



**HAL**  
open science

# Arbre couvrant partiel pseudo-optimal pour diffusion multipoint

Raymond Marie, Miklos Molnar

► **To cite this version:**

Raymond Marie, Miklos Molnar. Arbre couvrant partiel pseudo-optimal pour diffusion multipoint. [Rapport de recherche] RR-3636, INRIA. 1999. <inria-00073038>

**HAL Id: inria-00073038**

**<https://inria.hal.science/inria-00073038v1>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Arbre couvrant partiel pseudo-optimal  
pour diffusion multipoint*

Raymond Marie et Miklós Molnár

**N° 3636**

Mars 1999

———— THÈME 1 ————



*rapport  
de recherche*





# Arbre couvrant partiel pseudo-optimal pour diffusion multipoint

Raymond Marie\* et Miklós Molnár†

Thème 1 — Réseaux et systèmes  
Projet Model

Rapport de recherche n 3636 — Mars 1999 — 18 pages

**Résumé :** Dans un réseau de communication, la diffusion multipoint des messages est optimale, du point de vue topologique, si le support de la diffusion correspond à un arbre couvrant partiel minimal. La construction d'arbres couvrants partiels minimaux (le problème de Steiner des réseaux) étant NP-complète, plusieurs heuristiques ne nécessitant qu'un temps polynomial d'exécution ont été formulées. Pour la diffusion multipoint des messages, la longueur de l'arbre utilisé est très importante puisque le coût de la diffusion dépend de cette longueur. Une partie des heuristiques (comme l'heuristique de Takahashi et de Matsuyama) utilise les plus courts chemins du graphe pour construire l'arbre. Dans ce rapport, nous donnons un algorithme de construction de l'arbre approché de l'arbre de Steiner minimal. Dans certains cas, notre algorithme permet de diminuer la longueur de l'arbre construit en temps polynomial. L'algorithme proposé utilise des structures plus avantageuses que les plus courts chemins et fait, si nécessaire, des connexions sous forme d'arbres de Steiner minimaux simples.

**Mots-clé :** Théorie des graphes, problème de Steiner, heuristiques, réseaux à haut débit, diffusion multipoint.

*(Abstract: pto)*

\* {Raymond.Marie}@irisa.fr

† {Miklos.Molnar}@irisa.fr

# Pseudo-optimal Partial Spanning Trees for Multicast Diffusion

**Abstract:** The multicast diffusion in a network is optimal from the topology point of view, if the diffusion support corresponds to a partial minimum spanning tree. On the one hand, the construction of partial minimum spanning trees being NP-complete, several heuristic approximation algorithms with polynomial time have been formulated. On the other hand, the length of the tree is very important since the communication costs depend on this length. Many of the heuristics (like the heuristic of Takahashi and Matsuyama) use the shortest paths to connect the components of the tree. In this paper, we give an approximate construction algorithm of the minimum Steiner tree that can diminish the length of the multicast tree and still be in polynomial time. This construction is based on more advantageous structures than shortest paths and makes connections like simple Steiner trees when necessary.

**Key-words:** Graphs, Steiner problem, Heuristics, High speed networks, Multicast diffusion.

# 1 Introduction

Avec l'essor des réseaux de communications, de plus en plus d'applications multimédia nécessitent la diffusion *multicast* des messages. C'est le cas pour l'Internet mais aussi pour des réseaux orientés connexion à haut débit. Le nombre de services qui demandent la transmission simultanée d'une même séquence de messages à partir d'une source vers plusieurs destinataires est en constante augmentation. Quand les messages correspondent à une émission vidéo et que le groupe des destinataires est important et placé irrégulièrement (voire même aléatoirement) sur le réseau, une diffusion mal choisie peut augmenter considérablement le coût de la diffusion. La situation devient encore plus délicate, s'il s'agit d'une vidéo-conférence où n'importe quel membre du groupe *multicast* peut émettre et recevoir des vidéos. Il est vrai, même pour cette dernière situation, que le routage optimal des messages correspond à un arbre, si les noeuds du réseau sont capables de dupliquer le message entrant pour le diffuser en plusieurs exemplaires. Les membres du groupe pouvant être des serveurs de proximité. Dans le cas idéal, l'arbre de diffusion *multicast* est un des arbres couvrants partiels minimaux du groupe *multicast*.

La construction d'un arbre couvrant partiel minimal d'un groupe de noeuds est connu comme le problème NP-complet de Steiner des réseaux. L'arbre est souvent appelé *arbre minimal de Steiner*. Le temps de calcul demandé par les solutions connues (pour un état de l'art, cf. [1] [2]) dépend d'une façon exponentielle de la taille du problème (nombre de noeuds dans le groupe *multicast* et dimensions du graphe). Jusqu'à présent, la construction exacte de l'arbre pour un cinquantaine de noeuds a pu être réalisée. Dans les cas réels, on utilise des heuristiques fournissant des solutions approchées de l'arbre minimal de Steiner. Une des heuristiques les plus simples est l'algorithme de Takahashi et Matsuyama qui reprend l'idée de Prim pour construire un arbre couvrant. Dans la section 2, nous présentons cette heuristique qui donne un résultat selon un temps polynomial et au plus deux fois plus long que la solution optimale. Le but de ce rapport est de présenter un algorithme aussi simple (dans la mesure du possible) mais qui fournit un arbre plus proche de l'optimum, au moyen d'un paramètre de complexité. Nous proposons ici la connexion des éléments de l'arbre à l'aide des arbres minimaux de Steiner. Les principes de cette proposition sont introduits dans la section 3. Dans la section 4, nous démontrons que, poussé à l'extrême, l'algorithme proposé correspond à la construction exacte de l'arbre minimal de Steiner (naturellement avec un temps généralement prohibitif puisque le problème est NP-complet). C'est dans la section 5 que le lecteur trouvera nos propositions pour une

heuristique qui permet la construction d'arbres approchés de l'arbre minimal de Steiner plus favorables mais toujours selon un temps de calcul raisonnable.

## 2 Rappel de l'Algorithme de Prim

L'algorithme de base proposé par Prim construit un arbre couvrant global minimum. En partant de l'état initial où un seul noeud arbitraire du graphe compose l'arbre couvrant, cet algorithme ajoute à l'arbre le noeud le plus proche jusqu'au dernier noeud, Quand l'ensemble de tous les noeuds doit être couvert, la construction de l'arbre ne cause pas de difficulté majeure. Takahashi et Matsuyama [3] ont adapté l'idée de Prim pour la construction d'un arbre couvrant partiel, quand seul un sous-ensemble de noeuds du graphe est concerné.

Le squelette de l'algorithme est simple :

**Initialisation** : *Initialiser l'arbre en choisissant arbitrairement un noeud du groupe.*

**Jusqu'au dernier noeud du groupe** :

*Calculer les distances des noeuds du groupe non encore choisis avec l'arbre ainsi obtenu.*

*Ajouter à l'arbre courant le noeud le plus proche (en utilisant le plus court chemin entre le noeud et l'arbre).*

L'algorithme, comme tous les algorithmes qui utilisent les plus courts chemins, ne fournit qu'une solution approchée de l'arbre couvrant partiel minimal. L'arbre ainsi construit est au plus deux fois plus long que la solution optimale [1]. Souvent, la solution approchée est relativement proche de l'arbre minimal ; mais dans certains cas, le résultat fourni par une heuristique utilisant les plus courts chemins reste éloigné de la solution optimale (pour des exemples, cf.[4]).

## 3 Idées de base, Définitions

Dans la suite, nous proposons une heuristique qui reprend le scénario de base de l'algorithme de Takahashi et de Matsuyama mais cette fois-ci les noeuds successifs seront connectés à l'arbre courant à l'aide d'arbres minimaux de Steiner.

### 3.1 Idée de base

L'idée est simple. On peut connecter un nœud  $n$  à un arbre  $T$  à l'aide d'un arbre. Cette connexion n'est pas forcément un (plus court) chemin. La connexion sous forme d'un arbre minimum partiel permet la révision d'une partie de l'arbre déjà existant et la réorganisation de l'arbre (effacement des arêtes coûteuses) si nécessaire.

### 3.2 Définitions

Soit  $G = (X, U, V)$  un graphe valué non orienté, connexe dont les arêtes possèdent des longueurs positives; soit  $T$  un arbre couvrant un groupe de nœuds  $X_g$  dans  $G$  et  $n$  un nœud de  $G$  distinct des nœuds de  $T$  (cf. figure 1).

Dans la terminologie des arbres minimaux de Steiner, un  $S$ -nœud est un nœud ayant au moins 3 arêtes adjacentes.

#### **Définition 3.1** Arbre optimal d'association d'un arbre $T$ à un nœud $n$ via un arbre de Steiner à $k$ S-nœuds donné

Soit  $C$  un arbre de Steiner à  $k$  S-nœuds construits sur  $G$  ayant le nœud  $n$  comme feuille et dont les autres feuilles sont toutes des nœuds de  $T$ . De plus,  $C$  et  $T$  n'ont pas d'arête commune. Soit  $\mathcal{G}(T, C)$  le graphe défini comme l'union de  $T$  et de  $C$ . En général,  $\mathcal{G}(T, C)$  n'est plus un arbre car il contient des boucles. Soit  $\tau(\mathcal{G}(T, C))$  l'ensemble des arbres couvrant le groupe de nœuds  $X_g \cup \{n\}$  obtenus en supprimant des arêtes de  $T$  (et non de  $C$ ) incluses dans des boucles.

On appelle arbre optimal d'association d'un arbre  $T$  à un nœud  $n$  via un arbre de Steiner à  $k$  S-nœuds  $C$  l'arbre  $\tau^*(T, C)$  de longueur minimale sur l'ensemble  $\tau(\mathcal{G}(T, C))$ . On note gain d'association de  $C$  avec  $T$  la somme des longueurs des arêtes de  $T$  supprimées dans la construction de l'arbre optimal d'association. On note  $B(T, C)$  l'ensemble des arêtes de  $T$  supprimées dans cette construction optimale.

Remarque: insistons sur le fait que cette définition préserve les arêtes de  $C$ ; le but étant de trouver l'arbre de connexion qui nous permet de supprimer la plus longue partie de  $T$  tout en connectant les membres de  $X_g$  à  $n$ .

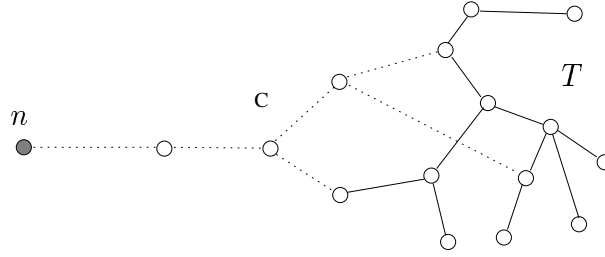


FIG. 1 – Association du noeud  $n$  à l'arbre  $T$  via l'arbre de Steiner  $C$ .

### Définition 3.2 $k$ -distance dans un graphe entre un noeud et un arbre

Soit  $C_k(n, T, G)$  l'ensemble des arbres de Steiner à  $k$   $S$ -noeuds construits sur  $G$  ayant le noeud  $n$  comme feuille et dont les autres feuilles sont toutes des noeuds de  $T$ . On note  $d(x)$  la longueur d'un arbre  $x$  (somme des longueurs de ses arêtes). On appelle  $k$ -distance entre le noeud  $n$  et l'arbre  $T$  :

$$k\text{-distance}(n, T, G) = \min_{C \in C_k(n, T, G)} (d(\tau^*(T, C)) - d(T)) \quad (1)$$

On note  $C_k^*(n, T, G)$  un arbre qui réalise ce minimum.

Par exemple la 0-distance  $(n, T, G)$  est la longueur du plus court chemin entre le noeud  $n$  et l'arbre  $T$  dans  $G$  (c'est la distance habituelle utilisée dans l'algorithme classique de Prim).

### Définition 3.3 $S$ -distance (ou distance de Steiner) entre un noeud et un arbre

On appelle  $S$ -distance entre le noeud  $n$  et l'arbre  $T$  :

$$S\text{-distance}(n, T, G) = \min_k (k\text{-distance}(n, T, G)) \quad (2)$$

On note  $C_S^*(n, T, G)$  un arbre qui réalise ce minimum.

Pour illustrer la  $S$ -distance  $(n, T, G)$ , prenons un exemple simple. Soit le graphe  $G$ , le noeud  $n$  et l'arbre  $T$  qui couvre le groupe  $X_g$  donnés selon la figure 2.

En tenant compte des longueurs des arêtes, la 0-distance  $(n, T, G)$  correspond à la longueur du chemin le plus court entre  $n$  et  $T$  et est égale à 11. La 1-distance  $(n, T, G)$  est égale à la longueur de l'arbre représenté en pointillé sur la figure 3 moins la longueur du seul chemin dans  $T$  qui peut être supprimé dans l'arbre couvrant final de  $n \cup X_g$ , soit 10.

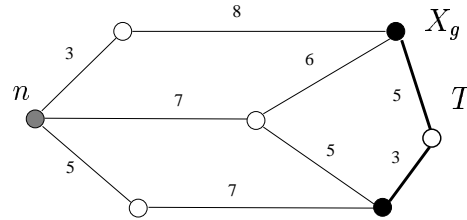


FIG. 2 – *Exemple : un noeud  $n$  et un arbre  $T$  couvrant  $X_g$*

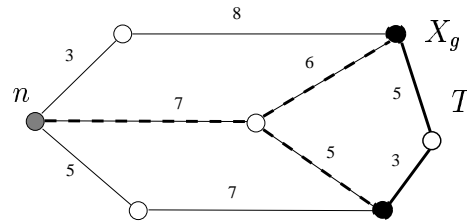


FIG. 3 – *1-distance entre le noeud et l'arbre*

Pour les  $k$ -distances, toujours pour cet exemple,  $k$  est limité à 1 (puisque l'arbre  $T$  est un chemin,  $X_g$  n'ayant que deux noeuds), les 2-distance, 3-distance, etc, n'ont pas de sens dans ce cas. La  $S$ -distance est ainsi de 10 .

**Lemma 3.4** *Si le groupe  $X_g$  couvert par l'arbre  $T$  contient  $(k + 1)$  noeuds, alors l'arbre d'association qui réalise la connexion optimale en terme de  $S$ -distance entre le noeud  $n$  et l'arbre  $T$  ne contient que  $k$   $S$ -noeuds au maximum.*

**Preuve.** Selon la définition de la  $S$ -distance, la connexion optimale de  $n$  à  $T$  est une révision de l'arbre couvrant partiel  $T$ . Cette nouvelle connexion, à l'aide de la révision des arêtes existantes et d'une éventuelle suppression de certaines parmi elles, crée un arbre minimum de Steiner de  $(n \cup X_g)$  . L'arbre minimum de Steiner d'un groupe de noeuds possédant  $(k + 2)$  noeuds ne peut avoir que  $k$   $S$ -noeuds (cf. [2]). ■

Remarque : de cette façon, le calcul de la  $S$ -distance et la recherche de l'arbre  $C_S^*(n, T, G)$  qui lui correspond sont limités. Pour énumérer les connexions possibles, il suffit de parcourir l'ensemble des arbres couvrants partiels ayant  $f = (k + 2)$  feuilles au maximum (y compris le noeud  $n$ ) et  $(f - 2)$   $S$ -noeuds au maximum si  $|X_g| = k + 1$ .

## 4 Construction de l'arbre minimum de Steiner exact

### 4.1 Algorithme principal

L'algorithme de Prim élargi par la notion de S-distance nous permet de construire l'arbre minimal de Steiner. Soit  $M$  le groupe *multicast* et  $T = (X_T, U_T)$  l'arbre en construction. Initialiser l'arbre à l'aide d'un noeud  $x_1$  arbitraire du groupe *multicast*.

Initialisation:  $X_T := \{x_1\}$ ;  $U_T := \{\phi\}$ ;  $M := M \setminus \{x_1\}$ ;

Jusqu'à ce que l'ensemble  $M$  soit vide faire :

Pour tout noeud  $x$  de  $M$  :

    Calculer la S-distance( $x, T, G$ ) ;

Pour le noeud  $y \in M$  le plus proche de  $T$  au sens de la S-distance :

    Ajouter à  $T$  l'arbre de Steiner  $C_S^*(y, T, G)$  puis éliminer les arêtes inutiles en construisant l'arbre  $\tau^*(T, C_S^*)$  ;

$M := M \setminus \{y\}$  ;

fait ;

Remarque : au cours de l'évolution de l'arbre couvrant (par augmentation du nombre de noeuds), le calcul des S-distances devient de plus en plus coûteux. La première itération nécessite le calcul des 0-distances, la deuxième nécessite le calcul des 0-distances et des 1-distances, etc.

Il ne s'agit pas ici de trouver l'algorithme le plus efficace (qui sera toujours *NP*) pour construire l'arbre minimum de Steiner exact mais de trouver un algorithme qui permette de construire une approximation de bonne qualité.

### 4.2 Algorithme de construction de l'arbre optimal d'association

$\tau^*(T, C)$

Au cours de la construction de l'arbre minimal couvrant par ajouts succesifs de noeuds à l'aide de l'algorithme précédent, on rencontre le problème de la construction de l'arbre optimal d'association  $\tau^*(T, C)$  et donc de la suppression des arêtes de  $T$  (et non de  $C$ ) incluses dans des boucles (cf. la définition 3.1). Pour déterminer l'ensemble des arêtes à enlever qui assure le gain maximal, nous proposons la solution suivante.

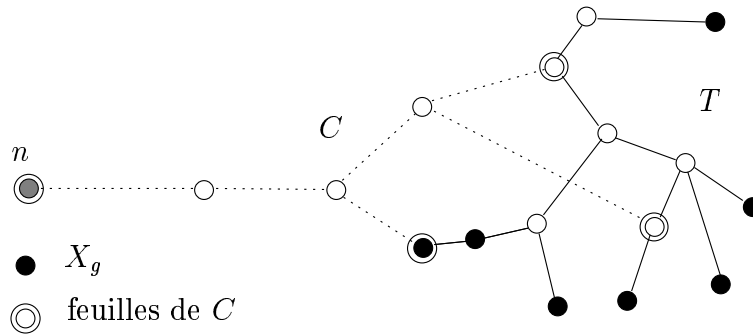


FIG. 4 – Arbre d'association optimal

Supposons que les arêtes de  $G$  soient valuées par des valeurs positives. Soit  $n$  le noeud à connecter à l'arbre  $T$  à l'aide de l'arbre  $C$  possédant  $k$  S-noeuds (il suffit d'analyser la situation où  $k > 0$ , puisque la connection par un plus court chemin ne crée pas de boucle et donc ne nécessite pas de suppression d'arête). Étant donné que  $C$  est un "arbre optimal d'association",  $C$  est tel que  $n$  est une feuille de  $C$  et les autres feuilles de  $C$  sont dans  $T$ . De plus,  $C$  et  $T$  n'ont pas d'arêtes communes. Soit  $X_g$  le sous ensemble du groupe *multicast*  $M$  couvert par  $T$  (cf. figure 4).

Afin d'obtenir l'arbre d'association optimal de  $T$  à  $n$  via  $C$ , il faut trouver un arbre couvrant l'ensemble  $(n \cup X_g)$  dans  $(C \cup T)$ . Pour cela, il faut éliminer certaines arêtes de  $T$ , en conservant l'accès à tous les noeuds de  $(n \cup X_g)$ . L'arbre d'association devient "optimal" si l'on élimine l'ensemble d'arêtes de longueur maximale de chaque boucle sans casser la connectivité entre les membres de  $(n \cup X_g)$ .

Dans ce but, énumérons les théorèmes suivants.

**Théorème 4.1** Notons  $T_c$  le sous-arbre de  $T$  limité par les feuilles de  $C$  (c-à-d : les feuilles de  $T_c$  sont aussi des feuilles de  $C$ ) tel qu'il ne possède pas de feuille hors l'arbre  $C$ . Les arêtes de  $T$  pouvant être supprimées pour obtenir l'arbre optimal d'association de  $n$  à  $T$  via  $C$  appartiennent à  $T_c$ .

**Preuve.** Si l'on enlève  $T_c$  de  $T$ , on obtient une forêt dont les arbres relient certains noeuds du groupe  $X_g$  à  $T_c$  (les feuilles de  $T$  appartiennent à  $X_g$ ). Si l'une des arêtes de ces arbres est enlevée, même si l'on rajoute  $T_c$  entièrement de nouveau à la structure, on perd

la connectivité d'au moins un noeud de  $X_g$  aux autres noeuds de  $(n \cup X_g)$ . La structure ainsi obtenue ne peut pas être un arbre. ■

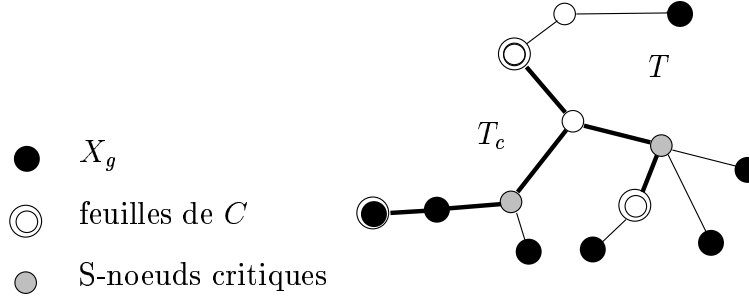


FIG. 5 – Sous-arbre critique pour éliminer les boucles

**Définition 4.2** Appelons *noeud de contact* du graphe  $G$  par rapport à un sous-graphe  $G_c$  les noeuds qui possèdent au moins une arête adjacente dans  $G$  n'appartenant pas à  $G_c$ .

**Définition 4.3** Appelons *S-noeud critique* de l'arbre  $T$  par rapport au sous arbre  $T_c$  les S-noeuds qui possèdent au moins une arête adjacente dans  $T$  n'appartenant pas à  $T_c$ . (Ces S-noeuds sont alors des noeuds de contact de  $T$  par rapport à  $T_c$ .)

**Théorème 4.4** La suppression des arêtes de  $T_c$  pour obtenir l'arbre d'association de  $n$  à  $T$  via  $C$  ne peut entraîner l'élimination des accès ni aux noeuds internes de  $T_c$  appartenant à  $X_g$  ni aux S-noeuds critiques de  $T$  par rapport à  $T_c$ .

**Preuve.** Trivialement, si l'on supprime tout accès vers un des noeuds de  $X_g$ , la structure obtenue ne peut pas être un arbre couvrant de  $(n \cup X_g)$ . De la même façon, si l'on supprime toutes les arêtes adjacentes dans  $T_c$  d'un S-noeud critique, cette suppression entraîne l'élimination de l'accessibilité d'au moins un noeud membre de  $X_g$ . La structure ainsi obtenue ne peut pas être un arbre. ■

**Définition 4.5** Soit  $T = (X, U)$  un arbre et  $P \subseteq X$  un sous-ensemble de noeuds internes de  $T$  (il n'y a pas de feuille dans  $P$ ). Appelons *forêt joignante partielle* une forêt  $F(T, P)$  qui relie chacun des noeuds de  $P$  à une feuille de  $T$ . Appelons *forêt joignante partielle minimale* du groupe la forêt  $F^*(T, P)$  de longueur minimale parmi les forêts joignantes partielles.

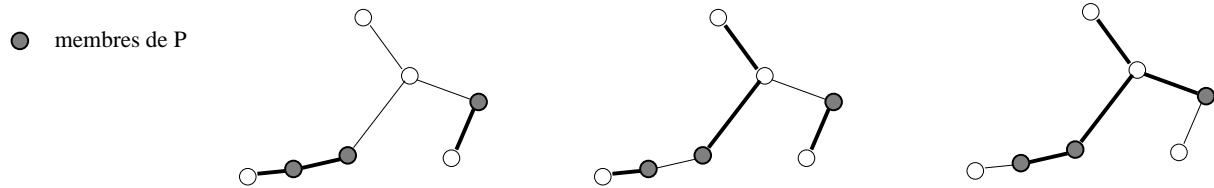


FIG. 6 – Forêts joignantes partielles de  $P$

**Théorème 4.6** Soit  $P_c$  l'ensemble des noeuds de  $T_c$  qui regroupe les noeuds internes soit appartenant à  $X_g$ , soit à l'ensemble des  $S$ -noeuds critiques de  $T$  par rapport à  $T_c$ . Une forêt joignante partielle minimale  $F^*(T_c, P_c)$  étant choisie, une réalisation de l'arbre optimal d'association de  $n$  à  $T$  via  $C$  s'obtient en supprimant de  $T_c$  toutes les arêtes qui n'appartiennent pas à  $F^*(T_c, P_c)$ .

**Preuve.** Pour pouvoir préserver l'accessibilité des noeuds dans  $(n \cup X_g)$ , il faut assurer l'accessibilité des noeuds de  $P_c$  à partir des feuilles de  $T_c$  (cf. théorème 4.4). Selon sa définition, c'est une forêt joignante partielle qui est capable d'assurer l'accessibilité des noeuds de  $P_c$  dans  $T_c$ . D'après la définition de l'arbre optimal d'association, la somme des longueurs des arbres composant cette forêt doit être minimale ; ce qui correspond à une forêt joignante partielle minimale. ■

**Théorème 4.7** La construction d'une forêt joignante partielle minimale correspond à la construction d'un arbre de Steiner minimal.

**Preuve.** Regroupons les feuilles de  $T_c$  en un seul noeud  $x_f$  composé. De cette façon, on obtient un graphe  $G_c$  qui contient le groupe  $P_c$  de noeuds (cf. figure 7). Les noeuds de  $P_c$  doivent être reliés à une des feuilles de  $T_c$ , donc au noeud  $x_f$  dans  $G_c$  à l'aide des arbres. Si l'on construit l'arbre de Steiner minimum de  $(x_f \cup P_c)$  dans  $G_c$  puis si l'on redécompose le noeud  $x_f$ , on obtient la forêt joignante partielle minimale. ■

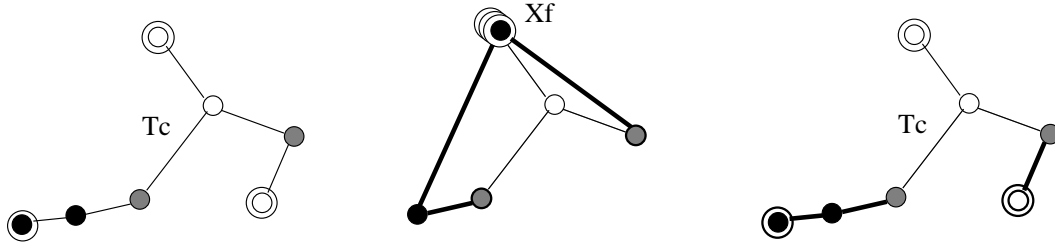


FIG. 7 – Construction de la forêt joignante partielle minimale

Les théorèmes ci-dessus indiquent comment les boucles peuvent être supprimées d'un arbre s'il est connecté par un autre arbre à un noeud. Le calcul de l'ensemble des arêtes à supprimer de l'arbre  $T = (X_T, U_T, V_T)$  qui couvre le sous-ensemble  $X_g$  du groupe *multicast* et qui est connecté par l'arbre  $C = (X_C, U_C, V_C)$  se décompose de la façon suivante :

Calculer les noeuds de connexion (les feuilles de  $C$ ) :  $X_{co} = X_T \cup X_C$  ;

Déterminer le sous arbre  $T_c = (X_c, U_c, V_c)$  limité par  $X_{co}$  ;

Déterminer l'ensemble  $S$  des S-noeuds critiques de  $T$  par rapport à  $T_c$  ;

Construire la forêt joignante partielle minimum de  $S \cup X_g$  dans  $T_c$  :

Créer le graphe  $G_c$  équivalent à  $T_c$  dont les noeuds de  $X_{co}$  sont regroupés en une feuille  $x_c$  ;

Construire l'arbre minimum de Steiner  $T_s = (X_s, U_s, V_s)$  de  $(S \cup X_g)$  dans  $G_c$  ;

Les arêtes à éliminer sont celles de  $U_c \setminus U_s$  ;

Remarques : en général, la construction de l'arbre de Steiner minimal demande un temps excessif en fonction de la taille du problème. Dans ce cas spécial, ce temps dépend fortement du nombre  $(k + 1)$  des feuilles de  $T_c$ . On peut constater que le nombre des boucles dans  $G_c$  (et de cette façon la complexité de l'algorithme dépendra de cette valeur) est égale à  $\binom{k+1}{2}$ .

Pour des valeurs faibles de  $k$ , le calcul est exécutable dans un temps raisonnable, étant donné qu'une bonne partie du graphe  $G_c$  est un arbre (correspondant à  $T_c$ ).

## 5 Heuristique pour approcher l'arbre minimum de Steiner exact

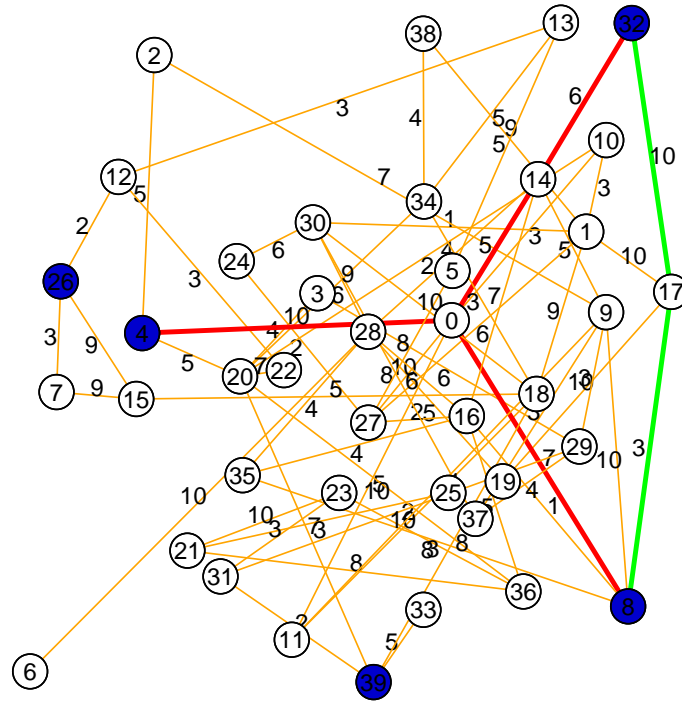
### 5.1 Limitations proposées

L'idée consiste à choisir une heuristique en temps polynomial. Pour cela, nous proposons deux limitations. Une réduction importante de temps de calcul peut être attendue, si l'on se limite à calculer toujours au plus la  $k$ -distance au lieu de la  $S$ -distance, où  $k$  est fixé et devient un paramètre réalisant le compromis entre la rapidité de calcul et la précision. Pour  $k = 0$ , l'heuristique correspond à celle de Takahashi et de Matsuyama où les noeuds sont associés à l'arbre par un plus court chemin. Pour  $k = 1$ , on calcule les 0-distances et les 1-distances des noeuds de l'arbre sous construction. Si la 1-distance est plus favorable, la connexion introduit une et une seule boucle. Dans ce cas-là, l'algorithme d'élimination de boucles est très simple. Le sous-arbre  $T_c$ , sur lequel les suppressions sont possibles, est un chemin ( $C$  n'a que deux sommets de contact avec  $T$ ) et il suffit d'éliminer le plus grand tronçon possible sur le chemin  $T_c$ .

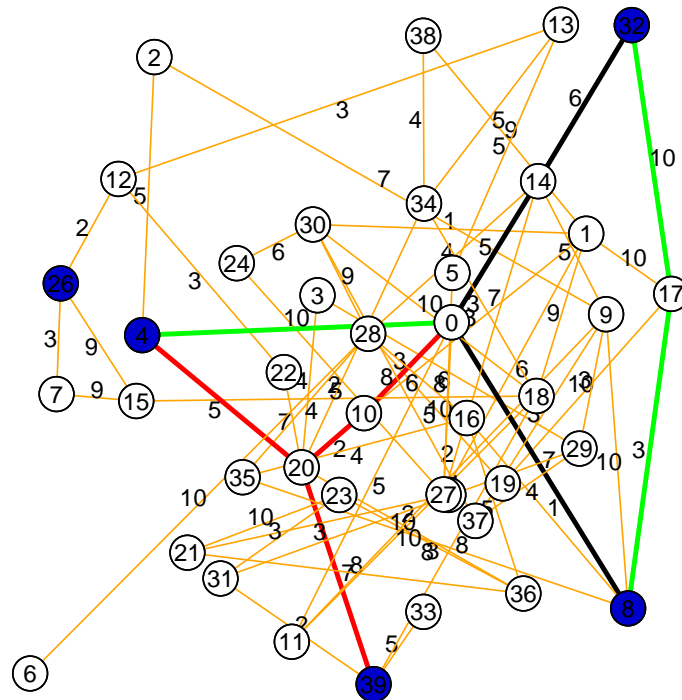
Une deuxième limitation peut concerner l'ensemble des noeuds de  $T$  pouvant être pris en compte pour calculer les  $k$ -distances. Intuitivement, on suppose que dans la réalisation d'un arbre optimal d'association à  $k$  S-noeuds, les noeuds de contact entre  $C$  et  $T$  ne sont pas "très loin" de  $n$ . Autrement dit : pour tenter de réaliser l'arbre optimal d'association à  $k$  S-noeuds entre  $n$  et  $T$ , on se limite à l'examen des  $m$  premiers noeuds de  $T$  qui sont les plus proches de  $n$ . Naturellement, pour  $k$  donné, il faut choisir  $m > k$ . Le tri des noeuds de  $T$  par rapport à un noeud externe  $n$  ne cause pas de problème ; étant donné que le calcul de la 0-distance nécessite la connaissance des distances habituelles entre  $n$  et les noeuds de  $T$ . En résumé : pour calculer les  $k$ -distances avec  $k$  supérieur à 0, on n'utilisera que les  $m$  premiers noeuds de la liste triée.

### 5.2 Exemple

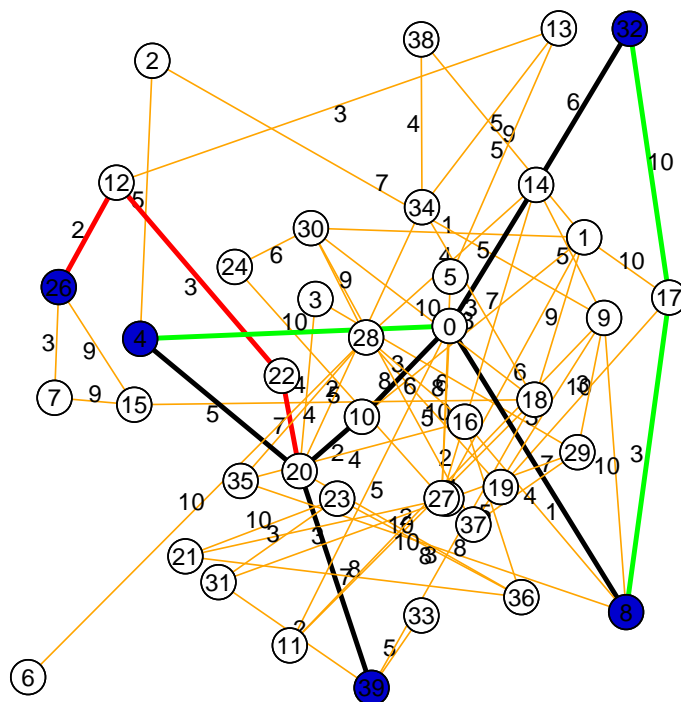
Dans la suite, nous illustrons les résultats d'une telle heuristique qui utilise un arbre minimal de Steiner à au plus un S-noeud pour connecter les noeuds à l'arbre sous construction. Dans cet exemple, vue la simplicité des arbres d'association, l'ensemble des noeuds de  $T$  pouvant être examinés et connectés n'est pas limité.



Création de l'arbre approché: phase 2



Création de l'arbre approché: phase 3



Création de l'arbre approché : phase 4

FIG. 8 – Exemple d'évolution de l'algorithme

Pour tester l'algorithme, nous avons généré des graphes connectés à 40 noeuds et à 70 arêtes à l'aide du générateur de graphes aléatoires de LEDA [5]. Les arêtes et les longueurs des arêtes ont été choisies aléatoirement (les longueurs étant choisies entre 1 et 10). Les membres du groupe *multicast* ont été tirés au sort sur l'ensemble des noeuds du graphe. La figure 8 nous montre l'évolution du résultat de l'heuristique sur un tel graphe. Les éventuelles arêtes noires indiquent l'état actuel de l'arbre en construction. Au cours des pas présentés de l'évolution, l'algorithme propose la connexion d'un membre du groupe *multicast* à l'arbre en utilisant une connexion sous forme d'un arbre qui est présenté en rouge sur la figure. Les arêtes vertes sont des arêtes à supprimer afin de créer la connexion sans laisser de boucle.

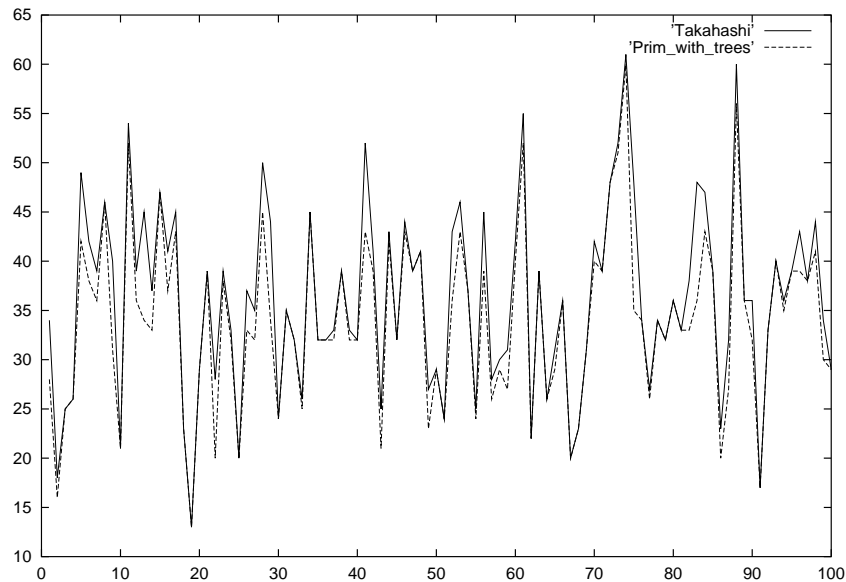


FIG. 9 – *La longueur des arbres construits*

Dans un jeu de simulations, nous avons lancé l'algorithme de Takahashi et Matsuyama puis l'heuristique que nous proposons ici sur 100 graphes différents générés comme ci-dessus. La longueur des arbres construits par les deux heuristiques sont présentées sur la figure 9. Nous avons calculé la moyenne des arbres par heuristique. L'évolution des valeurs moyennes est illustrée sur la figure 10. Dans certains cas, la nouvelle heuristique a comme résultat un arbre plus favorable. Sur l'ensemble des 100 graphes, le gain du nouvel algorithme par rapport à l'heuristique de Takahashi et Matsuyama est de 6 %.

Notons que le gain dépend fortement du graphe et du groupe *multicast*. Dans des cas "faciles", les deux algorithmes fournissent la même solution. L'avantage de la nouvelle heuristique se manifeste dans les cas où les algorithmes classiques ne peuvent pas donner une solution proche de l'optimum.

Il est important de noter que cette heuristique est particulièrement bien placée pour modifier l'arbre *multicast*, si l'on veut ajouter un (ou plusieurs) noeud(s) à un arbre déjà existant (cas de la mise à jour d'un arbre).

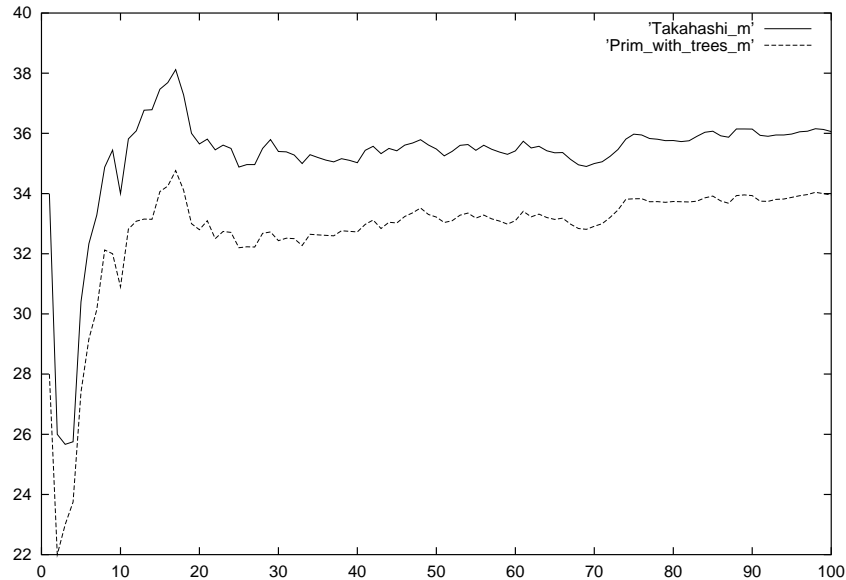


FIG. 10 – L'évolution de la moyenne des longueurs

## 6 Conclusion

L'heuristique de Takahashi et Matsuyama est facilement réalisable et donne des résultats relativement proche de l'optimum avec un temps faible de calcul. Toutefois, dans certains cas, cette heuristique, comme toutes les heuristiques qui utilisent les plus courts chemins pour construire l'arbre approché de l'arbre minimum de Steiner, n'est pas capable de trouver des sous-arbres couvrants partiels nécessaires pour résoudre des sous-problèmes de construction. A l'aide de la généralisation de la distance d'un noeud à un arbre sur un graphe, on peut reformuler l'algorithme simple de Takahashi et Matsuyama. Les distances basées sur les arbres minimaux de Steiner et les connexions qui réalisent la nouvelle distance minimale nous permettent de construire des arbres de Steiner plus favorables. Pour éviter l'explosion combinatoire due aux calculs des distances de plus en plus compliqués, on est obligé d'introduire des limitations. Si seule les plus simples distances sont calculées, on obtient une (des) heuristique(s) en temps polynomial, qui construisent, dans des cas difficiles, des arbres plus favorables. Dans les diffusions *multicast* où les applications peuvent être très sensibles à la longueur de l'arbre utilisé, l'amélioration des arbres *multicast* est important.

## Références

- [1] P. Winter, "*Steiner problem in networks: A survey*", Networks, vol. 17. pp. 129-167, 1987.
- [2] F.K. Hwang, D. S. Richards, "*Steiner Tree Problems*", Networks, vol. 22. pp. 55-89, 1992.
- [3] H. Takahashi, A. Matsuyama, "*An approximate solution for the Steiner problem in graphs*", Math. Japonica, vol. 24., 1981.
- [4] M. Molnár, "*Construction d'arbres de diffusion multicast basée sur une modification de l'heuristique de Kruskal*", rapport de recherche, IRISA. no.1214., 1998.
- [5] *LEDA, Library of Efficient Data types and Algorithms*,  
<http://www.mpi-sb.mpg.de/LEDA>



---

Unit ´e de recherche INRIA Lorraine, Technop ˆole de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unit ´e de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unit ´e de recherche INRIA Rh ˆone-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN  
Unit ´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unit ´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

´Editeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399