Protection par p-cycles dans les réseaux WDM

Hamza Drid* — **Bernard Cousin*** — **Samer Lahoud*** — **Miklos Molnar****

* *Université de Rennes I - IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France*
** *INSA de Rennes - IRISA, 35043 Rennes Cedex, France*

{hdrid, bcousin, molnar, slahoud}@irisa.fr

RÉSUMÉ. *Parmi les mécanismes de protection efficaces proposés pour les réseaux optiques, on trouve la protection par cycles préconfigurés ou p-cycles. La difficulté majeure de ce mécanisme réside dans le calcul de l'ensemble le plus efficace possible des p-cycles protégeant le réseau pour une certaine charge. Les solutions existantes génèrent des p-cycles candidats indépendamment de l'état du réseau, puis un sous-ensemble efficace de p-cycles protégeant le trafic du réseau est sélectionné. La solution que nous proposons dans cet article calcule l'ensemble des p-cycles protégeant le réseau sans passer par l'étape de génération de p-cycles candidats. Notre proposition est basée sur l'agrégation incrémentale des cycles en tenant compte de la topologie et de la distribution du trafic dans le réseau. Cela nous permet de calculer l'ensemble des p-cycles en une seule étape et en fonction des besoins.*

ABSTRACT. *One of the main methods for link protection proposed for optical WDM networks is pre-configured protection cycles (p-cycle). The major challenge of this method of protection resides in finding the most efficient set of p-cycles that protects the network for a given working capacity distribution. Existing heuristics are based on the generation of a large set of candidate p-cycles, which are independent of the network working capacity. Then the sub-set of p-cycles that protects the network is selected. In this paper, we propose an algorithm to compute a set of p-cycles that protects the network without going through the step of candidate p-cycles generation. Our algorithm is based on the incremental aggregation of cycles and takes into account the working capacity of the network. This enables us to compute in one step an efficient set of p-cycles.*

MOTS-CLÉS : *réseaux optiques, protection des réseaux WDM, survie des réseaux, p-cycle, complexité de la gestion du réseau.*

KEYWORDS: *optical networks, WDM protection, network survivability, p-cycle, network management complexity.*

1. Introduction

La tolérance aux pannes dans les réseaux optiques à multiplexage en longueur d'onde (WDM ou *Wavelength Division Multiplexing*) est une propriété très importante. En effet la technologie WDM permet de transporter des quantités d'informations importantes sur la même fibre. Récemment Alcatel-Lucent a réussi à transmettre un débit de 25,6 Tbit/s sur une seule fibre de 160 longueurs d'ondes, chaque longueur d'onde étant capable de transporter les informations à un débit de 160 Gbit/s. Par conséquent, la rupture d'une telle fibre optique peut causer d'énormes pertes de données et des millions de communications peuvent être interrompues. Pour minimiser les pertes, des mécanismes de protection doivent être mis en oeuvre.

Plusieurs mécanismes de protection ont été proposés dans la littérature. Un des principaux mécanismes de protection destiné aux réseaux optiques WDM est la protection par cycles préconfigurés ou p-cycles. Le concept de p-cycle a été introduit par (Grover *et al*., 1998). L'idée de base des p-cycles est inspirée de la protection en anneau. Dans ce type de protection, le réseau est organisé en anneau et les ressources de chaque lien sont divisées en deux : la première moitié des ressources est réservée pour les chemins primaires tandis que l'autre moitié est utilisée en tant que secours. Quand un lien tombe en panne, le trafic est rerouté sur le chemin de secours dans la direction de rotation inverse. A la différence de la protection en anneau, un p-cycle offre une protection non seulement aux liens constituant le p-cycle, mais il protège aussi ses cordes [1] et sans aucun besoin de réserver des ressources supplémentaires sur ces cordes. Cette propriété importante permet de réduire l'ensemble de ressources utilisées pour la protection du réseau.

Le temps de reprise en cas de panne est un critère très important pour tout mécanisme de protection. La protection par p-cycle, comme la protection par anneau, bénéficie d'une très bonne réactivité aux pannes. En effet, les noeuds voisins de la panne vont la détecter et réagir immédiatement en proposant un détour pour les chemins de secours.

La figure 1 décrit un exemple montrant le fonctionnement de la protection avec p-cycle. Sur la figure 1(a), a-b-c-d-e-a est un p-cycle composé de cinq liens, sur chacun de ses liens une longueur d'onde de protection est réservée. Le p-cycle a-b-c-d-e-a est capable de protéger une longueur d'onde primaire [2] sur chacun de ses liens et deux longueurs d'ondes primaires sur chaque corde qui lui appartient. Quand le lien d-c tombe en panne, le p-cycle offre un chemin de protection qui est d-e-a-b-c pour protéger une longueur d'onde sur le lien d-c (cf. Figure 1(b)). Lorsque la corde e-c tombe en panne, le p-cycle protège deux longueurs d'ondes primaires sur ce dernier en offrant deux chemins de protection qui sont ici e-d-c et e-a-b-c. Chacun de ces deux chemins protège une longueur d'onde sur le lien e-c (cf. Figure 1(c)).

1. On appelle corde une arête qui relie deux sommets non consécutifs d'un cycle.
2. Une longueur d'onde primaire est une longueur d'onde réservée pour le chemin primaire.

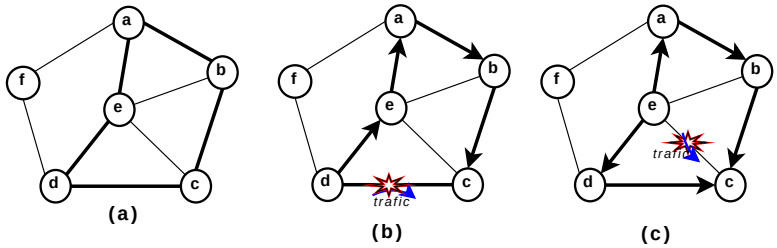

Figure 1. *Exemple de p-cycle*

L'efficacité d'un p-cycle est inversement proportionnelle à sa redondance. La redondance d'un p-cycle est définie comme le rapport entre les ressources utilisées pour la protection et les ressources protégées. Plus ce rapport diminue, plus l'efficacité du p-cycle s'accroît. Autrement dit, un p-cycle à grande efficacité est celui qui protège beaucoup de longueurs d'ondes primaires avec peu de longueurs d'ondes de protection. De plus, l'algorithme de calcul de p-cycles doit offrir un temps de calcul réduit. Cela permettra de tenir compte rapidement de tout changement dans la topologie du réseau. Le problème majeur de ce mécanisme de protection réside dans le calcul du meilleur ensemble possible de p-cycles protégeant le réseau qui est un problème NP-complet (Zhang *et al.*, 2003). Le calcul des p-cycles peut être formulé sous forme d'un problème d'optimisation combinatoire disjointe ou conjointe. Dans la première approche, après avoir construit les chemins primaires (par exemple en utilisant un algorithme de recherche du plus court chemin), l'ensemble de p-cycles efficaces est calculé à partir des ressources restantes (Grover *et al.*, 1998 ; Schupke *et al.*, 2002 ; Doucette *et al.*, 2003 ; Zhang *et al.*, 2003 ; Schupke, 2004 ; Onguetou *et al.*, 2007). Dans l'approche conjointe, les chemins primaires [3] et l'ensemble des p-cycles sont calculés simultanément en minimisant globalement toutes les ressources du réseau (Gruber, 2003 ; Nguyen *et al.,* 2006).

Nous étudions dans cet article les solutions traitant le problème de conception disjointe des p-cycles, en effet la complexité de l'approche conjointe est souvent trop élevée pour être réellement applicable. Comme nous allons voir dans la prochaine section, la majorité des solutions considère les p-cycles ayant plus de cordes comme étant les plus efficaces. En conséquence, la génération de p-cycles utilisée est basée uniquement sur la topologie du réseau indépendamment du volume du trafic du réseau (ressources disponibles, ressources occupées). L'inconvénient majeur de ces solutions réside dans le fait qu'elles génèrent un nombre très élevé de p-cycles candidats de manière non dirigée par les besoins. Dans cet article, nous proposons une nouvelle heuristique de génération de p-cycles qui tient compte de l'état du trafic du réseau et qui génère un nombre de p-cycles restreint tout en restant très efficaces. La suite de

3. Un chemin primaire est le chemin sur lequel les données de la communication circulent lorsqu'il n'y a pas de panne. Le chemin dit de secours sera utilisé lorsqu'une panne survient sur l'un des éléments du chemin primaire.

l'article est organisée comme suit. Dans la section 2, nous décrivons les principales solutions proposées dans la littérature pour résoudre le problème de la conception disjointe des p-cycles. Dans la section 3, nous présentons notre solution qui est basée sur le principe de la fusion incrémentale des cycles. Ensuite, nous analysons les résultats de la simulation obtenus par la comparaison de notre solution avec la solution exacte et les principales heuristiques proposées dans la littérature.

2. Etat de l'art

Plusieurs méthodes de conception des p-cycles ont été proposées dans la littérature. On peut distinguer deux grandes classes de méthodes pour résoudre ce problème d'optimisation : les solutions exactes qui utilisent souvent la programmation linéaire en nombres entiers (ILP) et les heuristiques. La première classe de méthodes résout de manière exacte le problème de calcul des p-cycles. Cette approche est applicable lorsque la taille du réseau est petite, car le nombre de p-cycles pouvant être construits dans le graphe augmente exponentiellement avec le nombre de noeuds et de liens du réseau. Dans la deuxième classe de méthodes, nous trouvons les heuristiques, qui sont à leur tour divisées en deux sous-classes : heuristiques basées sur la programmation linéaire en nombres entiers et les solutions purement heuristiques. Dans la première sous-classe (Zhang *et al*., 2002 ; Grover *et al*., 2002 ; Liu *et al* ., 2004) un ensemble limité de p-cycles est généré, puis la programmation linéaire intervient pour sélectionner les meilleurs p-cycles protégeant le réseau. Cette heuristique présente l'avantage de diminuer la complexité du problème et de faciliter ainsi le travail lors de la résolution du programme linéaire. La deuxième sous-classe correspond aux solutions purement heuristiques (Doucette *et al*., 2003 ; Zhang *et al*., 2003). Ce type de solutions essaie de trouver un ensemble efficace des p-cycles sans recours à la programmation linéaire en nombres entiers. Le but de cette dernière approche est de trouver une solution satisfaisante dans un temps réduit et admissible.

2.1. *Solution exacte*

Ici, nous présentons une solution exacte introduite dans (Grover *et al*., 1998) pour calculer l'ensemble optimal des p-cycles protégeant le réseau. La formulation du problème se base sur la programmation linéaire en nombres entiers. La solution du programme linéaire offre l'ensemble optimal des p-cycles minimisant les ressources utilisées pour la protection en assurant une protection totale des longueurs d'ondes primaires. Le principe de cette solution est de générer, dans un premier temps, tous les cycles élémentaires dans le réseau (modélisé par un graphe). Ensuite, la programmation linéaire intervient pour sélectionner l'ensemble optimal des p-cycles minimisant les ressources de protection sur chaque lien du réseau.

Dans cette formulation, *S* représente le nombre de liens dans le réseau. s_j et n_i sont les variables de ce programme linéaire indiquant respectivement le nombre de longueurs d'ondes de protection sur le lien *j*, le nombre de copies du cycle *i* dans

l'ensemble final des p-cycles. w_i indique le nombre de longueurs d'ondes primaire sur le lien *j*. $p_{i,j}$ est un paramètre binaire indiquant si le p-cycle *i* passe par le lien *j* ou non. $p_{i,j}$ prend la valeur 1 si le cycle *i* passe par le lien *j*, sinon il prend la valeur 0. *P* désigne le nombre de p-cycles candidats. $X_{i,j}$ est un paramètre indique le nombre de chemins que le p-cycle *i* peut fournir au lien *j* lorsque ce dernier tombe en panne. c_j est le coût du lien *j*.

$$X_{i,j} = \begin{cases} 0 & Si\ le\ pcycle\ i\ n'offre\ aucun\ chemin\ de\ secours\ au\ lien\ j. \\ 1 & Si\ le\ lien\ j\ est\ un\ element\ du\ pcycle\ i(i\ offre\ 1\ chemins \\ & de\ secours\ au\ lien\ j). \\ 2 & Si\ le\ lien\ j\ est\ une\ corde\ du\ pcycle\ i(i\ offre\ 2\ chemins \\ & de\ secours\ au\ lien\ j). \end{cases}$$

Fonction d'objectif :

$$Minimiser \quad \sum_{j=1}^{S} c_j s_j \qquad [1]$$

Contraintes :

$$s_j = \sum_{i=1}^{P} P_{i,j} \times n_i \qquad \forall j = 1, 2, ...S \qquad [2]$$

$$w_j \leq \sum_{i=1}^{P} X_{i,j} \times n_i \qquad \forall j = 1, 2, ...S \qquad [3]$$

$$n_i \geq 0 \qquad \forall i = 1, 2, ...P \qquad [4]$$

Pour que ce programme soit solvable, le graphe doit être b-connexe, sinon la contrainte [3] ne peut être satisfaite et le programme n'aura pas de solution. Cette condition est nécessaire pour toute solution qui cherche une protection totale de toutes les longueurs d'ondes primaires.

Cependant, la limitation principale de cette solution est qu'elle exige que l'ensemble de p-cycles candidats contienne tous les cycles élémentaires existants dans le réseau. Le nombre de cycles élémentaires peut être très grand dans le cas des réseaux à

grande échelle. Par conséquent, le temps de calcul de la solution optimale devient intolérable. Pour surmonter ce problème, des heuristiques ont été proposées pour calculer une solution (c.-à-d. un ensemble de p-cycles protégeant le réseau) qui est proche de l'optimal dans des temps de calcul raisonnables. Ces solutions, comme nous avons dit précédemment, peuvent être classées en deux classes : heuristiques basées sur l'ILP et les pures heuristiques.

2.2. *Heuristiques*

La première classe des heuristiques génère un ensemble limité de p-cycles candidats puis la programmation linéaire intervient sur cet ensemble limité pour sélectionner l'ensemble de p-cycles protégeant le réseau. Cette première classe d'heuristiques peut donner des solutions efficaces mais elle reste toujours limitée en terme de temps de calcul surtout quand il s'agit d'un réseau à grande échelle. Afin de réduire la taille de l'ensemble de p-cycles candidats, une heuristique a été présentée par (Grover *et al.* 2002) où les auteurs proposent une métrique pour sélectionner les cycles candidats efficaces dans le réseau. Dans ce contexte, un cycle efficace est défini comme un cycle qui peut potentiellement protéger un nombre très grand de communications avec peu de ressources. À cet effet, une métrique a été introduite pour sélectionner l'ensemble de cycles candidats, appelé PE (*a priori efficiency*).

$$PE(i) = \frac{\sum_{\forall j \in E} X_{i,j}}{\sum_{\forall j \in E / X_{i,j}=1} c_i} \qquad [5]$$

Ici, *E* est l'ensemble des liens du réseau, *j* est un lien qui appartient à *E*. Dans cette heuristique le choix des cycles se fait seulement en fonction de la topologie du réseau et sans tenir compte de la distribution du trafic. En effet, les p-cycles ayant une grande valeur de PE sont considérés comme étant les p-cycles les plus efficaces.

(Zhang *et al.*, 2002) ont proposé l'heuristique Straddling Link Algorithm (*SLA*). L'objectif principal de cet algorithme est de générer un ensemble relativement réduit de p-cycles candidats qui peut conduire, en utilisant la programmation linéaire en nombres entiers, à une solution efficace en termes d'utilisation de ressources et en temps de calcul. Les p-cycles de *SLA* contiennent une seule corde. *SLA* construit ces p-cycles de la manière suivante : pour chaque lien *j* du graphe, *SLA* cherche les deux plus courts chemins disjoints reliant les deux noeud d'extrémités de *j*. Puis ces deux chemins sont concaténés afin d'obtenir un p-cycle dont *j* est la corde. *SLA* est rapide quand on le compare à la solution exacte et à la solution présentée dans (Grover *et al.*, 2002) mais l'ensemble des p-cycles candidats n'est pas très efficace, car il est limité par le nombre de liens du réseau et ces p-cycles contiennent généralement peu de cordes.

(Doucette *et al.*, 2003) ont proposé Capacitated Iterative Design Algorithm (*CIDA*). L'algorithme de *CIDA* a pour objectif d'améliorer l'ensemble de p-cycles candidats générés par SLA et pour se défaire de la programmation linéaire en nombres entiers. Cet algorithme passe par deux étapes. La première étape consiste à générer l'ensemble des p-cycles candidats. La génération des p-cycles est basée sur la transformation des p-cycles fournis par *SLA* en p-cycles plus efficaces. Trois algorithmes de transformation ont été proposés. Leur principe consiste à remplacer un lien d'un p-cycle de *SLA* par un chemin reliant les deux noeuds d'extrémité de ce lien. Après cette transformation un nouveau p-cycle candidat est obtenu, qui contient une nouvelle corde. Dans la deuxième étape, les p-cycles les plus efficaces sont sélectionnés un par un. Chaque p-cycle sélectionné les longueurs d'ondes primaires protégées par le p-cycle sont enlevées de l'ensemble des longueurs d'ondes des liens à protéger. Le processus de sélection de p-cycles continue jusqu'à ce que la protection maximale du réseau soit atteinte.

L'efficacité réelle (AE) est la métrique utilisée par CIDA pour sélectionner les meilleurs p-cycles. L'efficacité réelle est définie comme suit :

$$AE(i) = \frac{\sum_{\forall j \in E} u_j \times X_{i,j}}{\sum_{\forall j \in E / X_{i,j}=1} c_i} \qquad [6]$$

Ici, u_j désigne le nombre de longueurs d'ondes primaires non encore protégées sur le lien *j*. En pratique, pour que *CIDA* puisse avoir de bons résultats il doit générer un ensemble très grand de p-cycles candidats. L'ensemble des p-cycles candidats générés par *CIDA* possèdent tous des cordes, cela ce n'est pas toujours intéressant en termes d'utilisation de ressources comme nous allons voir dans la prochaine section.

2.3. *Perspectives*

Nous pouvons constater que les solutions que nous venons de décrire passent par deux étapes pour calculer l'ensemble final des p-cycles. Ces deux étapes sont la génération des p-cycles candidats et la sélection de l'ensemble final de p-cycles. Nous avons mentionné aussi que l'efficacité de la solution finale (le taux d'utilisation de ressources pour la protection) ainsi que le temps de calcul dépend de l'ensemble des cycles candidats. Autrement dit, si l'ensemble des cycles candidats est grand, alors le taux d'utilisation de ressources dans la solution finale est meilleur mais le temps de calcul devient critique et vice-versa.

Pour surmonter ces problèmes nous proposons une solution qui ne dépend pas de l'ensemble des cycles candidats, elle génère des p-cycles en fonction de la distribution des longueurs d'ondes primaires dans le réseau. Cela nous permet de construire des p-cycles efficaces adaptés à la topologie et à la distribution du trafic. Notre proposition est basée sur l'agrégation incrémentale des cycles. Cela consiste premièrement à

choisir un cycle sans corde et en tenant compte de la topologie et la distribution courante du trafic dans le réseau, nous agrégeons ce dernier avec d'autres cycles voisins existants dans le réseau afin d'obtenir un cycle efficace.

3. Heuristique proposée

3.1. *Modélisation du système*

Nous modélisons un réseau optique WDM par un graphe connexe *G =(V, E)*, où les noeuds du graphe, $v \in V$ représentent les commutateurs optiques et les arêtes, $j \in E$ les fibres optiques. Chaque fibre optique *j* possède w_j longueurs d'ondes primaires. w_j est obtenu après avoir décidé le routage de toutes les demandes. Nous supposons que le réseau possède une capacité de conversion totale, c'est-à-dire tous les routeurs dans le réseau sont capables de convertir toutes les longueurs d'ondes. Nous considérons également que tous les liens du réseau sont bidirectionnels. Ces hypothèses sont réalistes car très généralement les câbles à fibres optiques contiennent effectivement un nombre suffisant de fibres (utilisation dans un sens ou l'autre) pour offrir des communications bidirectionnelles.

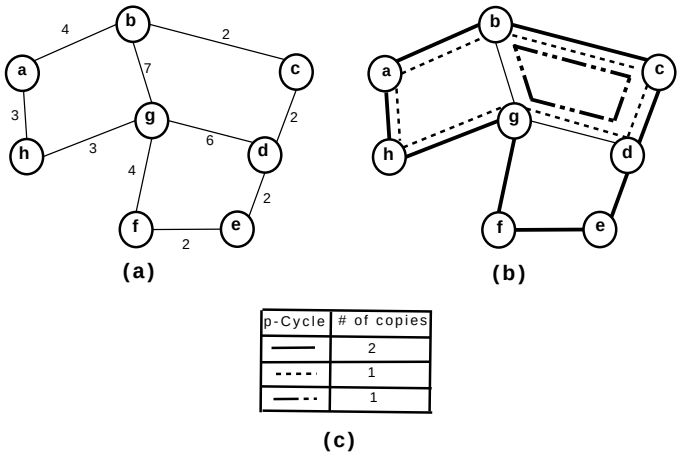

Figure 2. *Ensemble optimal de p-cycles protégeant le réseau*

La figure 2 (a) montre un réseau WDM avec huit commutateurs optiques. Sur chacun des liens du réseau le nombre de longueurs d'ondes primaires à protéger est indiqué. La figure 2 (b) présente un ensemble des p-cycles protégeant les longueurs d'ondes primaires du réseau. La première colonne du tableau 2 (c) contient l'ensemble final de protection qui est composé de trois p-cycles. La deuxième colonne de la table représente le nombre de copies de chaque p-cycle.

Dans ce travail, nous prenons l'hypothèse qu'une seule panne peut se produire à la fois dans le réseau. Il s'agit de la forme de panne la plus fréquente dans les ré-

seaux optiques WDM. En effet, la probabilité que deux pannes indépendantes arrivent simultanément est négligeable (Onguetou *et al.,* 2007). C'est une hypothèse qui est faite dans tous les travaux sur les p-cycles.

3.2. *Motivations de la proposition*

Les heuristiques existantes, celles qui se basent sur la programmation linéaire en nombres entiers ainsi que les heuristiques pures, utilisent une approche à deux étapes. La première étape consiste à générer l'ensemble de p-cycles candidats puis la sélection des p-cycles protégeant le réseau vient dans la deuxième étape. Par conséquent, la qualité de la solution dépend fortement de l'ensemble de p-cycles candidats. Plus le nombre de p-cycles candidats s'accroît plus la possibilité d'améliorer la solution finale augmente. Cependant, ceci mène à augmenter le temps de calcul de la solution finale considérablement, ce qui présente une limitation effective de ces approches. Dans cet article nous proposons une solution qui s'adapte aux besoins de la protection du trafic en calculant uniquement les p-cycles nécessaires à la protection. Cela nous permet de dépasser les limitations de l'approche à deux étapes.

Afin de réduire l'ensemble des p-cycles candidats, la plupart de ces solutions considèrent les p-cycles ayant le plus de cordes comme étant les plus efficaces. Cependant le choix de ce type de p-cycles (les p-cycles avec beaucoup de cordes) n'est pas toujours avantageux. Considérons, par exemple, le cas où un p-cycle contient plusieurs cordes et, en même temps, contient plusieurs liens sans longueur d'onde à protéger. Dans ce cas, ce type de p-cycle n'est pas efficace car le rapport entre les ressources utilisées pour la protection et les ressources protégées est très grand (4 longueurs d'ondes de protection pour protéger 3 longueurs d'ondes primaires).

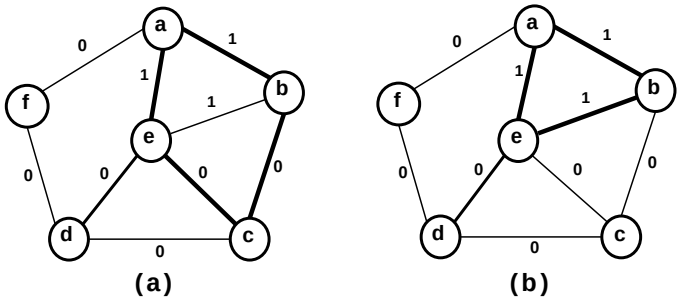

Figure 3. *Exemple sur l'utilité des p-cycles sans corde*

En prenant l'exemple de la figure 3, supposons que les liens e-c et c-b sont déjà protégés (il n'y a plus de longueur d'onde primaire à protéger sur ces liens). Dans ce cas, le p-cycle (a-e-c-b-a) de la figure 3(a) n'est pas efficace même s'il contient une corde. Il serait intéressant de construire le p-cycle (a-e-b-a) de la figure 3(b) (qui ne contient aucune corde). Dans notre heuristique, nous surmontons cet inconvénient

en prenant en considération la distribution du trafic dans le réseau (i.e., les longueurs d'ondes restant à protéger).

3.3. *Objectifs de la proposition*

Le premier objectif, qu'on vise à atteindre dans notre solution, est de réduire la valeur de la redondance. La redondance constitue un critère important permettant de mesurer l'efficacité d'une solution en terme de consommation de ressources : elle désigne le rapport entre les ressources utilisées pour la protection et les ressources protégées. Dans notre heuristique, nous contrôlons la redondance de chaque p-cycle au fil des itérations de l'algorithme pour que la redondance globale du système de protection soit faible.

Nous désignons la redondance globale du réseau par R. Par extrapolation, nous définissons la redondance du cycle *i*, notée par R(*i*), comme étant le rapport entre le nombre de longueurs d'ondes utilisées pour la protection (les longueurs d'ondes sur les liens du cycle) et le nombre de longueurs d'ondes protégées par le cycle (les longueurs d'ondes protégées sur les liens et sur les cordes du cycle). De même, la redondance R de la solution finale est définie comme le rapport entre le nombre de longueurs d'ondes utilisées pour la protection et le nombre de longueurs d'ondes protégées dans tout le réseau.

$$R(i) = \frac{\sum_{\forall j \in E} p_{i,j}}{\sum_{\forall j \in E} \pi_{i,j}} \qquad [7]$$

$$R = \frac{\sum_{\forall j \in E} s_j}{\sum_{\forall j \in E} w_j} \qquad [8]$$

Ici, $p_{i,j}$ indique si le lien *j* appartient au cycle *i*. $p_{i,j}$ prend la valeur 1 si *j* appartient à *i* ; $\pi_{i,j}$ représente le nombre de longueurs d'ondes que le cycle *i* peut effectivement protéger sur le lien *j*. Si *j* est une corde du cycle *i*, alors $\pi_{i,j}$ peut prendre 3 valeurs :

$$\pi_{i,j} = \begin{cases} 0 & S'il\ n'y\ a\ aucune\ longueur\ d'onde\ \ proteger\ sur\ le\ lien\ j. \\ 1 & S'il\ y\ a\ une\ seul\ longueur\ d'onde\ \ proteger\ sur\ le\ lien\ j. \\ 2 & S'il\ y\ a\ au\ moins\ deux\ longueurs\ d'ondes\ sur\ le\ lien\ j\ qui \\ & reste\ \ proteger. \end{cases}$$

Si *j* appartient au cycle *i* alors $\Pi_{i,j}$ peut prendre les valeurs suivantes :

$$\pi_{i,j} = \begin{cases} 1 & S'il\ y\ a\ des\ longueurs\ d'ondes\ primaires\ \ proteger\ sur\ le\ lien\ j. \\ 0 & Sinon. \end{cases}$$

Pour la redondance globale du réseau, la variable s_j indique le nombre de longueurs d'ondes utilisées pour la protection sur le lien *j*. w_j désigne le nombre de longueurs d'ondes primaires sur le lien *j*.

Le nombre de p-cycles dans l'ensemble final de protection est un deuxième critère d'évaluation que nous avons pris en considération dans notre solution. L'intérêt de ce critère vient du fait que les solutions qui génèrent un nombre restreint de p-cycles dans la solution finale de protection simplifie la gestion de réseau (Onguetou *et al.,* 2007). Par exemple, considérons le cas où les concepteurs de réseau mettent en oeuvre manuellement la configuration des p-cycles. Dans ce cas là, les solutions avec un nombre réduit de p-cycles facilitent la configuration et les mises à jour du réseau, réduisent le risque d'erreur pendant la configuration.

La solution, que nous proposons, calcule l'ensemble des p-cycles protégeant le réseau en une seule étape en tenant compte de l'état du réseau. De plus, notre algorithme tient compte explicitement des deux critères d'efficacité que nous avons introduits : la redondance et le nombre de p-cycles. Dans le cas où la mise en oeuvre est automatique, un nombre réduit permet de minimiser le délai de la configuration.

3.4. *Description de l'algorithme*

Dans cette section nous présentons notre algorithme de calcul de p-cycles. Nous allons décrire l'algorithme d'une manière générale et dans la section suivante nous expliquons plus en détail les motivations des choix effectués. Dans une phase préliminaire de notre méthode, nous calculons l'ensemble des plus courts cycles dans le réseau avec l'aide de l'algorithme présenté dans (Leeuwen, 1990). Un plus court cycle est un cycle élémentaire qui ne possède aucune corde et qui ne peut pas être obtenu par l'agrégation des autres cycles. Un exemple d'ensemble de plus courts cycles du réseau est montré dans la figure 4. Cet ensemble de cycles va nous servir pour construire l'ensemble final de p-cycles protégeant les longueurs d'ondes primaires du réseau, comme nous allons voir par la suite.

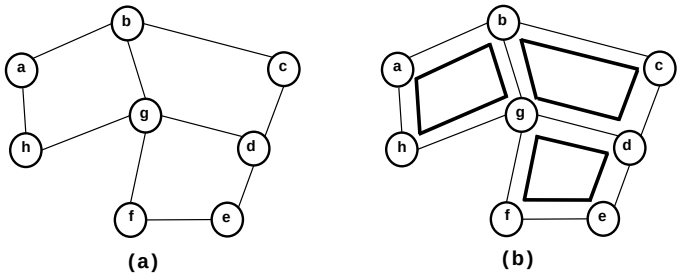

Figure 4. *Ensemble des plus courts cycles*

Les itérations de l'algorithme permettent de construire des p-cycles qui s'adaptent aux besoins de la protection des longueurs d'ondes primaires. Chaque itération de

notre algorithme commence par choisir un plus court cycle parmi tous les plus courts cycles dans le réseau. Typiquement, on choisit le cycle contenant le lien avec un nombre minimal de longueurs d'ondes primaires non encore protégées (cf. figure 5, étape 4). Nous désignons ce cycle par *c'*. Puis nous recherchons un plus court cycle qui va être agrégé avec le cycle *c'*. L'ensemble de cycles éligibles qui peuvent être agrégés avec le cycle *c'* sont les plus courts cycles qui remplissent les deux conditions suivantes : le plus court cycle *c* et le cycle *c'* partagent un seul lien entre eux et ils ne partagent aucun noeud sauf les noeuds aux extrémités du lien partagé (cf. figure 5, étape 5).

Nous calculons la redondance de chaque cycle obtenu par l'agrégation du cycle *c'* et d'un plus court cycle de l'ensemble des cycles éligibles. L'agrégation consiste à fusionner deux cycles voisins de façon à former un seul cycle avec une corde supplémentaire. Puis nous sélectionnons le cycle agrégé qui a la plus petite redondance parmi toutes les agrégations considérées. Nous comparons la redondance de ce cycle agrégé avec la redondance initiale de *c'* : si la redondance du cycle agrégé est inférieure à celle de *c'*, le processus d'agrégation continue en considérant cette fois-ci le cycle agrégé comme le cycle initial et il sera désormais noté par *c'*.

Le processus d'agrégation continue avec le nouveau cycle *c'* et il s'arrête quand l'ensemble de cycles éligibles est soit vide ou soit aucune agrégation ne mène à une réduction de redondance. L'arrêt du processus d'agrégation correspond à la fin d'une itération : le p-cycle obtenu sera ajouté à l'ensemble final des p-cycles de protection. Les longueurs d'ondes protégées par le p-cycle produit ne seront plus considérées dans les prochaines itérations. L'algorithme prend fin quand la protection totale du réseau est atteinte c'est-à-dire lorsque toutes les longueurs d'ondes primaires sont protégées.

3.5. *Explication de l'algorithme*

Notre premier objectif est de surmonter les problèmes liés à l'approche à deux étapes, qui commence dans un premier temps par la génération d'un ensemble de p-cycles candidats puis la sélection de l'ensemble final des p-cycles dans un deuxième temps. Notre idée pour éviter l'étape de la génération des cycles candidats est simple : nous construisons des p-cycles en tenant compte la topologie et les longueurs d'ondes primaires à protéger simultanément. Dans ce contexte, nous pouvons remarquer que tout p-cycle dans le réseau peut être construit à partir d'un ensemble limité des cycles élémentaires sans corde (ou plus courts cycles). Par conséquent, dans une étape préliminaire de notre algorithme, nous calculons l'ensemble des cycles les plus courts du réseau qui seront utilisés dans la construction des p-cycles. Afin de construire l'ensemble des p-cycles protégeant le réseau, nous proposons d'utiliser l'agrégation incrémentale des plus courts cycles. Le principe de l'agrégation incrémentale consiste à choisir un cycle et de l'agréger successivement avec d'autres plus courts cycles voisins pour construire un p-cycle. Chaque agrégation donne un nouveau cycle élémentaire (cf. Figure 5, étape 5, condition b) qui contient une corde supplémentaire (cf. Figure

Algorithme : generation de p-cycles

Input : Ensemble des plus courts cycles, longueurs d'ondes.
Output : Ensemble final de p-cycles

(1) $\forall j \in E, u_j = w_j$;
(2) Itération, i= 1 ;
(3) $\forall j, k \in E$, sélectionner *j* tel que $0 \leq u_j \leq u_k$;
(4) Sélectionner le plus court cycle c' contenant j
(tel que c' a le plus grand nombre de liens non protégés parmi
tous les plus courts cycles partageant le lien *j*) ;
(5) Sélectionner l'ensemble des plus courts cycles SSC={ c_k / k >0 } tel que chaque
cycle vérifie les deux conditions suivantes :
a. Les cycles c' et c_k partagent un seul lien entre eux ;
b. Les cycles c' et c_k ne partagent aucun noeud sauf les noeuds du lien partagé ;
(6) Si (SSC $\neq$ vide) alors
a. Sélectionner le cycle c_k qui a la plus petite valeur de R(c'+c_k) ;
b. Si (R(c'+ c_k) $\leq$ R(c')) alors c' :=c'+c_k et aller à l'étape **(5)** ;
(7) Pour chaque lien *j* protégé par le p-cycle *i* nous mettons à jour la valeur de u_j :
a. u_j=u_j-1 pour chaque lien *j* du cycle c' ;
b. u_j=u_j-2 pour chaque corde *j* du cycle c' ;
(8) S'il y encore des liens j∈ E avec $u_j > 0$ *i* = *i* + 1 et aller à l'étape **(3)** ;

w_j : Désigne le nombre de longueurs d'ondes primaires sur le lien *j*.
u_j : Désigne le nombre de longueurs d'ondes primaires restant à protéger sur le lien *j*.
(x+y) : Désigne l'agrégation des cycles *x* et *y*.
R(c) : Désigne la redondance du cycle *x*.

Figure 5. *Description de l'algorithme*

5, étape 5, condition a). La corde additionnelle permet d'améliorer la redondance du cycle et ainsi la redondance globale du réseau. Avec cette technique d'agrégation, nous pouvons construire des cycles avec le nombre maximum de cordes mais pour les raisons que nous avons citées dans la section 3.1 concernant l'efficacité des p-cycles, notre solution construit des p-cycles avec un nombre approprié de cordes suivant la quantité de ressources non encore protégées. En particulier, nous ne construisons que des p-cycles qui réduisent la redondance.

Notre idée pour diminuer le nombre de p-cycles dans l'ensemble final de protection vient du fait que si on choisit le même cycle initial dans plusieurs itérations successives on aura une forte probabilité de construire le même p-cycle dans chaque itération. En effet, notre sélection du cycle initial dépend d'un critère qui fixe le choix du cycle initial pour plusieurs itérations successives. Ce critère consiste à sélectionner un lien avec un nombre de longueurs d'ondes primaires non encore protégées minimal et en fonction de ce lien nous sélectionnons un des cycles qui le contient. Quand la construction du p-cycle atteint sa fin, nous enlevons toutes les longueurs d'ondes primaires protégées par ce dernier et nous passons à la prochaine itération (i.e., la construction du prochain p-cycle). A l'itération suivante, les liens qui peuvent avoir des u_j minimaux sont les cordes du p-cycle précédent ou le lien qu'on a choisi dans

la précédente itération. Cela veut dire que le cycle que nous allons choisir dans cette itération est probablement celui qui a été choisi dans la précédente itération.

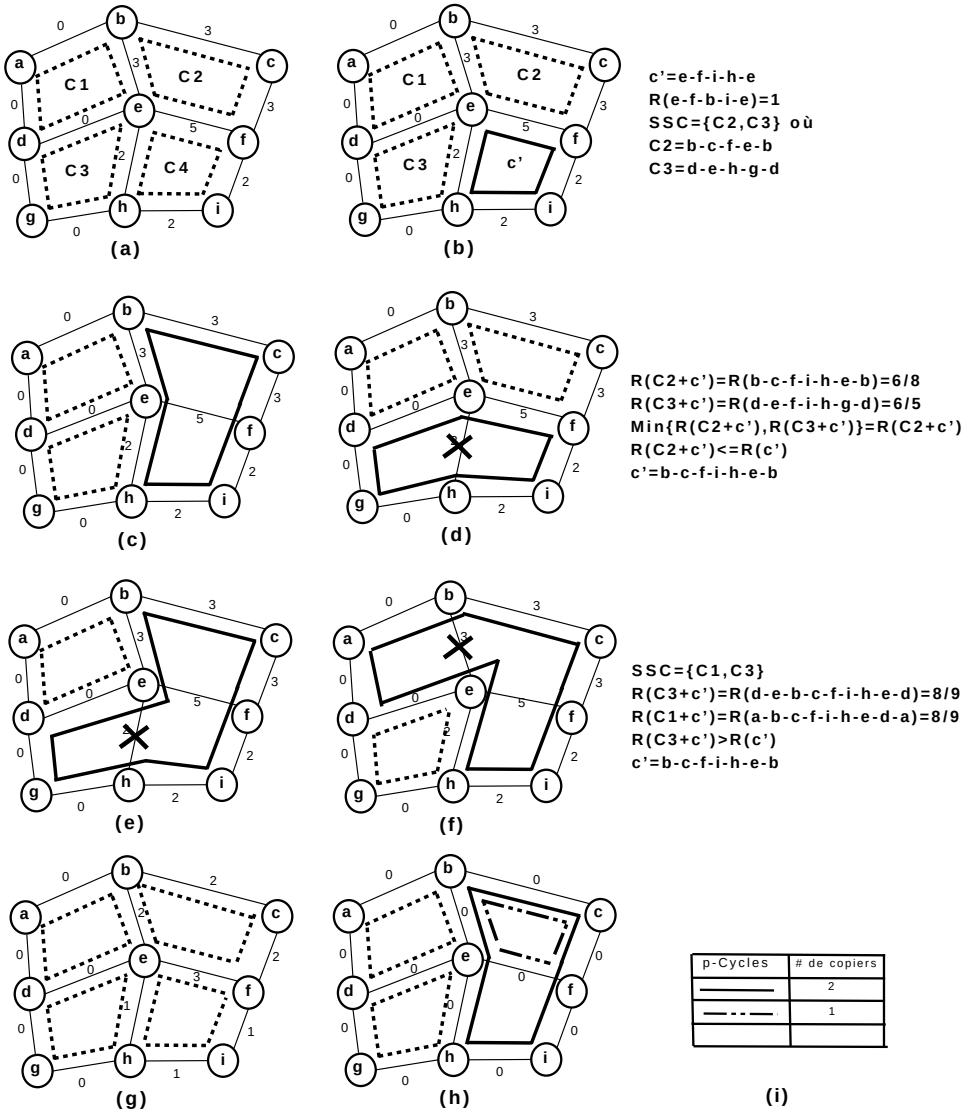

Figure 6. *Processus de construction d'un p-cycle*

La figure 6 montre les différentes phases du processus de construction de p-cycle. La figure 6(a) montre l'état du réseau après l'exécution de plusieurs itérations de l'algorithme (quelques liens n'ont aucune longueur d'onde à protéger), et aussi l'ensemble des plus courts cycles (C1, C2, C3, C4) que nous utilisons pour construire les

p-cycles protégeant le réseau. À ce stade, l'algorithme choisit le cycle (e-f-i-h-e) qui contient le lien avec le nombre minimal de longueurs d'ondes primaires non encore protégées. Ce cycle est nommé c' (cf. la figure 5, étape 3 et 4).

L'ensemble des cycles éligibles SSC obtenu après la sélection du cycle c' contient deux plus courts cycles : (b-c-f-e-b) et (d-e-h-g-d) (cf. figure 5, étape 5). Nous calculons la redondance de chaque cycle obtenu par l'agrégation du cycle c' et un plus court cycle de l'ensemble SSC : R(b-c-f-i-h-e-b) et R(d-e-f-i-h-g-d). Nous choisissons le cycle avec la plus petite valeur de la redondance, qui est (b-c-f-i-h-e-b) dans notre cas. Nous comparons la redondance du cycle choisi (b-c-f-i-h-e-b) à celle du cycle c'. Comme la redondance de (b-c-f-i-h-e-b) est inférieure à la redondance du c' (cf. figure 5, étape 6), alors le cycle (b-c-f-i-h-e-b) devient le nouveau c'.

La redondance du nouveau cycle c' est égale à 0.75. Cette dernière ne pourra pas être améliorée, même si nous agrégeons c' avec ses voisins C1, C3 comme il est présenté dans la figure 6 (e) et (f). Comme il n'y aucune amélioration du cycle c' alors le processus d'agrégation prend fin et le cycle obtenu est ajouté dans l'ensemble final de p-cycles. Les longueurs d'ondes protégées par le p-cycle obtenu ne seront pas considérées dans les prochaines itérations comme le montre la figure 6 (g) (cf. figure 5, étape 7). La table de la figure 6 (i) montre le nombre de copies de chaque p-cycle à l'issue de l'algorithme et protègeant toutes longueurs d'ondes primaire de la figure 6(a).

4. Simulations

Dans cette section, nous évaluons notre algorithme en termes de redondance, de temps de calcul ainsi qu'en nombre de p-cycles utilisés dans l'ensemble final de p-cycles protégeant le réseau. Les topologies que nous avons choisies pour évaluer les performances de notre solution sont les topologies les plus utilisées dans la littérature : la topologie COST239, la topologie KL et celle des Etats Unis. Elles offrent une assez grande variété de topologies. La première est une topologie composée de 11 noeuds et 26 liens, prise de (Onguetou *et al*., 2007). La deuxième est composée de 15 noeuds et 28 liens, prise de (Marzo *et al*., 2007). La topologie des Etats Unis est la plus grande topologie qui est composée de 28 noeuds et 45 liens, prise de (Liu *et al*., 2004).

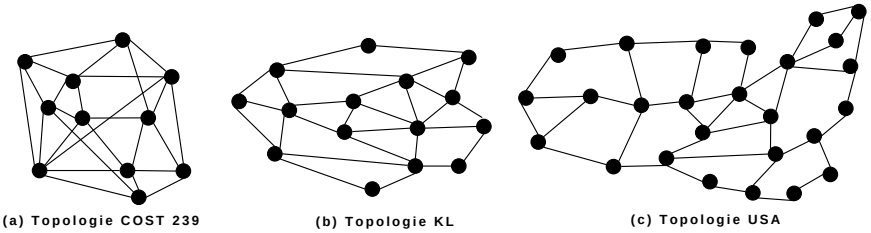

Figure 7. *Topologies de tests*

Caractéristiques topologiques \ Topologies	COST 239	KL	USA
Nombre de noeuds	11	15	20
Nombre de liens	26	28	45
Degré moyen	4.73	3.73	3.21
Nombre de cycles élémentaires	3531	1600	7321
Nombre de cycles courts	19	14	18

Tableau 1. . *Caractéristiques de COST 239, KL et USA*

	COST 239	KL	USA
Nombre de connexions	276.7	524.9	1911.3
Longueurs d'ondes primaires	422.2	1118.1	6432

Tableau 2. *Trafic moyen de COST 239 et KL*

Nous pouvons constater que la topologie COST 239 ainsi que celle des Etats Unis possèdent un nombre très élevé de cycles élémentaires, cela est dû au degré élevé des noeuds ainsi qu'à la taille de la topologie. Les trois topologies sont celles de la figure 7.

Le modèle de simulation que nous avons utilisé dans cet article pour évaluer les performances de notre solution est pris de l'article présenté dans (Onguetou et al., 2007). Les demandes de connexions sont distribuées uniformément entre chaque paire de noeuds source-destination du réseau. Pour chaque paire source-destination, un nombre entier entre zéro et le nombre maximal de demandes de connexions est aléatoirement généré selon une loi uniforme (le nombre maximal de connexions vaut 10 dans notre simulation). Après avoir routé toutes les demandes générées entre toutes les paires source-destination en utilisant le plus court chemin, nous obtenons le nombre de longueurs d'ondes primaires sur chaque lien du réseau. Dix expériences sont effectuées pour chaque topologie et les valeurs moyennes sont présentées dans le tableau 2.

Dans toutes nos simulations nous supposons que la capacité des liens du réseau est suffisante pour satisfaire toutes les demandes de connexions ainsi que pour assurer une protection totale du réseau. Cela est réaliste car le dimensionnement du réseau permet d'obtenir cette suffisance. Ces hypothèses sont conformes avec celles faites par les articles présentés dans l'état de l'art. Toutes nos simulations ont été effectuées sur une machine DELL Quadri Dual Core Xeon (3.4 GHz) avec une mémoire centrale de 4 giga-octets de RAM, le système d'exploitation utilisé est Windows server2003. Pour résoudre la solution exacte, utilisant la programmation linéaire en nombres entiers, nous avons utilisé la bibliothèque Matlog du logiciel MATLAB.

Pour évaluer notre solution en terme de redondance et ainsi qu'en terme de nombre de p-cycles dans la solution finale, nous comparons ses performances avec la solution

exacte présentée dans la section 2. Nous comparons également notre algorithme avec les deux heuristiques bien connues CIDA et SLA présentées précédemment. CIDA génère dans une première étape un ensemble limité de p-cycles candidats, puis il choisit l'ensemble de p-cycles protégeant le réseau en utilisant l'efficacité réelle comme critère de sélection. SLA passe aussi par une étape de génération des p-cycles, mais l'ensemble qu'il génère est très petit. L'ensemble de p-cycles protégeant le réseau est sélectionné en utilisant la programmation linéaire. Notez que dans la solution optimale, tous les cycles sont considérés comme des p-cycles candidats par le programme linéaire.

Solutions \ Topologies	**COST 239**	**KL**	**USA**
Solution optimale	72.5%	78.2%	-
Solution CIDA	89.4%	91.6%	98.9%
Solution SLA	98.4%	101%	105.35%
Notre solution	83.8%	88.9%	96.88%

Tableau 3. *Redondance moyenne*

À partir des résultats résumés dans le tableau 3, nous pouvons constater que la valeur de la redondance obtenue par notre solution est plus petite que les valeurs obtenue par les deux heuristiques CIDA et SLA. Notre redondance est à 11%, 10% de la solution optimale, respectivement pour les deux topologies COST 239 et KL. Les explications que nous donnons à cette efficacité est que dans notre approche, la construction des p-cycles est basée sur l'agrégation incrémentale des cycles, qui consiste à choisir un cycle et à l'agréger avec d'autres cycles sous certaines conditions topologiques et de trafic. Ce qui nous laisse contrôler la redondance de chaque p-cycle que nous construisons au fil des itérations de l'algorithme. Autrement dit, chaque nouveau p-cycle construit possède une petite valeur de redondance. Cela conduit à réduire la redondance globale du réseau. Nous constatons aussi que la solution exacte n'a pas pu produire une solution pour la topologie des Etats unis. La simulation a dû être suspendue au bout d'une dizaine de jours. Cela est dû à la taille importante de cette topologie ainsi qu'au nombre élevé de longueurs d'ondes primaires présents dans le réseau. Cela illustre parfaitement la nécessité d'utiliser les heuristiques.

Solutions \ Topologies	**COST 239**	**KL**	**USA**
Solution optimale	127728	11356	-
Solution CIDA	0,98	0,12	3,1
Solution SLA	159	8,2	401,9
Notre solution	0,07	0,09	2,53

Tableau 4. *Temps moyen de calcul (en s)*

Nous pouvons remarquer également que notre temps de calcul est très petit par rapport aux autres solutions. Cela est dû au fait que notre algorithme calcule uniquement les p-cycles nécessaires à la protection sans passer par une étape préliminaire de génération de p-cycles candidats.

Solutions \ Topologies	**COST 239**	**KL**	**USA**
Solution optimale	23,4	16	-
Solution CIDA	27,5	30	98
Solution SLA	17	13	24
Notre solution	21	25	66

Tableau 5. *Nombre moyen de p-cycles*

Comme le nombre de p-cycles dans l'ensemble final de p-cycles est une propriété importante pour la gestion et la configuration du réseau (Onguetou et al., 2007), nous évaluons également notre algorithme en terme de nombre de p-cycles distincts dans l'ensemble final des p-cycles. Un nombre restreint de p-cycles dans l'ensemble final de protection signifie une gestion facile du réseau. Les résultats de la simulation montrent que la solution finale qu'on obtient avec notre algorithme possède un nombre restreint de p-cycles. Cela est dû aux choix que nous avons présentés précédemment. Nous pouvons remarquer que SLA produit un petit nombre de p-cycles, nous expliquons ça par le fait que l'ensemble des p-cycles candidats utilisé par SLA est petit, en conséquence l'ensemble final des p-cycles sera relativement petit. On note toutefois que SLA a pris un temps de calcul plus élevé que notre solution et la redondance de la solution produite par SLA est moins bonne que notre solution.

5. Conclusion

Dans cet article, nous avons étudié la protection dans les réseaux optiques WDM en utilisant les p-cycles. C'est une technique de protection par cycles préconfigurés. Nous avons traité le problème de calcul de l'ensemble optimal des p-cycles protégeant le réseau. Après avoir décrit la solution exacte ainsi que les principales heuristiques proposées dans la littérature et montré leurs inconvénients (notamment le processus de sélection de l'ensemble de p-cycles qui est en deux phases, et le fait qu'ils ne tiennent pas compte de l'état du réseau), nous avons proposé un nouvel algorithme qui est basé sur l'agrégation incrémentale des plus courts cycles. Notre génération des p-cycles tient compte de l'état du trafic du réseau (c'est-à-dire du nombre de longueurs d'ondes primaires à protéger sur chaque lien du réseau). Cela nous permet de ne générer que des p-cycles avec une faible redondance et de générer un nombre très petit de p-cycles.

Les résultats de la simulation ont montré d'une part la difficulté, voire l'incapacité, de trouver des solutions exactes lorsqu'il s'agit de topologies de taille importante, et d'autre part l'intérêt de l'utilisation des heuristiques. Nous avons constaté aussi que la redondance obtenue par notre heuristique est meilleure que celle des autres heuristiques et le nombre de p-cycles produits par notre algorithme est petit par rapport au nombre de p-cycles produits par CIDA. Tout cela dans un temps de calcul extrêmement bref. Cette étude est très encourageante, nous allons poursuivre nos investiga-

tions en incluant d'autres contraintes telles que la taille des p-cycles. Car cela a une certaine influence sur le délai.

6. Bibliographie

Doucette J., He D., Grover W. D., Yang O., « Algorithmic Approaches for Efficient Enumeration of Candidate p-Cycles and Capacitated », *In Proc.of the Fourth International Workshop on the Design of Reliable Communication Networks.*p. 212-220, 2003.

Grover W. D., Doucette J., « Advances in Optical Network Design with p-Cycles : Joint optimization and pre-selection of candidate P-cycles, », *In Proc.of IEEE-LEOS Summer Topical Meeting on All Optical Networking.*p. 49-50, 2002.

Grover W. D., Stamatelakis D., « Cycle-Oriented Distributed Preconfiguration : Ring-like Speed with Mesh-like Capacity for Self-planning Network Restoration », *In Proc.of IEEE International Conference on Communication.*p. 537-543, 1998.

Gruber C. G., « Resilient Networks With Non-Simple p-Cycles », *In Proc. of the International Conference on Telecommunications*, 2003.

Liu C., Ruan L., « Finding Good Candidate Cycles for Efficient p-Cycle Network Design », *In Proc. 13 th International Conference on Computer Communication and Networks.*p. 321-326, 2004.

Marzo J. L., Calle E., Vilà P., A. U., « Performance evaluation of minimum interference routing in network scenarios with protection requirements », *Computer Communications.*p. 3161-3168, 2007.

Nguyen H. N., Habibi D., Phung V. Q., Lachowicz S., Lo K., Kang B., « Joint Optimization in Capacity Design of Networks with p-Cycle Using the Fundamental Cycle Set », *In Proc. of IEEE GLOBECOM.*, 2006.

Onguetou D.P.and Grover W., « p-Cycle Network Design : from Fewest in Number to Smallest in Size », *In Proc. of the 6 th International Workshop on Design and Reliable Communication Networks.*, 2007.

Schupke D., « An ILP for Optimal p-Cycle Selection without Cycle Enumeration », *In Proc. of the Eighth Working Conference on Optical Network Design and Modelling.*, 2004.

Schupke D., Gruber C., Autenrieth A., « Optimal configuration of p-cycles in WDM networks », *In Proc. IEEE International Conference on Communication.*p. 2761-2765, 2002.

Van Leeuwen J., *Algorithms and Complexity*, Handbook of Theoretical Computer Science., 1990.

Zhang H., Yang O., « Finding Protection Cycles in DWDM Networks », *In Proc. IEEE International Conference on Communication.*p. 2756-2760, 2002.

Zhang Z., Zhong W., Mukherjee B., « A Heuristic Algorithm for p-Cycles Configuration in WDM Optical Networks », *In Proc. of the Opto-Elecrtoinics and Communications Conference.*p. 568-569, 2003.

Hamza Drid est doctorant à l'université de Rennes 1. Il est membre d'une équipe de recherche au sein de l'IRISA. Ses travaux portent sur la protection des réseaux optiques à multiplexage en longueur d'onde (WDM). Il s'intéresse plus particulièrement aux problèmes de protection et de passage à l'échelle dans les réseaux multi-domaines.

Bernard Cousin est professeur à l'université de Rennes 1. Il est responsable d'une équipe de recherche sur les réseaux informatiques au sein de l'IRISA. Ses principaux domaines d'étude sont les réseaux à haut débit, l'ingénierie de trafic, le routage multicast, la gestion de la qualité de service, la protection et la sécurité des réseaux. Ses travaux portent sur IPv6, les réseaux d'accès sans fil et les réseaux tout optiques, et tout particulièrement sur l'amélioration de l'administration et l'optimisation du contrôle de ces réseaux.

Samer Lahoud a obtenu en 2006 une thèse de doctorat de l'ENST Bretagne et de l'Université de Rennes 1. Après une année passée au sein du laboratoire de recherche d'Alcatel-Lucent, il devient en 2007 enseignant chercheur à l'Université de Rennes 1 et membre du laboratoire Irisa. Ses recherches portent sur le routage, la qualité de service et l'ingénierie de trafic dans les réseaux de nouvelle génération.

Miklós Molnár habilité à diriger les recherches, est maître de conférences au Département Informatique de l'INSA de Rennes. Il mène sa recherche à l'IRISA sur des problèmes algorithmiques et d'optimisation combinatoire liés aux réseaux.