

**Titre:** Heuristique taboue pour le routage et l'affectation de longueurs  
Title: d'onde dans les réseaux optiques multigranulaires

**Auteur:** Jean-Marc Hyppolite  
Author:

**Date:** 2005

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Hyppolite, J.-M. (2005). Heuristique taboue pour le routage et l'affectation de longueurs d'onde dans les réseaux optiques multigranulaires [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.  
Citation: <https://publications.polymtl.ca/7630/>

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

**URL de PolyPublie:** <https://publications.polymtl.ca/7630/>  
PolyPublie URL:

**Directeurs de recherche:** Samuel Pierre, & Philippe Galinier  
Advisors:

**Programme:** Génie informatique  
Program:

UNIVERSITÉ DE MONTRÉAL

HEURISTIQUE TABOUE POUR LE ROUTAGE  
ET L'AFFECTATION DE LONGUEURS D'ONDE DANS  
LES RÉSEAUX OPTIQUES MULTIGRANULAIRES

JEAN-MARC HYPPOLITE  
DÉPARTEMENT DE GÉNIE INFORMATIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

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

OCTOBRE 2005

© JEAN-MARC HYPPOLITE, 2005.



Library and  
Archives Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-16798-4

*Our file* *Notre référence*  
ISBN: 978-0-494-16798-4

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

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

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

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

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

\*\*  
Canada

UNIVERSITÉ DE MONTRÉAL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

HEURISTIQUE TABOUE POUR LE ROUTAGE  
ET L'AFFECTATION DE LONGUEURS D'ONDE DANS  
LES RÉSEAUX OPTIQUES MULTIGRANULAIRES

Présenté par : JEAN-MARC HYPPOLITE  
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées  
a été dûment accepté par le jury d'examen constitué de :

Mme NICOLESCU Gabriela, Doct., présidente  
M. PIERRE Samuel, Ph.D., membre et directeur de recherche  
M. GALINIER Philippe, Doct., membre et codirecteur de recherche  
M. QUINTERO Alejandro, Doct., membre

## **REMERCIEMENTS**

Je voudrais tout d'abord remercier mon directeur de recherche, le professeur Samuel Pierre ainsi que mon codirecteur M. Philippe Galinier, pour avoir cru que je pouvais mener à bien ce travail et pour leur patience dans les moments difficiles.

Je remercie ensuite tous mes collègues du Laboratoire de Recherche en Réseautique et Informatique Mobile (LARIM), pour leurs critiques constructives et leurs conseils très appréciés lors de la réalisation du travail. Merci spécialement à Sedogbo Rolland W. Mewanou.

Ma reconnaissance va également à ma famille qui a su m'accompagner de ses prières et de son assistance tout au long de ce parcours.

Enfin, merci à tous mes amis, camarades de promotion qui m'ont aidé et encouragé lors de mes études.

## RÉSUMÉ

Au cours des dix dernières années, les réseaux optiques utilisant la technologie WDM (*Wavelength Division Multiplexing*) ont reçu de plus en plus d'attention, se présentant comme un moyen économique pour répondre à la demande croissante de trafic. Récemment, une hiérarchisation de ces réseaux a retenu l'attention de la communauté scientifique promettant de réduire davantage les coûts et de répondre au problème d'évolutivité dans de tels réseaux. Cette hiérarchisation permet notamment la commutation d'une fibre, d'une bande d'onde et d'une longueur d'onde à l'intérieur du même réseau. Le routage dans ces réseaux, en plus de trouver un chemin pour un trafic et lui affecter une longueur d'onde, doit chercher à regrouper les chemins optiques de manière efficace de façon à tirer avantage le plus possible de la hiérarchisation. Ceci constitue le problème du routage optique multigranulaire qui fait l'objet de ce mémoire.

L'objectif de ce mémoire est de proposer une méthode de résolution pour le problème de routage optique multigranulaire. Nous avons appliqué la contrainte de continuité de longueurs d'onde et nous avons considéré le cas où l'ensemble des connexions optiques à établir est connu dès le départ. Notre but est de minimiser le nombre de ports des commutateurs optiques globalement dans le réseau, tout en satisfaisant l'ensemble des demandes de connexion, étant donnés un nombre de longueurs d'onde par bande d'onde, un nombre de longueurs d'onde par fibre et un nombre de fibres par lien.

Le problème de routage et d'affectation de longueurs d'onde dans les réseaux multigranulaires ayant été répertorié NP-complet, nous avons choisi de développer une heuristique pour le résoudre. Notre solution traite conjointement la sélection des routes et l'affectation des longueurs d'onde, et est basée sur la recherche taboue. Nous avons réalisé des séries d'expériences pour analyser la performance de notre méthode, en mettant l'accent sur le nombre de ports globalement dans le réseau et le temps d'exécution. Nous avons étudié l'effet des paramètres que nous avons introduit au niveau de notre méthode, à savoir les  $k$  plus courts chemins entre une paire de nœuds, la

longueur de la liste taboue et le nombre maximum d'itérations. Nous avons remarqué que le fait d'augmenter le nombre de plus courts chemins entre les paires pour la sélection des routes n'impliquait pas une diminution du nombre de ports. Nous avons comparé les résultats obtenus pour le réseau NSFNET à ceux de l'algorithme BPHT de Cao et al. Dans l'ensemble, les gains par rapport à cet algorithme sont très bons. Toutefois, les temps d'exécution de notre méthode sont très supérieurs à ceux du BPHT, mais restent assez raisonnables. Ces résultats étaient prévisibles dans la mesure où le BPHT est une méthode de construction moins performante mais plus rapide qu'une heuristique basée sur la recherche locale comme les heuristiques taboues.

## ABSTRACT

For the past ten years, optical networks using WDM technology (Wavelength Division Multiplexing) have received more and more attention, presenting themselves as an economic way to meet the growing traffic demand. Recently, a hierarchisation of these networks has caught the attention of the scientific community promising to reduce further the cost and deal with the scaling problem in such networks. This hierarchisation enables the switching of whole fibers, wavebands and wavelengths in the same network. In addition to finding a path to route a traffic and assigning a wavelength on that path, routing in these networks has to group lighpaths efficiently in order to take advantage as much as possible of this hierarchisation. This is known as the routing problem in multigranular optical networks and will be studied in that paper.

The objective of this paper is to propose a solution to the optical multigranular routing problem, considering the wavelength continuity constraint and a given set of connections to satisfy. Specifically, our goal is to minimize the number of ports of optical nodes globally in the network, while satisfying the set of connection demands, and given a number of wavelengths per waveband, a number of wavelengths per fiber and a number of fibers per link.

The routing and wavelength assignment in multigranular optical networks has been proved to be NP-complete. Therefore, we chose to develop a heuristic to solve it. Our solution deals with the routing and wavelength assignment jointly and uses tabu search. For the performance analysis, we made several tests, and we compare our results with those of algorithm BPHT. We focused on the number of ports globally in the network and the execution time. Our results were better than BPHT except for the execution time which is higher. These results were predictable, since BPHT is a construction method less powerful than a heuristic based on tabu search which is an iterative method.

## TABLE DES MATIERES

REMERCIEMENTS .....	IV
RÉSUMÉ .....	V
ABSTRACT .....	VII
TABLE DES MATIERES .....	VIII
LISTE DES FIGURES.....	XI
LISTE DES TABLEAUX.....	XIII
SIGLES ET ABREVIATIONS.....	XV
CHAPITRE I INTRODUCTION.....	1
1.1.    Définitions et concepts de base .....	1
1.2.    Éléments de problématique .....	4
1.3.    Objectifs de recherche .....	6
1.4.    Esquisse méthodologique.....	6
1.5.    Plan du mémoire .....	6
CHAPITRE II ROUTAGE ET AFFECTATION DE LONGUEURS D'ONDE DANS LES RÉSEAUX OPTIQUES MULTIGRANULAIRES .....	8
2.1.    Réseaux à routage de longueurs d'onde.....	8
2.1.1.    La technologie WDM.....	9
2.1.2.    Routage optique .....	10
2.1.3.    Gestion des réseaux WDM.....	13
2.2.    Réseaux multigranulaires à trois niveaux de granularité .....	15
2.2.1.    Motivation et objectifs de la multigranularité.....	15
2.2.2.    Commutateurs multigranulaires .....	17
2.2.3.    Extensions proposées au GMPLS .....	20
2.2.4.    Routage optique .....	20

2.2.5. Gestion de pannes .....	23
2.3. Algorithmes de routage multigranulaire .....	23
2.3.1. Algorithmes de routage multigranulaire dynamique.....	24
2.3.2. Algorithmes de routage multigranulaire statique.....	27
2.4. Réseaux multigranulaires à deux niveaux de granularité.....	37
CHAPITRE III ALGORITHMES DE ROUTAGE PROPOSÉS .....	38
3.1. Définition du problème .....	38
3.1.1. Calcul du nombre de ports .....	41
3.1.2. Exemple illustratif.....	51
3.2. Algorithmes proposés .....	53
3.2.1. Schéma général .....	53
3.2.2. Algorithme de Yen.....	56
3.2.3. Tri des paires .....	57
3.2.4. Solution initiale .....	58
3.2.5. Mouvement .....	61
3.2.6. Mise à jour du nombre de ports au niveau d'un nœud .....	65
3.2.7. Présentation globale de l'algorithme TabouMRWA.....	75
3.3. Analyse de la complexité de l'algorithme TabouMRWA.....	76
CHAPITRE IV IMPLÉMENTATION ET RÉSULTATS.....	78
4.1. Structures de données utilisées.....	78
4.1.1. La classe Réseau .....	78
4.1.2. La classe Nœud .....	79
4.1.3. La classe Lien.....	81
4.1.4. La classe Route .....	82
4.1.5. La classe Paire.....	83
4.1.6. La classe Session.....	85
4.1.7. La classe Algorithmes .....	85
4.2. Données utilisées.....	86

4.2.1. Fichier de topologie.....	86
4.2.2. Modèle de trafic .....	87
4.3. Plan d'expérience.....	87
4.3.1. Les indices de performance.....	88
4.3.2. Les facteurs de simulation.....	88
4.3.3. Choix des niveaux des facteurs .....	89
4.3.4. Types d'expériences choisies.....	90
4.4. Expérimentation et tests .....	91
4.4.1. Étape 1 : expériences « un facteur à la fois ».....	92
4.4.2. Étape 2 : calibration des facteurs secondaires.....	96
4.4.3. Étape 3 : comparaison entre TabouMRWA et BPHT .....	98
CHAPITRE V CONCLUSION.....	111
5.1. Synthèse des travaux .....	111
5.2. Limitations des travaux .....	112
5.3. Travaux futurs .....	113
BIBLIOGRAPHIE .....	115

## LISTE DES FIGURES

Figure 1.1. Réseau WDM à routage de longueur d'onde.....	2
Figure 2.1 Liaison WDM point à point.....	9
Figure 2.2 Brasseur de longueurs d'onde.....	10
Figure 2.3 Architecture MG-OXC proposée par Mouftah et al. ....	17
Figure 2.4 Architecture MG-OXC proposée par Cao et al. ....	19
Figure 2.5 Algorithme WA-MF-WBS .....	34
Figure 3.1 Coût d'un coude.....	47
Figure 3.2 Calcul du nombre de ports d'un commutateur .....	50
Figure 3.3 Illustration du problème MIN-MRWA.....	52
Figure 3.4 Schéma général de l'algorithme TabouMRWA .....	55
Figure 3.5 Algorithme de Yen .....	57
Figure 3.6 Recherche d'une solution initiale .....	59
Figure 3.7 Algorithme d'affectation de longueurs d'onde WA-MF .....	60
Figure 3.8 Étapes d'un mouvement .....	62
Figure 3.9 Illustration des étapes d'un mouvement .....	64
Figure 3.10 Mise à jour du nombre de ports d'un nœud $v$ dans le cas de l'affectation d'une longueur d'onde .....	68
Figure 3.11 Mise à jour du nombre de ports d'un nœud $v$ dans le cas de la désaffectation d'une longueur d'onde .....	72
Figure 3.12 Mise à jour du nombre de ports d'un nœud $v$ dans le cas de l'affectation d'une bande d'onde .....	73
Figure 3.13 Mise à jour du nombre de ports d'un nœud $v$ dans le cas de la désaffectation d'une bande d'onde .....	74
Figure 3.14 Algorithme TabouMRWA.....	75
Figure 4.1 Topologie du réseau NSFNET.....	87
Figure 4.2 Nombre de ports en fonction des $k$ plus courts chemins .....	94
Figure 4.3 Temps d'exécution en fonction des $k$ plus courts chemins .....	94

Figure 4.4 Nombre de ports en fonction de la longueur de la liste taboue .....	95
Figure 4.5 Nombre de ports en fonction du nombre maximum d'itérations.....	96
Figure 4.6 Nombre de ports en fonction d'une demande uniforme pour TabouMRWA et BPHT (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $F = 2$ , $K = 120$ , $W = 4$ , $Tr =$ Uni (d)).....	101
Figure 4.7 Gain de TabouMRWA par rapport au BPHT en fonction de la demande uniforme (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $F = 2$ , $K = 120$ , $W = 4$ ) ....	102
Figure 4.8 Nombre de ports en fonction du nombre de longueurs d'onde par bande, $W$ , pour TabouMRWA et BPHT (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $F = 1$ , $K$ = 120, $Tr = \text{Rand5}$ ).....	103
Figure 4.9 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de longueurs d'onde par bande, $W$ .....	104
Figure 4.10 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de longueurs d'onde par bande et pour différentes valeurs de $F$ .....	106
Figure 4.11 Nombre de ports en fonction du nombre de longueurs d'onde par fibre $F$ , pour TabouMRWA et BPHT avec $K = 240/F$ (NSFNET, $k = 5$ , $LT = 20$ , $M =$ 1000, $W = 4$ , $Tr = \text{Rand20}$ ) .....	107
Figure 4.12 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de fibres par lien (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $F \times K = 240$ , $W = 4$ , $Tr =$ Rand20).....	107
Figure 4.13 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de longueurs d'onde par bande et pour différentes valeurs de $F$ et de $Tr$ .....	109

## LISTE DES TABLEAUX

Tableau 4.1 Taille du réseau utilisé.....	86
Tableau 4.2 Niveaux des facteurs .....	89
Tableau 4.3 Expériences réalisées.....	92
Tableau 4.4 Valeurs par défaut des facteurs (Étape 1).....	92
Tableau 4.5 Valeur des facteurs constants (Expérience 2.1) .....	97
Tableau 4.6 Nombre de ports pour $k$ et LT variables (NSFNET, $F=2$ , $K=120$ , $W=4$ , $Tr = Rand20$ ).....	98
Tableau 4.7 Temps d'exécution (en secondes) pour $k$ et LT variables (NSFNET, $F=2$ , $K=120$ , $W=4$ , $Tr = Rand20$ ) .....	98
Tableau 4.8 Valeur des facteurs secondaires pour TabouMRWA (Étape 3) .....	100
Tableau 4.9 Valeur des facteurs primaires constants (Expérience 3.1) .....	100
Tableau 4.10 Valeur des facteurs primaires constants (Expérience 3.2) .....	102
Tableau 4.11 Nombre de ports en fonction du nombre de longueurs d'onde par bande, $W$ , pour TabouMRWA et BPHT avec $F = 1$ et $K = 240$ (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $Tr = Rand20$ ).....	105
Tableau 4.12 Nombre de ports en fonction du nombre de longueurs d'onde par bande, $W$ , pour TabouMRWA et BPHT avec $F = 2$ et $K = 120$ (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $Tr = Rand20$ ) .....	105
Tableau 4.13 Nombre de ports en fonction du nombre de longueurs d'onde par bande, $W$ , pour TabouMRWA et BPHT avec $F = 4$ et $K = 60$ (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $Tr = Rand20$ ) .....	105
Tableau 4.14 Nombre de ports en fonction du nombre de longueurs d'onde par bande, $W$ , pour TabouMRWA et BPHT avec $F = 1$ et $Tr = Rand5$ (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $K = 60$ ) .....	108
Tableau 4.15 Nombre de ports en fonction du nombre de longueurs d'onde par bande, $W$ , pour TabouMRWA et BPHT avec $F = 2$ et $Tr = 2 \times Rand5$ (NSFNET, $k = 5$ , $LT = 20$ , $M = 1000$ , $K = 60$ ) .....	109

Tableau 4.16 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  
 $W$ , pour TabouMRWA et BPHT avec  $F = 4$  et  $Tr = 4 \times \text{Rand5}$   
(NSFNET,  $k = 5$ , LT = 20, M = 1000, K = 60) ..... 109

## SIGLES ET ABREVIATIONS

<b>BPHT</b>	Balanced Path Routing and Heavy Traffic First (Routage équilibré et trafic élevé en premier)
<b>BTF</b>	Band To Fiber multiplexer (Dispositif pour le multiplexage de bandes d'onde dans une fibre)
<b>BTW</b>	Band To Wavelength demultiplexer (Dispositif pour le démultiplexage d'une bande en longueurs d'onde)
<b>FTB</b>	Fiber To Band demultiplexer (Dispositif pour le démultiplexage d'une fibre en bandes)
<b>ILP</b>	Integer Linear Program (Programme en nombres entiers)
<b>MG-OXC</b>	Multigranular Optical Cross Connect (Brasseur optique multigranulaire)
<b>OXC</b>	Optical Cross Connect (Brasseur optique)
<b>RWA</b>	Routing and Wavelength Assignment (Routage et affectation de longueurs d'onde)
<b>WDM</b>	Wavelength Division Multiplexing (Multiplexage de longueurs d'onde)
<b>WTB</b>	Wavelength To Band multiplexer (Dispositif pour le multiplexage de longueurs d'onde dans une bande)

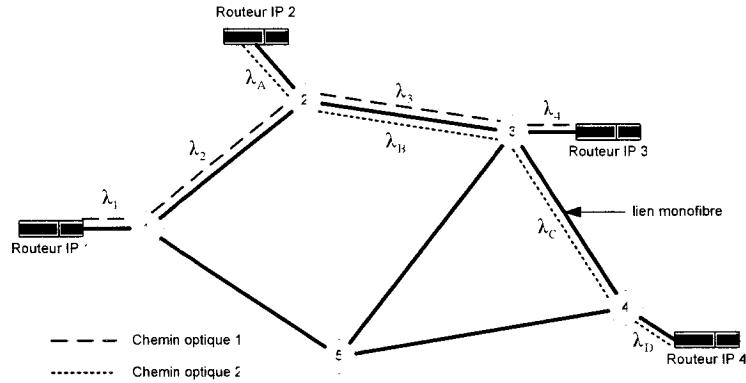
# CHAPITRE I

## INTRODUCTION

Au cours de la dernière décennie, le secteur des télécommunications a connu un essor important, dû notamment au succès de l'Internet et à l'avènement d'une multitude d'applications de tout genre. Le nombre grandissant d'utilisateurs, de nouveaux services, de nouvelles applications ne font qu'augmenter le besoin en bande passante, en particulier au niveau des réseaux d'infrastructure. Dans ce contexte, les réseaux optiques employant la technologie WDM (*Wavelength Division Multiplexing*), qui consiste à transmettre simultanément des données sur des longueurs d'onde différentes à l'intérieur d'une fibre, gagnent de plus en plus d'intérêt. Ils se présentent comme une alternative économique permettant de répondre à cette demande grandissante. Récemment, une hiérarchisation de ces réseaux a retenu l'attention de la communauté scientifique promettant de réduire davantage les coûts et de répondre au problème d'évolutivité dans ces réseaux, ce qui fait l'objet du présent mémoire. Dans ce chapitre d'introduction, après avoir présenté quelques définitions et les concepts de base liés aux réseaux WDM, nous exposerons la problématique liée à la hiérarchisation de ces réseaux, nous spécifierons ensuite les objectifs de notre recherche et enfin nous terminerons en indiquant le plan du mémoire.

### 1.1. Définitions et concepts de base

Un réseau WDM se compose de nœuds optiques (multiplexeurs optiques d'insertion et d'extraction de longueurs d'onde, brasseurs optiques, etc.) reliés ensemble par l'intermédiaire de liens optiques qui peuvent être monofibres ou multifibres. Le schéma d'interconnexion physique des nœuds optiques représente la *topologie physique* du réseau [27]. Elle peut être représentée par un graphe orienté dont les sommets sont les nœuds optiques du réseau et les arcs les liens optiques.



**Figure 1.1. Réseau WDM à routage de longueur d'onde**

Dans un réseau WDM dit à *routage de longueur d'onde*, les utilisateurs (des multiplexeurs SONET/SDH, des routeurs IP, des commutateurs ATM, etc.) communiquent entre eux via des canaux de communication tout optiques appelés *chemins optiques*. Un chemin optique est une séquence de liens de l'émetteur au récepteur où une longueur d'onde est assignée sur chacun des liens de celle-ci [4]. En l'absence d'un *convertisseur de longueur d'onde* (dispositif d'entrée-sortie qui convertit la longueur d'onde d'un signal optique arrivant à son port d'entrée en une longueur d'onde différente à son port de sortie) au niveau des nœuds du chemin, le chemin optique doit employer la même longueur d'onde sur tous les liens de la source à la destination. Il s'agit de la *contrainte de continuité de longueur d'onde*. À la Figure 1.1, cela revient à dire qu'il faut que  $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$  pour le chemin optique 1. De plus, à tous les chemins optiques partageant la même fibre doivent être assignés des longueurs d'onde distinctes. À la Figure 1.1, cela revient à dire qu'il faut que  $\lambda_3 \neq \lambda_B$  pour le lien (2, 3).

L'établissement d'un chemin optique résulte d'une demande de connexion optique entre deux utilisateurs et implique la détermination d'une route (une séquence de liens physiques) tout en réservant une longueur d'onde particulière sur chacun des liens de ce chemin. Ainsi, nous devons traiter à la fois le routage (choisir un chemin approprié) et l'assignation de longueur d'onde (allouer une longueur d'onde disponible pour le chemin). Le problème résultant est désigné sous le nom de *routage et d'assignation de longueur d'onde (RWA)*.

Les différentes variantes du problème RWA peuvent être classées en deux grandes catégories: un *RWA statique* et un *RWA dynamique*. Le RWA statique correspond à un modèle de *trafic statique* où toutes les demandes de connexions sont connues à l'avance. Le RWA dynamique correspond à un modèle de *trafic dynamique* où un chemin optique est établi sur demande (i.e. pour chaque requête de connexion lorsqu'elle arrive), et ce chemin optique est libéré après un temps fini. Chaque fois qu'une demande est faite, un algorithme doit être exécuté en temps réel pour déterminer si la demande peut être satisfaite, et, si oui, effectuer le routage et l'assignation de longueur d'onde. Dans le cas où la demande ne peut pas être satisfaite, elle est bloquée. Souvent, dans ce contexte, on parle de *probabilité de blocage* des demandes de connexion optique [24].

Les avancées rapides de la technologie WDM permettant le multiplexage de centaines de longueurs d'onde par fibre ont conduit à une augmentation importante de la taille des brasseurs optiques, du coût et de la difficulté à contrôler et gérer de tels dispositifs, en particulier dans les réseaux multifibres. Le concept de *multigranularité* dans les réseaux optiques permet de commuter différentes granularités à l'intérieur du réseau. Il a été proposé pour réduire la complexité et le coût à des niveaux raisonnables. Les différentes granularités correspondent à la longueur d'onde, la bande d'onde et la fibre. Trois types de commutation sont alors possibles dans ces réseaux : la *commutation de longueur d'onde*, la *commutation de bande d'onde* et la *commutation de fibre*.

La commutation de longueur d'onde consiste à commuter individuellement une longueur d'onde de son port d'entrée vers un port de sortie. La commutation de bande d'onde consiste à regrouper plusieurs longueurs d'onde en une bande et à commuter cette bande optiquement de son port d'entrée vers un port de sortie. Une bande d'onde peut être définie comme un ensemble de longueurs d'onde contiguës ou non. La commutation de fibre consiste à commuter toutes les longueurs d'onde d'une fibre en entrée vers une autre fibre en sortie, sans recourir au démultiplexage du signal optique contenant les longueurs d'onde.

L'idée centrale de la multigranularité dans les réseaux optiques est de regrouper le plus possible le trafic dans une fibre ou dans une bande d'onde afin de commuter la fibre ou la bande en utilisant deux ports uniquement (un port d'entrée et un port de sortie) au niveau du brasseur optique, ce qui évite d'aller jusqu'au niveau de la longueur d'onde. Les conséquences devraient être une réduction du coût global du réseau en diminuant la taille des brasseurs optiques (nombre de ports), facteur prépondérant dans le coût global de ceux-ci, et la taille de leur matrice de commutation. Une simplification de la gestion dans le réseau et une amélioration de l'évolutivité pourraient également en résulter [8].

## 1.2. Éléments de problématique

Le problème RWA dans les réseaux à routage de longueur d'onde a déjà été prouvé NP-complet. Dans les réseaux multigranulaires, plus d'états doivent être considérés, ainsi que de nouvelles contraintes. Par conséquent, le problème est encore plus difficile à résoudre dans ces réseaux. Un certain nombre de questions doivent être posées lorsqu'on aborde le problème de routage dans de tels réseaux. Notamment,

1. Comment regrouper les longueurs d'onde pour former des bandes à l'intérieur d'une fibre ?

D'abord, le nombre de bandes dans une fibre peut être fixe ou variable. Ensuite, le nombre de longueurs d'onde dans une bande d'onde peut être fixe ou variable. Enfin, pour un nombre fixé et donné de longueurs d'onde dans une bande, l'ensemble des longueurs d'onde dans la bande peut être prédéterminé, ou peut être adaptif, c'est-à-dire dynamiquement configuré [8].

2. Quelles stratégies de regroupement des chemins optiques utiliser ?

Il existe quatre stratégies de regroupement des chemins optiques en bande d'onde ou fibre.

- a) On peut regrouper uniquement les chemins optiques partageant la même paire source – destination.

- b) On peut regrouper uniquement les chemins optiques partageant la même source mais ayant des destinations différentes.
- c) On peut regrouper uniquement les chemins optiques partageant la même destination mais ayant des sources différentes.
- d) On peut regrouper uniquement les chemins optiques ayant un sous-chemin en commun.

Un algorithme de routage peut utiliser une de ces stratégies ou une combinaison des quatre.

3. Où doit-on regrouper et dégrouper les chemins optiques?
4. Comment sélectionner les routes des chemins optiques dans un tel contexte?
5. Comment assigner les bandes d'onde aux groupes de chemins optiques pour la commutation de bandes?

Plusieurs algorithmes ont été proposés dans la littérature pour répondre à ces questions dans le cas d'un modèle de trafic statique. Kim et al. [18] ont proposé un algorithme pour réduire le nombre de ports des commutateurs multigranulaires en essayant de regrouper uniquement les chemins optiques ayant la même destination. Il est possible de faire mieux que cet algorithme en explorant plus de possibilités de regroupement des chemins optiques. Mouftah et al. [3] ont proposé un algorithme pour dimensionner un réseau optique de façon à accommoder plus de trafic à long terme. Cao et al. [5] ont développé un algorithme, le BPHT (Balanced Path Routing and Heavy Traffic First), capable de minimiser le nombre de ports globalement dans le réseau et le nombre de ports du commutateur ayant le plus de ports parmi tous les commutateurs du réseau. Cet algorithme cherche à réduire le nombre de ports en considérant surtout l'affectation de longueurs d'onde aux chemins optiques. La sélection des routes utilisée dans cet algorithme ne cherche pas à regrouper les chemins optiques. Des améliorations peuvent donc être apportées au problème traité par cet algorithme.

### 1.3. Objectifs de recherche

L'objectif principal de ce mémoire est de concevoir un algorithme de routage statique dans les réseaux optiques multigranulaires qui permettrait de réduire le nombre de ports globalement dans le réseau, tout en satisfaisant un ensemble de demandes de connexion. De manière plus spécifique, le mémoire vise à :

- analyser les meilleurs algorithmes de routage multigranulaire statique disponibles dans la littérature afin de déceler leurs forces et leurs faiblesses ;
- proposer et implémenter un algorithme de routage statique réduisant le nombre de ports globalement dans le réseau ;
- évaluer les performances de cet algorithme et le comparer aux meilleurs algorithmes existants dans la littérature.

### 1.4. Esquisse méthodologique

Afin d'atteindre nos objectifs, nous commencerons par présenter en détail les algorithmes de routage multigranulaire proposés dans la littérature et une discussion des résultats obtenus par ces algorithmes, en particulier celui de Cao et al. [5]. Nous identifierons ensuite des facteurs d'amélioration de ce dernier. Par la suite, nous proposerons un algorithme de routage qui réduirait le nombre de ports globalement dans le réseau. Pour développer notre algorithme, nous considérerons que le nombre de bandes d'onde dans une fibre est fixe, que le nombre de longueurs d'onde dans une bande d'onde est fixe, que les longueurs d'onde d'une bande sont contigües et finalement que l'ensemble de ces longueurs d'onde est prédéterminé. De plus, les quatre stratégies de regroupement des chemins optiques seront utilisées. Pour finir, nous tiendrons compte de la contrainte de continuité au niveau des chemins optiques.

### 1.5. Plan du mémoire

Ce mémoire comprend cinq chapitres repartis de la façon suivante. Le chapitre 2 examinera plus en détail le concept de la multigranularité et présentera certains travaux déjà réalisés pour résoudre le problème de routage dans les réseaux optiques

multigranulaires. Le chapitre 3 exposera l'algorithme que nous utiliserons pour résoudre le problème. Le chapitre suivant, le quatrième, nous permettra de présenter l'implémentation de l'algorithme que nous proposons ainsi que les résultats obtenus. Dans le dernier chapitre, nous conclurons ce mémoire en présentant une synthèse des travaux que nous avons réalisés, les limites de ce dernier et des indications en vue de recherche future.

## CHAPITRE II

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

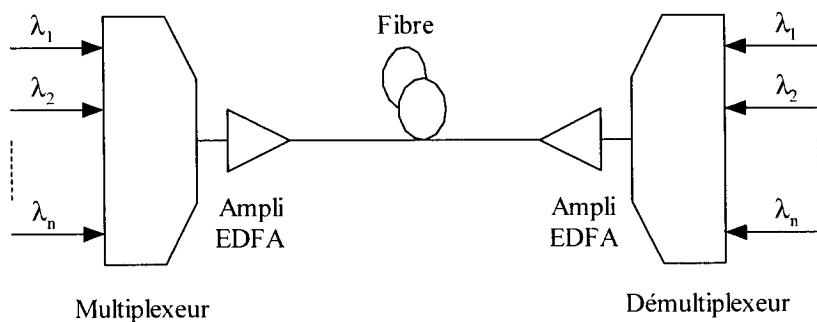
Les réseaux optiques utilisant la technologie WDM permettent de répondre au besoin grandissant de bande passante dû au nombre croissant d'utilisateurs, de nouveaux services, de nouvelles applications. Cependant, une augmentation importante du coût de ces réseaux et de la difficulté à les gérer a été récemment constatée. Ce fait s'explique par les avancées de la technologie WDM qui permet d'insérer plusieurs longueurs d'onde dans une fibre. Une hiérarchisation des réseaux WDM a donc été proposée pour en réduire de manière significative le coût et en faciliter la gestion. Afin de tirer avantage de la hiérarchisation, de nouvelles architectures pour les commutateurs optiques ont été proposées et de nouveaux algorithmes de routage et d'affectation de longueurs d'onde ont été développés. Dans ce chapitre, nous présenterons d'abord les réseaux à routage de longueurs d'onde où celles-ci sont traitées individuellement. Nous mettrons l'accent sur la technologie WDM, le routage optique et la gestion des réseaux optiques. Ensuite, nous présenterons les réseaux hiérarchiques à trois niveaux de granularité, en mettant l'accent sur la motivation et les objectifs de la multigranularité, le routage optique, la gestion des pannes et quelques algorithmes de routage et d'affectation de longueurs d'onde. Nous terminerons par une esquisse des réseaux hiérarchiques à deux niveaux de granularité.

### 2.1. Réseaux à routage de longueurs d'onde

Nous présenterons dans cette section les aspects importants des réseaux WDM à routage de longueurs d'onde, à savoir la technologie WDM, le routage optique et la gestion des réseaux.

### 2.1.1. La technologie WDM

La technologie WDM est essentiellement une technique de multiplexage de longueurs d'onde, au même titre que le multiplexage fréquentiel. L'idée est de transmettre des données simultanément à des longueurs d'onde porteuses différentes sur une fibre. La capacité résultante est l'agrégation des signaux optiques d'entrée, chacune des longueurs d'onde ou canaux optiques ayant sa propre bande passante. Le nombre de canaux optiques capables d'être injectés sur une fibre dépassait déjà la centaine en l'année 2000. La Figure 2.1 montre une liaison WDM point à point typique [1].



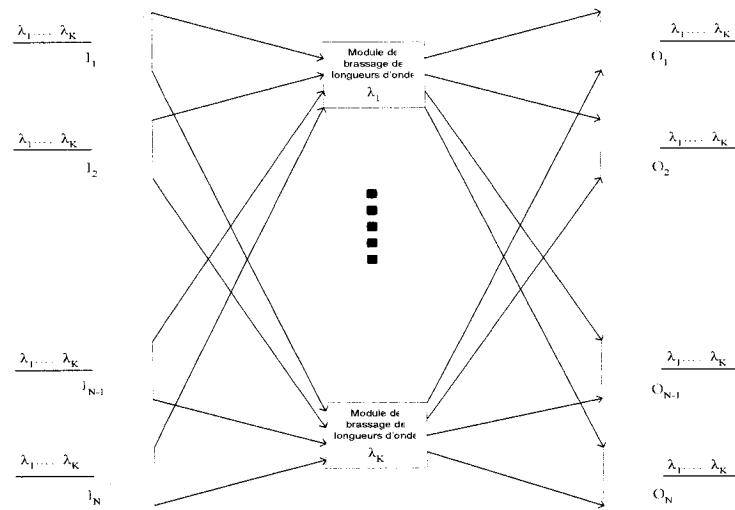
**Figure 2.1 Liaison WDM point à point**

L'amplificateur de fibre dopée à l'erbium (*Erbium Doped Fiber Amplifier : EDFA*) est l'un des éléments clés du WDM et ne fait qu'amplifier les signaux sur la fibre. Il ne régénère pas ces derniers. Les signaux peuvent être transmis entre 600 et des milliers de kilomètres sans avoir à être régénérés.

Une liaison WDM est en général bidirectionnelle, c'est-à-dire que les données sont transmises dans les deux sens sur la liaison. Pour rendre la liaison bidirectionnelle, deux fibres sont souvent utilisées, une pour chacune des directions. En plus, une liaison WDM peut être composée d'une ou de plusieurs fibres dans chacun des deux sens.

Un réseau WDM à routage de longueurs d'onde est l'interconnexion de liaisons WDM point à point par des multiplexeurs à insertion/extraction optiques (*Optical Add Drop Multiplexing : OADM*), ou des brasseurs optiques (*Optical Cross-Connect : OXCs*) capables de router les canaux optiques. L'OADM est un dispositif servant à insérer ou à

extraire sélectivement une ou plusieurs longueurs d'onde d'un signal WDM sans recourir à une conversion optoélectronique. Les OADMs sont typiquement déployés dans des technologies linéaires ou en anneau. L'OXC permet de commuter individuellement les longueurs d'onde d'un port d'entrée vers n'importe quel port de sortie. Sa matrice de commutation peut être reconfigurée sur demande, sous contrôle électronique.



**Figure 2.2 Brasseur de longueurs d'onde**

La Figure 2.2 illustre un brasseur optique ayant N ports d'entrée et N ports de sortie. Toutes les fibres ont le même nombre K de longueurs d'onde. L'OXC comprend alors K brasseurs de longueurs d'onde  $N \times N$  pour un total de  $K.N$  ports à l'entrée. Si la technologie MEMS (*MicroElectroMechanical Systems*) où des micros miroirs sont utilisés pour diriger la lumière d'une entrée vers une sortie est employée, le nombre total de miroirs est  $K.N^2$ . Les réseaux à routage de longueurs d'onde utilisent souvent des OXCs qui permettent le déploiement de topologies quelconques (maillées).

### 2.1.2. Routage optique

Un réseau WDM fournit des chemins optiques entre paires d'équipements supérieurs, tels que les routeurs IP, les commutateurs ATM, les multiplexeurs SONET/SDH. Une caractéristique unique de ces réseaux est la relation étroite entre le

routage et la sélection de longueur d'onde. Le problème résultant, le RWA, est classé en deux grandes catégories : le RWA statique et le RWA dynamique, comme mentionné dans le chapitre précédent.

Dans le cas du RWA statique, l'objectif typique est de minimiser le nombre de longueurs d'onde nécessaires pour établir un certain ensemble de chemins optiques pour une topologie physique donnée. Alternativement, le problème dual est de maximiser le nombre de connexions qui peuvent être établies (ou de minimiser le blocage) pour un nombre donné de longueurs d'onde et un ensemble donné de requêtes de connexion. Le problème RWA statique peut être formulé comme un programme linéaire en nombres entiers mixte (*Mixed-Integer Linear Program : MILP*) qui est connu comme étant NP-complet. Pour le rendre plus soluble, le routage et l'assignation de longueurs d'onde sont traités séparément. Le routage est souvent réalisé en utilisant des heuristiques, surtout pour résoudre le problème dans le cas de grands réseaux. Une approche pour résoudre le problème d'affectation de longueurs d'onde sous la contrainte de continuité de longueur d'onde est de le formuler comme un problème de coloriage de graphe. Ce problème est NP-complet. Le nombre minimum de couleurs nécessaires pour colorier le graphe est difficile à déterminer. Cependant, il existe des algorithmes efficaces de coloriage de graphe séquentiel qui sont optimaux pour le nombre de couleurs utilisées.

Une autre méthode de résolution du problème RWA statique est la recherche locale [1] qui est une classe de métaheuristiques basée sur la notion de voisinage. Le principe de ce type de méthode est simple. Étant donné un problème d'optimisation  $(S, f)$  défini par l'ensemble  $S$  des configurations et la fonction de coût  $f$ , on définit une fonction de voisinage et un mécanisme de parcours du voisinage. Pour commencer, un algorithme de recherche locale engendre tout d'abord une configuration initiale  $s_0$ , puis il effectue une suite d'itérations. À chaque itération, une configuration  $s'$  est choisie dans le voisinage de la configuration courante. Le processus s'arrête et retourne la meilleure configuration trouvée quand le critère d'arrêt est atteint.

La manière dont le voisinage de la solution courante est défini dans un algorithme de recherche locale est cruciale. Hyttia [20] a proposé deux définitions du

voisinage pour les réseaux optiques. La première définition consiste à changer le chemin d'une connexion, toutes les longueurs d'onde pouvant être différentes. Dans la pratique, lorsqu'une connexion est routée à nouveau, toutes les longueurs d'onde sont réaffectées tout en essayant de minimiser le nombre maximum de connexions utilisant la même longueur d'onde sur n'importe quel lien. Ce nombre définit la multiplicité de chaque lien. La deuxième définition correspond au cas où seulement une connexion possède soit un nouveau chemin, soit une nouvelle longueur d'onde, soit les deux. Toutes les autres connexions demeurent inchangées, c'est-à-dire qu'aucune affectation de longueurs d'onde n'est effectuée pour ces connexions.

Dans le cas du RWA dynamique, l'objectif est d'établir des chemins optiques et d'affecter des longueurs d'onde d'une manière qui minimise le nombre de connexions bloquées, ou qui maximise le nombre de connexions établies dans le réseau à n'importe quel moment. Une approche typique pour concevoir des algorithmes RWA efficaces dans un environnement dynamique est de découpler le problème tout comme dans le cas du RWA statique en deux sous problèmes séparés : le problème de routage et le problème d'affectation de longueurs d'onde. Pour le sous-problème de routage, trois approches de base peuvent être trouvées dans la littérature : le routage fixe (*Fixed routing*) dans lequel un chemin simple hors ligne est affecté à chaque paire de nœuds source/destination, le routage alternatif fixe (*Fixed-alternate routing*) où un nombre  $k$ ,  $k > 1$ , de chemins disjoints sont calculés et commandés hors line pour chaque paire de nœuds source/destination et le routage adaptif (*Adaptive routing*) qui peut calculer un chemin simple, ou un certain nombre de chemins alternatifs au moment de la demande. Dans un algorithme adaptatif, la route de la source à la destination est choisie dynamiquement, dépendamment de l'état du réseau. Des heuristiques peuvent aussi être utilisées pour résoudre le problème de routage ainsi que d'autres approches de routage tel que le routage résistant aux pannes (*Fault-tolerant routing*). Pour le problème d'affectation de longueurs d'onde, des heuristiques ont été proposées tel que le « First-Fit » dans lequel toutes les longueurs d'onde sont numérotées. Lorsque des longueurs d'onde disponibles sont recherchées, la longueur d'onde ayant un numéro plus faible est

considérée avant une longueur d'onde de numéro plus élevé. La première longueur d'onde disponible est alors choisie.

### 2.1.3. Gestion des réseaux WDM

Dans un réseau WDM, un mécanisme de contrôle est nécessaire pour établir et libérer les connexions optiques, pour gérer les liaisons et les pannes. Le GMPLS (*Generalized MultiProtocol Label Switching*) définie par l'IETF est une suite de protocoles qui fournit ce mécanisme.

Le GMPLS étend le MPLS (*MultiProtocol Label Switching*) adopté également par l'IETF. Le MPLS est à la base un modèle hybride associant les avantages de la transmission de paquets fondée sur la commutation de couche 2 avec ceux du routage de couche 3. Dans le MPLS, les paquets sont commutés le long d'un chemin prédéfini, appelé chemin commuté d'étiquettes (*LSP : Label Switch Path*), en se basant sur des étiquettes insérées en avant de l'entête des paquets IP au niveau d'un routeur périphérique commutateur d'étiquettes (*LER : Label Edge Router*) à l'entrée du réseau MPLS. Le LER classifie le trafic basé sur des classes d'équivalence (*FEC : Forwarding Equivalence Classes*). Celles-ci peuvent être considérées comme un groupe de paquets IP transmis de la même manière, sur le même chemin et avec le même traitement appliqué à la transmission. Des routeurs commutateurs d'étiquettes (*LSR : Label Switch Router*) le long du chemin effectuent l'échange des étiquettes et à la sortie du réseau MPLS, un LER enlève l'étiquette de chacun des paquets avant de les délivrer.

Le GMPLS ne se limite pas aux dispositifs permettant la commutation de paquets. Il contient des fonctions de signalisation et de routage pour les dispositifs pouvant commuter dans n'importe lequel de ces domaines : paquet, temps, longueur d'onde, et fibre. Il permet l'approvisionnement automatique de bout en bout des connexions, la gestion des ressources réseaux, la gestion des pannes. Il peut être divisé en trois parties : le routage, la signalisation et la gestion des liens. Les interfaces de contrôle sont les suivantes :

- PSC (*Packet Switch Capable*) : interface capable de commuter suivant l'entête d'un paquet;
- L2SC (*Layer-2 Switch Capable*) : interface capable de commuter suivant l'entête d'une trame;
- TDMC (*Time Division Multiplexing Capable*) : interface capable de commuter suivant un créneau de temps dans un cycle répétitif;
- LSC (*Lambda Switch Capable*) : interface capable de commuter suivant la longueur d'onde reçue;
- FSC (*Fiber Switch Capable*) : interface capable de commuter suivant le port d'entrée d'une fibre (il s'agit de la commutation spatiale de fibres).

Une connexion peut être établie uniquement entre interfaces de même type. Elle est dénommée G-LSP dans le contexte du GMPLS. Un L-LSP est donc un G-LSP utilisant des interfaces LSC. De même, un F-LSP correspond à un G-LSP utilisant des interfaces FSC. Les nœuds dans les réseaux optiques qui supportent la signalisation GMPLS sont définis comme étant des G-LSRs. Une hiérarchie peut être construite si une interface est capable de multiplexer plusieurs G-LSPs de la même technologie. Cette hiérarchie conduit à une agrégation des flots de trafic. L'agrégation est également possible entre interfaces de types différents. Une liste de la hiérarchie d'interfaces du plus élevé au moins élevé est comme suit : FSC, LSC, TDM, L2SC et PSC. Des G-LSPs de niveau inférieur (de même niveau) peuvent être groupés à l'intérieur d'un G-LSP de niveau supérieur. Les nœuds intermédiaires ne voient que le G-LSP supérieur. Ils ne maintiennent pas d'états de transmission pour les G-LSPs internes.

Dans un réseau WDM, la panne d'un élément de réseau (un lien ou un nœud) peut causer la panne de plusieurs canaux optiques, menant à de grandes pertes de données (et de revenu). Dans le cas du WDM, c'est extrêmement critique à cause des grosses quantités de trafic transportées. Il existe deux méthodes pour faire face à une panne du réseau : la méthode proactive et la méthode réactive.

La méthode proactive consiste à réserver à l'avance des ressources de sauvegarde dédiées pour protéger une connexion établie. Il existe trois approches de protection dans

ce contexte: la protection 1+1, 1:1 et 1:N ( $N > 1$ ). Pour la protection 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. Dans le cas de la protection 1:1, un chemin de secours disjoint (en arc) est réservé pour chaque chemin principal. Les données ne sont pas envoyées simultanément sur les deux chemins. Le chemin secondaire n'est utilisé qu'en cas de panne du chemin principal. Dans le cas de la protection 1:N,  $N$  chemins principaux partagent un même chemin de secours. Si plusieurs chemins principaux tombent en panne, seul le trafic de plus haute priorité est acheminé sur le chemin de secours.

La méthode réactive est une méthode de restauration dynamique. Elle consiste à trouver des solutions alternatives selon les ressources disponibles dans le réseau lorsqu'une panne survient. On distingue la restauration de chemin et la restauration de lien. Dans le cas de la restauration de chemin, un chemin alternatif est trouvé entre l'origine et la destination de la connexion affectée par la panne. La restauration de lien rétablit la communication autour du lien en défaillance. Les nœuds source et destination de la connexion ne sont pas informés de la panne.

## 2.2. Réseaux multigranulaires à trois niveaux de granularité

Pour bien comprendre la multigranularité dans les réseaux optiques, il est important d'en connaître la motivation et les objectifs. Nous commencerons donc la section en présentant la motivation et les objectifs de la multigranularité. Nous présenterons par la suite deux architectures de commutateurs multigranulaires et les extensions apportées au GMPLS.

### 2.2.1. Motivation et objectifs de la multigranularité

Comme mentionné au chapitre précédent, la commutation multigranulaire consiste à commuter simultanément différentes granularités à l'intérieur d'un réseau optique donné. Les différents granularités correspondent à la longueur d'onde, la bande d'onde et la fibre. L'idée centrale du concept de la multigranularité est de regrouper le

plus que possible le trafic dans une fibre ou dans une bande d'onde de façon à commutier la fibre ou la bande en utilisant un port au niveau du commutateur, ce qui évite d'aller jusqu'au niveau de la longueur d'onde. La motivation à la base de ce concept est l'augmentation substantielle de la taille des brasseurs optiques en terme de nombre de ports résultant en une augmentation de leur coût et de la difficulté à les gérer. Cette augmentation est due particulièrement aux avancées de la technologie WDM permettant le multiplexage de centaines de longueurs d'onde sur une fibre et est encore plus importante dans les réseaux multifibres. De plus, à partir d'une certaine taille requise du brasseur de longueurs d'onde, des contraintes technologiques (actuelles) empêchent la réalisation même du dispositif. On se trouve alors confronté à une impasse. L'objectif principal est la réduction du coût du réseau en réduisant le nombre de ports et la matrice de commutation (nombre de miroirs en utilisant la technologie MEMS) des nœuds de commutation, la taille des nœuds étant un facteur prépondérant dans le coût global du réseau.

Pour illustrer la réduction du nombre de ports en utilisant la commutation multigranulaire, prenons 10 fibres arrivant à un nœud optique ayant le même nombre de ports en entrée et en sortie. Chacune des fibres comprend 100 longueurs d'onde réparties en 20 bandes, ce qui implique que chaque bande comporte 5 longueurs d'onde. Une longueur d'onde est insérée dans le réseau à partir de ce nœud et une longueur d'onde en est extraite. Dans le cas des réseaux à routage de longueurs d'onde, les longueurs d'onde d'une fibre en entrée sont commutées individuellement et doivent par conséquent être toutes démultiplexées. Le nombre de ports requis en entrée s'élève donc à 1001, dont 1000 pour les longueurs d'onde incidentes et 1 pour la longueur d'onde à insérer. Par contre, si la commutation multigranulaire est utilisée, seulement les 20 bandes d'onde de la fibre contenant la longueur d'onde à extraire sont démultiplexées et seulement les 5 longueurs d'onde de la bande contenant la longueur d'onde à extraire parmi les 20 bandes sont démultiplexées. Il faut donc 11 ports pour les 10 fibres en entrée et 1 fibre pour l'insertion éventuelle du signal de toute une fibre. Il faut ensuite 21 ports pour le démultiplexage des 20 bandes et l'insertion éventuelle d'une bande d'onde. Pour finir, il

faut 6 ports pour le démultiplexage des 5 longueurs d'onde et l'insertion de la longueur d'onde. Le nombre total de ports requis pour le nœud s'élève à 38, soit près de 30 fois moins que le nombre de ports requis pour un nœud dans le cas des réseaux à routage de longueurs d'onde.

En plus de la réduction du nombre de ports, la multigranularité offre d'autres avantages, notamment une simplification de la gestion dans le réseau en réduisant le nombre d'entités à gérer et en permettant une gestion indépendante et hiérarchique des informations relatives aux bandes et aux longueurs d'onde. La réduction de la taille de la matrice de commutation améliore l'évolutivité dans les réseaux et diminue la complexité à contrôler la matrice elle-même et à approvisionner les connexions optiques.

### 2.2.2. Commutateurs multigranulaires

La multigranularité dans un réseau implique la présence de commutateurs capables de commuter, d'insérer et d'extraire du trafic aux différentes granularités. Le sigle MG-OXC sera utilisé pour désigner de tels commutateurs. Deux architectures ont été principalement proposées dans la littérature. La Figure 2.3 décrit l'architecture proposée par Mouftah et al. [4].

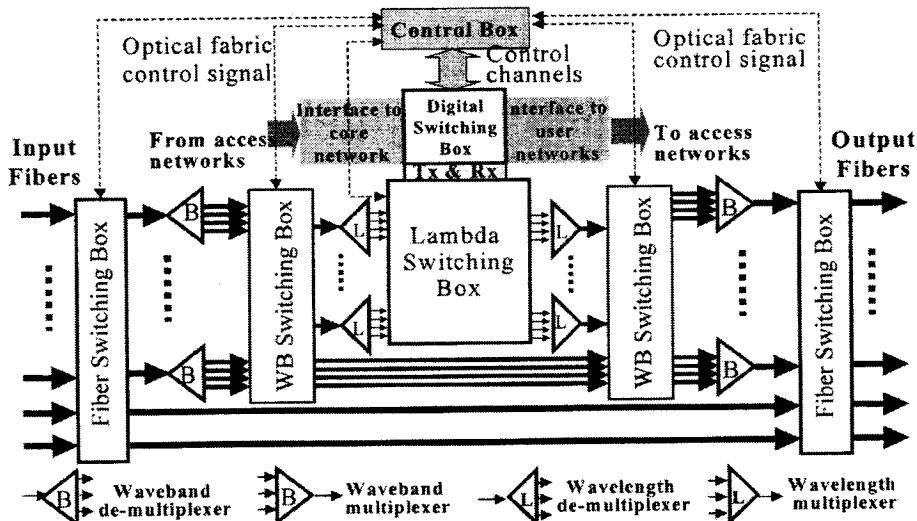


Figure 2.3 Architecture MG-OXC proposée par Mouftah et al.

Dans cette architecture, les boîtes de commutation de fibres et de bandes d'onde qui se trouvent du côté gauche jouent le rôle de sélecteurs pour les fibres et les bandes d'onde en entrée. La boîte de commutation de fibres sélectionne les fibres qui seront dirigées vers la boîte de sélection de bandes d'onde et celles qui seront commutées spatialement à la granularité fibre. De même, la boîte de commutation de bandes d'onde sélectionne les bandes d'onde qui seront séparées en longueurs d'onde individuelles et celles qui seront commutées spatialement à la granularité bande d'onde. Les boîtes de commutation de fibres et de bandes d'onde du côté droit accomplissent la commutation spatiale aux granularités fibre et bande d'onde respectivement. La boîte de commutation de fibres commute toute une fibre et la boîte de commutation de bandes d'onde commute toute une bande. La boîte du milieu est un brasseur de longueurs d'onde qui commute les longueurs d'onde individuellement.

Les ports d'entrée et de sortie sont affectés à l'un des trois types de commutation, soit la commutation de fibres, la commutation de bandes d'onde ou la commutation de longueurs d'onde. En d'autres termes, le trafic arrivant par exemple à un port affecté à la commutation de longueurs d'onde sera dirigé par la boîte de commutation de fibres de gauche vers la boîte de commutation de bandes d'onde de gauche qui, à son tour, dirigera le trafic vers le brasseur de longueurs d'onde où les longueurs d'onde individuelles seront commutées. En d'autres termes, les liens sont composés d'un nombre donné de fibres pour les différents types de commutation.

Pour finir, les mécanismes de signalisation sont générés à partir de la boîte de contrôle. Les paquets de contrôle générés peuvent être transmis *in-band*, c'est-à-dire en utilisant le réseau de transmission des données ou ils peuvent être transmis *out-of-band*, c'est-à-dire en utilisant un réseau différent de celui utilisé pour transporter les données.

La Figure 2.4 illustre l'architecture de commutateur multigranulaire proposée par Cao et al. [5]. Elle est fonctionnellement semblable à celle décrite plus haut.

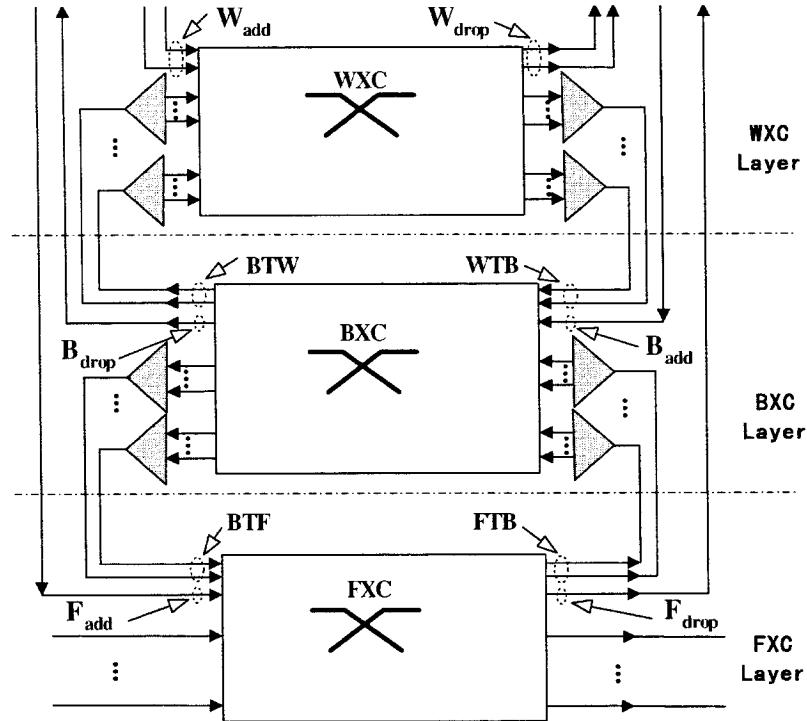


Figure 2.4 Architecture MG-OXC proposée par Cao et al.

Cette architecture comprend trois couches : la couche FXC, la couche BXC et la couche WXC. La couche WXC comprend un brasseur de longueurs d'onde, le WXC, qui est utilisé pour le by-pass, l'extraction et l'insertion de longueurs d'onde au niveau de cette couche. Elle comprend aussi des démultiplexeurs BTW qui séparent les bandes d'onde en longueurs d'onde individuelles, et des multiplexeurs WTB pour le multiplexage de longueurs d'onde en bande d'onde. La couche BXC se compose d'un brasseur de bandes d'onde pour le by-pass, l'insertion et l'extraction de bandes d'onde, de démultiplexeurs de fibres à bandes d'onde (FTB) et de multiplexeurs de bandes d'onde à fibre (BTF). La couche FXC comprend un brasseur de fibres qui sert au by-pass, à l'extraction et à l'insertion de fibres au niveau de cette couche.

Cette architecture permet une configuration (interconnexion) dynamique du commutateur multigranulaire. Elle est plus flexible que celle proposée par Mouftah et al. [4]. Aucun port en entrée n'est affecté à un type spécifique de commutation. Par exemple, au niveau de la couche FXC, du moment qu'il existe un port FTB libre, le

signal de n'importe quelle fibre en entrée peut être séparé en ses bandes d'onde individuelles. De même, au niveau de la couche BXC, le signal de n'importe quelle bande d'onde peut être séparé en longueurs d'onde individuelles du moment qu'il existe un port BTW libre. Cette architecture sera celle utilisée dans ce mémoire.

### 2.2.3. Extensions proposées au GMPLS

Le GMPLS supporte la commutation de bandes d'onde. Cependant, il considère cette dernière comme étant un cas spécial de commutation de longueurs d'onde. Dans le GMPLS, une bande d'onde est définie comme étant un ensemble contigu de longueurs d'onde pouvant être commutées ensemble vers une autre bande d'onde. Aucune interface de contrôle n'a été définie pour la commutation de bandes d'onde.

Dotaro et al. [26] ont proposé une nouvelle interface de contrôle, le WBSC (*Waveband Switch Capable Interface*), capable de commuter suivant une bande d'onde. Cette interface matérialise la réalité physique de la bande optique en tant qu'entité atomique ou granularité. La bande n'est plus vue dans ce cas comme étant un groupe nécessairement contigu de longueurs d'onde recevant le même traitement. Une nouvelle classe de LSP est également définie. Il s'agit du WB-LSP. La nouvelle hiérarchie se présente comme suit du plus élevé au moins élevé : FSC, WBSC, LSC, TDM, L2SC, PSC. Ainsi, des L-LSPs peuvent être combinés pour former un WB-LSP et des WB-LSPs peuvent être combinés pour former un F-LSP. Les F-LSPs agissent donc en tant que tunnels pour les WB-LSPs et les WB-LSPs en tant que tunnels pour les L-LSPs. Un F-LSP peut également agir comme tunnel directement pour des L-LSPs. Certains auteurs ont utilisé cette proposition dans leurs études des réseaux multigranulaires [3] - [5].

### 2.2.4. Routage optique

L'objectif principal de la multigranularité dans les réseaux optiques est de réduire la taille des commutateurs optiques en terme de nombre de ports. Le calcul du nombre de ports prend en compte au niveau de chacun des commutateurs du réseau les ports pour la commutation spatiale des fibres, des bandes d'onde et des longueurs

d'onde. Il prend en compte également les ports pour le multiplexage des bandes d'onde en fibre (signal composite occupant la bande passante d'une fibre), des longueurs d'onde en bande d'onde, et les ports pour la séparation du signal optique des fibres en bandes d'onde et des bandes d'onde en longueurs d'onde. L'algorithme de routage et d'affectation de longueurs d'onde doit prendre en considération la présence des commutateurs multigranulaires. Le nombre de ports utilisés au niveau d'un commutateur multigranulaire pour une connexion est maximal toutes les fois que le brasseur de longueurs d'onde est utilisé. Le routage doit chercher à regrouper les routes le plus que possible de façon à réduire l'utilisation du brasseur de longueurs d'onde. En effet, si deux connexions optiques entrant dans un nœud empruntent des liens différents à la sortie, les longueurs d'onde respectives devront être commutées individuellement et le brasseur de longueurs d'onde devra être utilisé. L'affectation des longueurs d'onde doit tenter de regrouper le plus que possible les longueurs d'onde des connexions optiques dans une même fibre et à l'intérieur d'une fibre dans une même bande d'onde.

Dans le cas de demandes de trafic statique, l'objectif du routage optique multigranulaire est de déterminer la taille des commutateurs optiques de façon à satisfaire les demandes de trafic tout en minimisant le nombre de ports globalement dans le réseau. Dans un réseau à routage de longueurs d'onde, le fait de minimiser le nombre de longueurs d'onde (en minimisant le nombre de sauts de longueurs d'onde) revient à minimiser le nombre de ports dans les commutateurs. Cependant, minimiser le nombre de ports dans un réseau multigranulaire n'équivaut pas nécessairement à minimiser le nombre de longueurs d'onde. En effet, parfois, pour réduire le nombre de ports, un chemin plus long (en terme de nombre de sauts) utilisant une longueur d'onde dans une bande peut être choisi bien qu'un chemin plus court ne permettant pas le groupement dans une bande existe. En fait, il doit exister un compromis entre le nombre requis de sauts et le nombre de ports [5].

Dans le cas de demandes de trafic dynamique, la taille des commutateurs est déterminée de façon à minimiser la probabilité de blocage des requêtes de connexion tout en minimisant le nombre de ports globalement dans le réseau. Les commutateurs

multigranulaires peuvent être reconfigurables. Dans ce contexte, un chemin optique répondant à une demande de connexion peut être établi à travers un chemin de bande d'onde (WB-LSP) ou de fibre (F-LSP) déjà établi. L'insertion du chemin optique se fait à partir de n'importe quel nœud le long du WB-LSP ou F-LSP. Une telle opération, cependant, provoque une interruption de service pour toutes les connexions optiques existantes utilisant ce WB-LSP ou F-LSP puisque ce dernier devra être établi à nouveau avec le nouveau chemin optique le traversant. Les algorithmes de routage utilisant la reconfiguration doivent minimiser le nombre de reconfigurations, le nombre de connexions affectées par une reconfiguration quelconque et le temps moyen pour une reconfiguration. D'autres algorithmes évitent la reconfiguration. Il existe quatre façons d'établir de nouvelles connexions sans altérer les connexions existantes :

- 1) Router le trafic à travers un WB-LSP ou F-LSP existant partageant le même nœud source et le même nœud destination;
- 2) Établir un nouveau chemin optique (*lightpath*) de la source à la destination;
- 3) Router le trafic à travers deux WB-LSPs ou F-LSPs ou plus dont le premier partage le même nœud source et le dernier le même nœud destination. Un WB-LSP ou F-LSP commence là où un autre termine;
- 4) Router le trafic à travers deux WB-LSPs ou F-LSPs et établir des chemins optiques pour les connecter entre la source et la destination.

Dans ce contexte, le taux d'utilisation des WB-LSPs et des F-LSPs établis est souvent inférieur à 100%, étant donné qu'on ne peut pas accéder aux longueurs d'onde individuelles le long des chemins existants sauf aux extrémités lors d'une nouvelle requête de connexion. Le débit global du réseau s'en trouve réduit comparé à un autre réseau semblable ayant le même nombre de longueurs d'onde disponibles mais étant à routage de longueurs d'onde. La réduction du coût du réseau en pénalise la flexibilité. Un autre objectif du routage multigranulaire serait donc d'améliorer le débit dans le réseau de façon à le rendre le plus proche possible du cas où des brasseurs de longueurs d'onde seraient les seuls à être utilisés.

### 2.2.5. Gestion de pannes

À cause des défaillances possibles des ports, des multiplexeurs et des démultiplexeurs au niveau d'un commutateur multigranulaire, une ou plusieurs bandes d'ondes dans une fibre ou plus peuvent être affectées sans que toute la fibre ou le lien soit affecté. Les approches de protection et de restauration actuelles traitent pour la plupart uniquement les failles des longueurs d'ondes individuelles, des fibres et/ou des liens. Cao et al. [8] ont introduit une nouvelle technique de récupération qui fusionne le trafic critique transporté dans une bande affectée par une panne de bande d'onde avec le trafic transporté par une bande non affectée. Ganguly et al. [17] ont proposé un algorithme dont la structure principale est un arbre de recouvrement (*spanning tree*) modifié, appelé « Waveband/wavelength Protection Tree » (WP-tree). Le réseau est protégé par un ou plusieurs WP-trees qui sont construits d'une façon qui prend en compte le nombre de chemins qui peuvent être protégés et le nombre de bandes d'onde disponibles. La protection peut se réaliser de deux façons possibles. Dans un premier cas, une bande de secours de la même capacité doit être trouvée pour le chemin de bande primaire à protéger. Dans le second scénario, le chemin de bande peut être protégé par plusieurs chemins optiques de secours. Pour chaque chemin de bande primaire à protéger, il faut décider laquelle des deux méthodes utiliser. La fonction de décision consiste à minimiser la capacité de secours.

## 2.3. Algorithmes de routage multigranulaire

Dans la littérature, on retrouve plusieurs algorithmes traitant du problème de routage et d'affectation de longueurs d'onde dans un réseau multigranulaire pour des demandes de trafic statique et des demandes de trafic dynamique. Nous présenterons ci-après, dans un premier temps, quelques algorithmes traitant le routage multigranulaire dynamique et par la suite quelques algorithmes traitant le cas statique.

### 2.3.1. Algorithmes de routage multigranulaire dynamique

Dans cette section, nous présenterons deux algorithmes de Mouftah et al., et un algorithme de Cao et al.

#### **Dynamic Tunnel Allocation (DTA) et Capacity-Balanced Static Tunnel Allocation (CB-STA)**

Mouftah et Pin-Han Ho [4] ont présenté deux algorithmes de routage et d'affectation de longueurs d'onde avec allocation de tunnels, l'allocation dynamique de tunnels (DTA) et l'allocation statique de tunnel à charge équilibrée (CB-STA) qui se proposent d'équilibrer la charge parmi les ressources de types de commutation différents, de façon à ce que le débit du système entier soit le plus proche possible du cas où uniquement des brasseurs de longueurs d'onde sont utilisés dans le réseau. Dans le DTA, le routage alternatif fixe avec  $k$  plus courts chemins est utilisé pour inspecter les ressources du réseau le long de chaque chemin alternatif afin d'établir dynamiquement les chemins optiques. Dans le CB-STA, les tunnels de bande et de fibre sont alloués dans le réseau dans la phase de planification (donc hors ligne) en se basant sur une métrique de lien définie par les auteurs, le « *weighted network link-state* » (W-NLS), qui vise à refléter la topologie du réseau, la charge potentielle (en terme de trafic) et l'emplacement de chaque paire source-destination. Dans ce contexte, un tunnel est défini comme étant un groupe de longueurs d'onde routées et commutées ensemble. Il existe alors deux types de tunnel : un *tunnel de bande d'onde* où le groupe de longueurs d'onde correspond à une bande d'onde et un *tunnel de fibre* où le groupe de longueurs d'onde correspond à l'ensemble des longueurs d'onde d'une fibre. Un tunnel comprend un nœud de départ qui est le nœud d'entrée du tunnel et un nœud de terminaison qui en constitue la sortie. Par ailleurs, les auteurs utilisent une conversion partielle de longueurs d'onde uniquement pour les transitions entre types de commutation différents. Chaque lien possède un nombre fixe de fibres, chacune de celles-ci contenant un nombre fixe de longueurs d'onde.

Le DTA est utilisé pour allouer dynamiquement des tunnels selon l'arrivée des demandes de connexion. La sélection dynamique d'un chemin optique considère à la fois les tunnels existants que celui-ci pourrait éventuellement emprunter et la possibilité d'établir un nouveau tunnel que le chemin optique pourra utiliser. Le DTA met l'accent sur l'habileté du réseau à s'adapter rapidement aux changements de trafic en établissant et libérant les tunnels selon le trafic en temps réel. L'algorithme pour l'implémentation du DTA se résume comme suit :

1. Trouver une combinaison faisable de tunnels le long d'un chemin alternatif sous inspection (incluant les tunnels existants et ceux pouvant être établis) de façon à établir un chemin optique entre la source et la destination.
2. Essayer tous les chemins alternatifs et trouver une combinaison optimale de tunnels le long d'un chemin. L'algorithme d'affectation de longueurs d'onde *First-Fit* est adopté.

La stratégie d'agrégation pour l'allocation dynamique est la suivante :

- l'ordre de priorité se présente comme suit du plus au moins prioritaire:
  1. utilisation d'un tunnel de fibre existant;
  2. utilisation d'un tunnel de bande existant;
  3. création d'un nouveau tunnel de fibre;
  4. création d'un nouveau tunnel de bande d'onde;
  5. utilisation d'un canal commuté en longueur d'onde;
- si un tunnel est enregistré comme étant de «longue vie», il ne sera jamais libéré même s'il n'est traversé par aucun chemin optique; dans le cas contraire, le tunnel sera libéré si aucun chemin optique ne le traverse.

Pour augmenter l'utilisation des fibres dédiées aux tunnels, le coût d'un lien avec différents types de commutation doit être tel que le coût du type de commutation de longueurs d'onde soit très supérieur à celui du type de commutation de bandes qui lui-même est supérieur au coût du type de commutation de fibres.

Le CB-STA vise à allouer des tunnels hors ligne dans la phase de planification du réseau en se basant sur la métrique W-NLS pour équilibrer la charge de trafic potentiel. L'algorithme du plus court chemin, Dijkstra, est utilisé pour chaque requête de connexion en prenant en considération les tunnels préétablis. Les tunnels préétablis ne seront jamais libérés. L'allocation des tunnels se divise en deux sous-tâches :

1. Déterminer les nœuds d'entrée et de sortie pour chaque tunnel en se basant sur la métrique W-NLS.
2. Router chaque tunnel en utilisant Dijkstra. La matrice de coût est dérivée en prenant en compte les tunnels. Le coût unitaire d'une commutation de fibres, de bandes d'onde et de longueurs d'onde est pris en considération. La longueur minimale, le nombre de tunnels d'un type de commutation, le nombre de longueurs d'onde disponibles dans un tunnel de fibre et de bande d'onde entre deux nœuds sont pris en compte également.

Pour la première sous-tâche, les auteurs ont défini le *potentiel* et l'*évier* (« *sink* ») d'un nœud pour quantifier la probabilité que des chemins optiques partent ou aboutissent à ce nœud. Le potentiel et l'évier sont exprimés en fonction de la métrique W-NLS. Le nœud ayant le plus grand potentiel possède la plus grande priorité d'être le nœud d'entrée d'un tunnel et celui ayant le plus grand évier la plus grande priorité d'être le nœud de sortie d'un tunnel. Après l'établissement d'un tunnel, le potentiel du nœud d'entrée et de ses voisins diminue. De même, l'évier du nœud de sortie et de ses voisins diminue.

### **L'algorithme MILB**

Cao et al. [7] ont proposé un algorithme, la longueur d'interférence maximum dans une bande (*Maximum Interference Length in Band : MILB*), dont l'objectif est de minimiser le nombre de ports utilisés au niveau des commutateurs multigranulaires qui sont reconfigurables. L'architecture utilisée pour les commutateurs est celle proposée par Cao et al. et présentée plus haut dans le chapitre. Les auteurs considèrent que le fait de minimiser le nombre de ports utilisés au niveau des commutateurs minimise la

probabilité de blocage des requêtes de connexion, ce qui paraît raisonnable. En effet, si moins de ports sont utilisés, il est plus probable de trouver un port libre pour établir la connexion.

Les auteurs assument qu'il n'existe aucune conversion de longueurs d'onde ni de bandes d'onde au niveau des commutateurs. Ils modélisent le réseau en  $B$  graphes de bande,  $B$  étant le nombre de bandes d'onde dans le réseau (un graphe pour chaque bande). Les nœuds dans un graphe pour une bande correspondent aux nœuds de la topologie physique du réseau, tandis que les liens entre deux nœuds correspondent à l'existence de la bande entre les nœuds. Pour une nouvelle demande de connexion, l'algorithme recherche les  $k$  plus courts chemins dans chacun des graphes de bande et détermine le poids du  $i$ ème plus court chemin emprunté dans le graphe de bande  $b$  qui a au moins une longueur d'onde disponible le long du  $i$ ème plus court chemin. Ce poids est dénoté  $Q_b^i$ . Afin de réduire la probabilité de blocage, l'algorithme choisit la paire  $(i, b)$  qui correspond au  $Q_b^i$  le plus élevé. La nouvelle demande de connexion est alors satisfaite en utilisant le  $i$ ème chemin et en affectant la première longueur d'onde disponible dans la bande d'onde  $b$  choisie.

### 2.3.2. Algorithmes de routage multigranulaire statique

Nous présenterons dans cette section quatre algorithmes traitant du routage et de l'affectation de longueurs d'onde dans un réseau multigranulaire dans le cas de demandes de trafic statique: celui de Kim et al., celui de Mouftah et al., celui de A. J. Glenstrup et celui de Cao et al.

#### Algorithme de Kim et al.

Kim et al. [18] ont présenté un algorithme pour le design de commutateurs multigranulaires dans les réseaux hiérarchiques. L'architecture des commutateurs est semblable à celle proposée par Cao et al. et présentée plus haut dans le chapitre. L'algorithme cherche à maximiser la réduction de la taille des commutateurs avec un minimum de longueurs d'onde. Notamment, il cherche à minimiser la taille des matrices

de commutation au niveau des couches WXC et WBXC (BXC dans l'architecture de Cao et al.). L'algorithme se déroule en trois étapes : le routage, le groupement des chemins optiques et l'affectation de longueurs d'onde.

L'algorithme cherche à router les chemins optiques de façon à regrouper ceux qui ont la même destination. Tous les chemins optiques du réseau suivent le principe d'optimalité suivant : si un nœud  $x$  fait partie du chemin optimal du nœud  $y$  au nœud  $z$ , alors le chemin optimal du nœud  $x$  au nœud  $z$  suit la même route sur cette partie du nœud  $y$  au nœud  $z$ . Le chemin optimal est défini comme étant un chemin à nombre minimal de sauts.

Après la sélection d'une route pour chaque chemin optique, l'algorithme détermine la bande d'onde à laquelle chaque chemin optique appartiendra. Pour ce faire, l'algorithme commence par classifier chaque chemin optique en des classes suivant le nœud de destination et liste tous les chemins optiques de chaque classe dans un ordre descendant suivant le nombre de sauts. Ensuite, il sélectionne le chemin optique en tête de liste dans une classe et cherche  $W_B - 1$  chemins optiques qui ont le plus de liens en commun afin de former un groupe,  $W_B$  étant le nombre maximal de longueurs d'onde dans une bande. Il affecte un numéro de bande aux chemins optiques du groupe et les enlève de la liste. La procédure est appliquée aux chemins optiques restants de la classe et s'étend ensuite aux autres classes.

Pour l'affectation de longueurs d'onde, la conversion de longueur d'onde est admise au niveau du WXC, mais pas au niveau du WBXC, et les longueurs d'onde sont contiguës à l'intérieur d'une bande. L'algorithme affecte les longueurs d'onde d'abord aux chemins de bande empaquetés, puis aux chemins de bande non empaquetés. Un chemin de bande empaqueté est défini par les auteurs comme étant un chemin de bande comprenant uniquement des chemins optiques ayant la même destination. Les autres chemins de bande sont des chemins de bande non empaquetés.

## Résolution du problème RWAT

Mouftah et al. [3] se proposent de dimensionner un réseau de façon à accommoder plus de demandes de trafic à long terme. Ils utilisent l'architecture de commutateur multigranulaire qu'ils ont eux-mêmes proposé et qui est présentée plus haut dans le chapitre. Cependant, en plus du fait de considérer qu'une fibre possède un type spécifique de commutation (fibre, bande d'onde ou longueur d'onde), les auteurs posent l'hypothèse que chaque commutateur possède un nombre fixe de ports (entrée/sortie) pour la commutation de longueurs d'onde, alors que le nombre de ports d'entrée/sortie pour les commutations de fibres et de bandes d'onde est sujet au design et à l'optimisation suivant le trafic. Les ressources du réseau à dimensionner sont le nombre de fibres et le nombre de ports d'entrée/sortie au niveau des commutateurs et ceci pour chaque type de commutation. Lorsque la demande de trafic croît, le nombre de fibres pour les commutations de fibres et/ou de bandes d'onde est graduellement augmenté au niveau des commutateurs. Afin de minimiser le nombre de fibres additionnelles nécessaires pour satisfaire une matrice de trafic donnée, les auteurs formulent le problème de routage et d'affectation de longueurs d'onde avec allocation de tunnels (RWAT) en un processus de programmation par contraintes (CP). Pour réduire la complexité de calcul induite par la formulation du CP, une heuristique est développée pour le design des commutateurs multigranulaires. Le CP est approximé par trois sous-tâches qui sont traitées séquentiellement.

La première sous-tâche consiste à déterminer les nœuds d'entrée et de sortie de tous les tunnels. Pour sélectionner les paires entrée/sortie des tunnels, les auteurs définissent le *potentiel* d'une paire de nœuds qui prend en compte la demande additionnelle de trafic à satisfaire entre les deux nœuds et la probabilité que le trafic d'une autre paire de nœuds emprunte un tunnel dont la paire entrée/sortie correspondrait à la paire de nœuds en question. La paire de nœuds ayant le plus grand potentiel est choisie comme paire entrée/sortie d'un tunnel. Si le potentiel est supérieur au nombre maximal de longueurs d'onde d'une fibre, noté  $K$  ( $K > 0$ ), le tunnel est un tunnel de

fibre, mais si le potentiel est compris entre 0 et K exclusivement, alors le tunnel est un tunnel de bande d'onde.

La seconde sous-tâche consiste à exécuter un processus RWA afin de router et affecter une longueur d'onde aux chemins optiques demandés dans la matrice de trafic et ceci en tenant compte de l'existence des paires entrée/sortie des tunnels. Une formulation de programmation linéaire en nombres entiers (ILP) est utilisée. La fonction objectif est de minimiser le nombre de chemins optiques qui ne peuvent pas être établis. Si toutes les demandes de la matrice de trafic ne sont pas satisfaites, plus de tunnels (fibre et/ou bande d'onde) sont alloués en répétant la première sous-tâche.

La troisième sous-tâche est de sélectionner des routes physiques pour tous les tunnels alloués dans la première sous-tâche. Un tunnel traverse le plus court chemin entre ses nœuds d'entrée et de sortie. Ensuite, le nombre de fibres pour chaque type de commutation est déterminé.

### **Algorithme glouton de Glenstrup**

A. J. Glenstrup [21] a développé un algorithme glouton pour configurer les commutateurs multigranulaires dans le réseau, en utilisant un nombre réduit de ports et de fibres. L'auteur utilise l'architecture de commutateur proposée par Cao et al.

L'algorithme proposé se déroule en cinq étapes :

1. déterminer un ensemble C de chemins (séquence de liens) pour toutes les demandes de trafic;
2. parmi les chemins trouvés à l'étape 1, déterminer les sous-chemins communs;
3. trier les sous-chemins communs suivant une valeur se basant sur leur volume et le nombre de sauts résultant en une liste S de sous-chemins. Le volume d'un sous-chemin est le nombre de chemins passant à travers sa séquence de liens et qui n'ont pas encore été routés;
4. tant que l'ensemble S des sous-chemins n'est pas vide,
  - a. enlever le premier sous-chemin de la liste;

- b. à partir des chemins de l'ensemble C, déterminer toutes les fibres et longueurs d'onde que les chemins devraient utiliser en entrant dans le nœud source du sous-chemin enlevé en a;
  - c. déterminer les ports utilisés le long du sous-chemin;
  - d. marquer tous les chemins qui sont partiellement routés le long du sous-chemin enlevé en a;
  - e. mettre à jour la liste des sous-chemins en enlevant les parties de sous-chemins qui chevauchent le sous-chemin enlevé en a. Trier la liste à nouveau;
5. configurer les commutateurs pour les insertions et extractions de fibres, de bandes et de longueurs d'onde.

Étant donné un ensemble de chemins, l'auteur a proposé un algorithme permettant de déterminer les sous-chemins communs. Pour chaque chemin de l'ensemble et pour chaque lien du chemin, l'algorithme recherche d'autres chemins qui utilisent ce lien. Il recherche ensuite les sous-chemins ayant le même nœud de terminaison et recherche leurs points de départ. L'algorithme ajoute ensuite les sous-chemins à la liste des sous-chemins. L'algorithme continue en recherchant les sous-chemins ayant le même point de départ et marque leur lien de départ.

### **L'algorithme BPHT**

Cao et al. [5] se proposent de déterminer le nombre de ports alloués à chaque couche au niveau des commutateurs multigranulaires du réseau. L'objectif est de minimiser soit le nombre de ports globalement dans le réseau, soit le nombre maximum de ports parmi tous les commutateurs du réseau, étant donné un ensemble donné de demandes de trafic à satisfaire sur une topologie de réseau donnée. Chaque lien du réseau peut posséder une seule ou plusieurs fibres. L'architecture de commutateur utilisée est celle proposée par les auteurs eux-mêmes et présentée plus haut dans le chapitre.

Pour minimiser le nombre de ports requis dans de grands réseaux, les auteurs ont développé un algorithme qu'ils ont appelé «*Balanced Path Routing with heavy-traffic first waveband assignment* » (BPHT). Avant de développer l'algorithme, les auteurs ont fait les suppositions suivantes :

1. chaque fibre a un nombre fixe de bandes d'onde;
2. chaque bande a un nombre fixe de longueurs d'onde;
3. l'ensemble des longueurs d'onde dans une bande est prédéterminé;
4. il n'y a aucune conversion de longueurs d'onde;
5. dans le calcul du nombre de ports, seuls les ports d'entrée des commutateurs multigranulaires sont pris en compte à cause de la symétrie de l'architecture de ces derniers;
6. la stratégie de regroupement des chemins optiques est celle où les chemins optiques possédant des liens intermédiaires en commun sont regroupés, indépendamment de leurs nœuds source et destination;
7. une demande de connexion correspond à une seule longueur d'onde.

L'algorithme BPHT se déroule en trois étapes :

- a) Routage à charge équilibrée
- b) Affectation de longueurs d'onde
- c) Configuration des commutateurs.

#### Routage à charge équilibrée

Cette étape détermine les routes en cherchant à équilibrer la charge dans le réseau. L'algorithme commence par déterminer les  $k$  ( $k > 1$ ) plus courts chemins pour toutes les paires ayant une demande de trafic non nulle. Il définit ensuite la charge d'un lien comme étant le nombre de routes traversant ce lien. Il continue en déterminant la route pour chaque paire de demande non nulle en essayant de minimiser la charge maximale parmi tous les liens et en commençant par la paire ayant le plus grand nombre de sauts le long de son plus court chemin.

### Affectation de longueurs d'onde

Les longueurs d'onde sont d'abord affectées aux chemins optiques qui utilisent deux sauts ou plus. Les auteurs se basent sur l'observation que ce trafic correspond à 60-80% du trafic total dans un réseau fédérateur. En même temps, l'algorithme veut donner la préférence aux chemins optiques qui en chevauchent d'autres de façon à tirer avantage le plus que possible des commutations de bandes et de fibres.

Les étapes suivantes sont utilisées pour affecter des longueurs d'onde à tous les chemins optiques, une fois le routage de l'étape a) terminé :

1. Pour toutes les paires de nœuds (s, d), dont la route est déterminée comme étant  $s = s_0 \rightarrow s_1 \rightarrow s_2 \dots s_{n-1} \rightarrow s_n = d$ , définir un ensemble  $Q_d^s$  qui inclut toutes les paires  $(s_i, s_j)$  dont la route est déterminée à l'étape a), où  $0 \leq i \leq n - 2$  et  $i + 2 \leq j \leq n$ .
2. Calculer le poids de chaque ensemble  $Q_d^s$  comme étant  $W_{sd} = \sum_{p \in Q_d^s} h_p \times t_p$ , où  $p$  est une paire  $(s_i, s_j)$  appartenant à l'ensemble  $Q_d^s$ ,  $h_p$  est le nombre de sauts et  $t_p$  est le nombre de chemins optiques requis entre  $s_i$  et  $s_j$ .
3. Trouver l'ensemble  $Q_d^s$  ayant le plus grand poids  $W_{sd}$ .
4. Cette étape est divisée elle-même en les étapes suivantes :
  - a. Affecter les longueurs d'onde aux chemins optiques demandés pour la paire de nœuds (s, d) en essayant de les grouper dans la même fibre et à l'intérieur de la fibre, dans une même bande. Les auteurs ont introduit alors un nouvel algorithme d'affectation de bandes d'onde appelé «*waveband assignment for multifiber WB*» (WA-MF-WBS), algorithme décrit à la Figure 2.5.

Soit  $K$  le nombre maximal de longueurs d'onde dans une fibre,  $B$  le nombre de bandes d'onde dans une fibre et  $F$  le nombre de fibres sur un lien.

Soit  $w$ ,  $b$  et  $f$  les indices d'une longueur d'onde, d'une bande d'onde et d'une fibre respectivement, tel que  $0 \leq w \leq K-1$ ,  $0 \leq b \leq B-1$ ,  $1 \leq f \leq F$ . Initialement,  $f = 1$  et  $w = b = 0$  pour toutes les fibres.

**Tant que  $t_p > W$  Faire**

Trouver une fibre qui a autant de bandes disponibles que possible en commençant par la fibre d'indice  $f$  (disons  $a \leq [t_p/W]$ )

{

Soit  $g$  la fibre trouvée, où  $g$  peut être égal ou non à  $f$ ;

Affecter les bandes de la fibre  $g$  aux  $a.W$  chemins optiques de la paire  $p$ ;

$t_p = t_p - a.W$ ;

Faire  $f = g$ , mettre à jour  $w$  et  $b$  pour la fibre  $g$ ;

}

**Fin Tant que**

**Tant que  $t_p > 0$  Faire**

Trouver une fibre  $g$  qui a au moins une longueur d'onde disponible en commençant par la fibre  $f$ ;

Affecter une longueur d'onde disponible  $x$  au chemin optique pour  $p$ , en commençant par l'index  $w$ . Il est très probable que  $x$  soit égal à  $w$ ;

$t_p = t_p - 1$ ;

Faire  $f = g$  et  $w = x + 1$ . Mettre à jour  $b$  pour la fibre  $g$ ;

**Fin Tant que**

**Figure 2.5 Algorithme WA-MF-WBS**

- b. Utiliser l'algorithme WA-MF-WBS pour affecter des longueurs d'onde aux chemins optiques demandés pour la paire  $(s_i, s_j)$ , en commençant par le plus grand  $j$ .
- c. Utiliser l'algorithme WA-MF-WBS pour affecter des longueurs d'onde aux chemins optiques demandés pour la paire  $(s_i, d)$  en commençant par le plus petit  $i$ .

- d. S'il existe toujours des paires  $(s_i, s_j)$  appartenant à l'ensemble  $Q_d^s$  qui n'ont pas encore été considérées, répéter à partir de l'étape 4 en traitant le  $s_i$  ayant le plus petit  $i$  comme étant  $s$ , et le  $s_j$  ayant le plus grand  $j$  comme étant  $d$ . Sinon, aller à l'étape 5.
- 5. Recalculer le poids des paires de noeuds dont les routes utilisent n'importe quelle partie de la route utilisée par la paire de noeuds  $(s, d)$ . Pour chaque fibre, faire  $b = (b+1) \bmod B$  et  $w = b \cdot W$  et ensuite aller à l'étape 3. Répéter jusqu'à ce que toutes les demandes de chemins optiques de plus de deux sauts soient satisfaites.
- 6. Affecter les longueurs d'onde aux chemins optiques entre deux noeuds séparés par un seul saut, en commençant par la paire de noeuds ayant la plus grande demande de chemins optiques.

#### Configuration des commutateurs

Une fois l'affectation de longueurs d'onde terminée, l'algorithme commute le plus de fibres que possible en utilisant des FXCs. Il commute ensuite autant de bandes d'onde que possible en utilisant les BXC. Les chemins optiques restants sont individuellement commutés au niveau de la couche WXC. Le nombre total de ports utilisés au niveau de chaque commutateur est alors déterminé. Au niveau de la couche WXC, le nombre de ports à l'entrée inclut les ports pour les chemins optiques arrivant d'autres noeuds et les ports pour les chemins optiques insérés dans le réseau à partir de ce commutateur. Le nombre de ports requis à l'entrée au niveau de la couche BXC est la somme du nombre de bandes insérées, extraites et commutées au niveau du brasseur de bandes d'onde. Cette somme inclut aussi le nombre de bandes provenant du multiplexage de longueurs d'onde au niveau des multiplexeurs WTB et le nombre de bandes séparées en longueurs d'onde au niveau des démultiplexeurs BTW. Le nombre de ports requis à l'entrée au niveau de la couche FXC est la somme du nombre de fibres insérées, extraites et commutées au niveau du brasseur de fibres et du nombre de fibres des multiplexeurs (démultiplexeurs) BTF (FTB).

Les heuristiques utilisées pour l'affectation de longueurs d'onde reposent sur des principes solides et permettent à l'algorithme de fournir de bons résultats dans son ensemble. L'analyse de performance de l'algorithme [5] montre que pour des réseaux de taille réduite, les solutions obtenues par le BPHT sont très proches de la solution optimale du modèle ILP (programmation linéaire en nombres entiers) trouvée en utilisant le solutionnaire ILP, CPLEX. Pour des réseaux de grande taille, l'algorithme fournit des résultats satisfaisants avec des réductions appréciables du nombre de ports comparé au nombre de ports nécessaires si les réseaux considérés étaient des réseaux à routage de longueur d'onde utilisant des commutateurs ordinaires (OXC). Pour ce faire, les auteurs ont défini des métriques de performance qui fournissent le rapport entre le nombre de ports des commutateurs multigranulaires calculé en utilisant le BPHT et le nombre de ports trouvé si des commutateurs ordinaires étaient les seuls utilisés dans le réseau et en utilisant toujours le même algorithme BPHT. Lorsque des commutateurs ordinaires sont uniquement utilisés dans le réseau, le fait de minimiser le nombre de ports dans le réseau est équivalent à minimiser le nombre de sauts dans le réseau. La formulation ILP du problème à l'étude est équivalente à celle pour le RWA optimal minimisant le nombre de sauts. Le BPHT, basée sur la formulation du problème, minimisera alors le nombre de sauts dans le réseau. Les valeurs des rapports des nombres de ports sont souvent très inférieures à l'unité indiquant des réductions appréciables.

Cependant, des modifications peuvent être apportées à l'algorithme BPHT, notamment au niveau de la sélection des routes, afin d'améliorer les solutions obtenues. La sélection des routes est le premier élément à considérer dans un algorithme cherchant à diminuer le nombre de ports. Si les chemins optiques ne se partagent pas des liens, l'affectation de longueurs d'onde n'est plus pertinente pour la réduction du nombre de ports. Le routage du BPHT cherche à équilibrer la charge à travers le réseau. C'est un bon début, mais le fait d'équilibrer la charge dans le réseau ne fournit pas un ensemble de routes conduisant à un nombre minimal de ports. Il est possible de faire mieux que l'algorithme proposé par Cao et al. , en améliorant notamment la partie routage.

#### **2.4. Réseaux multigranulaires à deux niveaux de granularité**

Dans ce mémoire, jusqu'à présent nous avons considéré uniquement les réseaux multigranulaires à trois niveaux de granularité. Cependant, dans la littérature, plusieurs études [9] – [17] ont été menées sur des réseaux multigranulaires à deux niveaux de granularité, les deux granularités étant la longueur d'onde et la bande d'onde. Les mêmes principes utilisés pour le routage multigranulaire à trois niveaux demeurent pour le routage multigranulaire à deux niveaux, sauf qu'il n'existe pas de commutation de fibres. Les architectures de commutateur diffèrent quelque peu aussi en plus du fait qu'il n'existe pas de couche pour la commutation de fibres.

La multigranularité à deux niveaux de granularité est mieux adaptée aux réseaux monofibres, alors que celle à trois niveaux est mieux adaptée aux réseaux multifibres. Dans ce mémoire, les réseaux qui seront considérés seront des réseaux métropolitains multifibres. De plus, dans la pratique, les réseaux optiques sont des réseaux multifibres.

## CHAPITRE III

### ALGORITHMES DE ROUTAGE PROPOSÉS

La multigranularité dans les réseaux optiques a été proposée principalement pour réduire le coût et la complexité à gérer de tels réseaux, notamment en diminuant le nombre de ports dans le réseau. Le routage et l'affectation de longueurs d'onde dans ces réseaux demeurent les éléments clés pour parvenir à une réduction appréciable du nombre de ports. Cette diminution dépend de la sélection des routes et de l'affectation des longueurs d'onde pour les chemins optiques. La sélection des routes doit chercher à regrouper les chemins le plus efficacement que possible. L'affectation des longueurs d'onde doit tenter de regrouper le plus que possible les longueurs d'onde dans une même fibre et, à l'intérieur d'une fibre, dans une même bande d'onde. Ce chapitre traite du problème du routage et d'affectation de longueurs d'onde qui cherche à minimiser le nombre de ports globalement dans le réseau dans le cas d'un modèle de trafic statique. Nous appellerons ce problème MIN-MRWA. Après avoir défini celui-ci, nous décrirons de manière détaillée l'algorithme que nous proposons pour le résoudre. Cet algorithme sera basé sur l'utilisation de la métaheuristique taboue et tentera d'obtenir des solutions dans des temps de calcul raisonnables. Enfin, nous terminerons le chapitre par une analyse de la complexité de notre approche.

#### 3.1. Définition du problème

Un exemplaire du problème de routage et d'affectation de longueurs d'onde que nous traitons dans ce mémoire est défini par :

- La topologie du réseau, représentée par un graphe non orienté noté  $G = (V, E)$ . L'ensemble  $V = \{v_1, \dots, v_N\}$  des sommets du graphe représente les nœuds du réseau. Les nœuds sont des commutateurs multigranulaires dont l'architecture a

été présentée au chapitre précédent.  $E = \{e_1, \dots, e_M\}$  est l'ensemble des arcs du graphe représentant les liens du réseau;

- $P$ , l'ensemble des paires de nœuds de demande non nulle. La demande d'une paire de nœuds  $p \in P$  est le nombre de chemins optiques à établir entre ses deux nœuds et est notée  $\text{multiplicity}(p)$ . Une paire de nœuds  $p$  est constituée d'un nœud source, noté  $s(p)$ , et d'un nœud destination, noté  $d(p)$ ;
- $F$ , le nombre de fibres par lien. Les fibres sont numérotées de 1 à  $F$ ;
- $K$ , le nombre de bandes d'onde par fibre. Les longueurs d'onde sont numérotées de 1 à  $K$  pour chaque fibre;
- $W$ , le nombre de longueurs d'onde par bande. Les longueurs d'onde sont numérotées de 1 à  $W$  pour chaque bande.

Le nombre de bandes d'onde par fibre, noté  $B$ , se déduit du nombre de longueurs d'onde par fibre et du nombre de longueurs d'onde par bande. Il est donné par la relation :  $B = K / W$ . Les bandes d'onde sont numérotées de 1 à  $B$  dans une fibre.

Afin de prendre en compte le fait que deux nœuds du réseau peuvent être interconnectés par plus d'une fibre, nous considérons le multigraphe non orienté  $G_F = (V, A)$ .  $V$  est le même ensemble de sommets du graphe  $G$  et  $A$  représente l'ensemble des fibres du réseau. Nous définissons en plus une fonction  $f$  de  $A \rightarrow \{\{v_1, v_2\} \mid v_1, v_2 \in V, v_1 \neq v_2\}$  faisant correspondre à une fibre la paire de nœuds interconnectés par celle-ci.

Un chemin  $\eta$  de  $v_1$  à  $v_2$  dans  $G_F$  est une séquence d'arcs  $a_1, a_2, \dots, a_\ell$  de  $G_F$  tels que  $f(a_1) = (o_1, o_2), f(a_2) = (o_2, o_3), \dots, f(a_\ell) = (o_\ell, o_{\ell+1})$  où  $v_1 = o_1$  et  $v_2 = o_{\ell+1}$ . Nous notons  $\text{PATHS}(G_F)$  l'ensemble de tous les chemins dans le multigraphe  $G_F$  et  $\text{PATHS}(v_1, v_2, G_F)$  l'ensemble de tous les chemins entre les nœuds  $v_1$  et  $v_2$  dans  $G_F$ . Nous disons que deux chemins  $\eta = (a_1, a_2, \dots, a_\ell)$  et  $\eta' = (a_1', a_2', \dots, a_{\ell'}')$  sont adjacents,

et nous notons  $\eta \cap \eta' \neq \emptyset$ , s'il existe des entiers  $i$  ( $1 \leq i \leq \ell-1$ ) et  $j$  ( $1 \leq j \leq \ell'-1$ ) tel que  $a_i = a'_{j+1}$ . Dans le cas où les chemins  $\eta$  et  $\eta'$  ne sont pas adjacents, nous notons  $\eta \cap \eta' = \emptyset$ .

Un chemin optique est un couple  $(\eta, \lambda)$  défini par un chemin  $\eta$  dans le multigraph  $G_F$  et une longueur d'onde  $\lambda : (\eta, \lambda) \in \text{PATHS}(G_F) \times \{1, \dots, K\}$ . Une solution potentielle au problème MIN-MRWA est un ensemble de chemins optiques, noté  $S$ , tel que  $S = \bigcup_{p \in P} S_p$ , chaque  $S_p$  représentant l'ensemble des chemins optiques pour satisfaire la demande de la paire  $p : S_p \subseteq \text{PATHS}(s(p), d(p), G_F) \times \{1, \dots, K\}$ .

Pour être une solution légale,  $S$  doit satisfaire les contraintes suivantes :

- $|S_p| = \text{multiplicity}(p), \forall p \in P$  (C1)

- $(\eta_1, \lambda) \in S, (\eta_2, \lambda) \in S, \eta_1 \neq \eta_2 \Rightarrow \eta_1 \cap \eta_2 = \emptyset$  (C2)

La contrainte (C1) impose que toutes les demandes de connexion optique soient satisfaites et que chaque ensemble de solution  $S_p$  ne contienne pas plus de chemins optiques que demandés. La contrainte (C2) interdit que deux chemins optiques se partageant une fibre utilisent la même longueur d'onde.

Lorsqu'un chemin optique traverse un nœud, il utilise un port à l'entrée et à la sortie de celui-ci. Cependant, deux chemins optiques se partageant deux arcs adjacents au même nœud utilisent le même port à l'entrée et à la sortie de ce dernier. La solution optimale du problème MIN-MRWA minimise le nombre de ports requis globalement dans le réseau en regroupant efficacement les chemins optiques dans la même fibre, tout en respectant les contraintes présentées ci-haut. L'objectif du problème est de trouver la solution optimale.

### 3.1.1. Calcul du nombre de ports

Soit  $\mathcal{T}$  l'ensemble de toutes les solutions légales du problème et soit la fonction  $\varphi$  de  $\mathcal{T} \rightarrow \mathbb{N}$  faisant correspondre à une solution  $S \in \mathcal{T}$  le nombre total de ports dans le réseau. Le nombre  $\varphi(S)$  représente le coût de la solution  $S$ . Il est à noter cependant que le nombre de ports ne tient compte que des ports d'entrée au niveau des nœuds (commutateurs multigranulaires). Ceci est dû à la symétrie de l'architecture des nœuds où le nombre de ports d'entrée est égal au nombre de ports de sortie.

Afin de déterminer le nombre de ports dans le réseau, nous introduisons la notion de *coudé* qui correspond au triplet  $(x, v, y) \in A \times V \times A$  tels que  $x \neq y$  et  $x$  et  $y$  sont adjacents au même sommet  $v$ . L'arc  $x$  est dit entrant et l'arc  $y$  est dit sortant. Le triplet  $(x, v, y)$  nous permet de traiter le cas où des chemins optiques sont routés d'un arc  $x$  à un arc  $y$  à travers un nœud  $v$ . Nous devons également considérer le cas où des chemins optiques partent d'un nœud et celui où ils aboutissent à un nœud. Le triplet  $(0, v, y)$  sera utilisé afin de nous permettre de traiter le cas particulier des départs et le triplet  $(x, v, 0)$  sera employé afin de traiter le cas des aboutissements.

Afin de calculer le nombre de ports, nous définissons aussi les termes suivants :

- $\mathcal{C}_v$  l'ensemble des coudes du réseau traversant le nœud  $v$ ;
- $u(a)$ , avec  $a \in A$ , le nombre de chemins optiques traversant l'arc  $a$ ;
- $u(a, b)$ , avec  $a \in A$  et  $b \in \{1, \dots, B\}$ , le nombre de chemins optiques traversant l'arc  $a$  et utilisant la bande  $b$ ;
- $u(x, v, y)$  le nombre de chemins optiques utilisant le coudé  $(x, v, y)$ , c'est-à-dire dont le trafic traverse le nœud  $v$  de l'arc  $x$  à l'arc  $y$ ;
- $u((x, v, y), b)$ , avec  $b \in \{1, \dots, B\}$ , le nombre de chemins optiques utilisant le coudé  $(x, v, y)$  et la bande  $b$ ;

- $cost(x, v, y)$  le coût du coude  $(x, v, y)$ , c'est-à-dire le nombre de ports requis au niveau du nœud  $v$  pour router des chemins optiques de l'arc  $x$  à l'arc  $y$ ;
- $cost((x, v, y), b)$ , avec  $b \in \{1, \dots, B\}$ , le coût du coude  $(x, v, y)$  au niveau de la bande  $b$ , c'est-à-dire le nombre de ports requis pour router des chemins optiques de l'arc  $x$  à l'arc  $y$  au niveau de la bande  $b$ ;
- $pénalités(v)$ , avec  $v \in V$ , le nombre de pénalités au niveau du nœud  $v$ . Cette notion de pénalité sera expliquée un peu plus loin dans cette section.

L'expression de la fonction  $\varphi$  est la suivante :  $\varphi = \sum_{v \in V} (\varphi_1 + \varphi_2 + \varphi_3)$ . Les termes

$\varphi_1$ ,  $\varphi_2$  et  $\varphi_3$  permettent de calculer le nombre de ports au niveau d'un nœud, mais chacun traite un cas différent.  $\varphi_1$  traite le cas où les chemins optiques passent à travers un nœud d'un arc à un autre,  $\varphi_2$  le cas où les chemins optiques partent du nœud et  $\varphi_3$  le cas où les chemins optiques aboutissent au nœud.

L'expression de  $\varphi_1$  est la suivante :

$$\varphi_1 = \sum_{(x, v, y) \in \mathcal{C}_r} cost(x, v, y) + pénalités_1(v)$$

où

$$\begin{cases} cost(x, v, y) = 0 & \text{si } u(x, v, y) = 0 \\ cost(x, v, y) = 1 & \text{si } u(x, v, y) = u(x) = u(y) \end{cases} \quad (3.1)$$

$$cost(x, v, y) = \sum_{b \in \{1, \dots, B\}} cost((x, v, y), b) \quad \text{sinon} \quad (3.2)$$

$$\text{de plus } \begin{cases} cost((x, v, y), b) = 0 & \text{si } u((x, v, y), b) = 0 \\ cost((x, v, y), b) = 1 & \text{si } u((x, v, y), b) = u(x, b) = u(y, b) \\ cost((x, v, y), b) = u((x, v, y), b) & \text{sinon} \end{cases} \quad (3.4)$$

$$(3.5)$$

$$(3.6)$$

L'expression de  $\varphi_2$  est la suivante :

$$\varphi_2 = \sum_{(0,v,y) \in \mathcal{C}_v} \text{cost}(0,v,y) + \text{pénalités}_2(v)$$

$$\text{où. } \begin{cases} \text{cost}(0,v,y) = 0 & \text{si } u(0,v,y) = 0 \\ \text{cost}(0,v,y) = 1 & \text{si } u(0,v,y) = u(y) \end{cases} \quad (3.7)$$

$$\begin{cases} \text{cost}(0,v,y) = \sum_{b \in \{1, \dots, B\}} \text{cost}((0,v,y), b) & \text{sinon} \end{cases} \quad (3.8)$$

$$\text{de plus. } \begin{cases} \text{cost}((0,v,y), b) = 0 & \text{si } u((0,v,y), b) = 0 \\ \text{cost}((0,v,y), b) = 1 & \text{si } u((0,v,y), b) = u(y, b) \end{cases} \quad (3.10)$$

$$\begin{cases} \text{cost}((0,v,y), b) = u((0,v,y), b) & \text{sinon} \end{cases} \quad (3.11)$$

$$\begin{cases} \text{cost}((0,v,y), b) = u((0,v,y), b) & \text{sinon} \end{cases} \quad (3.12)$$

L'expression de  $\varphi_3$  est la suivante :

$$\varphi_3 = \sum_{(x,v,0) \in \mathcal{C}_v} \text{cost}(x,v,0) + \text{pénalités}_3(v)$$

$$\text{où. } \begin{cases} \text{cost}(x,v,0) = 0 & \text{si } u(x,v,0) = 0 \\ \text{cost}(x,v,0) = 1 & \text{si } u(x,v,0) = u(x) \end{cases} \quad (3.13)$$

$$\begin{cases} \text{cost}(x,v,0) = \sum_{b \in \{1, \dots, B\}} \text{cost}((x,v,0), b) & \text{sinon} \end{cases} \quad (3.14)$$

$$\begin{cases} \text{cost}((x,v,0), b) = 0 & \text{si } u((x,v,0), b) = 0 \\ \text{cost}((x,v,0), b) = 1 & \text{si } u((x,v,0), b) = u(x, b) \end{cases} \quad (3.15)$$

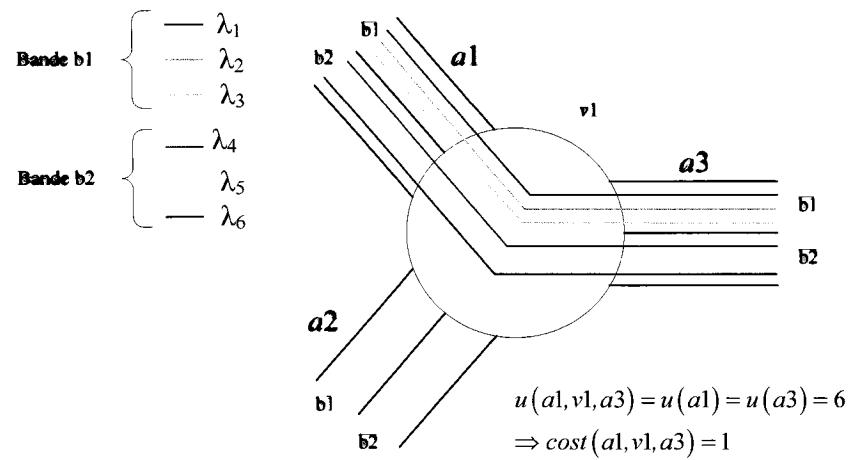
$$\begin{cases} \text{cost}((x,v,0), b) = u((x,v,0), b) & \text{sinon} \end{cases} \quad (3.16)$$

$$\begin{cases} \text{cost}((x,v,0), b) = u((x,v,0), b) & \text{sinon} \end{cases} \quad (3.17)$$

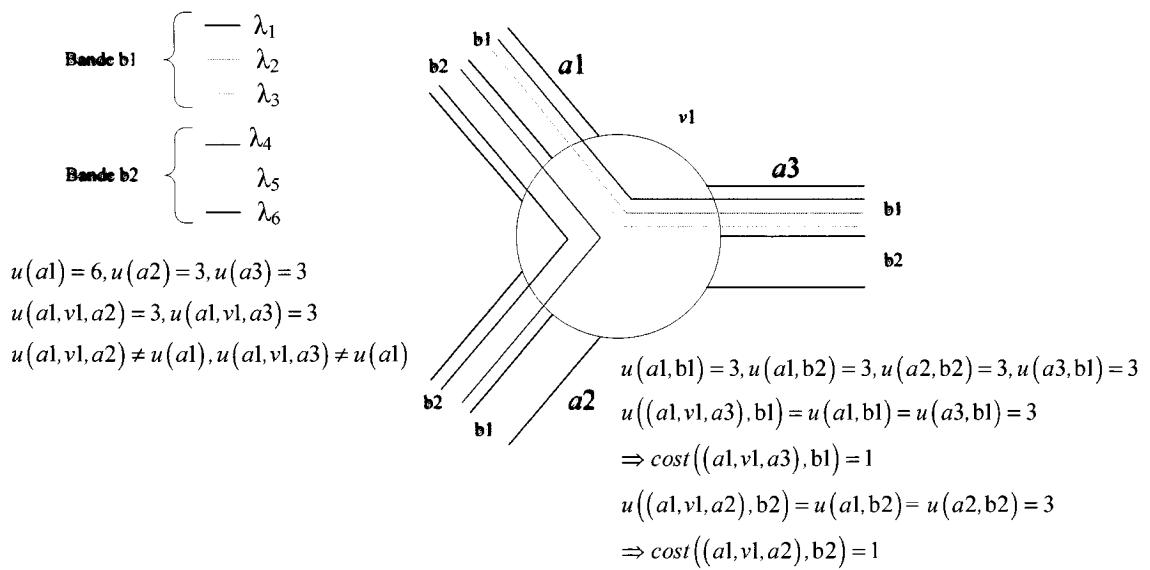
$$\begin{cases} \text{cost}((x,v,0), b) = u((x,v,0), b) & \text{sinon} \end{cases} \quad (3.18)$$

Un coude permet de déterminer le type de commutation des longueurs d'onde qui y sont utilisées. Les relations (3.2), (3.8) et (3.14) indiquent une commutation de fibre. Les chemins optiques utilisant le coude en considération ( $(x,v,y)$ ,  $(0,v,y)$  ou  $(x,v,0)$ ) sont les seuls à utiliser les arcs correspondants (lorsque c'est applicable). En d'autres termes, pour un coude  $(x,v,y)$ , toutes les longueurs d'onde utilisées sur l'arc entrant  $x$

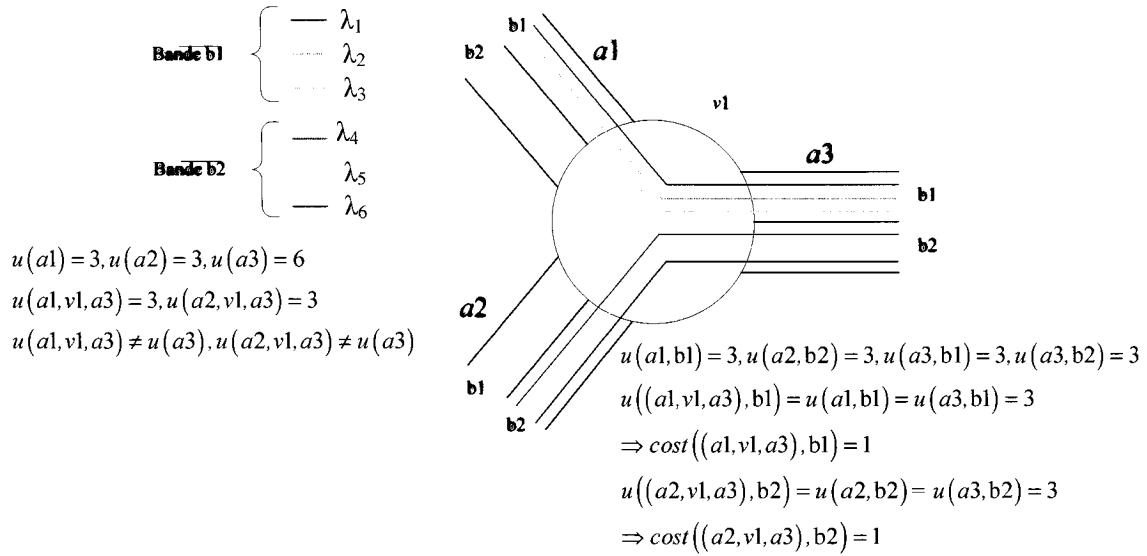
sont commutées vers le même arc sortant  $y$  et toutes les longueurs d'onde de celui-ci proviennent de l'arc  $x$ . Pour un coude  $(0, v, y)$  ( $(x, v, 0)$ ), toutes les longueurs d'onde de l'arc  $y$  (de l'arc  $x$ ) partent du (aboutissent au) même nœud  $v$ . La Figure 3.1a fournit un exemple d'une commutation de fibre dans le cas où des chemins optiques traversent un nœud d'un arc à un autre. Les relations (3.3), (3.9) et (3.15) indiquent que dans le cas où une commutation de fibre n'est pas possible, il est nécessaire d'inspecter les bandes d'onde individuellement et de comptabiliser le nombre de ports requis au niveau de chaque bande. Les relations (3.5), (3.11) et (3.17) indiquent une commutation de bande. Pour un coude  $(x, v, y)$ , toutes les longueurs d'onde d'une bande  $b$  de l'arc entrant  $x$  sont commutées vers le même arc sortant  $y$  et toutes les longueurs d'onde de la bande  $b$  de celui-ci proviennent de l'arc  $x$ . Pour un coude  $(0, v, y)$  ( $(x, v, 0)$ ), toutes les longueurs d'onde de l'arc  $y$  (de l'arc  $x$ ) au niveau d'une bande  $b$  partent du (aboutissent au) même nœud  $v$ . Les Figures 3.1b et 3.1c fournissent un exemple de commutation de bande dans le cas où des chemins optiques traversent un nœud d'un arc à un autre. Les relations (3.6), (3.12) et (3.18) indiquent qu'il est nécessaire de considérer de manière individuelle les longueurs d'onde de la bande en considération. Tous les chemins optiques utilisant l'arc  $x$  d'un coude  $(x, v, y)$  au niveau d'une bande  $b$  ne sont pas routés vers l'arc  $y$  ou encore tous les chemins optiques utilisant l'arc  $y$  (l'arc  $x$ ) ne partent pas du (n'aboutissent pas au) nœud  $v$ . Les Figures 3.1d et 3.1e fournissent un exemple où les longueurs d'onde d'une bande doivent être considérées individuellement. Les chemins optiques dans ces exemples sont routés d'un arc à un autre.



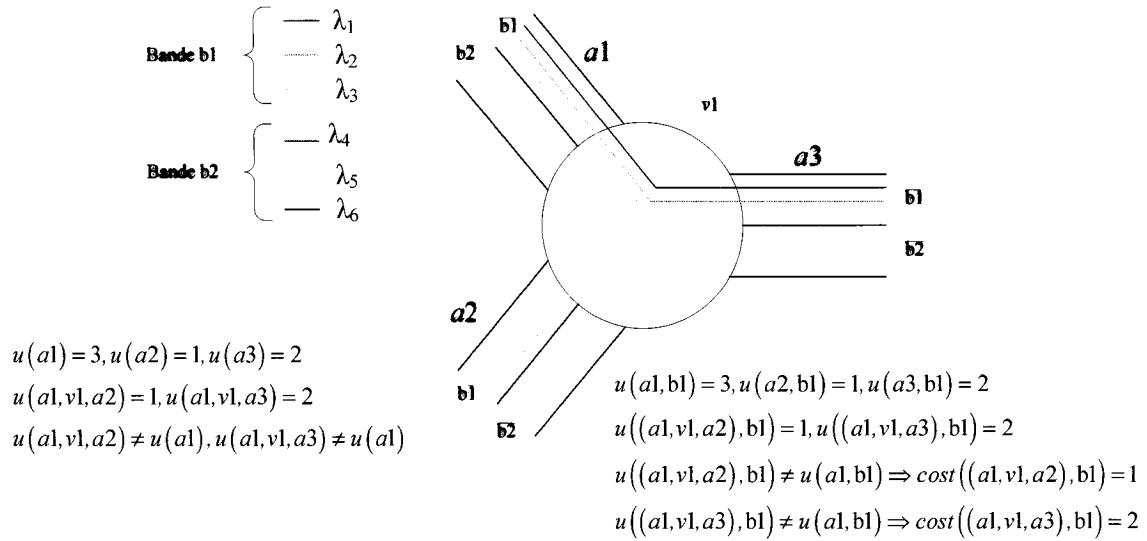
(a) commutation de fibre



(b) commutation de bande – exemple 1

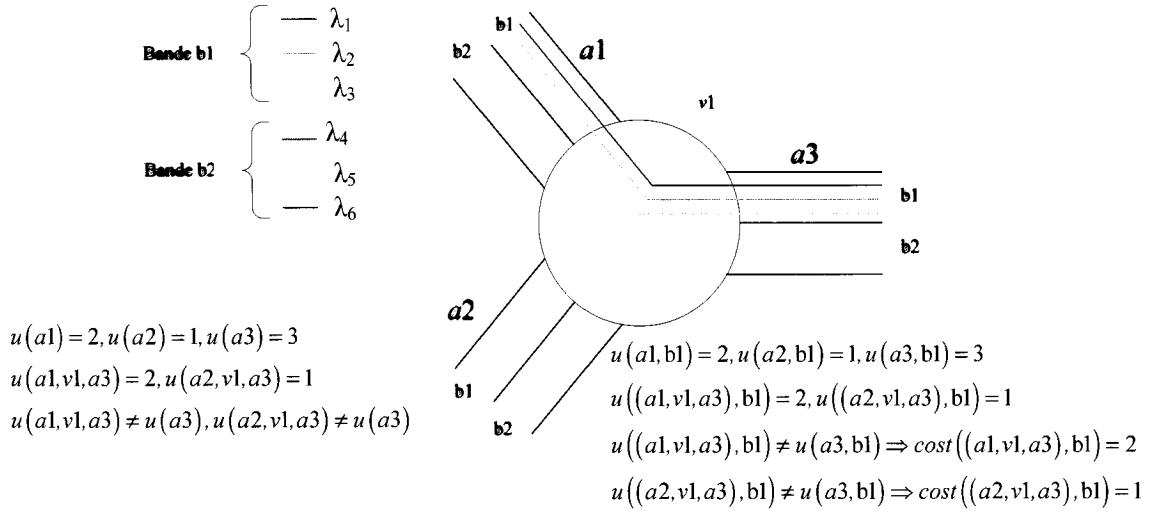


(c) commutation de bande – exemple 2



(d) commutation lambda – exemple 1

(commutation individuelle de longueurs d'onde)



(e) commutation lambda – exemple 2  
(commutation individuelle de longueurs d’onde)

**Figure 3.1 Coût d’un coude**

Les termes  $pénalités_1(v)$ ,  $pénalités_2(v)$  et  $pénalités_3(v)$  indiquent le nombre de pénalités au niveau du nœud  $v$  respectivement dans le cas de traversée, de départ et d’aboutissement de chemins optiques au niveau de celui-ci. Nous distinguons deux types de pénalités : une pénalité d’arc, notée  $pénalité(v, a)$  avec  $v \in V$  et  $a \in A$ , et une pénalité de bande, notée  $pénalité(v, a, b)$  avec  $v \in V$ ,  $a \in A$  et  $b \in \{1, \dots, B\}$ . Une pénalité d’arc survient toutes les fois qu’une commutation de fibre n’est pas possible, impliquant que les bandes individuelles de la fibre soient inspectées. La pénalité est associée à l’arc donc si plusieurs coudes se partageant un arc ne correspondent pas à une commutation de fibre, il n’y aura qu’une pénalité sur cet arc et non plusieurs. De plus, si les longueurs d’onde individuelles d’une bande doivent être considérées, signifiant qu’une commutation de bande n’est pas possible pour cette bande, une pénalité sera associée à la bande.

Soit  $A_v \subset A$ , l'ensemble des arcs adjacents au nœud  $v$ . Le nombre de pénalités d'un nœud  $v$ , dans le cas où les chemins optiques passent à travers celui-ci d'un arc à un autre, se calcule de la façon suivante :

$$pénalités_1(v) = \sum_{a \in A_v} pénalité_1(v, a) + \sum_{a \in A_v} \sum_{b \in \{1, \dots, B\}} pénalité_1(v, a, b)$$

$$\text{où } \begin{cases} pénalité_1(v, a) = 1 & \text{si } \exists x \in A_v \mid u(x, v, a) \neq u(x) \text{ ou } u(x, v, a) \neq u(a) \\ & \text{ou si } \exists y \in A_v \mid u(a, v, y) \neq u(a) \text{ ou } u(a, v, y) \neq u(y) \\ pénalité_1(v, a) = 0 & \text{sinon} \end{cases}$$

$$\begin{cases} pénalité_1(v, a, b) = 1 & \text{si } \exists x \in A_v \mid u((x, v, a), b) \neq u(x, b) \text{ ou } u((x, v, a), b) \neq u(a, b) \\ & \text{ou si } \exists y \in A_v \mid u((a, v, y), b) \neq u(a, b) \text{ ou } u((a, v, y), b) \neq u(y, b) \\ & \text{et si } pénalité_1(v, a) = 1 \\ pénalité_1(v, a, b) = 0 & \text{sinon} \end{cases}$$

Le nombre de pénalités d'un nœud  $v$  dans le cas où des chemins optiques partent de celui-ci se calcule de la façon suivante :

$$pénalités_2(v) = \sum_{a \in A_v} pénalité_2(v, a) + \sum_{a \in A_v} \sum_{b \in \{1, \dots, B\}} pénalité_2(v, a, b)$$

$$\text{où } \begin{cases} pénalité_2(v, a) = 1 & \text{si } u(a, v, 0) \neq u(a) \\ pénalité_2(v, a) = 0 & \text{sinon} \end{cases}$$

$$\begin{cases} pénalité_2(v, a, b) = 1 & \text{si } u((a, v, 0), b) \neq u(a, b) \text{ et si } pénalité_2(v, a) = 1 \\ pénalité_2(v, a, b) = 0 & \text{sinon} \end{cases}$$

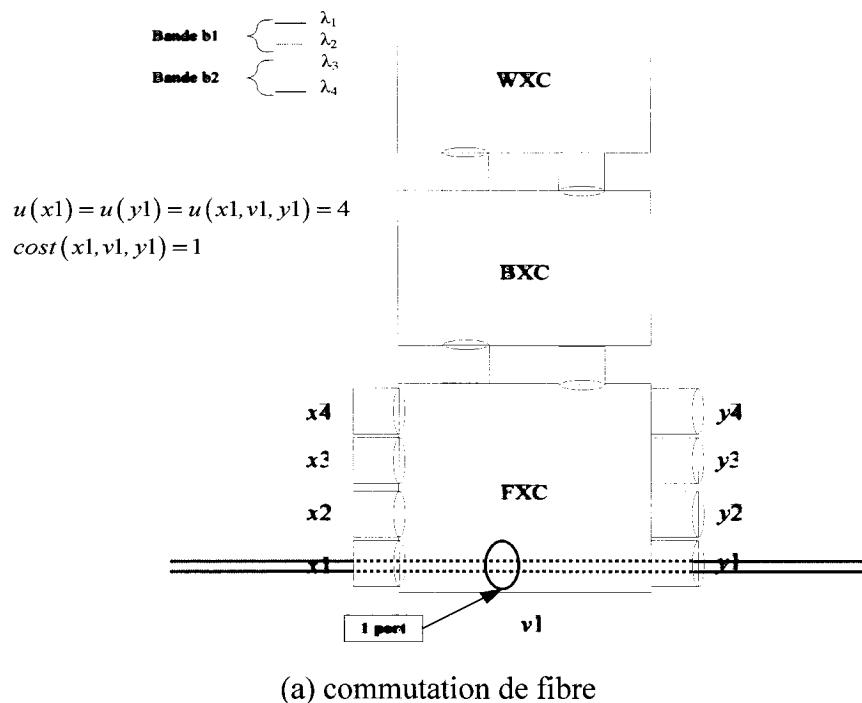
Le nombre de pénalités d'un nœud  $v$  dans le cas où des chemins optiques aboutissent à celui-ci se calcule de la manière suivante :

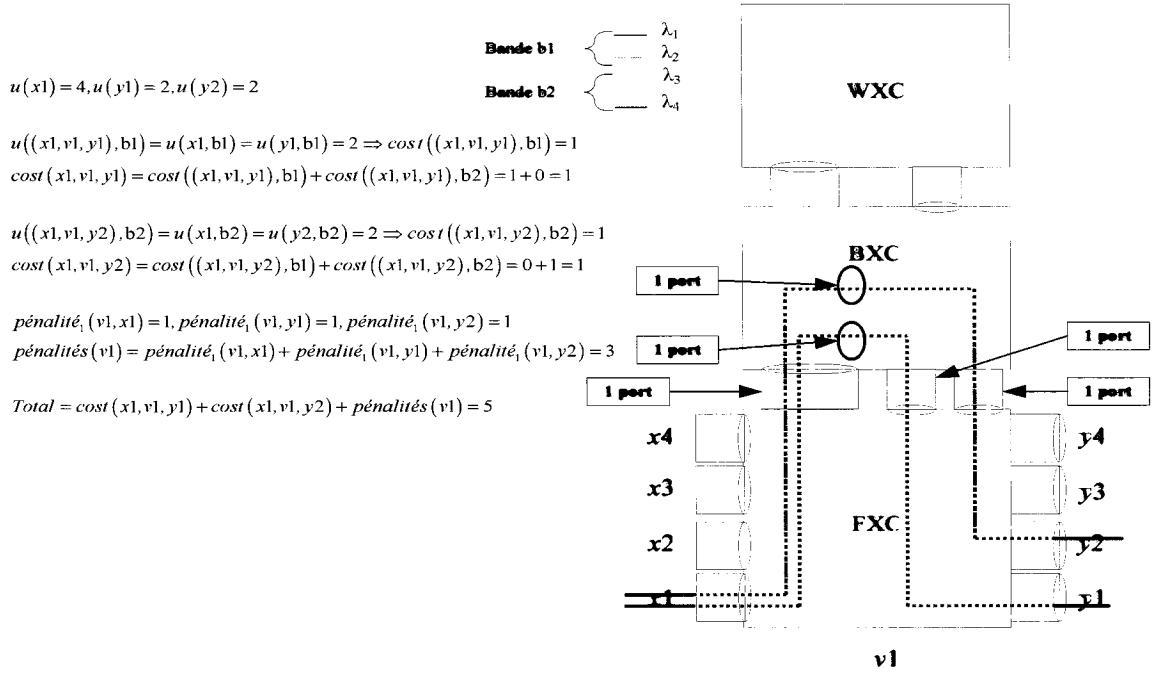
$$pénalités_3(v) = \sum_{a \in A_v} pénalité_3(v, a) + \sum_{a \in A_v} \sum_{b \in \{1, \dots, B\}} pénalité_3(v, a, b)$$

$$\text{où } \begin{cases} pénalité_3(v, a) = 1 & \text{si } u(0, v, a) \neq u(a) \\ pénalité_3(v, a) = 0 & \text{sinon} \end{cases}$$

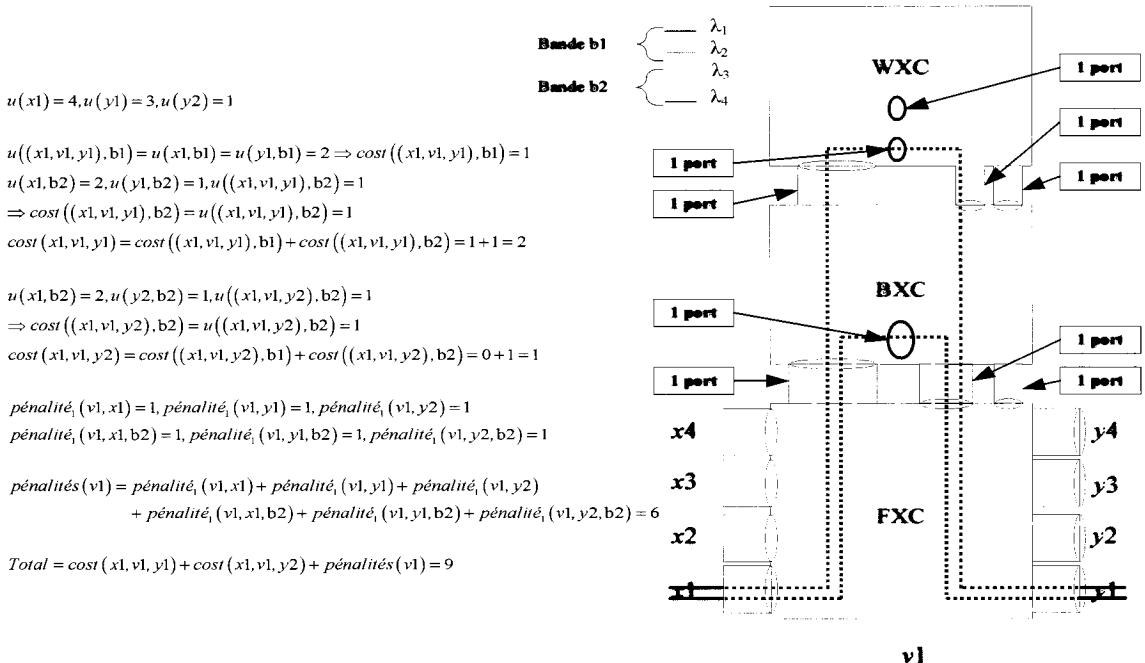
$$\begin{cases} \text{pénalité}_3(v, a, b) = 1 & \text{si } u((0, v, a), b) \neq u(a, b) \text{ et si } \text{pénalité}_3(v, a) = 1 \\ \text{pénalité}_3(v, a, b) = 0 & \text{sinon} \end{cases}$$

Une pénalité correspond en fait à l'utilisation d'un port interne au niveau d'un nœud. Ainsi, une commutation de fibre ne nécessite aucun port interne. Une commutation de bande pour un trafic qui passe à travers un nœud utilise deux ports internes. L'insertion (extraction) d'un trafic commuté bande au niveau d'un nœud nécessite l'utilisation d'un seul port interne. Une commutation individuelle de longueurs d'onde pour un trafic qui passe à travers un nœud utilise quatre ports internes par chemin optique. L'insertion (extraction) d'un trafic dont les longueurs d'onde sont commutées individuellement nécessite l'utilisation de deux ports internes par chemin optique. La Figure 3.2 illustre ce qui se passe réellement au niveau d'un commutateur, représenté par un nœud du multigraphe  $G_F$ , et fait le lien avec nos équations mathématiques pour le calcul du nombre de ports. Les ports internes permettent en fait de passer d'une couche à une autre. Les autres ports sont utilisés pour la commutation des fibres, des bandes d'onde ou des longueurs d'onde. Les différents types de commutation sont illustrés.





(b) commutation de bande



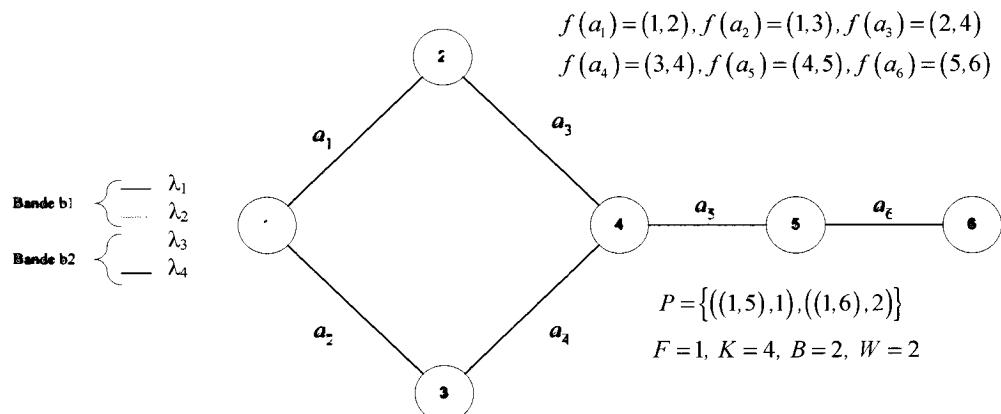
(c) commutation lambda

(commutation individuelle de longueurs d'onde)

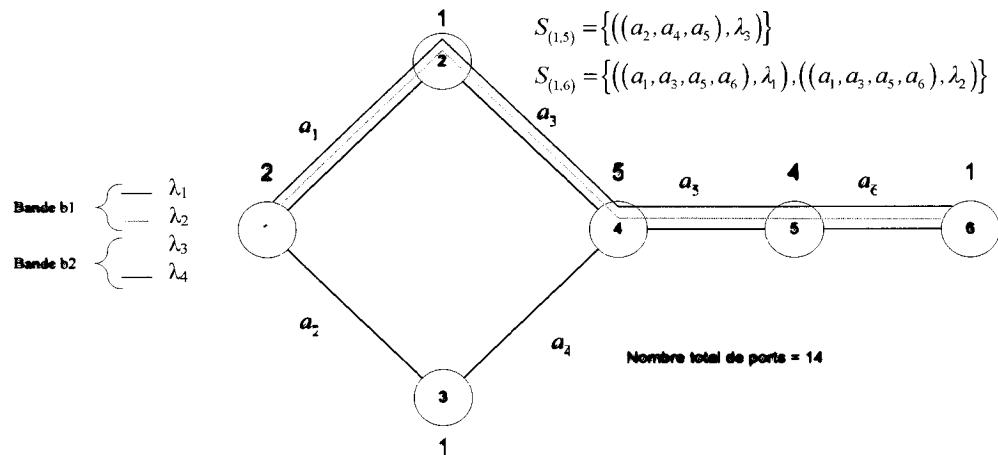
Figure 3.2 Calcul du nombre de ports d'un commutateur

### 3.1.2. Exemple illustratif

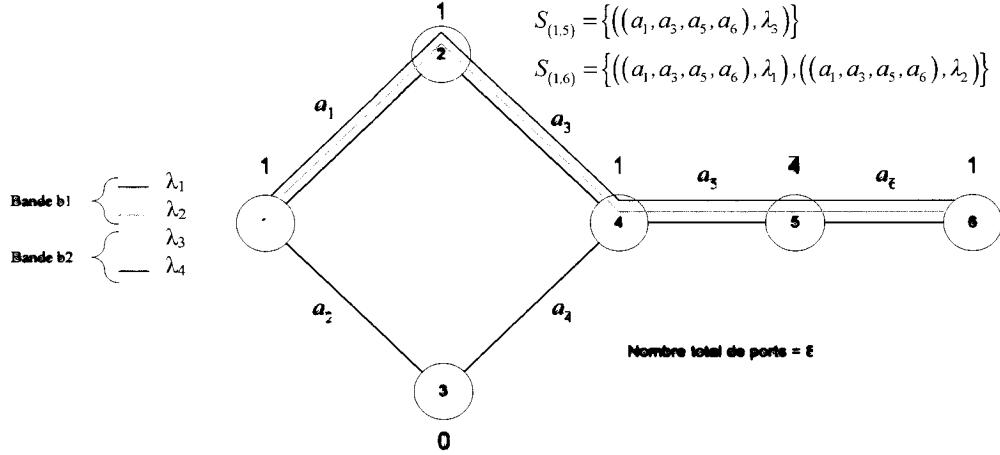
La Figure 3.3 est une illustration du problème MIN-MRWA. La topologie du réseau est représentée à la Figure 3.3a. Chaque lien du réseau contient une seule fibre. Une fibre contient quatre longueurs d'onde réparties en deux bandes b1 ( $\lambda_1, \lambda_2$ ) et b2 ( $\lambda_3, \lambda_4$ ). La paire de nœuds (1, 5) a une demande pour un chemin optique et la paire de nœuds (1, 6) une demande pour deux chemins optiques.



(a) topologie du réseau



(b) routage équilibré



(c) routage optimal

**Figure 3.3 Illustration du problème MIN-MRWA**

Un routage cherchant à équilibrer la charge au niveau des liens sélectionnerait (par exemple) le chemin  $(a_1, a_3, a_5, a_6)$  pour établir les deux chemins optiques demandés pour la paire  $(1, 6)$  et le chemin  $(a_2, a_4, a_5)$  pour établir le chemin optique demandé pour la paire  $(1, 5)$  (Figure 3.3b). Les chemins optiques pour la paire  $(1, 6)$  sont groupés dans la bande b1 tout le long du chemin sélectionné pour cette paire. Le chemin optique pour la paire  $(1, 5)$  utilise la bande b2 le long de la route qui lui est affectée. Au niveau du noeud 1, deux commutations de fibre sont nécessaires pour un total de 2 ports. Au niveau des noeuds 2 et 3, une commutation de fibre est requise pour un total de 1 port pour chacun d'eux. Au niveau du noeud 4, les trois chemins optiques se combinent dans la fibre de sortie  $a_5$ . Deux commutations de bande sont nécessaires. Trois ports internes sont également utilisés. Les ports du noeud 4 sont donc au nombre de 5. Deux commutations de bande sont nécessaires au niveau du noeud 5. Les chemins optiques qui aboutissent à ce noeud requièrent 2 ports dont un port interne. La deuxième commutation de bande nécessite 3 ports dont un port interne partagé avec les chemins optiques qui aboutissent au noeud. Ce port interne ne sera comptabilisé qu'une fois. Les ports du noeud 5 sont donc au nombre de 4. Au niveau du noeud 6, un seul port est requis pour une commutation de fibre. Le nombre de ports globalement dans le réseau est de 14.

Une solution optimale, montrée à la Figure 3.3c, consiste à router les quatre chemins optiques de façon à ce qu'ils se partagent les arcs  $a_1$  et  $a_3$  (ou  $a_2$  et  $a_4$ ). Le nombre de ports est réduit à 8, ce qui représente une réduction de près de 43%.

### 3.2. Algorithmes proposés

Dans cette section, nous présenterons le schéma général de l'algorithme que nous proposons. Nous détaillerons par la suite ses différentes parties. Il s'agit d'un algorithme de routage statique qui sera exécuté hors ligne au cours de la phase de planification par exemple.

#### 3.2.1. Schéma général

Le problème du routage et d'affectation de longueurs d'onde dans les réseaux multigranulaires pour des demandes de trafic statique a été classé NP-complet dans la littérature [5]. Pour des réseaux de petite taille, la solution optimale peut être trouvée sans grande difficulté. Mais, pour des réseaux de grande taille, celle-ci ne peut être trouvée par une méthode exacte dans un temps raisonnable. Des heuristiques fournissant une bonne solution sont alors souvent utilisées.

Dans ce mémoire, nous proposons d'utiliser la métaheuristique de recherche taboue (RT) pour trouver une solution permettant de diminuer le plus que possible le nombre de ports dans le réseau dans des temps de calcul raisonnables. Notre approche consiste à sélectionner une route (un chemin dans le graphe  $G$ ) par paire de nœuds et à modifier itérativement les routes du réseau. Il est à noter qu'un chemin  $\mu$  dans  $G$  du nœud  $s \in V$  au nœud  $d \in V$  est une séquence  $\mu = (s = v_1, v_2, \dots, d = v_n)$  de nœuds connectés par les arcs  $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)$  de  $G$ . La longueur du chemin  $\mu$  est égale à  $n-1$ , c'est-à-dire le nombre de nœuds dans la séquence moins un [29]. Lorsqu'une route est sélectionnée pour une paire, tout son trafic est acheminé le long de ce chemin (comme dans le BPHT) et nous noterons  $\mu(p)$  le chemin affecté à la paire  $p \in P$ . Donc, tous les chemins optiques d'une paire utilisent la même route dans le

réseau. Cette restriction devrait faciliter leur regroupement dans une même fibre. Nous appellerons notre méthode *TabouMRWA*.

L'espace de recherche d'une solution au problème est l'ensemble de toutes les combinaisons possibles de routes (dans  $G$ ) et de longueurs d'onde. Le nombre de routes pour une paire de nœuds et le nombre de paires pouvant être élevés, la taille de l'espace des solutions peut être très importante. De plus, bien que le fait de minimiser le nombre de sauts ne conduit pas nécessairement à un nombre minimal de ports dans le réseau, les routes des bonnes solutions, intuitivement, devraient être pour la plupart parmi les plus courts chemins entre les paires de nœuds (plus courts suivant le nombre de sauts) ou encore parmi les chemins dont la longueur est proche de celle des plus courts chemins. Nous avons alors décidé que l'espace à explorer serait composé des  $k$  ( $k > 0$ ) plus courts chemins non nécessairement optimaux pour chacune des paires ayant une demande de trafic non nulle. Il est à noter qu'il est possible qu'il existe moins de  $k$  chemins entre une paire de nœuds. La valeur de  $k$  doit être choisie de façon à ce que la taille de l'espace de recherche ne soit pas trop grande.

Soit  $Z \subseteq \mathfrak{S}$  l'espace des solutions du problème. Une solution  $z \in Z$  est l'ensemble des paires  $P$  à chacune desquelles ont été affectés un chemin dans  $G$  et un ensemble de longueurs d'onde le long de ce chemin afin de satisfaire la demande de trafic de la paire. (En d'autres, une solution est un ensemble de chemins optiques établis afin de satisfaire toutes les demandes de trafic du réseau). Le voisinage de  $z$  est l'ensemble des solutions obtenues en changeant la route d'une paire à la solution courante et en réaffectant des longueurs d'onde disponibles le long du chemin sélectionné. Le nombre de changements possibles (la taille du voisinage de  $z$ ) est donné par  $\sum_{p \in P} k_p$  avec  $k_p = \min(k_p^{\max}, k)$ ,  $k_p^{\max}$  représentant le nombre maximal de chemins entre la paire de nœuds  $p$ . Si  $k_p^{\max} \geq k$   $\forall p \in P$ , la taille du voisinage est donnée par  $k \cdot |P|$ .

La liste taboue peut être construite en déclarant taboue, pour un certain nombre d'itérations, la paire dont le changement de route détériore le moins ou améliore le plus le coût de la solution courante. En d'autres termes, la route d'une paire déclarée taboue

ne pourra pas être changée pendant un certain nombre d'itérations. La longueur de la liste taboue est alors donnée par le nombre d'itérations pendant lesquelles une paire reste taboue. La liste taboue contiendra autant de configurations interdites que le nombre d'itérations durant lesquelles une paire sera maintenue taboue. Une deuxième approche consiste à mettre taboue la route elle-même (de la paire). En d'autres mots, la route d'une paire peut changer mais la route déclarée taboue ne pourra pas être choisie pendant un certain nombre d'itérations.

Le critère d'aspiration consiste à conserver une solution obtenue avec un déplacement utilisant une paire ou une route taboue, mais ayant un coût inférieur à celui de la meilleure solution trouvée depuis le début de la recherche. Le coût d'une solution est le nombre total de ports dans le réseau.

Le schéma général de l'algorithme est présenté à la Figure 3.4.

**Données:**

Entrées du problème ( $G, F, K, W, P$ );  
 Nombre maximal d'itérations:  $Max\_Iter$ ;  
 Nombre de plus courts chemins entre deux nœuds :  $k$ ;  
 Longueur liste taboue :  $LT$ ;

**Préliminaires :**

Déterminer  $k$  plus courts chemins entre toutes les paires de nœuds dont la demande est non nulle en utilisant l'algorithme de Yen;

**Algorithme RT :**

Trier les paires en ordre décroissant suivant ((longueur PCC)  $\times$  demande);

Trouver une solution initiale réalisable;

$iter = 1$ ;

**Tant que**  $iter \leq Max\_Iter$  **Faire**

    Déterminer le mouvement le plus favorable tout en considérant les paires ou les routes taboues et le critère d'aspiration;

    Évaluer la paire ou la route taboue pour la meilleure transformation;

**Si** le coût de la solution courante est inférieur à celui de la meilleure solution trouvée depuis le début **Alors**

        Conserver la solution courante comme étant la meilleure solution;

**Fin Si**

$iter++$ ;

**Fin Tant que**

Figure 3.4 Schéma général de l'algorithme TabouMRWA

### 3.2.2. Algorithme de Yen

L'algorithme de Yen [30] est un algorithme de déviation qui permet de déterminer uniquement des chemins sans cycle. Pour un entier positif  $k$ , l'algorithme détermine l'ensemble  $R_k(p) = \{\mu_1, \dots, \mu_k\}$  de  $k$  plus courts chemins non nécessairement optimaux entre une paire de nœuds  $p = (s, d)$  tel que :

- $\mu_t$  est sans cycle pour tout  $t \in \{1, \dots, k\}$
- $c(\mu_t) \leq c(\mu_{t+1})$  pour tout  $t \in \{1, \dots, k-1\}$

$c(\mu_t)$  représente le coût du chemin  $\mu_t$  dans  $G$  et correspond à la somme du coût de tous ses liens (arcs). Nous avons  $c(\mu_t) = \sum_{(i,j) \in \mu_t} c_{ij}$ , où  $c_{ij}$  est une valeur réelle représentant le coût de l'arc  $(i, j)$ . Lorsque le coût du chemin est la longueur de celui-ci, le coût des arcs du réseau est mis à 1.

Le chemin  $\mu_t$  est déterminé avant le chemin  $\mu_{t+1}$ , pour tout  $t \in \{1, \dots, k-1\}$ . Alors, pour chaque nœud  $v'_i$  du chemin  $\mu_t$  à analyser, le plus court chemin  $\mu$  qui dévie de  $\mu_t$  à partir de  $v'_i$  est calculé. Le chemin sans cycle  $\mu$  est dit parent de  $\mu_t$  et le nœud  $v'_i$  est son nœud de déviation noté  $dev(\mu)$ . Pour calculer uniquement des chemins sans cycle, les nœuds du sous-chemin  $sub_{\mu_t}(s, v'_{i-1})$  ne doivent pas être répétés,  $sub_{\mu_t}(s, v'_{i-1})$  étant un chemin coïncidant avec  $\mu_t$  de  $s$  à  $v'_{i-1}$  ( $s$  et  $v'_{i-1}$  sont les nœuds source et destination du sous-chemin). Ces nœuds sont donc temporairement enlevés du réseau. De plus, l'arc  $(v'_i, v'_{i+1})$  et les arcs  $(v'_i, j)$  des chemins  $\mu_1, \dots, \mu_{i-1}, j \in V$  sont temporairement supprimés du réseau. Le plus court chemin de  $v'_i$  à  $d$  dans le réseau modifié résultant est ensuite déterminé. Soit  $q$  le plus court chemin trouvé. Le chemin  $\mu$  est la concaténation des deux chemins  $sub_{\mu_t}(s, v'_i)$  et  $q$ , c'est-à-dire le chemin de  $s$  à  $d$  formé de  $sub_{\mu_t}(s, v'_i)$  suivi de  $q$ . Il est noté  $cat(sub_{\mu_t}(s, v'_i), q)$ . Il est ajouté à la liste des chemins candidats maintenue par l'algorithme. Le nœud de déviation du chemin  $\mu$  est le nœud  $v'_i$ . À chaque itération, le

plus court chemin de la liste des candidats est tiré (et enlevé de la liste) et représente le nouveau plus court chemin calculé. La Figure 3.5 décrit l'algorithme.

```

 $R_k(p) = \{ \};$ 
 $\mu = \text{plus court chemin de } s \text{ à } d \text{ en utilisant Dijkstra};$ 
 $dev(\mu) = s;$ 
 $X = \{\mu\};$ 
 $t = 0;$ 

Tant que  $X \neq \{ \}$  Et  $t \leq k$  Faire
     $t = t + 1;$ 
     $\mu_t = \min(X);$ 
     $X = X - \{\mu_t\};$ 
     $R_k(p) = R_k(p) \cup \{\mu_t\};$ 

    Pour  $v'_i \in \{dev(\mu_t), \dots, v'_{t-1}\}$  Faire
        Supprimer  $sub_{\mu_t}(s, v'_{i-1});$ 
        Supprimer  $(v'_i, v'_{i+1});$ 
        Supprimer les arcs  $(v'_i, j) \in \mu_1, \dots, \mu_{t-1}, j \in V;$ 
         $q = Dijkstra(v'_i, d);$ 
         $\mu = cat(sub_{\mu_t}(s, v'_i), q);$ 
         $dev(\mu) = v'_i;$ 
         $X = X \cup \{\mu\};$ 

    Fin Pour
Fin Tant que

```

**Figure 3.5 Algorithme de Yen**

### 3.2.3. Tri des paires

Après avoir déterminé les  $k$  plus courts chemins pour chacune des paires, le poids de chacune des paires est calculé. Le poids d'une paire est égal à la longueur du plus court chemin entre les nœuds de la paire multipliée par la demande de celle-ci. L'ensemble  $P$  des paires de demande non nulle est alors trié en ordre décroissant suivant

leur poids. L'idée ici est de trouver un compromis entre la longueur d'un chemin et la demande. Nous nous basons ici sur les idées de Cao et al. [5]. Intuitivement, il est mieux d'affecter d'abord des longueurs d'onde aux chemins les plus longs afin de maintenir la continuité de longueur d'onde dans les réseaux à routage de longueur. Mais en plus, pour réduire le nombre de ports dans les réseaux multigranulaires, il est mieux d'affecter des longueurs d'onde d'abord aux chemins qui ont la demande la plus élevée de façon à augmenter la probabilité de commuter les longueurs d'onde au niveau fibre et/ou bande.

### 3.2.4. Solution initiale

La solution initiale, si elle est bien choisie, peut contribuer à améliorer le temps de convergence vers une bonne solution et par conséquent réduire le coût en temps de la recherche. En se basant sur le fait qu'intuitivement les routes des bonnes solutions devraient se trouver parmi les  $k$  plus courts chemins (non nécessairement optimaux), la méthode que nous proposons pour trouver une solution initiale tentera d'abord d'affecter le plus court chemin à chacune des paires de noeuds. Une fois une route sélectionnée, l'algorithme d'affectation de longueurs d'onde, le *WA-MF*, que nous utilisons et que nous présenterons par la suite, recherche des longueurs d'onde disponibles et les affecte à la paire courante. Si toute la demande de trafic n'a pu être satisfaite, une autre route est choisie. Ce processus continue jusqu'à ce qu'à obtention d'une route où tout le trafic peut être acheminé. Les routes sont choisies parmi les  $k$  plus courts chemins de la paire triés en ordre croissant suivant la longueur du chemin. Dans le cas où une route ne peut pas être trouvée, la solution est déclarée irréalisable. Il faudrait alors utiliser un autre algorithme capable de fournir une solution réalisable. Cependant dans le contexte du problème que nous traitons, ce cas ne devrait se présenter que rarement. Comme expliqué dans les chapitres précédents, la multigranularité dans les réseaux optiques a été proposée en raison des avancées de la technologie permettant le multiplexage de plusieurs centaines de longueurs d'onde par fibre. Le problème dans ce contexte est plutôt de gérer les ressources énormes du réseau. Le nombre de longueurs d'onde sur un

lien devrait être suffisant pour permettre de satisfaire toutes les demandes de connexion. La Figure 3.6 décrit l'algorithme pour la recherche d'une solution initiale.

**Pour chaque**  $p \in P$  **Faire**

    Récupérer les  $k$  plus courts chemins de  $p$ ,  $R_k(p)$ ;

**Tant que**  $t \leq k$  **Et**  $\text{stop} == \text{faux}$

        Affecter la route  $\mu_t \in R_k(p)$  à  $p$ ;  
        Affecter des longueurs d'onde à  $p$   
        en utilisant l'algorithme *WA-MF*;

**Si**  $WA-MF(p) == \text{vrai}$  **Alors**

$S = S \cup \{p\}$ ;  
             $\text{stop} = \text{vrai}$ ;

**Sinon**

$t++$ ;

**Fin Si**

**Fin Tant que**

**Si**  $t == k$  **Et**  $\text{stop} == \text{faux}$  **Alors** // (aucune affectation n'a réussi)

        Déclarer la solution irréalisable;  
        Quitter();

**Fin Si**

    Déterminer le nombre de ports du réseau en utilisant la fonction  $\varphi$ ;

**Fin Pour**

**Figure 3.6 Recherche d'une solution initiale**

La Figure 3.7 présente l'algorithme d'affectation de longueurs d'onde à une paire. (Nous nous sommes inspirés de l'algorithme d'affectation de longueurs d'onde de Cao et al. [5]). Si la demande de la paire est supérieure ou égale à  $W$  chemins optiques, l'algorithme essaie d'abord d'affecter des bandes disponibles de façon à regrouper  $W$  chemins optiques dans une même bande. Il affecte la première bande disponible trouvée sur le chemin en cherchant à partir de la première fibre du lien (la fibre numérotée 1). Il essaie d'affecter le plus de bandes que possible avant d'essayer d'affecter des longueurs

d'onde individuelles. Dans le cas de la recherche d'une longueur d'onde, tout comme dans le cas de la recherche d'une bande disponible, l'algorithme affecte la première longueur d'onde disponible trouvée sur le chemin en commençant par la première fibre. Lors de la recherche d'une bande ou d'une longueur d'onde, au niveau de chaque lien, plusieurs fibres peuvent être inspectées. Il est donc possible que des numéros de fibre différents soient utilisés le long du chemin pour une longueur d'onde ou une bande. Une autre possibilité serait d'utiliser le même numéro de fibre le long du chemin afin d'accélérer la recherche.

**Tant que**  $demande(p) \geq W$

    Trouver une bande  $b$  disponible le long du chemin en commençant par la fibre 1;

**Si** une bande n'a pas pu être trouvée **Alors**  
        Quitter la boucle;

**Fin Si**

    Affecter  $b$  à la paire  $p$ ;  
     $demande(p) -= W$ ;

**Fin Tant que**

**Tant que**  $demande(p) > 0$

    Trouver une longueur d'onde  $\lambda$  disponible le long du chemin en commençant par la fibre 1;

**Si** une longueur d'onde n'a pas pu être trouvée **Alors**  
        Libérer les ressources affectées à  $p$ ;  
        Retourner *faux*;

**Fin Si**

    Affecter  $\lambda$  à la paire  $p$ ;  
     $demande(p) --$ ;

**Fin Tant que**

    Retourner *vrai*;

**Figure 3.7 Algorithme d'affectation de longueurs d'onde WA-MF**

### 3.2.5. Mouvement

À chaque itération, pour chacune des paires de  $P$ , les  $k$  plus courts chemins sont inspectés (sauf celui couramment affecté à la paire). Chaque changement de route représente un mouvement et implique une réaffectation de longueurs d'onde et/ou de bandes d'onde le long de ce chemin en utilisant l'algorithme *WA-MF*. Pour une configuration (solution)  $z$ , le fait d'appliquer un mouvement permet d'obtenir une configuration  $z'$  voisine de  $z$ . Après chaque itération, le mouvement ayant la meilleure performance est choisi. Pour déterminer la performance d'un mouvement, nous introduisons la variable entière *gain*. Cette variable mesure l'influence du mouvement sur le coût de la solution, c'est-à-dire le nombre de ports dans le réseau. Une valeur positive de la variable indique une détérioration de la solution correspondant à une augmentation du nombre de ports. Plus la valeur de la variable est élevée, moins le mouvement est performant. Une valeur négative indique une amélioration de la solution correspondant à une diminution du nombre de ports. Plus le gain est négatif, plus le mouvement est performant. À chaque itération, on choisit alors le mouvement dont le gain est le plus faible, qu'il soit positif ou négatif.

Pour calculer le gain d'un mouvement, il faut d'abord récupérer la somme des ports des nœuds traversés par la route courante de la paire avant et après la désaffectation de cette route et des ressources utilisées (longueurs d'onde et/ou bandes d'onde) le long de celle-ci. Soient  $\beta_1$  la somme avant la désaffectation et  $\gamma_1$  la somme après celle-ci. Par la suite, on récupère la somme des ports des nœuds traversés par la nouvelle route avant et après l'affectation de longueurs d'onde et/ou de bandes d'onde le long du chemin. Soient  $\beta_2$  la somme avant l'affectation et  $\gamma_2$  la somme après. Le gain se calcule alors ainsi :  $gain = (\gamma_1 - \beta_1) + (\gamma_2 - \beta_2)$ . On peut réécrire la relation de la façon suivante :  $gain = (\gamma_1 + \gamma_2) - (\beta_1 + \beta_2)$ . La Figure 3.8 décrit les étapes d'un mouvement.

```

//Sauvegarder l'état courant de la paire (route, longueurs d'onde,..)
paireTemp = p;

//Récupérer la route courante de p,  $\mu(p)$ 
route =  $\mu(p)$ ;

//Récupérer la somme des ports des nœuds de la route
 $\beta_1 = RecupererNombrePorts(route)$ ;

//Libérer les ressources utilisées par p
LibererRessources(p);

//Récupérer la somme des ports des nœuds de la route
 $\gamma_1 = RecupererNombrePorts(route)$ ;

//Récupérer la somme des ports des nœuds de la nouvelle route,  $\mu_i \in R_k(p), \mu_i \neq \mu(p)$ 
 $\beta_2 = RecupererNombrePorts(\mu_i)$ ;

//Affecter la route  $\mu_i$  à la paire p
setRoute( $\mu_i, p$ );

//Affecter des longueurs d'onde et/ou des bandes sur la nouvelle route
WA - MF(p);

//Récupérer la somme des ports des nœuds de  $\mu_i$ ,
 $\gamma_2 = RecupererNombrePorts(\mu_i)$ ;

//Calcul du gain
gain = ( $\gamma_1 + \gamma_2$ ) - ( $\beta_1 + \beta_2$ );

//Libérer les ressources utilisées par p
LibererRessources(p);

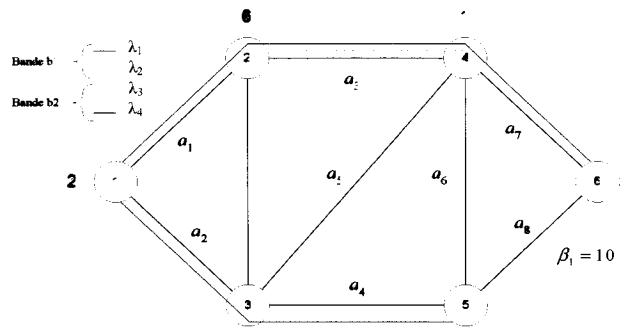
//Restaurer l'état de la paire avant le mouvement
p = paireTemp;

//RésERVER les ressources de la paire
ReserverRessources(p);

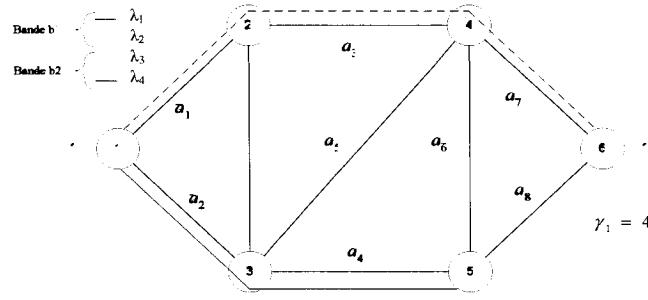
```

Figure 3.8 Étapes d'un mouvement

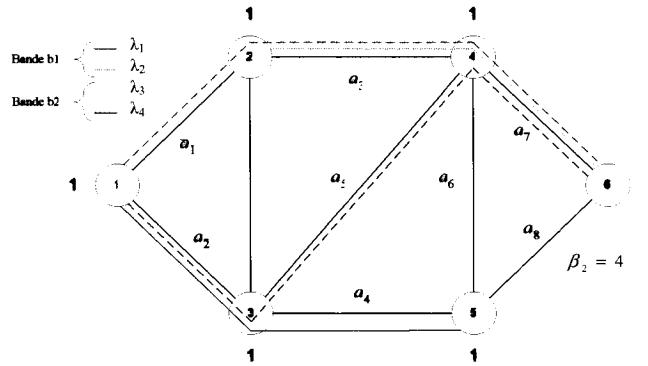
La Figure 3.9 illustre les étapes d'un mouvement (changement de route de la paire (1,6)). Dans le réseau, une seule fibre est utilisée par lien. Chaque fibre possède deux bandes  $b1$  et  $b2$  de deux longueurs d'onde chacune,  $(\lambda_1, \lambda_2)$  et  $(\lambda_3, \lambda_4)$  respectivement. Il existe une demande pour un chemin optique entre les paires (1, 5), (1, 6) et (2, 6). Soient  $\mu_1(p_{1,6}) = (1, 2, 4, 6)$  et  $\mu_2(p_{1,6}) = (1, 3, 4, 6)$  deux chemins entre la paire de nœuds (1,6),  $\mu(p_{1,5}) = (1, 3, 5)$  un chemin entre la paire de nœuds (1,5) et  $\mu(p_{2,6}) = (2, 4, 6)$  un chemin entre la paire de nœuds (2,6). La Figure 3.9a montre les chemins optiques établis dans le réseau :  $(\eta_{1,5}, \lambda_1)$ ,  $(\eta_{1,6}^1, \lambda_1)$  et  $(\eta_{2,6}, \lambda_2)$  où  $\eta_{1,5} = (a_2, a_4)$ ,  $\eta_{1,6}^1 = (a_1, a_3, a_7)$  et  $\eta_{2,6} = (a_3, a_7)$ . Avant de libérer le chemin optique  $(\eta_{1,6}^1, \lambda_1)$ , nous calculons la somme  $\beta_1$  des ports des nœuds 1, 2, 4 et 6. À la Figure 3.9b, le chemin optique  $(\eta_{1,6}^1, \lambda_1)$  est libéré et on calcule alors à nouveau la somme  $\gamma_1$  des ports des nœuds 1, 2, 4 et 6. Nous calculons ensuite la somme  $\beta_2$  des ports des nœuds 1, 3, 4 et 6 de la route  $\mu_2(p_{1,6})$  qui sera affectée à la paire (1,6), comme montré à la Figure 3.9c. L'étape suivante consiste à affecter une longueur d'onde disponible à la paire (1,6) le long du chemin  $\mu_2(p_{1,6})$ . La Figure 3.9d indique l'établissement du chemin optique  $(\eta_{1,6}^2, \lambda_3)$  où  $\eta_{1,6}^2 = (a_2, a_5, a_7)$ . Nous calculons alors la somme  $\gamma_2$  des ports des nœuds 1, 3, 4 et 6. Le gain est alors déterminé.



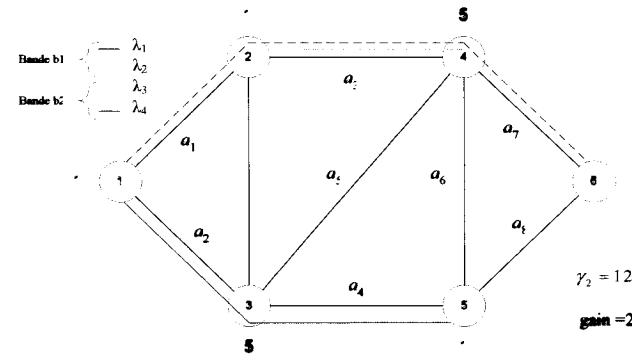
(a) Somme des ports des nœuds 1, 2, 4, 6 avant la libération des ressources utilisées par la paire (1,6)



(b) Somme des ports des nœuds 1, 2, 4, 6 après la libération des ressources utilisées par la paire (1,6)



(c) Somme des ports des nœuds 1, 3, 4, 6 avant l'affectation de ressources à la paire (1,6) sur la nouvelle route



(d) Somme des ports des nœuds 1, 3, 4, 6 après l'affectation de ressources à la paire (1,6) sur la nouvelle route

**Figure 3.9 Illustration des étapes d'un mouvement**

### 3.2.6. Mise à jour du nombre de ports au niveau d'un nœud

Lors d'un changement de route pour une paire, les longueurs d'onde et bandes d'onde associées à celle-ci sont libérées et d'autres longueurs d'onde et/ou bandes d'onde sont affectées le long de la nouvelle route. L'affectation et la désaffectation de longueurs d'onde et/ou de bandes d'onde impliquent respectivement une augmentation et une diminution du nombre de longueurs d'onde utilisées sur un ou plusieurs coudes au niveau de chacun des nœuds affectés. Ces deux opérations sont susceptibles de modifier le nombre de ports de chacun des nœuds. Cependant, la mise à jour du nombre de ports d'un nœud n'est pas triviale. La variation du nombre de longueurs d'onde utilisées sur un coude  $(x, v, y) \in \mathcal{T}_v$  est non seulement susceptible de modifier le coût de celui-ci, mais aussi celui d'un coude partageant l'arc entrant  $x$  et celui d'un coude partageant l'arc sortant  $y$ . De plus, la variation du nombre de longueurs d'onde n'est pas proportionnelle à celle du nombre de ports. Nous avons alors proposé quatre algorithmes de mise à jour du nombre de ports au niveau d'un nœud : le premier et le second respectivement dans le cas de l'affectation et de la désaffectation d'une longueur d'onde, le troisième et le quatrième respectivement dans le cas de l'affectation et de la désaffectation d'une bande d'onde. Ces algorithmes comprennent deux grandes étapes. La première consiste à déterminer l'impact de la variation du nombre de longueurs d'onde utilisées au niveau d'un coude sur le coût des autres coudes partageant un arc avec celui-ci. La seconde étape consiste à mettre à jour si possible le coût du coude dont le nombre de longueurs d'onde utilisées varie. A chacune des étapes, la variation du nombre de ports internes est aussi déterminée. Pour bien comprendre ces algorithmes, il est nécessaire de se rappeler l'existence de trois types de commutation et des conditions indiquant le type d'une commutation. Un coude  $(x, v, y)$  indique une commutation de fibre si aucune longueur d'onde n'est utilisée sur les autres coudes utilisant l'arc  $x$  ou l'arc  $y$ . Dans le cas où la commutation ne peut être classée fibre, les bandes d'onde individuelles au niveau du coude sont considérées. Une commutation de bande sur la bande  $b_i$  où  $1 \leq i \leq B$  est possible si aucune longueur d'onde n'est utilisée sur cette bande (bande  $b_i$ ) au niveau des

autres coudes utilisant l'arc  $x$  ou l'arc  $y$ . Dans le cas contraire, les longueurs d'onde de la bande  $b_i$  sont considérées individuellement.

La Figure 3.10 décrit le premier algorithme de mise à jour dans le cas de chemins optiques passant à travers un nœud. La première étape de l'algorithme est présentée à la Figure 3.10a. La longueur d'onde affectée à la paire est commutée de l'arc  $x$  à l'arc  $y$  et fait partie de la bande  $b$ . Le nombre de longueurs d'onde utilisées sur le coude  $(x, v, y)$  au niveau de la bande  $b$  est noté  $u((x, v, y), b)$  et celui au niveau du coude lui-même est noté  $u(x, v, y)$ . Dans cette étape, une modification du nombre de ports n'est possible que si la longueur d'onde affectée est la première utilisée sur la bande  $b$  au niveau du coude, c'est-à-dire si  $u((x, v, y), b)$  vaut un après son incrémentation. C'est le seul cas où les conditions indiquant le type d'une commutation sont susceptibles de changer pour les coudes utilisant l'arc  $x$  ou l'arc  $y$  (cf. 3-1 à 3-18). Lorsque ce cas se produit, une commutation de fibre (existante) pour un coude  $(x, v, j)$  (ou  $(i, v, y)$ ) n'est plus possible. Une réévaluation du coût de ce dernier et l'utilisation de deux ports internes sont alors nécessaires. Toutes les longueurs d'onde commutées de l'arc  $x$  (arc  $i$ ) à l'arc  $j$  (arc  $y$ ) sur une bande différente de  $b$  le seront suivant une commutation de bande. Si des longueurs d'onde sont utilisées sur la bande  $b$  de  $x$  ( $i$ ) vers  $j$  ( $y$ ), elles seront commutées individuellement conduisant à l'utilisation de deux ports internes additionnels. Si une commutation de fibre pour un coude partageant l'arc  $x$  (l'arc  $y$ ) n'est pas trouvée, une commutation de bande sur  $b$  pour un coude partageant l'arc  $x$  (l'arc  $y$ ) est recherchée. Dans le cas où la recherche est fructueuse, la commutation de bande trouvée n'est plus possible. Les longueurs d'onde de la bande  $b$  du coude correspondant doivent être alors commutées individuellement, provoquant une augmentation du coût de celui-ci et l'utilisation de deux ports internes additionnels. Ainsi, une première utilisation de la bande  $b$  du coude  $(x, v, y)$  est susceptible de conduire à une augmentation du nombre de ports en affectant d'autres commutations. Il est à noter aussi qu'une commutation de fibre n'est pas possible s'il ne s'agit pas de la première utilisation du coude  $(x, v, y)$  lui-

même. Il n'est pas nécessaire dans ce cas de vérifier l'existence d'une commutation de fibre pour un coude partageant l'arc  $x$  ou l'arc  $y$ .

```

 $u(x,v,y)++ ; u((x,v,y),b)++ ;$ 

Si  $u((x,v,y),b) = 1$  Alors
    Si  $u(x,v,y) = 1$  Alors
         $(x,v,j) = TrouverCommutationFibreSurX()$  ;
         $(i,v,y) = TrouverCommutationFibreSurY()$  ;
    Fin Si
    Si  $(x,v,j)$  est trouvé Alors
         $cost(x,v,j) = CalculerCout(x,v,j)$  ;  $nbPorts += cost(x,v,j) - 1$  ;
         $pénalité_1(x) = 1$  ;  $pénalité_1(j) = 1$  ;  $nbPorts += 2$  ;
        Si  $u((x,v,j),b) \neq u(x,b)$  ou  $u((x,v,j),b) \neq u(j,b)$  Alors
             $pénalité_1(x,b) = 1$  ;  $pénalité_1(j,b) = 1$  ;  $nbPorts += 2$  ;
    Fin Si
Sinon
     $(x,v,j) = TrouverCommutationBandeSurX(b)$  ;
    Si  $(x,v,j)$  est trouvé Alors
         $cost((x,v,j),b) = CalculerCout((x,v,j),b)$  ;  $nbPorts += cost((x,v,j),b) - 1$  ;
         $pénalité_1(x,b) = 1$  ;  $pénalité_1(j,b) = 1$  ;  $nbPorts += 2$  ;
    Fin Si
Fin Si
Si  $(i,v,y)$  est trouvé Alors
     $cost(i,v,y) = CalculerCout(i,v,y)$  ;  $nbPorts += cost(i,v,y) - 1$  ;
     $pénalité_1(i) = 1$  ;  $pénalité_1(y) = 1$  ;  $nbPorts += 2$  ;
    Si  $u((i,v,y),b) \neq u(i,b)$  ou  $u((i,v,y),b) \neq u(y,b)$  Alors
         $pénalité_1(i,b) = 1$  ;  $pénalité_1(y,b) = 1$  ;  $nbPorts += 2$  ;
    Fin Si
Sinon
     $(i,v,y) = TrouverCommutationBandeSurY(b)$  ;
    Si  $(i,v,y)$  est trouvé Alors
         $cost((i,v,y),b) = CalculerCout((i,v,y),b)$  ;  $nbPorts += cost((i,v,y),b) - 1$  ;
         $pénalité_1(i,b) = 1$  ;  $pénalité_1(y,b) = 1$  ;  $nbPorts += 2$  ;
    Fin Si
Fin Si
Fin Si

```

(a) Impact sur le coût des coudes partageant un arc avec le coude en considération

```

Si  $u(x, v, y) == u(x)$  Et  $u(x) == u(y)$  Alors
  Si  $u(x, v, y) == 1$  Alors
     $nbPorts++;$ 
  Fin Si
Sinon Si  $u(x, v, y, b) == u(x, b)$  Et  $u(x, b) == u(y, b)$  Alors
  Si  $u(x, v, y, b) == 1$  Alors
     $nbPorts++;$ 
  Si  $u(x) == 1$  Alors
     $pénalité_1(x) = 1; nbPorts++;$ 
  Fin Si
  Si  $u(y) == 1$  Alors
     $pénalité_1(y) = 1; nbPorts++;$ 
  Fin Si
Fin Si
Sinon
   $nbPorts++;$ 
  Si  $u(x) == 1$  Alors
     $pénalité_1(x) = 1; nbPorts++;$ 
  Fin Si
  Si  $u(x, b) == 1$  Alors
     $pénalité_1(x, b) = 1; nbPorts++;$ 
  Fin Si
  Si  $u(y) == 1$  Alors
     $pénalité_1(y) = 1; nbPorts++;$ 
  Fin Si
Fin Si

```

(b) Mise à jour du coût du coude en considération

**Figure 3.10 Mise à jour du nombre de ports d'un nœud  $v$  dans le cas de l'affectation d'une longueur d'onde**

La Figure 3.10b décrit la deuxième étape du premier algorithme. La variation du nombre de ports diffère selon le type de la commutation. L'algorithme détermine alors le type de la commutation pour le coude  $(x, v, y)$ . Si la commutation est de type fibre et si la longueur d'onde affectée est la première utilisée sur le coude correspondant, le nombre de ports augmente d'une unité. Dans le cas où il s'agit d'une commutation de

bande sur la bande  $b$  et si  $u((x, v, y), b)$  vaut un, le nombre de ports est d'abord incrémenté d'une unité. Puis, il augmente d'une unité si la longueur d'onde affectée à la paire est la seule utilisée sur l'arc  $x$ , indiquant l'utilisation d'un port interne. Le même raisonnement s'applique à l'arc  $y$ . Dans le cas d'une commutation individuelle de la longueur d'onde affectée à la paire, le nombre de ports est toujours augmenté d'une unité. L'algorithme vérifie ensuite si des ports internes doivent être comptabilisés en tenant compte des pénalités. Une pénalité sur la bande  $b$  de l'arc  $x$  est prise en compte si la longueur d'onde affectée est la seule utilisée sur cette bande, impliquant alors l'utilisation d'un port interne. Si, de plus, la longueur d'onde affectée est la seule utilisée sur l'arc  $x$ , on compte une pénalité sur celui-ci conduisant à l'utilisation d'un autre port interne. Le même raisonnement s'applique à l'arc  $y$ . Le nombre de ports du nœud augmente alors du nombre de ports internes considérés.

La Figure 3.11 décrit le deuxième algorithme de mise à jour où une longueur d'onde est désaffectée au niveau d'un nœud. La Figure 3.11a présente la première étape de l'algorithme. Dans cette étape, une modification du nombre de ports n'est possible que si la longueur d'onde désaffectée était la seule utilisée sur la bande  $b$  au niveau du coude  $(x, v, y)$ , c'est-à-dire si  $u((x, v, y), b)$  vaut zéro après sa décrémentation. C'est le seul cas où les conditions indiquant le type d'une commutation sont susceptibles de changer. Si ce cas est vérifié, l'algorithme de mise à jour recherche un coude  $(x, v, j)$  (ou  $(i, v, y)$ ) indiquant une commutation qui pourra être classée fibre. Si la recherche est fructueuse, le coût du coude trouvé vaut alors un et le nombre de ports diminue du coût précédent augmenté d'une unité. L'existence d'une commutation de fibre après la décrémentation de  $u((x, v, y), b)$  conduit nécessairement à la libération de deux ports internes et le nombre de ports diminue de deux unités. Si, de plus, une longueur d'onde (ou plus) est utilisée sur la bande  $b$  au niveau du coude  $(x, v, j)$  (ou  $(i, v, y)$ ), deux autres ports internes sont libérés, vu que cela signifie que la longueur d'onde désaffectée était commutée individuellement avant la décrémentation de  $u((x, v, y), b)$ . Le nombre de

ports est alors diminué de deux unités. Dans le cas où une commutation de fibre n'est trouvée pour aucun coude  $(x, v, j)$  (ou  $(i, v, y)$ ), l'algorithme recherche une commutation de bande sur la bande  $b$  pour un coude  $(x, v, j)$  (ou  $(i, v, y)$ ). Si une telle commutation est trouvée, le coût du coude correspondant est réévalué au niveau de la bande  $b$ . Le nombre de ports diminue du coût précédent de la bande augmenté d'une unité. Deux ports internes sont libérés et le nombre de ports du nœud est décrémenté encore de deux unités.

La Figure 3.11b décrit la deuxième étape du deuxième algorithme. La variation du nombre de ports diffère selon le type d'une commutation. L'algorithme détermine alors le type de la commutation pour le coude  $(x, v, y)$  avant la désaffectation de la longueur d'onde de la paire. Si la longueur d'onde désaffectée était la dernière utilisée au niveau du coude et des arcs correspondants, indiquant que la commutation était de type fibre, le nombre de ports diminue d'une unité. Dans le cas où il s'agissait d'une commutation de bande sur la bande  $b$  au niveau du coude, signifiant qu'aucune longueur d'onde n'est plus utilisée sur les arcs  $x$  et  $y$  au niveau de cette bande, le nombre de ports est d'abord décrémenté d'une unité. Puis, le nombre de ports diminue encore d'une unité si aucune longueur d'onde n'est plus utilisée sur l'arc  $x$ , indiquant la libération d'un port interne. Le même raisonnement s'applique à l'arc  $y$ , impliquant une nouvelle diminution du nombre de ports d'une unité. Dans le cas d'une commutation individuelle de la longueur d'onde désaffectée, le nombre de ports est toujours décrémenté d'une unité. L'algorithme vérifie ensuite si des ports internes sont libérés. Si aucune longueur d'onde n'est plus utilisée sur l'arc  $x$  au niveau de la bande  $b$ , un port interne est libéré. On libère un autre port interne si aucune longueur d'onde n'est plus utilisée sur l'arc  $x$  lui-même. Le même raisonnement s'applique à l'arc  $y$ . Le nombre de ports du nœud diminue alors du nombre de ports internes libérés.

```

 $u(x, v, y) -- ; u((x, v, y).b) -- ;$ 
Si  $u((x, v, y).b) = 0$  Alors
    Si  $u(x, v, y) == 0$  Alors
         $(x, v, j) = TrouverCommutationFibreSurX();$ 
         $(i, v, y) = TrouverCommutationFibreSurY();$ 
    Fin Si
    Si  $(x, v, j)$  est trouvé Alors
         $cost(x, v, j) = CalculerCout(x, v, j); nbPorts += 1 - cost(x, v, j);$ 
         $pénalité_1(x) = 0; pénalité_1(j) = 0; nbPorts -= 2;$ 
        Si  $u((x, v, j).b) > 0$  Alors
             $pénalité_1(x, b) = 0; pénalité_1(j, b) = 0; nbPorts -= 2;$ 
        Fin Si
    Sinon
         $(x, v, j) = TrouverCommutationBandeSurX(b);$ 
        Si  $(x, v, j)$  est trouvé Alors
             $cost((x, v, j), b) = CalculerCout((x, v, j), b); nbPorts += 1 - cost((x, v, j), b);$ 
             $pénalité_1(x, b) = 0; pénalité_1(j, b) = 0; nbPorts -= 2;$ 
        Fin Si
    Fin Si
    Si  $(i, v, y)$  est trouvé Alors
         $cost(i, v, y) = CalculerCout(i, v, y); nbPorts += 1 - cost(i, v, y);$ 
         $pénalité_1(i) = 0; pénalité_1(y) = 0; nbPorts -= 2;$ 
        Si  $u((i, v, y).b) > 0$  Alors
             $pénalité_1(i, b) = 0; pénalité_1(y, b) = 0; nbPorts -= 2;$ 
        Fin Si
    Sinon
         $(i, v, y) = TrouverCommutationBandeSurY(b);$ 
        Si  $(i, v, y)$  est trouvé Alors
             $cost((i, v, y), b) = CalculerCout((i, v, y), b); nbPorts += 1 - cost((i, v, y), b);$ 
             $pénalité_1(i, b) = 0; pénalité_1(y, b) = 0; nbPorts -= 2;$ 
        Fin Si
    Fin Si

```

(a) Impact sur le coût des coudes partageant un arc avec le coude en considération

```

Si  $u(x, v, y) == u(x)$  Et  $u(x) == u(y)$  Alors
    Si  $u(x, v, y) == 0$  Alors
         $nbPorts--;$ 
    Fin Si
Sinon Si  $u((x, v, y), b) == u(x, b)$  Et  $u(x, b) == u(y, b)$  Alors
    Si  $u((x, v, y), b) == 0$  Alors
         $nbPorts--;$ 
    Si  $u(x) == 0$  Alors
         $pénalité_1(x) = 0$  ;  $nbPorts--;$ 
    Fin Si
    Si  $u(y) == 0$  Alors
         $pénalité_1(y) = 0$  ;  $nbPorts--;$ 
    Fin Si
Fin Si
Sinon
     $nbPorts--;$ 
    Si  $u(x) == 0$  Alors
         $pénalité_1(x) = 0$  ;  $nbPorts--;$ 
    Fin Si
    Si  $u(x, b) == 0$  Alors
         $pénalité_1(x, b) = 0$  ;  $nbPorts--;$ 
    Fin Si
    Si  $u(y) == 0$  Alors
         $pénalité_1(y) = 0$  ;  $nbPorts--;$ 
    Fin Si
    Si  $u(y, b) == 0$  Alors
         $pénalité_1(y, b) = 0$  ;  $nbPorts--;$ 
    Fin Si
Fin Si

```

(b) Mise à jour du coût du coude en considération

**Figure 3.11 Mise à jour du nombre de ports d'un nœud  $v$  dans le cas de la désaffectation d'une longueur d'onde**

La Figure 3.12 décrit le troisième algorithme de mise à jour et la Figure 3.13 le quatrième. Ces algorithmes sont semblables aux deux premiers respectivement. Les principales différences résident dans le fait qu'à la première étape les deux derniers algorithmes ne recherchent que des commutations de type fibre et qu'à la deuxième étape une commutation de type lambda (commutation individuelle de longueurs d'onde)

n'est pas possible. Ce dernier cas n'est donc pas considéré. On peut noter aussi que la première utilisation d'un coude  $(x, v, y)$  est représentée par le fait que  $u(x, v, y)$  vaut  $W$  après son incrémentation.

```

 $u(x, v, y) += W ; u((x, v, y), b) += W ;$ 
Si  $u(x, v, y) == W$  Alors
     $(x, v, j) = TrouverCommutationFibreSurX()$  ;
    Si  $(x, v, j)$  est trouvé Alors
         $cost(x, v, j) = CalculerCout(x, v, j)$  ;  $nbPorts += cost(x, v, j) - 1$  ;
         $pénalité_1(x) = 1$  ;  $pénalité_1(j) = 1$  ;  $nbPorts += 2$  ;
    Fin Si
     $(i, v, y) = TrouverCommutationFibreSurY()$  ;
    Si  $(i, v, y)$  est trouvé Alors
         $cost(i, v, y) = CalculerCout(i, v, y)$  ;  $nbPorts += cost(i, v, y) - 1$  ;
         $pénalité_1(i) = 1$  ;  $pénalité_1(y) = 1$  ;  $nbPorts += 2$  ;
    Fin Si
Fin Si

```

(a) Impact sur le coût des coudes partageant un arc avec le coude en considération

```

Si  $u(x, v, y) == u(x)$  Et  $u(x) == u(y)$  Alors
    Si  $u(x, v, y) == W$  Alors
         $nbPorts ++$  ;
    Fin Si
Sinon
     $nbPorts ++$  ;
    Si  $u(x) == W$  Alors
         $pénalité_1(x) = 1$  ;
         $nbPorts ++$  ;
    Fin Si
    Si  $u(y) == W$  Alors
         $pénalité_1(y) = 1$  ;
         $nbPorts ++$  ;
    Fin Si
Fin Si

```

(b) Mise à jour du coût du coude en considération

**Figure 3.12 Mise à jour du nombre de ports d'un nœud  $v$  dans le cas de l'affectation d'une bande d'onde**

```

 $u(x, v, y) = W ; u((x, v, y), b) = W ;$ 
Si  $u(x, v, y) == 0$  Alors
     $(x, v, j) = TrouverCommutationFibreSurX();$ 
    Si  $(x, v, j)$  est trouvé Alors
         $cost(x, v, j) = CalculerCout(x, v, j) ;$ 
         $nbPorts += 1 - cost(x, v, j) ;$ 
         $pénalité_1(x) = 0 ; pénalité_1(j) = 0 ;$ 
         $nbPorts -= 2 ;$ 
    Fin Si
     $(i, v, y) = TrouverCommutationFibreSurY();$ 
    Si  $(i, v, y)$  est trouvé Alors
         $cost(i, v, y) = CalculerCout(i, v, y) ;$ 
         $nbPorts += 1 - cost(i, v, y) ;$ 
         $pénalité_1(i) = 0 ; pénalité_1(y) = 0 ;$ 
         $nbPorts -= 2 ;$ 
    Fin Si
Fin Si

```

(a) Impact sur le coût des coudes partageant un arc avec le coude en considération

```

Si  $u(x, v, y) == u(x)$  Et  $u(x) == u(y)$  Alors
    Si  $u(x, v, y) == 0$  Alors
         $nbPorts -- ;$ 
    Fin Si
Sinon
     $nbPorts -- ;$ 
    Si  $u(x) == 0$  Alors
         $pénalité_1(x) = 0 ;$ 
         $nbPorts -- ;$ 
    Fin Si
    Si  $u(y) == 0$  Alors
         $pénalité_1(y) = 0 ;$ 
         $nbPorts -- ;$ 
    Fin Si
Fin Si

```

(b) Mise à jour du coût du coude en considération

**Figure 3.13** Mise à jour du nombre de ports d'un nœud  $v$  dans le cas de la désaffectation d'une bande d'onde

### 3.2.7. Présentation globale de l'algorithme TabouMRWA

La Figure 3.14 présente l'algorithme TabouMRWA dans son ensemble.

```

Pour  $p \in P$  Faire
    Déterminer  $kPCC(p)$  ;
Fin Pour
Pour  $p \in P$  Faire
    Calculer  $Poids(p)$  ;
Fin Pour
    Trier( $P.poids$ ) ;
    meilleureCoutTrouve = SolutionInitiale() ;
    meilleureSolution =  $P$  ;
    itération = 1 ;

    Tant que  $itération \leq MaxIter$  Faire
        meilleurCout = INFINI ;
        Pour  $p \in P$  Faire
            paireTemp =  $p$  ;
             $\beta_1 = RecupererNombrePorts(\mu(p))$  ;
            LibérerRessources( $p$ ) ;
             $\gamma_1 = RecupererNombrePorts(\mu(p))$  ;
            Pour  $\mu_t \in R_k(p), \mu_t \neq \mu(p)$  Faire
                 $\beta_2 = RecupererNombrePorts(\mu_t)$  ;
                Si  $WA - MF(p) == vrai$  Alors
                     $\gamma_2 = RecupererNombrePorts(\mu_t)$  ;
                    gain =  $(\gamma_1 + \gamma_2) - (\beta_1 + \beta_2)$  ;
                    coutCourant = coutRéseau + gain ;
                    Si  $(coutCourant < meilleurCout$  et  $(tabouP(p) < itération$  ou
                         $coutCourant < meilleureCoutTrouve$  )) Alors
                        meilleurCout = coutCourant ;
                        paireMin = paireTemp ; etatMin = etat( $p$ ) ;
                    Fin Si
                    LibérerRessources( $p$ ) ;
                Fin Si
            Fin Pour
             $p = paireTemp$  ; RestaurerRessources( $p$ ) ;
        Fin Pour
        LibérerRessources(paireMin) ; etat(paireMin) = etatMin ;
        CalculerCoutRéseau() ; tabouP( $p$ ) = itération + listeTaboue ;
        Si  $meilleurCout < meilleureCoutTrouve$  Alors
            meilleureCoutTrouve = meilleurCout ; meilleureSolution =  $P$  ;
        Fin Si
        itération ++ ;
    Fin Tant que

```

Figure 3.14 Algorithme TabouMRWA

### 3.3. Analyse de la complexité de l'algorithme TabouMRWA

La complexité de l'algorithme que nous avons proposé est évaluée en calculant la complexité des principales étapes. L'algorithme commence par la recherche des  $k$  plus courts chemins entre toutes les paires de nœuds de demande non nulle. L'algorithme de Yen est utilisé afin d'y parvenir. La complexité de cet algorithme dans le pire cas est de l'ordre de  $O(kn(m + n \log n))$  où chaque chemin  $\mu_1, \dots, \mu_k$  nécessite l'analyse de  $n$  nœuds,  $n$  étant le nombre de nœuds du réseau et  $m$  le nombre de liens.

Dans la recherche d'une solution initiale, nous affectons à chaque paire la première route sur laquelle il est possible d'acheminer tout son trafic. À chaque sélection de route, l'algorithme *WA-MF* est utilisé pour affecter des longueurs d'onde à la paire. Cet algorithme cherche à affecter des bandes puis des longueurs d'onde. Le pire cas correspond à celui où aucune affectation de bandes d'onde n'est possible. Les longueurs d'onde doivent alors être affectées individuellement. Pour affecter une longueur d'onde, il faut d'abord en trouver une disponible le long du chemin. Dans le pire cas de la recherche d'une longueur d'onde, toutes les fibres de chacun des liens de la route sont inspectées de même que toutes les longueurs d'onde des fibres du premier lien. La complexité dans ce cas est  $O(\ell KF)$ , où  $\ell$  est la longueur du chemin. Le pire cas de la recherche d'une solution initiale correspond à celui où l'acheminement du trafic est possible sur la dernière route inspectée pour chacune des paires. De plus, les routes passent toutes par les  $n-1$  nœuds du réseau. Si  $T$  représente le trafic total à acheminer dans le réseau, c'est-à-dire  $T = \sum_{p \in P} \text{multiplicity}(p)$ , alors la complexité est donnée par :

$$O(Tk(n-1)KF) = O(TknKF).$$

Pour la boucle principale, il faut évaluer la complexité de chaque itération. À chaque itération, pour chaque paire, le nombre de ports des nœuds traversés par la route courante est susceptible d'être mis à jour après la désaffectation des ressources. Pour  $k-1$  routes, une mise à jour du nombre de ports des nœuds est possible après l'affectation et après la libération de ressources. Dans le pire cas, la mise à jour du nombre de ports s'effectue toujours et autant de fois que le nombre de longueurs d'onde affectées à la

paire, c'est-à-dire la demande de celle-ci notée  $multiplicity(p)$  pour  $p \in P$ . Le pire cas de l'algorithme de mise à jour lui-même correspond à celui où, à la première étape, aucune commutation de fibre ni de bande n'a été trouvée (ou encore une commutation de bande est trouvée après l'inspection de tous les coudes partageant un arc entrant ou sortant avec le coude en considération). La complexité de l'algorithme est alors donnée par  $O(\delta F)$ , où  $\delta$  est le degré du nœud (nombre de nœuds voisins), vu qu'à chaque recherche l'algorithme passe à travers tous les arcs sortants (entrants) pour un arc entrant (sortant). La complexité de l'affectation ou de la libération des ressources d'une paire s'écrit donc :  $O(multiplicity(p)\delta F)$ . Cependant, le pire cas inclut aussi que le réseau soit complètement maillé, impliquant un degré égal à  $n-1$  pour chaque nœud. La complexité devient :  $O(multiplicity(p)(n-1)F)$  ou  $O(multiplicity(p)nF)$ . Pour une paire, nous avons alors :  $O(multiplicity(p)nF) + O((k-1)multiplicity(p)nF) = O(multiplicity(p)knF)$ . Pour  $|P|$  paires, nous avons :  $O\left(\sum_{p \in P} multiplicity(p)knF\right) = O(TknF)$ . De plus, à chaque sélection de route, il faut affecter des longueurs d'onde à la paire en utilisant l'algorithme *WA-MF*. La complexité dans le pire cas pour une itération est donc :  $O(T(k-1)nKF) = O(TknKF)$ . La complexité pour  $M$  itérations (nombre maximal d'itérations) pour le pire cas est :  $O(MTknF) + O(MTknKF)$ .

En tenant compte du tri des paires avant la recherche d'une solution initiale, la complexité globale de l'algorithme *TabouMRWA* pour le pire cas est :

$$\begin{aligned}
 & O(kn(m + n \log n)) + O(TknKF) + O(MTknF) + O(MTknKF) + O(|P|^2) \\
 & = O(kn(m + n \log n)) + O(MTknF) + O(MTknKF) + O(|P|^2) \\
 & = O(kn(m + n \log n)) + O(MTknKF) + O(|P|^2).
 \end{aligned}$$

## CHAPITRE IV

### IMPLÉMENTATION ET RÉSULTATS

Dans ce chapitre, nous procéderons à l'évaluation de performance de l'algorithme de routage et d'affectation de longueurs d'onde que nous avons proposé au chapitre précédent. Dans un premier temps, nous allons décrire les détails de l'implémentation de l'algorithme, à savoir les structures de données utilisées ainsi que l'environnement de programmation. Par la suite, nous présenterons notre plan d'expérience et les données utilisées pour effectuer les simulations. Nous conclurons par une analyse des résultats obtenus.

#### 4.1. Structures de données utilisées

Les algorithmes ont été implémentés en C++ avec le logiciel Visual C++ 6.0. Nous avons utilisé une structure de programmation orientée objet. Celle-ci facilite la compréhension du code et simplifie la programmation.

##### 4.1.1. La classe Réseau

Elle représente la topologie physique du réseau à traiter. Ses principaux attributs sont les suivants :

- *nbrNoeuds* : le nombre de nœuds du réseau;
- *nbrLiens* : le nombre de liens du réseau;
- *Noeuds* : l'ensemble des nœuds du réseau. Il s'agit d'un vecteur contenant des pointeurs vers les nœuds créés;
- *Liens* : l'ensemble des liens du réseau. Il s'agit d'un vecteur contenant des pointeurs vers les liens créés.

La classe possède un constructeur qui prend en argument le nom du fichier de topologie, le nombre de fibres par lien  $F$ , le nombre de bandes par fibre et le nombre de longueurs d'onde  $W$ . Le constructeur crée les nœuds et les liens du réseau et les

initialise. Les informations nécessaires à la création de ces derniers sont lues dans le fichier de topologie. Il s'agit de la fonction la plus importante de la classe.

Le format du fichier de topologie est le suivant :

Ligne 1	<i>nbrNoeuds</i>		
Ligne 2 à <i>nbrLiens</i> + 1	<i>i</i>	<i>j</i>	<i>c<sub>ij</sub></i>

Les lignes 2 à *nbrLiens* + 1 représentent chacune un lien  $(i, j)$  et le coût associé. Le lien  $(j, i)$  lors de la lecture est automatiquement créé et possède le même coût.

Les autres fonctions importantes de la classe sont les suivantes :

- *Init()* : cette fonction initialise les nœuds et les liens du réseau;
- *CreerRessources()* : cette fonction crée la matrice de commutation des nœuds du réseau. Elle crée également les matrices qui indiquent l'utilisation des longueurs d'onde et des bandes sur les liens du réseau;
- *getNbrPorts()* : cette fonction retourne le nombre total de ports dans le réseau.

#### 4.1.2. La classe Nœud

Cette classe représente un nœud du réseau qui est un commutateur multigranulaire. Ses principaux attributs sont les suivants :

- *ID* : l'identifiant du nœud;
- *voisins* : l'ensemble des nœuds voisins du nœud. Il s'agit d'un vecteur contenant des pointeurs vers les nœuds voisins;
- *nbrVoisins* : le nombre de voisins du nœud;
- *F* : le nombre de fibres sur un lien;
- *B* : le nombre de bandes dans une fibre;
- *W* : le nombre de longueurs d'onde dans une bande;
- *K* : le nombre de longueurs d'onde dans une fibre;

- *liensI* : l'ensemble des liens entrant du nœud. Il s'agit d'un vecteur contenant des pointeurs vers les liens entrants;
- *liensO* : l'ensemble des liens sortant du nœud. Il s'agit d'un vecteur contenant des pointeurs vers les liens sortants;
- *nbrPorts* : le nombre total de ports du nœud;
- *FTB* : l'ensemble des ports pour le démultiplexage Fibre/Bandes. Il s'agit d'un tableau d'entiers prenant la valeur 0 ou 1. Une valeur de 0 indique que le port n'est pas utilisé et une valeur de 1 qu'il est utilisé;
- *BTF* : l'ensemble des ports pour le multiplexage Bandes/Fibre. Il s'agit d'un tableau d'entiers prenant la valeur 0 ou 1. Une valeur de 0 indique que le port n'est pas utilisé et une valeur de 1 qu'il est utilisé;
- *BTW* : l'ensemble des ports pour le démultiplexage Bande/Longueurs d'onde. Il s'agit d'une matrice dont la ligne représente la fibre entrante et la colonne la bande. Les éléments de la matrice sont des entiers prenant la valeur 0 ou 1. Une valeur de 0 indique que le port n'est pas utilisé et une valeur de 1 qu'il est utilisé;
- *WTB* : l'ensemble des ports pour le multiplexage Longueurs d'onde/Bande. Il s'agit d'une matrice dont la ligne représente la fibre sortante et la colonne la bande. Les éléments de la matrice sont des entiers prenant la valeur 0 ou 1. Une valeur de 0 indique que le port n'est pas utilisé et une valeur de 1 qu'il est utilisé;
- *commutateur* : la matrice de commutation. Il s'agit d'un tableau à trois dimensions. La ligne représente la fibre entrante, la colonne la fibre sortante et la troisième dimension la bande. Un élément de la matrice indique le nombre de longueurs d'onde en utilisation dans la bande;
- *tailleCom* : la taille de la matrice de commutation. Il s'agit du produit des deux premières dimensions uniquement;
- *sommeComX* : la somme des éléments d'un ligne de la matrice de commutation;
- *sommeComY* : la somme des éléments d'une colonne de la matrice de commutation;
- *coutCommutateur* : la matrice de coût des commutations.

Les principales méthodes de la classe sont les suivantes :

- *Init()* : cette fonction initialise le commutateur, les structures pour les ports intermédiaires et le nombre de ports;
- *ConfigurerCommutateurBande()* : cette fonction ajoute ou supprime une bande d'onde dans la matrice de commutation;
- *ConfigurerCommutateurLambda()* : cette fonction ajoute ou supprime une longueur d'onde dans la matrice de commutation;
- *CalculerCoutCommutation(x, y)* : cette fonction calcule le coût d'une commutation;
- *CalculerCoutCommutation(x, y, b)* : cette fonction calcule le coût d'une commutation au niveau d'une bande  $b$ ;
- *getNbrPorts()* : cette fonction retourne le nombre de ports du nœud.

#### 4.1.3. La classe Lien

Cette classe implémente un lien du réseau. Ses principaux attributs sont les suivants :

- *ID* : l'identifiant du lien;
- *idNoeud1* : l'identifiant du nœud de départ;
- *idNoeud2* : l'identifiant du nœud d'arrivée;
- *cost* : le coût du lien;
- *F* : le nombre de fibres sur le lien;
- *lambdas* : la matrice d'utilisation des longueurs d'onde. Cette matrice indique la disponibilité d'une longueur d'onde sur une fibre. Une valeur de 0 indique que la longueur d'onde n'est pas disponible et une valeur de 1 qu'elle est disponible;
- *bandes* : la matrice d'utilisation des bandes. Cette matrice indique le nombre de longueurs d'onde utilisées sur une bande. Une valeur positive et non nulle signifie que la bande n'est pas disponible (lors de la recherche d'une bande). Une

valeur nulle indique que la bande est disponible. Une valeur de  $W$  indique que la bande est pleine;

- $K$  : le nombre de longueurs d'onde sur une fibre;
- $B$  : le nombre de bandes dans une fibre;
- $W$  : le nombre de longueurs d'onde dans une bande.

La fonction importante de la classe est la suivante :

- $CreerMatricesRessources()$  : cette fonction crée les matrices qui indiquent l'utilisation des longueurs d'onde et des bandes sur les fibres du lien.

#### 4.1.4. La classe Route

Cette classe implémente une route (un chemin dans  $G$ ) dans un réseau. Ses principaux attributs sont les suivants :

- $source$  : l'identifiant du nœud source de la route;
- $destination$  : l'identifiant du nœud destination;
- $nodes$  : l'ensemble des nœuds de la route. Il s'agit d'un vecteur contenant des pointeurs vers les nœuds traversés par la route;
- $path$  : la séquence des liens du nœud source au nœud destination. Il s'agit d'un vecteur contenant des pointeurs vers les liens de la route;
- $nbrLiens$  : le nombre de liens de la route;
- $deviation$  : l'identifiant du nœud de déviation de la route;
- $cost$  : le coût de la route.

Les fonctions importantes de la classe sont les suivantes :

- $getSubPath(i, j)$  : cette fonction retourne le sous-chemin de la route entre les nœuds  $i$  et  $j$ . Les deux nœuds doivent appartenir à la route;
- $EstSousCheminDe(autreRoute)$  : cette fonction vérifie si la route est le sous-chemin d'une autre route;

- *RechercheBande()* : cette fonction recherche une bande disponible le long de la route;
- *RechercheLambda()* : cette fonction recherche une longueur d'onde disponible le long de la route;
- *ReservationBande()* : cette fonction réserve une bande le long de la route. La bande est mise pleine sur chacun des liens de la route;
- *ReservationLambda()* : cette fonction réserve une longueur d'onde le long de la route. La longueur d'onde est mise indisponible sur chacun des liens de la route;
- *LiberationBande()* : cette fonction libère une bande le long de la route. La bande est rendue disponible sur chacun des liens de la route;
- *LiberationLambda()* : cette fonction libère une longueur d'onde le long de la route. La longueur d'onde est mise disponible sur chacun des liens de la route;
- *ConfigurationCommutateurNoeudsBande()* : cette fonction ajoute ou supprime une bande au niveau de chacun des nœuds de la route;
- *ConfigurationCommutateurNoeudsLambda()* : cette fonction ajoute ou supprime une longueur d'onde au niveau de chacun des nœuds de la route;
- *getNbrPorts()* : cette fonction retourne le nombre de ports des nœuds de la route.

#### 4.1.5. La classe Paire

Cette classe représente une paire de nœuds dans le réseau. Ses principaux attributs sont les suivants :

- *source* : l'identifiant du nœud source de la paire;
- *destination* : l'identifiant du nœud de destination de la paire;
- *demande* : la demande de la paire en terme de chemins optiques;
- *route* : un pointeur sur la route affectée à la paire;
- *kRoutes* : l'ensemble des  $k$  plus courts chemins de la paire. Il s'agit d'un vecteur trié en ordre croissant suivant la longueur de chemin. Il contient des pointeurs vers les  $k$  routes déterminées;

- *indexRoute* : l'indice de la route dans le vecteur *kRoutes* ;
- *lambdaAssigne* : indique si toute la demande de la paire a été satisfaite ou non;
- *poids* : le poids de la paire;
- *lambdas* : l'ensemble des longueurs d'onde affectées à la paire. Il s'agit d'un tableau d'entiers dont les éléments indiquent la longueur d'onde affectée;
- *nbrLambdas* : le nombre de longueurs d'onde affectées à la paire;
- *bandes* : l'ensemble des bandes d'onde affectées à la paire. Il s'agit d'un tableau d'entiers dont les éléments indiquent la bande affectée;
- *nbrBandes* : le nombre de bandes affectées à la paire;
- *fibresLambdas* : la séquence des fibres utilisées le long de la route affectée à la paire pour chacune des longueurs d'onde affectées. Il s'agit d'une matrice. La ligne correspond à la position courante dans la structure *lambdas*. La colonne indique le lien (premier lien, deuxième lien, ...) de la route. Les éléments de la matrice indiquent le numéro de la fibre utilisée;
- *fibresBandes* : la séquence des fibres utilisées le long de la route affectée à la paire pour chacune des bandes d'onde affectées. Il s'agit d'une matrice. La ligne correspond à la position courante dans la structure *bandes*. La colonne indique le lien (premier lien, deuxième lien,...) de la route. Les éléments de la matrice indiquent le numéro de la fibre utilisée.

Les fonctions importantes de la classe sont les suivantes :

- *AssiguationBande()* : cette fonction affecte une bande d'onde à la paire en appelant la fonction correspondante sur la route affectée;
- *AssiguationLambda()* : cette fonction affecte une longueur d'onde à la paire en appelant la fonction correspondante sur la route affectée.
- *ConfigurationCommutateurNoeuds()* : cette fonction configure les nœuds traversés par la route affectée à la paire;

- *LiberationRessources()* : cette fonction libère les ressources utilisées par la paire (longueurs d'onde et bandes);
- *Reservation()* : cette fonction réserve le long de la route affectée des longueurs d'onde et des bandes d'onde.

#### 4.1.6. La classe Session

Cette classe représente une session du plan d'expérience. Elle stocke les entrées pour une session. Ses principaux attributs sont les suivants :

- *reseau* : un pointeur sur le réseau donné;
- *trafic* : la matrice de trafic;
- *L* : le nombre de longueurs d'onde par lien;
- *K* : le nombre de longueurs d'onde dans une fibre;
- *W* : le nombre de longueurs d'onde dans une bande;
- *LT* : la longueur de la liste taboue;
- *k* : le nombre de *k* plus courts chemins à calculer par paire de nœuds;
- *MaxIter* : le nombre maximum d'itérations;
- *Paires* : l'ensemble des paires de demande non nulle du réseau. Il s'agit d'un vecteur contenant des pointeurs vers les paires dont la demande est différente de zéro.

#### 4.1.7. La classe Algorithmes

Cette classe contient les algorithmes implémentés. Elle ne contient que des fonctions statiques. Il s'agit des fonctions :

- *PCC()* : algorithme PCC pour le calcul du plus court chemin d'une source à tous les autres nœuds du réseau;
- *Dijkstra(source)* : algorithme de Dijkstra du plus court chemin pour une source vers les autres nœuds du réseau. La fonction retourne l'ensemble des routes trouvées;

- $Dijkstra(source, destination)$  : algorithme de Dijkstra du plus court chemin entre deux nœuds. La fonction retourne la route trouvée;
- $Yen()$  : algorithme de Yen pour le calcul des  $k$  plus courts chemins entre deux nœuds;
- $SolutionInit()$  : recherche d'une solution initiale pour l'algorithme  $TabouMRWA$ ;
- $WA\_MF()$  : algorithme d'affectation de longueurs d'onde à une paire;
- $TabouMRWA v1()$  : algorithme RWA multigranulaire utilisant la recherche taboue (version 1).

## 4.2. Données utilisées

Les données que nous avons utilisées pour réaliser nos expériences sont celles qui définissent une instance du problème et sont les suivantes :

- la topologie du réseau;
- l'ensemble  $P$  des paires de nœuds de demande non nulle. Cet ensemble définit le trafic dans le réseau et est donné sous forme de matrice. Les éléments de cette matrice représentent la demande entre une paire de nœuds. Cette matrice sera notée  $Tr$ ;
- le nombre de fibres par lien,  $F$ ;
- le nombre de longueurs d'onde par fibre,  $K$ ;
- le nombre de longueurs par bande d'onde,  $W$ .

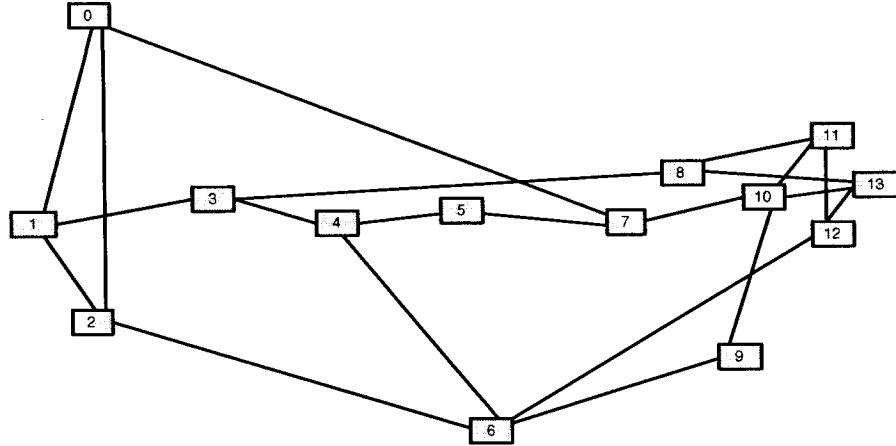
### 4.2.1. Fichier de topologie

Nous avons utilisé le réseau NSFNET (comme Cao et al. [5]) afin de réaliser nos simulations. Le Tableau 4.1 résume la taille du réseau.

**Tableau 4.1 Taille du réseau utilisé**

Réseau	Nombre de noeuds	Nombre de liens
NSFNET	14	21

Le réseau NSFNET est présenté à la Figure 4.1.



**Figure 4.1 Topologie du réseau NSFNET**

#### 4.2.2. Modèle de trafic

Le trafic dans les réseaux que nous avons utilisés dans nos simulations a été généré aléatoirement. La demande des paires de nœuds a été générée suivant une loi uniforme sur l'intervalle  $[0, demandeMax]$ . La graine que nous avons utilisée est basée sur le temps, en particulier l'heure de l'ordinateur utilisé pour réaliser les expériences. La charge totale dans le réseau dépend de la valeur attribuée à la variable *demandeMax*. Nous avons choisi d'utiliser un trafic aléatoire (employant une loi uniforme) de façon à généraliser nos résultats et à ne pas nous enfermer dans des cas particuliers, étant donné que le trafic joue un rôle important dans les résultats. Ceci est d'autant plus important que nous n'utilisons que le réseau NSFNET dans notre étude de performance.

#### 4.3. Plan d'expérience

Dans cette section, nous allons présenter la stratégie adoptée pour réaliser l'évaluation de performance de l'algorithme *TabouMRWA*. Nous commencerons par indiquer les indices de performance considérés pour notre évaluation. Nous identifierons par la suite les facteurs pertinents que nous ferons varier pour mesurer leur influence sur les indices de performance. Une fois ces facteurs déterminés, nous choisirons leurs

différents niveaux. Nous terminerons la section en présentant les types d'expériences que nous avons choisis.

#### 4.3.1. Les indices de performance

Le but de la multigranularité dans les réseaux optiques est de réduire le nombre de ports comparé à celui dans les réseaux à routage de longueurs d'onde. Donc, l'indice de performance de notre algorithme devrait permettre de comparer le nombre de ports trouvé à celui obtenu par un autre algorithme cherchant à réduire le nombre de ports dans les réseaux à routage de longueurs d'onde (WRNs). (Rappelons que diminuer le nombre de ports dans de tels réseaux est équivalent à réduire le nombre de sauts de longueur d'onde dans le réseau). En fait, le mieux serait de le comparer au meilleur algorithme connu dans la littérature. Cependant, nous cherchons essentiellement à comparer notre algorithme au BPHT présenté par Cao et al. [5]. Il a déjà été prouvé que ce dernier réduit le nombre de ports comparé à celui trouvé par un algorithme (dit « optimal ») réduisant le nombre de ports dans les WRNs. Il nous suffit donc de montrer que le nombre de ports que nous trouvons est inférieur à celui obtenu en appliquant le BPHT avec les mêmes données en entrée. Par conséquent, nous considérons le *nombre de ports* globalement dans le réseau comme un indice de performance dans notre évaluation et nous ferons la comparaison ensuite avec le *BPHT*.

Le temps d'exécution est un indice de performance moins important dans le contexte de notre évaluation. Le *BPHT* est un algorithme de construction. Notre algorithme étant une méthode itérative, nous ne pouvons pas réellement comparer les temps d'exécution. Cependant, nous analyserons le temps d'exécution de notre algorithme pour nous assurer que les temps de calcul restent raisonnables.

#### 4.3.2. Les facteurs de simulation

Les données du problème constituent des paramètres susceptibles d'influencer les résultats. Ces paramètres sont les facteurs les plus importants pour notre évaluation. Ils ont une influence directe sur le calcul du nombre de ports. De plus, ils nous servent

de base de comparaison avec l'algorithme *BPHT*. Cependant, parmi ces paramètres, la taille du réseau n'a pas été prise en compte vu qu'elle est fixe pour un réseau donné et nous n'utilisons qu'un réseau. Les paramètres  $F$ ,  $K$ ,  $W$  et  $Tr$  sont donc les seuls à être considérés parmi les données du problème. Ils constituent les facteurs primaires de notre évaluation.

En plus des données du problème, les paramètres suivants influencent les résultats :

- le nombre  $k$  des plus courts chemins entre une paire de nœuds ;
- la longueur de la liste taboue, LT ;
- le nombre maximum d'itérations, M.

Ces facteurs sont les paramètres de l'algorithme *TabouMRWA*. Leur influence est non négligeable mais moins importante pour notre évaluation par rapport aux données du problème. Ils n'influencent pas directement le calcul du nombre de ports. Ces facteurs sont dits secondaires.

#### 4.3.3. Choix des niveaux des facteurs

Afin de cerner les effets des paramètres sur la performance de notre algorithme, nous devons définir des niveaux pour chacun d'eux. Le niveau des facteurs correspond à des valeurs qu'ils peuvent prendre parmi un ensemble possible relatif à chaque facteur. Le Tableau 4.2 résume les niveaux choisis.

**Tableau 4.2 Niveaux des facteurs**

Facteurs		Niveaux
Paramètres	Algorithme	
Données du problème	$k$	5, 10, 15, 20, 25, 30, 35
	LT	0, 5, 10, 15, 20, 25, 30, 35, 40
	M	100, 1000, 10000
	$F$	1, 2, 3, 4, 5, 6, 10, 12, 15
	$K$	60, 120, 240, 360, 480
	$W$	1, 4, 6, 8, 10, 12, 15, 20, 30
	$Tr$	Rand5, Rand10, Rand20, Rand25, Uni (d)

Les niveaux choisis pour les facteurs primaires,  $F, K, W$ , représentent les valeurs utilisées par Cao et al. [5] dans l'évaluation de leur algorithme *BPHT*. Ce choix est guidé par le fait que nous voulons principalement comparer notre algorithme au *BPHT*. La sélection des niveaux des facteurs secondaires,  $k, LT, M$ , a été faite de manière à couvrir adéquatement l'étendue de variation des facteurs. Pour le facteur  $k$ , les valeurs 5, 10, 15, 20, 25, 30, 35 ont été choisies. En ce qui concerne le paramètre  $LT$ , nous avons décidé de commencer par 0 et d'incrémenter de 5 à chaque fois jusqu'à la valeur 40. Une valeur de 0 signifie que la liste sera toujours vide. Pour le facteur  $M$ , une valeur initiale de 100 a été choisie et les valeurs suivantes ont été obtenues en multipliant à chaque fois par 10. Nous nous sommes arrêtés à la valeur 10000. Ces valeurs ont été jugées suffisantes pour observer l'évolution du nombre de ports en fonction de ce paramètre. Nous avons utilisé différents niveaux de trafic aléatoire en faisant varier la valeur attribuée à la variable *demandeMax*. Le niveau de trafic est donné par le nombre ajouté au mot *Rand*. Par exemple, *Rand5* correspond à une matrice de trafic dont la valeur 5 a été affectée à la variable *demandeMax*. Le niveau *Uni* (d) est une matrice dont le trafic est uniforme. La demande par paire de nœuds est égale à  $d$ , sauf pour les paires dont les nœuds source et destination sont identiques. Dans ce cas, la demande est égale à 0.

#### 4.3.4. Types d'expériences choisies

Une fois les niveaux de chaque facteur spécifiés, les combinaisons de ces niveaux doivent être choisies afin de réaliser les expériences. Une combinaison particulière de niveaux des facteurs, c'est-à-dire du tuple  $(k, LT, M, F, K, W, Tr)$ , représente une session. Une combinaison possible serait par exemple (10, 20, 1000, 2, 120, 4, *Rand20*), soit 10 plus courts chemins par paire de nœuds, une liste taboue de longueur 20, un nombre maximum d'itérations de 1000, 2 fibres par lien, 120 longueurs d'onde au maximum par fibre, 4 longueurs d'onde au maximum par bande, un trafic dont chaque demande entre paire de nœuds est générée aléatoirement sur l'intervalle  $[0, 20]$ . Nos expériences sont à session multiple afin de déterminer les influences des facteurs sur la performance de notre algorithme.

Notre plan d'expérience comporte trois étapes, chacune regroupant un certain nombre d'expériences. Dans la première étape, les expériences seront de type «un facteur à la fois», afin d'étudier l'effet de chaque facteur. Dans ce type d'expérience, on fait varier un facteur à la fois tandis que les autres restent constants. De cette manière, on peut isoler l'effet d'un facteur en fonction des résultats obtenus. Cependant, dans cette étape, nous étudierons surtout l'effet de la variation des facteurs secondaires  $k$ , LT et M. Les effets de la variation des facteurs primaires,  $F$ ,  $K$  et  $W$ , seront étudiés à une autre étape. Dans la deuxième étape, nous choisirons des combinaisons de niveaux entre les facteurs secondaires afin de calibrer ceux-ci. Les valeurs choisies seront ensuite utilisées à la troisième étape. Dans celle-ci, nous reprendrons les expériences réalisées par Cao et al. [5] afin de comparer nos résultats aux leurs.

#### 4.4. Expérimentation et tests

Dans cette section, nous allons présenter et analyser les résultats que nous avons obtenus. Dans chacune des expériences que nous avons réalisées, nous avons fait varier le trafic, sans en changer le niveau, pour chaque valeur ou combinaison de valeurs des facteurs que nous étudions. Chaque résultat présenté correspond à la moyenne arithmétique des résultats obtenus en faisant varier le trafic (sans en changer le niveau) et en gardant constant tous les autres facteurs. Pour chaque expérience, nous avons utilisé 10 matrices de trafic différentes générées de manière aléatoire et dont la valeur de la variable *demandeMax* est la même. Le Tableau 4.3 résume les expériences que nous avons réalisées. La deuxième colonne indique les paramètres qui varient dans l'expérience. La troisième colonne indique certaines particularités éventuelles de l'expérience.

**Tableau 4.3 Expériences réalisées**

<b>Étape 1</b>		
<b>Expérience 1.1</b>	Effet de la variation de $k$	
<b>Expérience 1.2</b>	Effet de la variation de LT	
<b>Expérience 1.3</b>	Effet de la variation de M	
<b>Étape 2</b>		
<b>Expérience 2.1</b>	Effet de la variation de $k$ , LT	
<b>Étape 3</b>		
<b>Expérience 3.1</b>	Effet de la variation du trafic uniforme	
<b>Expérience 3.2</b>	Effet de la variation de $W$	
<b>Expérience 3.3</b>	Effet de la variation de $F$ , $W$	$F \times K = \text{constante}$
<b>Expérience 3.4</b>	Effet de la variation de $F$	$F \times K = \text{constante}$
<b>Expérience 3.5</b>	Effet de la variation de $F$ , $W$ , $Tr$	Variation $F$ = Variation $Tr$ initial

#### 4.4.1. Étape 1 : expériences « un facteur à la fois »

Dans cette section, nous faisons varier les facteurs secondaires  $k$ , LT, M mais un facteur à la fois. Les valeurs par défaut que nous utilisons pour les facteurs secondaires sont données au Tableau 4.4. Lorsqu'un facteur ne varie pas, la valeur correspondante dans le tableau lui sera affectée. (Il est à noter que la liste taboue est construite en mettant les paires taboues et non les routes des paires). Ce tableau indique aussi les valeurs utilisées pour les facteurs primaires.

**Tableau 4.4 Valeurs par défaut des facteurs (Étape 1)**

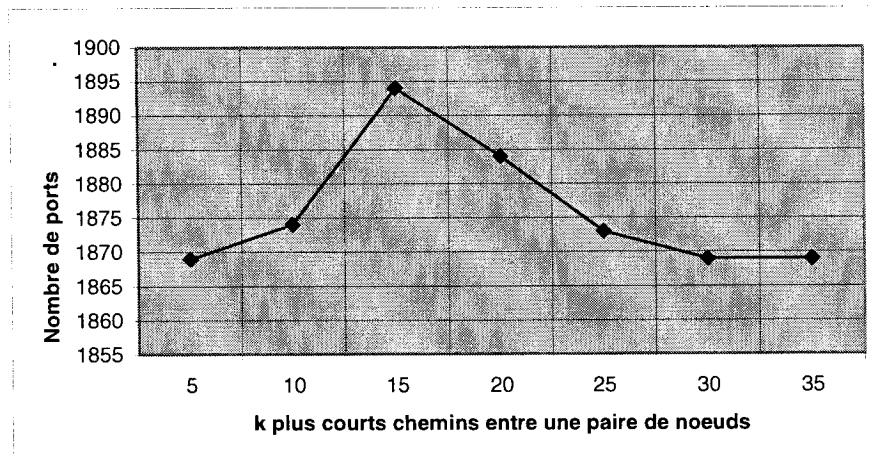
$k$	LT	M	$F$	$K$	$W$	$Tr$
5	20	1000	2	120	4	Rand25

#### Expérience 1.1 : Effet de la variation du nombre $k$ de plus courts chemins

Dans cette expérience, nous avons fait varier le nombre  $k$  de plus courts chemins entre une paire de nœuds. La Figure 4.2 montre le nombre de ports trouvé pour chacune

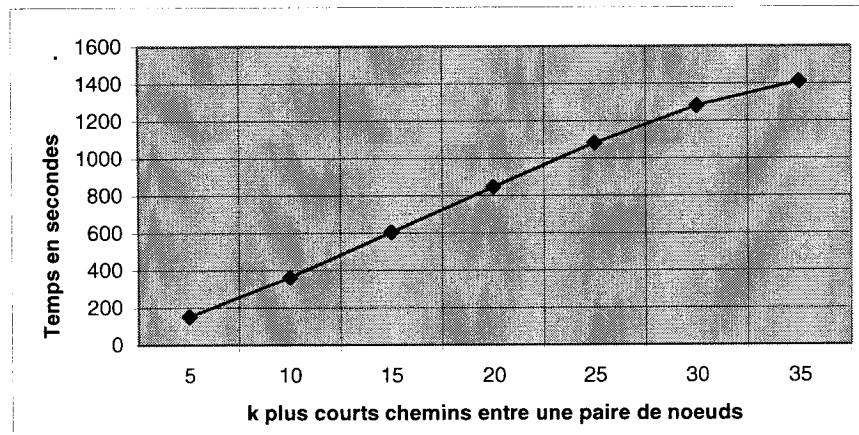
des valeurs de  $k$ . Nous remarquons que le fait d'augmenter la valeur de  $k$  n'améliore pas nécessairement les résultats. Le nombre de ports augmente entre les valeurs 5 et 15 et diminue pour les valeurs suivantes. Nous serions portés à penser qu'une augmentation de la valeur de  $k$  conduirait nécessairement à une diminution du nombre de ports. Cependant, étant donné que le coût d'une solution dépend de l'ensemble des chemins optiques établis dans le réseau, la réduction du nombre de ports ne dépend pas uniquement de la sélection des routes. Elle dépend aussi de l'affectation des longueurs d'onde. (Des solutions différentes peuvent utiliser un même plan de routage, mais l'ensemble de longueurs d'onde dans le réseau sera nécessairement différent conduisant éventuellement à un coût différent). De plus, il faut se rappeler que les routes que nous changeons sont des routes dans le graphe  $G$ . Les routes dans le multigraphe  $G_F$  associées aux chemins optiques sont déterminées après l'affectation de longueurs d'onde (qui détermine la fibre empruntée par un chemin optique sur un lien).

Notre algorithme priorise la sélection de routes. Il espère qu'en changeant les routes dans le graphe  $G$  et en réaffectant des longueurs d'onde à chaque changement de route, nous arriverons à une combinaison de routes (dans le multigraphe  $G_F$ ) et de longueurs d'onde qui diminue le nombre de ports de manière appréciable. Cependant, l'affectation de longueurs d'onde est plus importante que la sélection de routes pour de petites valeurs de  $k$ . Lorsque ce dernier varie sur un intervalle où les valeurs ne sont pas élevées, plus il augmente, plus le nombre de combinaisons de longueurs d'onde essayées (pour un plan de routage donné) diminue. En d'autres termes, moins la valeur de  $k$  est élevée, plus on a de chance d'inspecter des solutions dont le plan de routage est le même. Ceci reviendrait à essayer des combinaisons différentes de longueurs d'onde pour un même plan de routage. Donc la probabilité de rater certaines solutions qui nous auraient éventuellement conduit vers de bonnes régions de l'espace de recherche augmente avec  $k$  pour de petites valeurs de celui-ci. Lorsque nous augmentons progressivement la valeur de  $k$ , à partir d'un certain point, la sélection de routes devient plus importante que l'affectation de longueurs d'onde et le nombre de ports diminue lorsque  $k$  augmente.



**Figure 4.2 Nombre de ports en fonction des  $k$  plus courts chemins**  
**NSFNET, LT=20, M=1000, F=2, K=120, W=4, Tr = Rand25**

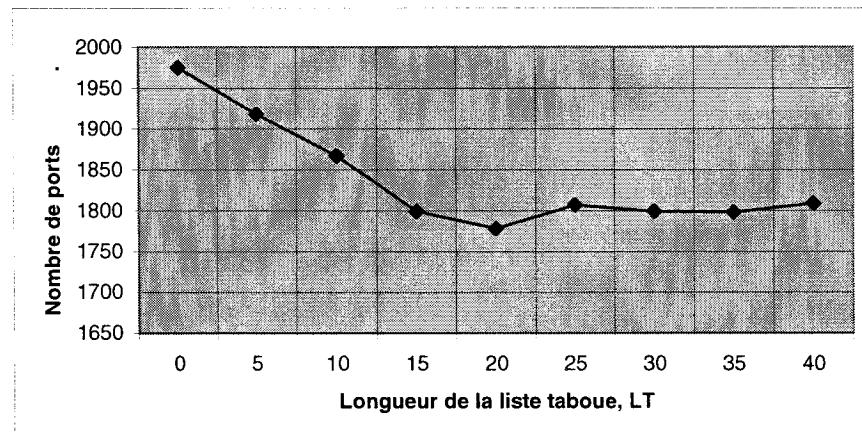
La Figure 4.3 montre le temps d'exécution en fonction de la valeur de  $k$ . Le temps d'exécution augmente de manière notable lorsque  $k$  augmente. Cela s'explique du fait que l'algorithme inspecte tous les  $k$  plus courts chemins de chacune des paires. Donc, plus  $k$  est élevé, plus le nombre de routes à inspecter augmente.



**Figure 4.3 Temps d'exécution en fonction des  $k$  plus courts chemins**  
**NSFNET, LT=20, M=1000, F=2, K=120, W=4, Tr = Rand25**

### Expérience 1.2 : Effet de la variation de la longueur de la liste taboue, LT

Dans cette expérience, nous faisons varier la longueur de la liste taboue. La Figure 4.4 présente les résultats obtenus en faisant varier la longueur de la liste taboue. Le nombre de ports est affecté tout comme dans le cas précédent. En analysant la Figure 4.4, on remarque que le nombre de ports ne diminue pas toujours lorsque la longueur de la liste augmente. En fait, il a même tendance à augmenter à partir d'une certaine valeur de la longueur de la liste. À partir d'une longueur de 40, le nombre de ports augmente. Il est minimal pour une longueur de 20.

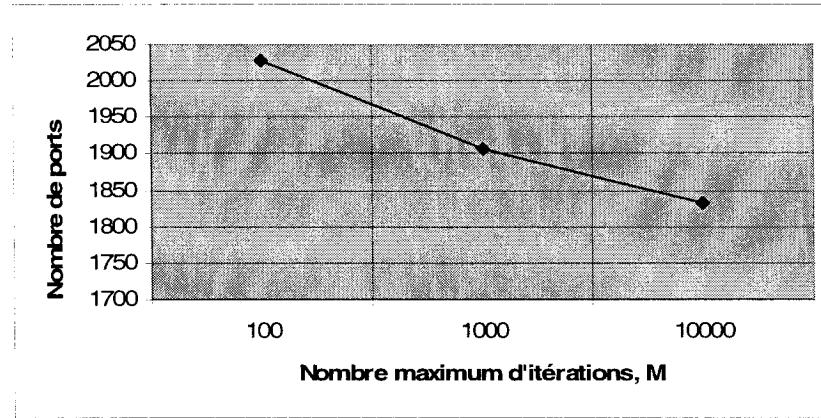


**Figure 4.4 Nombre de ports en fonction de la longueur de la liste taboue**

**NSFNET,  $k=5$ ,  $M=1000$ ,  $F=2$ ,  $K=120$ ,  $W= 4$ ,  $Tr = \text{Rand25}$**

### Expérience 1.3 : Effet de la variation du nombre maximum d'itérations, M

La Figure 4.5 montre la variation du nombre de ports en fonction du nombre maximum d'itérations. Le nombre de ports diminue lorsque M augmente, mais cette diminution est moins marquée entre les valeurs 1000 et 10000.



**Figure 4.5 Nombre de ports en fonction du nombre maximum d'itérations**

**NSFNET,  $k=5$ ,  $LT=20$ ,  $F=2$ ,  $K=120$ ,  $W=4$ ,  $Tr = \text{Rand25}$**

#### 4.4.2. Étape 2 : calibration des facteurs secondaires

Le but de cette étape est de choisir une valeur pour chacun des facteurs secondaires ( $k$ , LT et M). Les valeurs choisies seront utilisées par la suite dans les expériences de la troisième étape. Cependant, ces valeurs ne sont pas nécessairement optimales. Il s'agit plutôt ici de trouver de bonnes valeurs, c'est-à-dire une combinaison de valeurs des facteurs où le nombre de ports diminue le plus tout en gardant un temps d'exécution raisonnable (pas trop élevé). Cela nous permettra alors de réaliser plus d'expériences à l'étape 3. Dans cet ordre d'idées, nous avons décidé de fixer le nombre maximum d'itérations à 1000 ( $M=1000$ ). Nous ferons varier le nombre  $k$  de plus courts chemins pour différentes valeurs de la longueur de la liste taboue, LT, afin de fixer une valeur pour chacun de ces facteurs.

#### Expérience 2.1 : Effet de la variation du nombre $k$ de plus courts chemins et de la longueur de la liste taboue, LT

Dans cette expérience, nous faisons varier le nombre  $k$  de plus courts chemins et la longueur de la liste taboue, LT, afin de leur affecter une valeur. Le Tableau 4.5 indique la valeur des facteurs constants.

**Tableau 4.5 Valeur des facteurs constants (Expérience 2.1)**

LT	<i>F</i>	<i>K</i>	<i>W</i>	<i>Tr</i>
20	2	120	4	Rand20

Le Tableau 4.6 indique le nombre de ports obtenu pour différentes valeurs du nombre  $k$  et de la longueur de la liste taboue alors que le Tableau 4.7 indique le temps d'exécution. L'analyse des résultats du Tableau 4.6 montre que le plus petit nombre de ports est obtenu pour le couple (20, 20) où le premier nombre représente la valeur du facteur  $k$  et le deuxième la longueur de la liste taboue, LT. Cependant, nous ne pouvons conclure que la meilleure combinaison des facteurs  $k$  et LT est donnée par ce couple. Premièrement, les résultats dépendent également des facteurs primaires, en particulier du trafic. Le trafic utilisé pour cette expérience ne sera pas nécessairement le même pour la suite des tests. (Le niveau du trafic utilisé pour ce test est moyen). Ensuite, d'autres couples fournissent un nombre de ports très proche de la valeur minimale trouvée. Enfin, le temps d'exécution doit être pris en compte. Il est donc préférable de choisir un couple qui offre un bon compromis entre le nombre de ports et le temps d'exécution. En analysant conjointement les résultats des Tableaux 4.6 et 4.7, nous remarquons que le couple (5, 20) offre le meilleur compromis. D'autres couples fournissent un nombre de ports plus faible. Cependant, leur temps d'exécution est nettement plus élevé alors que les améliorations apportées au niveau du nombre de ports sont moins de 0.5% en moyenne. Nous avons ainsi décidé de garder les valeurs  $k = 5$  et  $LT = 20$  pour la suite des tests. De plus, de manière générale,  $k = 5$  fournit de bons résultats, proches de ceux obtenus pour  $k = 20$ . Et aussi, la valeur  $LT = 20$  fournit le meilleur résultat pour  $k = 5$ .

**Tableau 4.6 Nombre de ports pour  $k$  et LT variables**(NSFNET,  $F=2$ ,  $K=120$ ,  $W=4$ ,  $Tr = \text{Rand20}$ )

$k$	Longueur de la liste taboue, LT					
	10	15	20	25	30	35
5	1553	1493	1478	1485	1496	1500
10	1544	1512	1495	1480	1475	1477
15	1530	1484	1486	1476	1473	1483
20	1530	1498	1470	1478	1476	1486
25	1523	1490	1484	1486	1477	1473
30	1524	1493	1481	1478	1476	1475
35	1524	1493	1481	1478	1476	1471

**Tableau 4.7 Temps d'exécution (en secondes) pour  $k$  et LT variables**(NSFNET,  $F=2$ ,  $K=120$ ,  $W=4$ ,  $Tr = \text{Rand20}$ )

$k$	Longueur de la liste taboue, LT					
	10	15	20	25	30	35
5	125.796	123.828	125.498	124.883	125.339	125.488
10	307.758	307.369	306.94	309.284	312.151	309.973
15	527.877	530.966	526.51	529.735	534.881	528.126
20	781.311	775.372	773.561	779.823	773.745	770.151
25	1068.56	1054.24	1039.02	1056.98	1044.39	1044.24
30	1318.99	1314.93	1299.31	1325.01	1314.56	1306.37
35	1497.22	1491.75	1472.89	1500.08	1487.68	1465.77

#### 4.4.3. Étape 3 : comparaison entre TabouMRWA et BPHT

Afin d'évaluer la performance de notre algorithme, le *TabouMRWA*, nous avons comparé nos résultats à ceux obtenus par le *BPHT* en reprenant les expériences réalisées par Cao et al. [5] pour évaluer la performance de ce dernier. Cependant, nous n'avons pas pu utiliser les résultats présentés par Cao et al. [5]. Dans leur travail, les auteurs

n'ont pas fourni les matrices de trafic utilisées dans les expériences. De plus, les résultats présentés n'indiquent pas directement le nombre de ports trouvé, mais plutôt un ratio entre le nombre de ports obtenu en appliquant le *BPHT* et celui obtenu en appliquant un algorithme *RWA* dont les détails n'ont pas été fournis. Nous avons alors utilisé les résultats de notre propre implémentation de l'algorithme *BPHT* pour les comparaisons.

Nous avons opté pour une implémentation directe du *BPHT*. Cependant, il revient de souligner notre interprétation de certains points de l'algorithme. Tout d'abord, les indices de fibre, de bande et de longueur d'ondes ont été initialisés une fois au début de l'algorithme. Ensuite, lors de la recherche d'une fibre possédant le plus de bandes disponibles ou possédant au moins une longueur d'onde disponible, le même numéro de fibre n'est pas nécessairement utilisé tout le long de la route. Cependant, sur chacun des liens de celle-ci, une fibre est recherchée à partir du même numéro de la fibre trouvée sur le premier lien. Il est à noter que nous avons implémenté le cas où le même numéro de fibre est utilisé sur tous les liens de la route, mais les résultats étaient moins bons en général et ce cas n'a pas été pris en compte dans nos simulations.

Afin de mieux comparer notre algorithme au *BPHT*, nous utilisons la variable

$$Gain \text{ tel que } Gain = \left( \left( 1 - \frac{nbP(TabouMRWA)}{nbP(BPHT)} \right) \right) \times 100\%, \text{ où } nbP(TabouMRWA)$$

représente le nombre de ports trouvé en appliquant le *TabouMRWA* et  $nbP(BPHT)$  le nombre de ports obtenu en appliquant le *BPHT*. La variable *Gain* représente le gain en terme de réduction du nombre de ports. Un gain positif signifie alors que *TabouMRWA* fait mieux que le *BPHT*. Plus le gain est élevé, plus notre algorithme fait mieux que le *BPHT*. Un gain négatif signifie que le *BPHT* fait mieux que notre algorithme. Plus le gain est négatif, moins notre algorithme est performant comparé au *BPHT*.

Les facteurs secondaires sont maintenus fixes pour toutes les expériences de cette étape. Le Tableau 4.8 indique les valeurs des facteurs secondaires pour l'algorithme *TabouMRWA*. Les facteurs *LT* et *M* sont utilisés uniquement par ce dernier. Entre autres, l'algorithme *BPHT*, en plus des facteurs primaires, utilisent le facteur *k* pour les

plus courts chemins entre les paires de nœuds. La valeur de  $k$  est fixée à 3 pour le *BPHT*. Selon les auteurs de l'algorithme, ils ont obtenu de meilleurs résultats avec cette valeur, assertion que nous avons vérifiée.

**Tableau 4.8 Valeur des facteurs secondaires pour TabouMRWA (Étape 3)**

$k$	LT	M
10	20	1000

### Expérience 3.1 : Effet de la variation du trafic uniforme

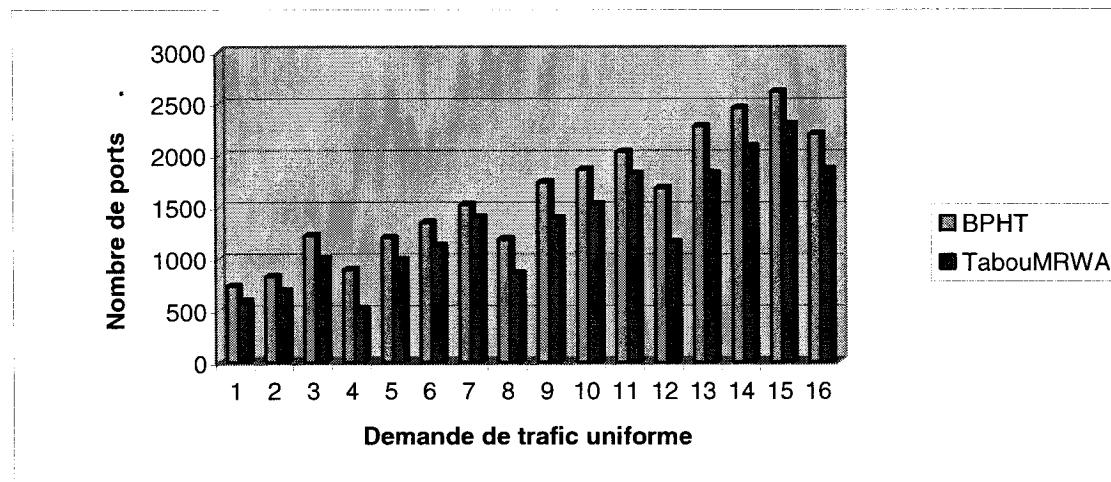
Dans cette expérience, nous utilisons un trafic uniforme dans le réseau et nous faisons varier la demande par paire de nœuds de 1 à 16 par incrément d'une unité. Le Tableau 4.9 fournit les valeurs utilisées pour les facteurs primaires constants.

**Tableau 4.9 Valeur des facteurs primaires constants (Expérience 3.1)**

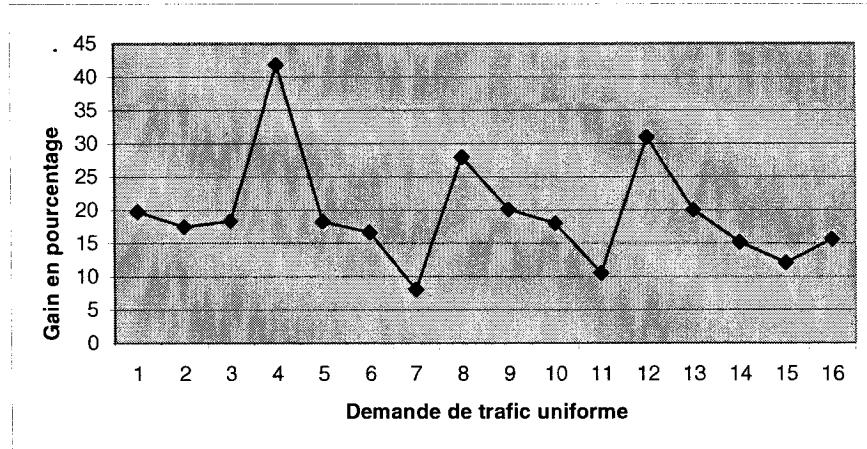
$F$	$K$	$W$
2	120	4

La Figure 4.6 montre les résultats obtenus en appliquant notre algorithme, le *TabouMRWA*, et ceux obtenus en appliquant l'algorithme *BPHT*. Le nombre de ports est calculé pour des valeurs de la demande de trafic uniforme variant de 1 à 16. En observant la figure, nous constatons qu'en joignant les extrémités des parallélépipèdes rectangles pour chacun des algorithmes, nous obtenons une courbe montante en dents de scie. Les minimums correspondent à des demandes de trafic dont la valeur est un multiple du nombre de longueurs d'onde par bande. Pour ces demandes, il est plus aisé de grouper des chemins optiques dans une même bande et de commuter toute la bande au niveau des nœuds traversés par ces derniers. Ceci est particulièrement vrai pour les chemins optiques d'une même paire. Pour les autres demandes, il est plus difficile de commuter toute une bande au niveau des nœuds. Le nombre de ports augmente alors plus rapidement en fonction du trafic.

Le gain de notre algorithme par rapport au BPHT est indiqué à la Figure 4.7. Les maximums de la courbe correspondent à des demandes de trafic qui sont des multiples du nombre de longueurs d'onde par bande. Ceci s'explique du fait qu'il est encore plus facile pour notre algorithme (comparé au BPHT) de router les chemins optiques de façon à commuter le plus de bandes que possible au niveau des nœuds. Les minimums correspondent aux demandes de trafic où il devient plus difficile pour les deux algorithmes de commuter des bandes au niveau des nœuds. Ces demandes précèdent tout juste celles dont la valeur est un multiple du nombre de longueurs d'onde par bande. (Il faut se rappeler que le nombre de ports augmente plus rapidement entre deux demandes multiples du nombre de longueurs d'onde par bande). Notre algorithme fait mieux que le BPHT pour ces demandes également, mais offre une bien moindre performance à cause de la difficulté à commuter des bandes au niveau des nœuds, ce qui explique le gain plus faible.



**Figure 4.6 Nombre de ports en fonction d'une demande uniforme pour TabouMRWA et BPHT**  
**(NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $F = 2$ ,  $K = 120$ ,  $W = 4$ ,  $Tr = \text{Uni}(d)$ )**



**Figure 4.7 Gain de TabouMRWA par rapport au BPHT en fonction de la demande uniforme (NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $F = 2$ ,  $K = 120$ ,  $W = 4$ )**

**Expérience 3.2 : Effet de la variation du nombre de longueurs d'onde par bande,  $W$**

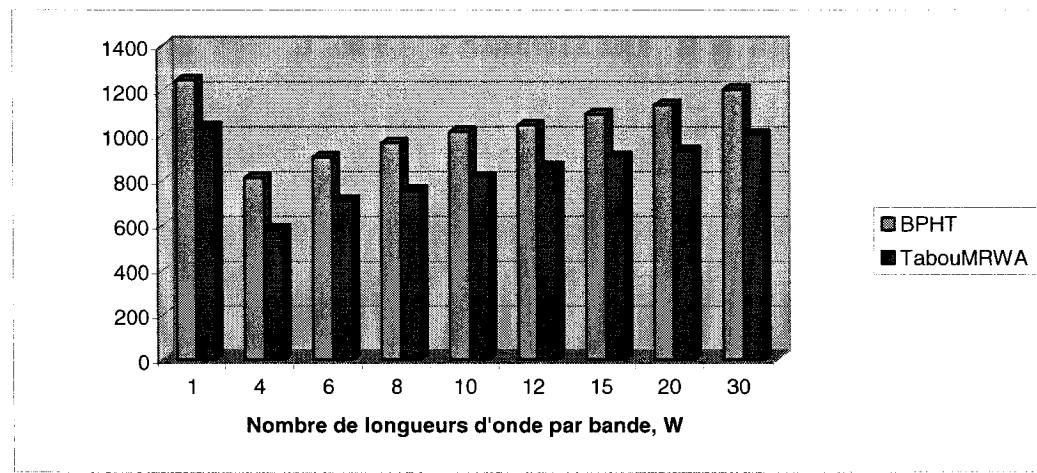
Dans cette expérience, nous faisons varier le nombre de longueurs d'onde par bande,  $W$ . Les différentes valeurs de  $W$  sont les suivantes : 1, 4, 6, 8, 10, 12, 15, 20 et 30. Le Tableau 4.10 indique les valeurs des facteurs primaires constants. Nous utilisons 10 matrices de trafic aléatoire, chacune de *demandeMax* égale à 5. Chaque résultat présenté est la moyenne arithmétique des résultats obtenus pour les différentes matrices.

**Tableau 4.10 Valeur des facteurs primaires constants (Expérience 3.2)**

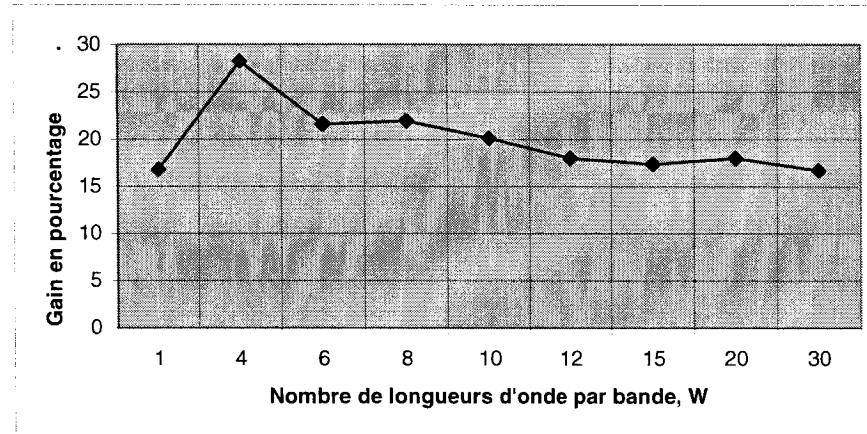
$F$	$K$	$Tr$
1	120	Rand5

La Figure 4.8 indique le nombre de ports pour les différentes valeurs de  $W$  dans le cas du *BPHT* et de *TabouMRWA*. Il est à noter que le nombre de ports est maximal pour  $W = 1$ . Cela s'explique du fait que cette valeur de  $W$  correspond au cas où la multigranularité n'est pas utilisée dans le réseau puisque le nombre de bandes par fibre est égal au nombre de longueurs d'onde par fibre. Cela montre bien que la

multigranularité diminue le nombre de ports dans le réseau (comparé au cas où elle n'est pas employée). Il est à remarquer aussi que le nombre de ports augmente à partir de 4 longueurs d'onde par bande. Ce comportement s'explique du fait que, dans le cas où la multigranularité est utilisée, plus le nombre de longueurs augmente, plus il devient difficile de commuter des bandes dans le réseau (en raison aussi de la contrainte de continuité de longueur d'onde). La Figure 4.9 montre le gain de *TabouMRWA* par rapport au *BPHT* en fonction du nombre de longueurs d'onde par bande. Encore une fois, *TabouMRWA* fait mieux que le *BPHT*. Le gain est plus net pour  $W = 4$ .



**Figure 4.8 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT**  
**(NSFNET,  $k = 5$ , LT = 20, M = 1000, F = 1, K = 120, Tr = Rand5)**



**Figure 4.9 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de longueurs d'onde par bande,  $W$**   
**(NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $F = 1$ ,  $K = 120$ ,  $Tr = \text{Rand5}$ )**

**Expérience 3.3 : Effet de la variation du nombre de fibres par lien,  $F$ , et du nombre de longueurs d'onde par bande,  $W$  avec  $F \times K = \text{constante}$**

Dans cette expérience, nous faisons varier le nombre de fibres par lien et le nombre de longueurs d'onde par bande en maintenant constant le nombre de longueurs d'onde par lien, c'est-à-dire  $F \times K = \text{constante}$ . Étant donné que le nombre de longueurs d'onde sur un lien est constant, lorsque le facteur  $F$  varie, le facteur  $K$  varie également et ses valeurs dépendent de celles de  $F$ . Les différentes valeurs de  $F$  sont 1, 2 et 4. Les niveaux du facteur  $W$  sont les suivants : 1, 4, 6, 10, 12, 15, 20, 30. Pour le nombre de longueurs d'onde par lien, nous avons  $F \times K = 240$ . Le niveau des matrices de trafic utilisées est Rand20 et comme dans l'expérience précédente, chaque résultat présenté correspond à la moyenne arithmétique des résultats obtenus pour les différentes matrices. Les Tableaux 4.11, 4.12 et 4.13 indiquent, pour chacun des algorithmes, le nombre de ports obtenu pour les différentes valeurs de  $F$  en faisant varier le nombre de longueurs d'onde par bande,  $W$ . La Figure 4.10 montre le gain de *TabouMRWA* par rapport au *BPHT* pour différentes valeurs du facteur  $F$ . On remarque que le gain est moindre lorsque le nombre de fibres augmente en maintenant constant le nombre de longueurs d'onde par lien.

**Tableau 4.11 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT avec  $F = 1$  et  $K = 240$   
(NSFNET,  $k = 5$ , LT = 20, M = 1000, Tr = Rand20)**

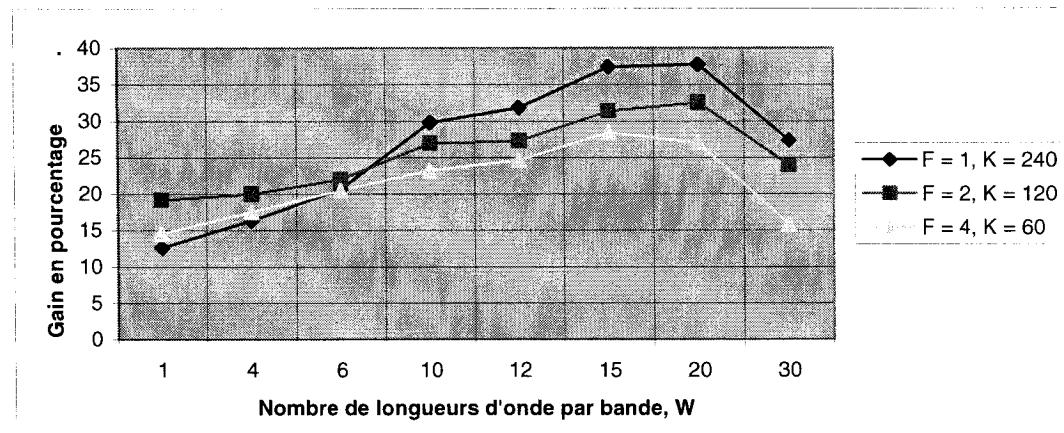
Algorithme	<b>F = 1, K = 240</b>							
	<b>W</b>							
	<b>1</b>	<b>4</b>	<b>6</b>	<b>10</b>	<b>12</b>	<b>15</b>	<b>20</b>	<b>30</b>
<b>BPHT</b>	5812	1916	1690	1853	2040	2387	3553	4162
<b>TabouMRWA</b>	5079	1603	1340	1301	1391	1493	2211	3022

**Tableau 4.12 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT avec  $F = 2$  et  $K = 120$   
(NSFNET,  $k = 5$ , LT = 20, M = 1000, Tr = Rand20)**

Algorithme	<b>F = 2, K = 120</b>							
	<b>W</b>							
	<b>1</b>	<b>4</b>	<b>6</b>	<b>10</b>	<b>12</b>	<b>15</b>	<b>20</b>	<b>30</b>
<b>BPHT</b>	5366	1848	1658	1910	2099	2450	3672	4269
<b>TabouMRWA</b>	4338	1478	1293	1396	1526	1681	2477	3245

**Tableau 4.13 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT avec  $F = 4$  et  $K = 60$   
(NSFNET,  $k = 5$ , LT = 20, M = 1000, Tr = Rand20)**

Algorithme	<b>F = 4, K = 60</b>							
	<b>W</b>							
	<b>1</b>	<b>4</b>	<b>6</b>	<b>10</b>	<b>12</b>	<b>15</b>	<b>20</b>	<b>30</b>
<b>BPHT</b>	4852	1833	1739	2038	2350	2748	4029	4478
<b>TabouMRWA</b>	4147	1512	1382	1565	1771	1967	2940	3771

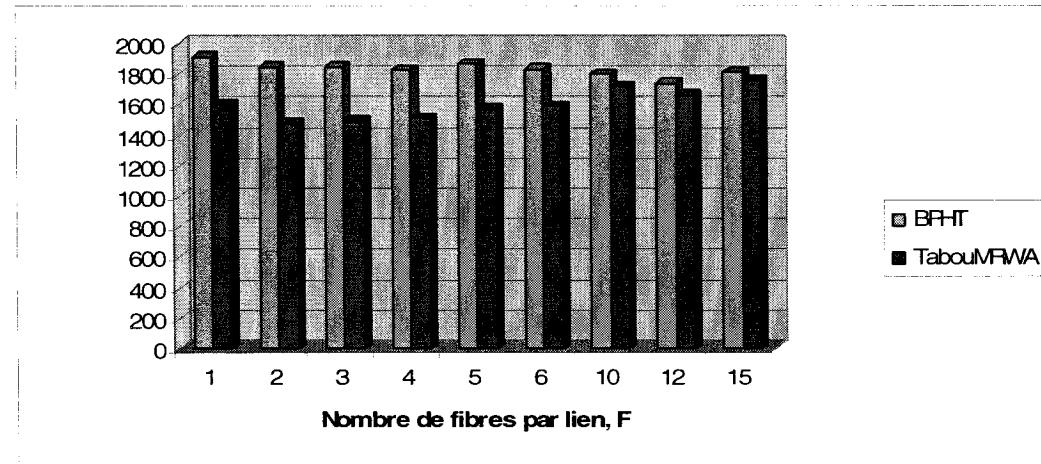


**Figure 4.10 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de longueurs d'onde par bande et pour différentes valeurs de  $F$  (NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $F \times K = 240$ ,  $Tr = \text{Rand20}$ )**

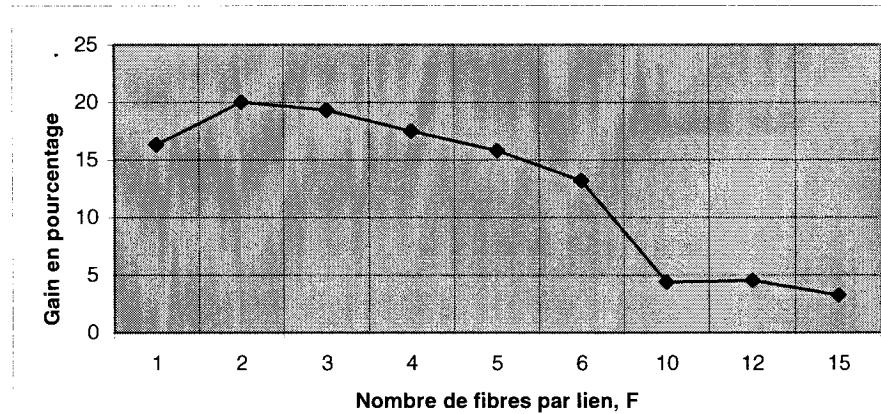
**Expérience 3.4: Effet de la variation du nombre de fibres par lien,  $F$ , et du nombre de longueurs d'onde par fibre,  $K$ , avec  $F \times K = \text{constante}$**

La Figure 4.11 présente, pour les algorithmes *TabouMRWA* et *BPHT*, les résultats obtenus en faisant varier le nombre de fibres par lien et en maintenant constant le nombre de longueurs d'onde par lien, soit  $F \times K = \text{constante}$ . Le facteur  $K$  varie et sa valeur dépend de celle de  $F$ . Par contre, le nombre de longueurs d'onde par bande,  $W$ , est gardé constant. Les différentes valeurs de  $F$  sont les suivantes : 1, 2, 3, 4, 5, 6, 10, 12, 15. Le nombre de longueurs d'onde par lien est égal à 240. La valeur de  $W$  est fixée à 4. Les différentes matrices de trafic utilisées ont un niveau de trafic égal à Rand20. (La moyenne arithmétique des résultats des matrices est présentée). La Figure 4.12 montre le gain de *TabouMRWA* par rapport au *BPHT* pour les différentes valeurs de  $F$ . On remarque qu'à partir d'une certaine valeur du nombre de fibres par lien, le gain diminue lorsque ce dernier augmente. À partir de 10 fibres par lien, le gain devient même très faible. En effet, en augmentant le nombre de fibres par lien, à partir d'une certaine valeur de ce dernier, le nombre de ports a tendance à augmenter si on garde constant le nombre de longueurs d'onde par lien. Il devient plus probable d'utiliser plusieurs fibres pour router des trafics possédant des liens en commun. Il est à rappeler que l'utilisation

d'une fibre implique l'utilisation d'au moins un port au niveau des nœuds. Dans ce contexte, les améliorations apportées par notre algorithme, *TabouMRWA*, sont moindres, d'où la diminution du gain.



**Figure 4.11 Nombre de ports en fonction du nombre de longueurs d'onde par fibre  $F$ , pour TabouMRWA et BPHT avec  $K = 240/F$  (NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $W = 4$ ,  $Tr = \text{Rand20}$ )**



**Figure 4.12 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de fibres par lien (NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $F \times K = 240$ ,  $W = 4$ ,  $Tr = \text{Rand20}$ )**

**Expérience 3.5: Effet de la variation du nombre de fibres par lien,  $F$ , du nombre de longueurs d'onde par bande,  $W$ , et du niveau de trafic,  $Tr$**

Dans cette expérience, nous faisons varier le nombre de longueurs d'onde par bande,  $W$ , pour différentes valeurs de  $F$  et pour différents niveaux de trafic. Les différentes valeurs de  $W$  sont : 1, 4, 6, 10, 12, 15, 20, 30. Les niveaux du facteur  $F$  sont les suivants : 1, 2 et 4. La variation du niveau de trafic est proportionnelle à celle du facteur  $F$ . En d'autres termes, si le nombre de fibres par lien double, le niveau du trafic double également. Le niveau du trafic initial est égal à  $\text{Rand5}$ . Lorsque le niveau augmente, le patron du trafic ne change pas. La demande de chaque paire est uniquement multipliée par le nombre de fibres par lien. Cependant, le niveau du facteur  $K$  est maintenu constant et vaut 60. Les Tableaux 4.14, 4.15 et 4.16 indiquent, pour chacun des algorithmes, le nombre de ports en fonction de  $W$  pour différentes valeurs de  $F$  et pour différents niveaux de trafic,  $Tr$ . La Figure 4.13 montre le gain de *TabouMRWA* par rapport au *BPHT*. Le gain est plus net pour un trafic quatre fois supérieur au trafic initial et pour un nombre de longueurs d'onde par bande supérieur à 10. Lorsque le trafic devient important dans le réseau, notre algorithme fait beaucoup mieux que le *BPHT* pour des bandes comprenant de plus de 10 longueurs d'onde. En effet, dans ces conditions, il devient plus difficile pour le *BPHT* de garder les chemins optiques dans les mêmes fibres et à l'intérieur d'une fibre dans une même bande d'onde.

**Tableau 4.14 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT avec  $F = 1$  et  $Tr = \text{Rand5}$**

**(NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $K = 60$ )**

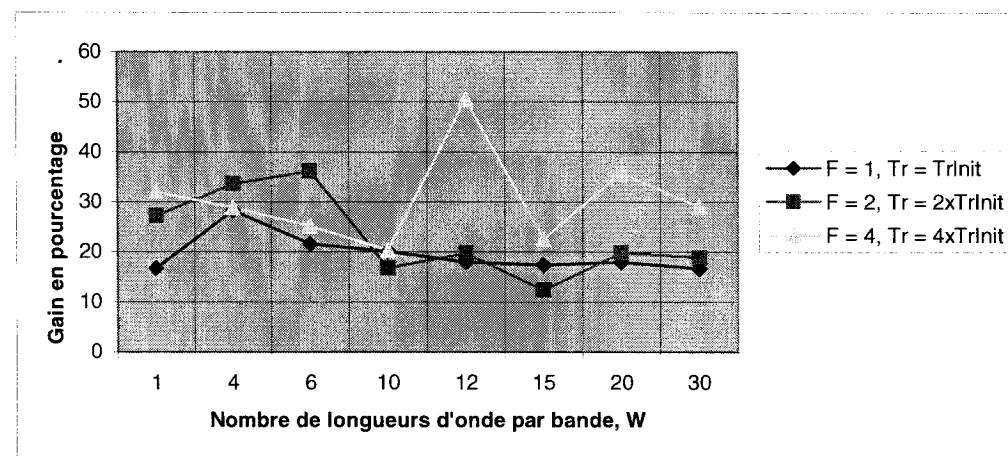
Algorithme	$F = 1, Tr = TrInit$							
	W							
	1	4	6	10	12	15	20	30
<b>BPHT</b>	1246	810	903	1016	1046	1094	1135	1204
<b>TabouMRWA</b>	1037	581	708	812	858	904	931	1003

**Tableau 4.15 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT avec  $F = 2$  et  $Tr = 2 \times TrInit$   
(NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $K = 60$ )**

Algorithme	$F = 2, Tr = 2 \times TrInit$							
	W							
	1	4	6	10	12	15	20	30
BPHT	2270	929	1040	1273	1392	1513	1652	1847
TabouMRWA	1653	617	664	1059	1118	1326	1323	1498

**Tableau 4.16 Nombre de ports en fonction du nombre de longueurs d'onde par bande,  $W$ , pour TabouMRWA et BPHT avec  $F = 4$  et  $Tr = 4 \times TrInit$   
(NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $K = 60$ )**

Algorithme	$F = 4, Tr = 4 \times TrInit$							
	W							
	1	4	6	10	12	15	20	30
BPHT	4118	1214	1233	1420	1636	1796	2544	3008
TabouMRWA	2797	864	922	1131	809	1390	1633	2135



**Figure 4.13 Gain de TabouMRWA par rapport au BPHT en fonction du nombre de longueurs d'onde par bande et pour différentes valeurs de  $F$  et de  $Tr$   
(NSFNET,  $k = 5$ ,  $LT = 20$ ,  $M = 1000$ ,  $K = 60$ )**

### **Temps d'exécution**

Le temps moyen d'exécution du *BPHT* avoisine la seconde. Ce temps est nettement inférieur à celui de notre algorithme qui est de l'ordre de plusieurs centaines de secondes. Ce fait était prévisible, vu que le *BPHT* est essentiellement une méthode de construction. La sélection des routes utilise une méthode itérative, mais elle influence peu l'algorithme lui-même. Le routage ne représente pas un élément essentiel du *BPHT*. Une méthode de routage, tel que Dijkstra aurait très bien pu être utilisé sans affecter pour autant l'essence de l'algorithme.

## CHAPITRE V

## CONCLUSION

Dans ce mémoire, nous avons traité du problème de routage et d'affectation de longueurs d'onde dans les réseaux optiques multigranulaires. Nous avons considéré le cas particulier de l'approche statique du problème. Nous avons proposé une méthode de résolution utilisant la recherche taboue. Dans ce chapitre, nous présenterons une synthèse des travaux réalisés. Par la suite, nous exposerons les limites des méthodes proposées, puis, nous donnerons un aperçu des travaux futurs.

### 5.1. Synthèse des travaux

Le problème de routage optique consiste à trouver un chemin et une longueur d'onde pour les chemins optiques à établir dans un réseau donné en évitant que deux chemins optiques traversant la même fibre aient la même longueur d'onde. Nous avons traité ce sujet en considérant la contrainte de continuité de longueurs d'onde dans le réseau ainsi que des commutateurs multigranulaires dont l'architecture a été proposée par Cao et al. [5]. Le but de notre travail était de minimiser le nombre de ports globalement dans le réseau tout en satisfaisant toutes les demandes de connexion optique.

Nous avons présenté en détail l'algorithme *TabouMRWA* que nous proposons. Dans notre méthode, nous avons traité conjointement le routage et l'affectation de longueurs d'onde. Lorsqu'une route est sélectionnée pour une paire de nœuds, nous lui affectons automatiquement des longueurs d'onde pour router tout son trafic le long du chemin retenu avant de procéder au routage du trafic d'une autre paire. La méthode de routage utilisée est basée sur les plus courts chemins et utilise le nombre de sauts comme métrique. La route affectée à une paire est sélectionnée parmi un nombre  $k$  de plus courts chemins possibles. L'algorithme d'affectation de longueurs d'onde à une paire

recherche des longueurs d'onde disponibles et affecte les premières longueurs d'onde trouvées à la paire.

Pour évaluer la performance de notre algorithme, nous avons comparé nos résultats à ceux obtenus avec l'algorithme *BPHT* proposé par Cao et al. [5]. Nous avons effectué nos comparaisons sur le réseau NSFNET et pour différentes valeurs des facteurs primaires à savoir : le nombre de fibres par lien, le nombre de longueurs d'onde par fibre et le nombre de longueurs d'onde par bande. Il est à noter que nous avons étudié les effets de la variation des trois facteurs secondaires (mais un facteur à la fois) sur le nombre de ports dans le réseau soit : le nombre de plus courts chemins entre une paire de nœuds, la longueur de la liste taboue et le nombre maximum d'itérations. Dans toutes nos comparaisons, nos résultats étaient meilleurs que ceux du *BPHT* et souvent nettement meilleurs. Cela peut se comprendre vu que le *BPHT* est un algorithme de construction alors que notre algorithme est une méthode itérative éprouvée. Souvent, les méthodes itératives conduisent à de meilleurs résultats en comparaison aux méthodes de construction. Le seul désavantage de notre algorithme par rapport au *BPHT* concerne le temps d'exécution. Le temps moyen d'exécution du *BPHT* est nettement inférieur à celui de notre algorithme. Cependant, les temps d'exécution de notre algorithme demeurent dans l'ensemble raisonnables. D'ailleurs, le temps d'exécution n'est pas un élément important dans une approche statique du problème de routage optique.

## 5.2. Limitations des travaux

Dans l'ensemble, nous avons obtenus de bons résultats. Cependant, notre travail comporte certaines limites. Tout d'abord, afin d'améliorer le temps d'exécution de notre algorithme, nous n'avons pas considéré tous les chemins possibles entre une paire de nœuds. Nous n'avons considéré qu'un certain nombre  $k$  de plus courts chemins non nécessairement optimaux. Ce choix se base sur l'idée que les routes des bonnes solutions du problème intuitivement devraient pour la plupart se trouver parmi les  $k$  plus courts chemins entre les paires de nœuds. Cependant, cela ne signifie pas que les routes des meilleures solutions se trouvent parmi ces derniers. Donc en se limitant à un sous-

ensemble des routes possibles entre une paire de nœuds, on risque de perdre de très bonnes solutions.

Ensuite, nous avons imposé que tout le trafic d'une paire soit routé sur le même chemin afin de faciliter le regroupement des chemins optiques de la paire dans une même fibre et à l'intérieur d'une fibre dans une même bande tout en réduisant le temps d'exécution de l'algorithme. Cette restriction rend impossible certaines combinaisons de routes et de longueurs d'onde qui pourraient éventuellement conduire à de très bonnes solutions, meilleures que celles qu'on a obtenues.

Puis, nous avons privilégié la sélection des routes au détriment de l'affectation des longueurs d'onde. Notre algorithme change essentiellement les routes des paires de nœuds et à chaque changement de route il ne fait qu'affecter les premières longueurs d'ondes trouvées disponibles. Il espère ainsi arriver à une combinaison de routes et de longueurs d'onde qui diminuera le nombre de ports de manière appréciable. Cependant, la diminution du nombre de ports dépend à la fois de la sélection de routes et de l'affectation de longueurs d'onde. Il est donc très probable que certaines combinaisons de routes et de longueurs d'onde ne soient pas considérées. Ces combinaisons pourraient éventuellement mener à de très bonnes solutions.

Finalement, nous n'avons autorisé que des solutions réalisables. Cependant, certaines solutions irréalisables pourraient éventuellement nous conduire vers de bonnes régions de l'espace de recherche. De plus, le fait d'autoriser des solutions non faisables augmente le nombre de solutions examinées sans trop affecter le temps d'exécution et ce plus grand nombre de possibilités en retour augmente les chances d'aboutir à de bonnes solutions. Cependant, un mécanisme de « rappel » peut s'avérer nécessaire afin de ramener l'exploration vers des solutions faisables si la recherche s'éloigne un peu trop de celles-ci [31].

### 5.3. Travaux futurs

Pour résoudre le problème de routage optique, nous avons priorisé la sélection des routes. Il serait intéressant d'accorder plus d'importance à l'affectation de longueurs

d'onde. Par exemple, à chaque sélection de route pour une paire, on pourrait rechercher les meilleures combinaisons de longueurs d'onde en utilisant une méthode itérative ou encore une approche basée sur le taux d'utilisation des fibres. Pour faciliter l'affectation de longueurs d'onde, on pourrait en plus permettre aux chemins optiques d'une paire d'utiliser des routes différentes. Cependant, ces changements pourraient conduire à une augmentation appréciable du temps d'exécution de notre algorithme. Une solution pourrait être de ne pas changer la route d'une paire systématiquement à chaque itération. On pourrait alors introduire un peu d'aléatoire dans l'algorithme. À chaque itération, le changement de route aurait une probabilité  $pr$  d'être exécuté. Lorsqu'un changement de route serait réalisé, le même algorithme d'affectation de longueurs d'onde proposé dans ce mémoire serait utilisé. Dans le cas contraire, une autre méthode (itérative par exemple) serait employée pour l'affectation des longueurs d'onde.

Notre algorithme ne tient pas compte de la possibilité de pannes dans le réseau. Il serait alors intéressant d'adapter l'algorithme à cette possibilité. Il serait également intéressant d'étudier l'effet que produirait l'introduction de convertisseurs de longueurs d'onde dans le réseau. Dans ce cas, le problème d'optimisation que nous résolvons rejoindrait un problème de design de placement des convertisseurs.

## BIBLIOGRAPHIE

- [1] Christiane Dzongang: "Heuristiques Taboues pour le Routage et l'Affectation de Longueurs d'Onde dans les Réseaux Optiques", Mémoire, Université de Montréal, Ecole Polytechnique, 2003, 89 p.
- [2] Sedogbo Rolland W. Mewanou : "Algorithmes de Routage Dynamique dans les Réseaux Optiques", Mémoire, Université de Montréal, Ecole Polytechnique, 2003, 88 p.
- [3] Pin-Han Ho, Hussein T. Mouftah, Jing Wu: "A Scalable Design of Multigranularity Optical Cross-Connects for the Next-Generation Optical Internet", IEEE Journal on Selected Areas in Communications, Vol. 211 No 7, September 2003, pp. 1133-1142.
- [4] Pin-Han Ho, Hussein T. Mouftah, Jing Wu: "Routing and Wavelength Assignment with Multi-Granularity Traffic in Optical Networks" Journal of Light wave Technology, pp. 1292-1303.
- [5] Xiaojun Cao, Vishal Anand, Yizhi Xiong, Chunming Qiao: "A Study of Waveband Switching with Multilayer Multigranular Optical Cross-Connects, IEEE Journal of Selected Areas in Communications, Vol 21 No 7, September 2003, pp. 1081-1095.
- [6] Xiaojun Cao, Vishal Anand, Yizhi Xiong, Chunming Qiao: "Performance Evaluation of Wavelength Band Switching in Multi-fiber All-Optical Networks, IEEE 2003, pp. 2251-2261.

- [7] Xiaojun Cao, Vishal Anand, and Chunming Qiao: "A Waveband Switching Architecture and Algorithm for Dynamic Traffic", IEEE Communications Letters, Vol. 7, No. 8, August 2003, pp. 397-399.
- [8] Xiaojun Cao, Vishal Anand, and Chunming Qiao: "A Waveband Switching in Optical Networks", IEEE Communications Magazine, April 2003, pp. 105-112.
- [9] Aleksandar Kolarov, Bhaskar Sengupta: "An algorithm for Waveband routing and wavelength assignment in hierarchical WDM Mesh Networks", IEEE 2003, pp. 29-36.
- [10] Biswanath Mukherjee, Shun Yao: "Design of Hybrid Waveband Switched Networks with OEO Traffic Grooming", OFC Vol. 1 2003, pp. 357-358.
- [11] Biswanath Mukherjee, Shun Yao, Canhui (Sam): "Design of Hybrid Optical Networks with Waveband and Electrical TDM Switching", 2003, 6 p.
- [12] Byrav Ramamurthy, Mengke Li: "A Graph Model for Dynamic Waveband Switching in WDM Mesh Networks", IEEE Communications Society 2004, pp. 1821-1825
- [13] Izmailov R., Kolarov A., Fan R., Araki S.: "Hierarchical Optical Switching: 'A Nodal-Level Analysis'", pp. 309-313.
- [14] Izmailov R., Y. Suemura, I. Nishioka, Y. Maeno, S. Araki., S. Ganguly: "Hierarchical Routing in Layered Ring and Mesh Optical Networks", IEEE 2002, pp. 2727-2733.

- [15] Izmailov Rauf, Samrat Ganguly, Ting Wang, Yoshihiko Suemura, Yoshiharu Maeno, Soichiro Araki: "Hybrid Hierarchical Optical Networks", IEEE 2002, pp. 88-94
- [16] Izmailov Rauf, Samrat Ganguly Viktor Kleptsyn Aikaterini C. Varsou: "Non-Uniform Waveband Hierarchy in Hybrid Optical Networks", IEEE 2003, 11 p.
- [17] Izmailov Rauf, Aikaterini C. Varsou, Samrat Ganguly: "Waveband Protection Mechanism in Hierarchical Optical Networks", IEEE 2003, pp.317-322.
- [18] Yongbum Kim, Myungmoon Lee, Jintae Yu, Chul-Hee Kang, and Jinwoo Park: "Design of Hierarchical Crossconnect WDM Networks Employing a Two-Stage Multiplexing Scheme of Waveband and Wavelength", IEEE 2002, pp. 166-171.
- [19] Yongbum Kim, Myungmoon Lee, Jintae Yu, Jinwoo Park: "WDM Network Design with Waveband and Wavelength Multiplexing Scheme", pp. 568-569.
- [20] Esa Hytyia: "Dynamic Control of All-Optical WDM Networks", Thesis, Helsinki university of technology, Department of Electrical and Communication Engineering Networking Laboratory, February 2001, 145 p.
- [21] Arne John Glenstrup: "Optimised Design and Analysis of All-Optical Networks", Thesis, 2002, 304p.
- [22] Tim Granger: "Reconfigurable wavelength-switched optical networks for the Internet core", Thesis, University of Cambridge November 2003, 184 p.
- [23] Rajiv Ramaswami, Kumar N. Sivarajan: "Optical Networks - A Practical Perspective", Second Edition, Academic Press 2002.

- [24] George N. Rouskas: “Routing and Wavelength Assignment in Optical WDM Networks”.
- [25] Cisco Systems, Inc.: “Introduction to DWDM Technology”, 2000.
- [26] E. Dotaro, D. Papadimitriou, L. Ciavaglia, M. Vigoureux, R. Douville, L. Noirie : “Optical Multi-Granularity Architectural Framework”, draft-dotaro-ipo-multi-granularity-01.txt, November 2001, pp.
- [27] Rajiv Ramaswami, Kumar N. Sivarajan, “Design of Logical Topologies for Wavelength-Routed Optical Networks”, IEEE Journal on selected areas in communications, Vol. 14, No. 5, June 1996, pp. 840-851.
- [28] Byrav Ramamurthy, Biswanath Mukherjee, “Wavelength Conversion in WDM Networking”, IEEE Journal on selected areas in communications, Vol. 16, No. 7, September 1998, pp. 1061-1073, pp.
- [29] Christianne Dzongang, Philippe Galinier, Samuel Pierre: “A Tabu Search Heuristic for the Routing and Wavelength Assignment Problem in Optical Networks”, Journal of Latex Class File, Vol. 1, No 11, November 2002, pp.
- [30] Ernesto Martins, Marta Pascoal: “A New Implementation of Yen’s Ranking Loopless Paths Algorithm”, Optimization 2001, Aveiro, July 2001, pp.
- [31] Samuel Pierre: “Réseaux et systèmes informatiques mobiles – Fondements, architectures et applications.”, Presses Internationales, École Polytechnique de Montréal, 2003.