

ORSAY  
N° D'ORDRE : 6344

**UNIVERSITÉ DE PARIS-SUD  
U.F.R. SCIENTIFIQUE D'ORSAY**

**THÈSE**

présentée  
pour obtenir

Le GRADE de DOCTEUR en SCIENCES  
DE L'UNIVERSITÉ PARIS XI ORSAY

Spécialité : INFORMATIQUE

par

**Mario E. VALENCIA-PABON**

**Complexité du problème de routage dans les réseaux de  
télécommunications : anneaux, arbres et grilles.**

Soutenue le 1<sup>er</sup> décembre 2000 devant la Commission d'examen

M. Euripides Bampis	Rapporteur
M. Dominique Barth	Directeur
M. Dominique Gouyou-Beauchamps	Président
Mme. Marie-Claude Heydemann	Examineur
M. Jean-Claude König	Rapporteur
Mme. Irena Rusu	Examineur



*A mis abuelos Gabriel e Isabel*

*A mis tías Clemencia, Mercedes y Rosario*

*A mi amada esposa Martina*



Je tiens à remercier ici les membres de mon jury :

Jean-Claude König et Evripidis Bampis pour avoir accepté le travail de rapporteurs de cette thèse. Je les remercie pour leur rigueur scientifique et leurs nombreuses suggestions.

Dominique Gouyou-Beauchamps pour l'honneur qu'il m'a fait de présider ce jury.

Marie-Claude Heydemann et Irena Rusu pour leurs participation au jury, pour leurs questions et pour l'intérêt qu'elles ont manifesté pour cette thèse.

Dominique Barth qui a dirigé mon travail, pour son appui constant aussi bien sur le plan scientifique que sur le plan personnel, sa très grande disponibilité et surtout pour sa grande patience et sa bonne humeur.

Je tiens également à remercier :

Camilo Rueda, Juan Francisco Diaz et Andrés Jaramillo, professeurs en Colombie, pour m'avoir initié à la recherche scientifique.

Diana Galindo et Christophe Genolini (camarades du DEA), Michel De Rouge-mont, Yannis Manoussakis, Jean-Claude König, Dominique Barth et Marie-Claude Heydemann pour m'avoir ouvert la voie qui m'a permis de réaliser ce travail.

Dominique Barth, Sylvie Corteel, Sophie Laplante, Moaiz Ben Dhaou, Pascal Berthomé et Alain Denise pour les innombrables relectures et corrections de ma thèse, ainsi que pour les répétitions de la soutenance.

Tous les membres de l'équipe GraFComm du LRI pour leur accueil. Je remercie en particulier Maryvonne et Jean-François pour leur amitié et pour leurs fameux pots.

José Ignacio Alvarez Hamelin (Nacho) pour son amitié et pour avoir toujours su me remonter le moral.

Toutes les personnes qui m'ont encouragé lors de toutes ces années de travail : mes grands-parents (Gabriel et Isabel), mes tantes (Clemencia, Mercedes, Rosario, Mabel et Veva) et oncles (Jesús, Javier, Félix, Hector, Carlos, Gabriel et José Luis), mes parents et sœurs, mes frères-cousins (Carlos et Mauricio), ma belle-famille italienne (Gianni, Francesca, el nonno Toni, Antonella et Ricky) et tous mes amis : Jaime et Flia. Guerrero Perini, Pedro, Leo, Flia. Forero Velasco, Andrés J., Camilo, Juan Fco., Jorge H., Carlos H., Eu, Andrés B., María Clara et Flia., Flia. Fort, Cesar, Alejandro, Sylvie, Sophie, Dominique, Moaiz, Franco et Lucia, Mauro et Monica, Velia et Vittorio, Marco, Nacho, Eduardo, Elina, Pablo, Gabriel, Carina, las Guilles, Andrea De LaRossa et Sabri LaRossa, ainsi que toutes les autres personnes non mentionnées.

Martina, mon épouse, pour tout son amour et sa compréhension. Elle me donne toujours de la force pour avancer.

Mes missions en Europe, Australie et Amérique du Sud ont été financées par les projets RNRT ROM et MADPoM CNRS.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Définitions et terminologie</b>	<b>5</b>
2.1	Éléments de théorie des graphes . . . . .	5
2.1.1	Définitions basiques . . . . .	5
2.1.2	Quelques graphes particuliers . . . . .	7
2.1.3	Coloration de graphes . . . . .	10
2.1.4	Plongements de graphes . . . . .	12
2.2	Permutations . . . . .	14
2.3	Éléments de théorie de la complexité algorithmique . . . . .	14
2.4	Réseaux de communications . . . . .	15
2.5	Le problème du routage . . . . .	16
2.6	Communications globales structurées . . . . .	18
<b>3</b>	<b>État de l'art et présentation des résultats</b>	<b>21</b>
3.1	Paramètres liés au routage . . . . .	21
3.2	Mode commutation de circuits . . . . .	22
3.2.1	Résultats connus . . . . .	23
3.2.2	Résultats trouvés dans cette thèse . . . . .	26
3.3	Mode commutation de paquets . . . . .	29
3.3.1	Routage store and forward . . . . .	29
3.3.2	Routage wormhole . . . . .	31
<b>4</b>	<b>Coloration de chemins dans l'anneau</b>	<b>33</b>
4.1	Le problème de la coloration de chemins dans l'anneau $C_n$ . . . . .	34
4.2	Généralisation du théorème de Tucker . . . . .	35
4.2.1	Collections particulières de chemins . . . . .	42
4.3	Instances polynomiales . . . . .	43
4.4	Problèmes ouverts . . . . .	47
4.5	Conclusion . . . . .	48
<b>5</b>	<b>Coloration de chemins dans les arbres</b>	<b>49</b>
5.1	Équivalence avec le routage de permutations . . . . .	49
5.2	Routage de permutations particulières . . . . .	53
5.3	Complexité en moyenne . . . . .	64

5.3.1	Une borne inférieure générale . . . . .	64
5.3.2	Le nombre moyen de couleurs dans les arbres . . . . .	66
5.4	Problèmes ouverts . . . . .	68
5.5	Conclusion . . . . .	68
<b>6</b>	<b>Commutation généralisée de circuits</b>	<b>71</b>
6.1	Le modèle . . . . .	72
6.2	Routage dans la chaîne et dans l’anneau . . . . .	74
6.2.1	Preuve du théorème 34 . . . . .	77
6.2.2	Coloration des requêtes dans l’anneau . . . . .	82
6.3	Routage dans les arbres . . . . .	82
6.3.1	Preuve du théorème 36 . . . . .	86
6.3.2	Un algorithme 2-approché pour les arbres . . . . .	90
6.4	Application au routage wormhole glouton . . . . .	91
6.5	Problèmes ouverts . . . . .	92
6.6	Conclusion . . . . .	93
<b>7</b>	<b>Mode commutation de paquets</b>	<b>95</b>
7.1	Paramètres nouveaux . . . . .	95
7.2	Routage dans les chaînes et les anneaux . . . . .	97
7.3	Émulation de réseaux dans les chaînes . . . . .	100
7.3.1	Bornes pour l’émulation d’un graphe dans la chaîne et dans l’anneau . . . . .	101
7.3.2	Émulation de l’hypercube binaire par la chaîne . . . . .	102
7.3.3	Émulation de l’hypercube par la grille . . . . .	105
7.4	Conclusion . . . . .	106
<b>8</b>	<b>Conclusions</b>	<b>107</b>



# Liste des tableaux

4.1	Résultats obtenus pour le problème de coloration de chemins dans l'anneau. . . . .	48
5.1	Résultats de complexité pire cas obtenus pour le problème de coloration de chemins dans les arbres. . . . .	69
5.2	Résultats obtenus pour le nombre moyen de couleurs nécessaires pour le problème de coloration de permutations dans les arbres. . . . .	69
6.1	Résultats sur le problème de minimisation du nombre de phases de communication dans le mode CGC( $\rho$ ). . . . .	94
7.1	Résultats obtenus pour l'émulation de l'hypercube dans la grille $d$ -dimensionnelle. . . . .	106



# Table des figures

2.1	(a) La chaîne $P_5$ , (b) le cycle $C_5$ , (c) le graphe complet $K_5$ et (d) un graphe biparti $G = (V_1, V_2, E)$ . . . . .	8
2.2	(a) Un arbre binaire $T$ à 8 sommets, (b) l'étoile $ST(7)$ et (c) l'étoile généralisée $GST(\lambda)$ à 10 sommets, où $\lambda = (3, 2, 2, 1, 1)$ est une partition de l'entier 9. . . . .	9
2.3	(a) (resp. (b)) La grille $M(3, 3)$ (resp. La grille torique $MT(3, 3)$ ) à 9 sommets. . . . .	10
2.4	L'hypercube binaire $H(3)$ . . . . .	11
2.5	(a) Une étoile $ST(4)$ non orientée et une collection de chemins $\mathcal{P} = \{p_0, p_1, p_2\}$ dans $ST(4)$ , et (b) le graphe de conflit $\mathcal{G}(\mathcal{P})$ associé, isomorphe à $C_3$ . . . . .	11
3.1	(a) (resp. (b)) Collection de requêtes de communication $\mathcal{C} = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), (a_4, b_4)\}$ dans le réseau $G$ avec une possible assignation de chemins $\mathcal{P}$ (resp. $\mathcal{P}'$ ) pour $\mathcal{C}$ , et le graphe de conflit associé à $\mathcal{P}$ (resp. $\mathcal{P}'$ ). . . . .	23
4.1	Exemple d'exécution de l'algorithme de Tucker. . . . .	36
4.2	(a) (resp. (b)) Exemple de configuration possible lors de la séquence $T_1$ . (c) Exemple de configuration interdite lors de la séquence $T_1$ . . . . .	37
4.3	Les seules dix configurations possibles pour lesquelles la propriété $cv(A_e^j) \cap cv(A_f^j) \neq \emptyset$ est vérifiée. . . . .	39
4.4	Exemple de configuration possible vérifiant la propriété 1 dans le cas $1 < j < \frac{l-2}{2}$ . . . . .	40
4.5	(a) Une collection $F$ de chemins sur $C_5$ . (b) La base $B(F)$ de $F$ , où le poids $W(A_i)$ de chaque chemin $A_i \in B(F)$ est égal à 1. . . . .	44
4.6	Exemple de transformation des chemins $A_1, A_2$ et $A_3$ en gras qui sont dans $B(F)$ , où $W(A_1) = 2, W(A_2) = 1$ et $W(A_3) = 1$ . . . . .	45
4.7	Exemple du cas (a) (resp. (b) et (c)) dans la preuve du théorème 25. . . . .	47
5.1	Construction pour le sommet $v$ dans l'arbre $T$ . . . . .	51
5.2	(a) Une collection de chemins sur un arbre non orienté et le graphe de conflit $G$ associé. (b) Une collection symétrique de chemins sur un arbre orienté symétrique et le graphe de conflit $G'$ associé. . . . .	54
5.3	(a) Un arbre $T$ et une collection de chemins $\mathcal{P}$ sur $T$ . (b) Le (multi)-graphe $G = \vec{G}_T(\mathcal{P})$ associé à $T$ et à $\mathcal{P}$ . . . . .	56

5.4	(a) Un arbre $T$ et une collection de chemins $\mathcal{P}$ sur $T$ . (b) Le (multi)- graphe orienté $G = \vec{G}_T(\mathcal{P})$ associé à $T$ et $\mathcal{P}$ . (c) Un circuit eulérien de $G$ . (d) Construction d'un arbre $T'$ et d'un ensemble de chemins $\mathcal{P}'$ représentant la permutation circulaire $\sigma = (1, 8, 9, 10, 6, 11, 5, 12, 13, 2,$ $7, 4, 3)$ des sommets de $T'$ , à partir de $T$ et $\mathcal{P}$ . . . . .	57
5.5	Construction partielle de l'instance $I'$ à partir de l'instance $I$ . . . . .	60
5.6	(a) L'arbre $T$ et la collection augmentée de chemins $\mathcal{P}$ sur $T$ dans la figure 5.3(a). (b) Le (multi)-graphe $G = \vec{G}_T(\mathcal{P})$ associé avec $T$ et $\mathcal{P}$ dans (a). . . . .	63
6.1	(a) (resp. (b)) Exemple du graphe de $\rho$ -conflit, avec $\rho = 2$ , associé à une collection quelconque de chemins $\mathcal{P}$ sur $P_6$ (resp. sur $C_{10}$ ). . . . .	75
6.2	(a) (resp. (b)) Application du raccourcissement des chemins fait par la construction donnée dans la preuve du lemme 3 (resp. 4) à l'instance montrée dans la figure 6.1.a (resp. 6.1.b), où $\rho = 2$ . . . . .	77
6.3	Exemples de la nouvelle solution $x'$ obtenue pour la propriété 10. . . . .	80
6.4	L'arbre binaire enraciné $T_\rho^n$ , avec $n = 6$ et $\rho = 1$ . . . . .	84
6.5	Sous arbre minimal dans $T_\rho^n$ contenant les sommets $v_{j-1}^i$ et $v_j^i$ dans $CL(i)$ . . . . .	85
6.6	(a) (resp. (b)) Cas 1 (resp. 2) du lemme 6. (c) La pire configuration possible pour le cas 2 du lemme 6. . . . .	86
6.7	Sous-arbre binaire isomorphe enraciné dans chacun des sommets $v_j^M \in$ $CL(M)$ construit dans la partie II. . . . .	87
6.8	Construction partielle de l'instance $I'$ . . . . .	88
6.9	(a) Arbre $T$ enraciné au sommet $r$ avec chemins $p_1, p_2$ et $p_3$ sur celui- ci, (b) les ensembles d'arcs $entrée(p_i)$ , et (c) les deux ordres totaux 2-candidat possibles sur les chemins $p_i, 1 \leq i \leq 3$ . . . . .	91

# 1 Introduction

Depuis quelques années, nous assistons à un développement important des réseaux de télécommunications et à la nécessité d'une réalisation efficace des communications dans ces réseaux. L'utilisation des réseaux de télécommunications, réseaux informatiques de petite taille (Local Area Network ou LAN) ou de très grande taille (Wide Area Network ou WAN) comme Internet, a augmenté considérablement. De plus, les perspectives actuelles d'intégrer différents types des données comme le texte, l'audio et la vidéo font également augmenter la demande d'une bande passante grande (partage des ressources) et d'une latence courte (garantie de qualité-service) dans les réseaux de communications. De même, les machines parallèles et distribuées demandent des mécanismes de communication très sophistiqués.

Malheureusement, la détermination d'une stratégie efficace pour le routage de l'information dans les réseaux mène souvent à des problèmes NP-difficiles. Beaucoup de propositions ont été faites pour palier à cette difficulté. La plupart des stratégies proposées pour le problème de routage de l'information dans les réseaux consistent à diviser le problème en deux sous-problèmes, sélectionner des routes dans un réseau de communication et trouver une planification des mouvements d'informations tout au long des routes affectées.

Cette thèse présente une étude de la complexité algorithmique liée à la détermination des stratégies de routage optimales dans les réseaux de communications. Plus précisément, nous allons étudier certains cas restrictifs du problème de routage comme le routage de permutations dans des réseaux ayant une topologie d'interconnexion assez simple comme les anneaux, les arbres et les grilles. En effet, d'une part dans les réseaux de communication standards actuels, qui utilisent la nouvelle technologie optique tels que les réseaux SONET (Synchronous Optical NETwork) (voir [2]), les topologies les plus fréquentes sont l'anneau, les arbres et les arbres d'anneaux (voir [51]). D'autre part, beaucoup des algorithmes généraux pour le problème du routage usent, comme stratégie de base, le routage des permutations (voir [45, 20]).

Dans le chapitre 2, nous présentons la terminologie et les outils mathématiques nécessaires tout au long de notre étude : les éléments de la théorie des graphes et de la théorie de la complexité algorithmique ainsi que les différents modèles de routage dans les réseaux de communication.

Le chapitre 3 présente "*l'état de l'art*" du problème de routage dans les réseaux de communication ainsi que les principaux résultats trouvés dans cette thèse. Nous avons étudié principalement deux modes de commutation utilisés pour le routage de l'information dans les réseaux de communications : le mode de commutation de circuits et le mode de commutation de paquets. Le mode de commutation de circuits

(routes virtuelles dans le réseau) est utilisé, par exemple, dans les réseaux dits *tout-optiques* qui utilisent la technologie WDM (Wavelength Division Multiplexing). Ce problème est fréquemment modélisé comme un problème particulier de coloration de chemins dans un graphe orienté symétrique qui représente la topologie du réseau. Les chapitres 4, 5 et 6 seront consacrés à ce mode de commutation. Le mode de commutation de paquets est le mode le plus classique pour le routage de l'information dans les réseaux. Nous étudierons brièvement les modèles de routage store-and-forward et wormhole dans le chapitre 7.

Le chapitre 4 est consacré au problème de coloration de chemins dans l'anneau. Étant donné un anneau et une collection de chemins sur celui-ci, nous montrons que si  $l \geq 4$  chemins sont nécessaires pour couvrir complètement l'anneau tels que leur graphe de conflit associé est un cycle, alors il existe un algorithme en temps polynomial qui utilise au plus  $(\frac{l-1}{2})$  fois le nombre optimal de couleurs nécessaires pour colorier une telle collection, de sorte que deux chemins ayant une arête en commun sont coloriés avec des couleurs distinctes. Ainsi, notre résultat généralise celui de Tucker qui avait montré, sous ces hypothèses, que le nombre de couleurs nécessaires était au plus  $\frac{3}{2}$  fois le nombre optimal. Nous montrons aussi quelques instances particulières du problème qui peuvent être résolues efficacement en temps polynomial.

Dans le chapitre 5 nous étudions le problème de la coloration de chemins dans les arbres. Nous nous concentrons sur le problème particulier de la coloration des ensembles de chemins représentant des permutations des nœuds d'un arbre. Nous montrons que le problème de calculer une coloration minimale pour ces ensembles de chemins dans les arbres est un problème NP-difficile, même pour des permutations particulières comme les *involutions* et les *permutations circulaires* dans les arbres de degré borné par une constante. Nous nous intéressons aussi dans ce chapitre au problème de déterminer le nombre moyen de couleurs nécessaires pour colorier toutes les permutations à  $n$  éléments dans les arbres à  $n$  nœuds. Plus précisément, nous montrons que le nombre moyen de couleurs nécessaires pour colorier toute permutation à  $n$  éléments dans une chaîne à  $n$  nœuds est égal à  $\frac{n}{4} + o(n)$ . Nous obtenons aussi des bornes inférieure et supérieure pour ce problème dans le cas des arbres quelconques.

Afin de mieux comprendre la difficulté inhérente de certaines instances du problème du routage par chemins arc-disjoints par rapport à d'autres instances du même problème et essayer ainsi de diminuer l'écart entre elles, nous généralisons dans le chapitre 6 le mode de commutation de circuits. En effet, les problèmes de coloration que nous étudions dans les chapitres 4 et 5 sont issus du problème du routage par chemins arc-disjoints qui fait partie du mode de commutation de circuits et dans lequel deux chemins qui partagent au moins un lien du réseau doivent être affectés à deux phases de communication (deux couleurs) différentes. Nous relaxons cette contrainte de conflit entre chemins en introduisant un paramètre entier  $\rho > 0$ . Nous disons ainsi que deux couleurs distinctes doivent être affectées à deux chemins s'ils partagent un même lien  $e$  du réseau tel que  $e$  se trouve dans la  $i$ -ème (resp.  $j$ -ème) position d'un des chemins (resp. de l'autre chemin) et  $|i - j| < \rho$ , pour un entier  $\rho$  donné. Ce nouveau paramètre entier  $\rho$  va nous permettre d'introduire une mesure

(i.e. une échelle de temps) pour le partage des ressources d'un réseau. Nous analysons la complexité algorithmique de cette généralisation du mode de commutation de circuits dans les chaînes, les anneaux et les arbres.

Finalement, dans le chapitre 7 nous étudions certains aspects du problème de routage dans le mode de commutation de paquets. Nous nous concentrons sur le problème de l'émulation de l'hypercube à  $n$  nœuds par la grille  $d$ -dimensionnelle à  $n$  nœuds sous le modèle de routage store-and-forward. D'une part, nous montrons qu'il existe une stratégie en temps polynomial pour émuler l'hypercube binaire dans la grille  $d$ -dimensionnelle avec un nombre optimal d'étapes de communication en introduisant des buffers de taille  $\log_2 n$  dans les liens de la grille. D'autre part, nous montrons qu'il existe une stratégie en temps polynomial pour émuler l'hypercube binaire dans la grille  $d$ -dimensionnelle sans utiliser de buffers dans les liens de la grille, avec un nombre d'étapes de communication qui est au plus  $\frac{3}{2}$  fois le nombre optimal.





## 2 Définitions et terminologie

Nous présentons dans ce chapitre la terminologie et les outils mathématiques nécessaires tout au long de cette thèse. Nous commençons par introduire certains éléments de la théorie des graphes et de la théorie de la complexité algorithmique. Ensuite, nous introduisons les modèles et la terminologie de la théorie du routage dans les réseaux de communications que nous utiliserons dans ce document.

### 2.1 Éléments de théorie des graphes

Dans cette section, nous introduisons les concepts de base de la théorie des graphes. Nous renvoyons à [15, 11] pour plus de détails.

Un *graphe*  $G = (V(G), E(G))$  (ou simplement  $G = (V, E)$  lorsqu'il n'y a pas d'ambiguïté) *non orienté* est constitué d'un ensemble fini de sommets  $V$  et d'un ensemble d'arêtes  $E$ . Le nombre de sommets de  $G$  est appelé *ordre* de  $G$ . Une *arête* entre les sommets  $u$  et  $v$  dans  $V$  est une paire de sommets que l'on notera  $\{u, v\} = \{v, u\}$ . De même on décrit un *graphe orienté*  $G = (V, A)$ , où  $A$  désigne un ensemble de *couples* (ordonnés) de sommets appelés les *arcs* du graphe. Un arc du sommet  $u$  vers le sommet  $v$  sera noté  $(u, v)$ . De plus, pour tout arc  $(u, v)$  du graphe  $G$ , on dit que le sommet  $u$  est l'*extrémité initiale* et le sommet  $v$  est l'*extrémité finale* d'un tel arc. On dit aussi que  $u$  est un *prédécesseur* immédiat de  $v$  et  $v$  est un *successeur* immédiat de  $u$  s'il existe un arc  $(u, v)$  dans  $G$ . Un arc de  $G$  de la forme  $(u, u)$  est appelé *boucle*. Un graphe  $G$  est dit *orienté symétrique* si, lorsque  $(u, v)$  est un arc de  $G$ , alors il en est de même pour  $(v, u)$ . Tout graphe non orienté  $G = (V, E)$  peut être transformé en un graphe orienté symétrique  $G' = (V, A)$  en remplaçant chaque arête  $\{u, v\}$  dans  $G$  par deux arcs  $(u, v)$  et  $(v, u)$  dans  $G'$ . Dans un graphe non orienté  $G = (V, E)$  (resp. orienté  $G = (V, A)$ ) nous avons les définitions suivantes.

#### 2.1.1 Définitions basiques

► Le *demi-degré extérieur* d'un sommet  $v$  de  $G$ , noté  $d_G^+(v)$ , est le nombre d'arcs de  $G$  d'extrémité initiale  $v$ . Le *demi-degré intérieur* de  $v$ , noté  $d_G^-(v)$ , est le nombre d'arcs de  $G$  d'extrémité finale  $v$ .

► Une arête  $\{u, v\}$  de  $G$  est dite *incidente* à  $u$  et à  $v$ , et  $u$  et  $v$  sont deux sommets dits *adjacents*. Une arête de la forme  $\{u, u\}$  est appelée *boucle*. L'ensemble des som-

ments adjacents à un sommet donné  $v$  de  $G$ , qu'on note  $N_G(v)$ , est appelé l'*ensemble de voisins* de  $v$ .

► On appelle *degré* d'un sommet  $v$  de  $G$ , et on note  $d_G(v)$ , le nombre d'arêtes incidentes à  $v$ . On appelle *degré maximum* de  $G$ , et on note  $\Delta_G$ , le maximum des degrés des sommets de  $G$ . On appelle *degré minimum* de  $G$ , et on note  $\delta_G$ , le minimum des degrés des sommets de  $G$ . Le graphe  $G$  est *régulier* si tous ses sommets ont le même degré, c'est-à-dire, si  $\Delta_G = \delta_G$ ;  $G$  est dit alors  $\Delta_G$ -régulier.

► Un graphe  $G$  orienté ou non est dit *simple* s'il ne contient ni boucles, ni arcs (arêtes) multiples. Par la suite, sauf précision, nous ne traiterons que de *graphes simples*.

► Dans le cas orienté ou non, un graphe  $G' = (V, E')$  est un *graphe partiel* de  $G$  si  $E' \subseteq E$ ;  $G'$  est aussi appelé *sous-graphe couvrant* de  $G$ . Un graphe  $G' = (V', E')$  est un *sous-graphe partiel* de  $G$  si  $V' \subseteq V$  et  $E'$  est un sous-ensemble d'arcs (d'arêtes) de  $G$  ayant leurs deux extrémités dans  $V'$ ; dans le cas où  $E'$  est l'ensemble de tous les arcs (arêtes) de  $G$  ayant leurs deux extrémités dans  $V'$ , on dit que  $G'$  est le *sous-graphe* de  $G$  induit par  $V'$ .

► On appelle *chaîne* entre deux sommets  $u$  et  $v$  d'un graphe  $G$  non orienté, une suite  $\mu = (e_1, e_2, \dots, e_k)$  d'arêtes de  $G$  telle que chaque arête  $e_i$ ,  $1 \leq i \leq k$ , ait une extrémité en commun avec l'arête précédente  $e_{i-1}$  (sauf  $e_1$  pour qui  $u$  joue le rôle de cette extrémité), et une extrémité en commun avec l'arête suivante  $e_{i+1}$  (sauf  $e_k$  pour qui  $v$  joue le rôle de cette extrémité). On appelle *longueur* d'une chaîne  $\mu$ , le nombre d'arêtes qui  $\mu$  contient.

► On appelle *chemin* entre deux sommets  $u$  et  $v$  d'un graphe  $G$  orienté, une suite  $\mu = (e_1, e_2, \dots, e_k)$  d'arcs telle que, pour tout arc  $e_i$  (avec  $i < k$ ), l'extrémité terminale de  $e_i$  coïncide avec l'extrémité initiale de  $e_{i+1}$ . Nous dirons qu'une chaîne entre les sommets  $u$  et  $v$  d'un graphe  $G$  non orienté est un *chemin*, si nous définissons le sommet  $u$  (resp.  $v$ ) comme l'extrémité initiale (resp. finale) d'une telle chaîne. Nous noterons par  $u \rightsquigarrow v$  (resp.  $\langle u, v \rangle$ ) un chemin orienté (non orienté) entre les sommets  $u$  et  $v$ .

► Un chemin (ou chaîne) qui n'utilise pas deux fois le même arc (arête) est dit *simple*. Un chemin (ou chaîne) qui ne passe pas deux fois par le même sommet est dit *élémentaire*. Par la suite, sauf précision, nous ne traiterons que des *chemins (ou chaînes) élémentaires*.

► Étant donnés deux sommets  $u$  et  $v$  d'un graphe  $G$ , on appelle *distance* entre  $u$  et  $v$ , et on note  $dist(u, v)$ , la plus petite longueur d'une chaîne d'extrémités  $u$  et  $v$ . On appelle *diamètre* de  $G$ , et on note  $D_G$ , le maximum, pris sur toutes les paires de sommets  $w$  et  $z$  de  $G$ , de la distance entre  $w$  et  $z$ .

- ▶ On appelle *cycle* dans un graphe (orienté ou non), un chemin simple dont les extrémités initiale et finale sont confondues. Par la suite, sauf précision, nous ne traiterons que de cycles élémentaires, c'est à dire n'utilisant pas deux fois le même sommet.
- ▶ On appelle *cycle eulérien* dans un graphe  $G$  (orienté ou non) un cycle (pas nécessairement élémentaire) passant, une fois et une seule, par tous les arcs (arêtes) de  $G$ . Un graphe qui possède un cycle eulérien est dit *graphe eulérien*.
- ▶ Un graphe  $G$  est dit *connexe* si deux sommets quelconques de  $G$  sont reliés par une chaîne.
- ▶ On appelle *stable* d'un graphe (orienté ou non), un ensemble de sommets deux à deux non adjacents.
- ▶ Un *couplage* dans un graphe non orienté  $G$  est un sous-graphe partiel  $G'$  de  $G$  avec  $d_{G'}(v) = 1$  pour tout sommet  $v$  de  $G'$ . Si  $G'$  est un graphe partiel de  $G$  tel que  $d_{G'}(v) = 1$  pour tout sommet  $v$ , alors  $G'$  est appelé un *couplage parfait* de  $G$ .
- ▶ Deux graphes  $G$  et  $G'$  sont dits *isomorphes* s'il existe une bijection  $f$  de  $V(G)$  sur  $V(G')$ , telle que, pour tous sommets  $u$  et  $v$  dans  $V(G)$ ,  $\{u, v\} \in E(G)$  si et seulement si  $\{f(u), f(v)\} \in E(G')$ .

### 2.1.2 Quelques graphes particuliers

- ▶ On note  $P_n$  le graphe *chaîne* non orienté défini par  $V(P_n) = \{0, 1, \dots, n-1\}$  et  $E(P_n) = \{(i, i+1) : 0 \leq i \leq n-2\}$  (voir figure 2.1(a)).
- ▶ On note  $C_n$  le graphe *cycle (ou anneau)* non orienté défini par  $V(C_n) = \{0, 1, \dots, n-1\}$  et  $E(C_n) = \{(i, i+1 \bmod n-1) : 0 \leq i \leq n-1\}$  (voir figure 2.1(b)).
- ▶ On appelle *graphe complet*, et on note  $K_n$ , le graphe non orienté à  $n$  sommets dont chaque sommet a pour ensemble de voisins à tous les autres (voir figure 2.1(c)).
- ▶ Un graphe  $G = (V, E)$  (orienté ou non) est dit *biparti* s'il existe une partition de  $V$  en deux sous-ensembles  $V_1$  et  $V_2$  tel que chacun d'eux soit un stable. On écrit également  $G = (V_1, V_2, E)$  (voir figure 2.1(d)).
- ▶ On appelle *arbre*, et on note par  $T$ , un graphe non orienté connexe et sans cycles. Un graphe  $T = (V', E')$  est appelé *arbre de recouvrement* d'un graphe  $G = (V, E)$  si  $V' = V$ ,  $E' \subseteq E$ , et  $T$  est un arbre. Un arbre est dit *raciné* s'il possède un sommet distingué appelé *racine*. On dit qu'un sommet  $u$  est le *père* d'un autre sommet  $v$  d'un arbre raciné  $T$ , si  $u$  est le premier sommet dans l'unique chemin élémentaire entre  $v$  et le sommet racine de  $T$ . Dans ce cas, le sommet  $v$  est appelé *fil* du sommet  $u$ . On appelle *feuille* d'un arbre, un sommet de degré 1. On appelle *hauteur* d'un

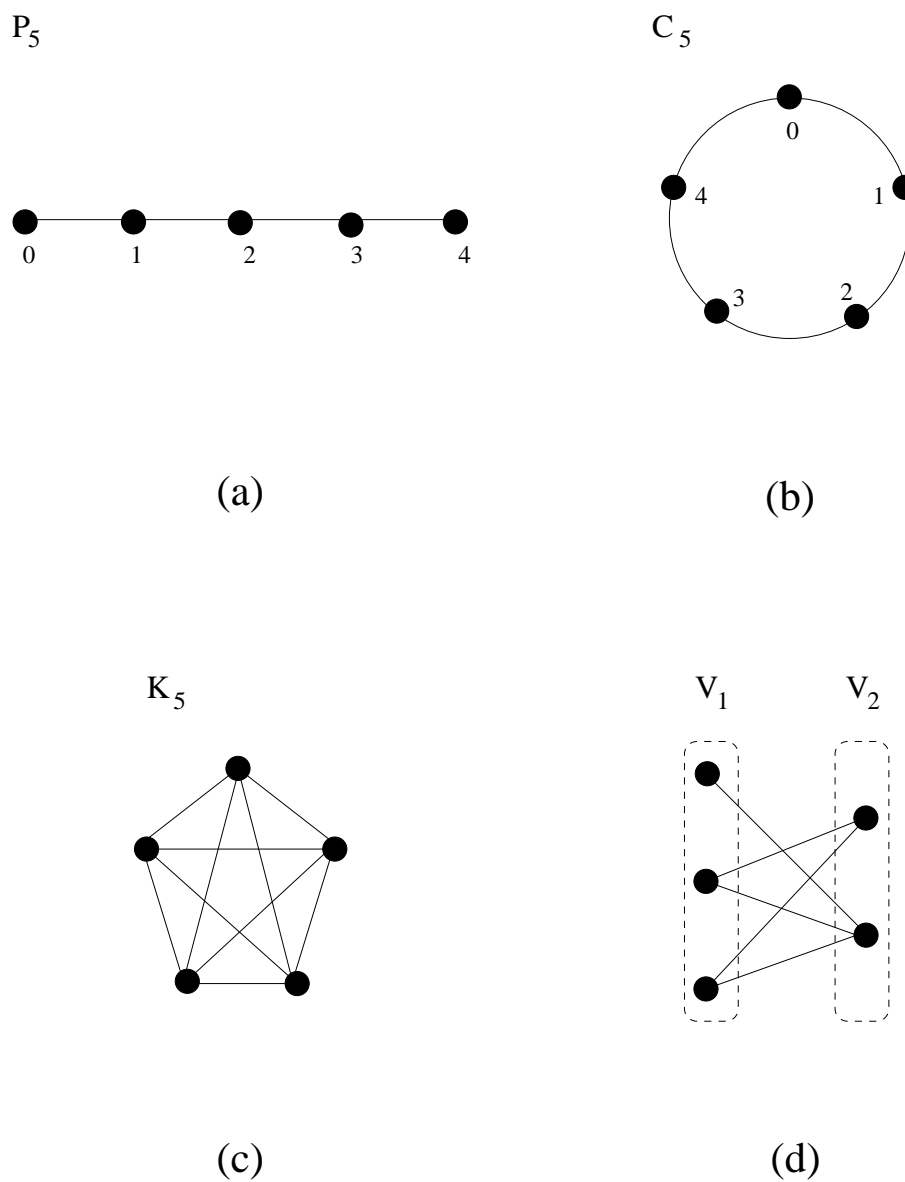


FIGURE 2.1 – (a) La chaîne  $P_5$ , (b) le cycle  $C_5$ , (c) le graphe complet  $K_5$  et (d) un graphe biparti  $G = (V_1, V_2, E)$ .

arbre raciné, la plus grande distance de la racine à une feuille. On appelle *arbre  $k$ -aire* un arbre raciné tel que le degré de la racine est au plus égal à  $k$ , et le degré de tout autre sommet est au plus égal à  $k + 1$  (voir figure 2.2(a)).

► On note  $ST(n)$ , et on appelle *étoile*, l'arbre à  $n$  sommets ayant  $n - 1$  feuilles (voir figure 2.2(b)).

► Pour un entier positif  $n$  et une partition  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$  de l'entier  $n - 1$  donnés, avec  $k > 2$ , on note  $GST(\lambda)$  et on appelle *étoile généralisée*, l'arbre à  $n$  sommets ayant  $k$  branches de longueur  $\lambda_1, \dots, \lambda_k$  (voir figure 2.2(c)).

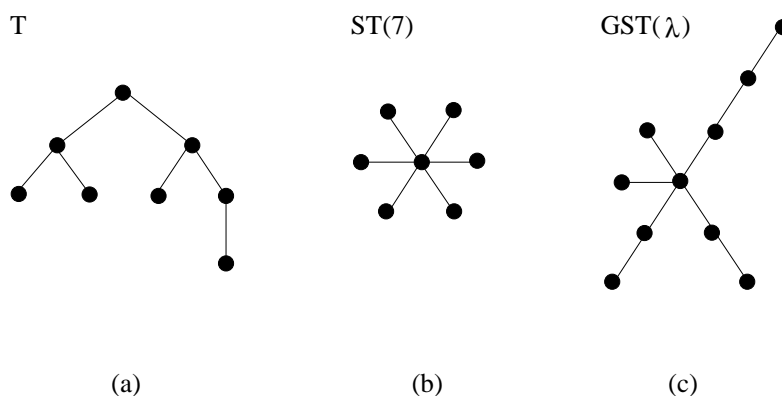


FIGURE 2.2 – (a) Un arbre binaire  $T$  à 8 sommets, (b) l'étoile  $ST(7)$  et (c) l'étoile généralisée  $GST(\lambda)$  à 10 sommets, où  $\lambda = (3, 2, 2, 1, 1)$  est une partition de l'entier 9.

► Soient  $G' = (V', E')$  et  $G'' = (V'', E'')$  deux graphes non orientés ; on note  $G' \square G''$ , et on appelle *somme cartésienne* de  $G'$  et  $G''$ , le graphe non orienté  $G$  dont les sommets sont tous les couples  $(x', x'')$  où  $x'$  est un sommet de  $G'$  et  $x''$  est un sommet de  $G''$ . Deux sommets  $(x', x'')$  et  $(y', y'')$  de  $G$  sont voisins si et seulement si  $x' = y'$  et  $x''$  est voisin de  $y''$  dans  $G''$  ou si  $x'' = y''$  et  $x'$  est voisin de  $y'$  dans  $G'$ . Par convention, chaque sommet  $(x', x'')$  de  $G$  sera noté  $x'x''$  afin d'éviter toute confusion.

► On note  $M(a_1, a_2, \dots, a_d)$  (resp.  $MT(a_1, a_2, \dots, a_d)$ ), et on appelle *grille* (resp. *grille torique*)  $d$ -dimensionnelle, le graphe composé de la somme cartésienne de  $d$  chaînes (resp.  $d$  cycles) à  $a_i$  sommets,  $1 \leq i \leq d$ . Précisément,  $M(a_1, a_2, \dots, a_d) = P_{a_1} \square P_{a_2} \square \dots \square P_{a_d}$  (resp.  $MT(a_1, a_2, \dots, a_d) = C_{a_1} \square C_{a_2} \square \dots \square C_{a_d}$ ) (voir figure 2.3).

► On appelle *hypercube binaire de dimension  $n$*  et on note  $H(n)$ , le graphe dont les sommets sont les mots de longueur  $n$  sur l'alphabet  $\{0, 1\}^n$ , et dont deux sommets sont adjacents si et seulement s'ils diffèrent en une seule coordonnée. Un sommet, noté  $x_1x_2 \dots x_i \dots x_n$ , est donc relié aux sommets  $x_1x_2 \dots \bar{x}_i \dots x_n$ , avec  $i = 1, 2, \dots, n$ .  $H(n)$  est un graphe  $n$ -régulier d'ordre  $2^n$ , qui peut être défini

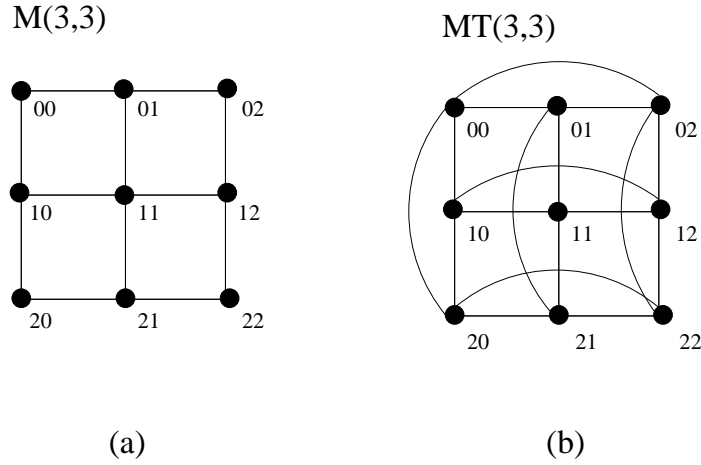


FIGURE 2.3 – (a) (resp. (b)) La grille  $M(3,3)$  (resp. La grille torique  $MT(3,3)$ ) à 9 sommets.

récurivement à partir de la chaîne  $P_2 : H(n) = P_2 \square H(n-1) = \underbrace{P_2 \square P_2 \square \dots \square P_2}_{n \text{ fois}}$ .

En effet,  $H(n)$  peut être vu comme le graphe obtenu à partir de deux copies de  $H(n-1)$  en reliant par une arête les sommets représentés par les mêmes mots. Les sommets du nouvel hypercube sont obtenus en préfixant les sommets des deux copies respectivement par 0 et par 1 (voir figure 2.4).

► Soit  $\mathcal{F}$  une famille de sous-ensembles non vides d'un ensemble fini. On appelle *graphe d'intersection* de  $\mathcal{F}$ , et on note  $\mathcal{G}(\mathcal{F}) = (V, E)$ , le graphe simple non orienté, où chaque sommet  $v_f \in V$  représente un sous-ensemble  $f \in \mathcal{F}$ , et où il existe une arête entre deux sommets  $u_f$  et  $v_g$  dans  $V$  si et seulement si l'intersection des sous-ensembles  $f$  et  $g$  dans  $\mathcal{F}$  est non vide (voir [30] pour plus de détails).

► Soient  $G$  un graphe (orienté ou non) et  $\mathcal{P}$  une collection de chemins élémentaires dans  $G$ . On appelle *graphe de conflit*, le graphe (non orienté) d'arête-intersection  $\mathcal{G}(\mathcal{P}) = (V', E')$  induit par les chemins dans  $\mathcal{P}$ . Dans la figure 2.5 nous présentons un exemple de graphe de conflit.

### 2.1.3 Coloration de graphes

► On appelle *sommet-coloration* (ou *coloration*) *propre* (resp. *arête-coloration propre*) d'un graphe  $G$  non orienté, toute coloration des sommets (resp. arêtes) de  $G$ , de sorte que toute paire de sommets adjacents (resp. d'arêtes incidents à un sommet commun) soient de couleurs différentes.

► On appelle *indice chromatique* d'un graphe  $G$  non orienté, et on note  $\chi'(G)$ , le nombre minimum de couleurs nécessaires à une arête-coloration propre de  $G$ , pris sur toutes les arête-colorations propres de  $G$ . Chaque ensemble d'arêtes de même

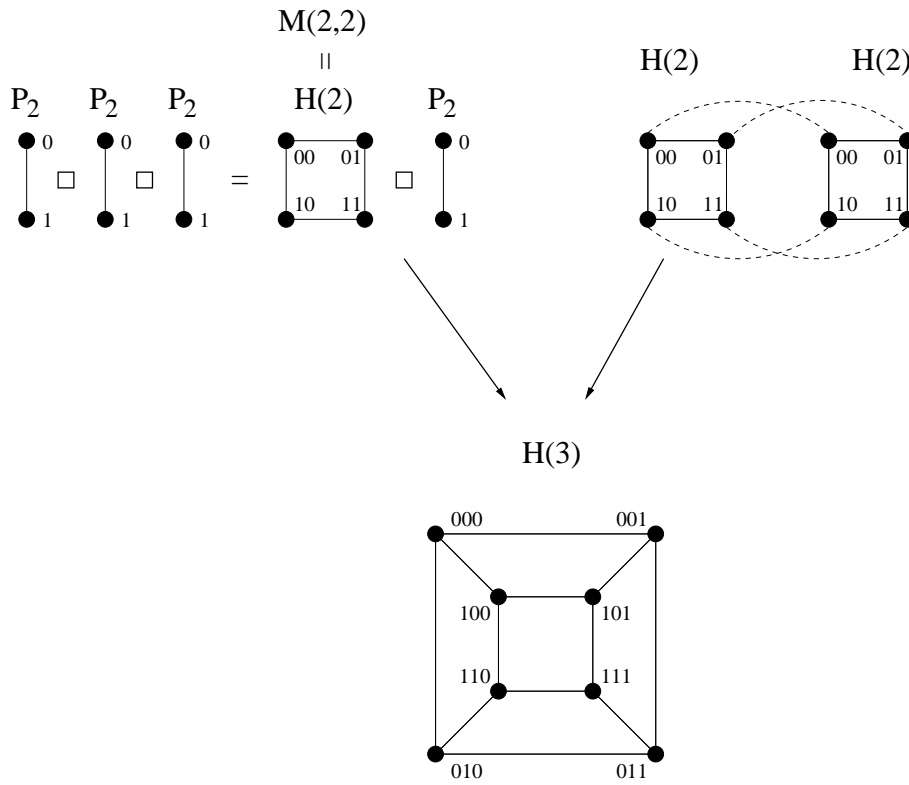


FIGURE 2.4 – L'hypercube binaire  $H(3)$ .

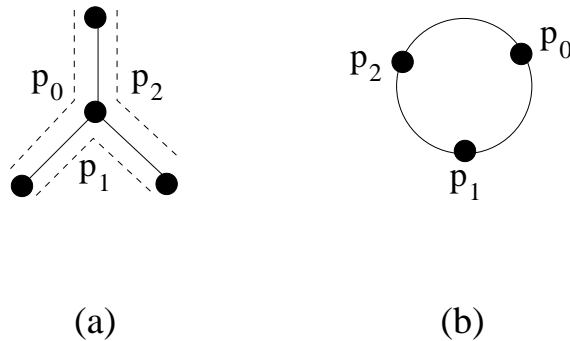


FIGURE 2.5 – (a) Une étoile  $ST(4)$  non orientée et une collection de chemins  $\mathcal{P} = \{p_0, p_1, p_2\}$  dans  $ST(4)$ , et (b) le graphe de conflit  $\mathcal{G}(\mathcal{P})$  associé, isomorphe à  $C_3$ .

couleur forme donc un couplage.

Vizing a montré le résultat suivant concernant l'indice chromatique d'un graphe.

**Théorème 1** (*Vizing, 1964 (voir [15] pag. 94)*) *Soit  $G$  un graphe simple non orienté de degré maximum  $\Delta_G$  alors,  $\Delta_G \leq \chi'(G) \leq \Delta_G + 1$ .*

► Une *clique* dans un graphe non orienté  $G$  est un sous-graphe induit complet de  $G$ . On note  $\omega(G)$  le nombre maximum de sommets dans  $G$  qui forment une clique.

► On appelle *nombre chromatique* d'un graphe  $G$  non orienté, et on note  $\chi(G)$ , le nombre minimum de couleurs nécessaires pour une sommet-coloration propre de  $G$ , pris sur toutes les sommet-colorations propres de  $G$ . Chaque ensemble de sommets d'une même couleur forme donc un stable. On sait que  $\chi(G) \geq \omega(G)$  pour tout graphe  $G$ .

► Un graphe non orienté  $G = (V, E)$  est dit *parfait* si l'on a  $\chi(G_A) = \omega(G_A)$ , pour tout sous-graphe  $G_A$  de  $G$  induit par tout  $A \subset V$  (voir [11, 30] pour plus de détails).

► Soient  $G$  un graphe (orienté ou non) et  $\mathcal{P}$  une collection de chemins élémentaires dans  $G$ . Une *coloration propre* de  $\mathcal{P}$  est l'affectation de couleurs aux chemins dans  $\mathcal{P}$ , de sorte que toute paire de chemins ayant au moins un arc (arête) dans  $G$  en commun se voient affectés à des couleurs distinctes. Il est clair que le nombre minimum de couleurs nécessaires pour une coloration propre de  $\mathcal{P}$  est égal au nombre chromatique  $\chi(\mathcal{G}(\mathcal{P}))$  du graphe de conflit  $\mathcal{G}(\mathcal{P})$  associé à  $\mathcal{P}$ .

### 2.1.4 Plongements de graphes

► Soient  $G$  et  $H$  deux graphes (orientés ou non). Un *plongement* du graphe  $G$  dans le graphe  $H$  est défini par la donnée d'une application injective  $f$  de l'ensemble des sommets de  $G$  dans l'ensemble des sommets de  $H$ , et une application injective  $\mathcal{R}_f$  de l'ensemble des arcs (arêtes) de  $G$  dans l'ensemble des chemins de  $H$ , qui associe à chaque arc  $(x, y)$  (arête  $\{x, y\}$ ) de  $G$ , un chemin  $f(x) \rightsquigarrow f(y)$  (un chemin  $\langle f(x), f(y) \rangle$ ) reliant les sommets  $f(x)$  et  $f(y)$  dans  $H$ . Si l'application  $f$  n'est pas injective (en particulier, si  $H$  a moins de sommets que  $G$ ), on parle de *placement*.

► Soit  $\mathcal{P}$  l'ensemble de chemins dans  $H$  associés, par le plongement  $(f, \mathcal{R}_f)$ , aux arcs (arêtes) de  $G$ . Beaucoup de paramètres ont été définis pour mesurer l'efficacité des plongements de graphes (voir [20] chapitre 5 et références), en particulier, la *dilatation* qui est la longueur maximale des chemins dans  $\mathcal{P}$  et la *congestion* qui est le nombre maximal des chemins dans  $\mathcal{P}$  contenant un même arc (arête) de  $H$ .

► Soient  $(f_1, \mathcal{R}_{f_1})$  un plongement du graphe  $G_1$  dans le graphe  $H_1$ , et  $(f_2, \mathcal{R}_{f_2})$  un plongement du graphe  $G_2$  dans le graphe  $H_2$ . Soit  $(u_1u_2, v_1v_2)$  un arc (arête) quelconque du graphe  $G_1 \square G_2$ . On définit le *plongement produit* qu'on note  $(f_{1,2}, \mathcal{R}_{f_{1,2}})$



du graphe  $G_1 \square G_2$  dans le graphe  $H_1 \square H_2$  par :

$$f_{1,2}(u_1 u_2) = f_1(u_1) f_2(u_2)$$

et si  $u_i \in V(G_i)$ ,  $u_i = v_i$  et  $(u_j, v_j) \in A(G_j)$ , avec  $1 \leq i \neq j \leq 2$ , étant  $f_j(u_j) \rightsquigarrow f_j(v_j) = f_j(u_j), x_0, x_1, \dots, x_k, f_j(v_j)$ , avec  $x_m \in V(H_j)$ ,  $1 \leq m \leq k$ , alors

$$\mathcal{R}_{f_{1,2}}(u_1 u_2, v_1 v_2) = \begin{cases} f_1(u_1) f_2(u_2), f_1(u_1) x_0, \dots, f_1(u_1) x_k, f_1(u_1) f_2(v_2), & \text{si } i = 1 \text{ et } j = 2 \\ f_1(u_1) f_2(u_2), x_0 f_2(u_2), \dots, x_k f_2(u_2), f_1(v_1) f_2(u_2), & \text{si } i = 2 \text{ et } j = 1 \end{cases}$$

Ho et Johnsson ont montré le résultat suivant concernant les paramètres d'un plongement produit.

**Théorème 2** (Ho et Johnsson [34]) Soient  $(f_1, \mathcal{R}_{f_1})$  un plongement du graphe  $G_1$  dans le graphe  $H_1$  avec une dilatation  $d_1$  et une congestion  $c_1$ ,  $(f_2, \mathcal{R}_{f_2})$  un plongement du graphe  $G_2$  dans le graphe  $H_2$  avec une dilatation  $d_2$  et une congestion  $c_2$ , et  $(f_{1,2}, \mathcal{R}_{f_{1,2}})$  le plongement produit du graphe  $G_1 \square G_2$  dans le graphe  $H_1 \square H_2$  avec une dilatation  $d_{1,2}$  et une congestion  $c_{1,2}$ . Alors,

$$d_{1,2} = \max(d_1, d_2) \text{ et } c_{1,2} = \max(c_1, c_2).$$

La définition de plongement produit peut être facilement généralisé comme suit.

► Soit  $(f_i, \mathcal{R}_{f_i})$  un plongement du graphe  $G_i$  dans le graphe  $H_i$ , avec  $1 \leq i \leq d$ . On considère le plongement produit  $(f_{1,2,\dots,d}, \mathcal{R}_{f_{1,2,\dots,d}})$  comme le plongement de la somme cartésienne des graphes  $G_1 \square G_2 \square \dots \square G_d$  dans la somme cartésienne des graphes  $H_1 \square H_2 \square \dots \square H_d$  défini par :

pour tout  $d$ -uplet  $u_1 u_2 \dots u_d \in V(G_1) \square V(G_2) \square \dots \square V(G_d)$ ,

$$f_{1,2,\dots,d}(u_1 u_2 \dots u_d) = f_1(u_1) f_2(u_2) \dots f_d(u_d)$$

et où pour tout arc (arête)  $(u_1 u_2 \dots u_d, v_1 v_2 \dots v_d)$  de  $G_1 \square G_2 \square \dots \square G_d$ , l'application injective  $\mathcal{R}_{f_{1,2,\dots,d}}(u_1 u_2 \dots u_d, v_1 v_2 \dots v_d)$  est définie de façon similaire à celle du plongement produit du graphe  $G_1 \square G_2$  dans le graphe  $H_1 \square H_2$  montré ci-dessus.

Soient  $d_i$  et  $c_i$  les valeurs respectives de la dilatation et de la congestion du plongement  $(f_i, \mathcal{R}_{f_i})$  du graphe  $G_i$  dans  $H_i$ , avec  $1 \leq i \leq d$ , et  $d^*$  et  $c^*$  les valeurs respectives de la dilatation et de la congestion du plongement produit  $(f_{1,2,\dots,d}, \mathcal{R}_{f_{1,2,\dots,d}})$  du graphe  $G_1 \square G_2 \square \dots \square G_d$  dans  $H_1 \square H_2 \square \dots \square H_d$ . Par la définition de somme cartésienne des graphes, le plongement produit  $(f_{1,2,\dots,d}, \mathcal{R}_{f_{1,2,\dots,d}})$  peut être vu comme le produit des plongements  $(f_{1,2,\dots,d-1}, \mathcal{R}_{f_{1,2,\dots,d-1}})$  et  $(f_d, \mathcal{R}_{f_d})$ . Ainsi, en utilisant cette hypothèse de récurrence et le théorème 2, on obtient le résultat suivant.

**Théorème 3** (Ho et Johnsson [34])

$$d^* = \max(d_i : 1 \leq i \leq d) \text{ et } c^* = \max(c_i : 1 \leq i \leq d).$$

## 2.2 Permutations

On note  $\mathcal{S}_n$ , le groupe symétrique de toutes les permutations sur  $[n] = \{1, 2, \dots, n\}$  (i.e., l'ensemble de toutes les applications bijectives de l'ensemble  $\{1, 2, \dots, n\}$  dans lui même). Soit  $\sigma$  une permutation dans  $\mathcal{S}_n$ , alors  $\sigma$  est appelée *involution* (resp. *permutation circulaire*) si elle peut être décomposée en *cycles de permutation* de longueur au plus deux (resp. si elle peut être décomposée en un seul cycle de permutation de longueur  $n$ ). Nous renvoyons à [58] (chapitre 6 et références) pour les définitions et résultats concernant les permutations.

## 2.3 Éléments de théorie de la complexité algorithmique

Dans cette section nous introduisons les concepts de base de la théorie de complexité algorithmique. Nous renvoyons à [27, 55, 44, 35] pour plus de détails sur les notions et les résultats de la théorie de la complexité et des algorithmes approchés.

La complexité introduit une mesure associée à des algorithmes décrits dans un modèle de calcul. Cette mesure nous permet de déterminer si un problème est plus difficile qu'un autre et de comprendre pourquoi certains problèmes ont un degré de difficulté inhérent. Deux mesures classiques sont introduites en informatique, *le temps* qui mesure le nombre d'étapes élémentaires nécessaires à un algorithme et *l'espace* qui mesure la taille de la mémoire nécessaire à cet algorithme. Ces deux mesures sont des fonctions asymptotiques de la taille des données.

► On dit qu'un problème est *traitable* s'il existe un algorithme polynomial pour le résoudre, c'est-à-dire un algorithme dont le nombre d'étapes élémentaires est borné pour les entrées de taille  $n$  par  $cn^k$  pour deux constantes  $c$  et  $k$ . Un des modèles classiques de calcul utilisé pour l'étude de la complexité d'un problème est celui de la *machine de Turing* (voir [27, 44] pour plus de détails).

► Un *problème de décision* est tel que, étant donnée une instance de celui-ci, sa solution est soit "oui", soit "non". On associe à un problème de décision un langage  $L \subseteq \Sigma^*$ , c'est-à-dire un sous-ensemble de mots de longueur finie sur un alphabet fini  $\Sigma$ , l'ensemble des symboles lus par une machine de Turing. Une instance positive  $x \in \Sigma^*$  du problème est un mot de  $L$ , alors qu'une instance négative  $x$  du problème est un mot du complément de  $L$ . On considère que les instances  $x$  du problème sont représentées par chaînes binaires.

► Un langage  $L$  est dans la classe NP (resp. dans la classe P) si toute instance  $x \in L$  est acceptée par une machine de Turing non-déterministe (resp. déterministe) en temps polynomial en fonction de  $|x|$ .

► Étant donnés deux langages  $L_1 \subseteq \Sigma_1^*$  et  $L_2 \subseteq \Sigma_2^*$ , le langage  $L_1$  est *Karp-réductible*

à  $L_2$  s'il existe une fonction polynomiale  $f$  qui transforme toute instance  $x \in \Sigma_1^*$  en  $f(x) \in \Sigma_2^*$  telle que  $x \in L_1$  si et seulement si  $f(x) \in L_2$ .

► Un langage  $L$  est NP-*difficile* si tout langage dans NP est Karp-réductible à  $L$ . Un langage  $L$  est NP-*complet* si  $L \in \text{NP}$  et  $L$  est NP-difficile.

► Un *problème d'optimisation* est défini par un triplet : un ensemble d'instances, un ensemble de solutions possibles pour chaque instance (c'est-à-dire qui satisfont certaines contraintes) et une fonction de coût sur cet ensemble de solutions. On cherche à trouver un optimum (maximum ou minimum) pour cette fonction de coût. Tout problème d'optimisation peut être transformé en un problème de décision, en introduisant un seuil  $k$  de telle façon que l'on doit décider s'il existe une solution de coût supérieur à  $k$  dans le cas d'un problème de maximisation, ou de coût inférieur à  $k$  dans le cas d'un problème de minimisation. Ainsi, on dit qu'un problème d'optimisation est NP-difficile si le problème de décision associé est NP-difficile.

► Étant donné un problème soit de décision, soit d'optimisation, on dit qu'il est *non traitable* si et seulement s'il est NP-difficile.

► Étant donné un problème d'optimisation  $\mathcal{P}$ , pour toute instance  $x \in \mathcal{P}$ , on note  $\text{OPT}(x)$  la valeur optimale parmi toutes les solutions possibles pour  $x$ , c'est-à-dire la valeur minimale (resp. maximale) de la fonction de coût associée à  $\mathcal{P}$  sur l'instance  $x$  si  $\mathcal{P}$  est un problème de minimisation (resp. maximisation).

Soit  $A$  un algorithme polynomial tel que, pour toute instance  $x \in \mathcal{P}$ , il trouve une solution pour  $x$  de valeur  $A(x)$ . On dit que  $A$  est un algorithme  $\epsilon$ -approché pour  $\mathcal{P}$  si pour toute instance  $x \in \mathcal{P}$ ,  $A(x) \leq \epsilon \text{OPT}(x)$ . Il faut remarquer que  $0 < \epsilon \leq 1$  si  $\mathcal{P}$  est un problème de maximisation, et  $\epsilon \geq 1$  si  $\mathcal{P}$  est un problème de minimisation ; étant égal à 1 uniquement dans le cas optimal.

## 2.4 Réseaux de communications

Un *réseau de communication* est constitué de différentes entités, appelées nœuds qui sont pas nécessairement identiques, et qui sont connectés par des liens de communication grâce auxquels ils échangent information.

► Un lien de communication reliant deux nœuds voisins peut être composé d'un ou plusieurs canaux bidirectionnels. On définira la *bande passante* d'un lien comme le nombre de canaux qu'il supporte. On appelle *paquet*, la quantité de données d'information qui peut être envoyée tout le long d'un canal dans une unité de temps. Ainsi, le nombre de paquets qui peut être échangés sur un même lien dans une unité de temps est au plus le nombre de canaux qu'il possède. A chaque canal est associée une *file d'attente* (ou "*buffer*" en anglais), ce qui permet d'échanger les données sur le canal indépendamment des autres canaux. Par la suite, sauf précision, on assume que la bande passante d'un lien est égale à 1.

- Une *requête de communication* est un message qui va être émis par un nœud  $u$  vers un nœud  $v$  dans le réseau. Ce message peut être : un ou plusieurs paquets de données, une demande d'établissement de canal de communication, ou une demande de libération de canal.
- Un nœud du réseau peut représenter simultanément deux entités :
  - *Un consommateur* : il s'agit d'une entité pouvant être l'origine et/ou la destination de requêtes de communication dans le réseau. Par exemple, un processeur et sa mémoire locale, une station de travail connectée à une carte réseau, ou un réseau local connecté à un réseau plus large.
  - *Un routeur* : c'est l'entité du nœud qui gère localement le trafic du réseau ainsi que les paquets émis ou reçus. Le routeur détermine, à chaque étape, le lien de sortie affecté à chaque paquet en entrée.
- Un réseau de communication sera modélisé par un graphe  $G = (V, A)$  orienté symétrique, où l'ensemble des sommets  $V$  du graphe représente les nœuds du réseau et chaque lien du réseau est représenté par un couple d'arcs orientés en sens opposés dans le graphe. Chaque arc du graphe représente un lien de communication monodirectionnel entre deux routeurs. Nous parlerons donc, indifféremment d'un réseau ou du graphe qui le modélise.

## 2.5 Le problème du routage

Considérons un réseau de communication modélisé par un graphe  $G = (V, A)$  orienté symétrique. Soient  $(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$  une collection de requêtes de communication dans  $G$  tel que pour  $i = 1, 2, \dots, m$ , le sommet  $a_i$  veut envoyer un message  $m_i$  au sommet  $b_i$ . Le problème du routage des requêtes de communication  $(a_i, b_i)$  dans  $G$  consiste à minimiser le temps pour délivrer tous les messages  $m_i$  de leur origine à leur destination, tout en respectant les contraintes imposées pour les ressources du réseau (c-à-d, tailles des buffers, bande passante, etc). Pour résoudre ce problème de routage, nous devons déterminer la façon comment chaque message  $m_i$  va être acheminé dans  $G$  et la façon dont ces messages vont transiter dans le réseau. Ainsi, nous devons déterminer une fonction dite *fonction de routage* dans le premier cas, et choisir un *mode de commutation* dans le deuxième cas.

- Une *fonction de routage* dans un graphe  $G = (V, A)$  est définie par une fonction  $R = V \times V \rightarrow \mathcal{P}(A)$  qui détermine, pour chaque couple  $(a_i, b_i)$ , un chemin dans le réseau allant du sommet  $a_i$  au sommet  $b_i$ . La fonction de routage est dite *statique* ou *déterministe* si elle associe un chemin unique à chaque requête de communication. Cependant, une fonction de routage statique peut poser des problèmes notamment de congestion quand le trafic n'est pas homogène et quand plusieurs messages provenant de sources différentes sont routés à travers un même lien. Pour faire face aux problèmes de congestion, on peut implanter des fonctions de routage dites *adaptatives*, où le chemin est construit dynamiquement, en fonction du trafic.

Les modes de commutation que nous considérons dans cette thèse sont les suivants.

► *Le mode commutation de paquets.* Dans ce mode, les communications dans le réseau s'effectuent par **étapes**. Chaque étape constitue une unité de temps. Lors d'une étape, chaque couple de noeuds du réseau peuvent échanger des données d'information de façon *synchrone*, c'est-à-dire, qu'une étape de communication est complètement réalisée avant que la suivante ne commence. De plus, par notre hypothèse de bande passante unitaire, à chaque étape de communication, chaque lien du réseau peut être traversé par au plus un seul paquet. Dans ce mode, nous distinguons deux types de routage :

- *Routage par stockage intermédiaire (store-and-forward).* Dans ce cas, on suppose que chaque message est composé d'un seul paquet d'information de taille fixée. À chaque étape, chaque message est totalement stocké (*store*) dans le buffer correspondant au canal courant traversé avant d'être propagé vers le noeud suivant (*forward*) (voir [45, 20, 57] et références).
- *Routage ver de terre (wormhole).* Dans ce cas, chaque message est décomposé en petits paquets de taille fixée appelés *flits*. En effet, la taille d'un flit est égale à la taille du buffer d'un canal. Le premier flit contient l'adresse du noeud destinataire, et le reste contiennent le message proprement dit. Ainsi, à chaque étape, la tête du message, c'est-à-dire, le premier flit avance d'un canal sur son chemin, chaque fois que cela est possible. Le reste du message (les flits restants) avancent derrière lui comme dans un pipeline, libérant le buffer du dernier canal, qui stocke la fin du message (le dernier flit du message). Ce dernier canal est alors disponible pour un nouveau message. Ainsi, une fois que le flit de tête a été affecté à un canal, ce canal ne peut transmettre aucun flit d'un autre message, tant que tout le message originel n'est pas passé. Si l'en-tête est bloquée, c'est-à-dire si le ou les canaux de sortie sont utilisés par d'autres messages, la propagation du message est stoppée et les flits restent stockés dans les buffers des canaux qu'ils occupent (voir [20, 57]).
- *Routage wormhole glouton.* Ce type de routage est une variation du routage wormhole dans lequel on suppose qu'il n'y a pas de buffers sur les canaux du réseau. Ainsi, une fois que le premier flit d'un message commence à avancer, tous les flits d'un tel message ne s'arrêteront que jusqu'à qu'ils atteignent sa destination finale (voir [13, 56]).

Ainsi, dans ce mode de commutation de paquets, le problème du routage d'une collection de requêtes dans un réseau, consiste à minimiser le nombre d'**étapes** nécessaires pour délivrer tous les messages de leur origine à leur destination. De plus, dans le mode store-and-forward il faut minimiser aussi la taille des **buffers** des canaux, puisque c'est une ressource limitée.

► *Le mode commutation de circuits.* Le mécanisme qu'il met en œuvre établit un chemin virtuel entre un noeud émetteur et un noeud récepteur à l'aide des routeurs. Après l'établissement du circuit de connexion, le message suit ce circuit de façon

transparente pour le routage. À la fin de la communication le circuit de connexion est libéré. Dans ce mode, les communications dans le réseau s'effectuent par **phases**. Chaque phase constitue une unité de temps. Une phase de communication est formée par un ensemble de circuits de connexion deux-à-deux sans conflit dans l'utilisation des ressources du réseau. En général on suppose que la communication des messages est synchrone<sup>1</sup>, c'est-à-dire, qu'une phase de communication est complètement réalisée avant que la suivante ne commence, et on suppose qu'il n'y a pas de buffers sur les canaux du réseau. Il existe plusieurs variantes de ce mode de commutation, qui dépendent de la façon de définir un conflit entre une paire de circuits de connexion. Pour ce qui concerne notre étude, nous analyserons principalement le suivant type de conflit :

- *Routage par chemins arc-disjoints*. Dans ce mode, on dit que deux circuits de connexion (ou chemins) sont en conflit s'ils partagent un même canal (ou arc) du réseau. Ainsi, une phase de communication est constituée par un ensemble de chemins deux-à-deux arc-disjoints. Ce type de routage est utilisé par exemple dans les réseaux *tout-optiques* qui utilisent la technologie *WDM* (i.e. multiplexage en longueur d'onde) (voir [52]).

Ainsi, dans ce mode de commutation de circuits, le problème du routage d'une collection de requêtes dans un réseau, consiste à minimiser le nombre de **phases** nécessaires pour délivrer tous les messages de leur origine à leur destination.

## 2.6 Communications globales structurées

Lorsqu'un mouvement de données doit être réalisé sur un réseau, il peut être totalement arbitraire, ou posséder une structure prédéterminée. Les communications structurées les plus fréquentes dans un réseau  $G = (V, E)$  sont :

► **l'échange total personnalisé** : (*All-to-All*) chaque nœud du réseau envoie un message différent à tous les autres. En fait, la collection de requêtes de communication  $\mathcal{C}$  dans  $G$  est définie par :  $\mathcal{C} = \{(u, v) : u, v \in V, u \neq v\}$ .

► **la distribution** : (*One-to-All*) un nœud émetteur  $u_0 \in V$  envoie à chacun des nœuds du réseau un message particulier. En fait, la collection de requêtes de communication  $\mathcal{C}$  dans  $G$  est définie par :  $\mathcal{C} = \{(u_0, v) : v \in V, v \neq u_0\}$ . Il existe une variation de ce type de communication structuré appelée *One-to-many* dans laquelle le nœud émetteur  $u_0$  envoie à chacun des nœuds dans un ensemble donné  $W \subset V$  un message particulier. Dans ce dernier cas, la collection de requêtes de communication  $\mathcal{C}$  dans  $G$  est définie par :  $\mathcal{C} = \{(u_0, v) : v \in W, v \neq u_0\}$ .

► **la permutation** : soit  $n$  le nombre de sommets dans  $G$ . On suppose que chaque sommet dans  $G$  est étiqueté de façon arbitraire par un entier différent dans l'ensemble  $\{1, 2, \dots, n\}$ . Une collection de requêtes de communication  $\mathcal{C}$  dans  $G$  est

---

1. Il faut remarquer que la notion de synchronisme dans les deux modes de commutation que nous étudions, à savoir, commutation de paquets et commutation de circuits, n'est pas la même.

dite une permutation, s'il existe une permutation  $\pi \in \mathcal{S}_n$  telle que  $\pi(i) = j$  si et seulement si  $(i, j) \in \mathcal{C}$ , avec  $i \neq j$ .

► **les  $k$ -relations** : tout nœud du réseau est la source et la destination de au plus  $k$  requêtes de communication. On peut remarquer que une permutation est une 1-relation et que une communication *All-to-All* est une  $(n - 1)$ -relation. Cependant, le contraire n'est pas toujours vrai.

Dans cette thèse, nous considérons principalement que les communications dans le réseau sont totalement arbitraires ou bien elles correspondent au cas de permutations ou  $k$ -relations.





# 3 État de l'art et présentation des résultats

Nous présentons dans ce chapitre l'état de l'art du problème du routage dans les réseaux de communications ainsi que les principaux résultats trouvés tout au long de cette thèse. Nous donnons les résultats les plus importants concernant la complexité algorithmique du problème, les algorithmes approchés dans le cas des communications arbitraires, et certains résultats optimaux dans le cas des communications structurées. Nous commençons par le routage dans le mode commutation de circuits et finalement nous abordons brièvement le routage classique dans le mode commutation de paquets.

## 3.1 Paramètres liés au routage

Soient  $G$  un graphe orienté symétrique modélisant un réseau de communication et  $\mathcal{C}$  une collection de requêtes de communication dans  $G$ . Nous noterons  $R_G(\mathcal{C})$  (resp.  $\mathcal{E}_G(\mathcal{C})$ ) le nombre minimum de phases (resp. d'étapes) de communication nécessaires pour délivrer tous les messages de  $\mathcal{C}$  de leur origine à leur destination, en utilisant le mode de commutation de circuits (resp. de paquets).

Pour résoudre un tel problème de routage, c'est-à-dire pour calculer  $R_G(\mathcal{C})$  ou  $\mathcal{E}_G(\mathcal{C})$  en fonction du mode de commutation considéré, certains auteurs (voir [45, 47, 20, 57, 7]) ont divisé le problème de routage en deux sous-problèmes :

► *Sélection des chemins* : tout d'abord on cherche une collection  $\mathcal{P}$  des chemins dans  $G$  connectant chaque requête de communication dans  $\mathcal{C}$  par où les données vont transiter, de sorte que ces chemins minimisent certains paramètres. Ces paramètres sont généralement<sup>1</sup> :

- *La congestion*, que nous noterons  $L_G(\mathcal{C})$ , où  $L_G(\mathcal{C}) = \min_{\mathcal{P}} L_G(\mathcal{C}, \mathcal{P})$ , et où  $L_G(\mathcal{C}, \mathcal{P})$  est la valeur maximale prise sur tous les arcs  $e$  de  $G$ , du nombre de chemins dans  $\mathcal{P}$  passant par  $e$ .
- *La dilatation*, que nous noterons  $D_G(\mathcal{C})$ , où  $D_G(\mathcal{C}) = \min_{\mathcal{P}} D_G(\mathcal{C}, \mathcal{P})$ , et où  $D_G(\mathcal{C}, \mathcal{P})$  est la longueur maximale des chemins dans  $\mathcal{P}$ .

---

1. La congestion et la dilatation ont été définies dans le chapitre précédent dans le cas particulier d'un plongement des graphes.

► *Itinéraire des messages* : on doit assigner à chaque message dans  $\mathcal{C}$  transitant par un des chemins de la collection  $\mathcal{P}$  choisie, un itinéraire qui va déterminer à chaque instant de temps, en fonction du mode de commutation considéré (c-à-d, à chaque phase ou à chaque étape de communication), si le message avance ou non vers sa destination. Nous noterons  $R_G(\mathcal{C}, \mathcal{P})$  (resp.  $\mathcal{E}_G(\mathcal{C}, \mathcal{P})$ ) le nombre minimum de phases (resp. d'étapes) de communication nécessaires pour délivrer tous les messages de  $\mathcal{C}$  à leur destination, à travers les chemins de  $\mathcal{P}$ .

Dans le cas du mode de commutation de paquets (le routage store-and-forward, le routage wormhole et le routage wormhole glouton), la congestion et la dilatation sont des paramètres importants puisque chacun d'eux est une borne inférieure pour le nombre minimum d'étapes. De plus, comme mentionné dans le chapitre précédent, dans le routage store and forward il est nécessaire de minimiser la taille des buffers des liens du réseau.

De même, dans le mode de commutation de circuits la congestion joue un rôle assez important. En effet, dans ce mode de commutation, la contention des canaux du réseau induite par des données d'information voulant utiliser la même ressource au même temps affecte considérablement le temps nécessaire pour délivrer les messages à leur destination. Ainsi, sous ce mode de commutation, la dilatation a une moindre importance.

Malheureusement, l'approche précédente consistant à diviser le problème de routage en deux sous-problèmes n'est pas en général satisfaisant. Par exemple, dans le cas du routage par chemins arc-disjoints, même si on a une collection de chemins  $\mathcal{P}$  associée à une collection de requêtes de communication  $\mathcal{C}$  dans un réseau  $G$ , telle que  $L_G(\mathcal{C}) = L_G(\mathcal{C}, \mathcal{P})$ , le nombre minimum de phases  $R_G(\mathcal{C}, \mathcal{P})$  associé à  $\mathcal{P}$ , représente uniquement une borne supérieure (parfois même grossière) pour  $R_G(\mathcal{C})$ . En effet, il peut exister une collection de chemins  $\mathcal{P}'$  telle que  $L_G(\mathcal{C}, \mathcal{P}') > L_G(\mathcal{C}, \mathcal{P}) = L_G(\mathcal{C})$ , et  $R_G(\mathcal{C}, \mathcal{P}) > R_G(\mathcal{C}, \mathcal{P}') = R_G(\mathcal{C})$ , comme le montre l'exemple de la figure 3.1.

## 3.2 Mode commutation de circuits

Nous présentons dans cette section, les résultats les plus importants liés au routage par chemins arc-disjoints que l'on considérera dans cette thèse. Nous renvoyons à [7, 57] pour plus d'informations.

Comme mentionné précédemment, dans le routage par chemins arc-disjoints, la longueur (c-à-d, la dilatation) des chemins choisis pour router un ensemble de requêtes de communication n'est pas réellement importante. Le paramètre important dans ce type de routage est la congestion des chemins. Cependant, la minimisation de la congestion induite par les chemins associés à une collection de requêtes de communication dans un réseau, ne permet pas forcément de minimiser le nombre de phases nécessaires pour délivrer ces messages comme on le voit dans l'exemple de la figure 3.1. Dans cette figure, on considère un réseau  $G$  et une collection de

requêtes de communication  $\mathcal{C} = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), (a_4, b_4)\}$  dans  $G$ . Dans la figure 3.1(a) (resp. figure 3.1(b)) on considère une collection de chemins  $\mathcal{P}$  (resp.  $\mathcal{P}'$ ) associée à  $\mathcal{C}$ . On peut voir que  $L_G(\mathcal{C}, \mathcal{P}) = L_G(\mathcal{C}) = 2$  et  $L_G(\mathcal{C}, \mathcal{P}') = 3$ . Cependant, à partir des graphes de conflit associés respectivement à  $\mathcal{P}$  et  $\mathcal{P}'$ , on peut constater facilement que  $R_G(\mathcal{C}, \mathcal{P}) = 4$  et  $R_G(\mathcal{C}, \mathcal{P}') = 3$ .

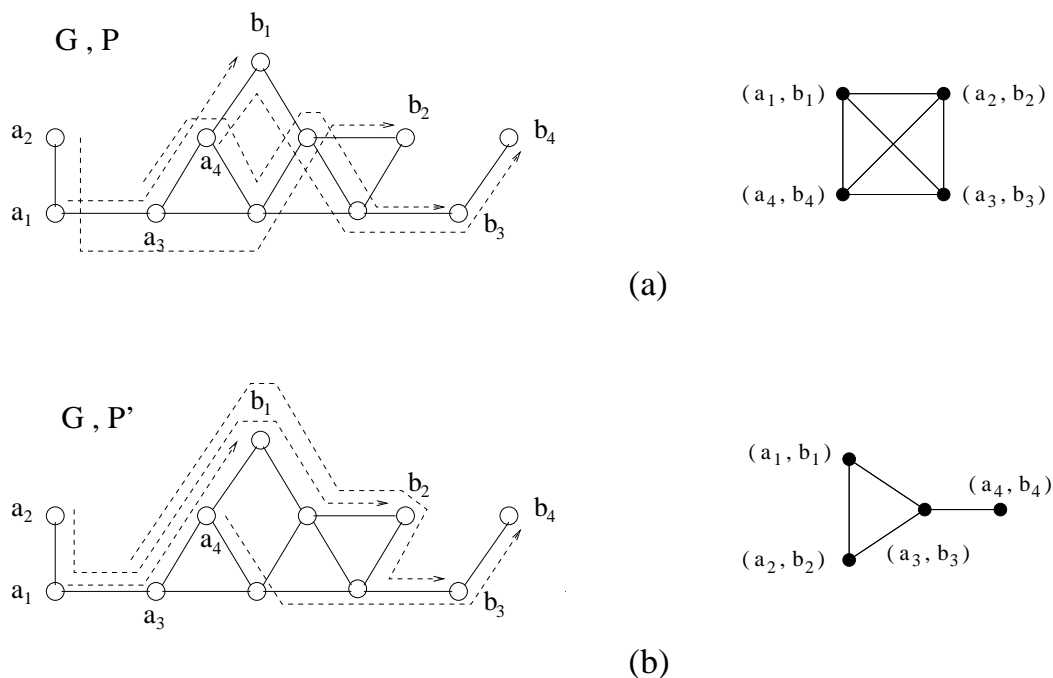


FIGURE 3.1 – (a) (resp. (b)) Collection de requêtes de communication  $\mathcal{C} = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), (a_4, b_4)\}$  dans le réseau  $G$  avec une possible assignation de chemins  $\mathcal{P}$  (resp.  $\mathcal{P}'$ ) pour  $\mathcal{C}$ , et le graphe de conflit associé à  $\mathcal{P}$  (resp.  $\mathcal{P}'$ ).

### 3.2.1 Résultats connus

Le problème de minimiser le nombre de phases de communication pour router une collection quelconque de requêtes de communication  $\mathcal{C}$  dans un réseau quelconque  $G$  (c-à-d, le problème de calculer  $R_G(\mathcal{C})$ ) est un problème NP-difficile même si  $G$  est un cycle [69] ou un arbre binaire [42, 22]. Le théorème suivant montre un encadrement pour  $R_G(\mathcal{C})$ .

**Théorème 4** (Aggarwal et al. [1]) *Étant donné un réseau  $G = (V, A)$  et une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ , alors*

$$L_G(\mathcal{C}) \leq R_G(\mathcal{C}) \leq 2\sqrt{|A|}L_G(\mathcal{C})$$

Malheureusement, même le problème de calculer  $L_G(\mathcal{C})$  dans le cas général est un problème difficile comme le montre le théorème suivant.

**Théorème 5** (Even et al. [24] (voir aussi [7])) *Étant donné un réseau  $G$  et une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ , le problème du calcul de  $L_G(\mathcal{C})$  est un problème NP-difficile.*

Les deux théorèmes précédents montrent que le problème de routage par chemins arc-disjoints est très difficile dans le cas général. Ainsi, beaucoup d'auteurs ont concentré leurs efforts soit sur quelques collections de requêtes de communication particulières, soit sur des classes de réseaux spécifiques. Dans ce qui suit, nous ferons un bilan des résultats les plus importants qui ont été trouvés à ce jour.

Le premier résultat concerne les réseaux de taille bornée (c-à-d, réseaux où le nombre de sommets est borné par une constante). Pour ces types de réseaux, Kumar et al. [42] montrent le théorème suivant.

**Théorème 6** (Kumar et al. [42]) *Soient  $G$  un réseau de taille bornée par une constante et  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $G$ . Le problème du routage de  $\mathcal{C}$  dans  $G$  par chemins arc-disjoints peut être résolu efficacement en temps polynomial.*

Dans le cas des requêtes de communication particulières comme les permutations et les  $k$ -relations, Pankaj [53, 54] (voir aussi [7]) a obtenu dans sa thèse, entre autres, les deux résultats suivants.

**Théorème 7** (Pankaj [53]) *Pour tout réseau  $G$  d'ordre  $N$  et de degré maximum  $\Delta$ , il existe une permutation pire cas pour  $\mathcal{C}$  telle que*

$$R_G(\mathcal{C}) \geq \frac{\lfloor \log_{\Delta} \frac{N}{2} \rfloor}{2\Delta}.$$

**Théorème 8** (Pankaj [53]) *Pour tout réseau  $G$  sommet transitif de diamètre  $D$  et degré  $\Delta$ , il existe une permutation pire cas  $\mathcal{C}$  telle que*

$$R_G(\mathcal{C}) \geq \left\lceil \frac{D}{\Delta} \right\rceil.$$

De plus, Pankaj a obtenu une borne inférieure de  $(\min\{k, N/2\} \cdot \lfloor \log_{\Delta} \frac{N}{2} \rfloor) / 2\Delta$  pour le nombre de phases nécessaires pour router une  $k$ -relation pire cas dans un réseau à  $N$  nœuds et de degré maximum  $\Delta$ .

Dans le cas des requêtes de communication “*One-to-All*” et “*One-to-Many*”, Bermond et al. ont donné dans [12] un algorithme polynomial pour résoudre ces problèmes de routage dans tout réseau. De plus, dans [12], il a été montré que la congestion optimale et le nombre minimum de phases de communication nécessaires pour router ces types de requêtes par chemins arc-disjoints, ont la même valeur.

Si le graphe qui modélise le réseau est un arbre, le problème du routage devient plus “simple” puisque le problème de trouver les chemins associés à une collection

de requêtes de communications donnée ne se pose pas. En effet, pour toute paire de nœuds dans un arbre, il existe un unique chemin élémentaire entre eux, et donc le problème du routage peut se voir comme un simple problème de coloration de chemins dans le réseau, ou ce qui est équivalent, comme un problème de coloration des sommets du graphe de conflit induit par les chemins associés à la collection des requêtes de communication dans le réseau. Cependant, comme nous l'avons énoncé au début de cette section, ce problème est NP-difficile, même dans le cas des arbres binaires. Si l'arbre est une chaîne, le problème de coloration des chemins sur ce réseau peut être résolu en temps polynômial. En effet, les graphes de conflit induits par une collection de chemins sur la chaîne sont des graphes dits *d'intervalles* [30] et le problème de trouver le nombre chromatique de ces graphes peut être résolu efficacement en temps polynômial [33]. Si l'arbre est une étoile, le problème de coloration des chemins sur ce réseau est équivalent à trouver une coloration optimale des arêtes d'un (multi)-graphe biparti non orienté ce qui, par le théorème de Hall, peut être fait de façon efficace en temps polynômial (voir [15] pp. 53-57).

Tant dans les chaînes que dans les étoiles, la congestion induite par toute collection de chemins sur ces réseaux est égale au nombre de couleurs (i.e. phases) nécessaires pour colorier ces chemins. En combinant ces résultats dans les chaînes et les étoiles, Gargano et al. montrent dans [29] que le problème de coloration des chemins dans une étoile généralisée peut aussi être résolu efficacement en temps polynômial. Particulièrement, dans [29] est montré le résultat suivant.

**Théorème 9** (Gargano et al. [29]) *Soit  $T$  un arbre. Pour toute collection  $\mathcal{C}$  de chemins dans  $T$  on a  $R_T(\mathcal{C}) = L_T(\mathcal{C})$  si et seulement si  $T$  est une étoile généralisée.*

Gargano et al. montrent aussi dans [29] que pour les arbres quelconques, le problème "All-to-All" peut être résolu efficacement en temps polynômial. Dans le cas des arbres binaires quelconques, Jansen [37] a construit un arbre binaire  $T$  et une collection des chemins  $\mathcal{C}$  dans  $T$  tels que  $R_T(\mathcal{C}) = 5$  et  $L_T(\mathcal{C}) = 3$ . Récemment, Erlebach et al. [23] ont montré le résultat suivant.

**Théorème 10** (Erlebach et al. [23]) *Soient  $T$  un arbre quelconque et  $\mathcal{C}$  une collection quelconque de chemins dans  $T$ . Il existe un algorithme en temps polynômial pour colorier les chemins dans  $\mathcal{C}$  qui utilise au plus  $\lceil \frac{5}{3}L_T(\mathcal{C}) \rceil$  couleurs.*

Dans le cas des cycles ou anneaux, Wilfong et Winkler montrent dans [69] que le problème du routage par chemins arc-disjoints dans ces réseaux est aussi NP-difficile. Cependant, ils montrent aussi les résultats suivants.

**Théorème 11** (Wilfong et Winkler [69]) *Soient  $G$  un anneau et  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $G$ . Il existe un algorithme en temps polynômial pour trouver une collection de chemins  $\mathcal{P}$  dans  $G$  associée à  $\mathcal{C}$  telle que  $L_G(\mathcal{C}, \mathcal{P}) = L_G(\mathcal{C})$ .*

À partir d'un résultat de Tucker [64] à propos du nombre chromatique des graphes *arc-circulaires*, qui correspondent aux graphes de conflit non orientés induits par des collections fixées de chemins dans un anneau, Wilfong et Winkler montrent le résultat suivant.

**Théorème 12** (Wilfong et Winkler [69]) *Soient  $G$  un anneau et  $\mathcal{C}$  une collection arbitraire de requêtes de communication dans  $G$ . Il existe un algorithme en temps polynomial pour trouver une collection de chemins  $\mathcal{P}$  dans  $G$  associée à  $\mathcal{C}$  et une coloration pour  $\mathcal{P}$  de sorte que  $R_G(\mathcal{C}, \mathcal{P}) \leq 2L_G(\mathcal{C}) - 1$ .*

Puisque on sait que  $R_G(\mathcal{C}) \geq L_G(\mathcal{C})$ , alors le théorème 12 de Wilfong et Winkler constitue un algorithme 2-approché pour le problème de routage par chemins arc-disjoints dans les anneaux. Récemment, Kumar [43] a trouvé un algorithme aléatoire  $(1.5 + 1/2e + o(1) \approx 1.68)$ -approché pour le problème de routage par chemins arc-disjoints dans les anneaux, avec une grande probabilité si le nombre optimal de couleurs nécessaires pour colorier une collection arbitraire de requêtes de communication dans ce réseau est  $\omega(\log n)$ , où  $n$  est le nombre de nœuds de l'anneau.

Gu et Tamaki [32] ont considéré le problème du routage de permutations dans les réseaux hypercubes et ont trouvé le résultat suivant.

**Théorème 13** (Gu et Tamaki [32]) *Pour tout ensemble de requêtes de communication  $\mathcal{C}$  représentant une permutation des sommets d'un hypercube, il existe un ensemble de chemins  $\mathcal{P}$  tel que  $R(\mathcal{C}, \mathcal{P}) = 2$ .*

Le théorème 13 de Gu et Tamaki n'est pas loin de prouver une ancienne conjecture de Szymansky [62] qui affirme qu'une seule couleur est nécessaire pour colorier toute permutation dans un hypercube. Finalement, Beauquier a montré dans [6] que le problème du routage "All-to-All" peut être calculé efficacement par des algorithmes en temps polynomial dans les anneaux, les grilles (grilles toriques)  $d$ -dimensionnelles, les hypercubes et autres graphes produit.

### 3.2.2 Résultats trouvés dans cette thèse

#### Chapitre 4 : Coloration de chemins dans l'anneau

Dans le cas des anneaux, nous traitons le problème de minimisation du nombre de phases de communication pour router une collection quelconque de requêtes de communication ayant associée une collection de chemins fixée à l'avance. En effet, ce problème est équivalent au problème de la coloration d'une collection de chemins sur un anneau non orienté. Les principaux résultats que nous avons obtenu sont les suivants.

**Résultat 1** (voir théorème 19) *Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$  avec une congestion  $L = L(F)$ . Soit  $l$  la longueur minimale de tout cycle induit du graphe de conflit associé à  $F$  tel que les chemins dans  $F$  associés à un tel cycle couvrent tout  $C_n$ . Si  $l \geq 4$ , alors il existe un algorithme en temps polynomial pour colorier  $F$  qui utilise au plus  $\lfloor (\frac{l-1}{l-2}) L \rfloor + 1$  couleurs.*

Dans le cas de collections propres de chemins (i.e. une collection de chemins est dite propre si aucun chemin n'est contenu dans un autre chemin), nous avons obtenu les résultats suivants.

**Résultat 2** (voir théorème 24) Soient  $F$  une collection de chemins sur l'anneau  $C_n$  et  $B(F)$  la base de  $F$  (i.e.  $B(F)$  est un sous-ensemble maximal de chemins dans  $F$  deux-à-deux distincts). Si  $B(F)$  est propre, alors  $F$  peut être transformée en temps linéaire en un ensemble équivalent  $F^*$  propre de chemins.

**Résultat 3** (voir théorème 25) Soit  $F$  une collection de  $m$  chemins sur l'anneau  $C_n$  telle que la longueur de chacun des chemins dans  $F$  soit dans l'ensemble  $\{1, \alpha, \alpha+1\}$ , pour tout entier  $\alpha \geq 2$ . Alors, une coloration optimale des chemins dans  $F$  peut être obtenue en temps  $O(m^{1.5})$ .

### Chapitre 5 : Coloration de chemins dans les arbres

Comme Beauquier et al. l'ont remarqué dans [7], le problème de coloration d'un ensemble de chemins représentant une permutation des sommets dans un arbre n'a pas été étudié jusqu'à présent. Ainsi, dans le chapitre 5 nous allons concentrer notre étude sur ce problème dans le cas où les chemins représentent une permutation des sommets dans les arbres. Les principaux résultats trouvés dans ce chapitre sont les suivants.

**Résultat 4** (voir théorème 26) Soient  $T$  un arbre orienté symétrique à  $n$  sommets de degré maximum  $\Delta$ , et  $\mathcal{P}$  une collection quelconque de chemins sur  $T$  avec une congestion  $L = L_T(\mathcal{P})$ . On peut construire en temps polynomial, un arbre  $T'$  orienté symétrique à  $O(n\Delta L)$  sommets de degré maximum  $\Delta' \leq 2\Delta - 1$ , et un ensemble de chemins  $\mathcal{P}'$  représentant une permutation des sommets dans  $T'$ , tels que  $R_T(\mathcal{P}) = k$  si et seulement si  $R_{T'}(\mathcal{P}') = k$ , pour un entier positif  $k$ .

**Résultat 5** (voir théorème 27) Soient  $T$  un arbre orienté symétrique et  $\mathcal{P}$  une collection de chemins orientés sur  $T$ . Alors, le problème de calculer  $R_T(\mathcal{P})$  est un problème NP-difficile dans les cas suivants :

- (a)  $T$  est un arbre binaire et  $\mathcal{P}$  est une collection symétrique de chemins sur  $T$ .
- (b)  $T$  est un arbre binaire et  $\mathcal{P}$  représente une involution des sommets de  $T$ .
- (c)  $T$  est un arbre de degré maximum  $\Delta \geq 4$ , et  $\mathcal{P}$  représente une permutation circulaire des sommets de  $T$ .
- (d)  $T$  est un arbre ayant uniquement deux sommets de degré supérieur à deux, et  $\mathcal{P}$  représente une involution des sommets de  $T$ .

Dans le cas de la complexité en moyenne du problème de coloration de chemins, nous avons obtenu les résultats suivants.

**Résultat 6** (voir proposition 3) Pour tout graphe  $G = (V, A)$  à  $n$  sommets et pour toute fonction de routage  $r$  dans  $G$ , la congestion moyenne  $\bar{L}_{G,r}$  induite par  $r$  pour router toutes les permutations dans  $\mathcal{S}_n$  vérifie,

$$\bar{L}_{G,r} \geq \frac{1}{n} \cdot \max_{U \subseteq V} \left( \frac{|U| \cdot (n - |U|)}{|c(U)|} \right)$$

où  $c(U)$  est la coupe dans  $G$  induite par tout  $U \subseteq V$ .

**Résultat 7** (voir théorème 29) *Le nombre moyen de couleurs nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets d'une chaîne à  $n$  sommets est*

$$\frac{n}{4} + \frac{\lambda}{2}n^{1/3} + O(n^{1/6})$$

où  $\lambda = 0.99615\dots$

**Résultat 8** (voir théorèmes 30 et 31) *Pour tout  $\epsilon$ , il existe  $n_0 = n_0(\epsilon)$  tel que, pour tout  $n \geq n_0$  et pour tout arbre  $T$  raciné à  $n$  sommets, étant le sommet étiqueté par l'entier  $n$  la racine de  $T$ , le nombre moyen de couleurs  $\bar{R}_T$  nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets de  $T$  vérifie,*

$$\bar{R}_T \leq \left(\frac{5}{3} + \epsilon\right) n\tilde{v}_T(1 - \tilde{v}_T)$$

où  $\tilde{v}_T = \max_i \min(|T(i)|/n, 1 - |T(i)|/n)$  et  $T(i)$  est le sous-arbre de  $T$  enraciné au sommet  $i$ , avec  $1 \leq i \leq n$ .

**Résultat 9** (voir théorème 32) *Le nombre moyen de couleurs nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets d'une étoile généralisée  $T = GST(\lambda)$  racinée à  $n$  sommets, étant le sommet étiqueté par l'entier  $n$  la racine de  $T$ , est*

$$\bar{R}_T = n\tilde{v}_T(1 - \tilde{v}_T) + O(\sqrt{n}).$$

où  $\tilde{v}_T = \max_i \min(|T(i)|/n, 1 - |T(i)|/n)$  et  $T(i)$  est le sous-arbre de  $T$  enraciné au sommet  $i$ , avec  $1 \leq i \leq n$ .

## Chapitre 6 : Commutation généralisée de circuits

Afin de mieux comprendre la difficulté inhérente de certains instances du problème de routage par chemins arc-disjoints par rapport à d'autres instances du même problème et essayer ainsi de diminuer l'écart entre elles, nous avons introduit dans ce chapitre un modèle théorique de routage qui relaxe la contrainte de conflit entre chemins. Dans ce nouveau modèle de routage, nous dirons que deux chemins virtuels  $p$  et  $q$  qui partagent un même canal  $e$  du réseau tel que  $e$  se trouve dans la  $i$ -ème (resp.  $j$ -ème) position sur  $p$  (resp.  $q$ ) sont en conflit (que nous appelons  $\rho$ -conflit) si et seulement si  $|i - j| < \rho$ , pour un certain entier positif  $\rho$  donné à l'avance. Ainsi, nous pouvons remarquer que dans ce mode de commutation généralisée de circuits paramétré par l'entier  $\rho$  (que nous appelons : mode CGC( $\rho$ )), deux requêtes de communication peuvent être délivrées lors de la même unité de temps (ou phase de communication) même si leurs chemins virtuels partagent certains canaux, à condition que ces deux chemins ne soient pas en  $\rho$ -conflit. De plus, le routage par chemins arc-disjoints que nous avons étudié dans les chapitres 4 et 5 est un cas spécial du mode CGC( $\rho$ ), où la valeur de  $\rho$  est assez grande, par exemple égale au nombre de canaux du réseau. Nous avons donc étudié la complexité algorithmique de ce nouveau mode de routage théorique sur les anneaux et sur les arbres.

Dans le cas des anneaux, nous avons obtenu les résultats suivants.



**Résultat 10** (voir théorème 34) Soient  $C_n$  l'anneau orienté symétrique à  $n$  sommets,  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $C_n$ , et  $\rho$  un entier positif quelconque. Alors, en temps polynomial, on peut obtenir une collection  $\mathcal{P}$  de chemins sur  $C_n$  pour router  $\mathcal{C}$  sous le mode  $CGC(\rho)$  telle que  $L_{C_n}^\rho(\mathcal{C}, \mathcal{P}) \leq L_{C_n}^\rho(\mathcal{C}) + 1$ , où  $L_{C_n}^\rho(\mathcal{C}, \mathcal{P})$  est le nombre maximum de chemins dans  $\mathcal{P}$  deux-à-deux en  $\rho$ -conflit et  $L_{C_n}^\rho(\mathcal{C}) = \min_{\mathcal{P}} L_{C_n}^\rho(\mathcal{C}, \mathcal{P})$ .

**Résultat 11** (voir théorème 35) Soient  $C_n$  un anneau orienté symétrique,  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $C_n$ , et  $\rho$  un entier positif quelconque. Alors, il existe un algorithme en temps polynomial  $\mathcal{A}$  pour router  $\mathcal{C}$  dans  $C_n$  sous le mode  $CGC(\rho)$  tel que :

- (i) si  $1 \leq \rho \leq \frac{n}{k}$ , pour tout  $k \geq 3$ , alors  $\mathcal{A}$  utilise au plus  $\lfloor (\frac{k}{k-1}) L_{C_n}^\rho(\mathcal{C}) \rfloor + 2$  phases de communication pour router  $\mathcal{C}$ .
- (ii) si  $\rho > \frac{n}{3}$ , alors  $\mathcal{A}$  utilise au plus  $2L_{C_n}^\rho(\mathcal{C}) + 1$  phases de communication pour router  $\mathcal{C}$ .

Dans le cas des arbres, nous avons obtenu les résultats suivants.

**Résultat 12** (voir théorème 36) Soient  $T$  un arbre binaire orienté symétrique,  $\mathcal{P}$  une collection quelconque de chemins associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $T$  et  $\rho$  un entier positif. Alors, pour toute valeur de  $\rho \geq 1$ , le problème de routage de  $\mathcal{C}$  dans  $T$  sous le mode  $CGC(\rho)$  est NP-difficile.

**Résultat 13** (voir théorème 37) Soient  $T$  un arbre orienté symétrique quelconque,  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $T$ , et  $\rho$  un entier positif. Alors, il existe un algorithme en temps polynomial 2-approché pour le problème du routage de  $\mathcal{C}$  dans  $T$  sous le mode  $CGC(\rho)$ .

### 3.3 Mode commutation de paquets

Nous présentons brièvement dans cette section, les résultats les plus importants liés au problème de routage sous le mode de commutation de paquets que l'on considérera dans cette thèse. Nous renvoyons à [20, 57] pour une information plus détaillée sur le sujet.

#### 3.3.1 Routage store and forward

Nous présentons dans cette section les résultats les plus significatifs concernant le routage “store and forward”. Pour une description plus approfondie sur ce type de routage, nous recommandons l'ouvrage de Scheideler [57].

#### Résultats connus

Le résultat le plus important concernant le routage store and forward a été trouvé par Leighton, Maggs et Rao [46, 47, 48]. Ils obtiennent qu'avec une forte probabilité, pour toute collection de chemins fixée à l'avance sur un réseau quelconque, le

nombre d'étapes de communication nécessaires pour délivrer une collection de messages transitant pour une telle collection de chemins est  $O(\text{congestion} + \text{dilatation})$ , en utilisant des buffers dans les canaux du réseau de taille constante. Formellement, le résultat de Leighton et al. est le suivant.

**Théorème 14** (Leighton et al. [46, 47, 48]) *Soient  $G$  un réseau quelconque et  $\mathcal{P}$  une collection quelconque de chemins associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ . Avec une forte probabilité, il existe un algorithme en temps polynomial pour router les messages dans  $\mathcal{C}$  en  $\mathcal{E}_G(\mathcal{C}, \mathcal{P}) = O(L_G(\mathcal{C}, \mathcal{P}) + D_G(\mathcal{C}, \mathcal{P}))$  étapes de communication, en utilisant des buffers dans les canaux de  $G$  de taille  $O(1)$ .*

Comme la congestion et la dilatation induites par une collection de chemins dans un réseau sont des bornes inférieures pour le nombre minimum d'étapes de communication, le résultat obtenu par Leighton et al. est asymptotiquement optimal. Cependant, les constantes cachées dans la notation  $O$  sont assez grandes. Par exemple, la taille des buffers peut être de l'ordre d'un million. Ainsi, Scheideler [57] a amélioré le résultat de Leighton et al., obtenant les deux résultats suivants.

**Théorème 15** (Scheideler [57]) *Soient  $G$  un réseau quelconque et  $\mathcal{P}$  une collection quelconque de chemins associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ . Avec une forte probabilité, il existe un algorithme en temps polynomial pour router les messages dans  $\mathcal{C}$  en  $\mathcal{E}_G(\mathcal{C}, \mathcal{P}) = 39(L_G(\mathcal{C}, \mathcal{P}) + D_G(\mathcal{C}, \mathcal{P}))$  étapes de communication, en utilisant des buffers dans les canaux de  $G$  de taille  $O(\log^3(L_G(\mathcal{C}, \mathcal{P}) + D_G(\mathcal{C}, \mathcal{P})))$ .*

**Théorème 16** (Scheideler [57]) *Soient  $G$  un réseau quelconque et  $\mathcal{P}$  une collection quelconque de chemins associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ . Avec une forte probabilité, il existe un algorithme en temps polynomial pour router les messages dans  $\mathcal{C}$  en  $\mathcal{E}_G(\mathcal{C}, \mathcal{P}) = O(L_G(\mathcal{C}, \mathcal{P}) + D_G(\mathcal{C}, \mathcal{P}))$  étapes de communication, en utilisant des buffers dans les canaux de  $G$  de taille 3.*

Dans le cas des chaînes et des anneaux, Kaufmann et Sibeyn [39, 40] ont montré que pour toute collection fixée  $\mathcal{P}$  de chemins associés à une collection quelconque  $\mathcal{C}$  de requêtes de communication dans ces réseaux, la stratégie de routage connue comme “*les plus loin d'abord*” est une stratégie optimale qui utilise  $\mathcal{E}_G(\mathcal{C}, \mathcal{P})$  étapes de communication pour router les messages de  $\mathcal{C}$  via les chemins de  $\mathcal{P}$ . Kaufmann et Sibeyn donnent aussi dans [39, 40] des algorithmes probabilistes pour le problème du routage de toute  $k$ -relation dans les grilles et les grilles toriques  $d$ -dimensionnelles.

Dans le cas du routage de permutations dans les arbres, Symvonis montre dans [61] qu'il existe un algorithme polynomial pour router toute permutation dans un arbre à  $n$  sommets en au plus  $n - 1$  étapes de communication, en utilisant des buffers de taille 1 dans les canaux de ces réseaux.

Pour le reste des résultats concernant certains types de communications structurées et/ou classes de réseaux particulières, nous renvoyons à [20, 57].

### Résultats trouvés dans cette thèse (chapitre 7)

Tout d’abord, nous proposons dans ce chapitre une preuve alternative à celle donnée par Kaufmann et Sibeyn dans [39, 40] pour montrer que la stratégie “*les plus loin d’abord*” est une stratégie optimale pour le problème de routage de messages dans les chaînes et les anneaux sous le mode store and forward. Cette preuve utilise deux concepts nouveaux que nous appelons “*vecteur d’état*” et “*encombrement*” associés aux canaux de ces réseaux. Ensuite, nous étudions l’émulation de l’hypercube par la chaîne et par la grille  $d$ -dimensionnelle dans ce mode de commutation de paquets. Les résultats que nous avons trouvés sur l’émulation de l’hypercube pour ces réseaux sont les suivants.

**Résultat 14** (voir théorème 40) *Une étape de communication dans l’hypercube binaire  $H(n)$  à  $2^n$  sommets peut être émulée par la chaîne  $P_{2^n}$  en  $\left\lfloor \frac{2^{n+1}}{3} \right\rfloor$  étapes de communication, en utilisant des buffers dans les liens de  $P_{2^n}$  de taille  $O(n)$ . De plus, cette émulation est optimale en fonction du nombre d’étapes.*

**Résultat 15** (voir théorème 41) *Une étape de communication dans l’hypercube binaire  $H(n)$  à  $2^n$  sommets peut être émulée par la chaîne  $P_{2^n}$  en  $2^n - 1$  étapes de communication, sans utiliser de buffers dans les liens de  $P_{2^n}$ .*

**Résultat 16** (voir théorème 43) *Soit  $n = n_1 + n_2 + \dots + n_d$  un entier positif avec  $n_i > 0$  pour  $1 \leq i \leq d$ . Une étape de communication dans l’hypercube binaire  $H(n)$  à  $2^n$  sommets peut être émulée par la grille  $d$ -dimensionnelle  $M(n_1, n_2, \dots, n_d)$  en :*

- (i)  $\max\{\left\lfloor \frac{2^{n_i+1}}{3} \right\rfloor : 1 \leq i \leq d\}$  étapes de communication, en utilisant des buffers dans les liens de  $M(n_1, n_2, \dots, n_d)$  de taille  $\max\{n_i : 1 \leq i \leq d\}$ . De plus, cette émulation est optimale en fonction du nombre d’étapes.
- (ii)  $\max\{2^{n_i} - 1 : 1 \leq i \leq d\}$  étapes de communication, sans utiliser de buffers dans les liens de  $M(n_1, n_2, \dots, n_d)$ . De plus, cette émulation utilise au plus  $\frac{3}{2}$  fois le nombre optimal d’étapes.

### 3.3.2 Routage wormhole

Nous présentons dans cette section quelques résultats concernant le routage “*wormhole glouton*”. Nous renvoyons à [20, 57, 45, 56] et références pour une description plus complète des résultats principaux sur le routage wormhole.

#### Résultats connus

Le routage wormhole glouton a été introduit par Bhatt et al. [13]. Leur principaux résultats sont les suivants.

**Théorème 17** (Bhatt et al. [13]) *Soient  $G$  une chaîne et  $\mathcal{P}$  une collection de chemins sur  $G$  associés à une collection quelconque  $\mathcal{C}$  de requêtes de communication dans  $G$ . Alors,  $O(L_G(\mathcal{P}) + D_G(\mathcal{P}))$  étapes de communication sont suffisants pour router  $\mathcal{C}$  dans  $G$ . Particulièrement,*

- (i) Si la longueur (i.e. le nombre de flits) de chaque message dans  $\mathcal{C}$  est arbitraire, alors il existe un algorithme déterministe en temps polynomial 7-approché pour router  $\mathcal{C}$  dans  $G$ .
- (ii) Si tous les messages dans  $\mathcal{C}$  ont la même longueur (i.e. le même nombre de flits), alors il existe un algorithme déterministe en temps polynomial 2-approché pour router  $\mathcal{C}$  dans  $G$ .

Dans le cas des arbres et grilles  $d$ -dimensionnelles, avec  $d$  borné par une constante, Ranade et al. obtiennent dans [56] le résultat suivant.

**Théorème 18** (Ranade et al. [56]) Soient  $G$  un réseau et  $\mathcal{P}$  une collection de chemins associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ . Si la longueur (i.e. le nombre de flits) de tous les messages dans  $\mathcal{C}$  est la même et si  $G$  est un arbre ou une grille  $d$ -dimensionnelle, avec  $d$  borné par une constante, alors  $O(L_G(\mathcal{C}, \mathcal{P}) + D_G(\mathcal{C}, \mathcal{P}))$  étapes de communication sont suffisants pour router  $\mathcal{C}$  dans  $G$ .

### Résultats trouvés dans cette thèse

Comme une application des techniques utilisées dans la section 3 du chapitre 6 pour montrer le résultat 12, nous obtenons dans la section 4 du même chapitre le résultat suivant qui montre la difficulté du problème du routage sous le mode wormhole glouton.

**Résultat 17** (voir théorème 38) Le problème de routage dans les arbres binaires sous le mode wormhole glouton est NP-difficile, même si la longueur (c-à-d, le nombre de flits) des messages est unitaire.

## 4 Coloration de chemins dans l’anneau

Considérons un anneau  $C_n$  orienté symétrique à  $n$  sommets et une collection  $\mathcal{C}$  de requêtes de communication dans  $C_n$ . Dans le chapitre 3, nous avons vu que, dans le mode de **commutation de circuits** par chemins arc-disjoints, le problème de routage des requêtes de  $\mathcal{C}$  dans  $C_n$  consiste à trouver une coloration de chemins  $\mathcal{P}$  sur  $C_n$  associés à  $\mathcal{C}$  tels qu’ils minimisent le nombre de couleurs (ou phases) nécessaires pour colorier proprement les chemins dans  $\mathcal{P}$ . De plus, nous avons vu que l’approche qui consiste à diviser ce problème de routage en deux sous-problèmes, à savoir : (i) trouver une collection de chemins  $\mathcal{P}$  pour router  $\mathcal{C}$  dans  $C_n$  telle que la congestion induite par  $\mathcal{P}$  soit minimale, c’est-à-dire,  $L_{C_n}(\mathcal{C}, \mathcal{P}) = L_{C_n}(\mathcal{C})$ ; et (ii) trouver une coloration propre minimale pour  $\mathcal{P}$ , est une approche assez utilisée. Nous savons aussi qu’une solution optimale pour chacun des sous-problèmes (i) et (ii) de façon indépendante, n’implique pas une solution optimale pour ce problème de routage. Cependant, dans le cas des anneaux ou des arbres, l’approche précédente peut donner des solutions qui sont assez proches de l’optimal. En effet, dans le cas des anneaux, Wilfong et Winkler montrent dans [69] que le sous-problème (i) peut être résolu efficacement en temps polynomial, tandis que le sous-problème (ii) est NP-difficile, mais 2-approché.

Dans ce chapitre nous allons étudier uniquement le sous-problème (ii). En effet, nous allons supposer que l’on connaît à l’avance la collection de chemins  $\mathcal{P}$  associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $C_n$ , et donc nous cherchons la solution “la plus approchée” au problème de coloration minimale des chemins dans  $\mathcal{P}$ . De plus, comme les chemins dans  $\mathcal{P}$  qui sont orientés dans le sens des aiguilles d’une montre et ceux orientés dans le sens opposé ne sont pas en conflit, on peut traiter ces deux sous-collections de chemins de façon indépendante. Ainsi, le sous problème (ii) peut être vu comme un problème de coloration de chemins non orientés sur un anneau non orienté.

Dans la section 4.1 nous donnons un aperçu des résultats concernant le problème de coloration de chemins dans l’anneau non orienté. Dans la section 4.2 nous généralisons un théorème dû à Tucker [64]. Nous montrons que si le nombre minimal de chemins nécessaires pour couvrir tout l’anneau est grand, tel que leur graphe de conflit associé est un cycle, alors on peut obtenir en temps polynomial une coloration des ces chemins qui utilise un nombre de couleurs assez proche de l’optimal. Dans la section 4.3 nous montrons certaines collections de chemins qui peuvent être coloriées opti-

malement en temps polynomial. Nous proposons quelques problèmes ouverts dans la section 4.4, et nous concluons ce chapitre dans la section 4.5. Les résultats que nous allons présenter dans ce chapitre apparaissent dans [67].

## 4.1 Le problème de la coloration de chemins dans l'anneau $C_n$

Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F = \{A_1, A_2, \dots, A_m\}$  une collection de chemins non orientés sur  $C_n$ , où chaque chemin  $A_i = \langle a_i, b_i \rangle$ ,  $1 \leq i \leq m$ , doit être vu comme le chemin allant du sommet  $a_i$  vers le sommet  $b_i$  dans le sens des aiguilles d'une montre. Soit  $cv(A_i) = \{a_{i+1}, a_{i+2}, \dots, b_i\}$  si  $a_i < b_i$ , ou  $cv(A_i) = \{a_{i+1}, a_{i+2}, \dots, n-1, 0, \dots, b_i\}$  si  $a_i > b_i$ , le sous-ensemble de sommets dans  $C_n$  traversés par le chemin  $A_i$ . Alors, le problème de la coloration de chemins dans l'anneau  $C_n$ , appelé *ARC-COLORING*, consiste à colorier les chemins dans  $F$  avec le nombre minimum de couleurs de telle façon que, pour toute paire de chemins  $A_i$  et  $A_j$  dans  $F$ , si  $cv(A_i) \cap cv(A_j) \neq \emptyset$  alors les chemins  $A_i$  et  $A_j$  doivent être coloriés avec des couleurs distinctes. Formellement, on définit ce problème de la façon suivante :

### Problème ARC-COLORING (*Tucker [64]*)

*Instance* : un anneau non orienté  $C_n$  à  $n$  sommets, une collection de chemins non orientés  $F$  sur  $C_n$  et un entier positif  $k$ .

*Question* : Existe-t-il une coloration des chemins dans  $F$  avec au plus  $k$  couleurs telle que pour toute paire de chemins  $A_i$  et  $A_j$  dans  $F$ , si  $cv(A_i) \cap cv(A_j) \neq \emptyset$ , alors les chemins  $A_i$  et  $A_j$  sont coloriés avec des couleurs distinctes ?

Le graphe de conflit non orienté associé à une collection quelconque de chemins  $F$  sur un anneau quelconque  $C_n$  est dit *graphe arc-circulaire* (voir [64, 30] et réf.). Ainsi, le problème ARC-COLORING consiste en trouver le nombre chromatique du graphe arc-circulaire induit par  $F$  et  $C_n$ . Garey et al. [28] ont montré que le problème ARC-COLORING est NP-complet. Cependant, Garey et al. ont montré aussi dans [28], que si l'on fixe le paramètre  $k$  (c-à-d, l'entier  $k$  ne fait plus partie de l'instance du problème), alors en temps  $O(mk!k \log k) = O(m)$  on peut déterminer si une collection de chemins  $F$  peut être coloriée ou non avec  $k$  couleurs, où  $m$  est le nombre de chemins dans  $F$ , et la même complexité suffit pour construire une telle coloration si elle existe, ce qui implique trivialement le corollaire suivant :

**Corollaire 1** *Soit  $L(F)$  la congestion induite par une collection de chemins  $F$  sur  $C_n$ , avec  $|F| = m$ . Si  $L(F)$  est bornée par une constante  $c$ , alors une coloration optimale de  $F$  peut être obtenue en temps  $O(mc!c \log^2 c) = O(m)$ .*

On note par  $\chi(F)$  le nombre minimum de couleurs nécessaires pour colorier une collection  $F$  de chemins sur  $C_n$ . On note par  $\omega(F)$  la plus grande taille d'un sous ensemble  $F' \subseteq F$ , tel que pour toute paire de chemins  $A_i$  et  $A_j$  dans  $F'$ ,  $cv(A_i) \cap cv(A_j) = \emptyset$ . Finalement, on note par  $L(F)$  la congestion induite par une collection

de chemins  $F$  sur  $C_n$ . De façon triviale, pour toute collection de chemins  $F$ , on a  $\chi(F) \geq \omega(F)$ . Tucker a montré dans [64] que  $L(F) \leq \chi(F) \leq 2L(F) - 1$  et que les deux bornes sont atteintes. Il faut remarquer que  $L(F)$  n'est pas nécessairement égal à  $\omega(F)$ . En fait, Tucker montre dans [64] qu'il y a des collections de chemins  $F$  pour lesquelles  $\omega(F) = 2L(F) - 1$ . Un autre résultat très intéressant trouvé par Tucker est le suivant.

**Théorème 19** (Tucker [64]) *Soient  $C_n$  l'anneau non orienté avec  $n$  sommets et  $F$  une collection de chemins non orientés sur  $C_n$  avec une congestion  $L = L(F)$ . Si au moins quatre chemins dans  $F$  sont nécessaires pour couvrir complètement  $C_n$  de telle façon que le graphe d'intersection induit pour ceux-ci est un cycle, alors le nombre de couleurs nécessaires pour colorier  $F$  est au plus égal à  $\lfloor \frac{3}{2}L \rfloor$ .*

De plus, Tucker a conjecturé que pour toute collection de chemins  $F$  sur  $C_n$ , le nombre de couleurs nécessaires pour colorier  $F$  est au plus égal à  $\lfloor \frac{3}{2}\omega(F) \rfloor$ . Cette conjecture a été démontrée par Karapetyan dans [38]. Récemment, Kumar propose dans [43] un algorithme  $(1 + 1/e + o(1))$ -approché pour ce problème qui utilise des méthodes probabilistes.

## 4.2 Généralisation du théorème de Tucker

Dans cette section, nous donnons une généralisation du théorème 19 et montrons que l'algorithme glouton proposé par Tucker dans [64] pour montrer le résultat de ce théorème a une très bonne performance dans le cas où le nombre de chemins nécessaires pour couvrir complètement l'anneau tels que leur graphe de conflit associé est un cycle est supérieure à trois. Tout d'abord, nous présentons l'algorithme approché pour le problème ARC-COLORING proposé par Tucker.

**Algorithme de Tucker :**

**Entrée :** le graphe non orienté  $C_n$  et une collection de chemins non orientés  $F$  sur  $C_n$ , avec une congestion  $L = L(F)$ .

**Sortie :** une coloration propre pour  $F$ .

1. Soit  $p$  un sommet de  $C_n$  traversée par exactement  $L$  chemins dans  $F$ . Choisir parmi tous les chemins  $A \in F$  tels que  $p \in cv(A)$ , le chemin de départ qu'on note par  $A_1 = \langle a_1, b_1 \rangle$ , tel que le sommet initial  $a_1$  du chemin  $A_1$  soit le plus proche du sommet  $p$ . On colorie le chemin  $A_1$  avec la couleur 1.
2. Pour  $i = 1, 2, \dots, |F| - 1$ , on détermine en suivant le sens des aiguilles d'une montre, le chemin  $A_{i+1} = \langle a_{i+1}, b_{i+1} \rangle$  dans l'ensemble  $F \setminus \{A_1, A_2, \dots, A_i\}$  par la règle suivante :  $A_{i+1}$  est le chemin tel que son sommet initial  $a_{i+1}$  est le plus proche sommet à droite du sommet final  $b_i$  du chemin  $A_i$ . On colorie le chemin  $A_{i+1}$  avec la couleur correspondante au plus petit entier positif non encore affecté à un autre chemin  $A_j$ ,  $1 \leq j \leq i$ , tel que  $cv(A_{i+1}) \cap cv(A_j) \neq \emptyset$ .

Dans la figure 4.1 nous donnons un exemple de l'exécution de l'algorithme de Tucker.

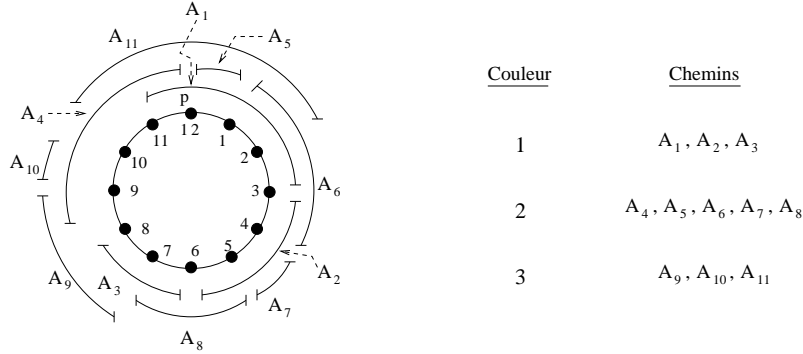


FIGURE 4.1 – Exemple d'exécution de l'algorithme de Tucker.

**Définition 1** Soient  $C_n$  l'anneau non orienté à  $n$  sommets,  $F$  une collection de chemins sur  $C_n$  et  $\mathcal{G}_{C_n}(F)$  le graphe arc-circulaire induit par l'intersection des chemins dans  $F$ . On définit la **couverture cyclique minimale** de  $F$  comme la longueur minimale de tout cycle induit du graphe  $\mathcal{G}_{C_n}(F)$  tel que les chemins dans  $F$  associés à un tel cycle couvrent tout  $C_n$ .

Dans le théorème suivant, nous donnons une généralisation du théorème 19 obtenu par Tucker.

**Théorème 20** Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$  avec une congestion  $L = L(F)$ . Soit  $l$  la couverture cyclique minimale de  $F$ . Si  $l \geq 4$ , alors le nombre de couleurs nécessaires pour colorier  $F$  en utilisant l'algorithme de Tucker est au plus égal à  $\lfloor (\frac{l-1}{l-2}) L \rfloor + 1$ .

**Preuve.** Par hypothèse, on sait que la couverture cyclique minimale de  $F$  notée par  $l$  vérifie  $l \geq 4$ . Considérons qu'on utilise l'algorithme glouton de Tucker décrit précédemment pour colorier  $F$ . Nous appellerons  $j$ -ième séquence qu'on notera par  $T_j$ , la séquence de chemins  $A_1^j, A_2^j, \dots, A_d^j, A_{d+1}^j, \dots, A_e^j, A_{e+1}^j, \dots, A_f^j$  considérés consécutivement dans cet ordre par l'algorithme de Tucker (voir figure 4.2 (a) et (b)) et qui est définie récursivement de la manière suivante : soit  $k \geq 1$  la couleur assignée par l'algorithme de Tucker au dernier chemin  $A_f^{j-1}$  de la séquence  $T_{j-1}$ . Alors,

- le premier chemin  $A_1^j$  de la séquence  $T_j$  est le premier chemin lors de la séquence  $T_{j-1}$  colorié avec la couleur  $k$ , où le chemin  $A_1^1$  de la séquence  $T_1$  correspond au chemin de départ  $A_1$  considéré par l'algorithme de Tucker. Nous noterons  $A^* = A_1^1$ .
- Le chemin  $A_d^j$  est le premier chemin de la séquence  $T_j$  après le chemin  $A_1^j$  tel que  $cv(A_d^j) \cap cv(A_1^j) \neq \emptyset$ .
- Le chemin  $A_e^j$  est le premier chemin de la séquence  $T_j$  après le chemin  $A_d^j$  tel que  $cv(A_e^j) \cap cv(A_d^j) \neq \emptyset$ .
- Finalement, le chemin  $A_f^j$  est le premier chemin après le chemin  $A_e^j$  qui traverse pour la deuxième fois, après le chemin  $A_f^{j-1}$  (pour  $j > 1$ ), l'arête à gauche (dans le sens des aiguilles d'une montre) adjacente au sommet initial du chemin  $A^*$ . En effet, le chemin  $A_f^j$  a ou bien comme sommet final le même sommet initial



du chemin  $A^*$ , ou bien  $\text{cv}(A_f^j) \cap \text{cv}(A^*) \neq \emptyset$ . Il faut remarquer que le chemin  $A_f^j$  peut correspondre au même chemin  $A_e^j$  (voir figure 4.2(b)).

Dans la figure 4.2 (a) et (b), nous montrons deux exemples pour la séquence de chemins  $T_1$ . Il faut remarquer que chaque paire de séquences consécutives d'arcs sont non disjointes.

Pour tout  $j \geq 1$ , soit  $F_j = \bigcup_{i=1}^j \{A_k^i : A_k^i \text{ est un chemin de la séquence } T_i\}$  la sous-collection de chemins qui ont été coloriés par l'algorithme de Tucker juste après la séquence  $T_j$ . Nous notons par  $\text{Col}(F_j)$  le nombre de couleurs utilisées par l'algorithme de Tucker pour colorier les chemins dans  $F_j$ . Finalement, on note par  $\omega(F_j)$  la taille maximale d'une clique du graphe de conflit induit par les chemins de  $F_j$ .

Nous montrerons par induction sur le nombre des séquences  $T_j$  que pour tout  $j \geq 1$ ,  $\text{Col}(F_j) \leq 2j + 1 + \lfloor \frac{2j}{l-2} \rfloor$ , et que la congestion  $L(F \setminus F_j)$  induite par la sous-collection de chemins  $F \setminus F_j$  vérifie  $L(F \setminus F_j) \leq L - 2j$ .

- *Cas (a) :  $j = 1$ .* Comme le chemin  $A_d^1$  est le premier chemin tel que  $\text{cv}(A_d^1) \cap \text{cv}(A_1^1) \neq \emptyset$ , où  $A_1^1 = A^*$  et  $A_d^1$  sont respectivement le premier et le  $d$ -ième chemin de la séquence  $T_1$ , alors par l'algorithme de Tucker, les chemins  $A_1^1, A_2^1, \dots, A_{d-1}^1$  sont deux à deux disjoints et donc ils peuvent être coloriés avec la même couleur. De la même façon, les chemins  $A_d^1, A_{d+1}^1, \dots, A_{e-1}^1$  sont deux à deux disjoints et donc on peut tous les colorier avec une nouvelle couleur. De plus, on a  $\text{cv}(A_e^1) \cap \text{cv}(A_f^1) = \emptyset$  (voir figure 4.2(a)) ou bien  $A_e^1 = A_f^1$  (voir figure 4.2(b)). On peut donc colorier tous les chemins  $A_e^1, A_{e+1}^1, \dots, A_f^1$  avec une nouvelle couleur. Supposons le contraire, c'est-à-dire,  $A_e^1 \neq A_f^1$  et  $\text{cv}(A_e^1) \cap \text{cv}(A_f^1) \neq \emptyset$  (voir figure 4.2(c)). Ceci implique que le graphe d'intersection induit par les chemins  $A_d^1, A_e^1$  et  $A_f^1$  est un cycle de longueur 3 qui couvre tout  $C_n$ , ce qui est une contradiction avec l'hypothèse que la couverture cyclique minimale de  $F$  est supérieure à trois. Ainsi, lors de la construction

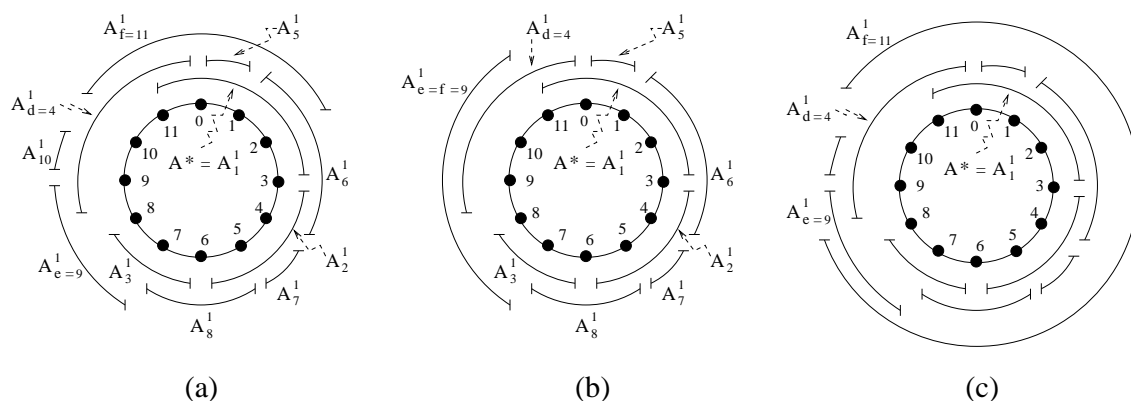


FIGURE 4.2 – (a) (resp. (b)) Exemple de configuration possible lors de la séquence  $T_1$ . (c) Exemple de configuration interdite lors de la séquence  $T_1$ .

de la séquence  $T_1$  on a besoin d'au plus trois couleurs pour colorier les chemins  $A_1^1, \dots, A_d^1, \dots, A_e^1, \dots, A_f^1$ . En fait, si le nombre de couleurs nécessaires lors de la construction de la séquence  $T_1$  est égal à 3, alors on a  $\text{cv}(A^*) \cap \text{cv}(A_f^1) \neq \emptyset$  ou bien le sommet final du chemin  $A_f^1$  est égal au sommet initial du chemin  $A^*$ . Dans le premier

cas, les chemins  $A^*$ ,  $A_d^1$  et  $A_f^1$  sont deux à deux non disjoints, ce qui implique que  $\omega(F_1)$  est égal à 3. Dans le deuxième cas, il est clair qu'il existe un sous-ensemble de chemins dans  $F_1$  qui couvrent tout  $C_n$  et tels que le graphe d'intersection induit par ceux-ci soit un cycle de longueur impaire au moins égale à  $l$ . On a donc que  $\text{Col}(F_1) \leq 3 \leq 2j+1 + \lfloor \frac{2j}{l-2} \rfloor$ . D'autre part, il est facile de voir que  $L(F \setminus F_1) \leq L-2$ , ce qui prouve le cas  $j = 1$ .

• *Cas (b) :  $j > 1$ .* On procède par récurrence. On suppose que le théorème est vrai pour  $j-1$  et on va montrer qu'il reste vrai pour  $j$ . On a donc, par hypothèse que  $\text{Col}(F_{j-1}) \leq 2(j-1) + 1 + \lfloor \frac{2(j-1)}{l-2} \rfloor$  et on a que  $L(F \setminus F_{j-1}) \leq L - 2(j-1)$ . Rappelons que pour tout  $j \geq 2$ , le premier chemin  $A_1^j$  de la séquence  $T_j$  est exactement le premier chemin de la séquence  $T_{j-1}$  colorié avec la couleur  $\text{Col}(F_{j-1})$ . Ceci implique que les chemins  $A_1^j, A_2^j, \dots, A_{d-1}^j$  peuvent être tous coloriés avec la couleur  $\text{Col}(F_{j-1})$ , puisqu'ils sont deux à deux disjoints. De plus, par construction, les chemins  $A_d^j, A_{d+1}^j, \dots, A_{e-1}^j$  sont eux aussi deux à deux disjoints et donc ils peuvent être coloriés avec une nouvelle couleur. Il reste à voir combien de couleurs sont nécessaires pour colorier les chemins  $A_e^j, A_{e+1}^j, \dots, A_f^j$ . En fait, si lors de la construction de la séquence  $T_j$ , avec  $j > 2$ , on a que  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) = \emptyset$ , alors les chemins  $A_e^j, A_{e+1}^j, \dots, A_f^j$  sont deux à deux disjoints, et donc on aura besoin d'une seule couleur en plus pour colorier tous ces chemins. Dans un autre cas, on aura besoin de deux couleurs en plus pour colorier les chemins  $A_e^j, A_{e+1}^j, \dots, A_f^j$ .

Dans la figure 4.3, nous montrons toutes les configurations possibles pour lesquelles on peut avoir lors de la construction d'une séquence  $T_j$ , avec  $j > 1$ , que  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) \neq \emptyset$ . Il faut remarquer qu'il n'y a plus que ces 10 configurations pour lesquelles la propriété  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) \neq \emptyset$  est vérifiée.

Grâce au comportement cyclique avec lequel l'algorithme de Tucker considère les chemins dans  $F$ , nous pouvons en déduire les deux propriétés suivantes :

★ **Propriété 1** *Si  $(j \bmod (\frac{l-2}{2})) \neq 0$ , alors  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) = \emptyset$  et donc,  $\text{Col}(F_j) \leq \text{Col}(F_{j-1}) + 2 \leq 2j + 1 + \lfloor \frac{2j}{l-2} \rfloor$ .*

*Preuve.* Supposons la propriété fautive, c'est-à-dire, que  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) \neq \emptyset$ .

Considérons tout d'abord le cas où  $1 < j < \frac{l-2}{2}$ . Comme  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) \neq \emptyset$ , alors lors de la construction de la séquence  $T_j$  on est dans une des seules dix configurations possibles montrées dans la figure 4.3. Dans la pire des hypothèses, les chemins  $\bigcup_{i=2}^j \{A_1^i, A_d^i\}$  ont pour graphe de conflit une chaîne de longueur au plus égale à  $2(j-1)$  (voir figure 4.4). Comme  $j$  est au plus égal à  $\frac{l-2}{2} - 1$  et comme on doit avoir forcément  $\text{cv}(A_e^j) \cap \text{cv}(A_f^j) \neq \emptyset$ , alors le graphe de conflit induit par les chemins  $(\bigcup_{i=2}^j \{A_1^i, A_d^i\}) \cup \{A_d^1, A_e^j, A_f^j\}$  est un cycle de longueur au plus égale à  $l-3$  qui couvre tout  $C_n$ , ce qui est une contradiction avec l'hypothèse que la couverture cyclique minimale de  $F$  est égale à  $l$ .

Considérons maintenant le cas où  $j > \frac{l-2}{2}$ , et supposons que lors de la construction de la séquence  $T_{j-1}$  on est dans une des seules dix configurations possibles

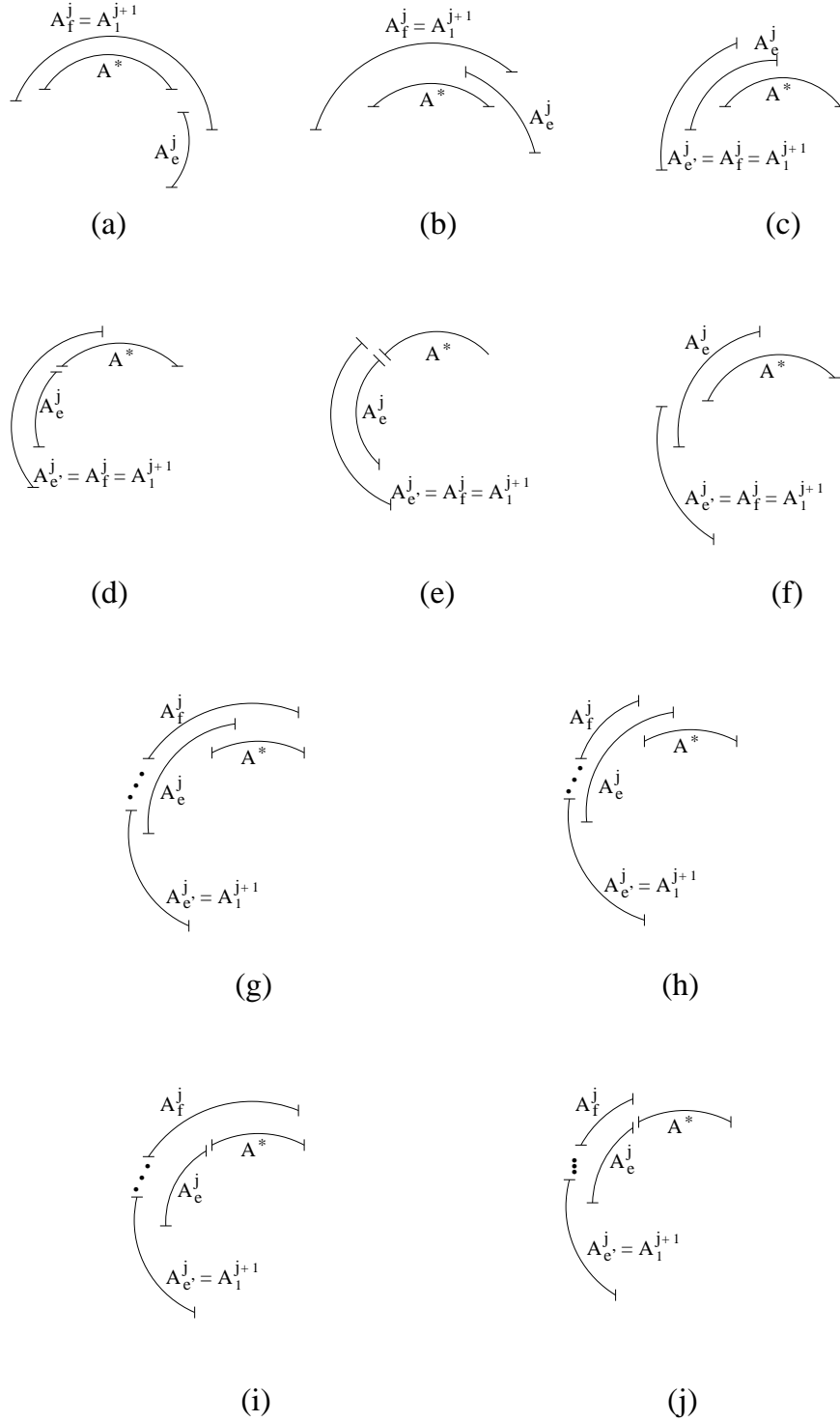


FIGURE 4.3 – Les seules dix configurations possibles pour lesquelles la propriété  $cv(A_e^j) \cap cv(A_f^j) \neq \emptyset$  est vérifiée.

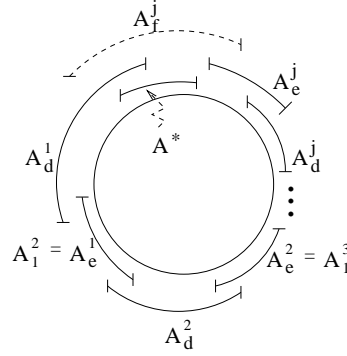


FIGURE 4.4 – Exemple de configuration possible vérifiant la propriété 1 dans le cas  $1 < j < \frac{l-2}{2}$ .

montrées dans la figure 4.3, c'est-à-dire, qu'on a  $cv(A_e^{j-1}) \cap cv(A_f^{j-1}) \neq \emptyset$ . Soit  $k$  un entier tel que  $1 \leq k < \frac{l-2}{2}$ . Supposons qu'après avoir considéré  $k$  séquences de chemins après la construction de la séquence  $T_{j-1}$  (c-à-d, juste après la construction de la séquence  $T_{j+k-1}$ ), on soit de nouveau dans une des seules dix configurations possibles montrées dans la figure 4.3, c'est-à-dire,  $cv(A_e^{j+k-1}) \cap cv(A_f^{j+k-1}) \neq \emptyset$ . Alors, il est facile de voir que dans la pire des hypothèses, les chemins  $\bigcup_{i=j}^{j+k-1} \{A_1^i, A_d^i\}$  ont pour graphe de conflit une chaîne de longueur au plus égale à  $2k$ . Comme  $k$  est au plus égal à  $\frac{l-2}{2} - 1$  et comme on doit avoir forcément  $cv(A_e^{j+k-1}) \cap cv(A_f^{j+k-1}) \neq \emptyset$ , alors on a deux possibilités :

1. Soit le premier chemin  $A_1^j$  de la séquence  $T_j$  correspond au dernier chemin  $A_f^{j-1}$  de la séquence  $T_{j-1}$  (voir figure 4.3 (a), (b), (c), (d) et (f)), et alors le graphe de conflit induit par les chemins  $(\bigcup_{i=j}^{j+k-1} \{A_1^i, A_d^i\}) \cup \{A_e^{j+k-1}, A_f^{j+k-1}\}$  est un cycle de longueur au plus égale à  $l-2$  qui couvre tout  $C_n$ , ce qui est une contradiction avec l'hypothèse que la couverture cyclique minimale de  $F$  est égale à  $l$ ;
2. Soit le premier chemin  $A_1^j$  de la séquence  $T_j$  correspond à un chemin  $A_{e'}^{j-1}$  différent au dernier chemin  $A_f^{j-1}$  de la séquence  $T_{j-1}$ , avec  $e' < f$  (voir figure 4.3 (g), (h), (i) et (j)), et alors le graphe de conflit induit par les chemins  $(\bigcup_{i=j}^{j+k-1} \{A_1^i, A_d^i\}) \cup \{A_e^{j-1}, A_e^{j+k-1}, A_f^{j+k-1}\}$  est un cycle de longueur au plus égale à  $l-1$  qui couvre tout  $C_n$ , ce qui est à nouveau une contradiction.

Ainsi, nous avons prouvé que les chemins  $A_e^j, A_{e+1}^j, \dots, A_f^j$  sont deux à deux disjoints et ils peuvent être coloriés en utilisant une nouvelle couleur. Au total, nous avons besoin d'au plus  $Col(F_{j-1}) + 2 \leq 2j + 1 + \lfloor \frac{2j}{l-2} \rfloor$  couleurs pour colorier  $F_j$ , ce qui prouve cette propriété.

★ **Propriété 2** Si  $j \equiv 0 \pmod{\frac{l-2}{2}}$ , alors dans le pire des cas on peut avoir que  $cv(A_e^j) \cap cv(A_f^j) \neq \emptyset$  et donc,  $Col(F_j) \leq Col(F_{j-1}) + 3 \leq 2j + 1 + \lfloor \frac{2j}{l-2} \rfloor$ .

*Preuve.* Comme une conséquence de la propriété 1, on peut avoir, dans le pire des cas, que pour tout entier  $j$  tel que  $j$  soit un multiple de  $\frac{l-2}{2}$ , les chemins dans la séquence  $T_j$  se trouvent dans une des seules dix configurations possibles

montrées dans la figure 4.3. Formellement, dans ce cas il peut exister un chemin  $A_e^j$  tel que  $e < e' \leq f$  et  $\text{cv}(A_e^j) \cap \text{cv}(A_{e'}^j) \neq \emptyset$ . Nous aurons donc besoin de deux nouvelles couleurs pour colorier les chemins  $A_e^j, A_{e+1}^j, \dots, A_f^j$ , ce qui implique qu'au total, nous avons besoin d'au plus  $\text{Col}(F_{j-1}) + 3$  couleurs pour colorier  $F_j$ . Par induction, nous obtenons alors que  $\text{Col}(F_j) \leq 2j + 1 + \lfloor \frac{2j}{l-2} \rfloor$  ce qui prouve cette propriété.

Par définition d'une séquence de chemins, il n'est pas difficile de vérifier qu'après la construction de chaque séquence de chemins  $T_j$ , avec  $j > 1$ , la congestion induite par la sous-collection de chemins  $F \setminus F_j$  est diminuée d'au moins deux unités par rapport à celle induite par la sous-collection de chemins  $F \setminus F_{j-1}$ , et donc  $L(F \setminus F_j) \leq L(F \setminus F_{j-1}) - 2 \leq L - 2j$ . Comme tous les cas possibles ont été considérés dans les propriétés 1 et 2, alors l'induction dans le cas (b) est prouvée.

Ainsi, on a montré que si le nombre des séquences de chemins construites en utilisant l'algorithme de Tucker est égal à  $r$ , alors le nombre de couleurs utilisées par cet algorithme pour colorier  $F$  est au plus égal à  $2r + 1 + \lfloor \frac{2r}{l-2} \rfloor$ . Comme après la construction de chaque séquence  $T_j$ , avec  $1 \leq j \leq r$ , la congestion  $L(F \setminus F_j)$  est au plus égale à  $L - 2j$ , alors si la valeur de la congestion  $L$  est paire, on a  $r = \frac{L}{2}$ . Ceci implique que le nombre de couleurs utilisées par l'algorithme de Tucker est au plus égal à  $L + 1 + \lfloor \frac{L}{l-2} \rfloor \leq \lfloor (\frac{l-1}{l-2})L \rfloor + 1$ . Si  $L$  est impair, on sait qu'après la construction de la séquence  $T_{r-1}$ , le nombre de couleurs utilisées par l'algorithme de Tucker est égal à  $\text{Col}(F_{r-1}) \leq L + \lfloor \frac{L-1}{l-2} \rfloor$ , puisque la congestion induite par la sous-collection de chemins  $F_{r-1}$  est au moins égale à  $L - 1$  qui est une valeur paire. De plus, le dernier chemin  $A_f^{r-1}$  de la séquence  $T_{r-1}$  fait partie aussi de la séquence  $T_r$ . Dans le pire des cas il existe un chemin  $A_i^r$  après le chemin  $A_f^{r-1}$  tel que  $\text{cv}(A_i^r) \cap \text{cv}(A_f^{r-1}) \neq \emptyset$ , où  $A_f^{r-1}$  est le premier des chemins lors de la séquence  $T_{r-1}$  colorié avec la couleur  $\text{Col}(F_{r-1})$ , avec  $e' \leq f$ , alors, on aura besoin d'au plus une nouvelle couleur pour colorier tous les chemins dans  $F$ . Donc,  $\text{Col}(F_r) \leq \text{Col}(F_{r-1}) + 1 \leq L + 1 + \lfloor \frac{L-1}{l-2} \rfloor \leq \lfloor (\frac{l-1}{l-2})L \rfloor + 1$ . Ceci conclue la preuve du théorème.  $\square$

On peut remarquer que dans le cas où la couverture cyclique minimale d'une collection de chemins est au moins quatre, nous obtenons le même résultat que celui de Tucker (voir théorème 19) à 1 près. Les deux corollaires suivants sont des conséquences directes du théorème 20.

**Corollaire 2** Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$  avec une congestion  $L = L(F)$ . Si la longueur des chemins dans  $F$  est au plus égale à  $n/k$ , pour tout  $k \geq 3$ , alors le nombre de couleurs nécessaires pour colorier  $F$  en utilisant l'algorithme de Tucker est au plus égal à  $\lfloor (\frac{k}{k-1})L \rfloor + 1$ .

**Corollaire 3** Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$  avec une congestion  $L = L(F)$ . Soit  $\rho$  la longueur maximale des chemins dans  $F$ , avec  $1 \leq \rho \leq n/3$ . Alors, le nombre de couleurs nécessaires pour colorier  $F$  en utilisant l'algorithme de Tucker est au plus égal à  $\lfloor (\frac{n}{n-\rho})L \rfloor + 1$ .

En utilisant les propriétés de coloration des graphes d'intervalles, Tucker avait montré dans [64] que  $2L$  couleurs suffisent pour colorier toute collection de chemins dans l'anneau avec une congestion  $L$ . Le même résultat peut être déduit facilement de l'algorithme glouton de Tucker comme suit.

**Corollaire 4** *Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$  avec une congestion  $L = L(F)$ . Dans le pire des cas, le nombre de couleurs nécessaires pour colorier  $F$  en utilisant l'algorithme de Tucker est au plus égal à  $2L$ .*

**Preuve.** Il est facile de voir que dans le pire des cas, lors de la construction de chaque séquence de chemins  $T_i$  (voir la preuve du théorème 20) nous avons besoin d'au plus quatre nouvelles couleurs pour colorier les chemins  $A_1^i, \dots, A_f^i$ , puisque on peut avoir que  $\text{cv}(A_e^i) \cap \text{cv}(A_f^i) \neq \emptyset$ . Cependant, après la construction de chaque séquence de chemins  $T_i$ , la congestion  $L(F \setminus F_i)$  induite par la sous-collection de chemins  $F \setminus F_i$  est au plus égale à  $L - 2i$ . Ceci prouve le corollaire.  $\square$

### 4.2.1 Collections particulières de chemins

Nous traitons ici quelques collections particulières de chemins ayant une couverture cyclique minimale donnée  $l$  et une congestion  $L$ , telles que le nombre minimum de couleurs nécessaires pour colorier ces chemins est égal à  $\lceil (\frac{l}{l-1}) L \rceil$ . Ceci montre que la borne supérieure obtenue dans le théorème 20 est assez bonne. Le résultat principal de cette section est le théorème suivant.

**Théorème 21** *Pour tout entier  $l$  impair, avec  $l \geq 3$ , et pour tout entier  $L$  pair, avec  $L \geq 2$ , il existe une collection de chemins  $F$  sur l'anneau  $C_l$  avec une congestion  $L$  qui vérifie :  $\chi(F) = \lceil (\frac{l}{l-1}) L \rceil$ .*

Pour prouver le théorème 21 nous allons introduire une nouvelle notion (voir [60]). Une  $r$ -uplet coloration d'un graphe non orienté  $G$  est une assignation de  $r$  couleurs différentes à chaque sommet dans  $G$  de telle façon que si deux sommets sont adjacents dans  $G$ , alors l'intersection de leurs  $r$ -uplets de couleurs doit être vide. On note par  $\chi_r(G)$  le nombre minimum de couleurs nécessaires pour une  $r$ -uplet coloration du graphe  $G$ . Dans le cas où le graphe  $G$  est un cycle, Stahl a montré dans [60] le résultat suivant.

**Théorème 22** (Stahl [60]) *Soient  $C_n$  le cycle non orienté à  $n$  sommets et  $r$  un entier positif. Alors,*

$$\chi_r(C_n) = \begin{cases} 2r, & \text{si } n \text{ est pair} \\ 2r + 1 + \lfloor \frac{2(r-1)}{n-1} \rfloor, & \text{si } n \text{ est impair} \end{cases}$$

## Preuve du théorème 21

La preuve du théorème 21 est obtenue par l'application du théorème 22. En effet, soit  $C_l$  l'anneau non orienté à  $l$  sommets et soit  $r = L/2$ , où  $L$  est la valeur de la congestion donnée. La collection de chemins  $F$  sur  $C_l$  avec une congestion égale à  $L$  est construit comme suit. Pour tout  $i = 0, 1, \dots, l-1$ , on ajoute  $r$  copies du chemin  $\langle i, i+2 \pmod{l} \rangle$  à  $F$ . Par la construction précédente, il est facile de vérifier que si  $L = 2$  (et donc  $r = 1$ ), alors le graphe d'intersection des chemins dans  $F$  est isomorphe à  $C_l$ . Ainsi, par le théorème 22, pour tout entier  $L = 2r$ , avec  $r \geq 1$ , on a que  $\chi(F) = \chi_r(C_l) = 2r + 1 + \lfloor \frac{2r-2}{l-1} \rfloor = L + 1 + \lfloor \frac{L-2}{l-1} \rfloor = \lceil (\frac{l}{l-1}) L \rceil$ .  $\square$

Il est évident que la couverture cyclique minimale des collections de chemins sur l'anneau  $C_l$  à  $l$  sommets, avec  $l \geq 3$ , construites dans la preuve du théorème 21 est exactement égale à  $l$ . Ainsi, par les théorèmes 20 et 21, pour tout entier  $l$  impair, avec  $l \geq 5$ , il existe des collections de chemins sur l'anneau avec une couverture cyclique minimale égale à  $l$  pour lesquelles, le nombre de couleurs utilisées par l'algorithme de Tucker pour les colorier est très proche de la valeur optimale.

## 4.3 Instances polynomiales

Dans cette section, nous présentons quelques instances du problème "ARC-COLORING" qui peuvent être résolues optimalement en temps polynomial.

Tout d'abord, il est facile de vérifier que si la longueur de chaque chemin dans  $F$  est strictement supérieure à la moitié du nombre de sommets dans l'anneau, alors toute paire de chemins  $A_i$  et  $A_j$  dans  $F$  vérifie  $cv(A_i) \cap cv(A_j) \neq \emptyset$ , et donc le graphe d'intersection induit par les chemins dans  $F$  est isomorphe au graphe complet à  $m = |F|$  sommets. Ceci implique le résultat suivant.

**Proposition 1** *Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins non orientés sur  $C_n$ . Si la longueur de chaque chemin dans  $F$  est strictement supérieure à  $n/2$ , alors on peut colorier optimalement les chemins dans  $F$  en temps linéaire.*

Avant de montrer d'autres instances du problème ARC-COLORING qui peuvent être calculées en temps polynomial, nous donnerons quelques définitions et résultats préliminaires.

**Définition 2** *Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$ . On définit la **base** de  $F$  qu'on note par  $B(F)$ , comme l'ensemble maximal de chemins dans  $F$  telle que toute paire de chemins  $A_i$  et  $A_j$  dans  $B(F)$  vérifie  $cv(A_i) \neq cv(A_j)$ . De plus, pour tout chemin  $A_i \in B(F)$ , on définit le **poinds** du chemin  $A_i$  qu'on note par  $W(A_i)$ , comme  $W(A_i) = |\{A_j : A_j \in F \setminus B(F) \text{ et } cv(A_j) = cv(A_i)\}|$ , et on note par  $l(A_i) = |cv(A_i)|$  la longueur de  $A_i$ .*

Dans la figure 4.5 nous présentons un exemple d'une collection de chemins sur l'anneau  $C_5$  (figure 4.5(a)), et la base de cette collection (figure 4.5(b)). Soit  $F$  une

collection de chemins sur l'anneau  $C_n$  non orienté à  $n$  sommets. Deux chemins  $A_i$  et  $A_j$  de  $F$  sont dits *propres* si et seulement si  $\text{cv}(A_i) \not\subseteq \text{cv}(A_j)$  et  $\text{cv}(A_j) \not\subseteq \text{cv}(A_i)$ . Ainsi,  $F$  est dite *propre* si toute paire de chemins de  $F$  est propre. L'ensemble de chemins montrés dans la figure 4.5(b) est un exemple d'une collection propre de chemins.

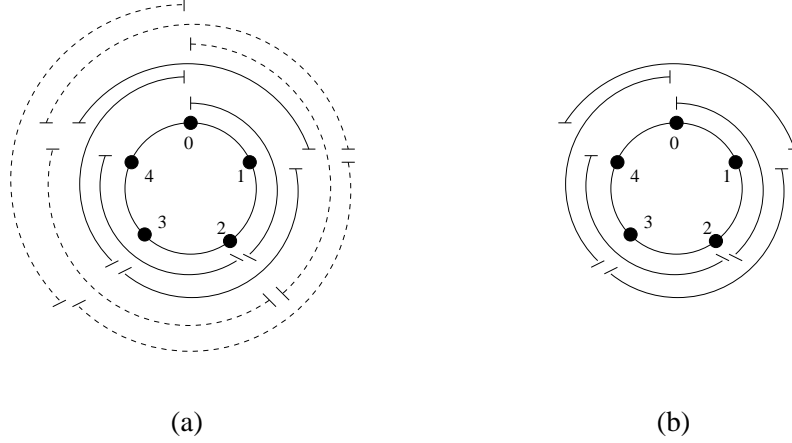


FIGURE 4.5 – (a) Une collection  $F$  de chemins sur  $C_5$ . (b) La base  $B(F)$  de  $F$ , où le poids  $W(A_i)$  de chaque chemin  $A_i \in B(F)$  est égal à 1.

**Théorème 23** (Shih et Hsu [59]) Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de  $m$  chemins sur  $C_n$ . Si  $F$  est propre, alors une coloration optimale des chemins dans  $F$  peut être effectuée en temps  $O(m^{1.5})$ .

L'algorithme proposé par Shih et Hsu dans [59] pour colorier optimalement les sommets des graphes arc-circulaires propres (graphes de conflit induits par collections propres de chemins) est le meilleur algorithme connu à ce jour. De plus, Deng, Hell et Hung donnent dans [21] le meilleur algorithme connu à ce jour pour reconnaître si un graphe  $G$  non orienté donné avec  $m$  sommets et  $e$  arêtes, est un graphe arc-circulaire propre. L'algorithme de Deng, Hell et Hung a une complexité en temps de  $O(e + m)$  et la même complexité suffit pour construire une représentation de  $G$  en une collection propre de chemins sur un anneau.

Dans la suite, nous allons montrer qu'on peut transformer de façon simple quelques collection de chemins qui ne sont pas à priori propres, en ensembles de chemins propres sans utiliser l'algorithme de Deng, Hell et Huang [21]. Nous dirons que deux collections de chemins sont *équivalentes* si leurs graphes de conflit induits sont isomorphes. Les principaux résultats que nous trouvons dans cette section sont les suivants.

**Théorème 24** Soient  $F$  une collection de chemins sur l'anneau  $C_n$  et  $B(F)$  la base de  $F$ . Si  $B(F)$  est propre, alors  $F$  peut être transformée en temps linéaire en une collection équivalente  $F^*$  de chemins propres.



En utilisant le théorème 24 et le théorème 23, nous montrerons le résultat suivant.

**Théorème 25** *Soit  $F$  une collection de  $m$  chemins sur l'anneau  $C_n$  telle que la longueur de chacun des chemins dans  $F$  soit dans l'ensemble  $\{1, \alpha, \alpha + 1\}$ , pour tout entier  $\alpha \geq 2$ . Alors, une coloration optimale des chemins dans  $F$  peut être obtenue en temps  $O(m^{1.5})$ .*

## Preuve du théorème 24

Nous supposons que  $F \neq B(F)$ . Sinon, comme par hypothèse  $B(F)$  est propre, alors  $F$  est aussi propre, ce qui conclue la preuve. Nous allons construire une collection  $F^*$  de chemins propres de la façon suivante. Initialement,  $F^*$  est vide. Pour chaque chemin  $A_i = \langle a_i, b_i \rangle$  dans  $B(F)$  de poids  $W(A_i) = p_i$  et de longueur courante  $l(A_i) = l_i$  (voir définition 2), on ajoute  $2p_i$  nouveaux sommets dans l'anneau  $C_n$  comme suit (voir exemple dans la figure 4.6). On ajoute  $p_i$  nouveaux sommets juste après le sommet  $a_i$  dans le sens des aiguilles d'une montre et on les note par  $a_i^1, a_i^2, \dots, a_i^{p_i}$ , et on ajoute  $p_i$  nouveaux sommets juste avant le sommet  $b_i$  dans le sens contraire à celui des aiguilles d'une montre, et on les note par  $b_i^1, b_i^2, \dots, b_i^{p_i}$ . Ensuite, on ajoute à  $F^*$  les chemins  $\langle a_i, b_i^1 \rangle, \langle a_i^1, b_i^2 \rangle, \langle a_i^2, b_i^3 \rangle, \dots, \langle a_i^{p_i-1}, b_i^{p_i} \rangle, \langle a_i^{p_i}, b_i \rangle$ . Dans la figure 4.6 nous montrons un exemple de cette transformation.

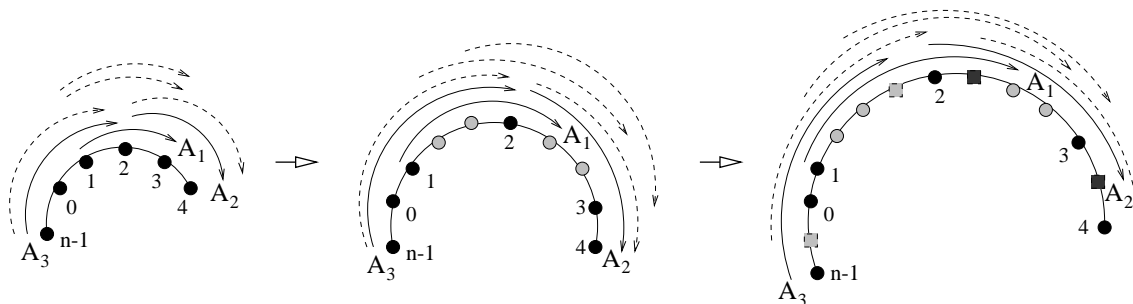


FIGURE 4.6 – Exemple de transformation des chemins  $A_1, A_2$  et  $A_3$  en gras qui sont dans  $B(F)$ , où  $W(A_1) = 2, W(A_2) = 1$  et  $W(A_3) = 1$ .

Il faut remarquer que les chemins associés au chemin  $A_i \in B(F)$  qui sont ajoutés à  $F^*$  sont deux à deux propres. De plus, soit  $A_j = \langle a_j, b_j \rangle$  un chemin dans  $B(F)$ , avec  $j \neq i$ , tel que  $\text{cv}(A_j) \cap \text{cv}(A_i) \neq \emptyset$ . Par hypothèse,  $A_i$  et  $A_j$  sont deux chemins propres. Nous dirons que  $a_j < a_i$  et  $b_j < b_i$  si le sommet initial  $a_j$  (resp. final  $b_j$ ) du chemin  $A_j$  se trouve à gauche (dans le sens des aiguilles d'une montre) du sommet initial  $a_i$  (resp. final  $b_i$ ) du chemin  $A_i$ .

Nous supposons sans perte de généralité que  $a_j < a_i$  et  $b_j < b_i$ . Il est facile de voir que les  $p_i + 1$  chemins dans  $F^*$  associés au chemin  $A_i \in B(F)$  ont pour sommet initial un sommet  $u \geq a_i$ , et ils ont pour sommet final, un sommet  $v$  qui vérifie  $a_i < a_i^1 < a_i^2 < \dots < a_i^{p_i} < a_i + 1 < a_i + 2 < \dots < a_i + l_i - 1 < v \leq b_i$ .

D'autre part,  $b_j \in \{a_i + 1, a_i + 2, \dots, a_i + l_i - 1\}$ , ce qui implique que la transformation précédente préserve la propriété d'intersection propre des chemins dans

$B(F)$  et produit une collection propre  $F^*$  de chemins qui est équivalente à la collection  $F$  de départ et qui peut être calculée en temps  $O(|F|)$ .  $\square$

Une conséquence directe du théorème précédent et du théorème 23 est le corollaire suivant.

**Corollaire 5** *Soit  $F$  une collection de  $m$  chemins sur l'anneau  $C_n$ . Si tous les chemins dans  $F$  ont la même longueur, alors une coloration optimale des chemins dans  $F$  peut être obtenue en temps  $O(m^{1.5})$ .*

## Preuve du théorème 25

Nous supposons qu'il y a au moins deux chemins  $A_i$  et  $A_j$  dans  $F$  avec une longueur égale à  $\alpha$  et  $\alpha + 1$  respectivement, et qu'il n'y a pas de chemins de longueur unitaire dans  $F$ . Dans le cas contraire, il suffit de colorier les chemins dans  $F$  de même longueur comme dans le cas du corollaire 5, et ensuite on peut colorier chacun des chemins de longueur unitaire qui restent avec une couleur différente de celles déjà utilisées par les chemins coloriés qui le contiennent, ce qui donne une coloration optimale de  $F$ .

Soit  $B(F)$  la base de  $F$  et soient  $B_\alpha$  et  $B_{\alpha+1}$  deux sous-ensembles disjoints de  $B(F)$  tels que  $B_\alpha = \{A_i \in B(F) : l(A_i) = \alpha\}$  et  $B_{\alpha+1} = B(F) \setminus B_\alpha$ . Nous supposons que  $B(F)$  n'est pas propre. Sinon, la preuve de ce théorème suit par l'application du théorème 24 et du théorème 23. Ainsi, tout d'abord, nous allons construire à partir de  $B(F)$  un nouveau ensemble  $F'$  de chemins propres. Pour cela, nous transformons les chemins de chacun des sous-ensembles  $B_\alpha$  et  $B_{\alpha+1}$  de la façon suivante. Pour tout chemin  $A_i = \langle a_i, b_i \rangle$  dans  $B_\alpha$ , nous considérons trois cas (voir figure 4.7) :

- *Cas (a) :*  $cv(A_i) \cap cv(A_j) \neq \emptyset$ , où  $A_j = \langle a_j, b_j \rangle$  est un chemin dans  $B_{\alpha+1}$  tel que  $a_i = a_j$ . Dans ce cas, on ajoute un nouveau sommet, qu'on note par  $p_j$ , dans l'anneau juste avant (à gauche) le sommet  $a_j$  dans le sens des aiguilles d'une montre. Ensuite, on transforme le chemin  $A_i$  en le chemin  $A'_i = \langle p_j, b_i \rangle$ , et on transforme tous les chemins  $A_k = \langle a_k, b_k \rangle$  dans  $B_\alpha \cup B_{\alpha+1}$  tels que  $b_k = a_i$  en les chemins  $A'_k = \langle a_k, p_j \rangle$ . On laisse les sommets initiaux et finaux des chemins restants dans  $B_\alpha \cup B_{\alpha+1}$  inchangés (voir figure 4.7(a)).
- *Cas (b) :*  $cv(A_i) \cap cv(A_j) \neq \emptyset$ , où  $A_j = \langle a_j, b_j \rangle$  est un chemin dans  $B_{\alpha+1}$  tel que  $b_i = b_j$ . Dans ce cas, on ajoute un nouveau sommet, qu'on note par  $q_j$ , dans l'anneau juste après (à droite) le sommet  $b_j$  dans le sens des aiguilles d'une montre. Ensuite, on transforme le chemin  $A_i$  en le chemin  $A'_i = \langle a_i, q_j \rangle$ , et on transforme tous les chemins  $A_k = \langle a_k, b_k \rangle$  dans  $B_\alpha \cup B_{\alpha+1}$  tels que  $a_k = b_i$  en les chemins  $A'_k = \langle q_j, b_k \rangle$ . On laisse les sommets initiaux et finaux des chemins restants dans  $B_\alpha \cup B_{\alpha+1}$  inchangés (voir figure 4.7(b)).
- *Cas (c) :*  $cv(A_i) \cap cv(A_j) \neq \emptyset$  et  $cv(A_i) \cap cv(A_s) \neq \emptyset$ , où les chemins  $A_j = \langle a_j, b_j \rangle$  et  $A_s = \langle a_s, b_s \rangle$  sont dans  $B_{\alpha+1}$  et sont tels que  $a_i = a_j$  et  $b_i = b_s$ . Dans ce cas, on applique de façon indépendante les cas (a) et (b) (voir figure 4.7(c)).

Après la transformation précédente, on pose  $F' = B_\alpha \cup B_{\alpha+1}$ . Pour tout chemin  $A_i \in B(F)$ , soit  $A'_i \in F'$  le chemin associé au chemin  $A_i$  après la transformation

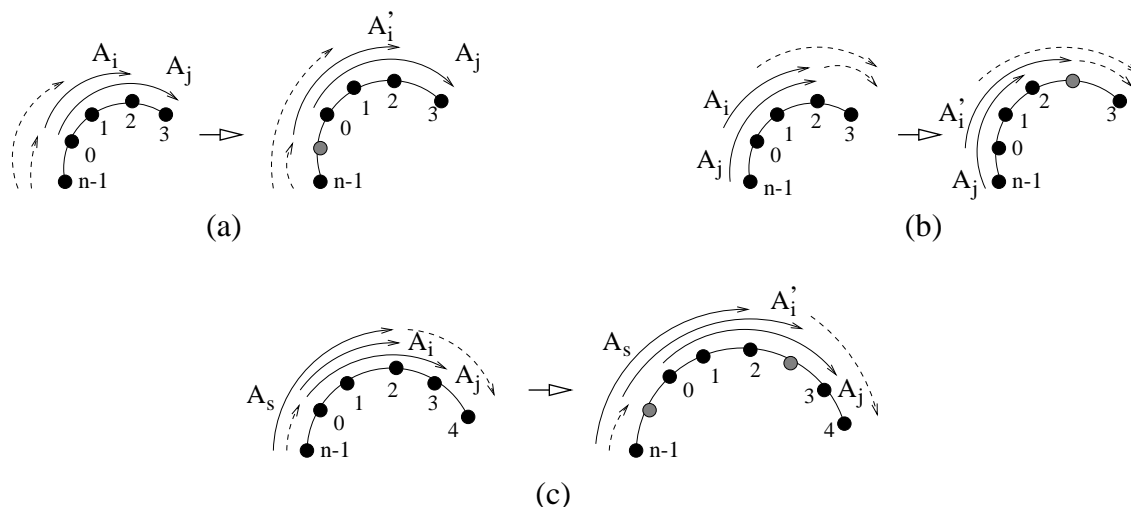


FIGURE 4.7 – Exemple du cas (a) (resp. (b) et (c)) dans la preuve du théorème 25.

précédente. Alors, il est clair que l'ensemble de chemins  $F'$  vérifie les deux propriétés suivantes :

**Propriété 3**  $F'$  est propre.

**Propriété 4** Pour toute paire de chemins  $A_i$  et  $A_j$  dans  $B(F)$ ,  $cv(A_i) \cap cv(A_j) \neq \emptyset$  si et seulement si  $cv(A'_i) \cap cv(A'_j) \neq \emptyset$ .

En effet, chaque chemin dans  $B_\alpha$  peut être contenu en au plus deux chemins différents dans  $B_{\alpha+1}$ , puisque la différence de longueur entre les chemins dans  $B_\alpha$  et ceux dans  $B_{\alpha+1}$  est d'une unité. Ainsi, les propriétés 3 et 4 sont une conséquence directe de la transformation précédente. Finalement, nous construisons une nouvelle collection de chemins  $F^*$  sur l'anneau augmenté résultant de la transformation précédente, qui contiendra pour chaque chemin  $A_i \in B(F)$ ,  $W(A_i) + 1$  copies de son chemin associé  $A'_i \in F'$ . Par une simple observation, on peut constater que la collection de chemins  $F^*$  est équivalente à la collection de chemins  $F$ , et donc si on pose  $B(F^*) = F'$ , le théorème suit par l'application directe du théorème 24 et du théorème 23.  $\square$

Il faut remarquer que les collections de chemins vérifiant les hypothèse du théorème 25 sont les uniques collections telles que leur base (étant non propre) peut être transformée en une base propre. En effet, si la différence des longueurs des chemins (non unitaires) dans la base d'une collection quelconque de chemins est supérieure ou égale à 2, alors il est clair qu'on ne peut pas toujours obtenir, à partir d'une telle base, une base propre.

## 4.4 Problèmes ouverts

Nous n'avons pas pu obtenir des instances pour lesquelles la borne supérieure du théorème 20 soit atteinte. Nous pensons que cette borne peut être améliorée.

En nous appuyant sur le résultat obtenu dans le théorème 21, nous posons donc la conjecture suivante :

**Conjecture 1** Soient  $C_n$  l'anneau non orienté à  $n$  sommets et  $F$  une collection de chemins sur  $C_n$  avec une congestion  $L = L(F)$ . Soit  $l$  la couverture cyclique minimale de  $F$ . Si  $l \geq 4$ , alors le nombre de couleurs nécessaires pour colorier  $F$  en utilisant l'algorithme de Tucker est au plus égal à  $\lceil (\frac{l}{l-1}) L \rceil$ .

Soient  $F$  une collection de chemins sur  $C_n$  et  $\rho$  la longueur maximale des chemins dans  $F$ . Une conséquence directe du théorème 25 est que si  $\rho \leq 3$ , alors on peut colorier les chemins dans  $F$  de façon optimale en temps polynomial. Cependant, le problème suivant reste ouvert :

**Problème 1** Soient  $F$  une collection de chemins sur  $C_n$ ,  $L(F)$  la valeur de la congestion induite par  $F$  et  $\rho$  la longueur maximale des chemins dans  $F$ . Supposons que  $L(F)$  est non bornée et  $\rho$  est bornée par une constante supérieure ou égale à 4. Quelle est la complexité algorithmique associée au problème de trouver une coloration minimale pour les chemins dans  $F$  ?

Il est clair que la congestion  $L(F)$  induite par  $F$  dans le problème 1 doit être non bornée, puisque en cas contraire, ce problème peut être résolu en temps polynomial comme une conséquence directe du corollaire 1.

## 4.5 Conclusion

Nous résumons les résultats trouvés dans ce chapitre dans la table 4.1 suivante.

Notation :  $F$  une collection quelconque de chemins dans l'anneau non orienté  $C_n$  à  $n$  sommets,  $L$  la congestion induite par  $F$ ,  $l$  la couverture cyclique de  $F$ ,  $m$  le nombre de chemins dans  $F$ ,  $\Lambda$  la longueur des chemins dans  $F$  et  $\chi$  le nombre minimal de couleurs nécessaires pour colorier proprement  $F$ .

$l$	$\Lambda$	$\chi$	complex.	référence
$\geq 4$	quelconque	$\leq \lceil (\frac{l-1}{l-2}) L \rceil + 1$	?	théorème 20
$\geq 4$	$\leq n/k, \forall k \geq 3$	$\leq \lceil (\frac{k}{k-1}) L \rceil + 1$	?	corollaire 2
$\leq 3$	$> n/2, \forall A_i \in F$	$m$	$O(m)$	proposition 1
quelconque	$\{1, \alpha, \alpha + 1\},$ $\forall \alpha \geq 2$ et $\forall A_i \in F$	?	$O(m^{1.5})$	théorème 25

TABLE 4.1 – Résultats obtenus pour le problème de coloration de chemins dans l'anneau.

# 5 Coloration de chemins dans les arbres

Dans ce chapitre nous allons étudier le problème de routage par chemins arc-disjoints dans les arbres orientés symétriques. Comme nous l'avons vu dans le chapitre 3, il existe beaucoup de résultats concernant ce problème. Cependant, comme Beauquier et al. l'ont remarqué dans [7], le problème de coloration d'un ensemble de chemins représentant une permutation des sommets dans un arbre n'a pas été étudié jusqu'à présent. Ainsi, nous concentrons notre étude sur ce problème dans le cas où les chemins représentent une permutation des sommets dans les arbres. A priori, nous pouvons penser que le problème de router des permutations est plus simple algorithmiquement parlant, que le problème général dans les arbres. Cependant, nous montrons dans la section 5.1 qu'en fait ces deux problèmes sont équivalents. De plus, dans la section 5.2 nous montrons que même dans le cas d'un arbre binaire et pour certains types particuliers de permutations, ce problème reste NP-difficile. Dans la section 5.3, nous étudions le comportement moyen du nombre de couleurs nécessaires pour colorier toute permutation dans les arbres. Nous obtenons des résultats exacts dans le cas où l'arbre est une chaîne ou une étoile généralisée, et nous donnons des bornes inférieure et supérieure pour les arbres quelconques. Dans la section 5.4 nous présentons quelques problèmes qui restent ouverts, et nous concluons ce chapitre dans la section 5.5. La plupart des résultats montrés dans ce chapitre ont été obtenus dans [3, 18].

## 5.1 Équivalence avec le routage de permutations

Dans cette section nous montrerons que dans les arbres, le problème de coloration d'une collection quelconque de chemins est équivalent au problème de coloration d'un ensemble de chemins représentant une permutation des sommets d'un arbre.

**Théorème 26** *Soient  $T$  un arbre orienté symétrique à  $n$  sommets de degré maximum  $\Delta$ , et  $\mathcal{P}$  une collection quelconque de chemins sur  $T$  avec une congestion  $L = L_T(\mathcal{P})$ . On peut construire en temps polynomial, un arbre  $T'$  orienté symétrique à  $O(n\Delta L)$  sommets de degré maximum  $\Delta' \leq 2\Delta - 1$ , et un ensemble de chemins  $\mathcal{P}'$  représentant une permutation des sommets dans  $T'$ , tels que  $R_T(\mathcal{P}) = k$  si et seulement si  $R_{T'}(\mathcal{P}') = k$ , pour un entier positif  $k$ .*

**Preuve.** Nous supposons sans perte de généralité que tous les arcs dans  $T$  sont traversés par exactement  $L$  chemins dans  $\mathcal{P}$ , où  $L$  est la valeur de la congestion induite par  $\mathcal{P}$ . Si ce n'est pas le cas, pour tout arc  $(u, v)$  dans  $T$  traversé par  $r < L$  chemins, nous additionons  $L - r$  chemins unitaires de la forme  $u \rightsquigarrow v$  à  $\mathcal{P}$ . Il est clair que les chemins de longueur unitaire ajoutés à  $\mathcal{P}$  n'affectent pas la coloration des chemins initiaux d'une telle collection.

Pour tout sommet  $v$  dans  $T$ , nous notons  $N(v) = \{v_1, v_2, \dots, v_j\}$  l'ensemble des voisins du sommet  $v$ . De plus, pour tout sommet  $v$  et pour tout sommet  $v_i \in N(v)$ , nous notons  $In(v, v_i)$  le sous-ensemble de chemins dans  $\mathcal{P}$  ayant comme destination finale le sommet  $v$  et traversant l'arc  $(v_i, v)$  de  $T$ . De la même façon, nous notons  $Out(v, v_i)$  le sous-ensemble de chemins dans  $\mathcal{P}$  ayant le sommet  $v$  comme origine et traversant l'arc  $(v, v_i)$  de  $T$ .

Finalement, pour tout sommet  $v$  de  $T$ , nous notons  $D_{In}(v) = \sum_{v_i \in N(v)} |In(v, v_i)|$  (resp.  $D_{Out}(v) = \sum_{v_i \in N(v)} |Out(v, v_i)|$ ), le nombre de chemins dans  $\mathcal{P}$  ayant le sommet  $v$  comme sommet final (resp. comme sommet initial), et pour tout sommet  $v_i \in N(v)$ , nous notons  $Min(v, v_i) = \min\{|In(v, v_i)|, |Out(v, v_i)|\}$ .

Ainsi, par nos hypothèses de départ, toute collection de chemins  $\mathcal{P}$  vérifie la propriété suivante.

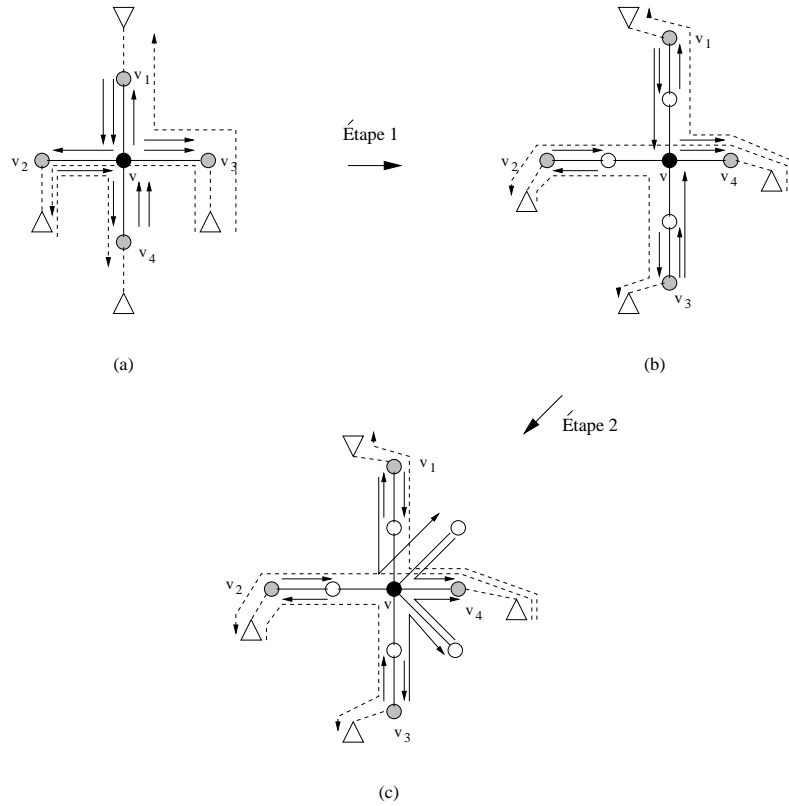
**Propriété 5** *Pour tout sommet  $v$  dans  $T$ , on a  $D_{In}(v) = D_{Out}(v)$ .*

La construction de l'arbre  $T'$  et de l'ensemble de chemins  $\mathcal{P}'$  sur  $T'$  représentant une permutation des sommets de  $T'$  est faite comme suit (voir figure 5.1 pour un exemple de cette construction). Initialement,  $T' = T$  et  $\mathcal{P}' = \mathcal{P}$ .

Pour chaque sommet  $v$  dans  $T$ , on exécute les deux étapes suivantes :

- **Étape 1** : pour chaque sommet  $v_i \in N(v)$ , posons  $\alpha_i = Min(v, v_i)$ . Si  $\alpha_i > 0$ , alors nous construisons une nouvelle chaîne orientée symétrique  $P_{\alpha_i}$  à  $\alpha_i$  sommets que nous notons  $w_1, w_2, \dots, w_{\alpha_i}$ . Ensuite, nous éliminons les arcs  $(v_i, v)$  et  $(v, v_i)$  dans  $T'$ , et nous connectons la nouvelle chaîne  $P_{\alpha_i}$  aux sommets  $v_i$  et  $v$ , c'est-à-dire, nous ajoutons à  $T'$  les arcs  $(v_i, w_1)$ ,  $(w_1, v_i)$ ,  $(v, w_{\alpha_i})$  et  $(w_{\alpha_i}, v)$ . Soient  $\beta_O(v, v_i)$  et  $\beta_I(v, v_i)$  deux sous-ensembles de chemins de cardinalité  $\alpha_i$  dans  $Out(v, v_i)$  et  $In(v, v_i)$  respectivement. Nous remplaçons chacun des chemins  $p_j = v \rightsquigarrow a_j$  dans  $\beta_O(v, v_i)$  par le chemin  $w_j \rightsquigarrow a_j$ ,  $1 \leq j \leq \alpha_i$ , et nous remplaçons chacun des chemins  $q_m = b_m \rightsquigarrow v$  dans  $\beta_I(v, v_i)$  par le chemin  $b_m \rightsquigarrow w_m$ ,  $1 \leq m \leq \alpha_i$ . Finalement, nous posons  $In(v, v_i) = In(v, v_i) \setminus \beta_I(v, v_i)$  et  $Out(v, v_i) = Out(v, v_i) \setminus \beta_O(v, v_i)$ . Dans la figure 5.1(b), nous présentons un exemple de cette construction. Il faut remarquer que après cette étape, la propriété 5 est préservée.

- **Étape 2** : nous allons supposer qu'il existe au moins deux sous-ensembles de chemins  $In(v, v_i)$  et  $Out(v, v_j)$  non vides après l'étape 1. Sinon, nous aurons fini la construction pour le sommet  $v$ . Nous notons  $I^*$  (resp.  $O^*$ ) la séquence de couples  $((|In(v, v_{i_1})|, v_{i_1}), (|In(v, v_{i_2})|, v_{i_2}), \dots, (|In(v, v_{i_r})|, v_{i_r}))$  (resp.  $((|Out(v, v_{j_1})|, v_{j_1}), (|Out(v, v_{j_2})|, v_{j_2}), \dots, (|Out(v, v_{j_s})|, v_{j_s}))$ ), avec  $1 \leq r \leq |N(v)|$  (resp.  $1 \leq s \leq |N(v)|$ ), ordonnées de façon décroissante selon leur premier élément, où tout couple

FIGURE 5.1 – Construction pour le sommet  $v$  dans l'arbre  $T$ .

$(|\text{In}(v, v_{i_k})|, v_{i_k}) \in I^*$  (resp.  $(|\text{Out}(v, v_{j_k})|, v_{j_k}) \in O^*$ ) vérifie  $|\text{In}(v, v_{i_k})| > 0$  et  $v_{i_k} \in N(v)$  (resp.  $|\text{Out}(v, v_{j_k})| > 0$  et  $v_{j_k} \in N(v)$ ). Nous notons par  $<_l^1$  l'ordre lexicographique entre séquences de couples, défini sur le premier élément des couples. Ainsi, nous considérons les deux cas suivants :

\* *Cas a* :  $I^* <_l^1 O^*$ . Pour chaque couple  $(q_{j_l}, v_{j_l}) \in O^*$ , nous construisons un ensemble que nous notons  $\tilde{O}_{j_l}$  en utilisant la procédure suivante :

**Procédure 1 :**

1. Initialement on a  $\tilde{O}_{j_l} = \emptyset$  ;
2. **Tant que**  $q_{j_l} > 0$  **faire** :
3. Soit  $(p_{i_k}, v_{i_k})$  le premier élément de  $I^*$  ;
4. **Si**  $q_{j_l} \geq p_{i_k}$  **alors faire** :
5.  $\tilde{O}_{j_l} \leftarrow \tilde{O}_{j_l} \cup \{(p_{i_k}, v_{i_k})\}$  ;
6.  $q_{j_l} \leftarrow q_{j_l} - p_{i_k}$  ;
7.  $I^* \leftarrow I^* \setminus \{(p_{i_k}, v_{i_k})\}$  ;
8. **Sinon, faire** :
9.  $\tilde{O}_{j_l} \leftarrow \tilde{O}_{j_l} \cup \{(q_{j_l}, v_{i_k})\}$  ;
10.  $q_{j_l} \leftarrow 0$  ;
11.  $(p_{i_k}, v_{i_k}) \leftarrow (p_{i_k} - q_{j_l}, v_{i_k})$  ;

L'idée de la procédure 1 est la suivante : supposons qu'on considère un couple  $(q_{j_l}, v_{j_l}) \in O^*$  et soit  $(p_{i_k}, v_{i_k})$  le premier couple courant dans  $I^*$ . Par l'étape 1 de la construction, il est clair que les sommets  $v_{j_l}$  et  $v_{i_k}$ , voisins du sommet courant  $v$ , sont différents. Imaginons par exemple que  $q_{j_l} \geq p_{i_k}$  (voir étapes 4-7 de la procédure 1). Dans ce cas, on se servira du couple  $(p_{i_k}, v_{i_k})$  additionné à l'ensemble  $\tilde{O}_{j_l}$  pour créer une nouvelle chaîne à  $p_{i_k}$  sommets, dont une des extrémités sera connectée au sommet  $v$ , et pour remplacer ainsi le sommet final (resp. initial) de  $p_{i_k}$  chemins dans  $\text{In}(v, v_{i_k})$  (resp. dans  $\text{Out}(v, v_{j_l})$ ) (c-à-d, le sommet  $v$ ) par un des nouveaux sommets de cette chaîne. Sinon (voir étapes 8-11 de la procédure 1), on pourra remplacer uniquement  $p_{i_k} - q_{j_l}$  de ces chemins.

Dans la figure 5.1(b) nous avons un exemple du cas (a). En effet, il est facile de voir dans la figure 5.1(b) que  $I^* = ((1, v_1), (1, v_3))$  et  $O^* = ((2, v_4))$ , et donc  $I^* \leq_l^1 O^*$ . Ainsi, par l'application de la procédure 1, on a que l'ensemble  $\tilde{O}_4$  associé au sommet  $v_4 \in N(v)$  est  $\tilde{O}_4 = \{(1, v_1), (1, v_3)\}$ . Suite à la procédure 1, pour chacun des ensembles  $\tilde{O}_{j_l}$  et pour chaque couple  $(\alpha, v_{i_k}) \in \tilde{O}_{j_l}$ , nous construisons une nouvelle chaîne  $P_\alpha$  orientée symétrique à  $\alpha$  sommets que nous notons  $y_1, y_2, \dots, y_\alpha$ . Ensuite, nous ajoutons à  $T'$  les arcs  $(v, y_1)$  et  $(y_1, v)$  (voir figure 5.1(c)). Soient  $A$  et  $B$  deux sous-ensembles quelconques de chemins de cardinalité  $\alpha$  dans  $\text{Out}(v, v_{j_l})$  et  $\text{In}(v, v_{i_k})$  respectivement. On remplace chaque chemin  $v \rightsquigarrow a_i \in A$  par le chemin  $y_i \rightsquigarrow a_i$ ,  $1 \leq i \leq \alpha$ , et on remplace chaque chemin  $b_m \rightsquigarrow v \in B$  par le chemin  $b_m \rightsquigarrow y_m$ ,  $1 \leq m \leq \alpha$ . Finalement, nous posons  $\text{Out}(v, v_{j_l}) = \text{Out}(v, v_{j_l}) \setminus A$  et  $\text{In}(v, v_{i_k}) = \text{In}(v, v_{i_k}) \setminus B$ . Dans la figure 5.1(c), nous présentons un exemple de cette construction.

\* *Cas b* :  $O^* \leq_l^1 I^*$ . On utilise une procédure similaire à la procédure 1 du cas (a) afin de construire les ensembles  $\tilde{I}_{i_k}$  pour chaque couple  $(p_{i_k}, v_{i_k})$  dans  $I^*$ . On construit ensuite, pour chacun des ensembles  $\tilde{I}_{i_k}$  et pour chaque couple  $(\beta, v_{j_l}) \in \tilde{I}_{i_k}$ , un nouvelle chaîne  $P_\beta$  orientée symétrique à  $\beta$  sommets qu'on note  $z_1, z_2, \dots, z_\beta$ . Ensuite, on connecte les sommets  $v$  et  $z_1$  par les arcs  $(v, z_1)$  et  $(z_1, v)$ . Nous prenons deux sous-ensembles de chemins  $A$  et  $B$  quelconques de cardinalité  $\beta$  dans  $\text{In}(v, v_{i_k})$  et  $\text{Out}(v, v_{j_l})$  respectivement, et nous remplaçons chaque chemin  $c_i \rightsquigarrow v \in A$  (resp.  $v \rightsquigarrow d_m \in B$ ) par le chemin  $c_i \rightsquigarrow z_i$  (resp.  $z_m \rightsquigarrow d_m$ ), avec  $1 \leq i \leq \beta$  (resp.  $1 \leq m \leq \beta$ ). Finalement, nous posons  $\text{Out}(v, v_{j_l}) = \text{Out}(v, v_{j_l}) \setminus B$  et  $\text{In}(v, v_{i_k}) = \text{In}(v, v_{i_k}) \setminus A$ .

Une fois appliquée les étapes 1 et 2 à tous les sommets  $v$  de l'arbre initial  $T$ , nous aurons fini la construction de l'arbre  $T'$  et de la collection de chemins  $\mathcal{P}'$ .

Nous pouvons remarquer facilement que, par la construction précédente, pour tout sommet  $v$  de l'arbre initial  $T$  et pour tout ensemble  $\text{Out}(v, v_j)$ , avec  $v_j \in N(v)$ , chaque chemin  $v \rightsquigarrow a$  dans  $\text{Out}(v, v_j)$  est remplacé par un nouveau chemin  $w_\alpha \rightsquigarrow a$ , où  $w_\alpha$  est un nouveau sommet ajouté à  $T$ , et aucun autre chemin n'utilise le sommet  $w_\alpha$  comme sommet initial. De plus, il y a un et seulement un chemin  $b \rightsquigarrow v$  dans un des ensembles  $\text{In}(v, v_k)$ , avec  $v_k \in N(v)$ , lequel est remplacé par le chemin  $b \rightsquigarrow w_\alpha$ . Ainsi, l'ensemble des chemins  $\mathcal{P}'$  représente une permutation des sommets de l'arbre



augmenté  $T'$ . De plus, par construction, il n'est pas difficile de vérifier que deux chemins  $a \rightsquigarrow b$  et  $c \rightsquigarrow d$  dans  $\mathcal{P}$  sont en conflit si et seulement si leurs chemins associés dans  $\mathcal{P}'$  sont en conflit. Ainsi,  $R_T(\mathcal{P}) = k$  si et seulement si  $R_{T'}(\mathcal{P}') = k$ , pour un entier positif  $k$ .

De plus, nous pouvons remarquer d'une part qu'après l'étape 1, pour tout sommet  $v$  de l'arbre  $T$ , les ensembles  $\{v_j : v_j \in N(v) \text{ et } \text{Out}(v, v_j) \neq \emptyset\}$  et  $\{v_i : v_i \in N(v) \text{ et } \text{In}(v, v_i) \neq \emptyset\}$  sont disjoints. Ceci implique que si le degré de  $v$  dans  $T$  est égal à  $\Delta$ , alors le degré de  $v$  dans l'arbre augmenté  $T'$  est au plus égal à  $2\Delta - 1$ . D'autre part, le nombre de sommets dans  $T'$  est au plus égal à  $(L\Delta + 1)n$ , ce qui termine la preuve de ce théorème.  $\square$

Une conséquence directe du théorème 26 est que s'il existe un algorithme polynomial  $\epsilon$ -approché, avec  $\epsilon \geq 1$ , pour le problème du routage de permutations dans les arbres par chemins arc-disjoints, alors il existe un algorithme polynomial  $\epsilon$ -approché pour le problème général du routage dans les arbres orientés symétriques par chemins arc-disjoints. De plus, comme nous l'avons vu dans le chapitre 3, le problème du routage par chemins arc-disjoints dans les arbres binaires est NP-difficile. Nous pouvons donc déduire facilement du théorème 26 que le problème de routage de permutations dans les arbres ayant degré borné supérieur ou égal à 5 est un problème NP-difficile. Cependant, dans la section suivante nous allons montrer que le problème de routage de permutations reste NP-difficile même dans des cas plus restrictifs comme par exemple, dans le cas du routage des involutions dans les arbres binaires, où les involutions sont des permutations ayant une décomposition en cycles de longueur au plus 2.

## 5.2 Routage de permutations particulières

Dans cette section nous étudierons la complexité de certaines instances particulières du problème du routage de permutations dans les arbres orientés symétriques. Plus précisément, nous allons étudier la complexité du routage des *involutions* (resp. *permutations circulaires*), qui sont des permutations ayant une décomposition en cycles de longueur au plus 2 (resp. composées d'un seul cycle), dans les arbres de degré borné par une constante et dans les arbres ayant au plus deux sommets de degré supérieur à 2. D'abord, nous allons répondre à une question posée par Caragiannis et al. [16] concernant la complexité du problème de coloration d'une collection *symétrique* de chemins dans les arbres binaires orientés symétriques. Une collection de chemins est dite *symétrique*, si pour tout chemin  $u \rightsquigarrow v$  de la collection, il existe aussi son chemin symétrique, c'est-à-dire, le chemin  $v \rightsquigarrow u$ . Caragiannis et al. [16] montrent que le problème de coloration d'une collection symétrique de chemins dans les arbres quelconques est NP-difficile.

Grâce à la symétrie de la collection de chemins, on peut voir l'arbre orienté symétrique comme étant un arbre non orienté et chaque couple de chemins symétriques  $u \rightsquigarrow v$  et  $v \rightsquigarrow u$  comme un seul chemin non orienté  $\langle u, v \rangle$ . Dans la figure 5.2(b) nous donnons un exemple d'une collection symétrique de chemins sur un arbre

orienté symétrique. Il n'est pas difficile de voir que la congestion induite par toute

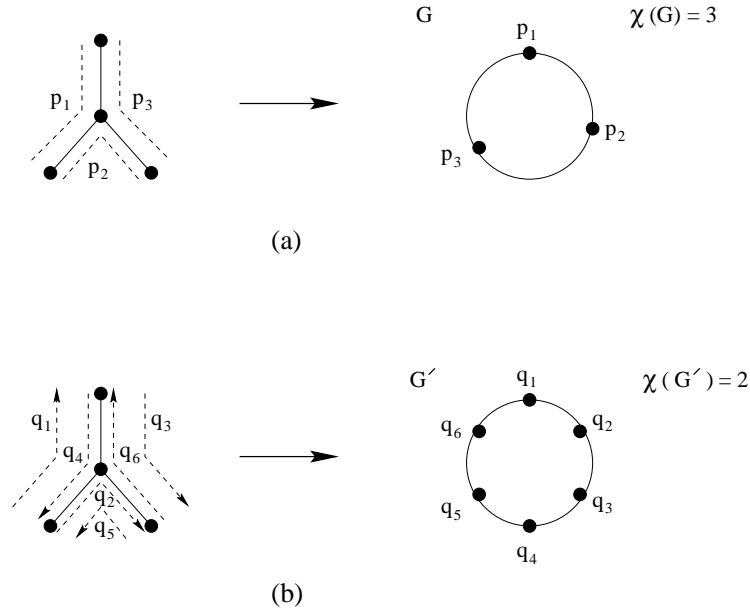


FIGURE 5.2 – (a) Une collection de chemins sur un arbre non orienté et le graphe de conflit  $G$  associé. (b) Une collection symétrique de chemins sur un arbre orienté symétrique et le graphe de conflit  $G'$  associé.

collection symétrique de chemins orientés et par celle de chemins non orientés ont la même valeur (voir figure 5.2). Malgré toutes ces ressemblances, ces deux problèmes de coloration sont différents comme on le voit dans la figure 5.2. En effet, Golubic et Jamison ont montré dans [31] que le problème de coloration de chemins dans les arbres non orientés est NP-difficile même si les arbres sont des étoiles. Tarjan a trouvé pour ce problème dans les arbres quelconques un algorithme  $\frac{3}{2}$ -approché [63]. De plus, Erlebach et Jansen ont montré dans [22] que si le degré des arbres non orientés est borné par une constante, alors le problème de coloration des chemins dans ceux-ci peut être résolu optimalement en temps polynomial.

Dans le cas des arbres orientés symétriques, Gargano et al. ont montré dans [29] que si les arbres sont des étoiles généralisées, alors le problème de coloration de chemins sur cette classe particulière des arbres orientés symétriques peut être calculé efficacement en temps polynomial. Dans le cas des arbres binaires orientés symétriques quelconques, ce problème est NP-difficile [22, 42] au contraire de celui dans les arbres non orientés.

Ainsi, Caragiannis et al. [16] demandent si le problème de coloration d'une collection symétrique de chemins dans les arbres binaires orientés symétriques peut être résolu efficacement en temps polynomial. Nous donnons une réponse négative à cette question, montrant que ce problème reste NP-difficile même pour les arbres binaires. Ensuite, nous utiliserons ce résultat pour montrer la NP-difficulté des certains cas restrictifs du problème de routage de permutations dans les arbres.

Le résultat principal de cette section est donc le suivant.

**Théorème 27** Soient  $T$  un arbre orienté symétrique et  $\mathcal{P}$  une collection de chemins orientés sur  $T$ . Alors, le problème de calculer  $R_T(\mathcal{P})$  est un problème NP-difficile dans les cas suivants :

- (a)  $T$  est un arbre binaire et  $\mathcal{P}$  est une collection symétrique de chemins sur  $T$ .
- (b)  $T$  est un arbre binaire et  $\mathcal{P}$  représente une involution des sommets de  $T$ .
- (c)  $T$  est un arbre de degré maximum  $\Delta \geq 4$ , et  $\mathcal{P}$  représente une permutation circulaire des sommets de  $T$ .
- (d)  $T$  est un arbre ayant uniquement deux sommets de degré supérieur à deux, et  $\mathcal{P}$  représente une involution des sommets de  $T$ .

Pour prouver le théorème 27, nous utiliserons une réduction polynomial du problème ARC-COLORING (voir chapitre 4) qui a été montré NP-complet par Garey et al. [28]. Il faut remarquer qu'une réduction similaire a été utilisée dans [22, 42] pour prouver la NP-difficulté du problème général du routage par chemins arc-disjoints dans les arbres binaires. Cependant, on ne peut pas adapter une telle réduction aux résultats de NP-difficulté sur les instances restrictives considérées dans le théorème 27. Avant de prouver le théorème 27, nous avons besoin des définitions et résultats préliminaires suivants.

**Définition 3** Soient  $H = (V, A)$  un graphe orienté symétrique et  $\mathcal{P}$  une collection de chemins sur  $H$ . Nous définissons le (multi)-graphe orienté  $\vec{G}_H(\mathcal{P}) = (V', A')$  associé à  $H$  et à  $\mathcal{P}$  tel que :

- $V' = \{v \in V : v \text{ est un des sommets terminaux d'au moins un chemin dans } \mathcal{P}\}$   
et
- $A' = \{(v, w) : v, w \in V' \text{ et } v \rightsquigarrow w \in \mathcal{P}\}$ .

Un graphe orienté  $G = (V, A)$  est dit *pseudo-symétrique*, si pour tout sommet  $v \in V$ , on a  $d^+(v) = d^-(v)$ , où  $d^+(v)$  (resp.  $d^-(v)$ ) représente le degré entrant (resp. sortant) du sommet  $v$ . Il est clair que si  $G$  est connexe et pseudo-symétrique, alors  $G$  est fortement-connexe.

**Théorème 28** (Fleischer [26]) Si  $G$  est un (multi)-graphe orienté connexe et pseudo-symétrique, alors  $G$  est eulérien. De plus, un circuit eulérien dans  $G$  peut être trouvé en temps linéaire.

Soit  $\vec{G}_T(\mathcal{P})$  le (multi)-graphe associé à un arbre  $T$  et à une collection de chemins  $\mathcal{P}$  sur  $T$ . Il est clair que même si  $\vec{G}_T(\mathcal{P})$  est pseudo-symétrique, il peut être non connexe, comme nous le montrons dans la figure 5.3. Par contre, si  $\mathcal{P}$  représente une permutation circulaire des sommets dans  $T$ , alors  $\vec{G}_T(\mathcal{P})$  est forcément connexe et pseudo-symétrique.

Le lemme suivant va nous permettre de montrer la partie (c) du théorème 27.

**Lemme 1** Soient  $T$  un arbre orienté symétrique et  $\mathcal{P}$  une collection de chemins sur  $T$ . Soit  $\vec{G}_T(\mathcal{P})$  le (multi)-graphe orienté associé à  $T$  et  $\mathcal{P}$ . Si  $\vec{G}_T(\mathcal{P})$  est connexe et pseudo-symétrique, alors on peut construire en temps polynomial un arbre  $T'$  orienté

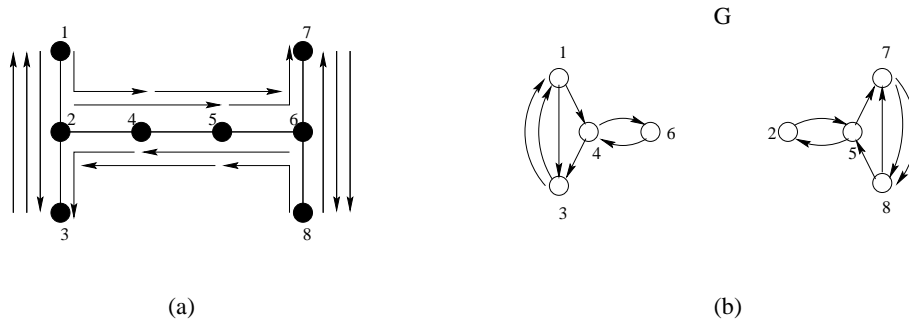


FIGURE 5.3 – (a) Un arbre  $T$  et une collection de chemins  $\mathcal{P}$  sur  $T$ . (b) Le (multi)-graphe  $G = \vec{G}_T(\mathcal{P})$  associé à  $T$  et à  $\mathcal{P}$ .

symétrique et un ensemble de chemins  $\mathcal{P}'$  représentant une permutation circulaire des sommets dans  $T'$  tels que  $R_{T'}(\mathcal{P}') = k$  si et seulement si  $R_T(\mathcal{P}) = k$ , pour un entier positif  $k$ .

**Preuve.** Soit  $L = L_T(\mathcal{P})$  la congestion induite par  $\mathcal{P}$  dans  $T$ . Comme dans la preuve du théorème 26, nous pouvons supposer sans perte de généralité que tous les arcs de  $T$  sont traversés par exactement  $L$  chemins dans  $\mathcal{P}$ . Soit  $n$  le nombre de sommets dans  $T$ . Nous supposons que chaque sommet dans  $T$  est étiqueté par un entier différent dans l'ensemble  $\{1, 2, \dots, n\}$ . Soit  $G = \vec{G}_T(\mathcal{P}) = (V', A)$  le (multi)-graphe orienté associé à  $T$  et  $\mathcal{P}$ . Par hypothèse, nous savons que  $G$  est connexe et pseudo-symétrique. Ainsi, par le théorème 28,  $G$  est eulérien et un circuit eulérien dans  $G$  peut être trouvé en temps linéaire. Dans la figure 5.4(b) et 5.4(c) respectivement, nous montrons un exemple d'un tel (multi)-graphe  $G$  orienté et d'un circuit eulérien de  $G$ . Soit  $v_0, v_1, \dots, v_k, v_0$  un circuit eulérien de  $G$ . Par construction, chaque arc  $(v_i, v_{i+1})$  dans le circuit eulérien représente un chemin  $v_i \rightsquigarrow v_{i+1}$  dans  $\mathcal{P}$ . La construction de l'arbre  $T'$  et de l'ensemble de chemins  $\mathcal{P}'$  représentant une permutation circulaire des sommets dans  $T'$  est faite de la façon suivante. Initialement,  $T'$  est identique à  $T$  et  $\mathcal{P}'$  est identique à  $\mathcal{P}$ . Soit  $\mathcal{M}$  une liste de sommets dans  $T$  (initialement  $\mathcal{M} = \{v_0\}$ ), et pour chaque sommet  $v$  dans  $T$ , soit  $\mathcal{C}(v)$  une liste de sommets dans  $T'$  (initialement,  $\mathcal{C}(v) = \emptyset$  pour tout sommet  $v$  de  $T$ ). Soit  $p_0, p_1, \dots, p_k$  un ordre des chemins dans  $\mathcal{P}'$  induit par le circuit eulérien, où  $p_i = v_i \rightsquigarrow v_{i+1}$ ,  $0 \leq i \leq k$  (l'addition est comprise modulo  $(k+1)$ ). Alors, pour chaque  $i = 0, 1, \dots, k-1$ , on exécute les étapes (1)-(4) de la procédure suivante (voir exemple dans la figure 5.4(d) qui utilise le circuit eulérien  $v_0 = 2, 7, 4, 3, 1, 2, 4, 3, 6, 7, 5, 6, 1 = v_{k=12}$  de la figure 5.4(c)).

### Procédure 2 :

Pour tout  $i = 0, 1, \dots, k$ , soit  $(a_i, b_i)$  (resp.  $(a_{i+1}, b_{i+1})$ ) le couple de sommets initial et final courants du chemin  $p_i$  (resp.  $p_{i+1}$ ) dans  $\mathcal{P}'$ , c'est-à-dire,  $p_i = a_i \rightsquigarrow b_i$  et  $p_{i+1} = a_{i+1} \rightsquigarrow b_{i+1}$ , avec  $b_i = a_{i+1}$ . Alors :

1. Si  $a_{i+1} \notin \mathcal{M}$  alors, on ajoute  $a_{i+1}$  à  $\mathcal{M}$  et on laisse inchangés les chemins  $p_i$  et  $p_{i+1}$  et on aura fini l'exécution de cette procédure pour  $i$ . Sinon, soit  $w_i$  (resp.  $w_{i+1}$ ) le prédécesseur (resp. successeur) immédiat du sommet  $a_{i+1}$  sur

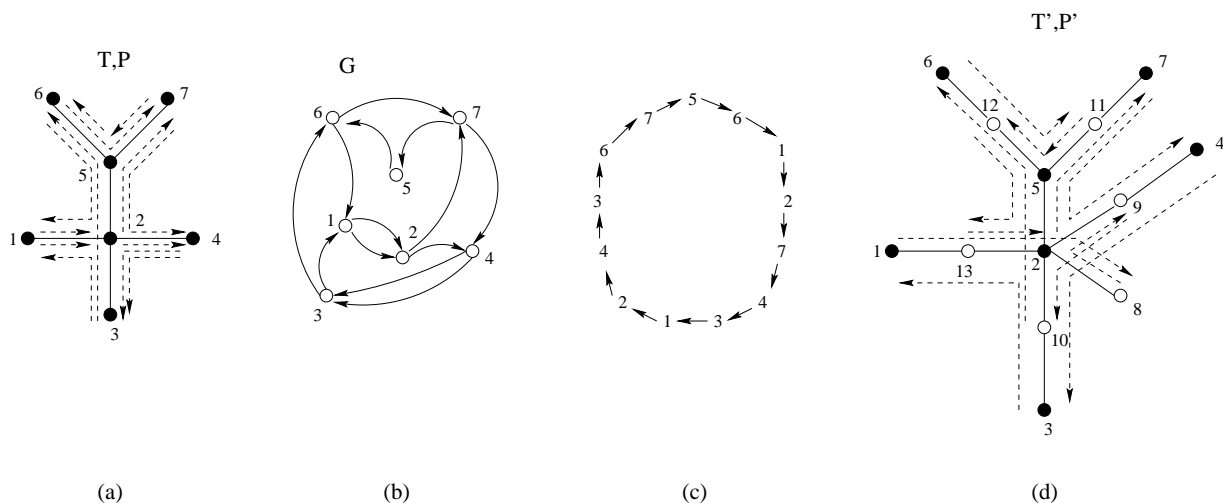


FIGURE 5.4 – (a) Un arbre  $T$  et une collection de chemins  $\mathcal{P}$  sur  $T$ . (b) Le (multi)-graphe orienté  $G = \vec{G}_T(\mathcal{P})$  associé à  $T$  et  $\mathcal{P}$ . (c) Un circuit eulérien de  $G$ . (d) Construction d'un arbre  $T'$  et d'un ensemble de chemins  $\mathcal{P}'$  représentant la permutation circulaire  $\sigma = (1, 8, 9, 10, 6, 11, 5, 12, 13, 2, 7, 4, 3)$  des sommets de  $T'$ , à partir de  $T$  et  $\mathcal{P}$ .

- le chemin  $p_i$  (resp.  $p_{i+1}$ ), et soit  $u_{i+1}$  un nouveau sommet qu'on va ajouter à l'arbre  $T'$ .
2. Si  $w_i = w_{i+1}$  alors, on déconnecte les sommets  $a_{i+1}$  et  $w_{i+1}$ , et on connecte le sommet  $u_{i+1}$  aux sommets  $a_{i+1}$  et  $w_{i+1}$ , et on passe à l'étape (4).
  3. S'il existe un sommet  $z \in \mathcal{C}(a_{i+1})$  qui soit le sommet final et initial respectivement de deux chemins  $q$  et  $q'$  dans  $\mathcal{P}'$  et  $q$  (resp.  $q'$ ) traverse l'arc  $(w_i, a_{i+1})$  (resp. l'arc  $(a_{i+1}, w_{i+1})$ ), alors on connecte le sommet  $u_{i+1}$  au sommet  $z$  et on élimine le sommet  $z$  de la liste  $\mathcal{C}(a_{i+1})$ . Sinon, on connecte le sommet  $u_{i+1}$  au sommet  $a_{i+1}$ . Finalement, on ajoute le sommet  $u_{i+1}$  à la liste  $\mathcal{C}(a_{i+1})$  et on passe à l'étape (4).
  4. On remplace les chemins  $p_i$  et  $p_{i+1}$  respectivement par les chemins  $a_i \rightsquigarrow u_{i+1}$  et  $u_{i+1} \rightsquigarrow b_{i+1}$ .

Dans la figure 5.4(d) on peut voir un exemple de la construction de l'arbre  $T'$  et de l'ensemble de chemins  $\mathcal{P}'$ . En effet, pour construire  $T'$  et  $\mathcal{P}'$  à partir de l'arbre  $T$  et la collection de chemins  $\mathcal{P}$  sur  $T$  montrés dans la figure 5.4(a) en utilisant la procédure 2, nous considérons le circuit eulérien  $v_0 = 2, 7, 4, 3, 1, 2, 4, 3, 6, 7, 5, 6, 1 = v_{k=12}$  (voir figure 5.4(c)) du (multi)-graphe orienté  $G = \vec{G}_T(\mathcal{P})$  (voir figure 5.4(b)). Ainsi, initialement on a  $\mathcal{M} = \{2\}$  et  $\mathcal{C}(j) = \emptyset$ , avec  $1 \leq j \leq 7$ . Après l'application de la procédure 2 sur tous les chemins de  $\mathcal{P}$ , on a  $\mathcal{M} = \{2, 7, 4, 3, 1, 6, 5\}$ ,  $\mathcal{C}(2) = \{8\}$ , et  $\mathcal{C}(j) = \emptyset$ , avec  $1 \leq j \neq 2 \leq 7$ . Il est clair que la construction précédente peut être effectuée en temps polynomial. Soit  $\Delta$  le degré maximum de l'arbre  $T$ , et soient  $n'$  et  $\Delta'$  le nombre de sommets et le degré maximum de l'arbre  $T'$  respectivement. Par la construction précédente, il est facile de prouver que  $n'$  est au plus égal à  $Ln\Delta$ , et

que  $\Delta'$  est au plus égal à  $\Delta(\Delta - 1)$ . D'autre part, nous pouvons remarquer qu'après la construction précédente, chacun des sommets  $v$  dans l'arbre final  $T'$  est soit le début et la fin d'exactly un et un seul chemin dans  $\mathcal{P}'$ , soit il n'y a aucun chemin dans  $\mathcal{P}'$  ayant le sommet  $v$  comme sommet initial ou sommet final. De plus, comme les chemins dans  $\mathcal{P}'$  suivent, par construction, l'ordre donné par le circuit eulérien de  $G$ , alors  $\mathcal{P}'$  représente une permutation circulaire des sommets dans  $T'$ . Finalement, par construction, il est facile de voir que toute paire de chemins  $p$  et  $q$  dans  $\mathcal{P}$  sont en conflit si et seulement si leurs chemins associés  $p'$  et  $q'$  respectivement, dans  $\mathcal{P}'$  sont en conflit. Ceci implique que  $R_T(\mathcal{P}) = k$  si et seulement si  $R_{T'}(\mathcal{P}') = k$ , pour un entier positif  $k$ . Nous avons donc prouvé ce lemme.  $\square$

## Preuve du théorème 27

On utilise une réduction polynomiale du problème *ARC-COLORING* (voir chapitre 4). Une instance du problème *ARC-COLORING* est donnée par un anneau non orienté  $C_n$  à  $n$  sommets, une collection de chemins non orientés  $F$  sur  $C_n$  et un entier positif  $k$ . Ce problème consiste à savoir si  $F$  peut être coloriée avec au plus  $k$  couleurs de sorte que toute paire de chemins dans  $F$  partageant une même arête dans  $C_n$ , sont coloriés avec des couleurs distinctes.

Nous supposons, sans perte de généralité, que chaque arête dans  $C_n$  est traversée par exactement  $k$  chemins dans  $F$ . S'il existe une arête  $\{i, i+1\}$  dans  $C_n$  traversée par  $r < k$  chemins dans  $F$ , alors on peut ajouter  $k - r$  chemins identiques à  $\langle i, i+1 \rangle$  (ou  $\langle i, 1 \rangle$  si  $i = n$ ) à  $F$ , sans affecter ses propriétés de coloration. Nous supposons aussi qu'aucun chemin dans  $F$  couvre tout  $C_n$ . Soit  $I$  une instance du problème *ARC-COLORING* définie comme ci-dessus. Nous allons construire à partir de l'instance  $I$ , une instance  $I'$  du problème de routage de permutations par chemins arc-disjoints dans les arbres, formée d'un arbre orienté symétrique  $T$ , d'un ensemble de chemins orientés  $\mathcal{P}$  représentant une permutation des sommets de  $T$  et d'un entier positif  $k'$  telle que  $I'$  vérifie les contraintes données dans la partie (a) (resp. (b), (c), (d)), et telle que  $F$  est  $k$ -coloriable si et seulement si  $\mathcal{P}$  est  $k'$ -coloriable.

Soit  $\langle i, j \rangle$  un chemin quelconque dans  $F$ . On dit que  $\langle i, j \rangle$  est un chemin de *type 1* (resp. *type 2*) si  $i < j$  (resp. si  $i > j$ ). Dans la figure 5.5 nous donnons des exemples de ces deux types de chemins dans  $F$ .

**Preuve de la partie (a).** L'arbre binaire  $T$  est construit de la façon suivante. Tout d'abord, on construit une chaîne  $P_n$  orientée symétrique à  $n$  sommets qu'on note par  $v_1, v_2, \dots, v_n$ . Ensuite, on construit  $2(n+k)$  copies différentes isomorphes au graphe étoile  $ST(4)$ . Pour  $n+k$  de ces  $2(n+k)$  graphes isomorphes, on note leurs trois feuilles par  $l_i, s_i$  et  $t_i$ , et on note par  $r_i, x_i$  et  $z_i$ , les trois feuilles des autres  $n+k$  graphes isomorphes, avec  $1 \leq i \leq n+k$ . Finalement, on connecte le sommet  $l_i$  (resp.  $r_i$ ) au sommet  $l_{i+1}$  (resp.  $r_{i+1}$ ),  $1 \leq i \leq n+k-1$ , et on connecte le sommet  $l_1$  (resp.  $r_1$ ) au sommet  $v_1$  (resp.  $v_n$ ) de la chaîne  $P_n$  (voir figure 5.5).

La collection symétrique de chemins  $\mathcal{P}$  est construite de la façon suivante (voir figure 5.5). Pour chaque chemin  $\langle i, j \rangle$  dans  $F$ , si  $\langle i, j \rangle$  est un chemin de type

1 (i.e.,  $i < j$ ), alors on ajoute à  $\mathcal{P}$  les chemins  $A_{i,j} = v_i \rightsquigarrow v_j$  et  $B_{j,i} = v_j \rightsquigarrow v_i$ . Dans le cas contraire, si le chemin  $\langle i, j \rangle$  est un chemin de type 2 (i.e.,  $i > j$ ), soit  $p$  (resp.  $q$ ) un entier dans l'ensemble  $\{1, 2, \dots, k\}$  tel que, aucun chemin dans la collection courante  $\mathcal{P}$  n'utilise les sommets  $x_p$  et  $z_p$  (resp.  $s_q$  et  $t_q$ ) dans  $T$  comme sommets initiaux ou finaux. Alors, on ajoute à  $\mathcal{P}$  les ensembles de chemins  $\bar{A}_{i,j} = \{v_i \rightsquigarrow z_p, x_p \rightsquigarrow t_q, s_q \rightsquigarrow v_j\}$  et  $\bar{B}_{j,i} = \{v_j \rightsquigarrow s_q, t_q \rightsquigarrow x_p, z_p \rightsquigarrow v_i\}$ .

Pour assurer que le (multi)-graphe orienté  $\vec{G}_T(\mathcal{P})$  associé à  $T$  et  $\mathcal{P}$  (voir définition 3) soit connexe (propriété que nous utiliserons dans la suite pour prouver la partie (c) du théorème), pour chaque  $j$ , avec  $k+1 \leq j \leq n+k$ , on ajoute à  $\mathcal{P}$  les ensembles de chemins  $C_{j-k} = \{s_j \rightsquigarrow v_{j-k}, v_{j-k} \rightsquigarrow z_{j'}, x_{j'} \rightsquigarrow v_{j'-k}, v_{j'-k} \rightsquigarrow t_j\}$  et  $D_{j-k} = \{t_j \rightsquigarrow v_{j'-k}, v_{j'-k} \rightsquigarrow x_{j'}, z_{j'} \rightsquigarrow v_{j-k}, v_{j-k} \rightsquigarrow s_j\}$ , où  $j' = j+1$  si  $j < n+k$ , sinon  $j' = k+1$ .

De plus, pour chaque  $i$ , avec  $1 \leq i \leq n+k$ , on ajoute  $2(n+k) - 1$  chemins identiques  $s_i \rightsquigarrow t_i$  (resp.  $x_i \rightsquigarrow z_i$ ) et  $2(n+k) - 1$  chemins identiques  $t_i \rightsquigarrow s_i$  (resp.  $z_i \rightsquigarrow x_i$ ) à  $\mathcal{P}$ . Finalement, on pose  $k' = 2(n+k)$ . Dans la figure 5.5, on présente un exemple de cette transformation polynomiale.

Par construction, il est facile d'observer que  $T$  est un arbre binaire et  $\mathcal{P}$  est une collection symétrique de chemins sur  $T$ . De plus, soient  $\langle i_1, j_1 \rangle, \langle i_2, j_2 \rangle, \dots, \langle i_k, j_k \rangle$  les  $k$  chemins dans  $F$  de type 2, et  $\bar{A}_{i_r, j_r} = \{v_{i_r} \rightsquigarrow z_{p_r}, x_{p_r} \rightsquigarrow t_{q_r}, s_{q_r} \rightsquigarrow v_{j_r}\}$  et  $\bar{B}_{j_r, i_r} = \{v_{j_r} \rightsquigarrow s_{q_r}, t_{q_r} \rightsquigarrow x_{p_r}, z_{p_r} \rightsquigarrow v_{i_r}\}$  les deux ensembles de chemins dans  $\mathcal{P}$  associés à chacun des chemins  $\langle i_r, j_r \rangle$  dans  $F$  de type 2, avec  $1 \leq r \leq k$ . Alors, s'il existe une  $k'$ -coloration propre pour la collection symétrique de chemins  $\mathcal{P}$ , alors  $\mathcal{P}$  vérifie les trois propriétés suivantes :

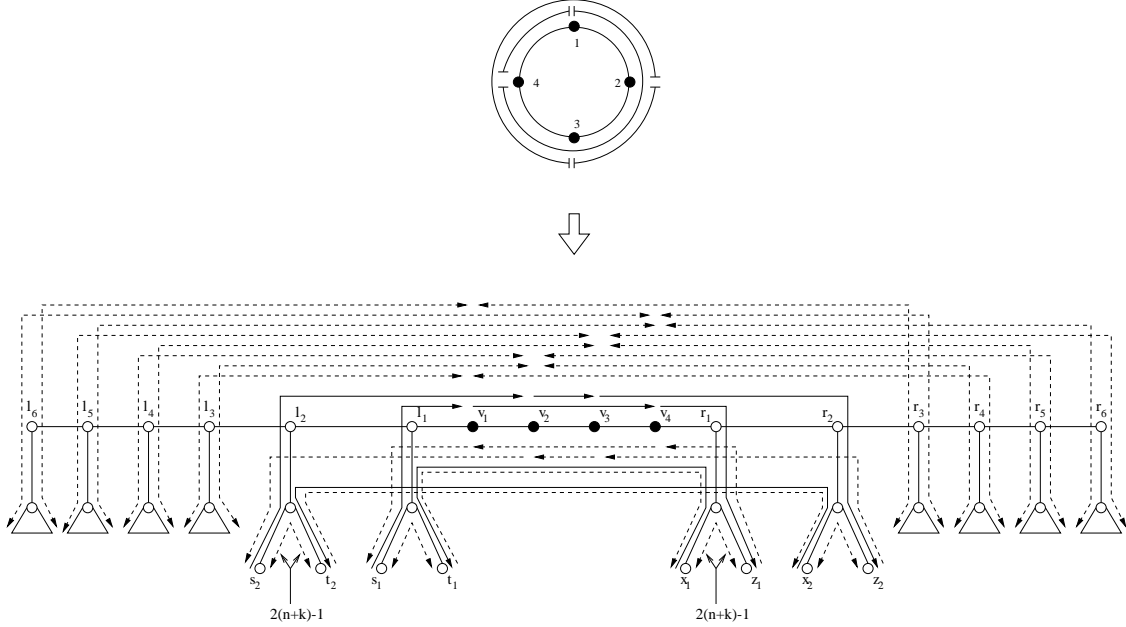
**Propriété 6** *Tous les chemins dans chacun des ensembles  $\bar{A}_{i_r, j_r}, \bar{B}_{j_r, i_r}, C_m$  et  $D_m$ , avec  $1 \leq r \leq k$  et  $1 \leq m \leq n$ , sont coloriés avec la même couleur dans toute  $k'$ -coloration de  $\mathcal{P}$ .*

**Propriété 7** *À chacun des ensembles  $\bar{A}_{i_r, j_r}, \bar{B}_{j_r, i_r}, C_m$  et  $D_m$ , avec  $1 \leq r \leq k$  et  $1 \leq m \leq n$ , doit être assigné une couleur différente dans toute  $k'$ -coloration de  $\mathcal{P}$ .*

**Propriété 8** *Chacun des chemins  $A_{a,b}$  (resp.  $B_{b,a}$ ) dans  $\mathcal{P}$  associé avec un chemin  $\langle a, b \rangle$  dans  $F$  de type 1, partage au moins un arc dans  $T$  avec tous les chemins dans  $\cup_{r=1}^k \{t_{q_r} \rightsquigarrow x_{p_r} \in \bar{B}_{j_r, i_r}\}$  (resp. dans  $\cup_{r=1}^k \{x_{p_r} \rightsquigarrow t_{q_r} \in \bar{A}_{i_r, j_r}\}$ ) et au moins avec un des chemins dans chacun des ensembles  $C_m$  et  $D_m$ ,  $1 \leq m \leq n$ .*

Pour tout  $j$ , avec  $k+1 \leq j \leq n+k$ , nous notons  $C_{j-k}^g$  (resp.  $C_{j-k}^d$ ) le sous-ensemble de  $C_{j-k}$  composé des chemins  $\{s_j \rightsquigarrow v_{j-k}, v_{j-k} \rightsquigarrow t_j\}$  (resp.  $\{v_{j-k} \rightsquigarrow z_{j'}, x_{j'} \rightsquigarrow v_{j'-k}\}$ ), où  $j' = j+1$  si  $j < n+k$ , sinon  $j' = k+1$ . De même, nous notons  $D_{j-k}^g = \{t_j \rightsquigarrow v_{j'-k}, v_{j'-k} \rightsquigarrow s_j\}$  et  $D_{j-k}^d = \{v_{j'-k} \rightsquigarrow x_{j'}, z_{j'} \rightsquigarrow v_{j-k}\}$  les sous-ensembles de  $D_{j-k}$ . D'une part, par construction, les  $k' - 1$  chemins identiques  $s_i \rightsquigarrow t_i$  (resp.  $t_i \rightsquigarrow s_i$ ) et les  $k' - 1$  chemins identiques  $x_i \rightsquigarrow z_i$  (resp.  $z_i \rightsquigarrow x_i$ ), avec  $1 \leq i \leq n+k$ , assurent que tous les chemins dans chacun des ensembles  $\bar{A}_{i_r, j_r}, \bar{B}_{j_r, i_r}, C_m^g, C_m^d, D_m^g$  et  $D_m^d$ , avec  $1 \leq r \leq k$  et  $1 \leq m \leq n$ , sont coloriés avec la même couleur dans toute  $k'$ -coloration de  $\mathcal{P}$ . D'autre part, il n'est pas difficile de voir

$$F = \{ \langle 1, 4 \rangle, \langle 2, 3 \rangle, \langle 4, 1 \rangle, \langle 3, 2 \rangle \}, n = 4 \text{ et } k = 2$$



$$\langle 1, 4 \rangle \longrightarrow A_{1,4} = v_1 \rightsquigarrow v_4 \text{ et } B_{4,1} = v_4 \rightsquigarrow v_1$$

$$\langle 2, 3 \rangle \longrightarrow A_{2,3} = v_2 \rightsquigarrow v_3 \text{ et } B_{3,2} = v_3 \rightsquigarrow v_2$$

$$\langle 4, 1 \rangle \longrightarrow \bar{A}_{4,1} = \{v_4 \rightsquigarrow z_1, x_1 \rightsquigarrow t_1, s_1 \rightsquigarrow v_1\} \text{ et } \bar{B}_{1,4} = \{v_1 \rightsquigarrow s_1, t_1 \rightsquigarrow x_1, z_1 \rightsquigarrow v_4\}$$

$$\langle 3, 2 \rangle \longrightarrow \bar{A}_{3,2} = \{v_3 \rightsquigarrow z_2, x_2 \rightsquigarrow t_2, s_2 \rightsquigarrow v_2\} \text{ et } \bar{B}_{2,3} = \{v_2 \rightsquigarrow s_2, t_2 \rightsquigarrow x_2, z_2 \rightsquigarrow v_3\}$$

$$C_1 = \{s_3 \rightsquigarrow v_1, v_1 \rightsquigarrow z_4, x_4 \rightsquigarrow v_2, v_2 \rightsquigarrow t_3\} \text{ et } D_1 = \{t_3 \rightsquigarrow v_2, v_2 \rightsquigarrow x_4, z_4 \rightsquigarrow v_1, v_1 \rightsquigarrow s_3\}$$

$$C_2 = \{s_4 \rightsquigarrow v_2, v_2 \rightsquigarrow z_5, x_5 \rightsquigarrow v_3, v_3 \rightsquigarrow t_4\} \text{ et } D_2 = \{t_4 \rightsquigarrow v_3, v_3 \rightsquigarrow x_5, z_5 \rightsquigarrow v_2, v_2 \rightsquigarrow s_4\}$$

$$C_3 = \{s_5 \rightsquigarrow v_3, v_3 \rightsquigarrow z_6, x_6 \rightsquigarrow v_4, v_4 \rightsquigarrow t_5\} \text{ et } D_3 = \{t_5 \rightsquigarrow v_4, v_4 \rightsquigarrow x_6, z_6 \rightsquigarrow v_3, v_3 \rightsquigarrow s_5\}$$

$$C_4 = \{s_6 \rightsquigarrow v_4, v_4 \rightsquigarrow z_3, x_3 \rightsquigarrow v_1, v_1 \rightsquigarrow t_6\} \text{ et } D_4 = \{t_6 \rightsquigarrow v_1, v_1 \rightsquigarrow x_3, z_3 \rightsquigarrow v_4, v_4 \rightsquigarrow s_6\}$$

FIGURE 5.5 – Construction partielle de l'instance  $I'$  à partir de l'instance  $I$ .

que par construction, chaque chemin  $x_{p_r} \rightsquigarrow t_{q_r} \in \bar{A}_{i_r, j_r}$  (resp.  $t_{q_r} \rightsquigarrow x_{p_r} \in \bar{B}_{j_r, i_r}$ ), avec  $1 \leq r \leq k$ , partage au moins un arc dans  $T$  avec tous les chemins dans  $\cup_{m=1}^k \{x_{p_m} \rightsquigarrow t_{q_m} \in \bar{A}_{i_m, j_m} : m \neq r\}$  (resp. dans  $\cup_{m=1}^k \{t_{q_m} \rightsquigarrow x_{p_m} \in \bar{B}_{j_m, i_m} : m \neq r\}$ ) et avec tous les chemins dans  $\cup_{m=1}^k \{z_{p_m} \rightsquigarrow v_{i_m}, v_{j_m} \rightsquigarrow s_{q_m} \in \bar{B}_{j_m, i_m}\}$  (resp. dans  $\cup_{m=1}^k \{v_{i_m} \rightsquigarrow z_{p_m}, s_{q_m} \rightsquigarrow v_{j_m} \in \bar{A}_{i_m, j_m}\}$ ) (c.f. figure 5.5). De plus, chacun des ensembles de chemins  $C_m$  (resp.  $D_m$ ), avec  $1 \leq m \leq n$ , partage au moins un arc dans  $T$  avec chacun des chemins dans  $\mathcal{P} \setminus C_m \setminus Q$  (resp.  $\mathcal{P} \setminus D_m \setminus Q$ ), où  $Q$  est la collection de chemins formée par tous les  $k' - 1$  chemins identiques  $s_i \rightsquigarrow t_i$  (resp.  $t_i \rightsquigarrow s_i$ ) et les  $k' - 1$  chemins identiques  $x_i \rightsquigarrow z_i$  (resp.  $z_i \rightsquigarrow x_i$ ), avec  $1 \leq i \leq k$  (c.f. figure 5.5).

Supposons maintenant qu'il existe un ensemble  $C_{j-k}$  quelconque tel que ses sous-ensembles de chemins  $C_{j-k}^g$  et  $C_{j-k}^d$  sont coloriés avec des couleurs distinctes.



Alors, par construction (les 4 sommets terminaux des deux chemins dans  $C_{j-k}^g$  (resp.  $C_{j-k}^d$ ) sont deux-à-deux différents) et par les remarques précédents, toute coloration propre optimale pour  $\mathcal{P}$  aura besoin d'au moins  $k' + 1$  couleurs, ce qui est une contradiction à la hypothèse initiale qui affirme que  $\mathcal{P}$  est  $k'$ -coloriable. Ainsi, tous les chemins dans chacun des ensembles  $C_m$ , avec  $1 \leq m \leq n$ , sont coloriés avec la même couleur dans toute  $k'$ -coloration de  $\mathcal{P}$ . De façon symétrique, nous prouvons pour les chemins dans chacun des ensembles  $D_m$ . Ceci prouve les propriétés 6 et 7. Finalement, la propriété 8 peut être facilement vérifiée par construction.

Maintenant, nous affirmons qu'il existe une  $k$ -coloration pour  $F$  si et seulement s'il existe une  $k'$ -coloration pour  $\mathcal{P}$ . Tout d'abord, supposons qu'il existe une  $k$ -coloration pour  $F$ , et soit  $\langle i, j \rangle$  un chemin dans  $F$  colorié avec la couleur  $\gamma$ , avec  $1 \leq \gamma \leq k$ . Une  $k'$ -coloration de  $\mathcal{P}$  peut être obtenue à partir d'une  $k$ -coloration de  $F$  de la façon suivante. Si le chemin  $\langle i, j \rangle$  dans  $F$  est un chemin de type 1, alors on colorie les chemins  $A_{i,j} = v_i \rightsquigarrow v_j$  et  $B_{j,i} = v_j \rightsquigarrow v_i$  dans  $\mathcal{P}$  avec les couleurs  $\gamma$  et  $\gamma + k$  respectivement. Par contre, si le chemin  $\langle i, j \rangle$  dans  $F$  est de type 2, alors on colorie tous les trois chemins dans l'ensemble  $\bar{A}_{i,j}$  (resp.  $\bar{B}_{j,i}$ ) avec la couleur  $\gamma$  (resp.  $\gamma + k$ ). Ensuite, pour chaque  $i$ , avec  $1 \leq i \leq n$ , on assigne à tous les chemins dans l'ensemble  $C_i$  (resp.  $D_i$ ) la couleur  $2k + i$  (resp. la couleur  $2k + n + i$ ). Finalement, pour chaque  $i$ , avec  $1 \leq i \leq n + k$ , on colorie les  $k' - 1$  chemins identiques  $s_i \rightsquigarrow t_i$  (resp.  $x_i \rightsquigarrow z_i$ ) et les  $k' - 1$  chemins identiques  $t_i \rightsquigarrow s_i$  (resp.  $z_i \rightsquigarrow x_i$ ) avec les  $k' - 1$  couleurs disponibles pour chacun de ces  $(k' - 1)$ -ensembles de chemins. Par les propriétés 6, 7 et 8, il est facile de vérifier que la coloration précédente est une  $k'$ -coloration propre de la collection de chemins  $\mathcal{P}$ .

Supposons maintenant qu'il existe une  $k'$ -coloration pour  $\mathcal{P}$ . Par les propriétés 6, 7 et 8, il est facile de déduire deux  $k$ -colorations propres différentes pour  $F$ . Si  $\langle i, j \rangle$  est un chemin dans  $F$  de type 1, on assigne au chemin  $\langle i, j \rangle$  la couleur assignée au chemin  $A_{i,j} = v_i \rightsquigarrow v_j$  (resp.  $B_{j,i} = v_j \rightsquigarrow v_i$ ) dans  $\mathcal{P}$ . Par contre, si  $\langle i, j \rangle$  est un chemin dans  $F$  de type 2, on assigne au chemin  $\langle i, j \rangle$  la couleur assignée aux trois chemins dans l'ensemble  $\bar{A}_{i,j}$  (resp.  $\bar{B}_{j,i}$ ) qui, par la propriété 6, est la même pour tous les trois. On a donc montré qu'il existe une  $k$ -coloration propre pour  $F$  si et seulement s'il existe une  $k'$ -coloration propre pour  $\mathcal{P}$ . Ceci termine la preuve de cette partie.

**Preuve de la partie (b).** La preuve de cette partie est une conséquence directe de la partie (a). En effet, soient l'arbre binaire  $T$  et la collection symétrique de chemins  $\mathcal{P}$  sur  $T$  construits dans la partie (a). Soient  $u$  et  $v$  deux sommets adjacents dans  $T$ , et soit  $\text{In}(v, u)$  (resp.  $\text{Out}(v, u)$ ) le sous-ensemble de chemins dans  $\mathcal{P}$  qui traversent l'arc  $(u, v)$  (resp. l'arc  $(v, u)$ ) et qui ont comme sommet initial (resp. sommet final) le sommet  $v$ . Comme  $\mathcal{P}$  est symétrique, alors il est clair que  $|\text{In}(v, u)| = |\text{Out}(v, u)|$  (voir preuve du théorème 26). On remplace les arcs  $(u, v)$  et  $(v, u)$  dans  $T$  par une chaîne  $P_\alpha$  orientée symétrique à  $\alpha = |\text{In}(v, u)|$  sommets qu'on note par  $w_1, w_2, \dots, w_\alpha$ . Ensuite, on remplace chaque paire de chemins symétriques  $a \rightsquigarrow v \in \text{In}(v, u)$  et  $v \rightsquigarrow a \in \text{Out}(v, u)$  pour les chemins  $a \rightsquigarrow w_j$  et  $w_j \rightsquigarrow a$  respectivement, où  $w_j$  est un sommet dans  $P_\alpha$  pas encore utilisé par aucun chemin dans la

collection courante  $\mathcal{P}$  comme sommet initial ou final. En utilisant la transformation précédente sur chaque paire de sommets adjacents dans  $T$ , on obtient une instance donnée par un arbre binaire augmenté  $T'$  et un ensemble de chemins  $\mathcal{P}'$  représentant une involution des sommets dans  $T'$ , laquelle est équivalente à celle-ci obtenue dans la partie (a). Ceci termine la preuve de cette partie.

**Preuve de la partie (c).** Soient l'arbre binaire  $T$  et la collection de chemins  $\mathcal{P}$  sur  $T$  obtenus dans la construction de la partie (a). Il est clair que le (multi)-graphe orienté  $\vec{G}_T(\mathcal{P})$  associé à  $T$  et à  $\mathcal{P}$  (voir définition 3) est connexe et pseudo-symétrique. Tout d'abord, on utilise une procédure similaire à celle employée dans la partie (b) laquelle préserve la connexité du (multi)-graphe associé à l'arbre et à la collection de chemins obtenus comme suit. Rappelons que le sommet de  $T$  étiqueté par  $v_i$  fait partie de la chaîne  $P_n$  construite dans le début de la preuve de la partie (a)), avec  $1 \leq i \leq n$ . Pour tout sommet  $v_i$ , soient  $u_i^1$  et  $u_i^2$  les deux sommets de  $T$  voisins de  $v_i$ . Ainsi, pour chaque sommet  $v_i$  de  $T$ , si la paire d'arcs  $(u_i^1, v_i)$  et  $(v_i, u_i^1)$  (resp.  $(u_i^2, v_i)$  et  $(v_i, u_i^2)$ ) doit être remplacée par une nouvelle chaîne à  $\alpha$  sommets qu'on note par  $w_1, w_2, \dots, w_\alpha$  (voir partie (b)), où  $w_1$  sera le nouveau sommet adjacent au sommet  $v_i$  et le sommet  $w_\alpha$  sera le nouveau sommet adjacent au sommet  $u_i^1$  (resp.  $u_i^2$ ), alors après un tel remplacement, on doit ajouter à  $\mathcal{P}$  les chemins  $w_j \rightsquigarrow w_{j+1}$  et  $w_{j+1} \rightsquigarrow w_j$ ,  $1 \leq j \leq \alpha$ , et les chemins  $w_1 \rightsquigarrow v_i$  et  $v_i \rightsquigarrow w_\alpha$ . Il n'est pas difficile de vérifier que cette nouvelle instance est équivalente (du point de vue de la coloration) à celle obtenue dans la partie (a), et que chaque sommet interne dans l'arbre courant est le sommet initial ou final d'au plus 3 chemins dans la collection courante de chemins. Soient  $T'$  et  $\mathcal{P}'$  l'arbre binaire et la collection de chemins courants après avoir appliqué la transformation précédente. Alors, le (multi)-graphe orienté  $\vec{G}_{T'}(\mathcal{P}')$  associé à  $T'$  et à  $\mathcal{P}'$  est connexe et pseudo-symétrique. Par le théorème 1,  $T'$  et  $\mathcal{P}'$  peuvent donc être transformés, en temps polynomial, en un arbre augmenté  $T''$  et un ensemble de chemins  $\mathcal{P}''$  respectivement, tels que  $\mathcal{P}''$  représente une permutation circulaire des sommets dans  $T''$ , et tel que  $R_{T'}(\mathcal{P}') = k'$  si et seulement si  $R_{T''}(\mathcal{P}'') = k'$ . De plus, on peut vérifier facilement que le degré de l'arbre  $T''$  construit par le procédure 2 donnée dans la preuve du théorème 1 est au plus égal à 4. En effet, les chemins dans  $\mathcal{P}'$  ayant pour sommet initial ou final un des sommets internes de  $T'$  exécuteront dans le pire des cas, une fois l'étape (1) et deux fois l'étape (3) de la procédure 2 donnée dans la preuve du théorème 1. Par construction (voir figure 5.5), tous les sommets internes de  $T'$  de degré supérieur à deux ne sont pas sommets initiaux ou finaux d'un chemin dans  $\mathcal{P}'$ . De plus, les  $2k'$  chemins dans  $\mathcal{P}'$  qui ont pour sommet initial ou final une feuille de  $T'$  exécuteront une fois l'étape (1) et  $2k' - 1$  fois l'étape (2) de la procédure 2 donnée dans la preuve du théorème 1. On a bien donc que le degré maximum de  $T''$  est au plus égal à 4. Ainsi, en faisant attention des chemins initiaux  $A_{i,j}$  et  $B_{j,i}$  (resp.  $\bar{A}_{i,j}$  et  $\bar{B}_{j,i}$ ) associés aux chemins  $\langle i, j \rangle$  de type 1 (resp. type 2) dans  $F$  (voir partie (a)), on obtient qu'il existe une  $k'$ -coloration propre pour  $\mathcal{P}''$  si et seulement s'il existe une  $k$ -coloration propre pour  $F$ . Ceci termine la preuve de cette partie.

**Preuve de la partie (d).** La preuve de cette partie suit directement des parties

(a) et (b). En effet, soient  $T$  l'arbre binaire et  $\mathcal{P}$  la collection symétrique de chemins construits dans la partie (a). Alors, on remplace tous les  $n + k$  sous-graphes isomorphes au graphe étoile  $ST(4)$  à 4 sommets, où l' $i$ -ème de ceux-ci a par feuilles les sommets dans  $T$  étiquetés par  $l_i$ ,  $s_i$ , et  $t_i$  (resp. par  $r_i$ ,  $x_i$ , et  $z_i$ ), par un seul graphe étoile  $ST(2(n + k) + 1)$  ayant par feuilles les sommets  $s_i$  et  $t_i$  (resp. les sommets  $x_i$  et  $z_i$ ), avec  $1 \leq i \leq n + k$ , et on note  $l_1$  (resp.  $r_1$ ) son unique sommet de degré  $2(n + k)$ . Ensuite, on connecte le sommet  $l_1$  (resp.  $r_1$ ) au sommet  $v_1$  (resp.  $v_n$ ) dans la chaîne  $P_n$  qui est un sous-graphe de  $T$ , et on laisse  $\mathcal{P}$  exactement comme dans la partie (a). Il n'est pas difficile de voir que cette nouvelle instance est équivalente à celle obtenue dans la partie (a) (du point de vue de la coloration). Finalement, en utilisant des arguments similaires à ceux-ci dans la partie (b), nous prouvons la NP-difficulté pour le cas des involutions. Nous avons donc prouvé cette partie. Ainsi le théorème est prouvé.  $\square$

Dans la figure 5.3 nous avons montré un exemple de (multi)-graphe  $\vec{G}_T(\mathcal{P})$  associé à un arbre  $T$  et une collection de chemins  $\mathcal{P}$  sur  $T$ , où  $\vec{G}_T(\mathcal{P})$  est pseudo-symétrique et non connexe. Cependant, il suffit d'ajouter pour tout arc  $(a, b)$  dans  $T$ , un chemin  $a \rightsquigarrow b$  dans  $\mathcal{P}$  de sorte que le (multi)-graphe orienté associé à  $T$  et à la nouvelle collection augmentée de chemins sur  $T$  soit connexe et pseudo-symétrique comme nous le montrons dans la figure 5.6.

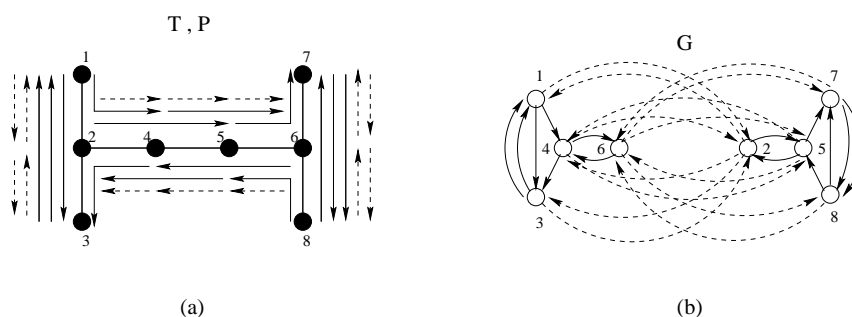


FIGURE 5.6 – (a) L'arbre  $T$  et la collection augmentée de chemins  $\mathcal{P}$  sur  $T$  dans la figure 5.3(a). (b) Le (multi)-graphe  $G = \vec{G}_T(\mathcal{P})$  associé avec  $T$  et  $\mathcal{P}$  dans (a).

Par la remarque précédente et le lemme 1, nous obtenons le résultat suivant.

**Proposition 2** *Si pour tout arbre  $T$  orienté symétrique et pour tout ensemble de chemins  $\mathcal{P}$  avec une congestion  $L$ , représentant une permutation circulaire des sommets dans  $T$ , il existe un algorithme polynomial pour colorier  $\mathcal{P}$  avec au plus  $\lceil \epsilon L \rceil$  couleurs, avec  $\epsilon > 1$ , alors le même algorithme sert à colorier n'importe quelle collection de chemins  $\mathcal{P}'$  avec une congestion  $L'$  sur n'importe quel arbre  $T'$  orienté symétrique, avec au plus  $\lceil \epsilon(L' + 1) \rceil$  couleurs.*

### 5.3 Complexité en moyenne

Dans cette section nous analysons la complexité en moyenne du problème de routage de permutations dans les arbres par chemins arc-disjoints. En effet, dans la section précédente, nous avons montré que ce problème est NP-difficile même pour des instances très particulières. Cependant, ces résultats nous donnent une idée de la complexité du problème dans le pire des cas. Ainsi, dans cette section nous donnons une borne inférieure pour la congestion moyenne induite pour toute permutation sur un réseau quelconque à  $n$  sommets, ce qui implique donc une borne inférieure pour le nombre moyen de couleurs nécessaires pour colorier toute permutation dans un réseau quelconque. Ensuite, nous obtenons des résultats exacts dans le cas où le réseau est une chaîne. Plus précisément, nous montrons que le nombre moyen de couleurs nécessaires pour colorier toute permutation sur une chaîne à  $n$  sommets est égal à  $n/4 + o(n)$ . Nous obtenons aussi des résultats exacts pour les étoiles généralisées. Finalement, nous généralisons les résultats obtenus dans la chaîne pour le cas des arbres quelconques et nous obtenons des bornes inférieure et supérieure pour ce problème dans ces réseaux. Tout d'abord, nous donnons quelques définitions.

Soit  $T$  un arbre raciné orienté symétrique à  $n$  sommets. Nous supposons que les sommets dans  $T$  sont étiquetés de façon arbitraire, par des entiers dans l'ensemble  $\{1, 2, \dots, n\}$  et que le sommet étiqueté par l'entier  $n$  est la racine de  $T$ . Nous notons par  $T(i)$  le sous-arbre de  $T$  raciné au sommet  $i$ , avec  $1 \leq i \leq n$ , où  $T(n) = T$ . Soit  $\mathcal{S}_n$  le groupe symétrique de toutes les permutations sur  $[n] = \{1, 2, \dots, n\}$ . Comme d'habitude dans ce chapitre, toute permutation  $\sigma \in \mathcal{S}_n$  sera représentée par un ensemble de chemins  $\mathcal{P}$  sur  $T$  tel que  $\sigma(i) = j$ , avec  $i \neq j$ , si et seulement si  $i \rightsquigarrow j \in \mathcal{P}$ . Nous notons par  $\bar{L}_T$ , la congestion moyenne induite par toutes les permutations  $\sigma \in \mathcal{S}_n$  sur  $T$ , c'est-à-dire,  $\bar{L}_T = \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} L_T(\sigma)$ , où  $L_T(\sigma)$  est la valeur de la congestion induite par l'ensemble de chemins sur  $T$  représentant une permutation  $\sigma \in \mathcal{S}_n$ . Finalement, nous notons par  $\bar{R}_T$ , le nombre moyen de couleurs nécessaires pour colorier proprement les chemins sur  $T$  associés à toute permutation  $\sigma \in \mathcal{S}_n$  sur  $T$ , c'est-à-dire,  $\bar{R}_T = \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} R_T(\sigma)$ , où  $R_T(\sigma)$  est le nombre minimum de couleurs nécessaires pour colorier les chemins sur  $T$  associés à une permutation  $\sigma \in \mathcal{S}_n$ .

#### 5.3.1 Une borne inférieure générale

Soient  $G = (V, A)$  un graphe orienté symétrique à  $n$  sommets ( $|V| = n$ ) étiquetés de façon arbitraire, par des entiers dans l'ensemble  $\{1, 2, \dots, n\}$ , et  $r$  une fonction de routage dans  $G$  laquelle associe un ensemble de chemins sur  $G$  à chacune des permutations  $\sigma \in \mathcal{S}_n$  des sommets de  $G$ . Soit  $\bar{L}_{G,r}$  la congestion moyenne induite pour les chemins sur  $G$  associés, pour la fonction de routage  $r$ , à toutes les permutations dans  $\mathcal{S}_n$  des sommets de  $G$ , et soit  $U \subseteq V$  un sous-ensemble quelconque de l'ensemble de sommets  $V$  dans  $G$ . Nous notons par  $c(U)$  la coupe dans  $G$  induite

par  $U$ , définie par  $c(U) = \{(u, v) \in A : u \in U, v \in V \setminus U\}$ .

**Proposition 3** *Pour tout graphe  $G = (V, A)$  à  $n$  sommets, et pour toute fonction de routage  $r$  dans  $G$ ,*

$$\bar{L}_{G,r} \geq \frac{1}{n} \cdot \max_{U \subseteq V} \left( \frac{|U| \cdot (n - |U|)}{|c(U)|} \right).$$

**Preuve.** Soit  $U \subseteq V$  un sous-ensemble quelconque de sommets dans  $V$ , et considérons une permutation  $\sigma \in \mathcal{S}_n$  des sommets de  $G$ . Il est clair que la congestion  $L_r(U, \sigma)$  des arcs dans l'ensemble  $c(U)$  induite par les chemins associés, par la fonction de routage  $r$ , à la permutation  $\sigma$  vérifie  $L_r(U, \sigma) \geq |\{j \in U : \sigma(j) \notin U\}|$ . Ainsi, la congestion totale des arcs dans  $c(U)$  vérifie,  $L_r(U) \geq \sum_{\sigma \in \mathcal{S}_n} L_r(U, \sigma)$ . En effet,

pour tout sommet  $j \in U$  et pour tout sommet  $k \in V \setminus U$ , chaque permutation  $\sigma \in \mathcal{S}_n$  telle que  $\sigma(j) = k$ , contribue pour au moins une unité à la congestion totale des arcs dans  $c(U)$ . Ainsi, la congestion moyenne des arcs dans  $c(U)$  vérifie  $\bar{L}_r(U) \geq \frac{1}{n!} \sum_{j \in U} \sum_{k \in V \setminus U} |\{\sigma \in \mathcal{S}_n : \sigma(j) = k\}|$ . De plus, pour toute paire de sommets  $j$  et  $k$  dans  $G$ , ils existent  $(n-1)!$  permutations  $\sigma \in \mathcal{S}_n$  telles que  $\sigma(j) = k$ .

Donc,  $\bar{L}_r(U) \geq \frac{1}{n!} \sum_{j \in U} \sum_{k \in V \setminus U} (n-1)! = \frac{1}{n} \sum_{j \in U} \sum_{k \in V \setminus U} 1 = \frac{|U|(n-|U|)}{n}$ . Ainsi, pour tout

arc  $\alpha \in c(U)$ , on a que la congestion moyenne dans  $\alpha$  vérifie  $\bar{L}_r(\alpha) \geq \frac{\bar{L}_r(U)}{|c(U)|}$ . Ceci implique que,  $\bar{L}_{G,r} \geq \frac{1}{n} \cdot \max_{U \subseteq V} \left( \frac{|U| \cdot (n-|U|)}{|c(U)|} \right)$ .  $\square$

Notons par  $\mathcal{C}(G)$  le paramètre  $\max_{U \subseteq V} \left( \frac{|U| \cdot (|V| - |U|)}{|c(U)|} \right)$ . Il n'est pas difficile de voir que  $\mathcal{C}(G)$  est égal à  $\frac{1}{\mathcal{S}(G)}$ , où  $\mathcal{S}(G)$  représente le paramètre appelé en anglais *the sparsest cut* du graphe  $G$  (voir [50]). En fait,  $\mathcal{S}(G)$  est défini par  $\min_{U \subseteq V} \left( \frac{|c(U)|}{|U| \cdot (|V| - |U|)} \right)$ . Matula et Shahrokhi montrent dans [50] que le problème de calculer le paramètre  $\mathcal{S}(G)$  est NP-difficile, ce qui implique que calculer le paramètre  $\mathcal{C}(G)$  est aussi NP-difficile. Déterminer la meilleure borne inférieure possible dans la proposition 3 est donc aussi un problème NP-difficile. Cependant, pour toute constante  $k$ , si la *bissection-arc* du graphe  $G$ , c'est-à-dire, la cardinalité minimale de toute coupe  $c(U)$  dans  $G$  telle que  $|U| = \lfloor \frac{n}{2} \rfloor$ , est au plus égale à  $k$ , alors le paramètre  $\mathcal{C}(G)$  peut être déterminé en temps polynomial. En effet, il est clair que  $\mathcal{C}(G) \geq \frac{\lfloor \frac{n}{2} \rfloor \lfloor \frac{n}{2} \rfloor}{k}$ . Considérons qu'il existe  $V_1 \subset V(G)$  tel que  $\frac{|V_1|(n-|V_1|)}{|c(V_1)|} > \frac{\lfloor \frac{n}{2} \rfloor \lfloor \frac{n}{2} \rfloor}{k}$ . Alors, comme  $|V_1|(n-|V_1|) \leq \lfloor \frac{n}{2} \rfloor \lfloor \frac{n}{2} \rfloor$ , on a que  $|c(V_1)| < k$ . Ainsi, pour calculer  $\mathcal{C}(G)$  il suffit en considérer tous les sous-ensembles dans  $A(G)$  qui déconnectent  $G$  avec cardinalité maximale au plus égale à  $k-1$ , c'est-à-dire, un nombre polynomial de tels sous-ensembles. Par exemple, pour toute grille-2D  $M(2n, k)$  avec  $2n$  lignes et un nombre constante  $k$  de colonnes,  $\mathcal{C}(M(2n, k)) = k \cdot n^2$ , et pour tout anneau  $C_n$ ,  $\mathcal{C}(C_n) = \frac{\lfloor \frac{n}{2} \rfloor \lfloor \frac{n}{2} \rfloor}{2}$ .

Finalement, nous donnons dans cette section, une borne inférieure pour le nombre moyen de couleurs nécessaires pour colorier toute permutation  $\sigma \in \mathcal{S}_n$  sur tout arbre

à  $n$  sommets.

Soit  $T$  un arbre raciné orienté symétrique à  $n$  sommets. Pour tout sommet  $i$  dans  $T$ , avec  $1 \leq i \leq n$ , soient  $v_T(i) = |T(i)|/n$  et  $\tilde{v}_T(i) = \min(v_T(i), 1 - v_T(i))$ , où  $T(i)$  est le sous-arbre de  $T$  raciné au sommet  $i$ . Soit  $\tilde{v}_T = \max_i \tilde{v}_T(i)$ .

**Corollaire 6**  $\bar{L}_T \geq n\tilde{v}_T(1 - \tilde{v}_T)$ .

**Preuve.** Par la proposition 3, nous pouvons en déduire que la congestion moyenne de tout arc  $i$  dans  $T$  (i.e. l'arc dans  $T$  allant du sommet  $i$  vers le sommet père de  $i$ ), qu'on note par  $\bar{L}_T(i)$ , vérifie  $\bar{L}_T(i) = \frac{|T(i)|(n - |T(i)|)}{n}$ . Alors, il est clair que  $\max_i \{\bar{L}_T(i)\} = n\tilde{v}_T(1 - \tilde{v}_T)$ . De plus, comme  $\bar{L}_T \geq \max_i \{\bar{L}_T(i)\}$ , alors on a  $\bar{L}_T \geq n\tilde{v}_T(1 - \tilde{v}_T)$ .  $\square$

De plus, soit  $T$  un arbre orienté symétrique à  $n$  sommets. Comme  $\bar{R}_T \geq \bar{L}_T$ , alors on obtient la borne inférieure suivante pour le nombre moyen de couleurs nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets de  $T$ .

**Lemme 2**  $\bar{R}_T \geq n\tilde{v}_T(1 - \tilde{v}_T)$ .

### 5.3.2 Le nombre moyen de couleurs dans les arbres

Dans cette section, nous donnons un aperçu des résultats sur la complexité en moyenne du problème de la coloration de chemins associés à les permutations des sommets dans les arbres orientés symétriques, sans entrer dans les détails. Toutes les preuves formelles des résultats que nous présentons ici, peuvent être trouvées dans [17, 18].

Considérons tout d'abord le cas d'une chaîne. Dans ce réseau, le problème de coloration de chemins peut être résolu optimalement en temps polynomial [33], puisque le graphe de conflit induit par toute collection de chemins sur ce réseau est un graphe *d'intervalles* [30]. De plus, le nombre de couleurs nécessaires pour colorier toute collection de chemins sur la chaîne est égal à la valeur de la congestion induite pour une telle collection. Le principal résultat dans cette section est le suivant.

**Théorème 29** *Le nombre moyen de couleurs nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets d'une chaîne à  $n$  sommets est*

$$\frac{n}{4} + \frac{\lambda}{2}n^{1/3} + O(n^{1/6})$$

où  $\lambda = 0.99615\dots$

Pour prouver le théorème 29, nous avons utilisé des techniques issues de la combinatoire énumérative et asymptotique [14, 25, 68, 49, 19]. Tout d'abord, nous avons utilisé une bijection donnée par Biane [14] entre les permutations dans  $\mathcal{S}_n$  et certains objets combinatoires appelés *chemins de Motzkin*, qui sont des chemins spéciaux définis dans  $\mathbb{N} \times \mathbb{N}$ . Cette bijection est telle que la hauteur de tout chemin de Motzkin

est égale à la congestion induite par la permutation sur la chaîne associée à un tel chemin. Donc, grâce à cette bijection, nous avons pu obtenir la fonction génératrice des permutations qui ont besoin d'exactly  $k$  couleurs pour être coloriées, pour tout  $k$  donné. Ensuite, en utilisant des résultats de la combinatoire énumérative trouvés par Flajolet [25] et Viennot [68], nous en avons déduit un algorithme polynomial pour déterminer le nombre moyen de couleurs nécessaires pour colorier toutes les permutations dans  $\mathcal{S}_n$  sur une chaîne à  $n$  sommets, pour tout  $n$  fixé à l'avance. Finalement, nous avons combiné les résultats énumératifs précédents avec des techniques de marches aléatoires introduites par Louchard [49] et Daniels et Skyrme [19] pour prouver finalement le théorème 29.

Nous avons généralisé le théorème 29 dans le cas des arbres arbitraires, et nous avons obtenu le résultat suivant.

**Théorème 30** *Soit  $T$  un arbre orienté symétrique à  $n$  sommets. La congestion moyenne induite par les chemins associés à toutes les permutations  $\sigma \in \mathcal{S}_n$  des sommets de  $T$  est*

$$\bar{L}_T = n\tilde{v}_T(1 - \tilde{v}_T) + O(\sqrt{n}).$$

De plus, par le résultat de Erlebach et al. [23], nous savons qu'il existe un algorithme polynomial pour colorier toute collection de chemins avec une congestion  $L$  sur un arbre  $T$  qui utilise au plus  $\lceil \frac{5}{3}L \rceil$  couleurs. Ainsi, pour tout arbre  $T$  à  $n$  sommets, le nombre moyen de couleurs  $\bar{R}_T$  nécessaires pour colorier toute permutation  $\sigma \in \mathcal{S}_n$  sur  $T$  vérifie  $\bar{R}_T \leq \frac{5}{3}\bar{L}_T + 1$ . Plus formellement, par l'algorithme d'Erlebach et al. [23] et par le théorème 30, nous obtenons la borne supérieure suivante pour  $\bar{R}_T$ .

**Théorème 31** *Pour tout  $\epsilon$ , il existe  $n_0 = n_0(\epsilon)$  tel que, pour tout  $n \geq n_0$  et pour tout arbre  $T$  à  $n$  sommets, le nombre moyen de couleurs  $\bar{R}_T$  nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets de  $T$  vérifie  $\bar{R}_T \leq (\frac{5}{3} + \epsilon)n\tilde{v}_T(1 - \tilde{v}_T)$ .*

Soient  $k$  un entier fixé,  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$  une partition de l'entier  $n - 1$  en  $k$  parties et  $\text{GST}(\lambda)$  l'étoile généralisée orientée symétrique associée à  $n$  sommets. Dans ce cas, on a  $\tilde{v}_{\text{GST}(\lambda)} = \min(\lfloor n/2 \rfloor, \lambda_1)$ . De plus, Gargano et al. ont montré dans [29] que toute collection de chemins  $\mathcal{P}$  sur une étoile généralisée  $\text{GST}(\lambda)$  vérifie,  $R_{\text{GST}(\lambda)}(\mathcal{P}) = L_{\text{GST}(\lambda)}(\mathcal{P})$ . Ainsi, par le théorème 30, nous avons obtenu dans [18] le résultat suivant.

**Théorème 32** *Le nombre moyen de couleurs nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_n$  des sommets d'une étoile généralisée  $\text{GST}(\lambda)$  à  $n$  sommets est*

$$\bar{R}_{\text{GST}(\lambda)} = n\tilde{v}_{\text{GST}(\lambda)}(1 - \tilde{v}_{\text{GST}(\lambda)}) + O(\sqrt{n}).$$

Nous avons aussi obtenu dans [18] le résultat suivant. Soit  $k \geq 2$  un entier fixé à l'avance.

**Théorème 33** *Le nombre moyen de couleurs nécessaires pour colorier les chemins associés à toute permutation  $\sigma \in \mathcal{S}_{nk+1}$  des sommets d'une étoile généralisée  $\text{GST}(\lambda)$  à  $nk + 1$  sommets et à  $k$  branches, chacune de longueur  $n$ , est  $n(k - 1)/k + O(\sqrt{n})$ .*

## 5.4 Problèmes ouverts

Comme Caragiannis et al. l'ont remarqué dans [16], toute collection symétrique de chemins avec une congestion  $L$ , sur un arbre orienté symétrique peut être coloriée avec au plus  $\lfloor \frac{3}{2}L \rfloor$  couleurs, puisque ce problème se réduit au problème de coloration de chemins dans les arbres non orientés pour lequel Tarjan a donné dans [63] un algorithme polynomial  $\frac{3}{2}$ -approché. Ainsi, tout ensemble de chemins avec congestion  $L$  représentant une involution des sommets dans un arbre, peut être colorié en utilisant au plus  $\lfloor \frac{3}{2}L \rfloor$  couleurs. De plus, on sait par le résultat de Erlebach et al. [23], que  $\lceil \frac{5}{3}L \rceil$  couleurs sont suffisantes pour colorier toute collection de chemins avec une congestion  $L$  sur un arbre quelconque. Cependant, les problèmes suivants restent ouverts :

**Problème 2** *Existe-il un algorithme polynomial pour colorier tout ensemble de chemins avec congestion  $L$  représentant une permutation circulaire des sommets d'un arbre quelconque, tel qu'il utilise au plus  $\lceil \epsilon L \rceil$  couleurs, avec  $1 < \epsilon < \frac{5}{3}$  ?*

Si la réponse au problème 2 est affirmative, alors par la proposition 2, nous aurons trouvé un meilleur algorithme en temps polynomial pour colorier toute collection de chemins sur un arbre orienté symétrique avec une congestion  $L$ , en utilisant au plus  $\lceil \epsilon L \rceil + 2$  couleurs.

**Problème 3** *Quelle est la complexité algorithmique du problème de la coloration de permutations circulaires dans les arbres binaires et dans les arbres ayant au plus deux sommets de degré supérieure à deux ?*

Dans le cas de la complexité en moyenne du problème de routage par chemins arc-disjoints dans les réseaux quelconques, il reste beaucoup de travail à faire. En effet, nous ne savons pas si les techniques utilisées dans le cas des arbres peuvent être appliquées à d'autres classes de graphes. Il faut remarquer que les résultats obtenus dans ce chapitre sur la complexité en moyenne du routage de permutations dans les arbres, utilisent la propriété du rapport constant (c-à-d, inférieure à 2) entre le nombre chromatique des graphes de conflit induits par les ensembles de chemins représentant une permutation des sommets d'un arbre et le nombre de clique de ces graphes de conflit. De plus, il existe peu de résultats sur la non-approximabilité du problème du routage de permutations par chemins arc-disjoints dans les graphes orientés symétriques quelconques. Ceci implique que nous ne connaissons pas le rapport entre le nombre minimal de couleurs nécessaires pour colorier une permutation dans un réseau quelconque et la congestion induite par les chemins affectés à une telle permutation.

## 5.5 Conclusion

Nous résumons les principaux résultats trouvés dans ce chapitre dans les tables 5.1 et 5.2 suivantes.



Tout d'abord, nous avons montré dans le théorème 26 que dans les arbres, le problème de coloration d'une permutation est équivalent au problème général de coloration d'une collection quelconque de chemins. Soient  $T$  un arbre orienté symétrique à  $n$  sommets et  $\mathcal{P}$  un ensemble de chemins dans  $T$ . Alors nous avons obtenu les résultats de complexité suivants pour ce problème de coloration de chemins dans les arbres :

$\mathcal{P}$	$T$	complexité	référence
involution	binaire	NP-difficile	théorème 27(b)
permutation circulaire	degré maximum $\geq 4$	NP-difficile	théorème 27(c)
involution	ayant au moins 2 sommets de degré $> 2$	NP-difficile	théorème 27(d)

TABLE 5.1 – Résultats de complexité pire cas obtenus pour le problème de coloration de chemins dans les arbres.

Supposons que l'arbre  $T$  est un arbre raciné, que les sommets dans  $T$  sont étiquetés de façon arbitraire par les entiers dans  $\{1, 2, \dots, n\}$  et que la racine de  $T$  est le sommet étiqueté par l'entier  $n$ . Considérons  $\tilde{v}_T = \max_i \min(|T(i)|/n, 1 - |T(i)|/n)$  où  $T(i)$  est le sous-arbre de  $T$  enraciné au sommet  $i$ , avec  $1 \leq i \leq n$ . Alors nous avons obtenu les résultats suivants concernant le nombre moyen de couleurs nécessaires pour colorier toute permutation dans  $\mathcal{S}_n$  sur les arbres :

$T$	# moyen de couleurs	référence
chaîne	$\frac{n}{4} + o(n)$	théorème 29
quelconque	au plus $(\frac{5}{3} + \epsilon) n \tilde{v}_T (1 - \tilde{v}_T)$	théorème 31
étoile généralisée	$n \tilde{v}_T (1 - \tilde{v}_T) + O(\sqrt{n})$	théorème 32

TABLE 5.2 – Résultats obtenus pour le nombre moyen de couleurs nécessaires pour le problème de coloration de permutations dans les arbres.



## 6 Commutation généralisée de circuits

Considérons deux chemins  $p$  et  $q$  associés respectivement à deux requêtes de communication  $(s_p, t_p)$  et  $(s_q, t_q)$  dans un réseau. Supposons que les chemins  $p$  et  $q$  utilisent un même lien  $e$  du réseau, et supposons qu'on utilise le mode de commutation de circuits par chemins arc-disjoints pour effectuer le routage des données dans un tel réseau. Il est clair, comme on a vu dans les deux chapitres précédents, que ces deux requêtes de communication doivent être routées dans deux phases de communication distinctes, puisque  $p$  et  $q$  sont en conflit pour le lien  $e$ . Supposons maintenant que la longueur des chemins  $p$  et  $q$  est suffisamment grande, et le lien  $e$  est le premier (resp. le dernier) lien dans  $p$  (resp.  $q$ ). Si la quantité d'information qui doit transiter pour ces deux chemins n'est pas considérable, alors réserver deux phases de communication distinctes pour router les requêtes de communication  $(s_p, t_p)$  et  $(s_q, t_q)$  dans le réseau peut paraître une hypothèse pessimiste. En fait, on peut imaginer que quand les données qui transitent pour le chemin  $q$  arrivent au lien  $e$ , celles qui transitent pour le chemin  $p$  ont déjà traversé ce lien, et donc une seule phase de communication aurait été nécessaire pour router ces deux requêtes. Ainsi, dans ce chapitre, nous allons introduire un modèle théorique de routage dans un réseau de communication, qui permettra de paramétrer l'utilisation par diverses communications d'un même lien du réseau. Pour cela, nous allons généraliser la notion de conflit entre chemins de communication sur un réseau. Nous dirons que deux chemins sur un réseau sont en conflit s'ils partagent un même lien du réseau, lequel se trouve dans la  $i$ -ème position sur un des chemins et sur la  $j$ -ème position sur l'autre chemin, et si la valeur  $|i - j|$  est inférieure à un certain paramètre entier  $\rho$ . Nous appellerons ce paramètre  $\rho$  la *tolérance*. Ainsi, la tolérance représente une mesure (i.e. une échelle de temps) pour le partage des ressources d'un réseau. Ce type de mesure joue un rôle important en problèmes d'affectation d'horaires dans les services de transport publique, notamment dans les trains (voir par exemple, le projet européen AMORE "Algorithmic Methods for Optimizing the Railways in Europe", <http://www.inf.uni-konstanz.de/algo/amore/>). La tolérance peut aussi être choisie comme la durée maximale d'une communication. En fait, en utilisant des hypothèses similaires à celle de la tolérance, Becchetti et al. [8] donnent des algorithmes approchés pour un problème particulier du routage optique dans les chaînes et les anneaux, qui consiste à minimiser le temps de réalisation de certaines communications optiques ayant une durée variable.

Dans ce chapitre nous présentons des résultats de complexité sur le problème de routage dans les anneaux et dans les arbres sous ce mode de commutation généralisée de circuits. Nous formalisons tout d'abord dans la section 6.1 le modèle de commutation généralisée de circuits. Dans la section 6.2 nous étudions ce problème de routage dans les chaînes et dans les anneaux. Dans le cas des chaînes, nous montrons que ce problème de routage peut être calculé optimalement en temps polynomial pour toute valeur de la tolérance. Dans le cas des anneaux, nous obtenons tout d'abord un algorithme polynomial pour minimiser le nombre maximum de conflits des chemins associés à toute collection de requêtes de communication, avec une garantie de performance égale à la valeur optimale plus un. Ensuite, nous dérivons pour toute valeur de la tolérance, un algorithme 2-approché pour ce problème de routage dans les anneaux. Dans la section 6.3 nous montrons que pour toute valeur de la tolérance, le problème de routage dans les arbres binaires sous ce mode de commutation généralisée est NP-difficile. Nous montrons aussi qu'il ne peut pas exister un algorithme  $(4/3 - \epsilon)$ -approché pour ce problème dans les arbres binaires, pour tout  $\epsilon > 0$ , sauf si  $P = NP$ . Cependant, nous donnons un algorithme polynomial 2-approché pour ce problème de routage dans les arbres quelconques. Nous appliquons dans la section 6.4, les résultats de complexité obtenus dans la section 6.3 au problème de routage wormhole glouton dans les arbres binaires, et nous montrons que ce problème est aussi NP-difficile, même si la longueur (i.e. le nombre de flits) de tous les messages est unitaire. Dans la section 6.5 nous posons quelques problèmes ouverts et nous concluons ce chapitre dans la section 6.6. La plupart de ces résultats apparaissent dans [5].

## 6.1 Le modèle

Nous commençons cette section avec la définition suivante qui généralise la notion de conflit entre chemins sur un réseau de communication.

**Définition 4** Soient  $G$  un graphe orienté symétrique,  $p$  et  $q$  deux chemins sur  $G$ , et  $\rho$  un entier positif. Nous disons que les chemins  $p$  et  $q$  sont en  $\rho$ -conflit si et seulement s'il existe un arc  $e$  dans  $G$  tel que  $e \in p \cap q$ ,  $e$  étant le  $i$ -ème (resp. le  $j$ -ème) arc sur le chemin  $p$  (resp.  $q$ ), et  $|i - j| < \rho$ .

**Définition 5** Soient  $G$  un graphe orienté symétrique,  $\mathcal{P}$  une collection quelconque de chemins sur  $G$ , et  $\rho$  un entier positif. Nous définissons le graphe non orienté  $\mathcal{H}_{(G,\mathcal{P})}^\rho$  qu'on appelle graphe de  $\rho$ -conflit, associé à  $G$ ,  $\mathcal{P}$  et  $\rho$ , comme étant le graphe qui a pour ensemble de sommets les chemins dans  $\mathcal{P}$  et où deux sommets dans  $\mathcal{H}_{(G,\mathcal{P})}^\rho$  sont adjacents si et seulement si les deux chemins dans  $\mathcal{P}$  qu'ils représentent sont en  $\rho$ -conflit.

Soient  $G$  le graphe orienté symétrique qui modélise un réseau de communication donné,  $\mathcal{C} = \{(a_1, b_1), (a_2, b_2), \dots, (a_i, b_i)\}$  une collection de requêtes de communication dans  $G$ , et  $\rho$  un entier positif. Le problème du routage de  $\mathcal{C}$  dans  $G$  sous le mode commutation généralisée de circuits avec une tolérance  $\rho$ , que nous appellerons par la suite le mode CGC( $\rho$ ), consiste à trouver une collection de chemins  $\mathcal{P}$

sur  $G$  qui minimise le nombre de phases de communication nécessaires pour router  $\mathcal{C}$  dans  $G$ , de sorte que toute paire de chemins appartenant à une même phase de communication ne soient pas en  $\rho$ -conflit.

Il est clair que le mode de commutation de circuits par chemins arc-disjoints que nous avons étudié dans les chapitres précédents est un cas particulier du mode CGC( $\rho$ ). En effet, il suffit de prendre  $\rho$  égal au nombre de canaux (arcs) sur un réseau, et on obtient ainsi que toute paire de chemins sur un réseau qui partagent un même arc (lien) sont en  $\rho$ -conflit. De plus, le problème de routage dans le mode CGC( $\rho$ ) sur un réseau peut être aussi vu comme un problème de coloration de chemins, dans lequel nous cherchons une collection de chemins sur un réseau, associée à une collection de requêtes de communication, ayant une coloration minimale de sorte que toute paire de chemins qui sont en  $\rho$ -conflit soient affectés des couleurs distinctes.

Nous notons par  $R_G^\rho(\mathcal{C}, \mathcal{P})$  le nombre minimum de couleurs nécessaires pour colorier la collection de chemins  $\mathcal{P}$  sur  $G$  associé à la collection de requêtes de communication  $\mathcal{C}$ , tel que toute paire de chemins dans  $\mathcal{P}$  étant en  $\rho$ -conflit soient coloriés avec des couleurs distinctes, et on note par  $R_G^\rho(\mathcal{C})$  le minimum des valeurs  $R_G^\rho(\mathcal{C}, \mathcal{P})$ , pris sur toutes les collections des chemins  $\mathcal{P}$  sur  $G$  associés à  $\mathcal{C}$ . Clairement, si  $\mathcal{H}_{(G, \mathcal{P})}^\rho$  représente le graphe de  $\rho$ -conflit associé à  $G$ ,  $\mathcal{P}$  et  $\rho$ , alors  $R_G^\rho(\mathcal{C}, \mathcal{P}) = \chi(\mathcal{H}_{(G, \mathcal{P})}^\rho)$ , où  $\chi(\mathcal{H}_{(G, \mathcal{P})}^\rho)$  est le nombre chromatique du graphe  $\mathcal{H}_{(G, \mathcal{P})}^\rho$ .

Nous rappelons que dans le mode de communication par chemins arc-disjoints, la congestion induite par une collection de chemins sur un réseau est une borne inférieure du nombre de couleurs nécessaires pour colorier ceux-ci. Cependant, par sa définition dans le mode arc-disjoint, la congestion ne représente pas une borne inférieure pour le nombre de couleurs nécessaires pour colorier une collection de chemins dans le mode CGC( $\rho$ ), surtout si  $\rho$  est petit. En effet, considérons un arc  $e$  quelconque d'un réseau donné à  $m$  arcs et une collection de chemins  $\mathcal{P}_e$  sur un tel réseau traversant l'arc  $e$ . Pour toute paire d'entiers positifs  $i$  et  $\rho$ , avec  $1 \leq i, \rho \leq m$ , soit  $\mathcal{P}_e^i \subseteq \mathcal{P}_e$  la sous-collection de chemins dans  $\mathcal{P}_e$  ayant l'arc  $e$  dans la  $j$ -ème position, avec  $i \leq j < i + \rho$ . D'une part, par définition de  $\rho$ -conflit, toute paire de chemins dans  $\mathcal{P}_e^i$  sont en  $\rho$ -conflit. D'autre part, tous les chemins dans  $\mathcal{P}_e^i$  qui ont l'arc  $e$  dans la  $i$ -ème position ne sont pas en  $\rho$ -conflit avec ceux dans  $\mathcal{P}_e$  qui ont l'arc  $e$  dans la  $(i + \rho)$ -ème position. Ainsi, le nombre maximum de chemins dans  $\mathcal{P}_e$  qui sont deux à deux en  $\rho$ -conflit est égal au nombre maximum des chemins dans chaque sous-collection  $\mathcal{P}_e^i$ , pris sur tout  $i$ , avec  $1 \leq i \leq m$ . Plus formellement, nous définissons la  $\rho$ -congestion induite par une collection de chemins sur un réseau de communication, comme suit.

**Définition 6** Soient  $G = (V, A)$  un graphe orienté symétrique modélisant un réseau de communication,  $\mathcal{P}$  une collection de chemins sur  $G$  associée à une collection de requêtes de communication  $\mathcal{C}$  dans  $G$ , et  $\rho$  un entier positif. Pour tout chemin  $p \in \mathcal{P}$  et pour tout arc  $e$  dans  $p$ , soit  $\text{pos}(e, p)$  la position de l'arc  $e$  sur le chemin  $p$ . On

définit la  $\rho$ -congestion induite par  $\mathcal{P}$  sur les arcs dans  $G$  par :

$$L_G^\rho(\mathcal{C}, \mathcal{P}) = \max_{e \in A} \left( \max_{1 \leq i \leq |A|} |\{p \in \mathcal{P} : e \in p \text{ et } i \leq \text{pos}(e, p) < i + \rho\}| \right).$$

Nous noterons par  $L_G^\rho(\mathcal{C})$  le minimum des valeurs  $L_G^\rho(\mathcal{C}, \mathcal{P})$ , pris sur toutes les collections possibles des chemins  $\mathcal{P}$  sur  $G$  associées à une collection de requêtes de communication  $\mathcal{C}$ . Ainsi, on a que la  $\rho$ -congestion  $L_G^\rho(\mathcal{C})$  est une borne inférieure pour le paramètre  $R_G^\rho(\mathcal{C})$ , c'est-à-dire,  $R_G^\rho(\mathcal{C}) \geq L_G^\rho(\mathcal{C})$ .

## 6.2 Routage dans la chaîne et dans l'anneau

Le résultat principal que nous allons montrer dans cette section est le suivant.

**Théorème 34** *Soient  $C_n$  l'anneau orienté symétrique à  $n$  sommets,  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $C_n$ , et  $\rho$  un entier positif quelconque. Alors, en temps polynomial, on peut obtenir une collection  $\mathcal{P}$  de chemins sur  $C_n$  pour router  $\mathcal{C}$  sous le mode  $CGC(\rho)$  telle que  $L_{C_n}^\rho(\mathcal{C}, \mathcal{P}) \leq L_{C_n}^\rho(\mathcal{C}) + 1$ .*

Avant de prouver ce théorème, nous allons introduire quelques définitions et résultats préliminaires.

Soient  $P_n$  la chaîne orientée symétrique à  $n$  sommets,  $\mathcal{P}$  une collection quelconque de chemins sur  $P_n$ , et  $\rho$  un entier positif quelconque. Nous notons par  $\overleftarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  (resp.  $\overleftarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$ ) le graphe non orienté de  $\rho$ -conflit associé aux chemins  $p \in \mathcal{P}$  tels que  $p = s_i \rightsquigarrow t_i$  et  $s_i < t_i$  (resp.  $t_i < s_i$ ). De la même façon, nous notons  $\overrightarrow{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho$  (resp.  $\overleftarrow{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho$ ) le graphe non orienté de  $\rho$ -conflit associé aux chemins dans  $\mathcal{P}$  orientés dans le sens des aiguilles d'une montre (resp. orientés dans le sens opposé à celui des aiguilles d'une montre) sur l'anneau orienté symétrique  $C_n$  à  $n$  sommets. Il est facile de voir que pour ces deux types de réseaux, on a  $R_{P_n}^\rho(\mathcal{C}, \mathcal{P}) = \max(\chi(\overrightarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho), \chi(\overleftarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho))$  et  $R_{C_n}^\rho(\mathcal{C}, \mathcal{P}) = \max(\chi(\overrightarrow{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho), \chi(\overleftarrow{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho))$  pour tous  $\mathcal{C}$ ,  $\mathcal{P}$  et  $\rho$  donnés. Dans la figure 6.1 nous donnons un exemple de graphes de  $\rho$ -conflit  $\overrightarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  et  $\overleftarrow{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho$ . Il est connu (voir [22]) que dans le mode commutation de circuits par chemins arc-disjoints, les graphes de conflit associés à  $P_n$  et  $\mathcal{P}$  (resp.  $C_n$  et  $\mathcal{P}$ ) sont des graphes dits *d'intervalles* (resp. *arc-circulaires*) (voir [30] pour plus d'informations sur les propriétés de ces types spéciaux de graphes). De plus, tout sous-graphe induit d'un graphe d'intervalles est aussi un graphe d'intervalles. Cependant, un sous-graphe partiel d'un graphe d'intervalles n'est pas toujours un graphe d'intervalles. Dans la suite, nous montrerons que, pour toute valeur de  $\rho$ , avec  $\rho \geq 1$ , les graphes de  $\rho$ -conflit associés à  $P_n$  (resp.  $C_n$ ) et  $\mathcal{P}$  sont ils aussi des graphes d'intervalles (resp. arc-circulaires), même s'ils sont sous-graphes partiels de graphes d'intervalles (resp. de graphes arc-circulaires).

**Lemme 3** *Soient  $P_n$  la chaîne orientée symétrique à  $n$  sommets et  $\mathcal{P}$  une collection quelconque de chemins sur  $P_n$ . Pour tout entier positif  $\rho$ , les graphes non orientés de  $\rho$ -conflit  $\overrightarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  et  $\overleftarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  sont des graphes d'intervalles.*

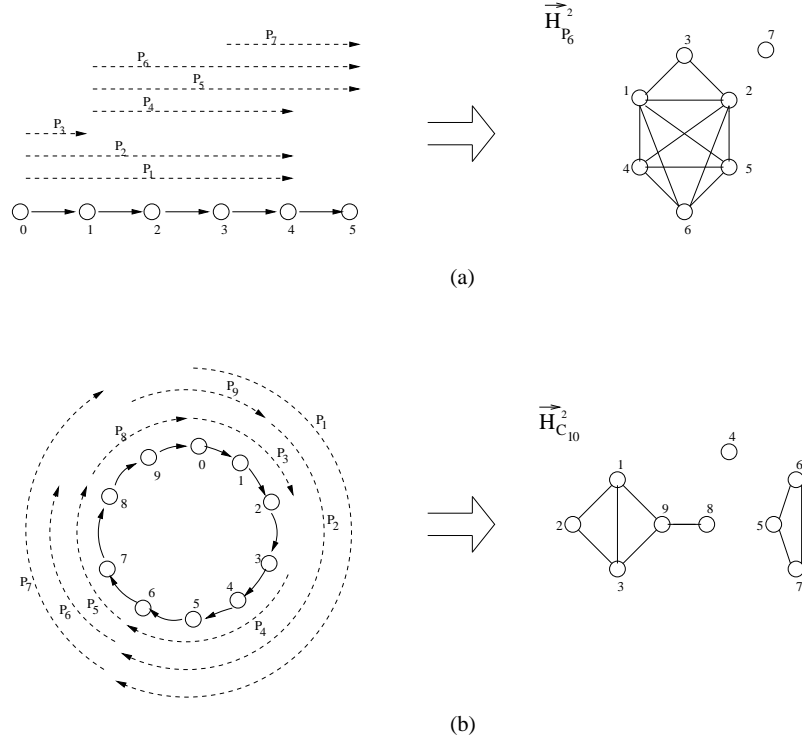


FIGURE 6.1 – (a) (resp. (b)) Exemple du graphe de  $\rho$ -conflit, avec  $\rho = 2$ , associé à une collection quelconque de chemins  $\mathcal{P}$  sur  $P_6$  (resp. sur  $C_{10}$ ).

**Preuve.** Soit  $\vec{\mathcal{P}} \subseteq \mathcal{P}$  le sous-ensemble de chemins sur  $P_n$  qui sont dirigés de gauche à droite, et qui constituent donc les sommets du graphe  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$ . Nous allons construire une collection d'intervalles semi-ouverts sur les nombres réels, qu'on note par  $\mathcal{I}$ , à partir des chemins dans  $\vec{\mathcal{P}}$  comme suit. Nous associons à chaque chemin  $p_i = a_i \rightsquigarrow b_i$  dans  $\vec{\mathcal{P}}$ , l'intervalle réel  $I_i = [a'_i, b'_i)$ , où  $a'_i = a_i$  et  $b'_i = a_i + \rho$  si  $b_i - a_i > \rho$ , ou  $b'_i = b_i$  dans le cas contraire. Soit  $\mathcal{G}(\mathcal{I})$  le graphe d'intervalles ayant pour ensemble de sommets  $\mathcal{I}$ , et où deux sommets dans  $\mathcal{G}(\mathcal{I})$  sont adjacents si et seulement si leurs intervalles associés s'intersectent. Alors, nous aurons prouvé ce lemme si nous prouvons la propriété suivante.

**Propriété 9** *Le graphe  $\mathcal{G}(\mathcal{I})$  est isomorphe au graphe  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$ .*

En effet, soient  $p_i = a_i \rightsquigarrow b_i$  un chemin dans  $\vec{\mathcal{P}}$  et  $I_i = [a'_i, b'_i)$  l'intervalle dans  $\mathcal{I}$  associé à  $p_i$  par la construction précédente. Soient  $v_i$  et  $v'_i$  les sommets dans les graphes  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  et  $\mathcal{G}(\mathcal{I})$  correspondant au chemin  $p_i$  et à l'intervalle  $I_i$  respectivement. Considérons maintenant l'application  $\phi : V(\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho) \rightarrow V(\mathcal{G}(\mathcal{I}))$  définie par  $\phi(v_i) = v'_i$  pour tout sommet  $v_i$  dans le graphe  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$ . Clairement, la propriété précédente est vraie si et seulement si  $\phi$  est un isomorphisme de  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  dans  $\mathcal{G}(\mathcal{I})$ , c'est-à-dire, les sommets  $v_i$  et  $v_j$  dans  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  sont adjacents si et seulement si les sommets  $\phi(v_i) = v'_i$  et  $\phi(v_j) = v'_j$  dans  $\mathcal{G}(\mathcal{I})$  sont adjacents. Notons que d'une

part, par définition, les chemins  $p_i = a_i \rightsquigarrow b_i$  et  $p_j = a_j \rightsquigarrow b_j$  sont en  $\rho$ -conflit si et seulement s'ils ont un arc en commun dans  $P_n$  et  $|a_j - a_i| < \rho$ . D'autre part, par construction, les intervalles  $I_i$  et  $I_j$  associés aux chemins  $p_i$  et  $p_j$ , sont tels que  $I_i = [a_i, a_i + \rho)$  (resp.  $I_j = [a_j, a_j + \rho)$ ) si la longueur du chemin  $p_i$  (resp.  $p_j$ ) est supérieure à  $\rho$ , ou bien  $I_i = [a_i, b_i)$  (resp.  $I_j = [a_j, b_j)$ ) sinon. Ainsi, si les chemins  $p_i$  et  $p_j$  sont en  $\rho$ -conflit, alors  $I_i \cap I_j \neq \emptyset$ . De plus, si les chemins  $p_i$  et  $p_j$  ont un arc en commun dans  $P_n$ , mais  $|a_j - a_i| \geq \rho$ , ce qui implique que ces deux chemins ne sont pas en  $\rho$ -conflit, alors  $I_i \cap I_j = \emptyset$ , parce que, par construction, la longueur des intervalles est au plus égale à  $\rho$ . Donc,  $\phi$  est un isomorphisme de  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  dans  $\mathcal{G}(\mathcal{I})$ , ce qui prouve la propriété, et donc prouve que le graphe  $\vec{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  est un graphe d'intervalles. De façon symétrique, nous pouvons montrer que le graphe  $\overleftarrow{\mathcal{H}}_{(P_n, \mathcal{P})}^\rho$  est aussi un graphe d'intervalles, ce qui termine la preuve de ce lemme.  $\square$

Il est connu que le problème de coloration des sommets d'un graphe d'intervalles  $G$  peut être résolu optimalement en temps polynomial [33]. De plus, comme les graphes d'intervalles sont des graphes parfaits [11, 30], alors  $\chi(G) = \omega(G)$ . Donc, le résultat suivant est une conséquence directe du lemme 3.

**Corollaire 7** *Soient  $P_n$  la chaîne orientée symétrique à  $n$  sommets,  $\mathcal{P}$  une collection quelconque de chemins sur  $P_n$ , et  $\rho$  un entier positif. Pour toute valeur de  $\rho \geq 1$ , le problème de calculer  $R_{P_n}^\rho(\mathcal{P})$  peut être résolu optimalement en temps polynomial. De plus,  $R_{P_n}^\rho(\mathcal{P}) = L_{P_n}^\rho(\mathcal{P})$ .*

En utilisant une construction similaire à celle du lemme précédent, nous obtenons le résultat suivant.

**Lemme 4** *Soient  $C_n$  l'anneau orienté symétrique à  $n$  sommets, et  $\mathcal{P}$  une collection quelconque de chemins sur  $C_n$ . Alors, pour tout entier positif  $\rho$ , les graphes de  $\rho$ -conflit  $\vec{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho$  et  $\overleftarrow{\mathcal{H}}_{(C_n, \mathcal{P})}^\rho$  sont des graphes arc-circulaires.*

En effet, les graphes arc-circulaires sont des graphes non orientés qui peuvent être représentés comme les graphes de conflit (ou d'intersection) associés à une collection de chemins sur un anneau non orienté (voir chapitre 4 pour plus d'informations sur les graphes arc-circulaires).

Donc, par les lemmes 3 et 4, étant donnés une collection quelconque de chemins  $\mathcal{P}$  sur la chaîne ou sur l'anneau (orientés symétriques), et un entier positif  $\rho$ , nous pouvons raccourcir les chemins dans  $\mathcal{P}$  qui ont une longueur supérieure à  $\rho$ , de sorte que tous les chemins dans  $\mathcal{P}$  aient une longueur au plus égale à  $\rho$ . Nous obtenons ainsi pour le problème de routage dans ces réseaux sous le mode CGC( $\rho$ ), des instances équivalentes aux instances initiales. Par les lemmes 3 et 4, nous avons donc le corollaire suivant.

**Corollaire 8** *Soient  $P_n$  (resp.  $C_n$ ) la chaîne (resp. l'anneau) orientée symétrique à  $n$  sommets,  $\mathcal{P}$  une collection quelconque de chemins sur  $P_n$  (resp.  $C_n$ ),  $\rho$  un entier positif quelconque, et  $\mathcal{P}'$  la collection de chemins raccourcis de longueur au plus égale*



à  $\rho$  obtenue par l'application de la transformation donnée dans la preuve du lemme 3 sur  $\mathcal{P}$ . Alors,  $L_{P_n}^\rho(\mathcal{P}) = L_{P_n}(\mathcal{P}')$  (resp.  $L_{C_n}^\rho(\mathcal{P}) = L_{C_n}(\mathcal{P}')$ ) et  $R_{P_n}^\rho(\mathcal{P}) = R_{P_n}(\mathcal{P}')$  (resp.  $R_{C_n}^\rho(\mathcal{P}) = R_{C_n}(\mathcal{P}')$ ).

Dans la figure 6.2, nous montrons un exemple de ce raccourcissement de chemins appliqué aux instances données dans la figure 6.1.

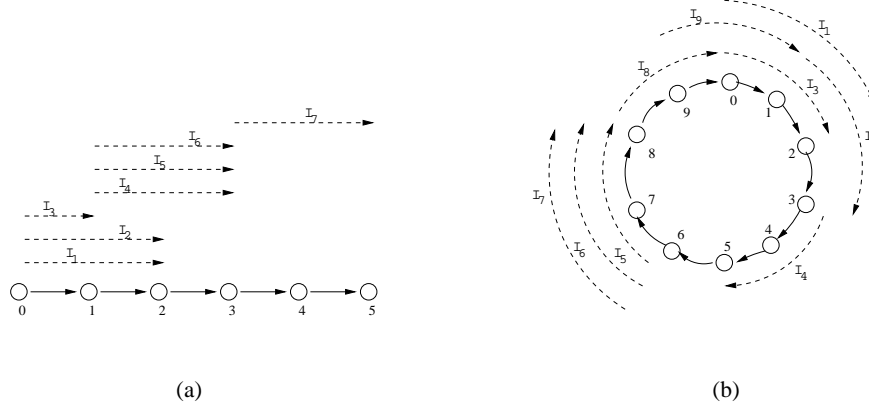


FIGURE 6.2 – (a) (resp. (b)) Application du raccourcissement des chemins fait par la construction donnée dans la preuve du lemme 3 (resp. 4) à l'instance montrée dans la figure 6.1.a (resp. 6.1.b), où  $\rho = 2$ .

### 6.2.1 Preuve du théorème 34

Comme nous l'avons mentionné dans le chapitre 3, Wilfong et Winkler ont montré dans [69] que, étant donné un réseau anneau  $C_n$  à  $n$  sommets, et une collection quelconque  $\mathcal{C}$  de requêtes de communication dans  $C_n$ , on peut trouver en temps polynomial une collection de chemins  $\mathcal{P}$  sur  $C_n$  pour router  $\mathcal{C}$  dans le mode de communication arc-disjoint, telle que  $L_{C_n}(\mathcal{C}) = L_{C_n}(\mathcal{C}, \mathcal{P})$ . Nous allons utiliser une approche similaire à celle de Wilfong et Winkler pour prouver le théorème 34. Malheureusement, pour notre problème d'optimisation de la  $\rho$ -congestion, nous ne pouvons pas exploiter certaines requêtes qui dans [69] sont appelés *requêtes parallèles*, et donc la valeur de la  $\rho$ -congestion induite par la collection de chemins trouvée, peut dépasser la valeur optimale d'au plus une unité.

Soient  $\mathcal{P}$  une collection de chemins sur  $C_n$ ,  $\vec{\mathcal{P}} \subseteq \mathcal{P}$  (resp.  $\overleftarrow{\mathcal{P}} \subseteq \mathcal{P}$ ) le sous-ensemble de chemins dans  $\mathcal{P}$  orientés dans le sens (resp. dans le sens opposé) des aiguilles d'une montre, et  $\rho$  un entier positif. Grâce au corollaire 8, nous pouvons redéfinir la  $\rho$ -congestion (voir déf. 6) induite par  $\mathcal{P}$  sur  $C_n$  comme :

$$L_{C_n}^\rho(\mathcal{P}) = \max(L_{C_n}^\rho(\vec{\mathcal{P}}), L_{C_n}^\rho(\overleftarrow{\mathcal{P}}))$$

et

$$L_{C_n}^\rho(\vec{\mathcal{P}}) = \max_{0 \leq i \leq n-1} |\{p_j = s_j \rightsquigarrow t_j : p_j \in \vec{\mathcal{P}} \text{ et } i \in [s_j, t_j] \text{ et } |s_j - i| < \rho\}|$$

$$L_{C_n}^\rho(\overleftarrow{\mathcal{P}}) = \max_{0 \leq i \leq n-1} |\{p_j = s_j \rightsquigarrow t_j : p_j \in \overleftarrow{\mathcal{P}} \text{ et } i \in [t_j, s_j] \text{ et } |s_j - i| < \rho\}|.$$

Donc, soient  $\mathcal{C} = \{(s_1, t_1), (s_2, t_2), \dots, (s_m, t_m)\}$ , avec  $s_i \neq t_i$ , une collection quelconque de requêtes de communication sur  $C_n$ , et  $\rho$  un entier positif. Le problème de trouver une collection de chemins  $\mathcal{P}$  sur  $C_n$  pour router  $\mathcal{C}$  telle que  $L_{C_n}^\rho(\mathcal{C}) = L_{C_n}^\rho(\mathcal{C}, \mathcal{P})$ , peut être formalisé comme un problème de programmation linéaire en nombres entiers (PLE) comme suit.

**Formalisation comme un problème PLE :** Soient  $n, m$ , et  $\rho$  trois entiers positifs, et soient  $(s_1, t_1), (s_2, t_2), \dots, (s_m, t_m)$  une collection de couples dans l'ensemble  $\{0, 1, \dots, n-1\}^2$ , avec  $s_i \neq t_i$ . Alors, il s'agit de trouver des valeurs  $x_1, x_2, \dots, x_m \in \{0, 1\}$  qui minimisent la fonction  $L^\rho$ , où

$$L^\rho = \max(\max_k A_k, \max_k B_k)$$

et

$$A_k = |\{i : |k - s_i| < \rho \text{ et } k \in [s_i, t_i - 1] \text{ et } x_i = 1\}|$$

$$B_k = |\{i : |k - s_i| < \rho \text{ et } k \in [t_i, s_i - 1] \text{ et } x_i = 0\}|$$

En effet, si la variable binaire  $x_i$ ,  $1 \leq i \leq m$ , vaut 1, alors le chemin  $s_i \rightsquigarrow t_i$  sur  $C_n$  par où la requête de communication  $(s_i, t_i)$  sera routée, sera dirigé dans le sens des aiguilles d'une montre. Sinon, le chemin  $s_i \rightsquigarrow t_i$  sera dirigé dans le sens opposé à celui des aiguilles d'une montre. Ainsi, pour tout  $k$ ,  $0 \leq k \leq n-1$ ,  $A_k$  représente la valeur de la  $\rho$ -congestion de l'arc  $(k, k+1)$  de  $C_n$ , et  $B_k$  représente la valeur de la  $\rho$ -congestion de l'arc  $(k+1, k)$  de  $C_n$ , où toutes les opérations arithmétiques sont considérées modulo  $n$ .

Suivant les idées de Wilfong et Winkler [69], on fera tout d'abord une relaxation du problème PLE précédent en un problème de programmation linéaire non entière (PL), et après l'obtention d'une solution optimale pour le problème relaxé PL, nous allons arrondir cette solution réelle afin d'obtenir une solution entière, que sera une solution approchée pour le problème PLE.

**Relaxation comme un problème PL :** Soient  $n, m$ , et  $\rho$  trois entiers positifs, et soient  $(s_1, t_1), (s_2, t_2), \dots, (s_m, t_m)$  une collection de couples dans l'ensemble  $\{0, 1, \dots, n-1\}^2$ , avec  $s_i \neq t_i$ . Alors, il s'agit de trouver des valeurs réelles  $x_1, x_2, \dots, x_m \in [0, 1]$  qui minimisent la fonction  $L_*^\rho$ , où

$$L_*^\rho = \max(\max_k A_k^*, \max_k B_k^*)$$

et

$$A_k^* = \sum_{\substack{i : |k-s_i| < \rho \\ \text{et} \\ k \in [s_i, t_i-1]}} x_i ; \quad B_k^* = \sum_{\substack{i : |k-s_i| < \rho \\ \text{et} \\ k \in [t_i, s_i-1]}} (1 - x_i)$$

Clairement, le problème PL précédent peut être résolu optimalement en temps polynomial [41], et la  $\rho$ -congestion optimale obtenue  $L_{*OPT}^\rho$  vérifie  $[L_{*OPT}^\rho] \leq L_{OPT}^\rho$ ,

où  $L_{OPT}^\rho$  représente la valeur de la  $\rho$ -congestion optimale pour le problème PLE.

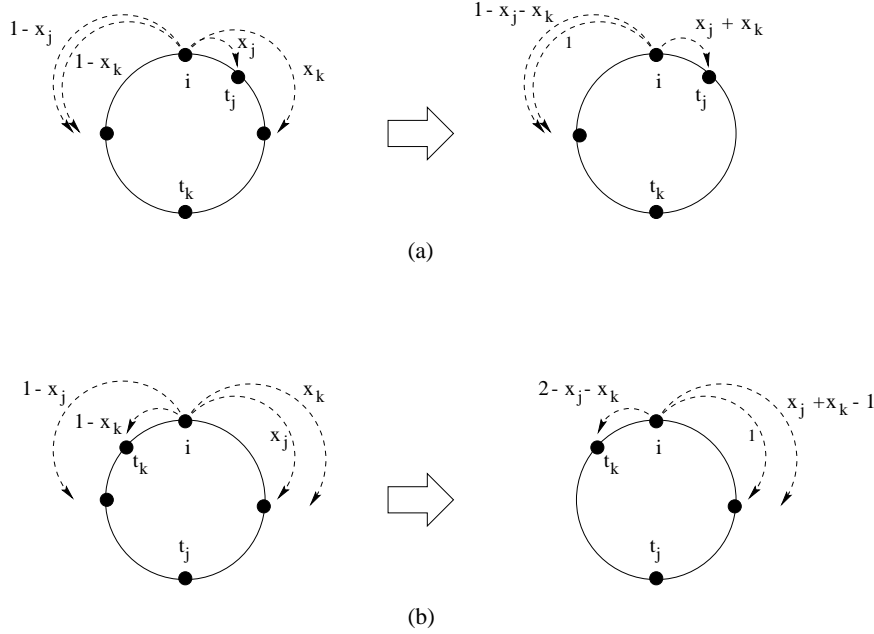
Nous dirons qu'une solution réelle  $x'$  au problème PL *cas*se la  $i$ -ème requête  $(s_i, t_i)$  si  $0 < x'_i < 1$ . De plus, si une solution réelle  $x'$  casse la  $i$ -ème requête  $(s_i, t_i)$ , alors nous associons à celle-ci, deux chemins  $\vec{p}_i$  et  $\overleftarrow{p}_i$ , tel que  $\vec{p}_i$  (resp.  $\overleftarrow{p}_i$ ) est le chemin orienté dans le sens (resp. dans le sens opposé) des aiguilles d'une montre allant du sommet  $s_i$  vers le sommet  $t_i$ , qui contribue par une valeur égale à  $x'_i$  (resp.  $1 - x'_i$ ) à la  $\rho$ -congestion des arcs dans  $C_n$  qu'il traverse. Par le corollaire 8, nous pouvons raccourcir les chemins associés à chacune des requêtes  $(s_i, t_i)$  par une solution réelle  $x'$ , de sorte que la longueur de tous ces chemins soit au plus égale à  $\rho$ , obtenant ainsi une instance équivalente. Par la suite, nous supposons qu'on a obtenu une solution réelle  $x'$  comme solution au problème PL, et que nous avons appliqué la technique de raccourcissement des chemins (voir la preuve du lemme 3) aux chemins induits par  $x'$  de sorte que la longueur de ceux-ci soit au plus égale à  $\rho$ . Ainsi, nous dirons que  $x'$  est une solution  $\rho$ -raccourcie.

**Propriété 10** *Étant donnée une instance du problème PL et une solution  $\rho$ -raccourcie  $x$  avec une  $\rho$ -congestion égale à  $L$ , on peut obtenir en temps polynomial, une solution  $\rho$ -raccourcie  $x'$  avec une  $\rho$ -congestion  $L' \leq L$ , telle que sur chaque sommet  $i$  de  $C_n$ , il existe au plus une requête  $(i, j)$  cassée par  $x'$ .*

*Preuve.* Soient  $(s_j, t_j)$  et  $(s_k, t_k)$  deux requêtes de communication cassées pour la solution  $x$  telles que  $s_j = s_k = i$ , pour un sommet  $i$  dans  $C_n$ . On a donc  $0 < x_j < 1$  et  $0 < x_k < 1$ . Nous allons montrer qu'on peut arrondir une des valeurs  $x_j$  ou  $x_k$  de sorte qu'une seule des deux requêtes soit cassée, et tel que la contribution de leurs chemins associés à la  $\rho$ -congestion des arcs dans  $C_n$  qu'ils traversent soit inférieure ou égale à la précédente. Nous rappelons que  $x$  est une solution  $\rho$ -raccourcie et donc, la longueur des chemins associés aux requêtes (cassées ou non) est au plus égale à  $\rho$ . Soient  $\vec{p}_j$  et  $\overleftarrow{p}_j$  (resp.  $\vec{p}_k$  et  $\overleftarrow{p}_k$ ) la paire de chemins associés à la requête  $(s_j, t_j)$  (resp.  $(s_k, t_k)$ ) par  $x$ . Nous supposons sans perte de généralité, que  $(s_j, t_j)$  est celle dont le chemin  $\vec{p}_j$  a une longueur inférieure à celle du chemin  $\vec{p}_k$ . Sinon,  $(s_j, t_j)$  est celle dont le chemin  $\overleftarrow{p}_j$  a une longueur supérieure ou égale à celle du chemin  $\overleftarrow{p}_k$ .

Ainsi, supposons d'abord que  $x_j \leq 1 - x_k$ . Dans ce cas, nous définissons la nouvelle solution  $x'$  par :  $x'_j = x_j + x_k$  et  $x'_k = 0$ , avec  $x'_q = x_q$  pour tout  $q \notin \{j, k\}$ . Dans la figure 6.3(a) nous montrons un exemple de cette transformation. Il est facile de voir que la  $\rho$ -congestion des arcs dans  $C_n$  traversés par les chemins  $\vec{p}_j$  et  $\overleftarrow{p}_k$  est exactement la même qu'avant, tandis que les autres arcs dans  $C_n$  ont une  $\rho$ -congestion inférieure ou égale à la précédente.

Si  $x_j > 1 - x_k$ , alors nous définissons la nouvelle solution  $x'$  par :  $x'_j = 1$  et  $x'_k = x_j + x_k - 1$ , avec  $x'_q = x_q$  pour tout  $q \notin \{j, k\}$ , qui a un effet sur la  $\rho$ -congestion des arcs dans  $C_n$  similaire au cas précédent. Dans la figure 6.3(b) nous montrons un exemple de cette transformation. En appliquant les transformations précédentes à chaque paire de requêtes cassées ayant le même sommet  $i$  comme sommet source, pour tout sommet  $i$  dans  $C_n$ , nous montrons la propriété 10.

FIGURE 6.3 – Exemples de la nouvelle solution  $x'$  obtenue pour la propriété 10.

Dans ce qui suit, nous allons supposer sans perte de généralité, que nous avons obtenu en temps polynomial une solution réelle  $x'$   $\rho$ -raccourcie qui vérifie la propriété 10. Maintenant, nous allons arrondir les requêtes cassées par  $x'$  qui restent. Nous obtenons ainsi une solution entière  $x$  avec une  $\rho$ -congestion  $L$  inférieure ou égale à  $L_{*OPT}^\rho + \frac{3}{2}$ , où  $L_{*OPT}^\rho$  est la  $\rho$ -congestion induite par  $x'$ .

Nous allons supposer que les requêtes cassées par  $x'$  sont ordonnées dans l'ordre croissant de leur sommet source dans  $C_n$ . On peut remarquer qu'un tel ordre sur les requêtes cassées est total puisque  $x'$  vérifie la propriété 10 et donc, on a pour chaque sommet de  $C_n$ , au plus une seule requête cassée dont ce sommet est l'origine. Pour tout sommet  $k$  dans  $C_n$ ,  $0 \leq k \leq n-1$ , nous notons par  $\vec{I}_k = \{(s_i, t_i) : 0 < x'_i < 1 \text{ et } k+1 \in \vec{p}_i\}$ , l'ensemble des requêtes cassées telles que leurs chemins associés orientés dans le sens des aiguilles d'une montre traversent l'arc  $(k, k+1)$  de  $C_n$ . De la même façon, nous notons par  $\overleftarrow{I}_k = \{(s_i, t_i) : 0 < x'_i < 1 \text{ et } k \in \overleftarrow{p}_i\}$ , l'ensemble des requêtes cassées telles que leurs chemins associés orientés dans le sens opposé à celui des aiguilles d'une montre traversent l'arc  $(k+1, k)$  de  $C_n$ . Il est clair que c'est l'arrondi des requêtes cassées qui affecte la  $\rho$ -congestion des arcs dans  $C_n$ . Supposons qu'on a une solution réelle  $x'$   $\rho$ -raccourcie qui vérifie la propriété 10, avec  $\rho$ -congestion des arcs égale à  $A'_k$  et  $B'_k$ , et que  $x$  est une  $\{0, 1\}$ -solution entière, avec  $\rho$ -congestion des arcs  $A_k$  et  $B_k$ , obtenue à partir d'un arrondi quelconque de la solution  $x'$ . Alors, on a

$$A_k = A'_k + \sum_{(s_i, t_i) \in \vec{I}_k} (x_i - x'_i)$$

et

$$B_k = B'_k + \sum_{(s_i, t_i) \in \overleftarrow{I}_k} (x'_i - x_i)$$

Maintenant, en considérant les requêtes cassées séquentiellement, en suivant l'ordre croissant défini sur celles-ci, nous allons définir une solution arrondie  $x$  entière de façon récursive comme suit : soit  $(s_j, t_j)$  la requête cassée considérée, alors

$$x_j = \begin{cases} 1 & \text{si } -x'_j + \sum_{(s_i, t_i) \in \overrightarrow{I}_{s_j}} (x_i - x'_i) < -\frac{1}{2} \\ 0 & \text{sinon} \end{cases}$$

Comme  $x'$  est une solution  $\rho$ -raccourcie, alors il est clair que par l'arrondi précédent, pour toute valeur de  $k$ , avec  $\rho - 1 \leq k \leq n - 1$ , on a

$$A_k - A'_k = \sum_{(s_i, t_i) \in \overrightarrow{I}_k} (x_i - x'_i) \leq \frac{1}{2}$$

Cependant, pour  $0 \leq k \leq \rho - 2$ , on a

$$A_k - A'_k = \sum_{(s_i, t_i) \in \overrightarrow{I}_k} (x_i - x'_i) \leq \frac{3}{2}$$

En effet, les chemins  $\overrightarrow{p}_j$  associés à toutes les requêtes cassées  $(s_j, t_j)$  telles que  $n - \rho \leq s_j \leq n - 1$  et  $1 \leq t_j \leq \rho - 1$  qui sont arrondies par  $x$  dans le sens des aiguilles d'une montre (c-à-d,  $x_j = 1$ ) affectent la  $\rho$ -congestion des arcs  $(k, k + 1)$  de  $C_n$ , avec  $0 \leq k \leq \rho - 2$ . Cependant, les chemins associés à ces requêtes sont tels que tous (sauf éventuellement un ayant pour sommet source le sommet  $n - 1$ ) traversent l'arc  $(n - 2, n - 1)$  de  $C_n$ , affectant ainsi la  $\rho$ -congestion de cet arc et donc, des arcs  $(k, k + 1)$ , par une valeur inférieure ou égale à  $\frac{1}{2}$ .

De plus, il peut exister un chemin associé à une requête cassée ayant pour sommet source le sommet  $n - 1$ , et qui par l'arrondi précédent peut aussi affecter la  $\rho$ -congestion des arcs  $(k, k + 1)$  de  $C_n$ , d'une valeur inférieure ou égale à  $\frac{1}{2}$ . Comme la  $\rho$ -congestion des arcs  $(k, k + 1)$  de  $C_n$ , pouvait déjà être augmentée par  $x$  d'une valeur inférieure ou égale à  $\frac{1}{2}$ , alors, en total, on a que la  $\rho$ -congestion des arcs  $(k, k + 1)$ , avec  $0 \leq k \leq \rho - 2$ , peut être augmentée par la solution arrondie  $x$  entière d'une valeur inférieure ou égale à  $\frac{3}{2}$ . Ainsi, on a que pour tout  $k$ ,  $0 \leq k \leq n - 1$ , la solution  $x$  est telle que  $A_k - A'_k \leq 3/2$ , et par un raisonnement symétrique, on peut déduire que  $B_k - B'_k \leq 3/2$ .

Donc, si  $L$  est la valeur de la  $\rho$ -congestion induite par  $x$ , alors  $L \leq L_{*OPT}^\rho + \frac{3}{2}$ , puisque la  $\rho$ -congestion induite par  $x'$  est optimale. Comme  $L_{OPT}^\rho \geq \lceil L_{*OPT}^\rho \rceil$  et comme  $L$  est entière, alors on a que  $L \leq L_{OPT}^\rho + 1$ , ce qui termine la preuve du théorème 34.  $\square$

Considérons une instance du problème PL précédente, et une solution réelle  $\rho$ -raccourcie  $x'$  vérifiant la propriété 10 de la preuve du théorème 34. Soient  $(s_{i_1}, t_{i_1})$ ,

$(s_{i_2}, t_{i_2}), \dots, (s_{i_r}, t_{i_r})$  l'ensemble de requêtes qui restent cassées par  $x'$ . Alors, le résultat suivant est une conséquence directe du théorème 34.

**Corollaire 9** *Si les chemins  $\overrightarrow{p_{i_j}}$  et  $\overrightarrow{p_{i_k}}$  (resp.  $\overleftarrow{p_{i_j}}$  et  $\overleftarrow{p_{i_k}}$ ) associés à toute paire de requêtes cassées  $(s_{i_j}, t_{i_j})$  et  $(s_{i_k}, t_{i_k})$  respectivement, avec  $1 \leq j \neq k \leq r$ , ne s'intersectent pas, alors, la solution entière  $x$  obtenue à partir de  $x'$  dans la preuve du théorème 34 a une  $\rho$ -congestion optimale, pour tout  $\rho \geq 1$ .*

**Proposition 4** *Soient  $C_n$  un anneau orienté symétrique,  $\mathcal{C} = \{(s_1, t_1), (s_2, t_2), \dots, (s_m, t_m)\}$  une collection de requêtes de communication dans  $C_n$ , et  $\rho$  un entier positif quelconque. Si pour tout  $i$ , avec  $1 \leq i \leq m$ , on a  $|t_i - s_i| \geq \rho$  et  $n - |t_i - s_i| \geq \rho$ , alors en temps polynomial, on peut obtenir une collection  $\mathcal{P}$  de chemins sur  $C_n$  pour router  $\mathcal{C}$  sous le mode  $CGC(\rho)$  de sorte que  $L_{C_n}^\rho(\mathcal{C}, \mathcal{P}) = L_{C_n}^\rho(\mathcal{C})$ .*

**Preuve.** En nous basant sur les idées de Wilfong et Winkler [69], pour toute instance du problème PL nous pouvons obtenir en temps polynomial, une solution réelle  $x'$  flux (une solution  $x$  est dite flux si  $\sum_{i=1}^m x_i$  est entière [69])  $\rho$ -raccourcie qui vérifie la propriété 10, telle qu'elle induite une  $\rho$ -congestion réelle optimale. En utilisant la même technique d'arrondi et une analyse similaire que dans [69], on obtient le résultat.  $\square$

## 6.2.2 Coloration des requêtes dans l'anneau

Grâce au corollaire 8, nous pouvons utiliser ici tous les résultats obtenus dans le chapitre 4 relatifs à la coloration des chemins dans l'anneau. Ainsi, nous donnons les résultats suivants pour le problème de routage dans les anneaux sous le mode de commutation  $CGC(\rho)$ , avec  $\rho \geq 1$ .

**Théorème 35** *Soient  $C_n$  un anneau orienté symétrique,  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $C_n$ , et  $\rho$  un entier positif quelconque. Alors, il existe un algorithme en temps polynomial  $\mathcal{A}$  pour router  $\mathcal{C}$  dans  $C_n$  sous le mode  $CGC(\rho)$  tel que :*

- (i) *si  $1 \leq \rho \leq \frac{n}{k}$ , pour tout  $k \geq 3$ , alors  $\mathcal{A}$  utilise au plus  $\lfloor \left(\frac{k}{k-1}\right) L_{C_n}^\rho(\mathcal{C}) \rfloor + 2$  phases de communication pour router  $\mathcal{C}$ .*
- (ii) *si  $\rho > \frac{n}{3}$ , alors  $\mathcal{A}$  utilise au plus  $2L_{C_n}^\rho(\mathcal{C}) + 1$  phases de communication pour router  $\mathcal{C}$ .*

**Preuve.** Par l'application du théorème 34 et du corollaire 2 (resp. 4) (voir chapitre 4), nous montrons (i) (resp. (ii)).  $\square$

## 6.3 Routage dans les arbres

Dans cette section, nous analysons la complexité du problème de routage dans les arbres sous le mode  $CGC(\rho)$ , avec  $\rho \geq 1$ . Le principal résultat de cette section est le suivant.

**Théorème 36** *Soient  $T$  un arbre binaire orienté symétrique,  $\mathcal{P}$  une collection quelconque de chemins sur  $T$  et  $\rho$  un entier positif. Alors, pour toute valeur de  $\rho \geq 1$ , le problème de calculer  $R_T^\rho(\mathcal{P})$  est NP-difficile.*

Avant de prouver le théorème 36, nous avons besoin quelques définitions et résultats préliminaires.

Pour toute paire d'entiers positifs  $n$  et  $\rho$ , nous allons construire un type spécial d'arbre binaire enraciné, que nous noterons par  $T_\rho^n$ . Soient  $n$  et  $\rho$  deux entiers positifs, et soient  $M = \lceil \log_2 n \rceil$  et  $h(i) = \lceil \frac{n}{2^{M-i}} \rceil$ ,  $1 \leq i \leq M$ , et  $h(0) = 1$ . Alors, l'arbre binaire enraciné  $T_\rho^n$  est construit comme suit (voir figure 6.4) :

1. D'abord, on considère  $M + 1$  ensembles ordonnés disjoints de sommets qu'on note par  $\text{CL}(i)$ ,  $0 \leq i \leq M$ , où chaque ensemble ordonné  $\text{CL}(i)$  est composé de  $h(i)$  sommets différents. Formellement,  $\text{CL}(i) = \{v_1^i, v_2^i, \dots, v_{h(i)}^i\}$ , avec  $0 \leq i \leq M$ . On définit l'unique sommet  $v_1^0 \in \text{CL}(0)$  comme la racine de l'arbre  $T_\rho^n$ , et les  $n$  sommets  $\{v_1^M, v_2^M, \dots, v_n^M\} \in \text{CL}(M)$  comme ses feuilles.
2. En suite, pour chaque sommet  $v_j^i \in \text{CL}(i)$  tel que  $j$  est pair,  $1 \leq i \leq M$ , on construit une nouvelle chaîne à  $\rho 2^{M-i+1}$  sommets, qu'on note par  $\mathcal{L}_j^i$ . On note le sommet initial et le sommet final dans  $\mathcal{L}_j^i$  par  $c_j^i$  et  $\bar{c}_j^i$  respectivement, et on connecte le sommet  $\bar{c}_j^i$  au sommet  $v_j^i$ .
3. Finalement, pour chaque  $i$ ,  $1 \leq i \leq M$ , et pour chaque  $j$ ,  $1 \leq j \leq h(i)$ , on connecte le sommet  $v_j^{i-1} \in \text{CL}(i-1)$  aux sommets  $v_{2j-1}^i \in \text{CL}(i)$  et  $c_{2j}^i \in \mathcal{L}_{2j}^i$  (s'il existe une telle chaîne  $\mathcal{L}_{2j}^i$ ).

Dans la figure 6.4, nous donnons un exemple de l'arbre binaire enraciné  $T_\rho^n$ , où  $n = 6$  et  $\rho = 1$ , avec les ensembles de sommets  $\text{CL}(i)$ ,  $0 \leq i \leq M = 3$ .

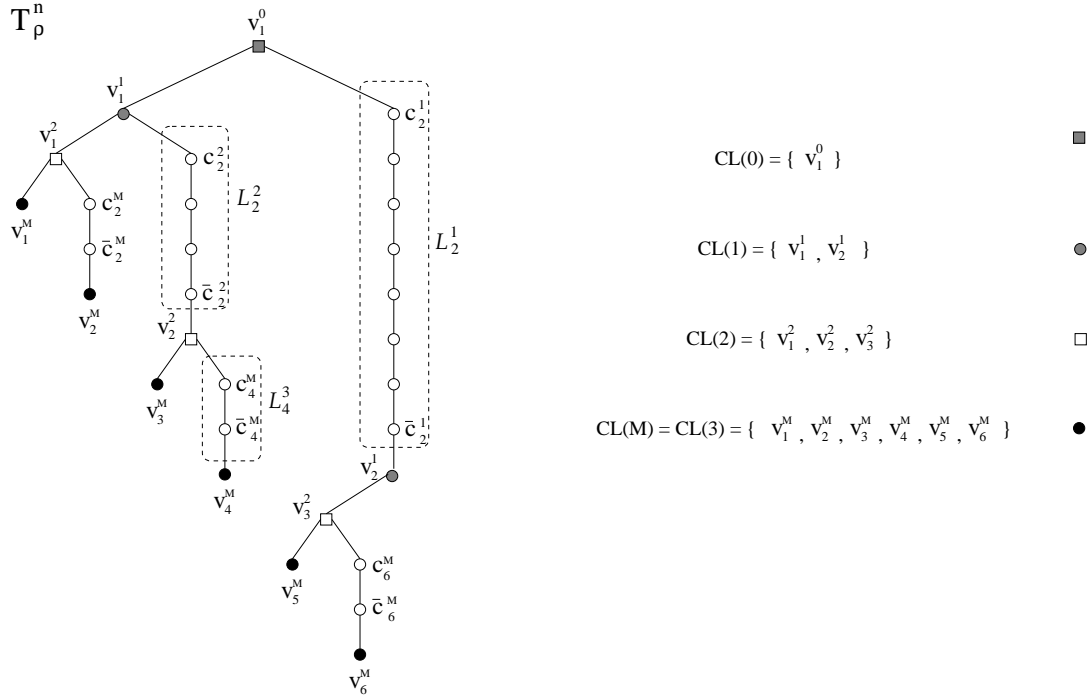
Pour toute paire d'entiers positifs  $n$  et  $\rho$ , l'arbre binaire enraciné  $T_\rho^n$  a des propriétés intéressantes que nous allons résumer dans les deux lemmes suivants.

**Lemme 5** *Soit  $v_j^i$  un sommet dans l'arbre binaire enraciné  $T_\rho^n$  tel que  $v_j^i \in \text{CL}(i)$ , avec  $0 \leq i \leq M$  et  $1 \leq j \leq h(i)$ . Alors, la hauteur du sommet  $v_j^i$ , c'est à dire, le nombre d'arcs sur l'unique chemin du sommet  $v_j^i$  au sommet racine  $v_1^0$ , qu'on note par  $N(v_j^i)$ , est  $N(v_j^i) = i + \rho(j-1)2^{M-i+1}$ .*

**Preuve.** Pour prouver ce lemme, nous montrerons que la hauteur d'un sommet  $v_j^i \in \text{CL}(i)$  dans l'arbre binaire  $T_\rho^n$  enracinée au sommet  $v_1^0 \in \text{CL}(0)$ , vérifie la relation de récurrence suivante :

$$\begin{cases} N(v_1^i) = i \\ N(v_j^i) = N(v_{j-1}^i) + \rho 2^{M-i+1}, \text{ si } j \geq 2 \end{cases}$$

Par construction de  $T_\rho^n$ , la relation de récurrence est vraie pour  $j = 1$  et pour tout  $j$  pair. Donc, il est uniquement nécessaire de montrer que cette relation de récurrence est vraie pour tout  $j$  impair,  $j \geq 3$ . Soit  $j$  un entier positif impair,  $j \geq 3$ , et soit  $j-1 = 2^{\alpha_1} + 2^{\alpha_2} + \dots + 2^{\alpha_{p(j-1)}}$ , avec  $\alpha_1 < \alpha_2 < \dots < \alpha_{p(j-1)}$ , où  $p(j-1)$  est le nombre de "1" dans la représentation binaire du nombre pair  $j-1$ . Soit  $v_c^{m-1} \in \text{CL}(m-1)$  le sommet racine du sous arbre minimal dans  $T_\rho^n$  contenant les sommets  $v_{j-1}^i$  et  $v_j^i$ ,

FIGURE 6.4 – L'arbre binaire enraciné  $T_\rho^n$ , avec  $n = 6$  et  $\rho = 1$ .

avec  $0 \leq m - 1 < i$  (voir figure 6.5). Soient  $v_a^m$  et  $v_{a+1}^m$  deux sommets consécutifs dans le sous-ensemble ordonné  $CL(m)$  tels que  $v_a^m$  est le fils gauche du sommet  $v_c^{m-1}$ . Par construction de  $T_\rho^n$ , l'entier positif  $a$  est impair et les sommets  $v_c^{m-1}$  et  $v_{a+1}^m$  sont connectés par une chaîne de longueur  $\rho 2^{M-m+1} + 1$  (voir figure 6.5). En plus, par construction de  $T_\rho^n$ , comme  $(j - 1)$  est pair et égal à  $2^{\alpha_1} + 2^{\alpha_2} + \dots + 2^{\alpha_{p(j-1)}}$ , avec  $\alpha_1 < \alpha_2 < \dots < \alpha_{p(j-1)}$ , alors la valeur de  $a$  est égale à  $\frac{j-1}{2^{\alpha_1}}$ , et les sommets  $v_{a+1}^m$  et  $v_j^i$  sont connectés par une chaîne de longueur  $\alpha_1$  comme il est montré dans la figure 6.5.

Les relations suivantes peuvent être facilement obtenues à partir de la construction de l'arbre binaire  $T_\rho^n$ .

- (r<sub>1</sub>)  $m = i - \alpha_1$
- (r<sub>2</sub>)  $N(v_{j-1}^i) = N(v_a^m) + \sum_{r=1}^{\alpha_1} (\rho 2^{M-(i-\alpha_1+r)+1} + 1)$
- (r<sub>3</sub>)  $N(v_{a+1}^m) = N(v_a^m) + \rho 2^{M-m+1}$
- (r<sub>4</sub>)  $N(v_j^i) = N(v_{a+1}^m) + \alpha_1$

En calculant la somme de la partie droite dans l'équation (r<sub>2</sub>), on obtient

$$(r_5) \quad N(v_{j-1}^i) = N(v_a^m) + \rho 2^{M-m+1} - \rho 2^{M-i+1} + \alpha_1,$$

et finalement, en remplaçant (r<sub>3</sub>) et (r<sub>5</sub>) dans (r<sub>4</sub>), on obtient que  $N(v_j^i) = N(v_{j-1}^i) + \rho 2^{M-i+1}$ , ce qui termine la preuve de ce lemme.  $\square$



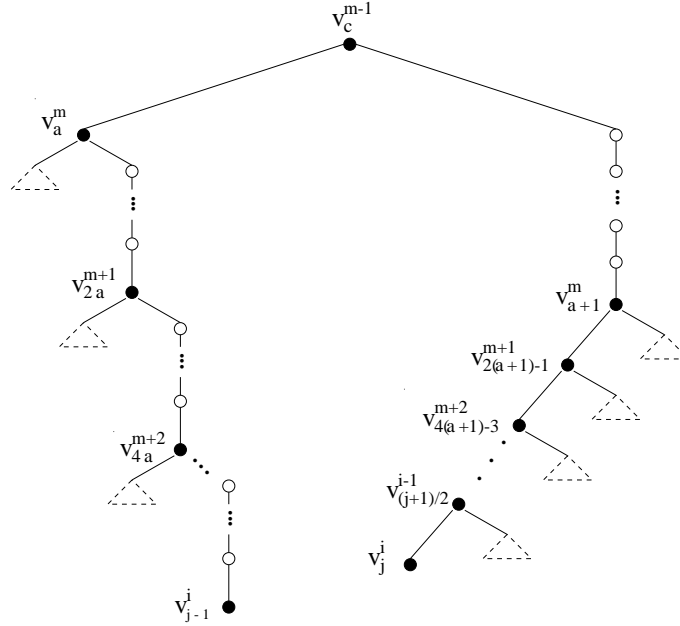


FIGURE 6.5 – Sous arbre minimal dans  $T_\rho^n$  contenant les sommets  $v_{j-1}^i$  et  $v_j^i$  dans  $CL(i)$ .

**Lemme 6** Soient  $v_l^i \rightsquigarrow u$  et  $v_j^i \rightsquigarrow w$ , avec  $l \neq j$ , deux chemins sur l'arbre binaire  $T_\rho^n$ , tels que les sommets initiaux  $v_l^i$  et  $v_j^i$  respectivement, appartiennent au même sous-ensemble ordonné de sommets  $CL(i)$ , avec  $1 \leq i \leq M = \lceil \log_2 n \rceil$ , et tels que ses sommets finaux  $u$  et  $w$  respectivement, sont deux sommets quelconques dans  $T_\rho^n$ . Alors, ces deux chemins ne sont pas en  $\rho$ -conflit.

**Preuve.** Soit  $v_a^m \in CL(m)$  le sommet racine du sous-arbre minimal dans  $T_\rho^n$  contenant les sommets  $v_l^i$  et  $v_j^i$ , et soit  $v_c^q \in CL(q)$  le sommet initial du premier arc possible dans  $T_\rho^n$  partagé par les chemins  $v_l^i \rightsquigarrow u$  et  $v_j^i \rightsquigarrow w$ . Alors, par construction de  $T_\rho^n$  (voir figure 6.4), il est suffisant de considérer les deux cas suivants :

- *Cas 1* :  $v_c^q = v_a^m$  (voir figure 6.6(a)). Par construction de  $T_\rho^n$ , il est clair que  $N(v_a^m) < N(v_j^i)$  et  $N(v_a^m) < N(v_l^i)$ . Soient  $\Delta_1 = N(v_l^i) - N(v_a^m)$  et  $\Delta_2 = N(v_j^i) - N(v_a^m)$ . Supposons que les chemins  $v_l^i \rightsquigarrow u$  et  $v_j^i \rightsquigarrow w$  sont en  $\rho$ -conflit, et supposons aussi sans perte de généralité que  $j > l$ . L'hypothèse de  $\rho$ -conflit entre ces deux chemins implique que  $\Delta_2 - \Delta_1 < \rho$ . En utilisant le lemme 5, l'inégalité précédente peut s'exprimer comme  $\rho(j-l)2^{M-i+1} < \rho$ . Cependant, par construction de  $T_\rho^n$ , on a que  $j-l \geq 1$  et  $2^{M-i+1} \geq 2$ , et donc on a que  $\rho(j-l)2^{M-i+1} > \rho$ , ce qui est une contradiction à l'hypothèse de départ.

- *Cas 2* :  $v_c^q \neq v_a^m$  (voir figure 6.6(b)). Nous supposons sans perte de généralité que  $j > l$ . Par construction de  $T_\rho^n$ , on a que  $i > q > m$ ,  $N(v_a^m) < N(v_l^i)$ ,  $N(v_a^m) < N(v_c^q)$ , et  $N(v_c^q) < N(v_j^i)$ . Soient  $\Delta_1 = N(v_l^i) - N(v_a^m)$ ,  $\Delta_2 = N(v_c^q) - N(v_a^m)$ , et  $\Delta_3 = N(v_j^i) - N(v_c^q)$ . Supposons que les chemins  $v_l^i \rightsquigarrow u$  et  $v_j^i \rightsquigarrow w$  sont en  $\rho$ -conflit. L'hypothèse de  $\rho$ -conflit entre ces deux chemins implique que  $\Delta_1 + \Delta_2 - \Delta_3 < \rho$ . De

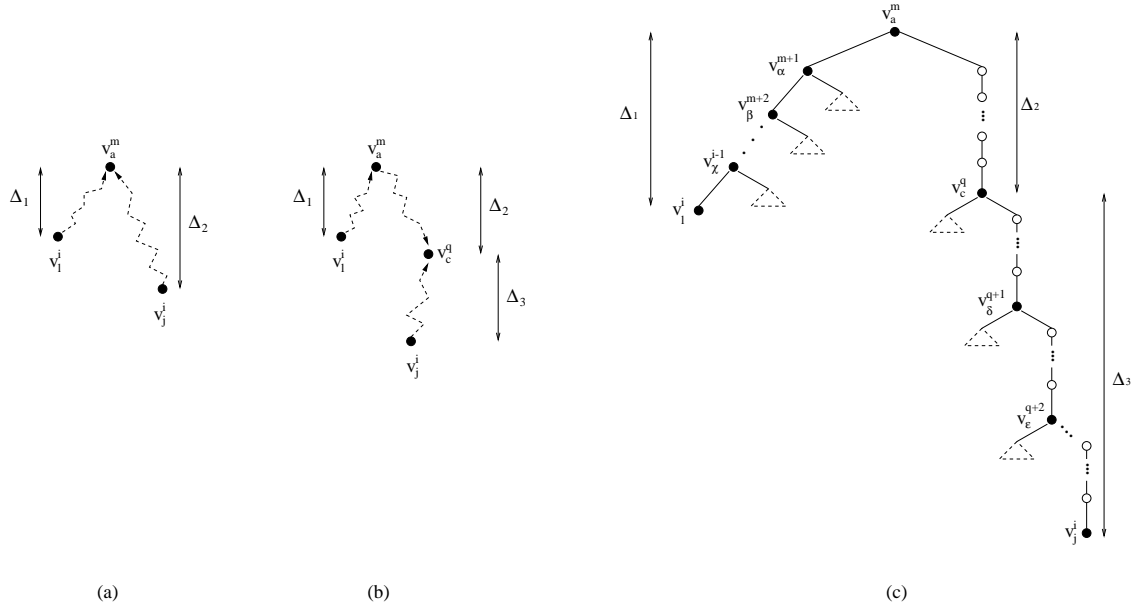


FIGURE 6.6 – (a) (resp. (b)) Cas 1 (resp. 2) du lemme 6. (c) La pire configuration possible pour le cas 2 du lemme 6.

plus, par construction de  $T_\rho^n$  et par le lemme 5, il est facile à vérifier les inégalités suivantes (voir figure 6.6(c)) :

$$\begin{aligned}
 (r1) \quad & \Delta_1 \geq i - m \\
 (r2) \quad & \Delta_2 \geq \rho 2^{M-q+1} + 1 \\
 (r3) \quad & \Delta_3 \leq \sum_{r=1}^{i-q} (\rho 2^{M-(q+r)+1} + 1) = \rho 2^{M-q+1} - \rho 2^{M-i+1} + i - q
 \end{aligned}$$

En utilisant les inégalités (r1), (r2) et (r3), on a que  $\Delta_1 + \Delta_2 - \Delta_3 \geq q - m + 1 + \rho 2^{M-i+1}$ . Cependant, par construction de  $T_\rho^n$ , on a que  $q - m + 1 > 1$  et  $2^{M-i+1} \geq 2$ , ce qui implique que  $\Delta_1 + \Delta_2 - \Delta_3 > \rho$ , ce qui est une contradiction à l'hypothèse de départ. Ainsi, ce lemme est prouvé.  $\square$

### 6.3.1 Preuve du théorème 36

Pour prouver ce théorème, nous utiliserons une réduction polynomiale du problème *ARÊTE-COLORATION* d'un graphe non orienté [15], qui à été montré par Holyer comme étant un problème NP-complet dans [36], même si le graphe considéré est un graphe *cubique*, c'est-à-dire, un graphe régulier de degré 3. Le problème *ARÊTE-COLORATION* d'un graphe cubique consiste à savoir si, étant donné un graphe cubique non orienté  $G$ , on peut obtenir une arête-coloration propre de  $G$  avec trois couleurs. Soit  $I$  une instance du problème *ARÊTE-COLORATION* d'un graphe cubique donnée par un graphe cubique  $G = (V, E)$  non orienté. Nous allons construire, à partir de  $I$ , une instance  $I'$  du problème de routage dans les arbres binaires sous le mode  $\text{CGC}(\rho)$  donnée par un arbre binaire orienté symétrique  $T$  et une collection

de chemins  $\mathcal{P}$  sur  $T$  tel que pour tout entier  $\rho \geq 1$ ,  $R_T^\rho(\mathcal{P}) = 3$  si et seulement si  $G$  admet une arête-coloration propre avec 3 couleurs. La construction de l'instance  $I'$  est faite en trois parties (voir figures 6.7 et 6.8) :

• **Partie I :** Soient  $|V| = n$  et  $M = \lceil \log_2 n \rceil$ . D'abord, nous construisons l'arbre binaire enraciné orienté symétrique  $T_\rho^n$ . Nous considérons que le sous-ensemble ordonné de sommets  $\text{CL}(M) = (v_1^M, v_2^M, \dots, v_n^M)$  dans  $T_\rho^n$  (par construction,  $\text{CL}(M)$  est l'ensemble des feuilles dans  $T_\rho^n$ ) représente l'ensemble de sommets  $V$  dans  $G$ . En fait, soit  $V = \{v_1, v_2, \dots, v_n\}$ , alors le sommet  $v_j^M \in \text{CL}(M)$  représente le sommet  $v_j \in V$ ,  $1 \leq j \leq n$ . Par définition, le sommet  $v_1^0 \in \text{CL}(0)$  est la racine de l'arbre  $T_\rho^n$ .

• **Partie II :** Ensuite, nous transformons l'arbre binaire enraciné  $T_\rho^n$  en un arbre binaire  $T$  comme suit : on remplace chacune des feuilles  $v_j^M \in \text{CL}(M)$  dans  $T_\rho^n$ ,  $1 \leq j \leq n$ , par un arbre binaire orienté symétrique, isomorphe à celui montré dans la figure 6.7, enraciné au sommet  $v_j^M$ . Les 9 feuilles de chaque sous-arbre binaire enraciné dans chacun des sommets  $v_j^M$  seront notées par  $v_{j,1}^M, v_{j,2,1}^M, v_{j,2,2}^M, v_{j,3}^M, v_{j,4,1}^M, v_{j,4,2}^M, v_{j,5}^M, v_{j,6,1}^M$ , et  $v_{j,6,2}^M$  comme il est montré dans la figure 6.7.

De plus, pour chaque arête  $e = \{v_i, v_j\}$  dans le graphe  $G$ , nous ajoutons à  $\mathcal{P}$  (initialement vide) deux chemins sur  $T$  qu'on note par  $p_1^e = v_{i,m}^M \rightsquigarrow v_{j,l,2}^M$  et  $p_2^e = v_{j,l-1}^M \rightsquigarrow v_{i,m+1,2}^M$ , où  $m$  (resp.  $l$ ) est un entier dans l'ensemble  $\{1, 3, 5\}$  (resp.  $\{2, 4, 6\}$ ) pas encore utilisé par aucun chemin associé à une autre arête incident au sommet  $v_i$  (resp.  $v_j$ ) dans  $G$ . Dans la figure 6.8, nous donnons un exemple de ces deux chemins correspondants à l'arête  $e = \{v_1, v_4\}$  d'un graphe cubique  $G$ , où les valeurs de  $m$  et  $l$  sont égaux à 3 et 2 respectivement.

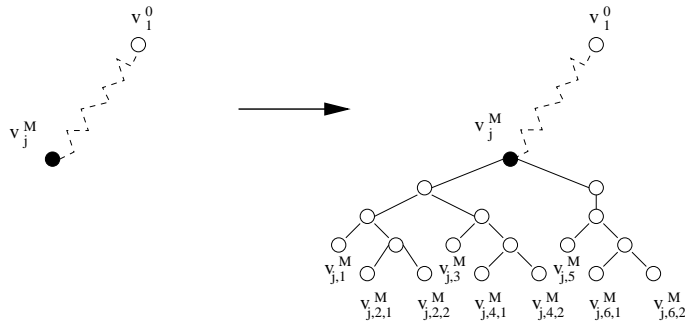
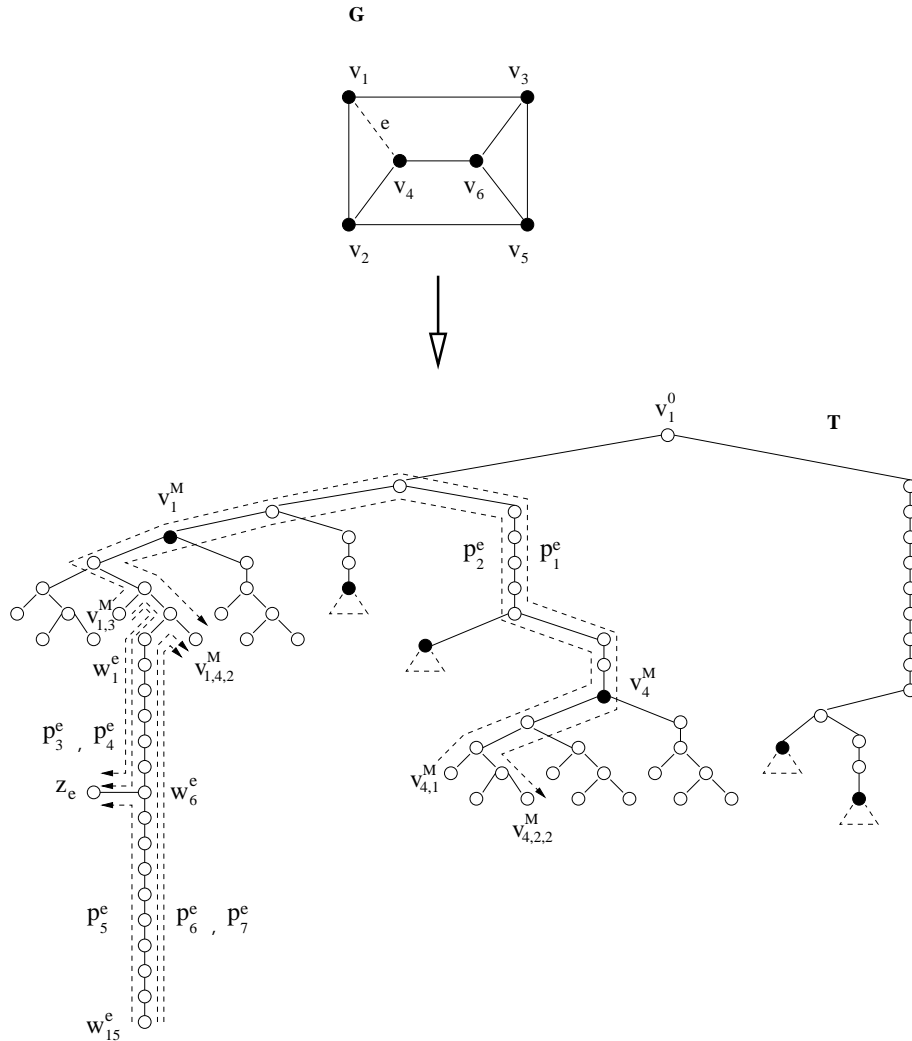


FIGURE 6.7 – Sous-arbre binaire isomorphe enraciné dans chacun des sommets  $v_j^M \in \text{CL}(M)$  construit dans la partie II.

**Partie III :** Finalement, nous devons assurer que les deux chemins  $p_1^e$  et  $p_2^e$  associés à chacune des arêtes  $e$  dans le graphe cubique  $G$  sont affectés la même couleur dans toute 3-coloration propre de  $\mathcal{P}$ . Pour cela, pour chaque arête  $e = \{v_i, v_j\}$  du graphe  $G$ , on ajoute une chaîne dans  $T$  de la manière suivante : soient  $p_1^e = v_{i,m}^M \rightsquigarrow v_{j,l,2}^M$  et  $p_2^e = v_{j,l-1}^M \rightsquigarrow v_{i,m+1,2}^M$  les deux chemins dans  $\mathcal{P}$  associés à l'arête  $e$  dans  $G$ , où  $m$  et  $l$  sont deux entiers dans l'ensemble  $\{1, 3, 5\}$  et  $\{2, 4, 6\}$  respectivement. Par construction de l'arbre courant  $T$  dans la partie II, il est facile de vérifier que les

FIGURE 6.8 – Construction partielle de l'instance  $I'$ .

longueurs des chemins  $p_1^e$  et  $p_2^e$ , sont égales à un entier impair  $d_e$  qui vérifie  $d_e \geq 2\rho + 9$ . Dans la figure 6.8, nous avons associé à l'arête  $e = \{v_1, v_4\}$  du graphe cubique  $G$  les chemins  $p_1^e = v_{1,3}^M \rightsquigarrow v_{4,2,2}^M$  et  $p_2^e = v_{4,1}^M \rightsquigarrow v_{1,4,2}^M$  sur l'arbre  $T$ , dont la longueur  $d_e$  de ces deux chemins est 17.

Soient  $l_e = d_e - 3$  et  $n_e = (l_e/2) - 1$ . Ainsi, nous construisons une nouvelle chaîne de longueur  $l_e$ , et on note ses sommets par  $w_1^e, w_2^e, \dots, w_{l_e+1}^e$ , où les sommets  $w_1^e$  et  $w_{l_e+1}^e$  sont les sommets initial et final respectivement d'une telle chaîne (voir figure 6.8). En suite, on connecte le sommet  $w_1^e$  dans la nouvelle chaîne au sommet  $v_{i,m+1,1}^M$  dans l'arbre courant  $T$ . On ajoute un nouveau sommet à l'arbre courant  $T$  qu'on note par  $z_e$ , et on connecte le sommet  $z_e$  au sommet étiqueté  $w_{n_e}^e$  dans cette nouvelle chaîne. Finalement, on ajoute à la collection courante de chemins  $\mathcal{P}$ , cinq nouveaux chemins associés à l'arête  $e$  dans  $G$  qui sont :  $p_3^e = p_4^e = v_{i,m}^M \rightsquigarrow z_e$ ,  $p_5^e = w_{l_e+1}^e \rightsquigarrow z_e$ , et  $p_6^e = p_7^e = w_{l_e+1}^e \rightsquigarrow v_{i,m+1,2}^M$ . En fait, les cinq chemins  $p_r^e$ ,  $3 \leq r \leq 7$ , vont assurer que les deux chemins  $p_1^e$  et  $p_2^e$  se voient affecter la même couleur dans toute 3-coloration propre de  $\mathcal{P}$  comme on va le voir dans ce qui suit. Dans la figure 6.8 on

présente un exemple partiel de cette construction à partir d'un graphe cubique  $G$  à 6 sommets, et où  $\rho = 1$ .

Maintenant, nous allons montrer qu'il existe une arête-coloration propre de  $G$  avec 3 couleurs si et seulement si  $R_T^\rho(\mathcal{P}) = 3$ . Tout d'abord, supposons qu'il existe une arête-coloration propre de  $G$  avec 3 couleurs. Alors, une 3-coloration propre de  $\mathcal{P}$  est obtenue de la façon suivante. Soit  $c_e \in \{1, 2, 3\}$  la couleur affectée à l'arête  $e$  dans  $G$ , alors on affecte la couleur  $c_e$  aux chemins  $p_1^e$  et  $p_2^e$ . Comme toute paire de chemins dans chacun des ensembles  $\{p_1^e, p_3^e, p_4^e\}$ ,  $\{p_2^e, p_6^e, p_7^e\}$ ,  $\{p_5^e, p_6^e, p_7^e\}$ , et  $\{p_3^e, p_4^e, p_5^e\}$  sont en  $\rho$ -conflit, alors les chemins  $p_r^e$ , avec  $3 \leq r \leq 7$ , peuvent être coloriés de la façon suivante. On affecte au chemin  $p_5^e$  la couleur  $c_e$ , et on affecte aux chemins  $p_3^e, p_4^e, p_6^e$  et  $p_7^e$  les deux couleurs encore disponibles différentes de la couleur  $c_e$ . Par construction, toute paire de chemins qui ont par sommet initial les feuilles  $v_{i,m}^M$  (resp.  $v_{j,n}^M$ ), avec  $m$  (resp.  $n$ ) dans l'ensemble  $\{1, 3, 5\}$ , sont en  $\rho$ -conflit. Cependant, par le lemme 6, ces chemins ne sont en  $\rho$ -conflit avec aucun autre chemin dans  $\mathcal{P}$  ayant comme sommet initial un sommet feuille  $v_{a,r}^M$ , avec  $a \neq i$  (resp.  $a \neq j$ ). Donc, la coloration précédente des chemins dans  $\mathcal{P}$  est une 3-coloration propre.

Supposons maintenant qu'il existe une 3-coloration propre de  $\mathcal{P}$ , c'est-à-dire,  $R_T^\rho(\mathcal{P}) = 3$ . Par construction de l'instance  $I'$ , dans toute 3-coloration propre de  $\mathcal{P}$ , les chemins  $p_1^e$  et  $p_2^e$  associés à l'arête  $e = \{v_i, v_j\}$  dans  $G$  sont affectés la même couleur. Ceci est fait par les cinq chemins  $p_r^e$ , avec  $3 \leq r \leq 7$ . Par la construction de l'instance  $I'$ , toute paire de chemins  $p_1^f$  et  $p_2^f$  associée à une arête  $f$  dans  $G$  adjacent à l'arête  $e$ , doivent être affectés une couleur différente à celle assignée aux chemins  $p_1^e$  et  $p_2^e$ , puis qu'ils sont en  $\rho$ -conflit. Cependant, si les arêtes  $e$  et  $f$  ne sont pas adjacents dans  $G$ , alors par le lemme 6, aucune paire de chemins dans l'ensemble  $\{p_1^e, p_2^e, p_1^f, p_2^f\}$  ne sont en  $\rho$ -conflit. Ainsi, si on affecte à chaque arête  $e$  dans  $G$  la couleur affectée aux chemins  $p_1^e$  et  $p_2^e$  dans  $\mathcal{P}$ , nous obtenons une arête-coloration propre de  $G$  avec 3 couleurs, ce qui termine la preuve du théorème.  $\square$

Le résultat suivant est une conséquence directe du théorème 36.

**Corollaire 10** *Soient  $T$  un arbre orienté symétrique,  $\mathcal{P}$  une collection de chemins sur  $T$ , et  $\rho$  un entier positif. Alors, il ne peut pas exister d'algorithme polynomial  $(4/3 - \epsilon)$ -approché, pour tout  $\epsilon > 0$ , pour calculer  $R_T^\rho(\mathcal{P})$ , même si  $T$  est un arbre binaire, à moins que  $P = NP$ .*

En effet, il est facile de voir que s'il existe un algorithme polynomial  $(4/3 - \epsilon)$ -approché, pour tout  $\epsilon > 0$ , pour résoudre le problème de routage dans les arbres binaires sous le mode CGC( $\rho$ ), pour tout  $\rho \geq 1$ , alors  $P = NP$ , puisque le même algorithme pourrait déterminer en temps polynomial si un graphe cubique admet ou pas une arête-coloration propre avec trois couleurs, ce qui est, d'après Holyer [36], un problème NP-complet.

### 6.3.2 Un algorithme 2-approché pour les arbres

Soient  $T$  un arbre orienté symétrique,  $\mathcal{P}$  une collection de chemins sur  $T$  représentant une collection de requêtes de communication dans  $T$ , et  $\rho$  un entier positif. Dans le théorème 36 nous avons prouvé que, pour tout  $\rho \geq 1$ , le problème de calculer  $R_T^\rho(\mathcal{P})$  est NP-difficile même si  $T$  est un arbre binaire. De plus, par le corollaire 10, nous savons qu'il ne peut pas exister un algorithme polynomial  $(4/3 - \epsilon)$ -approché, pour tout  $\epsilon > 0$ , pour résoudre ce problème, à moins que  $P = NP$ . Cependant, suivant les idées de Ranade et al. [56], nous donnons dans cette section un algorithme polynomial 2-approché pour résoudre ce problème dans les arbres quelconques et pour toute valeur de l'entier  $\rho$ . D'abord, nous définissons un ordre total sur les chemins dans  $\mathcal{P}$  comme suit.

**Définition 7** *Un ordre total  $<$  sur les chemins dans  $\mathcal{P}$  est dit 2-candidat s'il est possible d'associer à chaque chemin  $p \in \mathcal{P}$  un ensemble d'arcs dans  $T$  qui sont traversés par  $p$  et qu'on note  $\text{entrée}(p)$  tel que :*

- (i)  *$\text{entrée}(p)$  contient au plus deux arcs, et*
- (ii) *si  $p < p'$ , où  $p' \in \mathcal{P}$ , et les chemins  $p$  et  $p'$  sont en  $\rho$ -conflit, alors il y a un arc dans l'ensemble  $\text{entrée}(p')$  qui est traversé par  $p$ .*

Pour toute collection de chemins  $\mathcal{P}$  sur un arbre arbitraire  $T$ , nous pouvons trouver un ordre  $<^*$  2-candidat comme suit (voir figure 6.9). Nous considérons  $T$  comme étant un arbre raciné, où on prend comme sommet racine n'importe quel sommet de  $T$ . Définissons le *point plus haut* de chaque chemin  $p \in \mathcal{P}$  comme le sommet dans  $p$  le plus proche du sommet racine. De nouveau, pour chaque chemin  $p \in \mathcal{P}$  on définit l'ensemble  $\text{entrée}(p)$  comme l'ensemble composé des deux arcs traversés par  $p$  qui sont adjacents au point le plus haut de  $p$  si deux tels arcs existent dans  $p$ , sinon l'ensemble  $\text{entrée}(p)$  sera composé par l'unique arc traversé par  $p$  qui est adjacent au sommet initial ou final dans  $p$  qui représente le point le plus haut de  $p$ . Ensuite, pour toute paire de chemins  $p$  et  $p'$  dans  $\mathcal{P}$ , nous disons que  $p <^* p'$  si et seulement si le point le plus haut du chemin  $p$  est plus haut ou égal que celui du chemin  $p'$ . Par la définition 7, il est facile à voir que l'ordre  $<^*$  est un ordre total 2-candidat sur les chemins dans  $\mathcal{P}$ . Dans la figure 6.9, nous présentons un exemple de la construction des ensembles  $\text{entrée}$  et deux ordres totaux 2-candidats pour un arbre et une collection de chemins sur celui-ci donnés.

Dans la suite, nous présentons un algorithme polynomial glouton 2-approché pour le problème de routage dans les arbres sous le mode de communication CGC( $\rho$ ), pour tout  $\rho \geq 1$ .

**Algorithme  $\rho$ -colorie-arbre :**

1. On choisit un ordre total 2-candidat  $<^*$  sur les chemins dans  $\mathcal{P}$  comme indiqué précédemment.
2. On considère les chemins dans  $\mathcal{P}$  en ordre croissant par  $<^*$  et on affecte à chaque chemin la couleur la plus petite possible telle qu'il n'y ait pas de  $\rho$ -conflit avec des chemins auxquels on a déjà affecté une couleur.

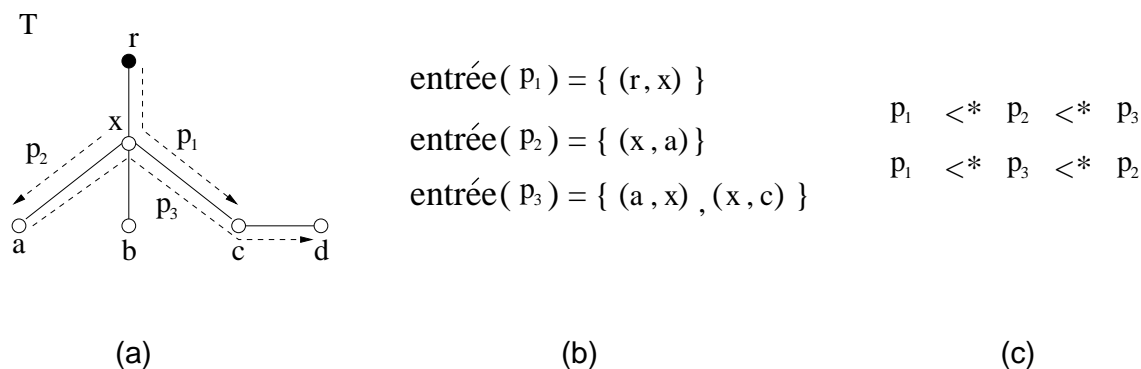


FIGURE 6.9 – (a) Arbre  $T$  enraciné au sommet  $r$  avec chemins  $p_1$ ,  $p_2$  et  $p_3$  sur celui-ci, (b) les ensembles d'arcs  $\text{entrée}(p_i)$ , et (c) les deux ordres totaux 2-candidat possibles sur les chemins  $p_i$ ,  $1 \leq i \leq 3$ .

**Théorème 37** Soient  $T$  un arbre orienté symétrique,  $\mathcal{P}$  une collection de chemins sur  $T$ , et  $\rho$  un entier positif. Soit  $L_T^\rho(\mathcal{P})$  la  $\rho$ -congestion des arcs dans  $T$  induite par  $\mathcal{P}$ . Alors, l'algorithme  $\rho$ -colorie-arbre utilise au plus  $2L_T^\rho(\mathcal{P}) - 1$  couleurs pour colorier proprement les chemins dans  $\mathcal{P}$ .

**Preuve.** Étudions ce qui arrive quand on affecte une couleur à un chemin  $p \in \mathcal{P}$ . Chaque chemin dans  $\mathcal{P}$  auquel on a déjà affecté une couleur et qui est en  $\rho$ -conflit avec  $p$ , traverse un arc dans  $T$  qui se trouve dans l'ensemble  $\text{entrée}(p)$ . Le nombre de tels chemins est au plus égal à  $2(L_T^\rho(\mathcal{P}) - 1)$ , parce que la  $\rho$ -congestion de chaque arc dans  $T$  est au plus  $L_T^\rho(\mathcal{P})$ , et parce qu'il y a au plus deux arcs dans l'ensemble  $\text{entrée}(p)$ . Ainsi, quand on doit affecter une couleur au chemin  $p$ , il y a au plus  $2(L_T^\rho(\mathcal{P}) - 1)$  couleurs non disponibles pour  $p$ . Nous pouvons donc choisir une nouvelle couleur pour le chemin  $p$ , ce qui fait qu'au pire on utilise  $2L_T^\rho(\mathcal{P}) - 1$  couleurs pour colorier proprement les chemins dans  $\mathcal{P}$ .  $\square$

## 6.4 Application au routage wormhole glouton

Dans cette partie, nous appliquons les résultats de complexité obtenus dans la section 6.3, au problème de routage wormhole glouton dans les arbres (voir définition du routage wormhole glouton dans le chapitre 2). Nous allons considérer que la longueur (c-à-d, le nombre de flits) de tous les messages est unitaire. Rappelons que dans ce mode de commutation de paquets, la communication est faite par étapes, où lors d'une étape de communication, chaque sommet du réseau ne peut communiquer qu'avec ses voisins. De plus, à chaque étape de communication, au plus un paquet (flit) peut traverser chaque lien (canal) du réseau. Ainsi, le problème de routage sous ce mode de commutation consiste à minimiser le nombre d'étapes nécessaires pour délivrer tous les paquets depuis leur origine vers leur destination, sachant qu'il n'y a pas de stockage intermédiaire dans le réseau, c'est-à-dire, une fois que l'en-tête (premier flit du message) commence à avancer sur le réseau, il ne s'arrête

que lorsque tout le message (le dernier flit) arrive à sa destination finale. Ce mode de commutation de paquets glouton a été introduit par Bhatt et al. [13] sous une forme plus générale, dans laquelle les messages peuvent avoir une taille variable. Dans le cas où les messages ont une taille variable, Bhatt et al. montrent dans [13] que ce problème de routage est NP-difficile même dans les chaînes. Dans le cas où la taille des paquets est unitaire, Ranade et al. donnent dans [56] un algorithme polynomial 2-approché pour le problème de routage dans les arbres sous ce mode de commutation de paquets. Nous montrerons, en utilisant les idées développées dans la preuve du théorème 36, que le problème de routage wormhole glouton dans les arbres est NP-difficile, même si les arbres sont binaires et si la longueur (c-à-d, le nombre de flits) des messages est unitaire.

**Théorème 38** *Le problème de routage wormhole glouton dans les arbres binaires est NP-difficile, même si la longueur (c-à-d, le nombre de flits) des messages est unitaire.*

**Preuve.** (Par réduction du problème ARÊTE-COLORATION d'un graphe cubique [36]). Soit  $I$  une instance du problème ARÊTE-COLORATION d'un graphe cubique consistant en un graphe cubique non orienté. Soit  $\rho = 2$  et soit  $I'$  l'instance du problème de routage dans les arbres binaires sous le mode CGC( $\rho$ ) construite à partir de l'instance  $I$  dans la preuve du théorème 36, et qui est composée par un arbre binaire  $T$  orienté symétrique et une collection de chemins  $\mathcal{P}$  sur  $T$ . Nous avons montré dans la preuve du théorème 36, qu'il existe une arête-coloration propre de  $G$  avec 3 couleurs si et seulement si  $R_T^2(\mathcal{P}) = 3$ . Soit  $d$  la longueur du plus long chemin dans  $\mathcal{P}$ . Nous allons transformer  $T$  et  $\mathcal{P}$  de sorte que la longueur de tous les chemins dans  $\mathcal{P}$  soit égale à  $d$  comme suit. Pour tout sommet  $v$  dans  $T$ , soit  $\mathcal{P}_v$  l'ensemble de chemins dans  $\mathcal{P}$  qui ont pour sommet final le sommet  $v$ . Alors, pour tout sommet  $v$  dans  $T$  tel que  $\mathcal{P}_v \neq \emptyset$ , nous construisons une nouvelle chaîne qu'on relie au sommet  $v$ , et on allonge les chemins dans  $\mathcal{P}_v$  sur la nouvelle chaîne de sorte que tous aient une longueur égale à  $d$ . Il est clair que la transformation précédente préserve la relation de 2-conflit entre chemins, et donc l'instance obtenue après cette transformation est équivalente à celle de départ. Chaque chemin  $p = u \rightsquigarrow v$  dans la collection courante  $\mathcal{P}$  représente donc l'unique route dans l'arbre binaire courant  $T$  sur laquelle va avancer un paquet unitaire du sommet  $u$  vers le sommet  $v$ . Ainsi, de façon similaire à la preuve du théorème 36, nous pouvons montrer qu'il y a une arête-coloration propre de  $G$  avec 3 couleurs si et seulement si le nombre d'étapes pour délivrer tous les paquets depuis leur origine vers leur destination sans stockage intermédiaire dans l'arbre binaire courant  $T$ , est égale à  $d + 2$ .  $\square$

## 6.5 Problèmes ouverts

Il y a beaucoup de problèmes ouverts intéressants concernant le mode de commutation généralisée de circuits que nous avons introduit dans ce chapitre (hormis la section 5.4). Nous présentons les problèmes suivants.



**Problème 4** Soient  $C_n$  un anneau orienté symétrique,  $\mathcal{C}$  une collection quelconque de requêtes de communication dans  $C_n$ , et  $\rho$  un entier positif quelconque. Existe-il un algorithme en temps polynomial pour trouver une collection  $\mathcal{P}$  de chemins sur  $C_n$  pour router  $\mathcal{C}$  sous le mode  $\text{CGC}(\rho)$  telle que  $L_{C_n}^\rho(\mathcal{C}, \mathcal{P}) = L_{C_n}^\rho(\mathcal{C})$  ?

**Problème 5** Soient  $T$  un arbre orienté symétrique,  $\mathcal{P}$  une collection quelconque de chemins sur  $T$  et  $\rho$  un entier positif. Existe-il un algorithme en temps polynomial  $\epsilon$ -approché pour calculer  $R_T^\rho(\mathcal{P})$ , avec  $4/3 < \epsilon < 2$  ?

Par les lemmes 3 et 4, nous savons que dans le cas de la chaîne et de l'anneau, les graphes de  $\rho$ -conflit induits par une collection quelconque de chemins, avec  $\rho \geq 1$ , sont respectivement, des graphes d'intervalles et des graphes arc-circulaires. En effet, dans le cas de ces réseaux, nous pouvons appliquer une technique de raccourcissement des chemins. Cependant, une telle technique ne fonctionne pas dans le cas des arbres. Ainsi, une question intéressante du point de vue de la théorie des graphes est la suivante.

**Problème 6** Soient  $T$  un arbre orienté symétrique,  $\mathcal{P}$  une collection quelconque de chemins sur  $T$  et  $\rho$  un entier positif. Quel type de graphes représentent les graphes de  $\rho$ -conflit associés à  $T$ ,  $\mathcal{P}$  et  $\rho$  ?

## 6.6 Conclusion

Nous résumons les principaux résultats trouvés dans ce chapitre dans la suite.

1. Nous avons prouvé dans le théorème 34 que pour tout  $\rho \geq 1$ , il existe un algorithme en temps polynomial pour trouver une collection de chemins sur l'anneau pour router une collection quelconque de requêtes de communication sous le mode  $\text{CGC}(\rho)$  de sorte que la  $\rho$ -congestion induite pour une telle collection de chemins dépasse la valeur optimale en au plus une unité.

2. Pour le problème de routage (minimisation du nombre de phases de communication) dans le mode  $\text{CGC}(\rho)$ , nous avons trouvé les résultats suivants que nous présentons dans la table 6.1. Soit  $L$  la valeur de la  $\rho$ -congestion minimale induite pour une collection quelconque de requêtes de communication dans un réseau à  $n$  nœuds. Alors :

3. Finalement, nous avons montré dans le théorème 38, par l'application des résultats obtenus dans le mode  $\text{CGC}(\rho)$ , que le problème de routage wormhole glouton dans les arbres binaires est NP-difficile, même si la longueur (c-à-d, le nombre de flits) des messages à router est unitaire.

réseau	$\rho$	# phases	complexité	référence
anneau	$1 \leq \rho \leq \frac{n}{k},$ $\forall k \geq 3$	$\leq \lfloor (\frac{k}{k-1})L \rfloor + 2$	?	théorème 35(i)
anneau	$> \frac{n}{3}$	$2L + 1$	?	théorème 35(ii)
arbres	$\geq 1$	$2L - 1$	NP-difficile et non $(\frac{4}{3} - \epsilon)$ -approché, $\forall \epsilon > 0$	théorèmes 37 et 36 et corollaire 10

TABLE 6.1 – Résultats sur le problème de minimisation du nombre de phases de communication dans le mode  $CGC(\rho)$ .

# 7 Mode commutation de paquets

Dans ce chapitre nous allons étudier certains aspects du problème du routage store and forward dans la chaîne et dans l'anneau. Tout d'abord, nous allons définir dans la section 7.1 deux paramètres nouveaux que nous appellerons *vecteur d'état* et *encombrement* associés aux canaux de ces réseaux. En nous basant sur ces deux paramètres nouveaux, nous proposons dans la section 7.2 une preuve alternative à celle donnée par Kaufmann et Sibeyn dans [39, 40] pour montrer que la stratégie gloutonne *les plus loin d'abord* est une stratégie optimale (en fonction du nombre d'étapes de communication) pour le problème du routage de messages dans les chaînes et dans les anneaux sous le mode store and forward. Ces nouveaux paramètres vont nous permettre d'obtenir dans la section 7.3 des bornes plus précises pour l'émulation des réseaux de communications dans les chaînes et les anneaux. Nous étudierons aussi dans la section 7.3 l'émulation de l'hypercube par les chaînes et les grilles  $d$ -dimensionnelles. Nous concluons ce chapitre dans la section 7.4. Les résultats montrés dans ce chapitre apparaissent dans [65, 66, 4].

## 7.1 Paramètres nouveaux

Soient  $G$  un réseau et  $\mathcal{P}$  une collection de chemins sur  $G$  associés à une collection quelconque  $\mathcal{C}$  de requêtes de communication dans  $G$ . Nous savons que, aussi bien la congestion  $L_G(\mathcal{C}, \mathcal{P})$  que la dilatation  $D_G(\mathcal{C}, \mathcal{P})$  induites par  $\mathcal{P}$ , sont des bornes inférieures pour le nombre minimal d'étapes de communication nécessaires pour délivrer tous les paquets dans  $\mathcal{C}$  de leur source vers leur destination dans  $G$ . Dans cette section, nous allons introduire un nouveau paramètre associé à  $G$  et à  $\mathcal{P}$ . Ce nouveau paramètre combine la congestion induite sur un lien du réseau par les chemins dans  $\mathcal{P}$  qui utilisent ce lien et la position de ce lien dans ces chemins. Ainsi, en nous basant sur ce nouveau paramètre, nous allons en déduire une borne inférieure plus fine pour le nombre minimal d'étapes nécessaires pour router  $\mathcal{C}$  dans  $G$  en utilisant les chemins dans  $\mathcal{P}$ .

Soit  $d = D_G(\mathcal{C}, \mathcal{P})$  la dilatation induite par les chemins dans  $\mathcal{P}$ . Nous associons à chaque arc  $e \in A(G)$ , un vecteur entier unidimensionnel de longueur  $d$ , que nous appelons *vecteur d'état* et que nous notons par  $E_e$ , où  $E_e[i]$ , avec  $1 \leq i \leq d$ , représente le nombre de chemins dans  $\mathcal{P}$  qui ont l'arc  $e$  dans la  $i$ -ème position.

**Définition 8** Soient  $G$  un réseau et  $\mathcal{P}$  une collection de chemins sur  $G$  avec une dilatation égale à  $d$ . Nous notons par  $\text{pos}(e, p)$ , la position de l'arc  $e \in A(G)$  sur

le chemin  $p \in \mathcal{P}$  (si  $e \in p$ ). Alors, nous définissons pour chaque arc  $e \in A(G)$ , un vecteur entier unidimensionnel  $E_e$ , de la façon suivante :

$$E_e[i] = |\{p \in \mathcal{P} : e \in p \text{ et } \text{pos}(e, p) = i\}|, \quad \text{avec } 1 \leq i \leq d$$

Maintenant, nous définissons un nouveau paramètre associé à un réseau  $G$  et une collection de chemins  $\mathcal{P}$  sur  $G$ , que nous appelons *encombrement*, de la façon suivante.

**Définition 9** Soient  $G$  un réseau de communication et  $\mathcal{P}$  une collection de chemins sur  $G$  associée à une collection quelconque de requêtes de communication dans  $G$ . Soit  $d$  la dilatation des chemins dans  $\mathcal{P}$ . L'encombrement induit par  $\mathcal{P}$ , qu'on note par  $\tau_G(\mathcal{P})$ , est défini par :

$$\tau_G(\mathcal{P}) = \max_{e \in A(G)} \tau_G(\mathcal{P}, e)$$

$$\text{où, } \tau_G(\mathcal{P}, e) = \max_{1 \leq i \leq d} \left( (i-1)\beta_{i,e} + \sum_{k=i}^d E_e[k] \right), \quad \text{avec } \beta_{i,e} = \begin{cases} 0, & \text{si } E_e[i] = 0 \\ 1, & \text{si } E_e[i] > 0 \end{cases}$$

**Proposition 5** Soient  $G$  un réseau de communication,  $\mathcal{P}$  une collection de chemins sur  $G$  associée à une collection  $\mathcal{C}$  de requêtes de communication dans  $G$ , et  $\tau_G(\mathcal{P})$  l'encombrement induit par  $\mathcal{P}$ . Le nombre minimal d'étapes de communication nécessaires pour router  $\mathcal{C}$  dans  $G$  via les chemins dans  $\mathcal{P}$  est au moins égal à  $\tau_G(\mathcal{P})$ .

**Preuve.** Soit  $d$  la dilatation des chemins dans  $\mathcal{P}$ . Soient  $e$  un arc dans  $A(G)$  et  $i$  un entier positif, avec  $1 \leq i \leq d$ . Par la définition 8, on a que  $E_e[i]$  représente le nombre de paquets dont les chemins associés dans  $\mathcal{P}$  utilisent l'arc  $e$  dans la  $i$ -ème position. Donc, si  $E_e[i] > 0$ , alors il faut au moins  $i-1$  étapes de communication pour que le premier de ces  $E_e[i]$  paquets puisse traverser l'arc  $e$ . Soit  $S_e^i$  le sous-ensemble de chemins dans  $\mathcal{P}$  qui ont l'arc  $e$  dans une position au moins égale à  $i$ . D'une part, on a par les hypothèses de communication dans le mode de commutation de paquets, que lors de chaque étape de communication, chaque lien du réseau peut être traversé au plus par un seul paquet. D'autre part, chaque chemin dans  $S_e^i$  représente le chemin par où va transiter un paquet différent. Les  $|S_e^i|$  paquets qui transitent par les chemins dans l'ensemble  $S_e^i$  auront donc besoin d'au moins  $i-1 + |S_e^i| = i-1 + \sum_{k=i}^d E_e[k]$  étapes de communication pour traverser l'arc  $e$ . Ainsi, le nombre d'étapes de communication nécessaires pour que tous les paquets, tels que son chemin associé dans  $\mathcal{P}$  utilise l'arc  $e$  dans  $G$ , est au moins égale à la valeur maximale prise sur toute position  $i$ ,  $1 \leq i \leq d$ , des valeurs  $i-1 + \sum_{k=i}^d E_e[k]$ , ce qui, par la définition (9), est l'encombrement de l'arc  $e$ . Donc, le nombre d'étapes de communication nécessaires pour délivrer tous les paquets transitant par les chemins dans  $\mathcal{P}$ , de son origine vers sa destination dans  $G$ , est au moins égal à l'encombrement  $\tau_G(\mathcal{P})$ .  $\square$

Nous pouvons aussi obtenir des bornes inférieure et supérieure pour l'encombrement  $\tau_G(\mathcal{P})$  comme il est montré dans la proposition suivante.

**Proposition 6** Soient  $G$  un réseau et  $\mathcal{P}$  une collection de chemins sur  $G$ . Soient respectivement  $d$  et  $c$  la dilatation et la congestion induites par  $\mathcal{P}$ . Alors, l'encombrement  $\tau_G(\mathcal{P})$  vérifie :

$$c \leq \tau_G(\mathcal{P}) \leq c + d - 1$$

**Preuve.** Soit  $e$  un arc quelconque dans  $G$ . Par la définition (9), on a  $\tau_G(\mathcal{P}, e) \geq \sum_{i=1}^d E_e[i]$ . Cependant,  $\sum_{i=1}^d E_e[i]$  n'est rien d'autre que la valeur de la congestion de l'arc  $e$  induite par tous les chemins dans  $\mathcal{P}$  qui utilisent l'arc  $e$ . Donc,  $\tau_G(\mathcal{P}) \geq \max_{e \in A(G)} \sum_{i=1}^d E_e[i] = c$ . De plus, pour tout  $i$ , avec  $1 \leq i \leq d$ , on a trivialement que  $i - 1 \leq d - 1$  et  $\sum_{k=i}^d E_e[k] \leq \sum_{k=1}^d E_e[k] \leq c$ . Donc, pour tout arc  $e \in A(G)$ ,  $\tau_G(\mathcal{P}, e) \leq c + d - 1$  et ainsi  $\tau_G(\mathcal{P}) \leq c + d - 1$ .  $\square$

## 7.2 Routage dans les chaînes et les anneaux

Le résultat principal dans cette section est le suivant.

**Théorème 39** Soient  $P_n$  (resp.  $C_n$ ) le réseau chaîne (resp. anneau) à  $n$  sommets et soit  $\mathcal{P}$  une collection de chemins fixée à l'avance sur lesquels va transiter une collection de paquets. Alors, en utilisant la stratégie gloutonne “les plus loin d'abord” pour router ces paquets sur  $P_n$  (resp. sur  $C_n$ ), on obtient que le nombre d'étapes de communication nécessaires pour une telle stratégie, pour délivrer tous les paquets de leur origine à leur destination, est égal à l'encombrement  $\tau_{P_n}(\mathcal{P})$  (resp.  $\tau_{C_n}(\mathcal{P})$ ).

Nous allons montrer le résultat précédent dans le cas des chaînes. Dans le cas des anneaux, la preuve est identique. Tout d'abord nous rappelons la bien connue stratégie de routage *les plus loin d'abord*, que nous appellerons dans la suite la stratégie PLA.

**Stratégie les plus loin d'abord (PLA) :**

Lors de chaque étape de communication, chaque sommet  $i$  dans  $P_n$  fait :

1. Envoie un paquet à chacun des ses voisins  $i - 1$  et  $i + 1$  (s'il existe un tel voisin et un tel paquet) tel que ce paquet a la plus longue distance à parcourir parmi tous les paquets qui se trouvent stockés soit dans sa mémoire locale, soit dans le buffer associé au lien qui le relie à son voisin.
2. Reçoit un paquet provenant de chacun de ses voisins  $i - 1$  et  $i + 1$  (s'il existe un tel voisin et un tel paquet), et
3. si son voisin  $i - 1$  (resp.  $i + 1$ ) lui a envoyé un paquet, alors si la destination d'un tel paquet correspond à son adresse, le sommet  $i$  consomme ce paquet, sinon le paquet est stocké dans le buffer associé au lien qui relie le sommet  $i$  au sommet  $i + 1$  (resp.  $i - 1$ ).

Initialement, tous les paquets se trouvent stockés dans la mémoire locale de leurs sommets source  $i$  dans  $P_n$  et chacun des buffers associés à chaque lien dans  $P_n$  est vide. Considérons une collection quelconque de chemins  $\mathcal{P}$  sur  $P_n$ , associée à une

collection de requêtes de communication dans  $P_n$ . Soit  $d$  la valeur de la dilatation dans  $\mathcal{P}$  et supposons qu'on va utiliser la stratégie PLA pour router les paquets. Soit  $e$  un arc (i.e. un lien) quelconque dans  $P_n$  et soit  $E_e$  le vecteur entier unidimensionnel de longueur  $d$  associé à l'arc  $e$  (voir définition 8). Il est clair que la collection de chemins  $\mathcal{P}$  évolue dynamiquement après chaque étape de communication. En effet, après chaque étape de la stratégie PLA, certains paquets avancent vers leur destination et donc leurs chemins sont raccourcis. Afin de tenir compte de cette évolution dynamique de la collection de chemins associés aux requêtes de communication, nous allons définir, pour chaque arc  $e$  de  $P_n$ , un vecteur entier unidimensionnel de longueur  $d$ , que nous notons par  $S_e^{(r)}$ , tel que  $S_e^{(r)}$  représente l'état du vecteur  $E_e$  après la  $r$ -ème étape de la stratégie PLA, avec  $r \geq 0$ . En effet,  $S_e^{(r)}[i]$ , avec  $1 \leq i \leq d$ , représente le nombre courant de chemins associés aux paquets qui ne sont pas encore arrivés à leur destination après avoir effectué  $r$  étapes de la stratégie PLA, et qui ont l'arc  $e$  dans la  $i$ -ème position. Pour tout arc  $e$  dans  $P_n$  on a  $S_e^{(0)} = E_e$ .

**Lemme 7** *L'état du vecteur  $S_e^{(r)}$  associé à tout arc  $e$  dans  $P_n$  après avoir effectué  $r$  étapes de la stratégie PLA, avec  $r \geq 0$ , peut être obtenu à partir de la relation de récurrence suivante. Pour tout  $i$ , avec  $1 \leq i \leq d$ ,*

$$S_e^{(r)}[i] = \begin{cases} E_e[i], & \text{si } r = 0 \\ S_e^{(r-1)}[i] + \xi(S_e^{(r-1)}, i), & \text{si } r > 0 \end{cases}$$

$$\text{où, } \xi(S_e^{(r-1)}, i) = \begin{cases} 1, & \text{si } S_e^{(r-1)}[i] = 0 \text{ et } S_e^{(r-1)}[i+1] > 0 & \text{(a)} \\ -1, & \text{si } S_e^{(r-1)}[i] > 0 \text{ et } S_e^{(r-1)}[i+1] = 0 & \text{(b)} \\ 0, & \text{sinon} & \text{(c)} \end{cases}$$

**Preuve.** Soient  $i$  et  $r$  deux entiers tels que  $1 \leq i \leq d$  et  $r \geq 1$ . Considérons le cas où  $S_e^{(r-1)}[i] = 0$  et  $S_e^{(r-1)}[i+1] > 0$ . Par définition, l'entier  $S_e^{(r-1)}[i]$  (resp.  $S_e^{(r-1)}[i+1]$ ) représente le nombre courant de chemins associés aux paquets qui ne sont pas encore arrivés à leur destination après avoir effectué  $r-1$  étapes de la stratégie PLA, et qui ont l'arc  $e$  dans la  $i$ -ème (resp. la  $(i+1)$ -ème) position. Ainsi, après la  $(r-1)$ -ème étape de la stratégie PLA, il n'existe aucun paquet dans les sommets de  $P_n$  dont le chemin courant associé ait l'arc  $e$  dans la  $i$ -ème position. De la même façon, après la  $(r-1)$ -ème étape de la stratégie PLA, il y a au moins un paquet dans les sommets de  $P_n$  dont le chemin courant associé ait l'arc  $e$  dans la  $(i+1)$ -ème position. Donc, lors de la  $r$ -ème étape de la stratégie PLA, le paquet  $p$  ayant la plus longue distance à parcourir et ayant pour chemin associé un chemin qui a l'arc  $e$  dans la  $(i+1)$ -ème position sera déplacé vers son voisin immédiat (voir (1) dans la stratégie PLA). Ceci implique qu'après la  $r$ -ème étape de la stratégie PLA, la longueur du chemin courant associé au paquet  $p$  est réduite d'une unité, et donc ce chemin a l'arc  $e$  dans la  $i$ -ème position. Autrement dit,  $S_e^{(r)}[i] = S_e^{(r-1)}[i] + 1$ , ce qui correspond au cas (a). Les cas (b) et (c) sont obtenus d'une façon similaire.  $\square$

L'importance du lemme 7 est fondée sur le déterminisme des vecteurs  $S_e$ , pour tout arc  $e$  de  $P_n$ . En effet, grâce à la stratégie PLA, il suffit de connaître l'état initial du vecteur  $S_e$ , c'est-à-dire,  $S_e^{(0)} = E_e$ , pour déterminer son état après avoir

effectué  $r \geq 1$  étapes de la stratégie PLA, c'est-à-dire, pour connaître le vecteur  $S_e^{(r)}$ . Ainsi, l'état d'un vecteur  $S_e$  peut être déterminé indépendamment de l'état d'un autre vecteur  $S_f$ , avec  $e \neq f$ . De façon plus générale, considérons un vecteur entier semi-positif quelconque et analysons sa dynamique dans le temps déterminée par l'équation du lemme 7.

Soit  $T$  un vecteur entier unidimensionnel de longueur  $d$ , avec  $T[i] \geq 0$ ,  $1 \leq i \leq d$ . L'évolution de l'état du vecteur  $T$  dans le temps, basée sur l'équation donnée dans le lemme 7, peut se décrire par la procédure suivante, que nous noterons par EV.

*Procédure EV :*

Pour chaque instante du temps  $r \geq 1$  et pour tout  $i$ ,  $1 \leq i \leq d$ , faire :

1. Calculer la fonction  $\xi(T, i)$  (voir lemme 7)
2.  $T[i] \leftarrow T[i] + \xi(T, i)$ .

Soit  $\tau_T = \max_{1 \leq i \leq d} \left( (i-1)\beta_i + \sum_{k=i}^d T[k] \right)$ , où  $\beta_i = 0$  si  $T[i] = 0$  ou  $\beta_i = 1$  sinon. Nous noterons par  $T^{(r)}$ , avec  $r \geq 0$ , l'état du vecteur  $T$  après l'instant  $r$ , c'est-à-dire, après avoir appliqué  $r$  fois la procédure EV sur  $T$ , où  $T^{(0)}$  représente l'état initial du vecteur  $T$  à l'instant 0.

**Lemme 8** Soit  $m_T = \min\{r : \sum_{i=1}^d T^{(r)}[i] = 0\}$ . Alors,  $m_T = \tau_T$ .

**Preuve.** Soit  $z_T = |\{r : 0 \leq r < m_T \text{ et } T^{(r)}[1] = 0\}|$ . Nous allons prouver par induction sur  $d$  que  $m_T = \tau_T$  et  $z_T = \tau_T - \sum_{i=1}^d T^{(0)}[i]$ .

**Cas  $d = 1$ .** Il est facile de vérifier que  $m_T = T^{(0)}[1]$  et  $z_T = 0$ . Donc, pour ce cas le lemme est vrai.

**Cas  $d > 1$ .** Nous allons considérer que le lemme est vrai pour tout  $i < d$ . Soit  $T'$  un vecteur entier unidimensionnel de longueur  $d-1$  tel que  $T'[i] = T[i+1]$ , pour  $1 \leq i \leq d-1$ . Alors, par hypothèse inductive, on a  $m_{T'} = \tau_{T'}$  et  $z_{T'} = \tau_{T'} - \sum_{i=2}^d T^{(0)}[i]$ . Par la définition de  $\tau_T$  et par l'hypothèse inductive, on a  $\tau_T = \max(\tau_{T'} + 1, \sum_{i=1}^d T^{(0)}[i])$ . Ainsi, nous considérons les deux sous-cas suivants :

- $\tau_T = \tau_{T'} + 1$ . Dans ce cas, par l'hypothèse inductive, après avoir appliqué  $r = \tau_{T'} = m_{T'}$  fois la procédure EV sur  $T$ , on a  $T^{(r)}[i] = 0$  pour tout  $i$ , avec  $2 \leq i \leq d$ . En effet, soit  $p$  un entier,  $0 \leq p < r$ , tel que  $T^{(p)}[2] = 0$ . Par application du cas (b) du lemme 7, on a  $T^{(p+1)}[1] = T^{(p)}[1] - 1$ , si  $T^{(p)}[1] > 0$ . Donc, après avoir appliqué  $r$  fois la procédure EV sur  $T$ , on a  $T^{(r)}[1] = T^{(0)}[1] - z_{T'}$ . De plus, par l'hypothèse inductive,  $T^{(r-1)}[2] = 1$ , ce qui implique, par application du cas (a) du lemme 7, que  $T^{(r)}[1]$  est au moins égal à 1. Donc,  $T^{(r)}[1] = \max(T^{(0)}[1] - z_{T'}, 1) = 1$ , et par définition de  $z_{T'}$  et par l'hypothèse inductive,  $T^{(r)}[1] = \max(\sum_{i=1}^d T^{(0)}[i] - \tau_{T'}, 1) = 1$ . Ainsi,  $m_T = m_{T'} + 1 = \tau_{T'} + 1 = \tau_T$ .

De plus, si  $T^{(p)}[1] = 0$ , alors par application du cas (c) du lemme 7,  $T^{(p+1)}[1] = T^{(p)}[1]$ , et donc, par l'hypothèse inductive et par application des cas (b) et

- (c) du lemme 7,  $z_T = \max(z_{T'} - T^{(0)}[1] + 1, 0) = \tau_{T'} + 1 - \sum_{i=1}^d T^{(0)}[i] = \tau_T - \sum_{i=1}^d T^{(0)}[i]$ .
- $\tau_T = \sum_{i=1}^d T^{(0)}[i] > \tau_{T'} + 1$ . Dans ce cas, après avoir appliqué  $r = \tau_{T'} = m_{T'}$  fois la procédure EV sur  $T$ , on a  $T^{(r)}[i] = 0$ , pour  $2 \leq i \leq d$ . De plus, par application des cas (a) et (b) du lemme 7,  $T^{(r)}[1] = \max(T^{(0)}[1] - z_{T'}, 1) = \tau_T - \tau_{T'}$ . Donc,  $m_T = \tau_T$  et  $z_T = 0$ .

Ainsi, nous avons montré que le lemme est vrai pour tout  $d > 0$ .  $\square$

### Preuve du théorème 39

Soit  $d$  la dilatation induite par la collection de chemins  $\mathcal{P}$  sur  $P_n$ . Soit  $e$  un arc quelconque dans  $P_n$ , et soit  $E_e$  le vecteur entier unidimensionnel de longueur  $d$  associé à l'arc  $e$  et qui est induit par les chemins dans  $\mathcal{P}$  (voir définition 8). Par le lemme 8, le nombre minimal de fois qu'il faut appliquer la procédure EV sur  $E_e$  de sorte que  $E_e$  devienne un vecteur nul (c-à-d,  $E_e[i] = 0$ ,  $1 \leq i \leq d$ ) est égal à  $m_{E_e} = \max_{1 \leq i \leq d} \left( (i-1)\beta_{i,e} + \sum_{k=i}^d E_e[k] \right)$ , qui correspond à l'encombrement de l'arc  $e$  induit par  $\mathcal{P}$  (voir définition 9). Donc, par l'application des lemmes 7 et 8 sur tous les arcs  $e$  dans  $P_n$ , on obtient le résultat.  $\square$

## 7.3 Émulation de réseaux dans les chaînes

Une propriété importante d'un réseau de communication  $H$  est de savoir si on peut y simuler efficacement l'exécution d'un algorithme parallèle exécuté dans un autre réseau  $G$ . On dira alors que  $H$  *émule* cet algorithme. On parlera de l'émulation de  $G$  par  $H$  si on s'intéresse à la manière dont  $H$  peut simuler n'importe quel algorithme s'exécutant dans  $G$  (les notions et résultats généraux relatifs à ce domaine peuvent être trouvés dans [45, 20]). Afin de pouvoir émuler efficacement un réseau  $G$  par un autre  $H$ , on utilise la notion de *plongement* de graphes. Soient  $G$  et  $H$  deux graphes orientés symétriques à  $n$  sommets. Comme on l'a défini dans la section 1, un plongement de  $G$  dans  $H$  consiste en un couple d'applications injectives  $(f, \mathcal{R}_f)$ , où  $f$  associe à chaque sommet de  $G$  un sommet de  $H$ , et  $\mathcal{R}_f$  associe à chaque arc  $(u, v)$  de  $G$  un chemin orienté  $f(u) \rightsquigarrow f(v)$  de  $H$ . Ainsi, la qualité d'un plongement  $(f, \mathcal{R}_f)$  de  $G$  dans  $H$  est évaluée basiquement par deux paramètres : la dilatation  $D_H(\mathcal{P})$  et la congestion  $L_H(\mathcal{P})$  de la collection de chemins  $\mathcal{P}$  sur  $H$  induite par  $(f, \mathcal{R}_f)$ . De plus, nous savons par le célèbre résultat de Leighton et al. [46] (voir aussi [47, 48, 57]), que pour toute collection de requêtes de communication dans un réseau  $H$ , à laquelle est associée une collection de chemins  $\mathcal{P}$  sur  $H$  avec une dilatation égale à  $d$  et une congestion égale à  $c$ , le nombre d'étapes nécessaires de communication pour réaliser toutes ces requêtes de communication dans le mode de commutation de paquets est égal à  $O(c + d)$ , avec buffers dans les liens de  $H$  de taille  $O(1)$  (voir section 2). Ainsi, si la dilatation d'un plongement de  $G$  dans  $H$  est égale à  $d$  et si la congestion est égale à  $c$ , alors un algorithme parallèle s'exécutant en  $T$  étapes de communication dans  $G$  sera émulé en au plus  $O(c + d)T$  étapes de



communication dans  $H$ . Donc, le problème d'émuler un réseau  $G$  dans un autre  $H$  sous le mode de commutation de paquets, revient à trouver un plongement de  $G$  dans  $H$  qui minimise le facteur *congestion + dilatation*, plus une stratégie pour déplacer les paquets de leur origine à leur destination, suivant leurs chemins associés dans  $H$  par un tel plongement.

### 7.3.1 Bornes pour l'émulation d'un graphe dans la chaîne et dans l'anneau

Nous allons montrer des bornes inférieures et supérieures pour l'émulation d'un graphe quelconque dans les chaînes et les anneaux. Nous montrerons ces résultats que dans le cas des chaînes. Dans le cas des anneaux, les résultats sont identiques à ceux obtenus pour les chaînes.

Soit  $G$  un graphe orienté symétrique à  $n$  sommets et de degré maximum  $\Delta$ . Soit  $(f, \mathcal{R}_f)$  un plongement quelconque de  $G$  dans la chaîne  $P_n$  induisant un ensemble de chemins  $\mathcal{P}$  sur  $P_n$ . Soient  $d$  et  $c$  respectivement, la valeur de la dilatation et de la congestion induites par  $\mathcal{P}$ , et soit  $\gamma$  le nombre minimum d'étapes de communication dans  $P_n$  nécessaires pour émuler une seule étape de communication dans  $G$ .

**Proposition 7**  $c \leq \gamma \leq c + d - \left\lfloor \frac{c}{\Delta} \right\rfloor - \left\lfloor \frac{\Delta - 1}{2} \right\rfloor$

**Preuve.** Grâce au théorème 39, il suffit d'obtenir ces bornes pour l'encombrement  $\tau_{P_n}(\mathcal{P})$  induit par l'ensemble de chemins  $\mathcal{P}$ . Par la proposition 6, on a directement la borne inférieure. Soit  $e$  un arc quelconque dans  $P_n$  et soit  $k_*$  le plus grand entier tel que  $E_e[i] = 0$ , pour  $1 \leq i < k_*$ , et tel que  $\sum_{i=k_*}^d E_e[i] = c$ . Alors,  $\tau_{P_n}(\mathcal{P}) \leq k_* - 1 + \sum_{i=k_*}^d E_e[i] = k_* - 1 + c$ . Comme l'application  $f$  est par définition injective, alors il est clair que  $E_e[d] \leq 1, E_e[d-1] \leq 2, \dots, E_e[d-\Delta+2] \leq \Delta-1$ , et pour tout  $i$ , avec  $k_* \leq i \leq d-\Delta+1$ ,  $E_e[i] \leq \Delta$ . Soit  $\alpha = d - \Delta + 1$ . Il n'est pas difficile de voir que dans le pire des cas, l'encombrement le plus grand qu'on puisse obtenir est quand  $E_e[i] = \Delta$ , avec  $k_* \leq i < \alpha$ , et  $E_e[m] = \Delta - (m - \alpha)$ , avec  $\alpha \leq m \leq d$ . Dans ce cas, la congestion  $c$  vérifie la équation suivante :

$$c = (\alpha - k_*)\Delta + \sum_{m=\alpha}^d (\Delta - (m - \alpha))$$

$$c = (\alpha - k_*)\Delta + \sum_{m=1}^d (\Delta - (m - \alpha)) - \sum_{m=1}^{\alpha-1} (\Delta - (m - \alpha))$$

$$c = (\alpha - k_*)\Delta + \frac{d}{2}(2\Delta + 2\alpha - d - 1) - \frac{(\alpha - 1)}{2}(2\Delta + \alpha) \quad (1)$$

En remplaçant  $\alpha$  par  $d - \Delta + 1$  dans l'équation (1), nous obtenons

$$k_* = d - \frac{c}{\Delta} - \frac{\Delta - 3}{2} \quad (2)$$

Finalement, en utilisant l'équation (2) on obtient

$$\tau_{P_n}(\mathcal{P}) \leq k_* - 1 + c \leq c + d - \left\lfloor \frac{c}{\Delta} \right\rfloor - \left\lfloor \frac{\Delta - 1}{2} \right\rfloor$$

□

### 7.3.2 Émulation de l'hypercube binaire par la chaîne

Dans cette section, nous allons prouver les deux résultats suivants :

**Théorème 40** *Une étape de communication dans l'hypercube binaire  $H(n)$  à  $2^n$  sommets peut être émulée par la chaîne  $P_{2^n}$  en  $\left\lfloor \frac{2^{n+1}}{3} \right\rfloor$  étapes de communication, en utilisant des buffers dans les liens de  $P_{2^n}$  de taille  $O(n)$ . De plus, cette émulation est optimale en fonction du nombre d'étapes.*

**Théorème 41** *Une étape de communication dans l'hypercube binaire  $H(n)$  à  $2^n$  sommets peut être émulée par la chaîne  $P_{2^n}$  en  $2^n - 1$  étapes de communication, sans utiliser de buffers dans les liens de  $P_{2^n}$ .*

Nous rappelons (voir chapitre 1), que l'hypercube binaire  $H(n)$  est un graphe à  $2^n$  sommets étiquetés par les entiers  $0, 1, \dots, 2^n - 1$ , et deux sommets dans  $H(n)$  sont adjacents si et seulement si leur représentation binaire diffère en une seule coordonnée. Soit  $(f, \mathcal{R}_f)$  un plongement de  $H(n)$  dans  $P_{2^n}$ . Bel-Hala a montré dans [9], que si l'application  $f$  est la fonction identité, alors  $(f, \mathcal{R}_f)$  est un plongement à congestion optimale de  $H(n)$  dans  $P_{2^n}$ . De plus, Bel-Hala montre dans [9] que si  $f$  est la fonction identité, alors la valeur de la congestion optimale induite par  $(f, \mathcal{R}_f)$  est égale à  $\left\lfloor \frac{2^{n+1}}{3} \right\rfloor$  et la valeur de la dilatation induite par ce plongement identité est égale à  $2^{n-1}$ .

Pour prouver les théorèmes 40 et 41, nous allons utiliser dorénavant le plongement identité de  $H(n)$  dans  $P_{2^n}$ . De plus, pour prouver le théorème 40, nous allons utiliser la stratégie PLA (c-à-d, "les plus loins d'abord"), pour laquelle on a montré dans le théorème 39 que c'est une stratégie optimale pour le routage dans la chaîne sous le mode de commutation de paquets.

Avant de calculer le nombre d'étapes de communication dans  $P_{2^n}$  pour émuler une étape de communication dans  $H(n)$ , en utilisant le plongement identité et la stratégie PLA, nous allons calculer la valeur minimale et la valeur maximale des étiquettes des sommets dans  $P_{2^n}$  qui reçoivent un paquet de leurs voisins à gauche lors de la dernière étape de communication. Il est clair que ces deux sommets existent, puisque le plongement identité  $(f, \mathcal{R}_f)$  de  $H(n)$  dans  $P_{2^n}$  est symétrique. De plus, par cette symétrie, nous pouvons appliquer le même type de raisonnement pour les paquets dans  $P_{2^n}$  qui transitent de droite à gauche.

Soient  $i_n$  et  $j_n$  deux sommets dans  $P_{2^n}$  tels que  $i_n$  est le sommet avec l'étiquette la plus petite, et  $j_n$  est le sommet avec l'étiquette la plus grande, qui reçoivent un paquet provenant de leur sommet voisin à gauche, lors de la dernière étape de

communication réalisée par la stratégie PLA dans  $P_{2^n}$  pour émuler une étape de communication dans  $H(n)$ . Alors, on a le résultat suivant.

**Lemme 9**  $i_n = \frac{1}{3}(2^n + (-1)^{n+1})$  et  $j_n = 2^n - i_n$ .

**Preuve.** Soient  $G_{2^{n-1}}$  et  $G'_{2^{n-1}}$  les deux sous-graphes de  $H(n)$  induits respectivement par les ensembles de sommets  $\{f^{-1}(0), \dots, f^{-1}(2^{n-1}-1)\}$  et  $\{f^{-1}(2^{n-1}), \dots, f^{-1}(2^n-1)\}$ , qui sont isomorphes à deux sous-hypercubes  $H(n-1)$  par symétrie de  $H(n)$  et par définition du plongement identité  $(f, \mathcal{R}_f)$  de  $H(n)$  dans  $P_{2^n}$ . Il est clair que chacun des sommets dans  $G_{2^{n-1}}$  a un seul voisin dans  $G'_{2^{n-1}}$  à distance  $2^{n-1}$  dans  $P_{2^n}$  et  $n-1$  voisins dans  $G_{2^{n-1}}$ . Ceci implique que les paquets que les sommets dans  $G_{2^{n-1}}$  envoient à leurs voisins dans  $G'_{2^{n-1}}$ , avanceront successivement par les sommets intermédiaires  $j$  dans  $P_{2^n}$ , avec  $1 \leq j \leq 2^{n-1} - 1$ , étape par étape, sans interruption, lors des premières  $2^{n-1}$  étapes de communication. En effet, par le plongement choisi, la distance que doit parcourir chacun de ces paquets est plus grande que celle de tous les autres  $n-1$  paquets originaires du même sommet. Ainsi, le sommet dans  $G_{2^{n-1}}$  placé sur le sommet  $2^{n-1} - 1$  dans  $P_{2^n}$  commencera à recevoir les paquets de ses  $n-1$  voisins dans  $G_{2^{n-1}}$  qu'au moins lors de la  $(2^{n-1})$ -ème étape de communication. Donc,  $i_n \in [1, 2^{n-1} - 1]$ , et par symétrie du plongement identité  $(f, \mathcal{R}_f)$ , on a  $j_n \in [2^{n-1} + 1, 2^n - 1]$ .

Si on continue l'analyse récursive pour le graphe  $G_{2^{n-1}}$  qui est composé, par définition du réseau  $H(n)$ , par deux sous-hypercubes de dimension  $n-2$ , nous obtenons que  $i_{n-1} \in [1, 2^{n-2} - 1]$  et  $j_{n-1} \in [2^{n-2} + 1, 2^{n-1} - 1]$ . Ceci implique que  $i_n \in [2^{n-1} - 2^{n-2}, 2^{n-1} - 1]$  et  $j_n \in [2^{n-1} + 1, 2^n - (2^{n-1} - 2^{n-2})]$ . En continuant la récurrence sur  $n$ , et étant donné que  $H(1)$  est isomorphe à  $P_2$ , et ainsi  $i_1 = j_1 = 1$ , nous obtenons l'équation de récurrence suivante :

$$\begin{cases} i_1 = 1 \\ i_n = 2^{n-1} - i_{n-1}, \text{ pour } n \geq 2 \end{cases}$$

et par symétrie du plongement identité  $(f, \mathcal{R}_f)$ , on a  $j_n = 2^n - i_n$ .

Pour résoudre l'équation de récurrence précédente, notons que,

$$i_{2n} = 2^{2n-1} - i_{2n-1} \quad \text{et} \quad i_{2n+1} = 2^{2n} - i_{2n}$$

et ainsi,

$$\begin{cases} i_2 = 1 \\ i_{2n} = 2^{2n-1} - i_{2n-1} = 2^{2n-2} + i_{2n-2} \end{cases}$$

Donc,

$$i_{2n} = 1 + \sum_{k=1}^{n-1} 2^{2n-2k} = \frac{1}{3}(2^{2n} - 1)$$

et

$$i_{2n+1} = 2^{2n} - i_{2n} = \frac{1}{3}(2^{2n+1} + 1)$$

d'où on obtient finalement que  $i_n = \frac{1}{3}(2^n + (-1)^{n+1})$ . □

**Preuve du théorème 40**

Nous notons par  $r_n$ , le nombre d'étapes de communication dans  $P_{2^n}$  pour émuler une étape de communication dans  $H(n)$ , en utilisant le plongement identité de  $H(n)$  dans  $P_{2^n}$  et la stratégie PLA. Nous allons raisonner par récurrence sur  $n$ . Pour  $n = 1$ , on a que  $H(1)$  est isomorphe à  $P_2$ , et donc  $r_1 = 1$ . Pour tout  $n \geq 2$ , nous avons, par la stratégie PLA, que tout sommet  $k$  dans  $P_{2^n}$ , avec  $0 \leq k \leq 2^{n-1} - 2$ , ne peut commencer à transmettre un paquet destiné à un sommet  $j$  dans  $P_{2^n}$ , avec  $k < j \leq 2^{n-1} - 1$ , que après la  $k$ -ème étape de communication, puisque lors de cette étape (c-à-d, la  $k$ -ème), celui qui reçoit un paquet destiné à un sommet  $i$  dans  $P_{2^n}$ , avec  $2^{n-1} \leq i \leq 2^n - 1$  est prioritaire. Comme  $i_n$  est le sommet dans  $P_{2^n}$  avec l'étiquette la plus petite qui reçoit un paquet lors de la dernière étape de communication (voir lemme 9), et il est tel que  $i_n \in [1, 2^{n-1} - 1]$ , alors nous pouvons déduire facilement l'équation de récurrence suivante pour  $r_n$  :

$$\begin{cases} r_1 = 1 \\ r_n = r_{n-1} + i_n = r_{n-1} + (2^{n-1} - i_{n-1}), \text{ pour } n \geq 2 \end{cases}$$

Cette équation de récurrence peut être exprimée comme :

$$r_n = 1 + \sum_{k=1}^{n-1} 2^{n-k} - \sum_{k=1}^{n-1} i_{n-k} = 2^n - 1 - \sum_{k=1}^{n-1} i_{n-k}.$$

En résolvant la dernière somme et en utilisant le résultat du lemme 9, nous obtenons,

$$\sum_{k=1}^{n-1} i_{n-k} = \sum_{k=1}^{n-1} i_k = \frac{1}{3} \sum_{k=1}^{n-1} 2^k + \frac{1}{3} \sum_{k=1}^{n-1} (-1)^{k+1} = \frac{1}{3}(2^n - 2) + \frac{1}{3}(1 - (n \bmod 2)).$$

Donc, on a finalement que  $r_n = \frac{1}{3}(2^{n+1} - 2 + (n \bmod 2)) = \left\lfloor \frac{2^{n+1}}{3} \right\rfloor$ .

De plus, comme la mémoire interne de chaque sommet dans  $P_{2^n}$  contient au plus  $n$  paquets, alors il est clair que la taille des buffers dans les liens de  $P_{2^n}$  nécessaire pour la stratégie PLA pour émuler une étape de communication de  $H(n)$  est au plus égale à  $n$ . Ceci termine la preuve du théorème.  $\square$

**Preuve du théorème 41**

Rappelons qu'on utilise le plongement identité  $(f, \mathcal{R}_f)$  de  $H(n)$  dans  $P_{2^n}$  pour l'émulation de  $H(n)$  par  $P_{2^n}$ . Alors, nous allons utiliser dans  $P_{2^n}$  une stratégie de communication gloutonne. En effet, dans cette stratégie, une fois qu'un paquet est parti de son sommet source, il ne s'arrête que lorsqu'il arrive à son sommet destination. Cette stratégie est assez simple. Lors de la  $(2^{n-i})$ -ème étape de communication, avec  $1 \leq i \leq n$ , chaque sommet  $f(j) \in P_{2^n}$  envoie l'unique paquet destiné au sommet  $f(k) \in P_{2^n}$  qui se trouve à une distance égale à  $2^{n-i}$  de lui. Il n'est pas difficile de vérifier que pour cette stratégie, il n'existe pas une paire de paquets voulant emprunter le même lien dans  $P_{2^n}$  lors de la même étape de communication. Donc, une telle

stratégie n'a pas besoin de buffers dans les liens de  $P_{2^n}$ . Ainsi, le nombre d'étapes de communication nécessaires pour une telle stratégie de communication dans  $P_{2^n}$  pour émuler une étape de communication dans  $H(n)$  est égal à  $\sum_{i=1}^n 2^{n-i} = 2^n - 1$ .  $\square$

Une conséquence des théorèmes 40 et 41 est le corollaire suivant.

**Corollaire 11** *La stratégie gloutonne donnée dans la preuve du théorème 41 est une stratégie  $\frac{3}{2}$ -approchée pour le problème de l'émulation de l'hypercube  $H(n)$  par la chaîne  $P_{2^n}$  sous le mode de commutation de paquets sans stockage intermédiaire.*

**Preuve.** Soit  $r_1$  (resp.  $r_2$ ) le nombre d'étapes de communication nécessaires dans  $P_{2^n}$  pour émuler une étape de communication dans  $H(n)$  en utilisant la stratégie de communication PLA (resp. celle de la preuve du théorème 41). Alors, par les théorèmes 40 et 41 on a  $\frac{r_2}{r_1} \leq \frac{3}{2}$ , ce qui prouve le corollaire.  $\square$

### 7.3.3 Émulation de l'hypercube par la grille

Nous allons généraliser les résultats présentés précédemment concernant l'émulation de l'hypercube par la chaîne au cas de l'émulation de l'hypercube par une grille. Une propriété de l'hypercube  $H(n_1 + n_2 + \dots + n_d)$  est qu'il peut être représenté comme la somme cartésienne des sous-hypercubes  $H(n_1) \square H(n_2) \square \dots \square H(n_d)$ , avec  $n_i \geq 0$  et  $1 \leq i \leq d$ . De plus, une grille à  $d$  dimensions, notée  $M(n_1, n_2, \dots, n_d)$ , est une somme cartésienne de  $d$  chaînes  $P_{n_i}$  ayant chacune  $n_i$  sommets.

Soit  $(f_i, \mathcal{R}_{f_i})$  le plongement identité de  $H(n_i)$  dans  $P_{n_i}$ , avec  $1 \leq i \leq d$ , et soit  $(f_{1,2,\dots,d}, \mathcal{R}_{f_{1,2,\dots,d}})$  le plongement produit identité de  $H(n)$  dans  $M(n_1, n_2, \dots, n_d)$ . En utilisant le résultat de Ho et Johnsson [34] (voir chapitre 1), Bel-Hala montre dans [10] le résultat suivant.

**Théorème 42** (Bel-Hala [10]). *La congestion induite par le plongement produit identité  $(f_{1,2,\dots,d}, \mathcal{R}_{f_{1,2,\dots,d}})$  de  $H(n)$  dans  $M(n_1, n_2, \dots, n_d)$  est optimale. De plus, la congestion  $c^*$  et la dilatation  $d^*$  d'un tel plongement vérifient :*

$$c^* = \max\left\{\left\lfloor \frac{2^{n_i+1}}{3} \right\rfloor : 1 \leq i \leq d\right\} \text{ et } d^* = \max\{2^{n_i-1} : 1 \leq i \leq d\}.$$

Ainsi, en utilisant les théorèmes 40, 41 et 42, nous obtenons les résultats suivants concernant l'émulation de l'hypercube binaire par la grille  $d$ -dimensionnelle.

**Théorème 43** *Soit  $n = n_1 + n_2 + \dots + n_d$  un entier positif avec  $n_i > 0$  pour  $1 \leq i \leq d$ . Une étape de communication dans l'hypercube binaire  $H(n)$  à  $2^n$  sommets peut être émulée par la grille  $d$ -dimensionnelle  $M(n_1, n_2, \dots, n_d)$  en :*

- (i)  $\max\left\{\left\lfloor \frac{2^{n_i+1}}{3} \right\rfloor : 1 \leq i \leq d\right\}$  étapes de communication, en utilisant des buffers dans les liens de  $M(n_1, n_2, \dots, n_d)$  de taille  $\max\{n_i : 1 \leq i \leq d\}$ . De plus, cette émulation est optimale en fonction du nombre d'étapes.
- (ii)  $\max\{2^{n_i} - 1 : 1 \leq i \leq d\}$  étapes de communication, sans utiliser de buffers dans les liens de  $M(n_1, n_2, \dots, n_d)$ . De plus, cette émulation utilise au plus  $\frac{3}{2}$  fois le nombre optimal d'étapes.

## 7.4 Conclusion

Nous résumons les résultats trouvés dans ce chapitre dans la table 7.1 suivante.

Une étape de communication de l'hypercube  $H(n)$ , avec  $n = n_1 + n_2 + \dots + n_d$  et  $n_i > 0$  pour  $1 \leq i \leq d$ , peut être émulée par la grille  $d$ -dimensionnelle  $M(n_1, n_2, \dots, n_d)$  dans :

# étapes	taille du buffer	référence
$\max\{\lfloor \frac{2^{n_i+1}}{3} \rfloor : 1 \leq i \leq d\}$	$\max\{n_i : 1 \leq i \leq d\}$	théorème 43(i)
$\max\{2^{n_i} - 1 : 1 \leq i \leq d\}$	0	théorème 43(ii)

TABLE 7.1 – Résultats obtenus pour l'émulation de l'hypercube dans la grille  $d$ -dimensionnelle.

## 8 Conclusions

Dans cette thèse, nous avons étudié la complexité algorithmique liée à la détermination des stratégies de routage optimales dans les réseaux de communications. Nous avons montré que sous certaines hypothèses de communication, le problème de trouver une stratégie efficace pour le routage des données dans les réseaux de communications est un problème NP-difficile, même si les instances considérées du problème sont assez particulières comme le routage de permutations dans les arbres binaires. Nous avons analysé la complexité en moyenne du problème de routage par chemins arc-disjoints dans les réseaux et nous avons obtenu des résultats sur ce problème dans les arbres. Afin de mieux comprendre la difficulté inhérente de certaines instances du problème de routage par rapport à d'autres instances du même problème et essayer ainsi de diminuer l'écart entre elles, nous avons généralisé le mode de commutation de circuits. Finalement, nous avons trouvé des résultats optimaux concernant l'émulation de l'hypercube par la grille  $d$ -dimensionnelle sous les hypothèses du routage store-and-forward. À la fin de chaque chapitre sont données des conclusions propres à chaque thème. Cependant, nous allons reprendre ici les principaux résultats obtenus et donner des perspectives de travail envisagées.

### Coloration de chemins dans l'anneau

Bien que le problème de la coloration de chemins dans l'anneau que nous avons étudié dans le chapitre 4 ne résout pas complètement le problème du routage par chemins arc-disjoints dans ce réseau, il peut s'avérer être de grande importance au moment de trouver des solutions qui sont assez proches de l'optimal. Nous avons montré dans le chapitre 4 que pour un anneau et une collection de chemins donnés, si  $l \geq 4$  chemins sont nécessaires pour couvrir complètement l'anneau tels que leur graphe de conflit associé est un cycle, alors il existe un algorithme en temps polynomial qu'utilise au plus  $\binom{l-1}{l-2}$  fois le nombre optimal de couleurs nécessaires pour colorier cette collection. Notre résultat généralise celui de Tucker qui avait montré, sous ces hypothèses, que le nombre de couleurs nécessaires était au plus  $\frac{3}{2}$  fois le nombre optimal. Nous avons aussi étudié ce problème de coloration de chemins dans l'anneau en tenant compte la longueur des chemins. En particulier, nous avons trouvé que si la longueur de tous les chemins est dans l'ensemble  $\{1, \alpha, \alpha + 1\}$ , pour toute valeur entière  $\alpha \geq 2$ , alors il existe un algorithme en temps polynomial pour colorier efficacement ces chemins. Ainsi, si la longueur de tous les chemins est au plus égale à 3, alors ce problème peut être résolu efficacement en temps polynomial. Cependant, nous ne savons rien sur la complexité algorithmique de ce problème de

coloration de chemins dont la longueur est au moins égale à 4.

Une perspective de travail plus ambitieuse en rapport avec le dernier problème mentionné est celle d'étudier la complexité algorithmique du problème d'une  $r$ -uplet coloration minimale des graphes arc-circulaires  $k$ -coloriables (voir chapitre 4 section 4.2.1). En effet, supposons que nous avons un graphe arc-circulaire  $k$ -coloriable quelconque, avec  $k \geq 1$ , et supposons que nous pouvons calculer en temps polynomial une  $k$ -coloration optimale propre pour ce graphe. Pour un entier positif  $r$  fixé, par exemple  $r = 2$ , qu'est-ce que pouvons nous dire sur la complexité d'une  $r$ -uplet coloration de ce graphe ?

## Coloration de chemins dans les arbres

Nous avons montré dans le chapitre 5 que le problème de coloration de chemins dans les arbres est un problème très difficile. En effet, nous avons prouvé dans ce chapitre que colorier des ensembles de chemins représentant une permutation des sommets d'un arbre est un problème NP-difficile, même pour des permutations particulières comme les involutions ou les permutations circulaires dans les arbres de degré borné par une constante. Nous avons aussi analysé la complexité en moyenne du problème de coloration de permutations dans les arbres, trouvant ainsi des résultats exacts dans le cas des chaînes et des étoiles généralisées, et des bornes inférieure et supérieure dans le cas des arbres quelconques. Pour obtenir ces résultats sur la complexité en moyenne du problème de coloration de permutations dans les arbres, nous avons combiné des techniques issues de la combinatoire énumérative et asymptotique avec des techniques de marches aléatoires. Malheureusement, nous ne savons pas si ces techniques peuvent être appliquées à d'autres classes de graphes. De plus, notre résultat utilise la propriété du rapport constant (inférieure à 2) entre le nombre chromatique des graphes de conflit induits par les ensembles de chemins représentant une permutation des sommets d'un arbre et le nombre de clique de ces graphes de conflit. Dans le cas du problème de routage de permutations par chemins arc-disjoints dans les graphes orientés symétriques quelconques, il y a peu de résultats sur la non-approximabilité du problème. Il reste donc beaucoup de travail à faire.

## Commutation généralisée de circuits

Afin de mieux comprendre la difficulté inhérente de certaines instances du problème de routage par chemins arc-disjoints par rapport à d'autres instances du même problème et essayer ainsi de diminuer l'écart entre elles, nous avons généralisé dans le chapitre 6 le mode de commutation de circuits. Ainsi, nous avons introduit un paramètre entier que nous avons appelé *la tolérance* du modèle, qui relaxe la notion de conflit entre chemins. Nous avons obtenu des résultats de complexité pour ce problème de routage dans les chaînes, les anneaux et les arbres. Nous avons donc constaté que ce type de relaxation ne donne pas des meilleurs résultats vis-à-vis de la complexité par rapport au problème de routage par chemins arc-disjoints. Nous avons appliqué les résultats de complexité obtenus pour le problème de routage sous



ce mode de commutation généralisée au problème de routage wormhole glouton (i.e. sans buffers) dans les arbres binaires et nous avons montré que ce dernier problème de routage est aussi NP-difficile.

## Commutation de paquets

Finalement, dans le chapitre 7 nous avons étudié le problème de l'émulation de l'hypercube binaire à  $n$  nœuds par la grille  $d$ -dimensionnelle à  $n$  nœuds sous le modèle de routage store-and-forward. Nous avons montré qu'il existe une stratégie en temps polynomial pour émuler efficacement l'hypercube par la grille avec un nombre optimal d'étapes de communication en introduisant des buffers de taille  $\log_2 n$  dans les liens de la grille. De plus, nous avons montré qu'il existe une stratégie en temps polynomial  $\frac{3}{2}$ -approchée pour émuler l'hypercube par la grille sans utiliser de buffers dans les liens de celle-ci. Un problème intéressant est celui de savoir s'il existe une stratégie en temps polynomial pour l'émulation de l'hypercube dans la grille sans utiliser de buffers dans les liens de celle-ci, avec un nombre optimal d'étapes de communication.



# Bibliographie

- [1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient routing and scheduling algorithms for optical networks. In *Proc. of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '94*, pages 412–423, 1994.
- [2] J. Babcock. SONET : A practical perspective. *Business Comm. Rev.*, pages 59–63, Sept. 1990.
- [3] D. Barth, S. Corteel, A. Denise, D. Gardy, and M. Valencia-Pabon. On the complexity of routing permutations on trees by arc-disjoint paths. In *Proc. of 4th LATIN 2000 : Theoretical Informatics*, LNCS 1776, pages 308–317. Springer-Verlag, April 2000. Punta del Este, Uruguay.
- [4] D. Barth and M. Valencia-Pabon. Optimal packet routing schedules on linear arrays and ring networks under various communication models. Technical Report 1171, LRI, Université Paris-Sud, Centre d'Orsay, 1998.
- [5] D. Barth and M. Valencia-Pabon. Message scheduling on trees under a generalized line-communication model. In *Proc. of the 4th Int. Symp. on Parallel Architectures, Algorithms and Networks, I-SPAN'99*, pages 10–15. IEEE CS Press, Juin 1999. Fremantle-Australia.
- [6] B. Beauquier. All-to-all communication in some wavelength-routed all-optical networks. *Networks*, 33(3) :179–187, 1999.
- [7] B. Beauquier, J-C. Bermond, L. Gargano, P. Hell, S. Pérennes, and U. Vaccaro. Graph problems arising from wavelength-routing in all-optical networks. In *Proc. of IPPS'97, 2nd IEEE Workshop on Optics and Computer Science (WOCS)*. IEEE CS Press, 1997.
- [8] L. Becchetti, M. Di Ianni, and A. Marchetti-Spaccamela. Approximation algorithms for routing and call scheduling in all-optical chains and rings. In *Proc. of FSTTCS'99*, LNCS 1738, pages 201–212. Springer-Verlag, 1999.
- [9] A. Bel-Hala. Congestion optimale du plongement de l'hypercube  $H(n)$  dans la chaîne  $P_{2^n}$ . *RAIRO, Info. Théor. et Applic.*, 27(5) :465–481, 1993.
- [10] A. Bel-Hala. *Communications dans les réseaux d'interconnexion : plongements optimaux de l'hypercube*. PhD thesis, Université Bordeaux I, 1995.
- [11] C. Berge. *Graphes*. Gauthiers-Villars, 1983.
- [12] J-C. Bermond, L. Gargano, S. Perennes, A. Rescigno, and U. Vaccaro. Efficient collective communication in optical networks. In *Proc. of 23th Internatio-*

- nal Colloquium on Automata, Languages and programming (ICALP'96)*, LNCS 1099, pages 574–585. Springer-Verlag, 1996.
- [13] S. Bhatt, G. Bilardi, G. Pucci, A. Ranade, A. Rosenberg, and E. Schwabe. On bufferless routing of variable length messages. In *Proc. of ESA'93*, LNCS 726, pages 49–60. Springer-Verlag, 1993.
- [14] Ph. Biane. Permutations suivant le type d'excédance et le nombre d'inversions et interprétation combinatoire d'une fraction continue de heine. *Eur. J. Comb.*, 14(4) :277–284, 1993.
- [15] B. Bollobàs. *Graph Theory. An Introductory Course*. Number 63 in Graduate Texts in Mathematics. Springer-Verlag, 1979.
- [16] I. Caragiannis, Ch. Kaklamanis, and P. Persiano. Wavelength routing of symmetric communication requests in directed fiber trees. In *Proc. of the 5th International Colloquium on Structural Information and Communication Complexity (SIROCCO'98)*. Carleton Scientific, 1998.
- [17] S. Corteel. *Problèmes énumératifs issus de l'Informatique, de la Physique et de la Combinatoire*. PhD thesis, Université Paris-Sud, Janvier 2000.
- [18] S. Corteel, M. Valencia-Pabon, D. Gardy, D. Barth, and A. Denise. The permutation-path coloring problem on trees. Technical Report 1256, LRI, Université Paris-Sud, Centre d'Orsay, juillet 2000. À paraître dans *Theoretical Computer Science*.
- [19] H.E. Daniels and T.H.R. Skyrme. The maximum of a random walk whose mean path has a maximum. *Adv. Appl. Probab.*, 17 :85–99, 1985.
- [20] J. de Rumeur. *Communications dans les réseaux de processeurs*. Études et recherches en informatique. Masson, 1994.
- [21] X. Deng, P. Hell, and J. Hung. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM J. Comput.*, 25(2) :390–403, 1996.
- [22] T. Erlebach and K. Jansen. Call scheduling in trees, rings and meshes. In *Proc. of the 30th Hawaii International Conference on Systems Sciences HICSS-30*, volume vol. 1, pages 221–222. IEEE CS Press, 1997.
- [23] T. Erlebach, K. Jansen, Ch. Kaklamanis, M. Mihail, and P. Persiano. Optimal wavelength routing on directed fiber trees. *Theoret. Comput. Sci.*, 221(1-2) :119–137, 1999.
- [24] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multi-commodity flow problems. *SIAM J. Comput.*, 5(4) :691–703, 1976.
- [25] Ph. Flajolet. Combinatorial aspects of continued fractions. *Discrete Math.*, 32 :125–161, 1980.
- [26] H. Fleischner. *Eulerian graphs and related topics*. Number 45 in Annals of Discrete Mathematics vol. 2. Elsevier Science Publisher B.V., 1991.
- [27] M. Garey and D. Johnson. *Computers and Intractability : A guide to the theory of NP-completeness*. W.H. Freeman and Company, 1979.

- [28] M.R. Garey, D.S. Johnson, G.L. Miller, and Ch. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM J. Alg. Disc. Meth.*, 1(2) :216–227, 1980.
- [29] L. Gargano, P. Hell, and S. Perennes. Coloring all directed paths in a symmetric tree with applications to WDM routing. In *Proc. of 24th International Colloquium on Automata, Languages and programming (ICALP'97)*, LNCS 1256, pages 505–515. Springer-Verlag, 1997.
- [30] M.C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.
- [31] M.C. Golumbic and R.E. Jamison. The edge intersection graphs of paths in a tree. *J. Comb. Theory Series B*, 38(1) :8–22, February 1985.
- [32] Q-P. Gu and H. Tamaki. Routing a permutation in the hypercube by two sets of edge-disjoint paths. In *Proc. of 10th International Parallel Processing Symposium (IPPS)*. IEEE CS Press, 1996.
- [33] U.I. Gupta, D.T. Lee, and J. Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12 :459–467, 1982.
- [34] C.-T. Ho and S.L. Johnsson. Embedding meshes in boolean cubes by graph decomposition. *J. of Parallel and Distributed Computing*, 8(4) :325–339, 1990.
- [35] D.S. Hochbaum, editor. *Approximation Algorithms for NP-hard problems*. PWS Publishing Company, 1997.
- [36] I. Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4) :718–720, 1981.
- [37] K. Jansen. Approximation results for wavelength routing in directed trees. In *Proc. of IPPS'97, 2nd IEEE Workshop on Optics and Computer Science (WOCS)*. IEEE CS Press, 1997.
- [38] I.A. Karapetyan. Coloring of arc graphs (in russian). *Akad. Nauk Armyan. SSR Dokl.*, pages 306–311, 1980. Armenian summary.
- [39] M. Kaufmann and J.F. Sibeyn. Optimal multi-packet routing on the torus. In *Proc. of 3rd Scandinavian Workshop on Algorithm Theory (SWAT'92)*, LNCS 621, pages 118–129. Springer-Verlag, 1992.
- [40] M. Kaufmann and J.F. Sibeyn. Randomized multipacket routing and sorting on meshes. *Algorithmica*, 17 :224–244, 1997.
- [41] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Doklady*, 20 :191–194, 1979.
- [42] S.R. Kumar, R. Panigrahy, A. Russel, and R. Sundaram. A note on optical routing on trees. *Inf. Process. Lett.*, 62 :295–300, 1997.
- [43] V. Kumar. Approximating circular arc coloring and bandwidth allocation in all-optical ring networks. In *Proc. of APPROX'98*, LNCS 1444, pages 147–158. Springer-Verlag, 1998.
- [44] R. Lassaigne and M. de Rougemont. *Logique et Complexité*. Hermès, 1996.

- [45] F.T. Leighton. *Introduction to Parallel Algorithms and Architectures : Arrays • Trees • Hypercubes*. Morgan Kaufman, 1991.
- [46] F.T. Leighton, B.M. Maggs, and S. Rao. Universal packet routing algorithms. In *Proc. of the 29th IEEE Symp. on Foundations of Computer Science*, pages 256–271, 1988.
- [47] F.T. Leighton, B.M. Maggs, and S. Rao. Packet routing and job-shop scheduling in  $O(\text{congestion} + \text{dilation})$  steps. *Combinatorica*, 14 :167–186, 1994.
- [48] F.T. Leighton, B.M. Maggs, and A.W. Richa. Fast algorithms for finding  $O(\text{congestion} + \text{dilation})$  packet routing schedules. Technical Report CMU-CS-96-152, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.
- [49] G. Louchard. Random walks, gaussian processes and list structures. *Theoret. Comput. Sci.*, 53 :99–124, 1987.
- [50] D.W. Matula and F. Shahrokhi. The maximum concurrent flow problem and sparsest cuts. Technical report, Southern Methodist University, March 1986.
- [51] D. Minoli. *Telecommunications Technology Handbook*. Artech House, 1991.
- [52] B. Mukherjee. *Optical Communication Networks*. McGraw-Hill series on computer communications, 1997.
- [53] R.K. Pankaj. *Architectures for Linear Lightwave Networks*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1992.
- [54] R.K. Pankaj and R.G. Gallager. Wavelength requirements of all-optical networks. *IEEE/ACM Trans. on Networking*, 3 :269–280, 1995.
- [55] Ch. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.
- [56] A. Ranade, S. Schleimer, and D.S. Wilkerson. Nearly tight bounds for wormhole routing. In *Proc. of the 35th IEEE Symp. on Foundations of Computer Science*, pages 347–355, 1994.
- [57] Ch. Scheideler. *Universal Routing Strategies for Interconnection Networks*. Number 1390 in Lectures Notes in Computer Science. Springer-Verlag, 1998.
- [58] R. Sedgewick and Ph. Flajolet. *Introduction à l'analyse des algorithmes*. International Thomson Publishing France, 1996.
- [59] W.-K. Shih and W.-L. Hsu. An  $O(n^{1.5})$  algorithm to color proper circular arcs. *Discrete Appl. Math.*, 25 :321–323, 1989.
- [60] S. Stahl.  $n$ -tuple colorings and associated graphs. *J. Comb. Theory Series B*, 20 :185–203, 1976.
- [61] A. Symvonis. Routing on trees. *Inf. Process. Lett.*, 57(4) :215–223, 1996.
- [62] T. Szymanski. On the permutation capability of a circuit-switched hypercube. In *Proc. of International Conference on Parallel Processing (ICPP)*, volume 1, pages 103–110, 1989.
- [63] R.E. Tarjan. Decomposition by clique separators. *Discrete Math.*, 55 :221–232, 1985.

- [64] A. Tucker. Coloring a family of circular arcs. *SIAM J. Appl. Maths.*, 29(3) :493–502, 1975.
- [65] M. Valencia-Pabon. Congestion et émulation de réseaux de processeurs. Master's thesis, Université Paris-Sud, Centre d'Orsay, 1996.
- [66] M. Valencia-Pabon. mulation dynamique de réseaux d'interconnexion. In A. Schiper and D. Trystram, editors, *Proc. of RENPAR'97*, pages 135–138, Mai 1997. Lausanne-Suisse.
- [67] M. Valencia-Pabon. Revisiting Tucker's Algorithm to Color Circular-Arc Graphs. Technical Report 1267, LRI, Université Paris-Sud, Centre d'Orsay, 2000.
- [68] X. Viennot. A combinatorial theory for general orthogonal polynomials with extensions and applications. *Lect. Notes Math.*, 1171 :139–157, 1985.
- [69] G. Wilfong and P. Winkler. Ring routing and wavelength translation. In *Proc. of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'98*, pages 333–341, 1998.