Adaptation d'algorithmes multicast aux réseaux optiques

Alexandre Guitton, Raymond Marie et Miklos Molnar

Afin de pouvoir réaliser des communications *multicast* dans les réseaux de télécommunications traditionnels, de nombreux algorithmes utilisant la notion d'arbre couvrant existent déjà. Il est possible de trouver dans la littérature de nombreux travaux à ce sujet, tant au niveau théorique (bornes théoriques d'approximation, complexité précise des algorithmes) qu'au niveau pratique (étude de performance, comparaison des algorithmes). L'apparition des réseaux entièrement optique apporte plusieurs contraintes propres qu'il est nécessaire de gérer lors de la conception d'algorithmes. Cette étude décrit trois schémas d'adaptation et propose une étude comparative étendue de ces différents modèles.

Keywords: Routage *multicast*, réseaux WDM, méta algorithmes, heuristiques, problème de Steiner sous contraintes.

1 Introduction

Dans les réseaux de télécommunication traditionnels, une communication *multicast* peut être réalisée par un arbre de diffusion unique couvrant la source et les différents membres du groupe *multicast*. Dans la littérature, plusieurs de ces algorithmes *multicast* ont déjà été largement étudiés et comparés.

L'apparition des réseaux tout optique réduit le temps de traitement des informations par rapport aux réseaux traditionnels. Ce nouveau type de réseau, très prometteur en terme de débit et de délai, apporte cependant plusieurs contraintes. Plus précisément certains routeurs entièrement optiques ne sont pas capables de transférer un signal d'entrée vers plusieurs sorties. Cette contrainte sera appelée par la suite «contrainte de duplication» et les routeurs sont dits respectivement de type I ou de type II selon qu'ils ne peuvent pas ou qu'ils peuvent dupliquer. Afin de construire des arbres *multicast* adaptés aux réseaux tout optique, il est nécessaire de prendre en compte ces contraintes, c'est-à-dire qu'il est important d'avoir à disposition des heuristiques *multicast* adaptées aux réseaux tout optique. La manifestation la plus évidente de ces contraintes est qu'il n'est pas toujours possible de réaliser un unique arbre optique couvrant une source et tous les membres. Il faut dans le cas général recourir à une forêt optique, dont chaque composante couvrira la source et une partie des membres du groupe. Nous nous proposons de présenter trois schémas d'adaptation permettant de faire évoluer un algorithme conçu pour les réseaux traditionnels en un algorithme pour les réseaux tout optique.

Les protocoles *multicast* basés sur des arbres peuvent en outre être classés en deux catégories : ceux dont les arbres sont basés sur une source (*source based tree*) et ceux qui utilisent un arbre partagé par plusieurs sources (*shared tree*) existent. Les méthodes pour offrir des solutions optimales ou approchées varient en fonction des objectifs. Dans la plupart des cas, on cherche soit un arbre qui contient des chemins de longueur minimale à partir d'une source soit un arbre de longueur totale minimale (arbre minimal de Steiner).

L'arbre des plus courts chemins est calculable en temps polynomial, tandis que le calcul des arbres minimaux de Steiner est NP-difficile. Pour calculer des arbres approchés de la solution optimale du problème de Steiner, des heuristiques de faible complexité sont utilisées dans les protocoles. En général, plus un algorithme garantit un petit ratio d'approximation, c'est-à-dire plus la solution obtenue est efficace, plus son temps d'exécution est élevé.

La plupart des protocoles basés sur une source proposent la construction d'un arbre des plus courts chemins allant de la source aux destinations. En général cet arbre est très favorable pour minimiser les délais dans les feuilles (si la métrique correspond au délai des liens considéré constant) mais il peut être coûteux. On ne peut pas borner le ratio d'approximation au problème de Steiner dans le cas d'un arbre des plus courts chemins par une constante.

Si l'objectif vise la construction des arbres de longueur minimale, des heuristiques offrant des arbres approchés de la solution optimale sont utilisées. Les algorithmes les plus souvent appliqués sont des approximations dont le ratio est borné par 2, comme l'heuristique de Takahashi et Matsuyama (cf. [TM81]), celle de Kruskal (cf. [Kru56]) ou encore celle de Kou, Minkowski et Berman (cf. [KMB81]). Ces algorithmes donnent un arbre au plus deux fois plus long que l'optimum avec une complexité très raisonnable.

Dans le deuxième paragraphe, nous décrirons trois schémas d'adaptation. Ces adaptations permettent de conserver une partie des travaux déjà effectués sur les algorithmes *multicast* dans le cadre des réseaux traditionnels. Dans le troisième paragraphe, nous comparerons ces diverses adaptations. Nous terminerons en récapitulant les avantages et inconvénients des adaptations proposées.

2 Adaptation des algorithmes

Ce paragraphe propose trois schémas d'adaptation. Dans les différentes figures, un nœud carré représentera un nœud de type I, un nœud rond représentera un nœud de type II et un nœud grisé un membre du groupe *multicast*. Les différents arbres construits seront représentés par des flèches.

2.1 Schéma d'adaptation par post-traitement

En général, les arbres construits par les algorithmes pour les réseaux traditionnels ne peuvent pas être utilisés dans les réseaux tout optique où peu de nœuds ont la possibilité de dupliquer les messages. Plus formellement, si un nœud de branchement de l'arbre construit est incapable de dupliquer les messages, l'arbre ne peut pas être réalisé tel quel dans le réseau. L'idée du premier schéma d'adaptation est de transformer cet arbre invalide en une forêt respectant les contraintes des nœuds de type I. Il s'agit d'un post-traitement puisqu'il n'est appliqué qu'à l'issue du déroulement de l'algorithme.

La racine est capable d'émettre autant d'exemplaires du message qu'il est nécessaire. Un premier post-traitement peut être décrit comme suit : on extrait de l'arbre chaque chemin de la source à une destination, et on fusionne ces arbres deux à deux tant que c'est possible. La forêt obtenue, valide pour les contraintes optiques, couvre les mêmes arêtes que l'arbre initial. Cette méthode est identique à celle proposée par Zhang et al. (cf [ZWQ00]) pour l'heuristique *Reroute-to-Source*. Cette solution est facile à implémenter mais elle conduit à une grande utilisation des arêtes proches de la source, comme le montre la figure 1. L'algorithme permettant de transformer un arbre en une forêt valide est le suivant :

Entrée : un arbre couvrant une source s et un ensemble de destinations D

Sortie : une forêt F couvrant s et D, valide pour les contraintes optiques

$F \leftarrow$ l'ensemble des chemins dans l'arbre de s à D

tantque il existe dans F deux arbres A_1 et A_2 tels que $A_1 \cup A_2$ vérifie les contraintes optiques **faire**

$F \leftarrow F \setminus \{A_1, A_2\}$

$F \leftarrow F \cup \{A_1 \cup A_2\}$

fintantque

Un traitement particulier est proposé dans [ZGM03a]. Ce traitement s'applique à une forêt des plus courts chemins et tente de réduire le nombre d'arbres créés (qui peut être très important en pratique). Il procède en unifiant des arbres de la façon suivante : si deux arbres T_x et T_y ont des feuilles x et y pouvant dupliquer suffisamment proches (la longueur $d(x,y)$ du plus court chemin valide entre x et y est plus petite que la distance dans l'arbre T_x de x à la source et que la distance dans l'arbre T_y de y à la source), les deux arbres sont reliés par un plus court chemin $path(x,y)$, et la partie inutile de la boucle introduite est supprimée. Cette connexion permet de diminuer le nombre d'arbres générés et de libérer des arêtes près de la source. Sur la figure 2, on peut voir le résultat de l'application de ce post-traitement à une forêt des plus courts chemins. Les mêmes membres sont couverts par seulement deux arbres. La suppression de chacune des boucles est représentée par des croix sur les liens correspondant. Outre une diminution du nombre d'arbres de la forêt, on peut remarquer que le nombre de canaux de longueur d'onde nécessaires est moindre. La description de l'algorithme suit :

Entrée : une forêt F
Sortie : une forêt F comportant moins d'arbres
  pourtout membre x **faire**
    $y \leftarrow$le plus proche membre de x tel que
    – T_x et T_y sont deux arbres différents
    – x et y sont des feuilles pouvant dupliquer
    si $d(x,y) \leq \min(d_{T_x}(x, source), d_{T_y}(y, source))$ **alors**
      si T_x et T_y peuvent être reliés **alors**
        $T_{xy} \leftarrow T_x \cup T_y \cup path(x,y)$
        retirer la boucle de T_{xy} en supprimant le plus long chemin
        retirer T_x et T_y de la forêt
        ajouter T_{xy} à la forêt
      finsi
    finsi
  finpourtout

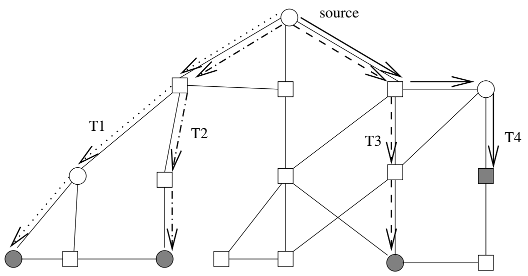

FIG. 1 – *Une forêt des plus courts chemins.*

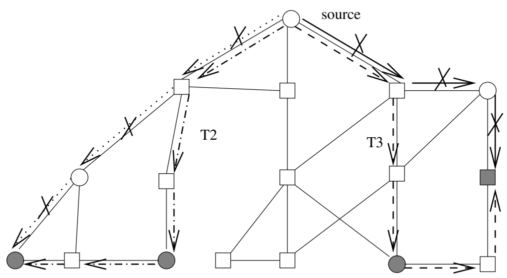

FIG. 2 – *La forêt résultante après application du post-traitement.*

2.2 Schéma d'adaptation par extension

Le deuxième schéma d'adaptation proposé est l'adaptation directe d'algorithmes utilisés dans les réseaux sans contraintes. Pour pouvoir adapter les algorithmes il faut traiter les contraintes directement au sein des algorithmes.

Tous les algorithmes ne sont pas faciles à faire évoluer. Par exemple, étant donné que l'heuristique de Kou et al. calcule l'arbre dans la fermeture métrique du graphe et de cette façon n'utilise pas les détails des connaissances topologiques du graphe, elle ne peut pas être aisément adaptée au problème de l'arbre couvrant partiel minimal sous contraintes de duplication.

Il existe d'autres algorithmes qui ont étés (ou qui peuvent être) modifiés pour satisfaire les objectifs du routage *multicast* dans les réseaux WDM (*Wavelength-Division Multiplexed*, réseaux à multiplexage de longueurs d'onde).

2.2.1 Construction des forêts de plus courts chemins sous contraintes

Les algorithmes de construction des arbres pour lesquels le délai de la source à chacune des destinations est minimale sont très simples : en fusionnant les différents plus courts chemins, on obtient l'arbre souhaité. Comme cela a déjà été mentionné, si l'algorithme doit tenir compte de la capacité de duplication des nœuds, les nœuds de branchement de l'arbre doivent correspondre aux nœuds qui peuvent dupliquer les messages entrants. Une simple adaptation de l'algorithme au problème posé peut être envisagée de la façon suivante. Si le nœud de branchement de l'arbre n'a pas la capacité de duplication, on remonte dans l'arbre vers la racine jusqu'à la source. Cette dernière peut en effet émettre les messages sur des arbres différents.

2.2.2 Construction des forêts de longueur minimale sous contraintes

Pour fournir des solutions approchées au problème de Steiner sous contraintes de duplication, une première adaptation de l'algorithme de Takahashi et Matsuyama a très vite été proposée ; elle est connue en tant qu'algorithme *Member-Only* ([ZWQ00]). Dans cet algorithme, un arbre est initialisé avec la source donnée et les membres du groupe *multicast* sont ajoutés successivement à l'arbre par l'intermédiaire d'un plus court chemin d'un nœud à capacité de duplication libre de l'arbre au membre en question. Si cet ajout n'est pas possible, un nouvel arbre est initialisé depuis la source.

L'heuristique de Kruskal peut elle aussi être appliquée pour construire une forêt assurant la distribution des messages pour un groupe *multicast*. Dans la version d'origine de cette heuristique du problème de Steiner, le point de départ est une forêt où chaque arbre isolé ne contient qu'un seul membre du groupe. À chaque itération, les deux arbres les plus proches sont reliés par le plus court chemin existant entre eux. L'adaptation simple de cette heuristique est proposée dans [Mol99]. Dans cette adaptation, les connexions ne peuvent partir que des nœuds qui possèdent encore la capacité de duplication. Si un arbre reste isolé à la fin de la construction, une liaison entre la source et l'arbre est établie, conduisant à la réalisation d'une forêt.

Pour améliorer la qualité des arbres obtenus, nous avons proposé des modifications de certaines heuristiques. Dans les algorithmes mentionnés ci-dessus, des sous-arbres sont connectés à chaque itération à l'aide de plus courts chemins. La connexion des sous-arbres peut aussi être réalisée sous forme d'un arbre de Steiner plutôt que par un simple chemin. Ce genre de connexions est appelé «connexion utilisant des k-distances» et des heuristiques basées sur ce type de connexion sont décrites dans [Mol99] et [MM99]. Au lieu d'utiliser un plus court chemin pour relier à l'arbre en cours le membre le plus proche, on utilise un «arbre de connexion». Afin de limiter l'espace de recherche, le nombre de nœuds de branchement de cet *arbre de connexion* est limité par k. L'ajout par un arbre entraîne une boucle qu'il est aisé de supprimer pour une valeur de k égale à 1.

Dans les réseaux WDM où les nœuds pouvant dupliquer sont rares, l'application de ces heuristiques présente un double intérêt : non seulement la longueur des arbres peut diminuer par rapport aux arbres construits à l'aide des heuristiques classiques adaptées (comme l'algorithme *Member-Only*), mais les connexions par arbres peuvent profiter de la capacité de duplication des nœuds qui ne sont pas sur les plus courts chemins calculés entre les éléments pendant la construction. L'extension des algorithmes de Takahashi et Matsuyama et de Kruskal sur les connexions utilisant des k-distances en tenant compte des capacités des nœuds sont décrite dans [ZGM03b]. La figure 3 réalise la connexion à un arbre courant d'un membre par une 1-distance. On peut noter que le coût de la connexion n'est que de 6 (un arbre de poids 16 est ajouté à l'arbre en cours tandis qu'un chemin de poids 10 lui est retiré) alors qu'il aurait été de 11 si le membre avait été connecté par un plus court chemin à l'arbre courant.

2.3 Schéma d'adaptation par changements de topologie

Le dernier schéma permettant de prendre en compte les contraintes des nœuds incapables de dupliquer consiste à modifier la topologie du graphe, soit par le biais du poids des arêtes, soit par sa structure.

Dans le cas où l'on ne s'intéresse qu'au changement du poids des arêtes, il est tout de même possible de prendre en compte les contraintes de duplication. Ce type d'algorithme considère à chaque itération que tous les nœuds du réseau sont de type II et applique la connexion de l'algorithme initial. Après chaque connexion d'un nouveau membre à l'arbre courant, le poids de certaines arêtes adjacentes au chemin ajouté peut être augmenté. Cette augmentation prend en considération le fait que certains nœuds impliqués dans ce chemin voient leur capacité à dupliquer épuisée. Dans un cas extrême, on peut considérer que le poids des arêtes est augmenté de $+\infty$, ce qui revient à retirer les arêtes du réseau. On peut néanmoins proposer une augmentation finie ; dans un tel cas, on autorise l'arbre construit à violer la contrainte de duplication. À l'issue du déroulement de l'algorithme, il faudra reconstruire une forêt à partir de l'arbre, de telle sorte que cette forêt respecte les contraintes initiales imposées. L'heuristique dite *directe* décrite dans [GM03] utilise

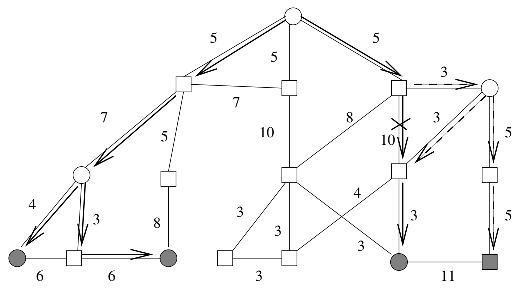

FIG. 3 – *Un exemple de connexion d'un membre à un arbre par une* 1*-distance.*

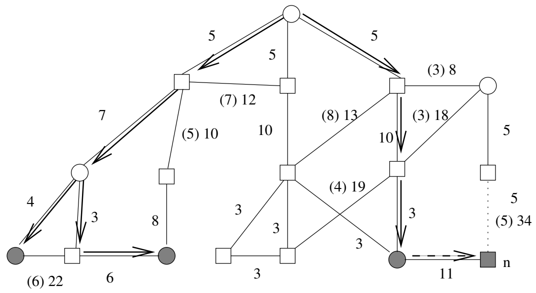

FIG. 4 – *Une itération du schéma par modification de la topologie.*

ce schéma d'adaptation appliqué à l'heuristique de Takahashi et Matsuyama. Le coût des arêtes adjacentes à un nœud n de type I du chemin est augmenté de la distance $d_{T,G}(source,n)$ du plus court chemin de la source au nœud, à laquelle on a ajouté une valeur fixée α correspondant à une pénalité. Cet algorithme peut être écrit de la manière suivante :

Entrée : un graphe G, une source s, un ensemble de membres D, un paramètre α
Sortie : une forêt F couvrant s et D
  $\bar{G}$ ←ignorer-incapacité-à-dupliquer(G)
  tantque $D \neq \emptyset$ **faire**
    (m,T) ←connexion($\bar{G},T,D$)
    pourtout nœud n appartenant au chemin ajouté, à l'exception de l'extrémité de départ **faire**
      si n est incapable de dupliquer alors **alors**
        augmenter le poids dans $\bar{G}$ de toutes les arêtes adjacentes à n qui ne sont pas dans T de la valeur $\alpha + d_{T,G}(source,n)$
      finsi
    finpourtout
  fintantque

La figure 4 présente une itération de cette heuristique ayant pour paramètre $\alpha = 0$. Les valeurs indiquées entre parenthèses correspondent aux valeurs initiales. Lors de la connexion du dernier membre, étant donné qu'il est incapable de dupliquer, le poids de l'arête qui en part (représentée en pointillés) va passer de 5 à $5+\alpha+d_{T,G}(source,n)$, c'est-à-dire à 34. En effet, utiliser cette arête dans les itérations futures va nécessiter la reconstruction d'un arbre partant de la source jusquà ce nœud.

Une autre méthode, un peu plus compliquée, consiste à modifier la topologie du réseau pour faire disparaître les nœuds introduisant des contraintes, l'algorithme à adapter étant réappliqué après chaque transformation de la topologie. Ces nœuds peuvent être supprimés *a priori* c'est-à-dire avant même d'exécuter l'heuristique, ou encore au cours du déroulement de l'heuristique. L'arbre *multicast* résultant va devoir être retransformé en une forêt satisfaisant aux contraintes d'origine en appliquant le procédé de transformation inverse. L'heuristique dite *itérative* décrite dans [GM03] utilise ce schéma d'adaptation appliqué à l'heuristique de Takahashi et Matsuyama. Les transformations se font au cours du déroulement de l'heuristique où seuls les nœuds étant à la fois nœuds de branchement de l'arbre et incapables de dupliquer sont supprimés.

3 Analyse

Nous nous concentrons dans ce paragraphe sur une analyse des trois schémas d'adaptation proposés et sur certaines propriétés que peuvent avoir les algorithmes utilisant ces schémas.

3.1 Complexité induite par les adaptations

Un aspect important de cette analyse est la complexité ajoutée par le schéma, tant au niveau structurel qu'au niveau du temps de calcul supplémentaire introduit.

3.1.1 Complexité structurelle des schémas

Le schéma le plus simple à implémenter est le schéma d'adaptation par post-traitement. Dans sa version simple, il suffit d'ajouter à l'algorithme existant un algorithme transformant un arbre en une forêt valide. Ce dernier peut être le même quelque soit l'algorithme à adapter et s'écrit aisément.

Adapter un algorithme aux contraintes optiques (schéma d'adaptation par extension) est souvent plus délicat : il est parfois difficile de trouver les parties pouvant être conservées. De plus, certains algorithmes (notamment les heuristiques utilisant des k-distances) présentent des singularités dans le cadre des réseaux tout optique qu'il faut considérer. On peut toutefois noter qu'en général, les algorithmes ajoutant à chaque itération un nouveau membre par un chemin s'adaptent facilement.

Le troisième schéma proposé est plus complexe. D'une part il faut spécifier le type de modifications à appliquer à la topologie et d'autre part il faut pouvoir être capable d'appliquer les transformations inverses, pour obtenir à la fin de l'algorithme une forêt valide pour le graphe initial. De plus, le choix des paramètres dépend du graphe initial et influence grandement la qualité de l'algorithme.

3.1.2 Temps de traitements induits

La complexité introduite par le premier schéma est souvent faible. L'algorithme d'adaptation le plus simple peut s'exécuter avec la même complexité qu'un calcul de plus court chemin à partir de la source.

La complexité temporelle ne change pas non plus avec les adaptations du deuxième schéma. En effet, le comportement asymptotique des algorithmes est le même avec et sans adaptation : dans le pire des cas, tous les nœuds du réseau sont de type II et le comportement de l'algorithme n'est pas changé ; quand certains nœuds sont de type I, l'espace de recherche de l'algorithme est réduit, impliquant donc moins de traitements à effectuer.

Le schéma d'adaptation par changement de topologie quant à lui nécessite plus de calculs puisqu'il faut opérer des changements parfois complexes sur le réseau et être capable à la fin de l'algorithme d'appliquer dans l'ordre inverse les différentes modifications.

3.2 Le nombre d'arbres construits

Les algorithmes adaptés en utilisant le deuxième schéma d'adaptation construisent souvent un petit nombre d'arbres. En effet, ils tentent d'agrandir le plus possible l'arbre en construction et ne passent à un nouvel arbre que lorsque toute extension est impossible.

Les algorithmes auxquels on ajoute un traitement ultérieur pour obtenir une forêt respectant les contraintes ont tendance à augmenter le nombre d'arbres construits. En effet, ce traitement ultérieur essaye de construire une forêt valide à partir de l'arbre qu'il a en entrée. Le support initial, c'est-à-dire l'arbre *multicast*, peut utiliser comme nœuds de branchement un grand nombre de nœuds incapables de dupliquer, résultant en la création de plusieurs arbres.

Le comportement des heuristiques modifiant la topologie est intermédiaire. Dans ces heuristiques on s'autorise à construire plus d'arbres, au bénéfice d'autres critères, comme le montrent les figures 5 et 6. Sur la figure 5, on peut voir que le nombre d'arbres crée par une heuristique suivant le schéma par adaptation construit moins d'arbres que celle suivant le schéma modifiant la topologie. Cependant, on peut voir sur la figure 6 que le comportement est inversé pour le nombre de canaux utilisés par les différentes forêts. Cette autorisation est contrôlée par l'augmentation du poids des arêtes, qui est censée reflèter l'ensemble des inconvénients de la construction d'un nouvel arbre utilisant cette arête.

3.3 Le nombre de canaux de longueur d'onde utilisés

Sans introduire la notion de conversion de longueur d'onde, nous considèrerons que le réseau peut contenir des nœuds appelés «sources virtuelles», qui peuvent se comporter comme la source du groupe. Nous observons ici la performance des différents schémas en termes de canaux de longueur d'onde utilisés, tout d'abord en l'absence de sources virtuelles, puis lorsque le réseau possède des sources virtuelles.

3.3.1 En l'absence de sources virtuelles

Dans le cas des algorithmes adaptés selon le schéma d'adaptation par extension, le nombre de canaux nécessaires dépend grandement du ratio d'approximation de l'algorithme de départ à l'optimal. Par exemple,

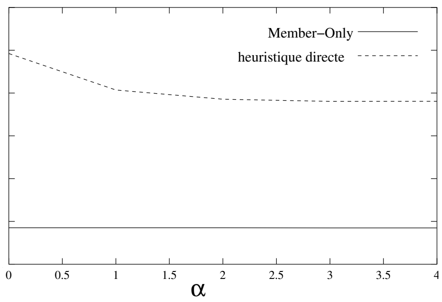

FIG. 5 – *Comparaison entre le nombre d'arbres construits par une heuristique suivant le schéma par adaptation et une heuristique suivant le schéma par modification de la topologie.*

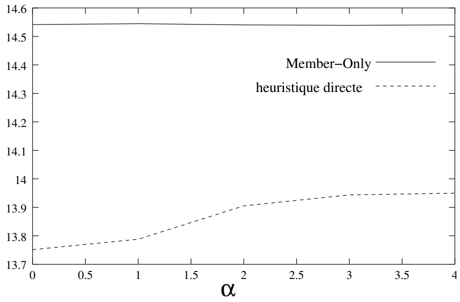

FIG. 6 – *Comparaison entre le nombre de canaux de longueurs d'onde utilisés par les forêts construites par une heuristique suivant le schéma par adaptation et une heuristique suivant le schéma par modification de la topologie.*

s'il s'agit d'une solution heuristique pour le problème de Steiner, le ratio d'approximation pour le problème de Steiner sous contraintes est en relation avec le ratio d'approximation pour le problème de Steiner sans contrainte. La figure 7 présente une comparaison entre le nombre de canaux utilisés par les arbres construits par l'heuristique *Member-Only* et par l'heuristique de Takahashi utilisant les 1-distances, en fonction du pourcentage de nœuds de type II. La comparaison est effectuée sur des graphes réalistes générés aléatoirement comportant 100 nœuds.

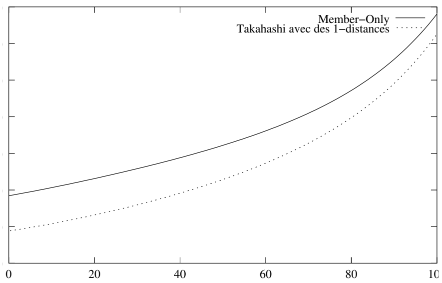

FIG. 7 – *Nombre de canaux de longueurs d'onde pour une heuristique du schéma d'adaptation.*

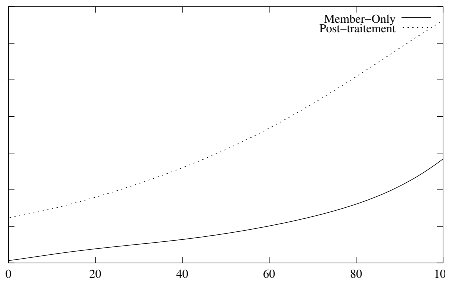

FIG. 8 – *Comparaison du nombre de canaux de longueurs d'onde pour une heuristique du schéma d'adaptation et pour une heuristique du schéma de post-traitement.*

Dans le cas d'algorithmes utilisant le schéma d'adaptation par post-traitement, le nombre d'arbres construit peut être très grand et on ne peut donc pas borner le ratio d'approximation dans le cas général. La figure 8 montre que les performances en terme de nombre de canaux de longueurs d'onde sont supérieures pour une heuristique suivant le schéma d'adaptation (dans l'exemple, *Member-Only*) que pour une heuristique auquel a été appliqué le post-traitement.

On remarque que les algorithmes modifiant la topologie peuvent utiliser moins de canaux de longueur d'onde que d'autres algorithmes. Ce gain est obtenu principalement au prix d'une augmentation du nombre d'arbres à réaliser et dépend grandement de la méthode de changement du réseau et des paramètres choisis.

3.3.2 En présence de sources virtuelles

Le schéma d'adaptation par post-traitement s'adapte facilement à la présence de sources virtuelles. En effet, les forêts construites par ces algorithmes se recouvrent fortement, la superposition des différentes composantes de la forêt formant un arbre. La présence d'une source virtuelle comme nœud de branchement de cet arbre permet de réduire le nombre de canaux nécessaires : au lieu de repartir de la source du groupe, différents arbres peuvent repartir de cette source virtuelle, réduisant ainsi le nombre de canaux nécessités.

Les algorithmes basés sur les deux autres schémas peuvent aussi être adaptés à la présence de sources virtuelles. Au lieu de revenir à la source pour la création de nouveaux arbres, on peut se contenter de retourner à la source virtuelle la plus proche déjà couverte par la forêt *multicast* en cours de construction. L'heuristique *Spawn-from-VS* décrite dans [SMM01] est un exemple d'heuristique adaptée. Il semble cependant que le bénéfice apporté soit moindre, les arbres ayant moins de parties communes.

4 Conclusions

Parmi les technologies récentes du domaine de la télécommunication, les réseaux tout optique possèdent un fort potentiel. Avec l'augmentation de la présence des applications multimédia et multi-utilisateur (jeux massivement multi-joueur, diffusion efficace de la télévision, etc.), la communication *multicast* présente un grand intérêt pour la réduction de l'utilisation des ressources. Les contraintes technologiques et le prix des commutateurs optiques limitent l'applicabilité des algorithmes de routage connus : certains nœuds du domaine optique ne seront pas capables de dupliquer la lumière entrante. Il faut donc adapter les algorithmes de routage *multicast* aux contraintes des réseaux tout optique. Dans notre étude, nous avons mentionné quelques algorithmes souvent utilisés pour la construction de supports *multicast*. Nous avons présenté trois classes de méthodes pour permettre d'adapter les algorithmes de routage *multicast* dans le cadre des réseaux tout optique. Les techniques d'adaptation présentées ne changent pas toutes la complexité asymptotique des algorithmes et rendent possible le calcul des arbres couvrants partiels correspondants aux contraintes du réseau avec un investissement raisonnable au niveau algorithmique. Nous avons de plus présenté une analyse des performances obtenues par ces schémas d'adaptation. Selon le critère privilégié, il peut être intéressant d'appliquer un schéma ou un autre à un algorithme bien connu.

Références

[GM03] A. Guitton and R. Marie. Nouvelles heuristiques de routage *multicast* dans les réseaux optiques basés sur WDM. In *Sciences Électroniques, Technologies de l'Information et des Télécommunications (SETIT 2003)*, March 2003.

[KMB81] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15:141–145, 1981.

[Kru56] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, 16:48–50, 1956.

[MM99] R. Marie and M. Molnár. Arbre couvrant partiel pseudo-optimal pour diffusion multipoint. Technical report, INRIA, 1999.

[Mol99] M. Molnár. Construction of advantageous multicast diffusion trees. In *International Conference on Dynamic Control and Systems (DYCONS 1999)*, August 1999.

[SMM01] N. Sreenath, G. Mohan, and C. Siva Ram Murthy. Virtual source based multicast routing in WDM optical networks. *Photonic Network Communications*, 3(3):217–230, July 2001.

[TM81] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24, 1981.

[ZGM03a] A. Zsigri, A. Guitton, and M. Molnár. Construction of light-trees for WDM multicasting under splitting capability constraints. In *International Conference on Telecommunications (ICT 2003)*, February 2003.

[ZGM03b] A. Zsigri, A. Guitton, and M. Molnár. Two multicast algorithms for sparse splitting capable networks. In *Optical Network Design and Modelling (ONDM 2003)*, volume 1, pages 3–21, February 2003.

[ZWQ00] X. Zhang, J. Wei, and C. Qiao. Constrained multicast routing in WDM networks with sparse light splitting. In *INFOCOM (3)*, pages 1781–1790, 2000.