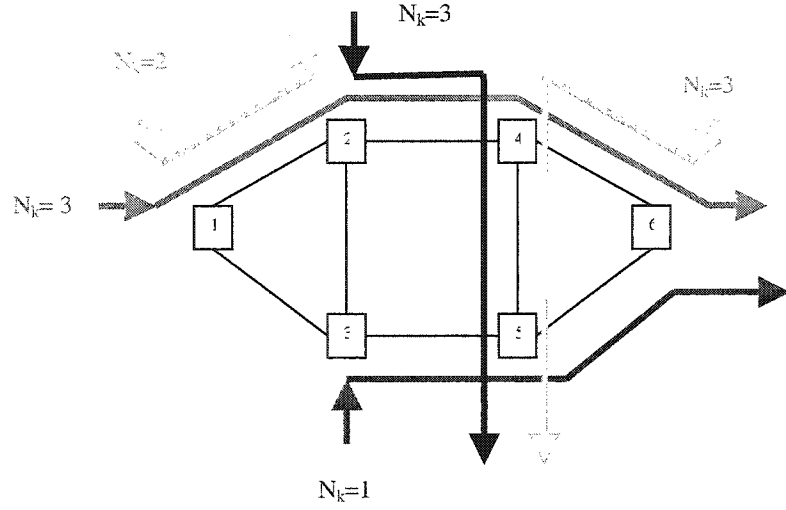


Figure 3.1 Exemples pour les valeurs de N_k

On définit $d(o,d)$ comme étant la distance en nombre de sauts optiques entre une origine o et une destination d . La distance de la connexion 1-6 est de 3. \sum^d représente la *distance inter-nodale totale* du réseau et s'exprime comme suit :

$$\sum^d = \sum_{i=1}^K d(o_i, d_i) \quad (3.12)$$

Dans le cas de la connexion 1-6, cette distance inter-nodale totale est de 7 sauts optiques car on ne considère que les connexions adjacentes. La capacité minimale utilisée pour router efficacement tout ce trafic correspond à \sum^d lien-canaux. En supposant que chaque lien unidirectionnel du réseau contienne f fibres qui supportent chacune $w = |\Omega|$ longueurs d'onde, le réseau peut offrir au maximum $|M|$ liens qui contiennent fw canaux donc $|M|fw$ liens-canaux. Il faut donc que :

$$|M|fw \geq \sum^d \Leftrightarrow w \geq \frac{\sum^d}{|M|f} \quad m \neq 0, f \neq 0 \quad (3.13)$$

On obtient donc une borne inférieure pour le nombre de longueurs d'onde nécessaires à l'établissement de l'ensemble des connexions. Puisque nous allons utiliser

cette borne pour comparer les différents trafics, nous pouvons omettre la division car m et f sont des constantes. On aura donc :

$$n_k = \sum^d \quad (3.14)$$

À chaque itération de notre méthode, nous allons supprimer la connexion qui aura la plus grande valeur de n_k . Chaque fois qu'une connexion est désactivée, on met à jour les indices n_k . On reclasse ensuite les connexions par ordre décroissant de n_k avant de procéder à une affectation.

Déroulement de la méthode

Supposons que nous ayons un réseau avec $k = |K| = 3000$ demandes dont $b = |B| = 1000$ étaient bloquées après l'affectation initiale. Cela suppose qu'on pourrait retirer jusqu'à 1000 connexions et qu'on devrait répéter l'algorithme H 1000 fois pour évaluer la solution à chaque itération. Nous l'avons essayé et cela était très long. Nous nous sommes basés sur deux observations que nous avons effectuées sur les figures 3.1 à 3.4 pour raccourcir le temps de calcul. Sur ces figures, le gain représente l'amélioration de la solution en désactivant un certain nombre de connexions par rapport à la solution initiale. Ce gain est exprimé en nombre de connexions.

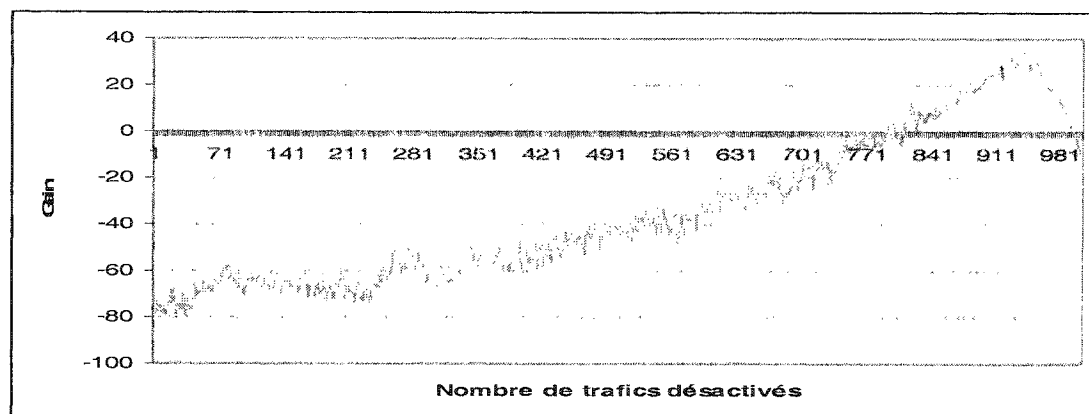


Figure 3.2 Gain en nombre de connexions pour 80 nœuds et $\lambda = 8$

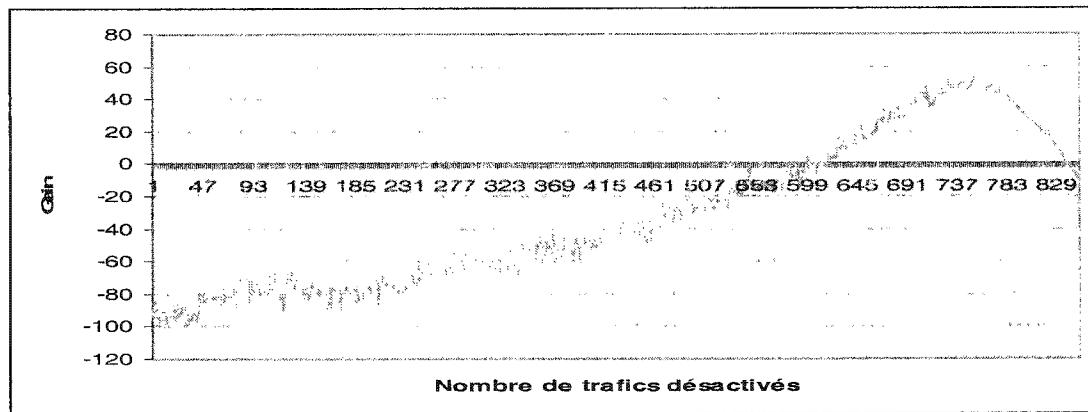


Figure 3.3 Gain en nombre de connexions pour 80 nœuds et $\lambda = 16$

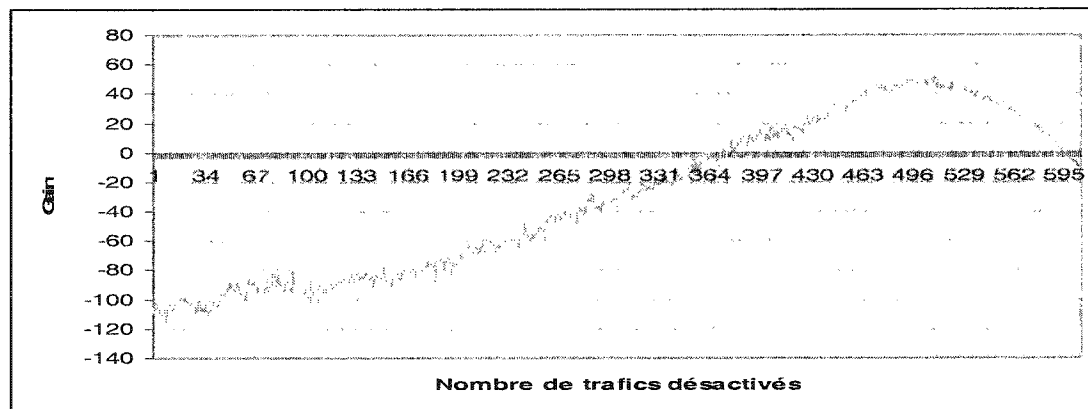


Figure 3.4 Gain en nombre de connexions pour 80 nœuds et $\lambda = 32$

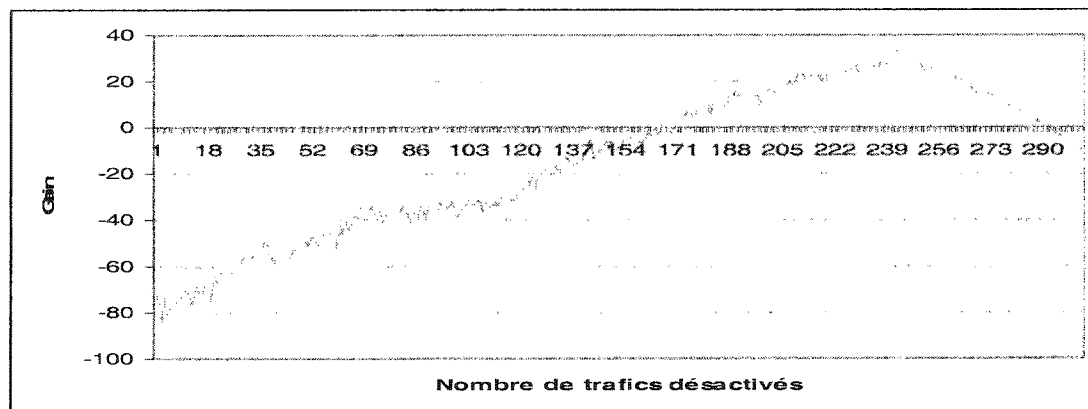


Figure 3.5 Gain en nombre de connexions pour 80 nœuds et $\lambda = 64$

Nous avons fait les observations suivantes :

- la meilleure solution apparaît toujours après avoir retiré $b/2$ connexions ;
- on peut supposer un *effet de cloche* dans la zone de la meilleure solution (montée graduelle puis descente graduelle).

La première observation nous permet de diviser en deux l'espace de recherche, ce qui permet de faire des gains considérables en temps. La deuxième observation nous donne l'idée d'utiliser un pas plus grand que 1. Dans l'exemple de la Figure 3.5, on désactive d'abord $b/2$ connexions. Ensuite, on évalue nos solutions après un certain intervalle. Le pas que nous choisissons ici dépend du résultat de la solution initiale, donc de b . Bien entendu, le pas peut être modifié en fonction de la taille du problème et du nombre de connexions bloquées à la première affectation. Voici les différents pas que nous avons choisis :

$$\begin{aligned} b \leq 100 &\Rightarrow pas = 1; \\ b \leq 800 &\Rightarrow pas = 10; \\ b \leq 1500 &\Rightarrow pas = 50; \\ b \geq 1500 &\Rightarrow pas = 100. \end{aligned} \tag{3.14}$$

On va alors identifier le nombre de connexions retirées n qui nous donne la meilleure solution après cette itération. Dans l'exemple de la figure 3.6, c'est la valeur identifiée par la flèche. L'étape suivante consiste à définir un espace centré autour de n où on effectuera la recherche avec un pas de 1, si le pas initial diffère de 1. Voici les espaces que nous avons choisis :

$$\begin{aligned} pas = 1 &\Rightarrow espace = 0; \\ pas = 10 &\Rightarrow espace = 40; \\ pas = 50 &\Rightarrow espace = 200; \\ pas = 100 &\Rightarrow espace = 200. \end{aligned} \tag{3.15}$$

La deuxième observation nous suggère aussi d'arrêter la recherche si la perte entre la meilleure solution trouvée et la solution courante devient trop grande. En effet, une

fois la meilleure valeur trouvée, on assiste à une décroissance du gain obtenu pour les itérations suivantes. Dans notre cas, nous avons décidé d'arrêter la recherche si la perte dépasse 50 connexions, mais ce seuil peut être choisi différemment.

Bien évidemment, ce processus peut mener à des solutions moins bonnes que celles obtenues en utilisant un pas de 1 tout au long de la recherche de solutions. Le pas que nous choisissons ici dépend du résultat de la solution initiale, donc de b . Bien entendu, le pas peut être modifié en fonction de la taille du problème et du nombre de connexions.

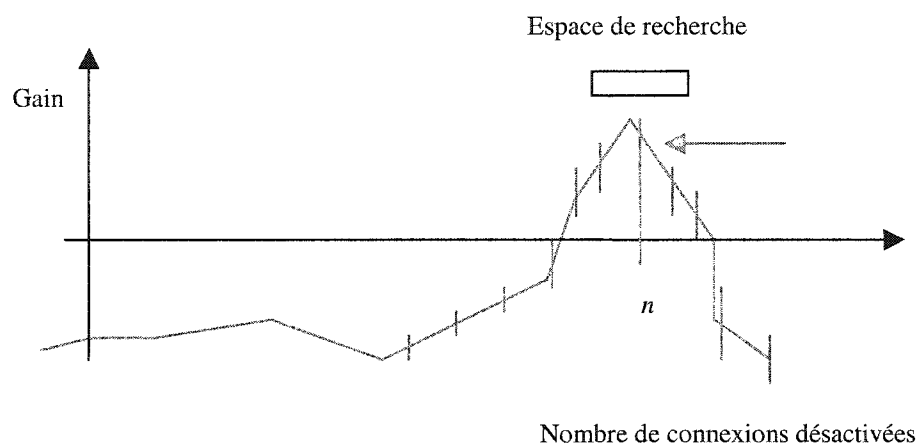


Figure 3.6 Exemple pour la première itération

Supposons qu'une première affectation nous donne un résultat $b = 5000$. La première étape nécessitera $2500/100 = 25$ itérations. Si on définit un espace de recherche de 200 points autour de la valeur de n trouvée, il faudra au total 225 itérations comparées à 5000 itérations avec un pas de 1. Pour certains réseaux, nous sommes passés de 8h à 40 minutes de temps d'exécution.

Une fois que nous avons identifié la solution voulue, il est encore possible d'améliorer légèrement cette solution. En effet, on peut essayer d'affecter séquentiellement tous les trafics qui ont été désactivés au départ en considérant les longueurs d'ondes déjà utilisées sur les différents liens. Cette opération devrait

améliorer la solution obtenue tout en ne dégradant pas le temps de calcul de manière problématique.

Dans le cas de l'algorithme H, cette méthode s'applique au scénario sans panne. Nous appelons cette modification de l'algorithme H, l'algorithme H+. Dans la majorité des cas, la méthode devrait avoir comme répercussion une diminution du nombre total de connexions bloquées sur tous les scénarios. On décide de ne pas l'appliquer aux autres scénarios car ceux-ci dépendent du scénario sans panne. En effet, comme nous ne reconfigurons pas totalement le réseau dans les scénarios de pannes, l'amélioration sur les scénarios de pannes sera minime voire inexistante.

L'algorithme H+ peut être vu comme une méthode qui serait appliqué à n'importe quel algorithme d'affectation. La figure 3.6 illustre ce fait.

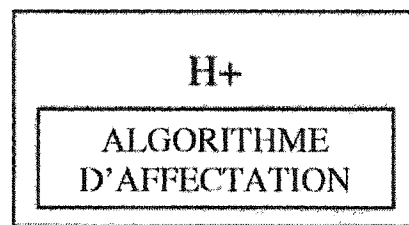


Figure 3.7 H+ vu comme une méthode basée sur une heuristique

Algorithme amélioré pour le scénario sans panne

Étape 1 : Calculer b en effectuant une affectation initiale avec l'algorithme H.

Étape 2 : Désactiver $b/2$ connexions en désactivant à chaque fois la connexion qui a la plus grande valeur de n_k .

Étape 3 : Première itération

Étape 3.1 :

Répéter

 Désactiver un nombre de connexions égal au pas et évaluer la solution avec l'algorithme AFF. (voir paragraphe 3.2.3)

Jusqu'à ce que b connexions soient désactivées.

Retenir le nombre n de connexions supprimées au départ qui donne la meilleure solution.

Étape 3.2 :

Définir un espace de recherche centré autour de n .

Étape 4 : Deuxième itération

Pour tout l'espace de recherche défini à l'Étape 3.2,

Désactiver une connexion à la fois et évaluer la solution avec l'algorithme AFF.

Retenir la meilleure solution.

Étape 5 : Essai pour les trafics désactivés au départ

Réactiver tous les trafics supprimés au départ

Évaluer la solution avec l'algorithme AFF.

3.2.3 Algorithme d'affectation de longueur d'onde

L'algorithme que nous avons utilisé est dénoté AFF. Nous le présentons ici:

Étape 1 : Affecter les trafics sur les liens

Pour chaque trafic $k \in K$,

Si le plus court chemin de k passe par le lien $(i,j) \in M$,

Affecter k sur (i,j) .

Étape 2 : Établir les adjacences entre chemins

Pour chaque lien $(i,j) \in M$,

Pour chaque trafic $k \in K$ passant par le lien (i,j)

Pour chaque trafic $p \in K$ passant par le lien (i,j)

Affecter p à k dans le tableau des adjacences.

Affecter k à p dans le tableau des adjacences.

Étape 3 : Affecter les longueurs d'onde

Étape 3.1 : Premier essai

Pour chaque trafic $k \in K$,

Étape 3.1.1 :

Pour chaque longueur d'onde $\lambda \in \Omega$,

Si aucun des trafics adjacents à k n'utilise λ ,

Affecter λ à k .

Passer au trafic suivant.

Étape 3.2 : Deuxième essai pour les trafics qui ont plusieurs plus courts chemins.

Pour chaque trafic $k \in K$ n'ayant pas été affecté à l'Étape 3.1,

Étape 3.2.1 :

Si k possède un autre plus court chemin n'ayant pas été testé,

Modifier les adjacences en fonction de ce chemin.

Répéter l'étape 3.1.1.

Comme nous l'avons dit, cet algorithme transforme le problème d'affectation en un problème de coloration de graphe associé. L'étape 1 réalise l'affectation des trafics sur les liens en fonctions des chemins qui ont été prédéterminés. Cette étape est essentielle car c'est à partir de là qu'on va pouvoir déterminer si un trafic k partage un ou plusieurs liens avec le trafic p . Toutefois, dans le cas de notre algorithme principal, cette opération est déjà réalisée lors du calcul des plus courts chemins et de l'affectation des trafics aux liens. Il ne sera donc pas nécessaire de la répéter lors de l'affectation des longueurs d'ondes. L'étape 2 de l'algorithme d'affectation consiste à établir les adjacences entre les trafics. Rappelons que, dans notre graphe à colorier, deux nœuds, qui représentent des trafics, ne seront adjacents que si les trafics correspondant partagent un ou plusieurs nœuds dans le réseau réel. Il faut aussi faire attention au cas où deux trafics partagent plus d'un lien afin de ne pas créer d'adjacences redondantes. L'étape 3 consiste en l'affectation proprement dite. Une fois le graphe de coloriage construit après l'étape 2, on va examiner les nœuds l'un après l'autre selon l'ordre arbitraire dans lequel seront les trafics correspondants. Pour chaque trafic, on va essayer séquentiellement les longueurs d'ondes disponibles. Si aucun trafic adjacent n'utilise la longueur d'onde λ , elle est affectée au trafic étudié. À l'étape 3.2, on fait un deuxième

essai pour les trafics bloqués ayant un autre plus court chemin. Si c'est le cas, il faut modifier les adjacences avant de réessayer l'affectation.

Maintenant que nous avons présenté les algorithmes étudiés, nous devons en évaluer la complexité.

3.2.4 Complexité des algorithmes

Pour évaluer la complexité des algorithmes, nous allons procéder étape par étape. Nous allons commencer par l'algorithme d'affectation car sa complexité participe à celle de l'algorithme principal. Dans cette section, nous ne considérerons que le pire cas. On note $n = |N|$, $m = |M|/2$, $w = |\Omega|$, $t = |T|$, $k = |K|$.

Dans l'étape 1 de l'algorithme d'affectation, chaque trafic étudié passe au maximum par $m - 1$ liens. La complexité de cette étape est donc $O(k(m-1)) = O(km)$. De la même manière, l'étape 2 de AFF est de complexité $O(k^2m)$ car chaque lien peut avoir au maximum k trafics qui le traversent. Pour l'étape 3.1.1, chaque trafic peut avoir un maximum de $k-1$ trafics adjacents. L'étape 3.1.1 est donc de complexité $O(w(k-1)) = O(kw)$. Par conséquent, on peut dire que l'étape 3.1 s'effectue en un temps $O(wk^2)$. Passons à l'étape 3.2. En analysant d'abord l'étape 3.2.1, on peut dire qu'on considère au maximum deux plus courts chemins pour chaque trafic. En considérant l'étape 3.1.1 qui est répétée, la complexité de l'étape 3.2 est donc $O(k^2w)$. En additionnant les étapes 3.1 et 3.2, on obtient pour la complexité de l'étape 3 :

$$O(wk^2) + O(k^2w) = O(wk^2)$$

En faisant la somme totale des trois étapes de cet algorithme, on obtient :

$$O(km) + O(k^2m) + O(wk^2) = O(k^2(m+w))$$

Même si, dans le cas de notre algorithme principal, nous n'aurons pas besoin de refaire l'étape 1 de AFF, la complexité de l'algorithme d'affectation reste la même lorsque utilisé dans l'algorithme H.

Nous allons maintenant traiter l'algorithme principal H. L'algorithme de Dijkstra que nous utilisons à l'étape 1.1 a été démontré de complexité $O(m \log_2 n)$. Cet algorithme, connu aussi sous le nom de *Dijkstra-Monceau* car il utilise un monceau dans son processus, permet de trouver les plus courts chemins entre un nœud origine quelconque et tous les autres nœuds du réseau. C'est pour cela que l'étape 1.1 de H considère n itérations au lieu de k . Le temps mis pour effectuer cette étape est donc $O(nm \log_2 n)$. Dans l'étape suivante, chaque trafic utilise au maximum $m-1$ liens, d'où la complexité $O(k(m-1)) = O(km)$. Pour l'étape 1.3, il faut se référer à la complexité de l'algorithme d'affectation qui a été démontrée précédemment. Comme l'étape 1.4.1 est de complexité $O(1)$, on conclut que l'étape 1.4 est $O(km)$. On obtient donc pour l'étape 1 de l'algorithme principal :

$$O(nm \log_2 n) + O(km) + O(k^2m + wk^2) + O(km) = O(nm \log_2 n + k^2m + wk^2)$$

Étudions maintenant l'étape 2 de l'algorithme principal. Premièrement, on remarque que l'étape 2.1 est $O(k)$ car l'ensemble K^P contient au maximum k trafics. De la même façon, on peut dire que l'étape 2.2.1 est aussi $O(k)$. En se référant à l'étape 1.1, on peut affirmer que la complexité de l'étape 2.2.2 est $O(nm \log_2 n)$ car l'algorithme de *Dijkstra-monceau* sera appliqué au maximum n fois. La dernière étape importante est l'étape 2.2.3 qui consiste à affecter les trafics ayant été reroutés. Le maximum de trafic rerouté étant k , cette étape est donc de complexité égale à celle de l'étape 1.3, c'est-à-dire $O(k^2m + wk^2)$. En additionnant tout cela, on obtient provisoirement pour l'étape 2 :

$$O(k) + O(k) + O(nm \log_2 n) + O(k^2m + wk^2) = O(nm \log_2 n + k^2m + wk^2)$$

Toutefois, il ne faut pas oublier que l'étape 2 se répète pour chaque scénario de panne. Comme il y a t scénarios, il y a donc $t-1$ scénarios de pannes. Or, chaque scénario de panne est caractérisé par le bris d'une des liaisons du réseau. On peut donc dire qu'il existe m scénarios de pannes. La complexité de l'étape 2 devient donc :

$$mO(nm \log_2 n + km + wk^2) = O(nm^2 \log_2 n + k^2m^2 + wmk^2)$$

Enfin, l'étape 3 est $O(1)$. On peut donc évaluer la complexité totale de l'algorithme H comme suit :

$$\begin{aligned} O(nm \log_2 n + k^2m + wk^2) + O(nm^2 \log_2 n + k^2m^2 + wmk^2) + O(1) &= (m+1)O(nm \log_2 n + k^2m + wk^2) \\ &= O(nm^2 \log_2 n + k^2m^2 + wmk^2) \end{aligned}$$

Nous allons maintenant traiter de la version améliorée de H qu'on dénotera H+. H+ utilise l'algorithme amélioré pour le scénario sans panne et garde la même procédure que H pour les scénarios de pannes. L'étape 1 de l'algorithme amélioré est de même complexité que H soit $O(nm^2 \log_2 n + k^2m^2 + wmk^2)$. Comme il peut y avoir k demandes bloquées au maximum, l'étape 2 est $O(k/2) = O(k)$. A l'étape 3.1, on effectue k/pas itérations de l'algorithme d'affectation AFF. La complexité est donc $O(k^3m + wk^3)$. L'étape 3.2 est $O(1)$, ce qui revient à dire que l'étape 3 est de complexité :

$$O(k^3m + wk^3) + O(1) = O(k^3m + wk^3).$$

Pour l'étape 4, on applique AFF un nombre de fois équivalent à la taille de l'espace de recherche. Comme la taille de cet espace ne peut dépasser le nombre de connexions à affecter, cette étape est $O(k^3m + wk^3)$. Enfin, l'étape 5 est de même complexité que l'algorithme d'affectation, sans considérer l'établissement des adjacences entre chemins. On peut donc évaluer la complexité de l'algorithme amélioré pour le scénario sans panne à :

$$O(nm^2 \log_2 n + k^2 m^2 + wmk^2) + O(k) + 2 O(k^3 m + wk^3) + O(wk^2) = \\ O(nm^2 \log_2 n + k^2 m^2 + wmk^2 + k^3 m + wk^3)$$

Fort de cela, on évalue la complexité de $H+$ qui est :

$$O(nm^2 \log_2 n + km^2 + wmk^2 + k^2 m + wk^3) + O(nm^2 \log_2 n + km^2 + wmk^2) + O(1) = \\ O(nm^2 \log_2 n + k^2 m^2 + wmk^2 + k^3 m + wk^3)$$

CHAPITRE IV

IMPLÉMENTATION ET ANALYSE DES RÉSULTATS

Une fois le modèle défini, les algorithmes présentés et analysés, nous devons maintenant passer à l'étape d'implémentation qui nous permettra d'évaluer la qualité des solutions. Dans ce chapitre, nous commencerons par présenter le langage informatique utilisé ainsi que la configuration du code que nous avons implanté. Après cela, nous traiterons des données utilisées pour réaliser nos tests, à savoir les réseaux et les ensembles de demandes utilisés. Nous essaierons d'utiliser des réseaux de différentes tailles afin d'évaluer le comportement des algorithmes en fonction de la taille et de la charge du réseau. Ensuite, nous aborderons l'étape de recherche de solution optimale avec le logiciel d'optimisation CPLEX. À défaut de la solution optimale, nous essaierons d'obtenir des bornes inférieures pour le taux de blocage. Cela devra nous permettre d'évaluer la distance de nos solutions par rapport à l'optimum. Enfin, nous présenterons et analyserons les résultats de nos tests.

4.1 Détails d'implémentation

La programmation de nos algorithmes a été réalisée à partir du langage C. Le logiciel de programmation que nous avons utilisé est Microsoft Visual C++ 6.0 installé sur une machine avec un processeur à 1.7 MHz, un disque dur de 40 GB et une mémoire vive de 512 Mo.

Le code programmé se répartit sur trois fichiers : *dijmonceau.cpp*, *coloriage.cpp* et *central.cpp*. Le fichier *dijmonceau.cpp* contient les routines nécessaires pour effectuer le calcul des plus courts chemins à partir de l'algorithme de Dijkstra et en utilisant un monceau pour accélérer le processus. Rappelons que ce processus est de complexité

$O(m \log_2 n)$ et permet de trouver les plus courts chemins d'une origine donnée à toutes les destinations possibles dans le réseau. Le fichier *coloriage.cpp* contient les fonctions permettant de réaliser l'affectation des longueurs d'onde aux trafics. On y retrouve aussi les fonctions préliminaires à cette affectation, à savoir la fonction qui assigne les trafics aux liens et celle qui établit les adjacences entre trafics afin de bâtir le graphe de coloriage. Comme son nom l'indique, le fichier *central.cpp* représente le fichier principal. C'est le fichier qui gère toutes les procédures. Typiquement, il fonctionne de la manière suivante :

- La fonction *chargerTraffic* se charge d'aller lire les informations sur le trafic dans un fichier. Voici la nomenclature générale du fichier des demandes :

Nombre de demandes

Origine Destination Quantité Permanence

Origine Destination Quantité Permanence

Quantité représente le nombre de demandes qu'il y a sur une paire origine-destination et *Permanence* indique si le trafic est robuste ou non. Ces informations sont chargées dans la variable *traffic* qui est un tableau contenant toutes les demandes du réseau. Chaque élément de la variable *traffic* contient des informations comme l'origine, la destination, la permanence, les chemins possibles, les chemins adjacents et des informations pour savoir si le trafic est actif ou non. Pour des raisons liées à l'espace mémoire, la variable *traffic* garde en mémoire les informations sur deux scénarios : le premier scénario est celui sans panne qu'on garde comme référence et le deuxième est le scénario courant. Les valeurs correspondant au deuxième scénario sont réinitialisés à chaque nouveau scénario.

- La fonction *InitDijkstra* a pour rôle d'initialiser les variables qui seront utilisées dans le calcul des plus courts chemins. Elle se charge d'aller lire les informations sur la topologie du réseau dans un fichier. Voici la nomenclature générale du fichier de la topologie :

Nombre de nœuds *Nombre de liens*

Origine *Destination* *Coût du lien*

Origine *Destination* *Coût du lien*

Ces informations sont chargées dans la variable *Liaison* qui est un tableau contenant tous les liens unidirectionnels du réseau. Chaque élément de la variable *Liaison* contient des informations comme l'origine, la destination, le coût, l'état du lien (panne ou non) et les trafics dont les plus courts chemins empruntent le lien considéré. Pour les mêmes raisons que la variable *traffic*, *Liaison* garde en mémoire les informations sur deux scénarios.

- La fonction *principale* représente le traitement principal. Lorsqu'elle est appelée, elle étudie les scénarios séquentiellement. À chaque scénario de pannes correspond un lien endommagé. Pour un scénario donné, elle commande d'abord, si nécessaire, la détermination des plus courts chemins à partir des fonctions contenues dans le fichier *dijmonceau.cpp*. Ensuite, elle appelle la fonction chargée d'affecter les longueurs d'ondes et de répertorier les résultats dans un fichier. Cette dernière fait appel aux routines contenues dans le fichier *coloriage.cpp*. Le fichier de résultats, quant à lui, contient les informations comme le nombre de nœuds du réseau, le nombre de demandes de connexions, le nombre total de connexions bloquées sur l'ensemble des scénarios qui correspond à la fonction objectif, le nombre moyen de connexions bloquées et le temps d'exécution du programme. Avant d'appeler la fonction principale, on décide d'appliquer l'algorithme H où l'algorithme H+.

Après avoir parlé de notre implémentation, nous allons maintenant parler du jeu de données que nous avons utilisé pour nos tests, à savoir les topologies de réseau et les demandes de trafics.

4.2 Jeux de données pour les tests

Les simulations que nous allons effectuer sont essentiellement constituées de trois composantes : la topologie physique du réseau, la charge du réseau et le nombre de longueurs d'onde utilisées. Pour vérifier la robustesse de nos algorithmes, nous avons générés plusieurs ensembles de données pseudo-aléatoirement.

4.2.1 Topologie des réseaux

Pour créer nos réseaux, nous utilisons un programme qui requiert comme variables d'entrées le nombre de nœuds n , le nombre de liens L qui doit être supérieur au nombre de nœuds, les bornes inférieures et supérieures du degré d'adjacence des nœuds du réseau. Nous avons imposé que chaque nœud soit relié à au moins deux autres nœuds du réseau, condition nécessaire mais non suffisante pour assurer la 2-connexité du réseau. En effet, on souhaite que le réseau puisse survivre à une panne simple. La borne supérieure du degré d'adjacence a été fixée à trois afin d'assurer une meilleure répartition des liens dans le réseau.

Pour assurer la 2-connexité du réseau, on décide de construire au départ un cycle hamiltonien, c'est-à-dire une boucle qui passe une seule fois par tous les nœuds du réseau. Cette boucle est construite en reliant chaque nœud à son suivant en suivant une numérotation séquentielle de 0 à $n-1$. La prochaine étape consiste à relier le dernier nœud au premier pour fermer la boucle. À cette étape, le réseau contient déjà n liens et chaque nœud est relié à au moins deux autres nœuds. Les $L-n$ liens restant sont créés aléatoirement en respectant les contraintes sur le degré d'adjacence. Les paires de nœuds choisies sont équiprobables. La Figure 4.1 représente le réseau généré de 20 nœuds avec 25 liens.

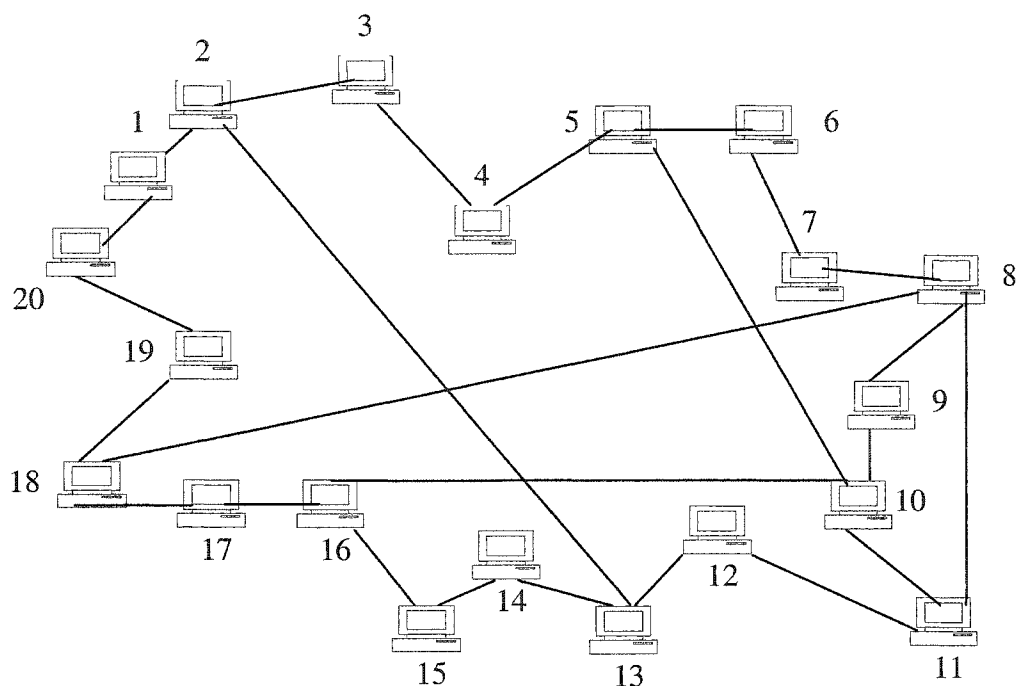


Figure 4.1 Réseau de 20 nœuds

Il faut aussi mentionner que nous utilisons deux versions pour chaque réseau. Dans la première version, les coûts des liens sont générés aléatoirement entre 0 et 10 tandis que la seconde version ne contient que des liens de coût unitaire. Dans cette version, le coût d'un chemin représente le nombre de sauts et donc de liens physiques sur ce chemin. Le tableau 4.1 présente les six réseaux de tests utilisés.

Tableau 4.1 Réseaux utilisés pour les tests

Nombre de Nœuds	Nombre de Liens
5	7
20	25
40	47
60	70
80	100
100	130

4.2.2 Demandes de trafic

Nous avons dû simuler la charge statique de nos réseaux. Pour le faire, nous avons exploré toutes les paires origine-destination possibles dans un réseau. Pour chaque paire, il y a la possibilité d'avoir au maximum un trafic permanent et un trafic robuste. C'est le lieu de rappeler qu'il y a une agrégation du trafic qui se fait en amont du réseau. Les probabilités pP d'avoir une demande permanente et pR d'avoir une demande robuste sont fixées au départ et déterminent la charge du réseau. Pour un réseau de 100 nœuds avec une probabilité $p = pP = pR$ de 20% par exemple, on a $100 \times 100 = 10\,000$ paires origine-destination possibles. On obtiendra donc environ 2000 demandes permanentes et autant de demandes robustes. Le taux de demandes par rapport au nombre de paires origine-destination est donc $pP + pR = 2p$. Le Tableau 4.2 résume les charges appliquées aux différents réseaux.

Tableau 4.2 Charge en nombre de demandes

Nombre de Nœuds	$p = 0.02$	$p = 0.05$	$p = 0.10$	$p = 0.20$
5	1	1	3	11
20	15	40	79	152
40	66	150	307	608
60	163	339	709	1401
80	279	629	1264	2539
100	421	977	1969	3948

4.2.3 Nombre de longueurs d'onde

Pour voir l'effet de la variation du nombre de longueurs d'onde, nous avons utilisé sept valeurs différentes de nombres de longueurs d'onde disponibles qui sont : 1, 2, 4, 8, 16, 32 et 64. Théoriquement, l'accroissement du nombre de longueurs d'onde devrait entraîner une diminution du nombre de trafics bloqués pour un réseau et une demande donnés.

4.3 Analyse des résultats des algorithmes

Après avoir présenté notre modèle, nous devons maintenant effectuer les simulations. Premièrement, on constate qu'on a 6 réseaux, 4 types de charges et 7 possibilités pour le nombre de longueurs d'onde. Il y a donc 168 combinaisons possibles. Pour des commodités de présentation, nous ne présenterons que les résultats obtenus pour les réseaux de 20 et 80 nœuds, ce qui nous ramènera à 54 combinaisons.

4.3.1 Algorithme H

Dans cette section, nous étudions les performances de notre algorithme de base H. Les Figures 4.2 à 4.9 présentent les résultats pour les coûts de liens aléatoires (courbe mauve) et ceux pour les coûts de liens unitaires (courbe cyan).

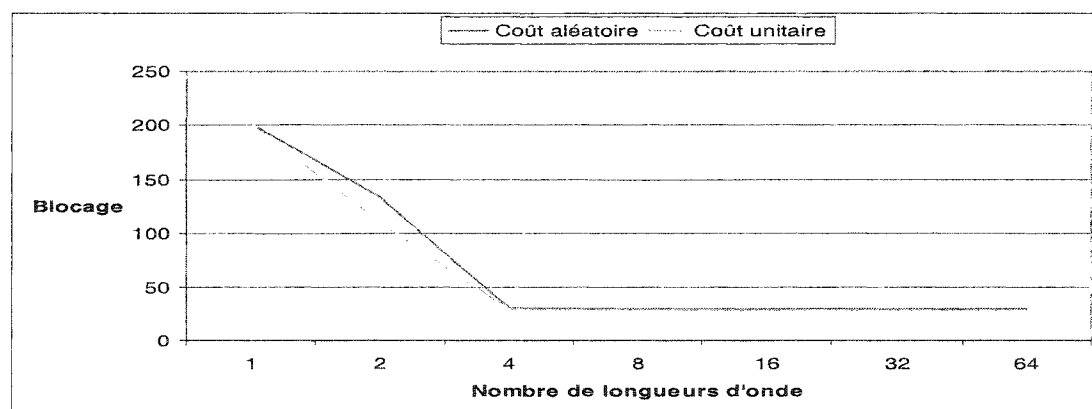


Figure 4.2 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.02$)

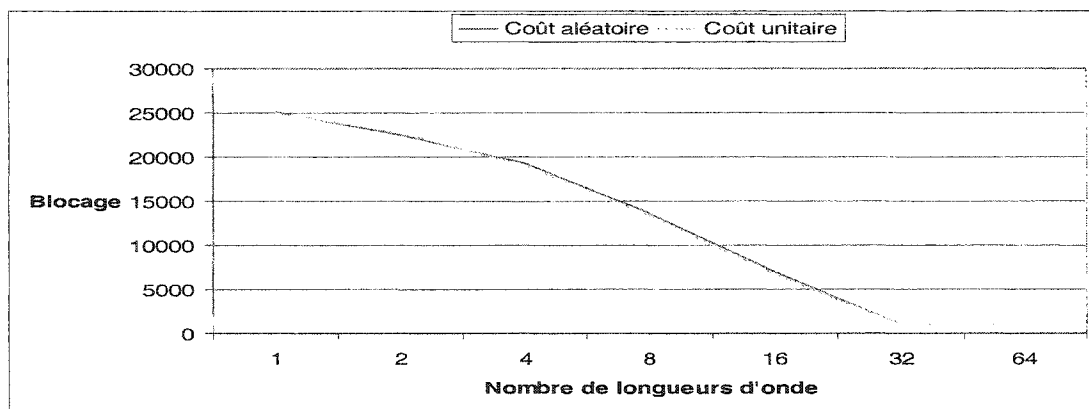


Figure 4.3 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.02$)

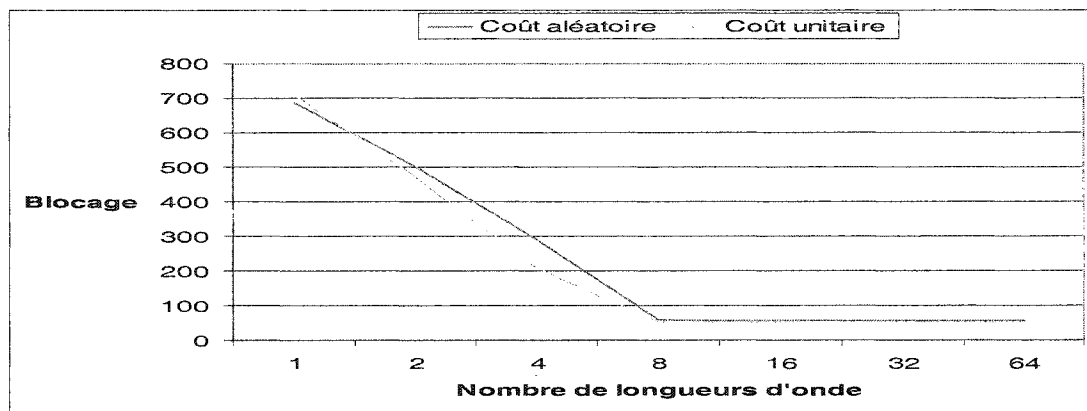


Figure 4.4 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.05$)

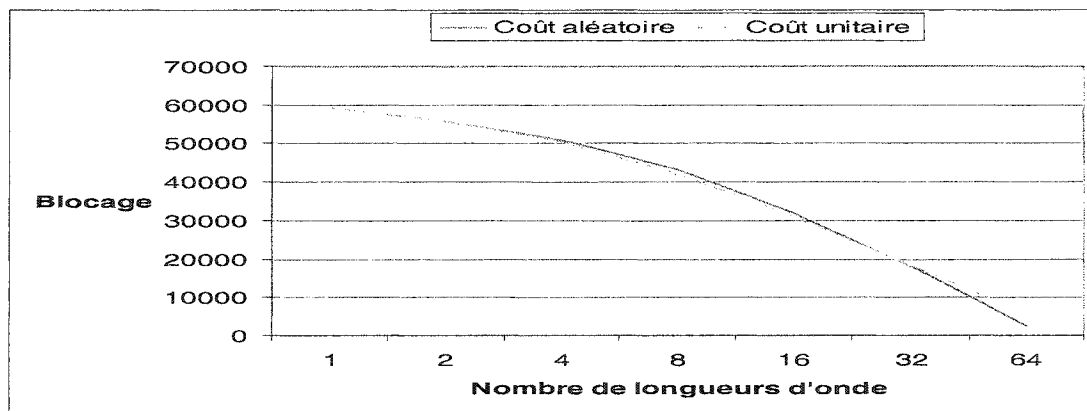


Figure 4.5 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.05$)

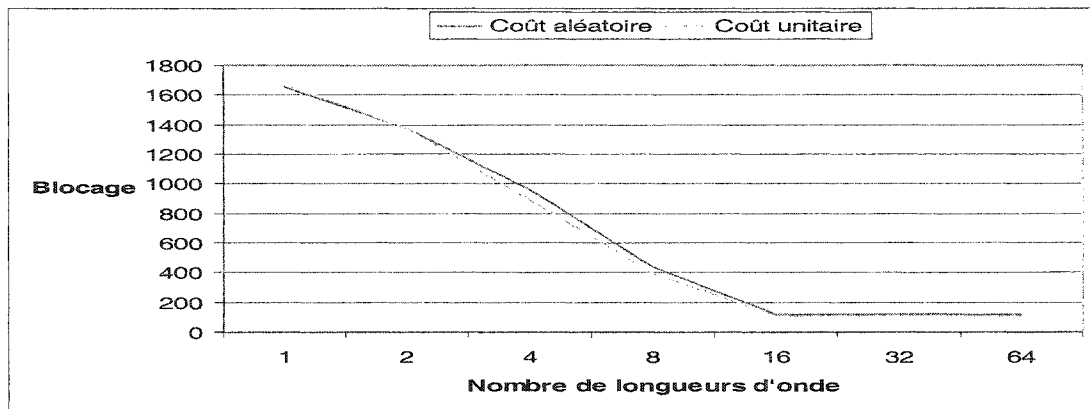


Figure 4.6 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.1$)

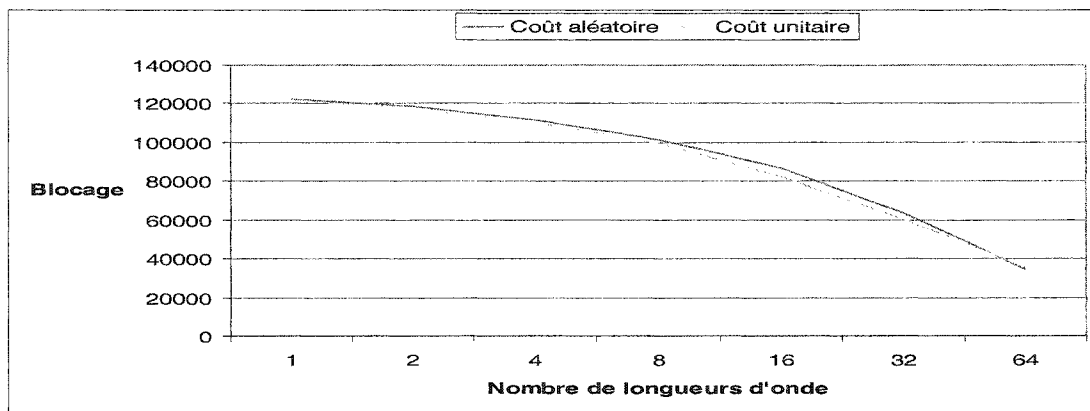


Figure 4.7 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.1$)

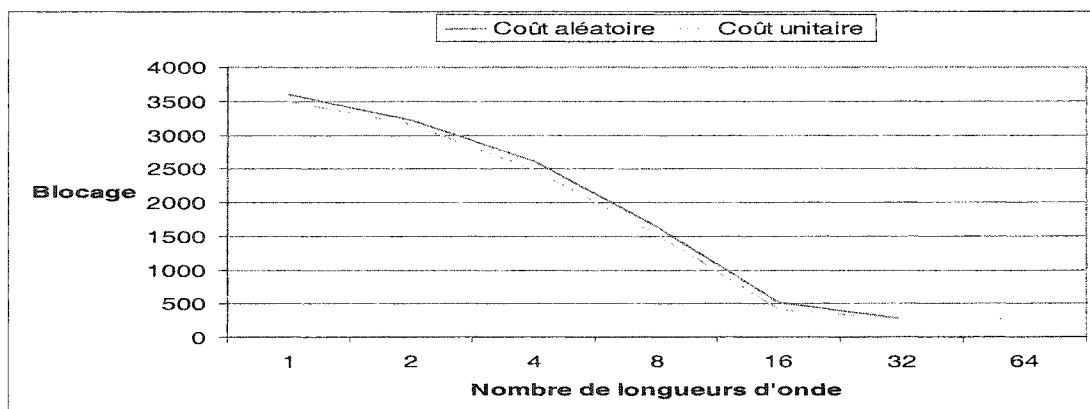


Figure 4.8 Nombre total de demandes bloquées ($n = 20$ nœuds, $p = 0.2$)

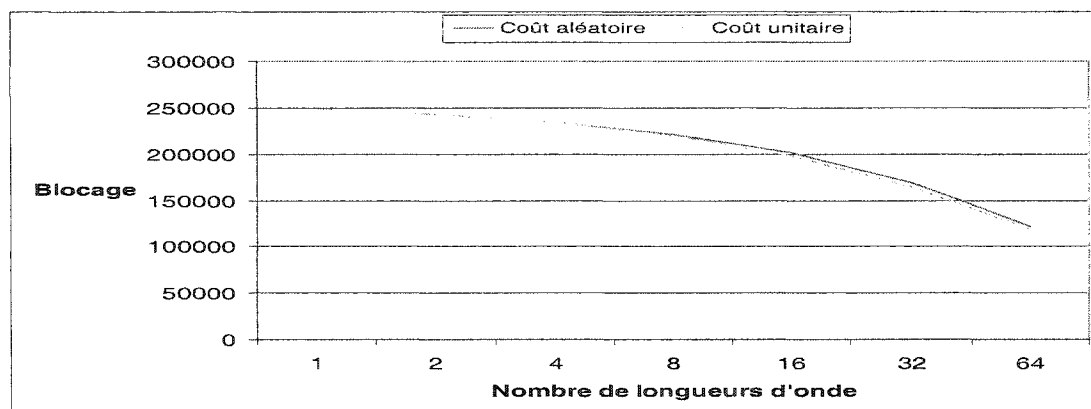


Figure 4.9 Nombre total de demandes bloquées ($n = 80$ nœuds, $p = 0.2$)

Nous pouvons remarquer que le taux de blocage diminue avec l'augmentation du nombre de longueurs d'onde, ce qui est tout à fait normal. Plus on a de capacités sur nos liens, plus on pourra acheminer de trafics. On observe aussi que, pour un réseau et un nombre de longueurs d'onde fixés, l'augmentation de la charge entraîne une augmentation du nombre total de connexions bloquées.

Dans la majorité des tests effectués, pour une charge et un nombre de longueurs d'onde donnés, le résultat pour les réseaux aux coûts de liens unitaires est inférieur à celui obtenu en utilisant des coûts aléatoires. Cela s'explique aisément par le fait que nous utilisons une politique de plus court chemin. En effet, le coût d'un chemin dans un réseau de coûts unitaires représente la distance en nombre de sauts optiques, autrement dit le nombre de liens traversés. Or, l'objectif de minimisation du nombre total de trafics bloqués sur tous les scénarios est intimement lié à la distance en nombre de sauts de chaque chemin. Plus un chemin traverse de liens, plus le risque d'augmenter le nombre de blocage est élevé car chaque scénario de panne est caractérisé par le bris d'un lien. Dans le cas d'un réseau de coûts unitaires, cette distance, qui représente le coût d'un chemin, est minimisée, d'où les meilleurs résultats obtenus avec ces réseaux. Nous allons donc utiliser les réseaux avec les coûts de liens unitaires pour continuer nos expérimentations.

4.3.2 Algorithme H+

Cette section traite des résultats de l'algorithme H+, censé améliorer ceux de l'algorithme H. Nous avons travaillé sur des réseaux de 20, 40 et 80 nœuds avec des coûts de liens unitaires. Les tests ont été menés avec 4 types de charge caractérisés par $p = 0.02$, $p = 0.05$, $p = 0.10$ et $p = 0.20$. Les nombres de longueurs d'onde utilisés sont 1, 2, 4, 8, 16, 32 et 64. Les figures 4.10 à 4.12 illustrent ces résultats et visent à faire ressortir les pourcentages d'amélioration de la solution apportés par l'algorithme H+ par rapport à l'algorithme H.

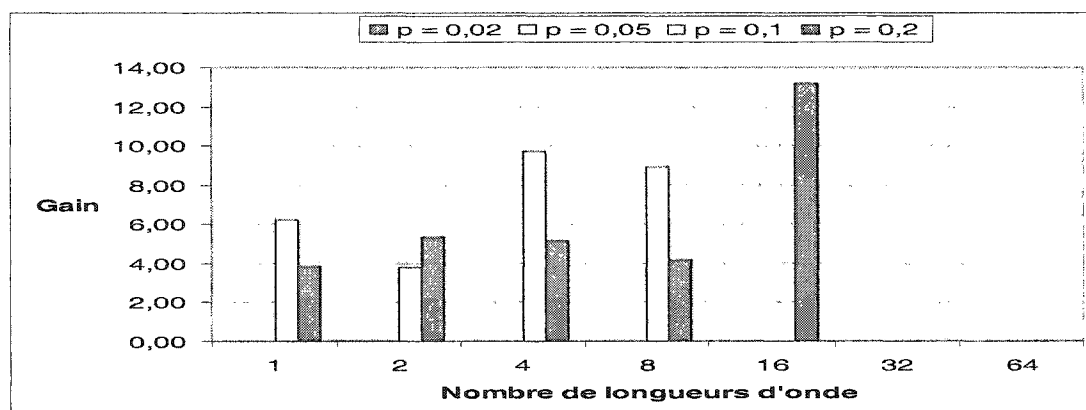


Figure 4.10 Gain en pourcentage de demandes bloquées (n = 20 nœuds)

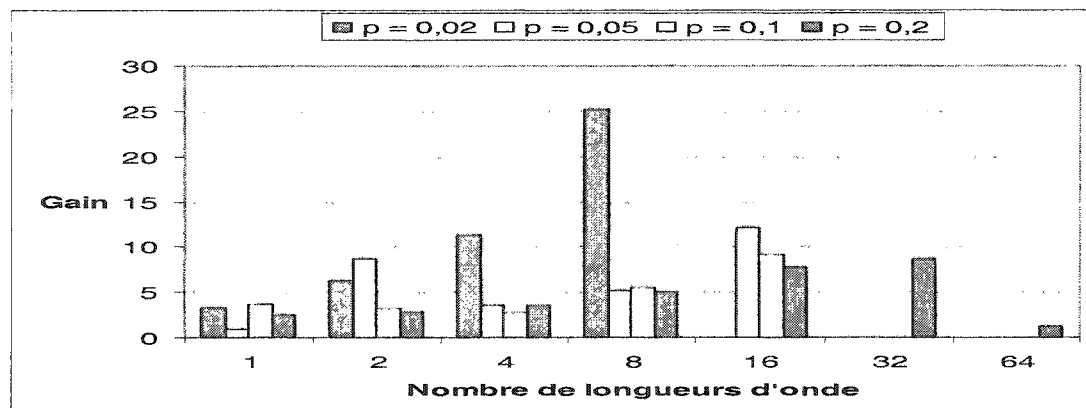


Figure 4.11 Gain en pourcentage de demandes bloquées (n = 40 nœuds)

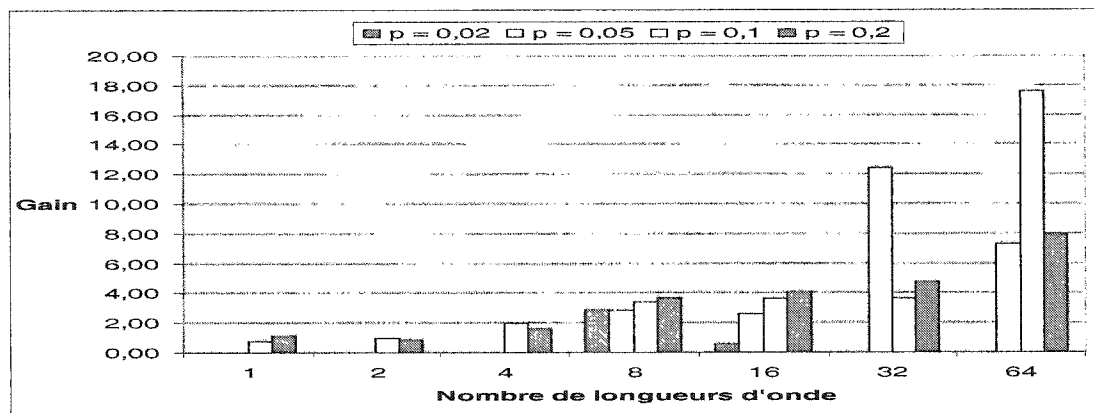


Figure 4.12 Gain en pourcentage de demandes bloquées (n = 80 nœuds)

La première chose à remarquer est que, pour un réseau et une charge donnés, aucune amélioration n'est possible si on dispose de suffisamment de longueurs d'onde pour satisfaire la demande. Dans ce cas, l'algorithme H fournit de bons résultats. À l'opposé, si le nombre de longueurs d'onde disponible est faible, les gains deviennent quasiment inexistantes à cause de la saturation du réseau.

Nos graphiques nous indiquent une tendance à la croissance puis à la décroissance en fonction de l'augmentation du nombre de longueurs d'onde pour un réseau et une charge donnée. Au départ, l'amélioration est faible car on a très peu de canaux disponibles. Plus le nombre de longueurs d'onde augmente, plus il est possible de réorganiser les trafics de manière à diminuer le nombre total de connexions bloquées. C'est pourquoi cet accroissement du gain en pourcentage s'effectue jusqu'à un certain seuil. À partir de ce seuil, on peut supposer que les résultats de l'algorithme H se rapprochent de plus en plus de l'optimum. En effet, il apparaît évident que, pour un très grand nombre de longueurs d'onde disponibles, le résultat du blocage se rapprochera de plus en plus de 0, qui représentera aussi dans ce cas l'optimum de notre fonction objectif. Il devient donc très difficile d'améliorer les résultats d'une heuristique dont l'écart entre la solution obtenue et l'optimum s'amenuise.

Ces observations sont confortées par la Figure 4.12. Pour $p = 0.02$, on remarque que la décroissance est déjà amorcée pour $\lambda = 8$ et $\lambda = 16$ parce que la charge du réseau est faible. Pour $p = 0.05$, on note un seuil atteint dans la zone de $\lambda = 32$ avant une

décroissance du gain à partir de $\lambda = 64$. Enfin, pour $p = 0.10$ et $p = 0.20$, le gain croît de $\lambda = 1$ à $\lambda = 64$. On peut présager que le seuil sera atteint pour un plus grand nombre de longueurs d'onde.

Pour conclure cette section, on peut dire que l'algorithme H+ permet, dans une certaine mesure, d'améliorer les résultats de l'algorithme H. Nous allons maintenant discuter de la possibilité d'éprouver les résultats de H+ en utilisant une méta-heuristique.

4.3.3 Utilisation d'une méta-heuristique : la recherche taboue

La recherche taboue (RT) est une technique adaptative d'optimisation combinatoire. Elle peut s'appliquer à différents types de problèmes et conduire à de bonnes solutions, d'où son qualificatif de méta-heuristique. Cette méthode a été introduite sous sa forme actuelle par Glover (1989) et s'étend aujourd'hui à plusieurs domaines comme la coloration de graphe, le design de circuit intégré ou le problème du commis voyageur. Le principal avantage de la RT est qu'elle est capable de fournir des solutions d'assez bonne qualité tout en nécessitant un effort de calcul nettement moins important que les méthodes exactes.

La plupart des méthodes et procédures proposées pour trouver des solutions à des problèmes d'optimisation combinatoire partent d'une solution initiale et lui appliquent de petites perturbations pour aboutir à de meilleures solutions. Ces perturbations qui permettent de se déplacer d'une solution à une autre définissent des *mouvements*. Ces mouvements sont inversibles en appliquant la perturbation inverse. On définit le *voisinage* d'une solution comme l'ensemble des solutions qu'on peut obtenir après avoir appliqué un mouvement sur la solution courante. RT s'inspire du principe de l'algorithme général de descente simple. Celui-ci part d'une solution initiale qu'il essaie d'améliorer de façon itérative en générant un sous-ensemble du voisinage de la solution courante et en choisissant la solution qui minimise l'objectif comme solution courante si elle est moins coûteuse que la meilleure solution obtenue depuis le début. Les itérations se poursuivent jusqu'à ce qu'il ne soit plus possible d'améliorer la solution

courante. Le grand désavantage de cette méthode, c'est que l'optimum final est souvent un minimum local, contrairement au minimum global recherché. RT explore intelligemment l'espace de recherche tout en essayant d'éviter le piège des minima locaux. Pour ce faire, RT accepte quelques fois des solutions qui dégradent la fonction objectif en espérant retomber sur de meilleures solutions. Toutefois, on s'expose ainsi à un risque de cycle. Pour éviter cela, RT définit ce qu'on appelle une *liste taboue* regroupant un certain nombre de solutions déjà visitées. Un mouvement ne peut se faire dans une direction dont la solution appartient à la liste taboue. La liste contient les h dernières itérations, d'où son appellation de *mémoire à court terme*. L'algorithme s'arrête si aucune amélioration n'est intervenue depuis un nombre $hmax$ d'itérations, si toutes les solutions du voisinage sont taboues ou si le nombre maximal d'itération est atteint.

Nous avons appliqué une version simplifiée de cette méthode à notre problème d'affectation de longueurs d'onde, plus particulièrement à l'algorithme H+. L'objectif est de voir l'effet du coût des liens sur la fonction objectif. Nous avons défini un ensemble de cinq valeurs possibles pour les coûts des liens : $L = \{1, 3, 5, 7, 10\}$. Chaque mouvement est défini par la variation, dans l'ensemble L , du coût d'un lien choisi aléatoirement. Après chaque mouvement, nous appliquons l'algorithme H+ pour évaluer la solution du voisinage. En fait, nous évaluons environ 20% du voisinage de la solution courante. La meilleure solution réfèrera donc à une métrique particulière pour un lien donné. On bâtit notre liste taboue en déclarant ce lien tabou pendant un certain nombre d'itérations qui varie pour nous entre 5 et 10. La solution initiale est obtenue en fixant le coût de tous les liens à 1, résultat que nous avons déjà calculé précédemment. Compte tenu des différents temps d'exécution que nous avons recueillis pour l'algorithme H+, nous avons fixé le nombre d'itérations maximal à 100. Nous avons aussi décidé d'interrompre le processus si aucune amélioration de la solution ne se produit après 30 itérations consécutives. Bien évidemment, la probabilité d'obtenir de bonnes solutions augmente avec le nombre d'itérations, mais nous nous contenterons de

ceux-ci car l'aspect de la variation des coûts ne constitue pas la tâche principale de ce mémoire.

Nous allons maintenant discuter des résultats obtenus par l'algorithme de recherche taboue. Pour des raisons de temps, nous avons étudié les réseaux de 5, 20, 40, 60 et 80 nœuds avec des coûts de liens unitaires. Le Tableau 4.3 résume les améliorations en pourcentage apportées par l'algorithme de recherche taboue.

Tableau 4.3 Pourcentage d'amélioration apportée par la recherche taboue

	p = 0.02			p = 0.05			p = 0.10		p = 0.20
	$\lambda=8$	$\lambda=32$	$\lambda=64$	$\lambda=8$	$\lambda=32$	$\lambda=64$	$\lambda=8$	$\lambda=64$	$\lambda=8$
n = 5	0	0	0	0	0	0	0	0	0
n = 20	0	0	0	0	0	0	1.6	0	0
n = 40	3.8	0	0	1.10^{-03}	0	0	0	0	0
n = 60	8.10^{-04}	9.10^{-03}	0	0	1.02	0	0	0	0
n = 80	0	5.10^{-03}	0	0	0	1.2	0	0	0

Comme nous pouvons le remarquer, les améliorations apportées par la recherche taboue dans ce cas, sont minimes, voire insignifiantes. Toutefois, il faut convenir que ces améliorations pourraient être plus importantes si l'on augmentait le nombre d'itérations et si l'on élargissait la plage des valeurs possibles pour les coûts des liaisons. D'autre part, il a déjà été discuté de la qualité des solutions pour les réseaux de coûts unitaires. Or, la solution initiale de notre recherche taboue se base sur ce type de réseaux. Cela pourrait expliquer la difficulté à améliorer la solution initiale. Pour montrer la longueur des temps d'exécution, on peut donner comme exemple les 40 heures dont on a eu besoin pour un réseau de 60 nœuds avec une charge $p = 0.10$.

4.3.4 Temps d'exécution

Un aspect important de tout algorithme repose sur le temps d'exécution. Il est donc intéressant de pouvoir obtenir nos solutions de la manière la plus rapide possible. Le Tableau 4.3 présente les temps de calculs obtenus pour nos charges les plus importantes, à savoir pour $p = 0.2$.

Tableau 4.4 Temps d'exécution en secondes pour $p = 0.2$

	$\lambda = 8$	$\lambda = 16$	$\lambda = 32$	$\lambda = 64$
$n = 5$	<1	<1	<1	<1
$n = 20$	<1	<1	<1	<1
$n = 40$	5	7	9	8
$n = 60$	25	29	49	60
$n = 80$	135	136	153	197
$n = 100$	453	446	480	614
Total	618	618	691	879

Il faut mentionner que le temps n'est pas aussi important dans notre cas parce que nous sommes dans un cas statique et que cette opération est effectuée avant la mise en service du réseau. En observant le Tableau 4.4, on note que la plus longue durée est de 614 secondes, soit 10 minutes et 14 secondes pour un réseau de 100 nœuds utilisant 64 longueurs d'onde. C'est une durée qu'on peut qualifier de tout à fait raisonnable comparativement aux temps de calcul faramineux qu'on obtient en cherchant la solution optimale. Cette recherche de solutions optimales constitue la substance de la section 4.4. Nous allons maintenant traiter de l'effet de la proportion de demandes robustes sur nos résultats.

4.3.5 Effet de la proportion de demandes robustes

Dans ce mémoire, nous avons considéré avoir simultanément sur le réseau, des demandes permanentes et des demandes robustes. Nous avons donc voulu évaluer le comportement de l'algorithme H+ en fonction de la proportion de demandes robustes sur le réseau. Nous avons effectué nos tests sur les réseaux de 20 et 80 nœuds avec toute la plage de longueur d'onde allant de 1 à 64. Nous avons évalué plusieurs types de charge qui sont :

Charge A : $pP = 0$ et $pR = 0.4$;	Charge E : $pP = 0.25$ et $pR = 0.16$;
Charge B : $pP = 0.06$ et $pR = 0.33$;	Charge F : $pP = 0.33$ et $pR = 0.06$;
Charge C : $pP = 0.16$ et $pR = 0.25$;	Charge G : $pP = 0.4$ et $pR = 0$;
Charge D : $pP = 0.2 = pR = 0.2$.	

Comme on peut le remarquer, pour chaque charge, la somme $pP + pR$ avoisine 0.4. Cela nous permettra de comparer les résultats. La charge A représente le cas où toutes les demandes sont robustes et la charge E réfère au cas où toutes les demandes sont permanentes. D'autre part, les trafics étant générés aléatoirement, le nombre total de demandes de connexions ne sera pas exactement le même d'une charge à l'autre. Dans ce cas, nous avons décidé de comparer les résultats par rapport à un taux moyen de blocage qui représente la division de la somme des connexions bloquées au produit du nombre de connexions par le nombre de scénarios. Supposons un réseau avec 200 demandes de connexions, 30 scénarios et un résultat de 3600 connexions bloquées fourni par l'algorithme H+. Dans ce cas, le taux moyen de blocage sera :

$$3600/(200*30) = 60\%$$

Les figures 4.13 et 4.14 illustrent les résultats obtenus pour les réseaux de 20 et 80 nœuds. Les abscisses représentent les différentes charges tandis que la légende est faite en fonction du nombre de longueurs d'onde utilisée.

Les résultats que nous obtenons nous montrent que l'algorithme se comporte de la même manière, quelle que soit la charge. Cependant, on observe que, pour le réseau de 20 nœuds avec 64 longueurs d'onde, la courbe subit un accroissement quasi linéaire.

Cela est dû au fait qu'on dispose d'assez de longueurs d'onde pour affecter nos connexions. Dans ce cas, on remarque que, pour la charge A qui ne contient que des demandes robustes, le blocage est nul ce qui est tout à fait normal. En revanche, on observe un accroissement du taux de blocage avec l'accroissement de la proportion de demandes permanentes. En effet, à chaque scénario de pannes, toutes les connexions robustes affectées par la panne seront reroutées, tandis que les demandes permanentes seront bloquées, d'où l'accroissement linéaire. Ce comportement sera observé chaque fois que le réseau disposera d'assez de ressources pour affecter toutes les connexions.

En revanche, si le réseau ne recèle pas un nombre suffisant de longueurs d'onde pour affecter toutes les connexions, on remarque que le taux de blocage ne varie pas significativement en fonction de la proportion de demandes robustes. Les résultats pour le réseau de 80 nœuds confirment cette observation.

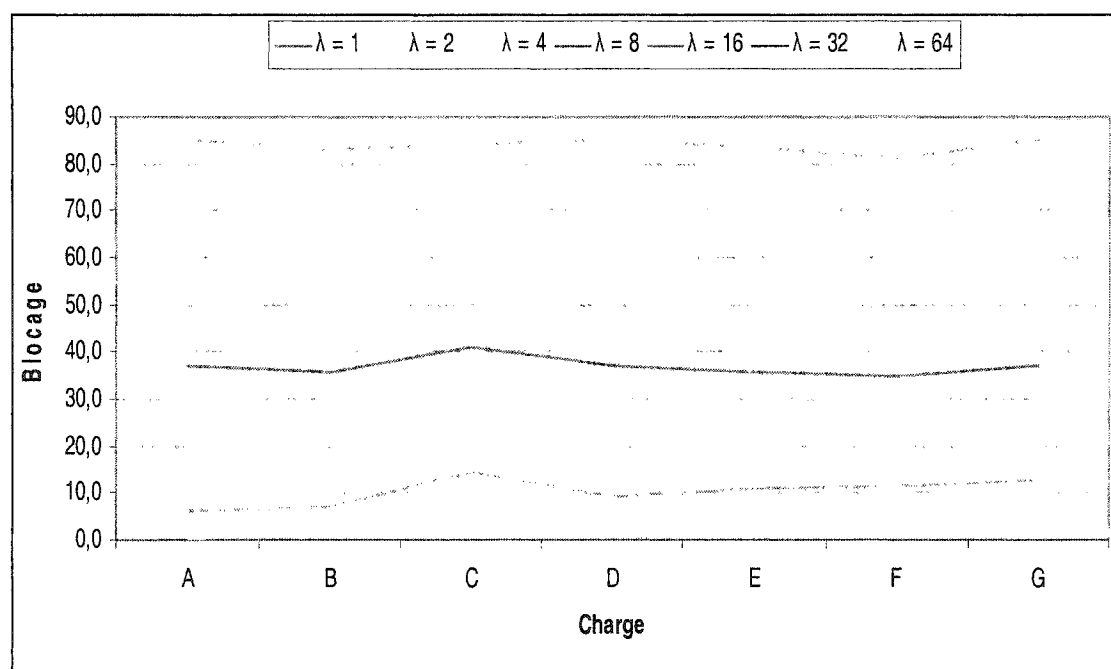


Figure 4.13 Blocage en pourcentage (n = 20 nœuds)

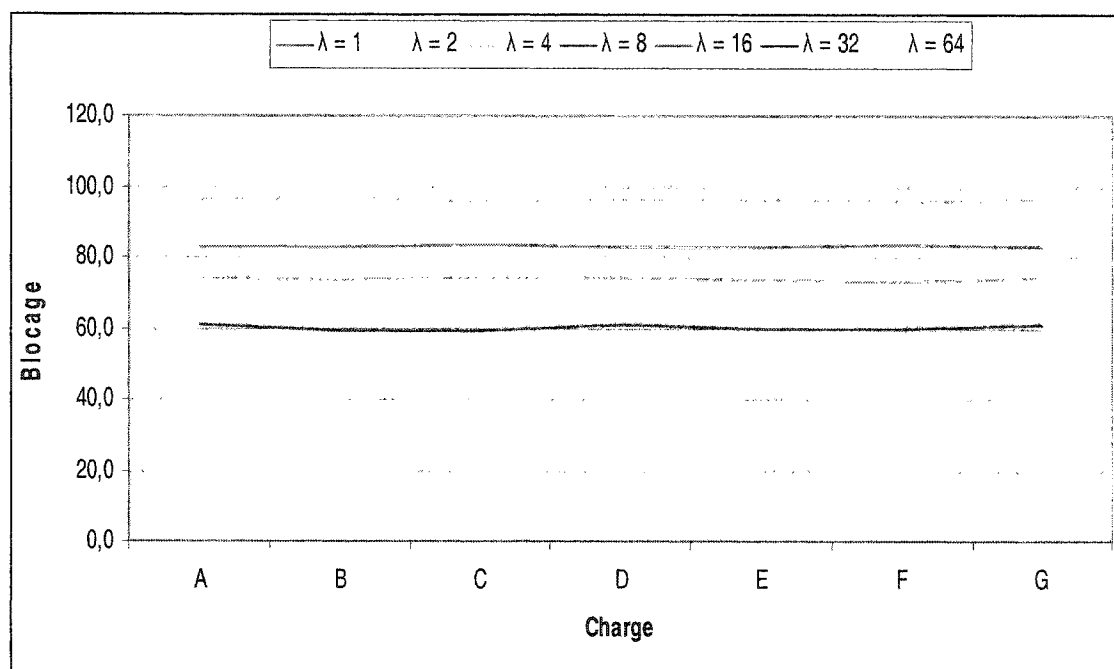


Figure 4.14 Blocage en pourcentage (n = 80 nœuds)

4.3.6 Effet de la connexité du réseau

Nous avons effectué nos tests sur des réseaux 2-connexes. Toutefois, nous avons des raisons de penser que l'utilisation de réseaux de connexité supérieure n'affectera pas significativement les performances de notre algorithme. En effet, nous avons appliqué une politique de plus courts chemins avec possibilité d'utiliser un deuxième plus court chemin en cas de blocage du premier. Dans ce cas, le nombre de chemins disjoints d'arcs disponibles n'influe pas réellement sur l'algorithme car H^+ ne sélectionne que les plus courts chemins. Par contre, le fait d'avoir plusieurs possibilités de chemins dans un réseau k -connexe peut, dans certains cas seulement, amener à obtenir des plus courts chemins de moindres coûts que ceux obtenus avec des réseaux de connexité inférieure.

4.4 Solutions optimales

Pour pouvoir évaluer nos solutions, nous allons tenter de trouver les valeurs optimales ou, à défaut, les bornes inférieures de la solution optimale. Nous avons utilisé le logiciel CPLEX, reconnu pour ses performances. Nous avons procédé étape par étape, en relâchant les contraintes et nous avons utilisé des charges avec $p = 0.20$.

Il faut mentionner que nous n'avons pas tenu compte des contraintes (3.12) car elles introduisent une non linéarité dans le modèle et il aurait fallu utiliser des méthodes de programmation plus complexes. La relaxation linéaire du modèle RWAP sans les contraintes (3.2a) et (3.2b) donne une borne inférieure trop petite. De fait, pour les réseaux de tests, la valeur de cette borne est 0, ce qui n'est d'aucune utilité pour évaluer la performance de notre heuristique. Si on ajoute les variables c dans $\{0,1\}$, le temps de calcul devient très grand, et le seul exemplaire que l'on peut traiter est celui de 5 nœuds avec 11 connexions, ce qui est très petit. Dans le cas de cet exemplaire, la borne trouvée est 0. Cela peut s'expliquer par la petite taille de l'exemplaire et par le fait que la majorité des variables x est fractionnaire. Si on ajoute les variables c et x dans $\{0,1\}$, on ne peut traiter que l'exemplaire de 5 nœuds. L'algorithme de branch-and-bound a trouvé une borne inférieure. Le temps de calcul CPU est d'environ 5 jours, la borne est 5 pour un GAP de 20%. Comme la solution de l'heuristique est aussi 5, on peut dire que nous sommes à 20% de l'optimalité. Cependant, l'exemplaire est trop petit pour conclure de la performance de l'heuristique. Considérant qu'il est difficile de résoudre des versions relaxées de RWAP, cela montre que les approches exactes sont à éviter pour RWAP et prône l'utilisation d'heuristiques. Cependant, nous avons quand même essayé d'obtenir d'autres moyens de comparaison.

4.4.1 Borne pour le scénario sans panne

Nous avons décidé de ne rechercher les bornes que pour le scénario sans panne. Appelons S^t une solution pour le scénario t , S^{*t} la solution optimale pour le scénario t avec S^{*0} , la solution optimale pour le scénario sans panne. Comme nous l'avons vu

précédemment, la solution sans panne aura un grand impact sur l'objectif final. Nous allons donc essayer de voir comment se comporte notre algorithme H+ pour le cas idéal. D'autre part, il faut remarquer que l'algorithme H+ tente d'optimiser chaque scénario individuellement, tandis que CPLEX a pour objectif une optimisation globale sur tous les scénarios. Cette optimisation globale peut, dans certains cas, entraîner le choix d'une valeur S^0 supérieure à S^{*0} , ce qui ne correspond pas totalement à la démarche de l'algorithme H+. Dans ce cas, il devient intéressant de pouvoir comparer les solutions obtenues avec H+ et celles obtenues avec CPLEX, et ce, pour le scénario sans panne. Les valeurs que nous obtenons avec CPLEX pour le scénario sans pannes sont des bornes inférieures de la solution optimale. Pour éviter des temps de calcul astronomiques, nous avons travaillé sur les réseaux de 10, 20 et 30 nœuds avec coûts de liens unitaires, en appliquant des charges correspondant à $p = 0.10$ et $p = 0.20$ lorsque cela était possible. Nous avons effectué nos tests avec 1, 2, 4 et 8 longueurs d'onde. Les tableaux 4.5 à 4.9 résument les résultats.

Le GAP représente l'écart entre la solution fournie par l'algorithme et la borne inférieure obtenue à partir de CPLEX. Il est calculé en effectuant le rapport de la valeur absolue de la différence de ces deux valeurs sur le nombre total de demande de connexions. Ces résultats viennent étayer certaines observations que nous avons faites précédemment. En effet, on remarque que l'écart entre la borne et la solution de l'heuristique augmente en fonction du nombre de longueurs d'onde disponibles avant de décroître. Ces caractéristiques devraient se vérifier pour des réseaux de tailles supérieures.

Au vu de ces résultats, on peut dire que l'algorithme H+ se comporte bien car l'écart vis-à-vis de la borne inférieure est raisonnable.

Tableau 4.5 Nombre de demandes bloquées (n = 10, p = 0.10)

λ	Solutions du scénario sans pannes			TEMPS (s)	
	H+	CPLEX	GAP (%)	H+	CPLEX
1	4	4	0	<1	<1
2	1	0	6.66	<1	<1
4	0	0	0	<1	<1
8	0	0	0	<1	<1

Tableau 4.6 Nombre de demandes bloquées (n = 10, p = 0.20)

λ	Solutions du scénario sans pannes			TEMPS (s)	
	H+	CPLEX	GAP (%)	H+	CPLEX
1	25	25	0	<1	<1
2	13	12	2.38	<1	8.33
4	2	1	2.38	<1	8.34
8	0	0	0	<1	3.77

Tableau 4.7 Nombre de demandes bloquées (n = 20, p = 0.10)

λ	Solutions du scénario sans pannes			TEMPS (s)	
	H+	CPLEX	GAP (%)	H+	CPLEX
1	59	57	2.53	<1	3.57
2	48	43	6.32	<1	60.46
4	26	22	5.06	<1	2492
8	8	3	6.32	<1	2309

Tableau 4.8 Nombre de demandes bloquées (n = 20, p = 0.20)

λ	Solutions du scénario sans pannes			TEMPS (s)	
	H+	CPLEX	GAP (%)	H+	CPLEX
1	128	126	1.32	<1	12.5
2	112	107	3.29	<1	450
4	84	75	5.92	<1	15 583
8	44	32	7.89	<1	267 416

Tableau 4.9 Nombre de demandes bloquées (n = 30, p = 0.10)

λ	Solutions du scénario sans pannes			TEMPS (s)	
	H+	CPLEX	GAP (%)	H+	CPLEX
1	137	135	1.18	<1	22
2	121	115	3.55	<1	4964
4	99	89	6.58	<1	16 569
8	71	57	8.28	2	270 703

4.4.2 Estimations des bornes de la solution optimale

Pour aller un peu plus loin, nous avons essayé d'estimer les bornes de la solution optimale sur l'ensemble des scénarios possibles. Pour ce faire, on a supposé que les bornes inférieures de la solution sans panne calculées précédemment représentent la solution choisie pour le scénario sans panne. Pour chaque scénario de pannes, le résultat du blocage sera supérieur ou égal à celui du scénario idéal car les connexions bloquées dans le scénario sans panne sont définitivement inactives. Pour estimer la borne inférieure, on va considérer le meilleur cas dans chaque scénario puisque notre objectif est une minimisation. Cela revient à dire que, lors d'une panne, toutes les demandes robustes passant par le lien défectueux sont réaffectées tandis que les demandes permanentes sont bloquées. Or nous savons que la distance en sauts optiques d'un

chemin représente le nombre maximal de scénarios de pannes qui peuvent affecter le trafic considéré. Dans le meilleur cas, le nombre de demandes bloquées dans le scénario t équivaut au résultat du scénario sans panne additionné du nombre de demandes permanentes passant par le lien défectueux car les demandes robustes sont réaffectées. De la même façon, nous évaluons la borne supérieure en utilisant le pire cas, qui revient à considérer qu'aucun trafic n'est rerouté lors d'une panne. Appelons n^{Pt} et n^{Rt} respectivement le nombre de connexions permanentes et robustes passant par le lien brisé dans le scénario t . Chaque connexion peut être affectée par une panne un nombre de fois égal à sa distance en sauts optiques. On peut donc dire que sur tous les scénarios de pannes, l'ensemble des connexions permanentes peut être affecté n^P fois, n^P représentant la somme des distances en sauts optiques de toutes les connexions permanentes affectées dans le scénario sans panne. De la même manière, on définit n^R pour les connexions robustes. Ces valeurs sont présentées dans les équations (4.1).

$$n^P = \sum_{k \in K^P} d_j^{kt_0} \quad j = d(k) \quad (4.1)$$

$$n^R = \sum_{k \in K^R} d_j^{kt_0}$$

On peut donc écrire l'inéquation suivante pour le scénario $t \in T$:

$$\sum_{k \in K} b^{kt_0} + n^{Pt} + n^{Rt} \geq \sum_{k \in K} b^{kt} \geq \sum_{k \in K} b^{kt_0} + n^{Pt} \quad (4.2)$$

Par généralisation, on peut effectuer une sommation sur l'ensemble des scénarios, ce qui donne :

$$\begin{aligned}
\sum_{t \in T} \left(\sum_{k \in K} b^{kt_0} + n^{Pt} + n^{Rt} \right) &\geq \sum_{t \in T} \sum_{k \in K} b^{kt} \geq \sum_{t \in T} \left(\sum_{k \in K} b^{kt_0} + n^{Pt} \right) \\
|T| \times \sum_{k \in K} b^{kt_0} + \sum_{t \in T} n^{Pt} + \sum_{t \in T} n^{Rt} &\geq \sum_{t \in T} \sum_{k \in K} b^{kt} \geq |T| \times \sum_{k \in K} b^{kt_0} + \sum_{t \in T} n^{Pt} \\
|T| \times \sum_{k \in K} b^{kt_0} + n^P + n^R &\geq \sum_{t \in T} \sum_{k \in K} b^{kt} \geq |T| \times \sum_{k \in K} b^{kt_0} + n^P
\end{aligned} \tag{4.3}$$

Concernant la qualité de l'estimation, plusieurs observations peuvent être faites. Premièrement, il peut exister plusieurs solutions de coût optimal pour le scénario sans panne. Les bornes sont intimement liées à la proportion de chemin permanents et de chemins robustes actives dans le cas idéal et à la distance de ces chemins. Si une solution optimale S^{0*1} affecte des chemins permanents de distance totale supérieure à une solution optimale S^{0*2} , la borne inférieure obtenue à partir de S^{0*1} sera supérieure à celle obtenue à partir de S^{0*2} . Si S^{0*1} affecte des chemins robustes de distance totale inférieure à S^{0*2} , la borne supérieure obtenue à partir de S^{0*1} sera inférieure à celle obtenue à partir de S^{0*2} . On voit donc qu'en fonction de la solution S^{0*} choisie, l'écart entre la borne inférieure et la borne supérieure peut être réduit, permettant ainsi de mieux estimer la solution optimale. Lorsque toutes les demandes sont permanentes, les deux bornes sont identiques. En revanche, si toutes les demandes sont robustes, la borne inférieure est nulle tandis que la borne supérieure peut être élevée, aspect moins utile pour comparer des résultats.

Pour obtenir la meilleure estimation, il aurait fallu répertorier toutes les solutions possibles pour le cas idéal et comparer les bornes obtenues. Dans notre cas, la borne importante est la borne inférieure car la solution de notre heuristique sera toujours supérieure ou égale à la solution optimale. En plus de cela, il nous est difficile d'évaluer la borne inférieure de manière absolue avec l'inéquation (4.3) à cause de la multiplicité

des solutions pour le cas sans pannes. Cependant nous savons que la valeur de n^P est positive. Nous pouvons donc écrire :

$$\begin{aligned} \sum_{t \in T} \sum_{k \in K} b^{kt} &\geq |T| \times \sum_{k \in K} b^{kt_0} + n^P \\ \sum_{t \in T} \sum_{k \in K} b^{kt} &\geq |T| \times \sum_{k \in K} b^{kt_0} \end{aligned} \quad (4.4)$$

C'est la borne inférieure que nous avons utilisée. Elle donne une idée sur la solution. Cependant, elle est très proche de la solution optimale si le nombre de longueurs d'onde disponibles sur le réseau est faible ou, dans le cas contraire si on dispose d'assez de ressources pour satisfaire une très grande majorité de demandes. Dans un cas intermédiaire, il est à craindre que cette borne soit trop distante de la solution optimale. Dans ce cas, cette borne peut être aussi très éloignée de notre solution et la comparaison devient difficile.

Nous avons aussi implémenté un autre algorithme d'affectation de longueurs d'onde pour servir de comparaison. C'est un algorithme simple dont le routage est aussi basé sur l'algorithme de Dijkstra pour trouver les plus courts chemins. Une fois ces chemins trouvés, l'affectation de longueurs d'onde s'effectue en inspectant séquentiellement les longueurs d'onde. On sélectionne une longueur d'onde selon une numérotation donnée. Ensuite, on parcourt séquentiellement l'ensemble des trafics et si aucun trafic adjacent au trafic courant n'utilise la longueur d'onde étudiée, on affecte cette longueur d'onde au trafic. Une fois tous les trafics inspectés, on recommence le processus avec une autre longueur d'onde. L'algorithme s'arrête lorsque tous les trafics sont affectés ou lorsque toutes les longueurs d'onde ont été utilisées. Cet algorithme est similaire à l'algorithme de chemins disjoints d'arcs présenté à la section 2.4.5. Nous appellerons cet algorithme CDA+. Nous allons maintenant présenter les résultats obtenus.

Tableau 4.10 Estimation des bornes ($n = 10$, $p = 0.10$)

λ	Résultats			Écart (%)	
	Inf	CDA+	H+	CDA+	Inf
1	64	84	84	0,00	8,33
2	0	37	37	0,00	15,42
4	0	8	8	0,00	3,33
8	0	8	8	0,00	3,33

Tableau 4.11 Estimation des bornes ($n = 10$, $p = 0.20$)

λ	Résultats			Écart (%)	
	Inf	CDA+	H+	CDA+	Inf
1	400	426	426	0,00	3,87
2	192	254	254	0,00	9,23
4	16	101	84	-2,53	10,12
8	0	41	41	0,00	6,10

Tableau 4.12 Estimation des bornes ($n = 20$, $p = 0.10$)

λ	Résultats			Écart (%)	
	Inf	CDA+	H+	CDA+	Inf
1	1482	1677	1573	-5,06	4,43
2	1118	1376	1324	-2,53	10,03
4	572	896	809	-4,24	11,54
8	78	392	357	-1,70	13,58

Tableau 4.13 Estimation des bornes ($n = 20$, $p = 0.20$)

λ	Résultats			Écart (%)	
	Inf	CDA+	H+	CDA+	Inf
1	3276	3503	3368	-3,42	2,33
2	2782	3156	2988	-4,25	5,21
4	1950	2481	2354	-3,21	10,22
8	832	1528	1465	-1,59	16,02

Tableau 4.14 Estimation des bornes ($n = 30$, $p = 0.10$)

λ	Résultats			Écart (%)	
	Inf	CDA+	H+	CDA+	Inf
1	5130	5341	5219	-1,90	1,39
2	4370	4893	4631	-4,08	4,06
4	3382	4111	3878	-3,63	7,72
8	2166	2906	2696	-3,27	8,25

Dans les tableaux 4.10 à 4.14, les résultats de la borne inférieure, de l'algorithme CDA+ ainsi que de l'algorithme H+ sont présentés. On y retrouve aussi les écarts en pourcentages entre notre solution, la solution du CDA+ et les bornes évaluées. En observant ces résultats, on constate que les solutions obtenues par notre algorithme H+ sont relativement proches des bornes obtenues et qu'elles sont meilleures que celles obtenues à partir du CDA+. Comme nous l'avons mentionné précédemment, l'écart entre notre solution et la solution optimale croît avant de décroître en fonction de l'augmentation du nombre de longueurs d'onde disponibles pour une topologie et une demande données.

Après les différentes estimations et analyses que nous avons effectuées, nous pouvons dire que l'algorithme H+ apporte un avantage important par rapport aux méthodes exactes. Premièrement, les solutions obtenues semblent relativement bonnes comparativement aux bornes de la solution optimale. D'autre part, les temps de calculs sont nettement meilleurs dans le cas de l'algorithme H+ comparé à la méthode exacte, dû au caractère NP-difficile du problème.

CHAPITRE V

CONCLUSION

Le problème de routage et d'affectation de longueurs d'onde dans les réseaux optiques WDM est un problème difficile que nous avons essayé de résoudre dans ce mémoire en proposant des heuristiques. Dans ce chapitre de clôture, nous présentons d'abord une synthèse générale des travaux suivie des limitations de nos algorithmes. Enfin, nous terminons en abordant les indications de travaux futurs.

5.1 Synthèse des travaux

Le problème que nous avons étudié consiste à trouver un chemin et une longueur d'onde disponible sur l'ensemble du chemin. Nous avons considéré la continuité de longueur d'onde sur tout le chemin. Par ailleurs, nous considérons la possibilité d'avoir des demandes permanentes et des demandes fiables. En cas de pannes les affectant, les demandes fiables bénéficient d'un reroutage tandis que les demandes permanentes sont bloquées. Le routage est un routage classique de plus court chemin utilisant une version améliorée de l'algorithme de Dijkstra. L'affectation de longueurs d'onde se fait, quant à elle, à partir de la transformation du problème initial en un problème de coloriage de graphe. Dans ce cas, l'affectation de longueurs d'onde est un problème NP-difficile. L'objectif fixé était de minimiser le nombre de demandes bloquées sur l'ensemble des scénarios, soient le scénario sans pannes et tous les scénarios de pannes de liens simples. Comme ce problème est difficile, il ne peut être résolu de manière exacte pour des instances de tailles moyennes et élevées. On a dû alors recourir à des heuristiques.

Dans ce mémoire, nous avons d'abord proposé une formulation mathématique du problème répertoriant l'ensemble des contraintes de notre problème et tenant compte de la présence de demandes fiables et permanentes. Par la suite, nous nous sommes basé

sur une heuristique simple, l'algorithme H, pour proposer l'algorithme H+ dont le but est d'améliorer les résultats initiaux d'un algorithme d'affectation quelconque. L'implémentation de cet algorithme consiste premièrement à calculer la solution obtenue à partir d'un algorithme de base qui a été l'algorithme H pour cette recherche. Une fois cette valeur connue, nous avons cherché à réduire le nombre de demandes de connexions à affecter pour améliorer la valeur de la solution. Nous avons obtenu des résultats intéressants. Nous avons en outre montré que la proportion de demande robuste par rapport au nombre de demandes totales du réseau n'influe pas significativement sur le comportement de l'algorithme.

Pour évaluer la performance de notre algorithme, nous avons tenté d'obtenir des bornes de solutions optimales à l'aide du logiciel CPLEX. Cependant, la complexité du problème nous a empêché d'obtenir des solutions couvrant l'ensemble des scénarios. Nous avons donc décidé de rechercher des estimations de solutions optimales seulement pour le scénario sans panne. Le résultat de ce scénario étant déterminant pour la suite du problème, cette valeur apparaît intéressante pour la comparaison. À partir de ces résultats, nous avons fait une estimation d'une borne inférieure de la solution optimale. Nous avons aussi implémenté un algorithme simple de résolution, dénommé CDA+, pour la comparaison des résultats. Au vu de tous les résultats, nous pouvons dire que l'algorithme H+ est intéressant car il fournit des solutions très acceptables en des temps de calculs raisonnables.

5.2 Limitations des travaux

En dépit des performances intéressantes obtenues avec notre algorithme, il subsiste quelques limitations. En effet, les estimations que nous avons effectuées quant aux bornes de l'heuristique peuvent être éloignées de la solution optimale. Il serait intéressant de pouvoir raffiner ces estimations de bornes inférieures, ce qui permettrait de réduire l'écart entre notre solution et ces bornes. À cause des temps de calculs

impliqués par CPLEX, nous n'avons pu comparer le comportement de notre réseau pour des réseaux de grandes tailles avec des charges importantes.

D'autre part, notre simulation considère que chaque lien unidirectionnel ne comporte qu'une seule fibre, ce qui n'est pas toujours de mise dans les réseaux actuels. Pour un même nombre de longueurs d'onde par lien, les résultats pourraient être différents, voire meilleur, si ce nombre de longueurs d'ondes étaient partagés sur plusieurs fibres.

5.3 Indications des travaux futurs

Le problème de routage et d'affectation de longueurs d'onde demeure un problème ouvert. Cette recherche recèle de nombreuses pistes d'améliorations. Premièrement, il faudrait s'atteler à déterminer des bornes inférieures pour la solution optimale de l'heuristique. Cela pourrait s'effectuer en améliorant le modèle, ce qui permettrait à des logiciels comme CPLEX de déterminer les bornes sur l'ensemble des scénarios. Comme changement au modèle, on pourrait relâcher la contrainte du plus court chemin en explorant les k premiers plus courts chemins pour un trafic donné.

Pour l'algorithme H+, nous avons défini un indice nous permettant de classifier les connexions. Il faudrait essayer de définir d'autres indices et évaluer les performances obtenues à partir de ces indices. Nous avons aussi vu que l'algorithme H+ peut être appliqué en utilisant différents algorithmes d'affectation. On pourrait donc se livrer à une classification des algorithmes d'affectations auxquels l'algorithme H+ serait applicable. Par ailleurs, il faudrait évaluer explicitement l'effet de la connexité du réseau sur les performances de l'algorithme. Enfin, des heuristiques nouvelles ou adaptées pourraient être proposées pour résoudre ce problème en considérant des demandes permanentes et robustes. Dans ce cas, l'algorithme H+ pourrait servir de base de comparaison.

BIBLIOGRAPHIE

Ahuja R., Magnanti T., Orlin J., "Networks Flows", Prentice-Hall, 1993.

Alanyali M., Ayanoglu E., "Provisioning Algorithms for WDM Optical Networks", INFOCOM '98, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol.2, 1998, San Francisco (CA), pp. 910 -918.

Bampis E., Rouskas G.N., "The Scheduling and wavelength Assignment Problem in Optical WDM Networks", IEEE Journal of Lightwave Technology, vol. 20, No. 5, May 2002, pp. 782 -789.

Banerjee S., Chen C., "Design of Wavelength-Routes Optical Networks for Circuit Switched Traffic", Global Telecommunications Conference, GLOBECOM '96, Communications: The Key to Global Prosperity, vol. 1, 1996, London (UK), pp. 306-310.

Banerjee S., Mukherjee B., "A practical approach for routing and wavelength assignment in large wavelength routed optical networks", IEEE J.Sel. Areas in Communs., vol. 14, No. 5, June 1996, pp 903-908.

Chamberland S., "On the Joint Link Dimensionning and Routing Metric Assignment Problem for Reliable WDM Networks", soumis à IEEE\ACM Transactions on Networking, 2001.

Chamberland S., Oulaï D., Pierre S., "Routing and Wavelength Assignment for Permanent and Reliable Wavelength Path in WDM Networks", IEEE CCECE 2002, Canadian Conference on Electrical and Computer Engineering, 2002. vol. 1, Winnipeg (MB), pp. 301-306.

- Chen B., Wang J., "Efficient Routing and Wavelength Assignment for Multicast in WDM Networks", IEEE Journal on Selected Areas in Communications, vol. 20, No. 1, January 2002, pp. 97-109.
- Fumagalli A., Valcarengh L., "The preplanned weighted restoration scheme", IEEE Workshop on High Performance Switching and Routing, 2001, Dallas (Texas), pp. 36-41.
- Gangxiang S., Bose S.; Hiang C., "Low Delay Heuristic Algorithm for Wavelength Assignment in WDM Networks", Proceedings, IEEE International Conference on Networks (ICON '99), 1999, Brisbane (Australia), pp. 3 -10.
- Gangxiang S., Bose S.; Hiang C., Chao Lu, "Designing WDM Optical Network for Reliability: Routing Light Paths Efficiently for Path Protection", Optical Fiber Communication Conference, vol. 3, 2000, Baltimore, (USA), pp. 50 -52.
- Gary M.R., Johnson D.S. "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman & Co., San Francisco, CA (1979).
- Glover F., "Tabu Search- Part I", ORSA Journal on Computing, vol. 1, No.3, 1989, pp.190-206.
- Handley M., Leeson M., Phillips A., Senior J., Ball P., Wright I., "Comparison and simulation of wavelength allocation algorithms on passive optical networks", IEEE Colloquium on Multiwavelength Optical Networks: Devices, Systems and Network Implementations, No. 1998/296, 1998, London, pp. 8/1-8/6.

Harder E., Lee S., Choi H., "On Wavelength Assignment in WDM Optical Networks", Proceedings of the Fourth International Conference on Massively Parallel Processing Using Optical Interconnections, 1997, pp. 32-38.

Hediblé C., "Algorithme génétique pour l'affectation de cellules à des commutateurs dans les réseaux mobiles", Mémoire de maîtrise, École Polytechnique de Montréal, Septembre 2000.

Houéto F., "Affectation de cellules à des commutateurs dans les réseaux de communications personnelles", Mémoire de maîtrise, École Polytechnique de Montréal, Août 1999.

Inkret R., Mikac B., Podnar I., "A heuristic approach to Wavelength Assignment in All-Optical Networks", Mediterranean Electrotechnical Conference, MELECON 98, vol. 2, 1998, Tel Aviv (Israel), pp. 759-763.

Jia X., Du D., Hu X., Lee M., Gu J., "Optimization of Wavelength Assignment for QoS Multicast in WDM Networks", IEEE Transactions on Communications, vol. 49, No. 2, February 2001, pp. 341-350.

Lima E., Stubkjaer K., "An algorithm for link restoration of wavelength routing optical networks", IEEE International Conference on Communications, ICC '99, vol. 3, 1999, Vancouver (AB), pp. 2055-2061.

Manohar Pallavi, Manjunath D., Shevgaonar R., "Routing and Wavelength Assignment in Optical Networks From Edge Disjoint Path Algorithm", IEEE Communications Letters, Vol. 6, No. 5, May 2002, pp. 211-213.

Mikac B., Inkret R., Podnar I., "An Application of Heuristic search Techniques in Telecommunication System Design", Proceedings Second International Conference on University Knowledge-Based Intelligent Electronic Systems, vol. 1, 1998, Adelaide (Australia), pp. 311-317.

Modiano E., Narula-Tam A., "Designing Survivable Networks using Effective Routing and Wavelength Assignment (RWA)", Optical Fiber Communication Conference and Exhibit, vol. 2, 2001, Dallas (Texas), pp. TuG5 -T1-3.

Mohan G., Murthy S., Somani A., "Efficient Algorithms for Routing Dependable Connections in WDM Optical Networks", IEEE/ACM Transactions on Networking, vol. 9, No. 5, October 2001, pp.553-566.

Mokhtar A., Azizoglu M., "Dynamic Route selection and Wavelength Assignment in All-Optical Networks", Lasers and Electro-Optics Society Annual Meeting, vol. 1, 1995, San Francisco (CA), pp. 220-221.

Mokhtar A., Azizoglu M., "Adaptative Techniques for Routing and Wavelength Assignment in All- Optical WANS", IEEE 39th Midwest symposium on Circuits and Systems, vol. 3, 1996, Ames (Iowa), pp. 1195-1198.

Mokhtar A., Azizoglu M., "Adaptive Wavelength Routing in All-Optical Networks", IEEE/ACM Transactions on Networking, vol. 6, No. 2, April 1998, pp. 197-206.

Mukherjee B., Ramamurthy S., Banerjee D., Mukherjee A., "Some principles for designing a wide-area optical network", INFOCOM '94 Networking for Global Communications., 13th Proceedings IEEE, vol.1, 1994, Toronto (Canada), pp. 110-119.

Nagatsu N., Hamazumi Y., Sato K., "Optical path accommodation designs applicable to large scale networks", IEICE Trans. On Commun., Vol. E78-B, No. 4, Apr. 1995, pp. 597-607.

Ohta S., Greca A., "Comparison of Routing and Wavelength Assignment Algorithms for Optical Networks", IEEE Workshop on High Performance Switching and Routing, 2001, Dallas (Texas), pp. 146-149.

Peng W., Wei C., "Distributed Wavelength Assignment Protocols with Priority for WDM All-Optical Networks", Proceedings Ninth International Conference on computer Communications and Networks, 2000, Las Vegas (Nevada), pp. 625-630.

Prins C., "Algorithmes de graphes", Eyrolles, 1994.

Pujolle G., "Les réseaux ", Eyrolles, 1999.

Pujolle G., "Télécommunication et réseaux", Eyrolles, 1992.

Ramswami R., Sivarajan K., "Routing and Wavelength Assignment in All-Optical Networks", IEEE/ACM Transactions on Networking, vol. 3, No. 5, October 1995, pp.489-500.

Rossier-Ramuz D., Rodellar D., Scheurer R., "Dynamic Protection Set-up in Optical VPN using Mobile Agent Ecosystem", DRCN 2001, Budapest, 2002.

Somani A., Azizoglu M., "Wavelength Assignment Algorithms for Wavelength Routed Interconnection of LANs", Journal of Lightwave Technology, vol. 18, No.12, December 2000, pp. 1807-1817.

Stern Thomas, Krishna Bala, "Multiwavelength Optical Networks", Addison-Wesley, 2000.

Sreenath N., Phanibhushan P., Mohan G., Murthy C., "Design of survivable WDM networks for carrying ATM traffic", Journal of Computer Communications 25, 2002, pp 485-500.

Taechan L., Kyungsik L., Sungsoo P., "Optimal Routing and Wavelength Assignment in WDM Ring Networks", IEEE Journal on Selected Areas in Comm., vol. 18, No. 10, October 2000, pp 2146-2154.

Weifa L., Xiaojun S., "Permutation Routing in All-Optical networks", IEEE Transact. on circuits and Systems-I: Fundamental Theory and Applications, vol. 49, No. 4, April 2002, pp. 533-538.

Wigderson Avi, "Improving the Performance Guarantee for Approximate Graph Coloring", Journal of ACM, vol. 30, No. 4, October 1983, pp 729-735.

Zang Z., Acampora A., "A Heuristic Wavelength Assignment Algorithm for Multihop WDM Networks with Wavelength Routing and Wavelength Re-Use", IEEE/ACM Transactions on Networking, vol. 3, No. 3, June 1995, pp. 281 -288.

Zhiang X., Qiao C., "Wavelength Assignment for Dynamic traffic in Multi-fiber WDM Networks", Proceedings 7th International Conference on Computer Communications and Networks, 1998, Lafayette (Louisiana), pp. 479-485.

Zhou B., Mouftah H., "Adapative Least Loaded Routing for Multi-fiber WDM Networks Using Approximate Congestion Information", IEEE International Conference on Communications, Vol.5, 2002, New-York (NY), pp. 2745-2749.

Zyane Abdellah, "Les réseaux optiques sous la technologie WDM : Principes et perspectives", Rapport de stage, École Polytechnique de Montréal, 2001.