

# Construction d'arbres de diffusion multicast basée sur une modification de l'heuristique de Kruskal

Miklos Molnar

► **To cite this version:**

Miklos Molnar. Construction d'arbres de diffusion multicast basée sur une modification de l'heuristique de Kruskal. [Rapport de recherche] RR-3587, INRIA. 1998. <inria-00073093>

**HAL Id: inria-00073093**

**<https://hal.inria.fr/inria-00073093>**

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.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Construction d'arbres de diffusion multicast  
basée sur une modification de l'heuristique  
de Kruskal*

Miklós Molnár

**N°3587**

Décembre 1998

THEME 1B

A large blue rectangle occupies the lower half of the page. Overlaid on it is the text 'Rapport de recherche' in a white serif font. The 'R' is significantly larger and more stylized than the other letters. A horizontal white line is positioned below the text.

*Rapport  
de recherche*





# Construction d'arbres de diffusion multicast basée sur une modification de l'heuristique de Kruskal

Miklós Molnár\*

Programme 1B - Réseaux, systèmes, évaluation de performance

Projet Model

Rapport de recherche n°3587

Décembre 1998

22 pages

**Résumé :** La construction d'arbres couvrants partiels de poids minimum est un problème NP-complet pour lequel plusieurs heuristiques approximatives ont déjà été formulées. La plupart d'entre elles (telle l'heuristique de Kruskal) sont basées sur la recherche des plus courts chemins pour relier différentes composantes de l'arbre. Dans ce rapport, nous présentons un algorithme de construction d'une approximation de l'arbre minimum de Steiner (qui est l'arbre optimal d'une diffusion multicast) basé sur des structures plus avantageuses que celles des plus courts chemins. L'algorithme réalise les connexions sous forme des arbres minimum de Steiner limités en nombre de noeuds.

**Mots-clé :** Réseaux de communication, diffusion multicast, arbre de Steiner, algorithmes heuristiques.

*(Abstract: pto)*

\* Email : Miklos.Molnar@irisa.fr

# **Construction of Multicast Diffusion Trees Based on a Modified Kruskals Heuristic**

**Abstract:** The construction of partial minimum spanning trees being NP-complete, several heuristic algorithm have been already formulated. Many of these heuristics (like the heuristic of Kruskal) use the shortest paths to connect the components of the tree. In this work, we give an approximate construction algorithm for the minimum Steiner tree (which is the optimal tree of a diffusion multicast). This construction is based on structures more advantageous than shortest paths. The algorithm makes connections like simple Steiner trees if necessary.

**Key-words:** Communication networks, multicast diffusion, graph theory, Steiner trees, heuristical algorithms.

## **1 Introduction**

Dans les réseaux de communication actuels (nous parlons ici de réseaux à haut débit orientés connexions, eg. ATM), la diffusion d'un même message vers plusieurs destinataires peut se faire par l'intermédiaire d'un arbre. Si nécessaire, le message qui transite par les arêtes en un seul exemplaire est copié au passage des noeuds. Cette diffusion dite multicast est une source d'économie mais elle pose deux problèmes majeurs : d'une part la construction de l'arbre optimal de diffusion est un problème NP-complet et d'autre part, si la diffusion est réalisée sous forme d'un circuit virtuel (VC) et si le groupe contient plusieurs sources, la séparation des messages provenant de différentes sources du groupe multicast est difficile. Ce dernier problème provient du fait que l'en-tête des cellules ATM ne contient pas d'informations sur la source. Dans ce rapport, nous nous intéressons au premier problème lié à la construction de l'arbre de diffusion multicast.

On constate aisément que la diffusion optimale des messages entre les membres d'un même groupe multicast utilise un arbre couvrant partiel minimum du groupe. Quelque soit la source ou la racine, l'arbre de diffusion doit être minimal et couvrir les membres du groupe. La détermination d'un arbre couvrant partiel minimum est connue sous le terme du problème (NP-complet) de Steiner dans les réseaux et l'arbre associé est généralement appelé arbre minimum de Steiner[1,8]. Il n'existe pas, à notre connaissance, d'algorithme donnant une construction exacte d'arbre minimum de Steiner avec un temps polynomial [2]. Par contre, plusieurs heuristiques permettant d'approcher la solution optimale ont été proposées. Une bonne partie de ces heuristiques reprend soit les idées de base utilisées pour la construction d'arbres couvrant minimum (dont la complexité est facilement maîtrisable) soit une méthode qui utilise les plus courts chemins entre les noeuds pour construire un arbre couvrant partiel relativement avantageux. Dans la section 2, nous présentons quelques heuristiques connues de construction de l'arbre de Steiner et des exemples qui illustrent dans certains cas leur inefficacité. La section 3 présente une étude des arbres de Steiner simples et les définitions nécessaires à la démarche proposée. Dans la section 4, nous proposons une heuristique basée sur celle de Kruskal. Cette construction permet non seulement de trouver les plus courts chemins entre les composantes potentielles de l'arbre couvrant partiel mais aussi de chercher les connexions sous forme des arbres minimum de Steiner simples. Nous analysons également la complexité de cette solution approximative. Dans ce rapport nous ne considérons que des graphes non-orientés (le plus souvent, les réseaux de communications peuvent être modélisé par de tels graphes), mais l'heuristique proposée s'adapte facilement en cas d'asymétries dans les liaisons. Afin de faciliter la compréhension de la solution proposée, le lecteur pourra se référer à l'annexe qui contient quelques détails relatifs à la construction d'arbres de Steiner simples.

## 2 Quelques heuristiques de construction d'arbres multicast

### 2.1 Formulation du problème et concepts de base des heuristiques

Soit  $G = (X, U, V)$  un graphe connexe non orienté où  $X, U$  et  $V$  désignent respectivement les ensembles des noeuds, des arêtes et des longueurs associées à ces arêtes. On notera  $X_g = \{i \in X\}$  un sous-ensemble de noeuds du graphe  $G$  correspondant à un groupe multicast de cardinalité  $g$ . Dans la suite, on suppose que la longueur (le poids) d'une arête n'est pas négative. On rappelle qu'un *arbre couvrant partiel* de  $X_g$  dans  $G$  est un arbre  $T = (X_T, U_T)$  tel que :  $X_g \subseteq X_T$ . Un *arbre couvrant partiel minimum* (ou arbre couvrant partiel de poids minimum, ou arbre minimum de Steiner) du groupe  $X_g$  dans  $G$  est un arbre couvrant partiel  $T_m = (X_{T_m}, U_{T_m})$  tel que la somme des longueurs des arêtes de  $U_{T_m}$  est minimale parmi les arbres couvrants partiels de ce groupe  $X_g$ . Autrement dit : la longueur d'un arbre est la somme des longueurs des arêtes qui le composent :

$$C(T) = \sum_{(i, j) \in U_T} v_{ij}$$

Si  $\tau$  désigne l'ensemble des arbres couvrants partiels d'un groupe  $X_g$ , alors l'arbre minimum de Steiner de ce groupe est donné par :

$$C(T_m) = \min_{T \in \tau} C(T)$$

*Remarques :*

- En général dans un graphe connexe, il existe plusieurs arbres couvrants partiels parmi lesquels se trouve au moins un arbre couvrant partiel minimum.
- Dans un graphe connexe sans cycle (c-à-d dans un arbre), il n'y a, pour un groupe donné, qu'un seul arbre couvrant partiel minimum.

Dans le cas général, la recherche d'un arbre de Steiner est un problème NP-complet [1, 2, 8].

La diminution de cette complexité liée à la construction de l'arbre minimal couvrant un groupe de noeuds dans un graphe connexe non-orienté a fait l'objet de nombreux travaux.

Takahashi et Matsuyama ont présenté dans [3] une des plus simples construction. Cette construction, voisine de celle proposée par Prim pour construire un arbre couvrant minimum (cf. par exemple [9]), se limite aux membres du groupe multicast. Le squelette de l'algorithme est simple :

*Initialisation : Initialiser l'arbre en choisissant un noeud arbitraire du groupe. Calculer les distances entre ce noeud et les autres membres du groupe.*

*Jusqu'au dernier noeud du groupe : ajouter à l'arbre courant le noeud le plus proche (en utilisant le plus court chemin entre le noeud et l'arbre). Calculer les distances des noeuds restants avec l'arbre ainsi obtenu.*

Une autre construction simple correspond à l'heuristique de Dijkstra tronquée (Pruned Dijkstra Heuristic [1, 2]). L'algorithme simplifié en est :

*Sélectionner la racine de l'arbre  $s$  arbitrairement du groupe multicast ; Déterminer les plus courts chemins de  $s$  aux autres membres du groupe (selon l'algorithme de Dijkstra) ;*

*Éliminer les boucles.*

Kruskal présente un algorithme qui considère simultanément tous les membres du groupe multicast [4]. Il propose d'initialiser une forêt (un ensemble d'arbres) avec les membres du groupe (chacun des noeuds initialise un arbre isolé). Son algorithme permet d'obtenir un seul arbre couvrant à l'aide des connexions successives entre les arbres isolés. Dans son algorithme, les connexions sont réalisées à l'aide des plus courts chemins :

*Initialisation : On considère chaque membre du groupe, comme un arbre isolé initial;*

*Jusqu'à l'obtention d'un seul arbre : Calculer les distances entre les arbres ; Relier les deux arbres les plus proches en utilisant le plus court chemin entre eux.*

Notons ici que cet algorithme peut être parallélisé, puisque la recherche des connexions possibles peut se faire à l'aide de plusieurs calculateurs situés dans les différents arbres isolés. Une version distribuée est présentée dans [5].

Kou, Markowsky et Berman [6] transposent la recherche d'un arbre couvrant partiel dans la fermeture métrique du graphe du problème :

*Créer la fermeture métrique pour le groupe  $X_g$  (c'est un graphe complet ayant  $X_g$  comme ensemble de sommets et chacune de ses arêtes est évaluée avec la longueur du plus court chemin entre les deux noeuds concernés) ;*

*Construire un arbre  $T_1$  couvrant minimum (MST) dans cette fermeture ;*

*Construire un sous-graphe  $G_g$  dans  $G$  en remplaçant les arêtes de  $T_1$  par les chemins correspondants*

*Construire un arbre  $T_2$  couvrant minimum (MST) dans ce graphe  $G_g$*

*Éliminer les branches inutiles*

Cette dernière idée a été reprise par Kampella et al. [7]. L'algorithme proposé permet d'optimiser les routages lorsqu'on impose des contraintes sur le délai de communication.



En utilisant certaines heuristiques, il est prouvé qu'on obtient un arbre qui est au plus deux fois plus cher que l'arbre minimum de Steiner du groupe [1, 8]. Cependant, Zelikovsky propose un algorithme qui permet de construire un arbre dont le quotient *coût du résultat / coût de l'arbre minimum de Steiner* est inférieur à 11/6, [10]. Reprenant l'idée de travailler sur la fermeture métrique du problème, Zelikovsky utilise (à la place des plus courts chemins) des arbres pour couvrir des triplets de noeuds. Pour décrire cet algorithme, utilisons la notation suivante :  $\overline{G}_{X_g}$  désigne la fermeture métrique du groupe  $X_g$  dans le graphe  $G$ , l'opération  $F[z]$  représente la contraction d'un triplet  $z$  (i.e. l'annulation des distances entre les noeuds qui le composent) dans le graphe  $F$  et  $mst(F)$  est une fonction qui retourne la longueur de l'arbre couvrant minimum du graphe  $F$ . Les grandes lignes de l'algorithme de Zelikovsky sont les suivantes :

*Initialisation* :  $F = \overline{G}_{X_g}$ ;  $W = \{\}$ ;  $Triplets = \{z = (a, b, c), a \in X_g, b \in X_g, c \in X_g\}$ ;

*Pour tout  $z$  dans Triplets faire* :

*Trouver le sommet  $v(z)$  dans  $X$  tel que* :  $d(z) = \sum_{s \in z} d(v, s)$  *est minimale* ;

*Tant que la stabilité n'est pas atteinte faire* :

*Sélectionner  $z$  dans Triplets qui maximise la fonction d'objectif* :

$$f(z) = mst(F) - mst(F[z]) - d(z) ;$$

*Si  $f(z) \leq 0$ , alors sortir par stabilité*

*sinon  $F = F[z]$ ;  $W = W + [v(z)]$ ;*

*fait*;

*Trouver un arbre de Steiner pour  $W \cup X_g$  dans le graphe  $G$  (en utilisant l'algorithme de Kou et al. par exemple).*

Notons que l'ensemble  $W$  qui apparaît dans l'algorithme permet de fixer un ensemble de  $S$ -noeuds (cf. section 2.2) facilitant la construction de l'arbre de Steiner final.

Ramanathan propose dans [2] des généralisations de plusieurs heuristiques ainsi que l'extension du problème aux réseaux de communication asymétriques. Dans sa solution, il utilise l'algorithme de Prim combiné avec une file à priorité pour stocker les noeuds de l'arbre déjà construit. Dans cette file, la racine de l'arbre est prioritaire, elle est suivie par les feuilles et puis les noeuds intermédiaires. Quand on calcule la distance d'un membre non encore rajouté à l'arbre, on ne tient compte que des  $k$  premiers noeuds de la file.  $k$  est un paramètre qui permet d'obtenir un algorithme paramétrable pour choisir un compromis entre le temps de calcul et la performance (ou coût) de l'arbre. Pour  $k=1$ , on retrouve la solution de Dijkstra tronquée et pour

$k$  maximal, on obtient la performance de l'algorithme de Kruskal. Notons la file à priorité par  $Q$ . L'algorithme s'écrit ainsi :

*Initialisation avec un noeud  $s$  arbitraire :  $Q=\{s\}$ ;  $X_t=\{s\}$ ;  $U_t=\{\}$ ;  $Y = X_g$  ;*

*Tant que  $Y$  n'est pas vide faire :*

*Sélection :  $B = k$  premiers noeuds de  $Q$  ;  $P=$  un chemin arbitraire de  $B$  à  $Y$  ;  $c_p=c(P)$ ;*

*Pour tout  $b$  dans  $B$  faire :*

*Pour tout  $m$  dans  $Y$  faire :*

*$P' =$  le plus court chemin de  $b$  à  $m$  ;*

*Si  $c(P') < c_p$  alors  $P=P'$  ;  $c_p=c(P')$  ;*

*Insérer les noeuds de  $P$  dans  $Q$  (en respectant les priorités) ;*

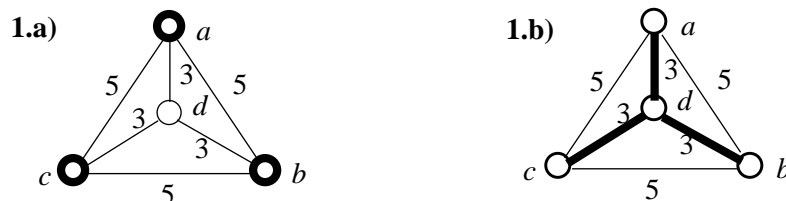
*$X_t = X_t \cup$  (noeuds de  $P$ ) ;  $U_t = U_t \cup$  (arcs de  $P$ ) ;  $Y = Y -$  (extrémités de  $P$ ) ;*

*fait.*

## 2.2 Problèmes posés par les heuristiques

La plupart des algorithmes rappelés dans la section 2.1 ont un temps d'exécution très raisonnable et le coût de l'arbre construit reste acceptable (proche de la valeur optimale). Lorsque les approximations fournies par les heuristiques connues ne sont pas acceptables (c'est le cas par exemple d'un coût à peu près égal au double de celui de l'arbre optimal en présence d'une application très gourmande en débit), il est intéressant de chercher des solutions plus proches de l'optimum. Intuitivement, les algorithmes qui donnent des solutions plus performantes, nécessitent un temps d'exécution plus important.

L'insuffisance des algorithmes cités vient du fait que la topologie utilisée pour relier les bouts de l'arbre est une structure linéaire contrairement à la structure cherchée (arbre couvrant optimal). A titre d'exemple, considérons un graphe non-orienté complet avec quatre noeuds dont les longueurs des arêtes sont indiquées sur la figure 1.a. On note  $X_g = [a, b, c]$  le groupe multicast. L'arbre minimum de Steiner de ce groupe est un arbre qui contient le noeud  $d$  (cf. figure 1.b).



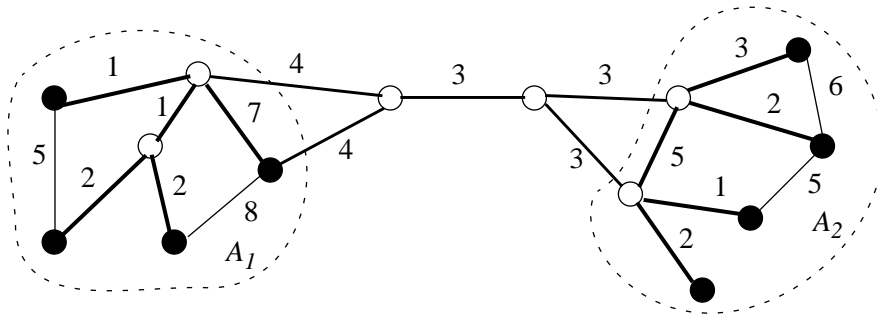
**Figure 1 :** Exemple d'un groupe et de son arbre minimum de Steiner qui ne peut pas être construit à l'aide des plus courts chemins

On notera que cet exemple ne peut pas être résolu par les algorithmes utilisant les plus courts chemins parmi les connexions. Sur l'exemple : les plus courts chemins sont les arêtes  $(a,b)$ ,  $(a,c)$  et  $(b,c)$ . Le noeud de branchement  $d$  nécessaire à la construction de l'arbre optimal est appelé *noeud de Steiner* ou *S-noeud* [9].

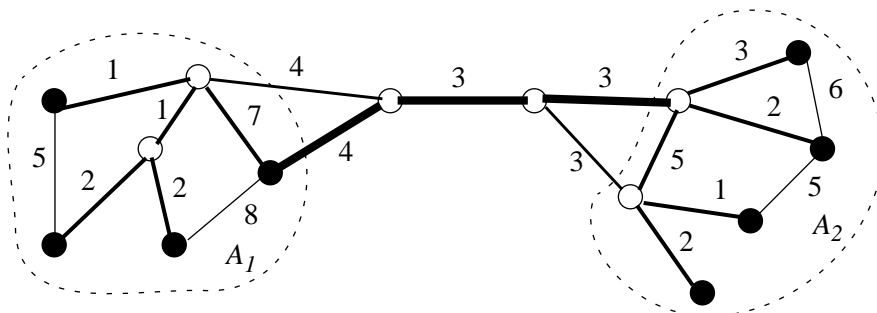
En théorie, il est toujours possible de trouver au moins une solution optimale. Il suffit par exemple de construire l'arbre minimum de Steiner en testant tous les arbres couvrants partiels possibles. Quelques méthodes exactes, dont la méthode de l'énumération des topologies, sont présentées dans [1]. Bien que le nombre des topologies abstraites possibles soit limité (on ne peut avoir que  $(g-2)$  S-noeuds au maximum dans l'arbre couvrant d'un groupe qui possède  $g$  noeuds [9]), le problème ainsi posé est NP-complet. De manière concrète pour la résolution d'une communication multicast dans un réseau de taille réelle, on ne peut envisager que des algorithmes fournissant des solutions approchées en temps polynomial. Dans les heuristiques présentées dans la section 2.1., la recherche des connexions au cours de la construction est limitée à l'ensemble des plus courts chemins. C'est aussi valable pour algorithmes utilisant la fermeture métrique du problème, puisque les arêtes de la fermeture métrique correspondent aux plus courts chemins du graphe d'origine du problème. Seul l'algorithme de Zelikovsky [10] utilise des arbres minimum de Steiner à la place des plus courts chemins pour couvrir certains triplets de noeuds du groupe multicast.

### 2.3 Principes d'une nouvelle heuristique

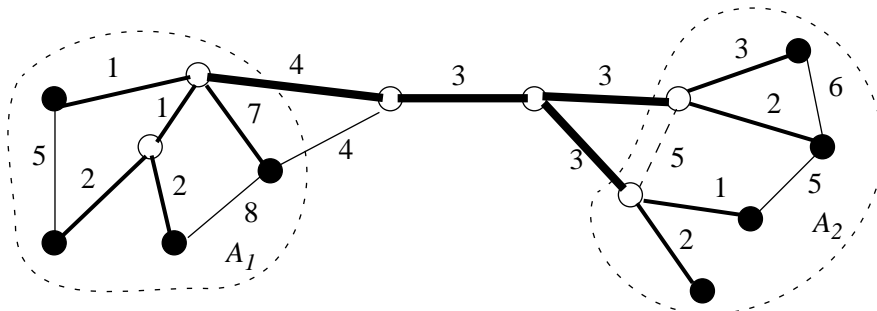
La connexion optimale entre deux parties de l'arbre en construction peut être différente d'un éventuel plus court chemin (par exemple : la connexion entre deux arbres dans la forêt de l'heuristique de Kruskal, ou la connexion entre le noeud sélectionné et l'arbre déjà construit dans l'algorithme de Kou et al.). Supposons que l'idée de base de l'heuristique de Kruskal soit retenue pour la construction de l'arbre multicast : on construit l'arbre couvrant partiel à partir des arbres isolés correspondant aux noeuds du groupe, à l'aide des réunifications successives des arbres. La figure 2 présente un cas où il n'existe que deux arbres ( $A_1$  et  $A_2$ ) dans la forêt et le pas suivant de l'algorithme doit les connecter pour obtenir l'arbre couvrant du groupe multicast donné par des noeuds noircis sur la figure. Les arêtes du graphe sont évaluées par leur longueur. Si l'on utilise l'algorithme de Kruskal, les deux arbres seront connectés à l'aide du plus court chemin entre eux (cf. figure 3). La figure 4 indique le cas où l'on utilise un arbre couvrant partiel minimum à un S-noeud pour relier les deux arbres et dans le cas de la figure 5 le nombre des S-noeuds dans l'arbre de connexion est égale à deux. Dans cet exemple, la connexion optimale est celle présentée sur la figure 5. Ici, l'arbre de connexion permet de remplacer deux arêtes coûteuses se trouvant dans les arbres couvrant  $A_1$  et  $A_2$  (arêtes en pointillé de longueur 7 et 5 sur la figure).



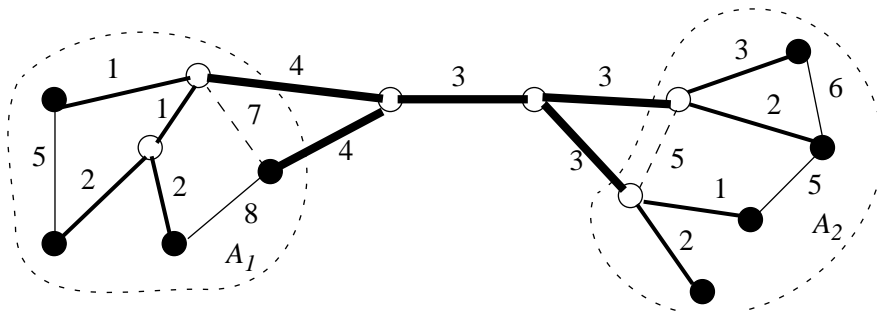
**Figure 2 :** Deux arbres de Steiner à connecter



**Figure 3 :** Connexion à l'aide du plus court chemin



**Figure 4 :** Connexion sous forme d'un arbre à un S-noeud



**Figure 5 :** Connexion sous forme d'un arbre à deux S-noeuds

En ne tenant compte que des connexions sous forme des plus courts chemins (le nombre de  $S$ -noeuds d'un chemin est nul), on obtient des solutions limitées en performance mais l'arbre est construit en temps polynomial (cf. les heuristiques dans [4, 6]). On peut supposer que si l'on se limite à l'utilisation des arbres minimum de Steiner à  $m$   $S$ -noeuds au maximum pour couvrir ou connecter un sous-ensemble de  $m+2$  noeuds au maximum, alors la recherche de l'arbre couvrant partiel se fait aussi avec un temps polynomial (cf. l'exemple de la section 4.). Dans le cas où la complexité et le nombre des structures des connexions ne sont limités que par la taille du problème, on peut construire l'arbre de Steiner exact ; la contre partie réside dans une explosion combinatoire du temps d'exécution. La valeur de  $m$  (où  $0 \leq m \leq g - 2$ ) peut être considérée comme un compromis entre les deux solutions extrêmes.

La proposition de Ramanathan [2] pour construire une famille paramétrable d'algorithmes nous montre que le choix des points à connecter influence également la complexité de l'algorithme de recherche. On peut limiter le nombre des noeuds qui sont susceptibles de participer aux liaisons optimales entre deux arbres (ces noeuds correspondent aux feuilles de l'arbre qui réalise la connexion). Cette limitation est similaire à la gestion des  $k$  premiers éléments de la file de priorité dans l'algorithme de Ramanathan et permet de diminuer la complexité de l'algorithme.

Dans la section 4, nous proposons un algorithme de construction d'arbres approchés d'arbres minimum de Steiner. Cette approche, qui est basée sur l'heuristique de Kruskal, permet de définir des liaisons sous la forme d'arbres. On se limitera :

- aux arbres (minimum de Steiner) ayant au maximum quatre feuilles pour réaliser les connexions ;
- à deux noeuds au maximum par arbre de la forêt pour examiner et établir une éventuelle connexion.

### 3 Etude des arbres de Steiner

Afin de justifier notre choix et de faciliter la compréhension de l'algorithme proposé, nous rappelons ici quelques notions qui concernent la topologie des connexions proposées et les propriétés des arbres de Steiner.

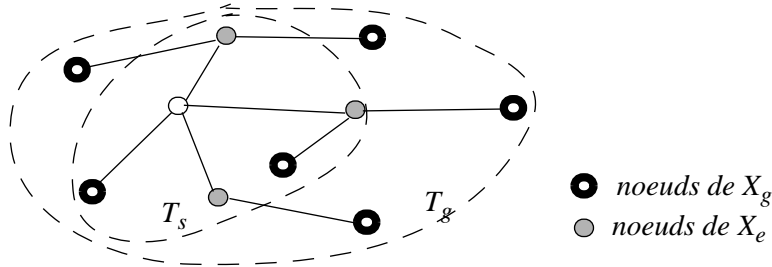
#### 3.1 Quelques propriétés des arbres minimum de Steiner

*Propriété 1* : Les feuilles d'un arbre minimum de Steiner  $T_g$  relatif à un groupe  $X_g$  de noeuds appartient à  $X_g$ .

En effet, supposons qu'il existe une feuille dans  $T_g$  n'appartenant pas à  $X_g$ . Si l'on l'enlève (avec la branche qui le relie dans  $T_g$  à un autre noeud de  $X_g$  ou à un noeud de branchements de l'arbre) l'arbre reste un arbre couvrant de  $X_g$ . Donc,  $T_g$  ne peut pas être un arbre couvrant minimum.

**Théorème 1 :** Soit  $T_s$  un sous-arbre de l'arbre minimum de Steiner  $T_g$  relatif à un groupe  $X_g$ . On désigne par  $X_f$  l'ensemble des feuilles de  $T_s$  et par  $X_e$  l'ensemble de noeuds de  $T_s$  possédant une arête adjacente qui n'a pas son autre extrémité dans  $T_s$ .  $T_s$  est un arbre minimum de Steiner relatif à  $X_f \cup X_e$ .

*Preuve :* Supposons que le sous-arbre  $T_s$  ne soit pas minimal pour l'ensemble  $X_f \cup X_e$ . Dans ce cas, il existe un arbre minimum de Steiner  $T_m$  pour le même groupe  $X_f \cup X_e$  tel qu'en remplaçant  $T_s$  par  $T_m$  dans  $T_g$ , on obtient un arbre couvrant de longueur inférieur à la longueur de  $T_s$ . Ceci contredit le fait que  $T_g$  est un arbre minimum de Steiner.



**Figure 6 :** Un sous-arbre d'un arbre minimum de Steiner

**Lemme 1 :** Un arbre privé d'un de ses sous-arbres est une forêt.

*Preuve :* Si le sous-arbre enlevé ne contient pas de coupe de l'arbre, alors le graphe obtenu est un arbre. En effet, dans le cas contraire le graphe restant est composé de plusieurs composantes connexes dont chacune est un arbre.

**Théorème 2 :** Soit  $T_g$  un arbre minimum de Steiner du groupe  $X_g$  dans le graphe  $G$ . Notons par  $T_{g_i}$  l'arbre minimum de Steiner couvrant le sous-groupe  $X_{g_i} \subseteq X_g$  dans  $T_g$  (et non dans  $G$ ) et par  $\bar{X}_{g_i}$  l'ensemble de noeuds de cet arbre  $T_{g_i}$ . Soit  $P = \{X_{g_i}\}$  une partition du groupe  $X_g$ , i.e., telle que

$$\bigcup_{X_{g_i} \in P} X_{g_i} = X_g$$

$$X_{g_i} \cap X_{g_j} = \emptyset \quad X_{g_i} \in P, X_{g_j} \in P, i \neq j$$

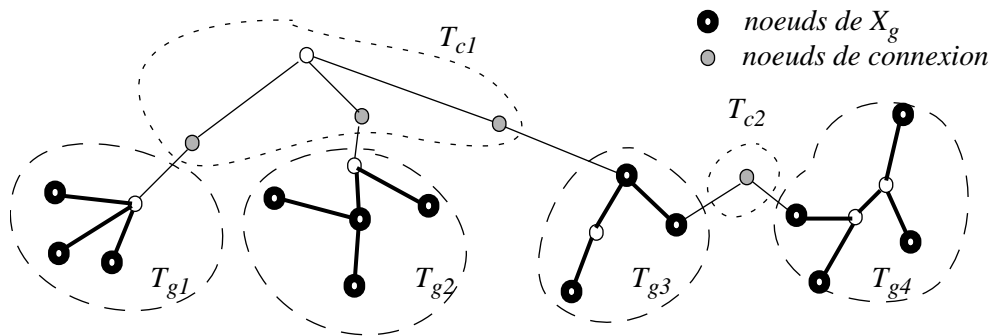
Deux arbres  $T_{g_i}$  et  $T_{g_j}$  minimum de Steiner relatifs aux deux groupes disjoints  $X_{g_i}$  et  $X_{g_j}$  de  $P$  n'ont ni arête ni noeud en commun.

La structure qui relie les différents arbres  $T_{g_i}$  dans  $T_g$  est une forêt dont chaque arbre est un arbre minimum de Steiner  $T_{c_j}$  relatif aux noeuds qui possèdent des arêtes adjacentes dans  $T_g$  telles que leur autre extrémité ne soit pas dans l'arbre  $T_{c_j}$  mais dans

$$\bigcup_{X_{g_i} \in P} \bar{X}_{g_i}.$$

Autrement dit : Les arbres couvrants induits par la partition  $P$  sont connectés dans  $T_g$  à l'aide d'arbres minimum de Steiner (cf. figure 7).

*Preuve :* Si l'on enlève les différents sous-arbres  $T_{g_i}$  de  $T_g$  on obtient une forêt (lemme 1). Chaque arbre de cette forêt est un sous-arbre de l'arbre minimum de Steiner  $T_g$ . D'après la définition de la partition  $P$ , aucun de ces arbres ne contient de noeud appartenant à  $X_g$ . En vertu du théorème 1, un tel arbre est alors un arbre minimum de Steiner de l'ensemble de noeuds qui possèdent des arêtes adjacentes dans  $T_g$  telles que leur autre extrémité n'est pas dans l'arbre en question.



**Figure 7 :** Connexion des membres d'une partition d'un arbre minimum de Steiner

Ces propriétés d'arbres minimum de Steiner permettent d'envisager la construction de ce type d'arbres à l'aide des connexions qui sont, elles aussi, des arbres minimum de Steiner.

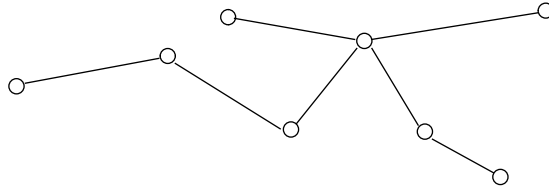
### 3.2 Quelques arbres minimum de Steiner élémentaires

Un *arbre simple* (ou étoile) est un arbre pour lequel il n'existe qu'au plus un noeud de degré supérieur à 2 dit noeud de branchement.

Soit  $X_g = \{x_i, i=1, \dots, g\}$  un groupe de noeuds dans un graphe  $G = (X, U, V)$  non orienté connexe. Appelons *centre du groupe*  $X_g$  le noeud  $x_c \in X$  pour lequel la somme des plus courts chemins aux membres du groupe  $X_g$  est minimale :

$$\sum_{x_i \in X_g} a'_{x_c x_i} \leq \sum_{x_i \in X_g} a'_{x_j x_i} \quad \forall x_j \in X$$

où  $a'_{ij}$  désigne la longueur des plus courts chemins de  $i$  à  $j$ .



**Figure 8 :** Arbre simple (étoile)

*Remarques :*

- Un centre n'appartient pas nécessairement au groupe de noeuds.
- Un groupe de noeuds peut avoir plusieurs centres dans le même graphe.
- La matrice des plus courts chemins du graphe permet de calculer facilement les centres d'un groupe.

Cette notion du centre nous semble intéressante pour calculer l'arbre de Steiner minimum de trois noeuds étant donné que le seul S-noeud de l'arbre ne peut être qu'un centre des trois noeuds.

Afin de maîtriser la complexité de l'algorithme de la construction, nous nous limitons à des topologies relativement simples et proposons l'énumération des cas où l'arbre de connexion doit couvrir quatre noeuds au maximum.

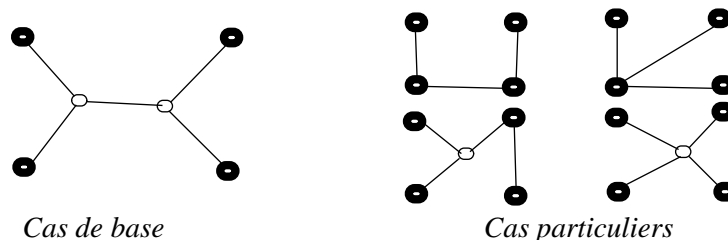
*Cas de deux noeuds :* les arbres minimum de Steiner du groupe composé de deux noeuds sont les plus courts chemins entre ces deux noeuds.

*Cas de trois noeuds :* si le groupe contient trois noeuds, l'arbre minimum de Steiner ne possède au plus qu'un S-noeud et c'est un arbre simple (étoile). Soit  $x_c$  un centre d'un tel groupe noté  $X_g$ . On peut envisager deux cas. Si  $x_c \in X_g$ , alors il n'y a pas de S-noeud dans l'arbre minimum de Steiner qui ne contient que les deux plus courts chemins reliant  $x_c$  avec les deux autres membres du groupe. Si  $x_c \notin X_g$ , alors l'arbre minimum de Steiner relie les trois noeuds au centre par les plus courts chemins. Dans tous cas, il existe au plus un noeud de branchement dans l'arbre et c'est le centre dans le deuxième cas de figure. Le(s) centre(s) du groupe détermine(nt) la structure minimale.

*Cas de quatre noeuds :* La topologie générique des arbres de Steiner d'un groupe formé de quatre noeuds est illustrée par la figure 9. Les noeuds du groupe à couvrir sont représentés en gras. Les autres noeuds de la figure indiquent les éven-



tuels S-noeuds qui n'appartiennent pas à l'ensemble des noeuds à recouvrir. Pour une analyse plus détaillée des topologies possibles, le lecteur peut se reporter à l'annexe.



**Figure 9 :** Topologie des arbres de Steiner d'un groupe à quatre noeuds

## 4 Algorithme de construction de l'arbre approché de l'arbre minimum de Steiner

Nous avons choisi l'algorithme de Kruskal[4] comme algorithme de base parce qu'il offre la possibilité d'une éventuelle parallélisation. Rappelons que dans cet algorithme, on initialise une forêt avec les noeuds du groupe à couvrir. Chaque pas de l'algorithme correspond à une phase d'unification des deux arbres les plus "proches". On les connecte par un plus court chemin dans la version originale de l'algorithme (cf. section 2.1.). La structure de connexion est un arbre minimum de Steiner dans l'heuristique que nous proposons (cf. l'idée de base dans la section 2.3.). Pour maîtriser le temps de calcul dans cette dernière, on choisit une topologie parmi celles qui sont données dans la section 3. Naturellement, cette heuristique nécessite la redéfinition de la notion distance des arbres (cf. plus loin).

Dans la suite, appelons *représentation* d'un arbre l'ensemble des noeuds à partir desquels on autorise la création des connexions. Au cours de la construction d'un arbre de Steiner à l'aide de l'algorithme de réunification, nous supposons que les connexions du niveau supérieur (les connexions dernièrement établies dans les différents arbres) sont susceptibles de participer à la création des nouvelles connexions, i.e., la création suit la hiérarchie de l'arbre.

A part les noeuds à recouvrir, les structures de connexion énumérées dans la section précédente peuvent être caractérisées à l'aide de leurs S-noeuds. Dans un premier temps, limitons la *représentation* de chacun des arbres de l'algorithme de Kruskal aux S-noeuds de la dernière connexion de l'arbre. Si la connexion ne possède pas de S-noeud, les extrémités du chemin utilisé comme connexion deviennent la représentation. Un noeud isolé est représenté par lui-même. De cette façon à chaque instant, chaque arbre de la forêt est représenté par 2 noeuds au maximum et seuls les noeuds représentatifs des arbres sont pris en compte pour examiner la distance des différents arbres ainsi que pour établir une nouvelle connexion. En fonction de la to-

pologie de la dernière connexion, on calcule la représentation du nouvel arbre et ses distances par rapport aux autres arbres.

Le squelette de l'algorithme s'écrit ainsi :

*Initialiser autant d'arbres isolés que de membres du groupe. Calculer les distances initiales à l'aide des plus courts chemins ;*

*Jusqu'à l'obtention d'un seul arbre, faire :*

*Sélectionner les deux arbres les plus proches ;*

*Relier ces arbres par l'intermédiaire d'un arbre minimum de Steiner couvrant les représentations des deux arbres ; Si le nombre des noeuds dans les deux représentations est égal à :*

*2 points : utiliser un des plus courts chemins,*

*3 points : utiliser un arbre simple avec son centre,*

*4 points : utiliser un arbre de Steiner avec deux S-noeuds*

*pour connecter les deux arbres.*

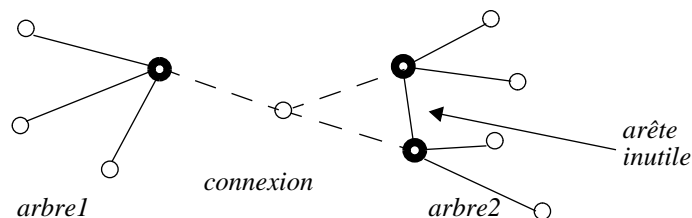
*Calculer la représentation de l'arbre ainsi créé et ses distances par rapport aux autres arbres.*

La "distance" entre deux arbres est définie et calculée de la manière suivante. Si chacune des deux représentations ne contient qu'un seul noeud, alors la distance correspond à la longueur des plus courts chemins entre les deux noeuds. Si la somme des noeuds dans les deux représentations est supérieure à deux, alors la distance est la longueur de l'arbre minimum de Steiner construit sur l'ensemble des noeuds des représentations diminuée de la longueur des arêtes qui deviennent inutiles dans les arbres ainsi connectés (la figure 10 illustre le cas de trois noeuds).

Eventuellement, la connexion basée sur trois points peut être réalisée en utilisant l'algorithme très simple ci-dessous :

*Calculer le centre des trois noeuds ;*

*Construire l'arbre de Steiner simple à partir du centre du groupe et éliminer la boucle obtenue.*

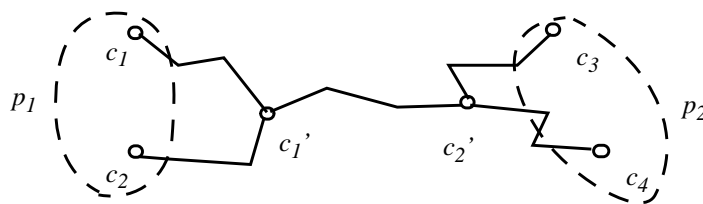


**Figure 10 :** Connexion basée sur trois noeuds

Insistons sur le fait que la distance des deux arbres correspond à la longueur de l'arbre obtenu diminuée par la longueur des éventuelles arêtes supprimées.

Dans le cas où chaque représentation possède deux points, la construction de l'arbre de connexion (construction d'un arbre de Steiner couvrant quatre noeuds) est différente. Soit  $C = \{c_1, c_2, c_3, c_4\}$  l'ensemble des noeuds des deux représentations et  $c_1', c_2'$  les deux S-noeuds possibles de l'arbre couvrant partiel. Notons par  $p_1$  le sous-ensemble de  $C$  dont les noeuds sont connectés à  $c_1'$  dans l'arbre couvrant à l'aide d'un plus court chemin et notons par  $p_2$  le sous ensemble complémentaire à  $C$  (cf. la figure 11). Pour trouver la connexion minimale à deux S-noeuds, nous proposons un algorithme d'énumération de topologies et de noeuds. Dans cet algorithme, on parcourt l'ensemble  $X$  des noeuds du graphe  $G$  pour énumérer les possibilités des S-noeuds  $c_1'$  et  $c_2'$ . Pour chaque couple  $(c_1', c_2')$ , on examine toutes les partitions  $(p_1, p_2)$  de  $C$ . Parmi les arbres possibles, on choisit l'arbre  $T'$  qui entraîne une longueur ajoutée minimale (l'énumération des arbres et les impossibilités sont présentées plus en détails dans l'annexe). L'algorithme de construction de la connexion s'écrit ainsi :

*Initialiser la longueur  $L = + \infty$  .*  
*Pour tout  $(c_1', c_2')$  dans  $X \times X$  faire :*  
*Pour tout  $(p_1, p_2)$  faire :*  
*S'il n'y a pas de contradiction entre  $(c_1', c_2')$  et  $(p_1, p_2)$ , alors*  
*Calculer le surplus de la connexion  $T'$  :*  
 $L' = \text{longueur}(T') - \text{longueur}(\text{arêtes inutiles});$   
*Si  $L' < L$  alors  $L = L'$ ; sauvegarder  $(T')$ ;*  
*Construire l'arbre de Steiner selon  $(T')$ ;*  
*Eliminer les boucles de la solution obtenue.*



**Figure 11 :** Topologie d'un arbre de Steiner de quatre noeuds

#### 4.1 Remarques sur l'algorithme

Nous supposons que la connexion dernièrement faite d'un arbre est susceptible de participer à de nouvelles connexions.

Dans notre heuristique, la recherche de l'arbre couvrant partiel minimum est limitée de deux points de vue. La première limitation est que, quand à la recherche des liaisons optimales entre deux arbres de la forêt, nous nous plaçons au niveau su-

périeur des arbres à connecter (i.e. les représentations ne contiennent que les S-noeuds de la dernière connexion des arbres). Dans d'autres heuristiques de construction d'arbres approchés de l'arbre minimum de Steiner, Ramanathan[2] propose un contrôle extensible pour régler le temps de calcul. Il trie les noeuds de l'arbre en construction et considère la racine puis les feuilles de l'arbre comme des noeuds prioritaires pour réaliser de nouvelles connexions. Il autorise une nouvelle connexion sur les  $k$  premiers éléments de la pile de priorité ainsi créée. Notre démarche correspond à l'utilisation d'une telle pile de priorité, en se limitant aux S-noeuds de la dernière connexion qui sont placés en-tête de la pile.

La deuxième limitation provient du fait que nous n'utilisons comme connexion que des arbres couvrants partiels à deux S-noeuds au maximum.

Ces limitations ne sont pas indépendantes. Si le nombre de noeuds dans les deux représentations est égale à  $m$  (e.g.  $m=2k$ , on veut examiner les  $k$  premiers noeuds des piles de priorité), le nombre des S-noeuds dans la connexion optimale ne peut pas dépasser  $(m-2)$ .

Des résultats plus performants pourraient être obtenus en descendant plus profondément dans la pile de priorité des noeuds des arbres et en utilisant des arbres minimum de Steiner plus complexes pour les connexions si nécessaire. La contrepartie de ces améliorations serait un coût plus important et une tendance à l'explosion combinatoire.

L'algorithme de Kruskal n'est pas le seul candidat qui peut être amélioré par l'utilisation des arbres pour connecter des structures intermédiaires au cours de la construction de l'arbre couvrant partiel. D'autres heuristiques (par exemple l'algorithme basé sur l'idée de Prim [3]) qui utilisent les plus courts chemins pour établir les connexions peuvent aussi être améliorées en appliquant des arbres de Steiner simples.

## 4.2 La complexité de l'algorithme et l'illustration des résultats

Pour pouvoir situer notre heuristique dans l'ensemble des heuristiques fournissant des arbres approchés de l'arbre minimum de Steiner, l'analyse de la complexité de l'algorithme et celle de la performance des arbres obtenus sont nécessaires. La performance de l'arbre d'une éventuelle diffusion multicast peut être caractérisée par la longueur de l'arbre.

On désigne par  $N > 1$  et  $g > 1$  le nombre de noeuds respectivement dans  $X$  et  $X_g$ . On suppose que la matrice des plus courts chemins existe.

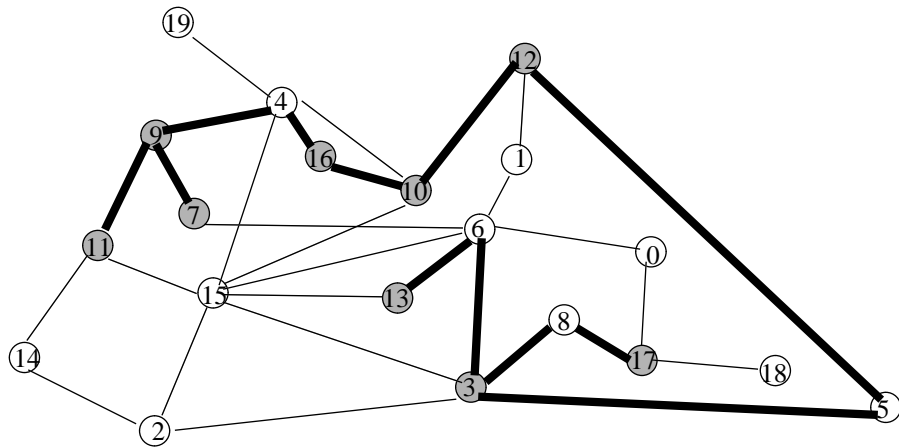
L'algorithme proposé nécessite  $(g-1)$  unifications d'arbres. Après chaque regroupement, il faut déterminer la distance minimale du nouvel arbre par rapport aux autres arbres. Le calcul de la distance implique aussi la détermination de la structure qui correspond à la distance minimale. Il y a donc  $g(g-1)/2$  calculs à faire.

Le calcul le plus coûteux est celui réalisé pour le cas à quatre noeuds ( $|C| = 4$ ). Dans le pire des cas : il faut toujours utiliser ce calcul (selon nos expériences, dans un tiers des cas on utilise cet algorithme coûteux et dans les autres, la distance et la connexion peuvent être calculées sur la base de deux ou de trois noeuds). La détermination d'arbre minimum de Steiner d'un groupe de quatre noeuds à l'aide de l'algorithme présenté nécessite l'examen de  $6 \cdot N^2$  configurations différentes. On peut choisir  $N$  fois  $c_1$  et  $c_2$  chacun ; pour  $c_1$  et  $c_2$  donnés, il existe 6 possibilités pour créer des partitions  $(p_1, p_2)$  valides et différentes (cf. l'annexe).

Finalement, dans le pire des cas, le temps d'exécution est borné par  $O(g^2 N^2)$ .

Pour illustrer la qualité du résultat de cette heuristique, nous avons choisi un graphe et un groupe multicast générés aléatoirement. Le graphe est composé de 20 noeuds et ses arêtes ont été générées aléatoirement jusqu'à l'obtention d'un graphe connexe sur les 20 noeuds. Pour simplifier l'illustration, les poids associés aux différentes arêtes sont identiques (tous égaux à 1, par exemple). Les noeuds du groupe multicast sont aussi choisis aléatoirement ;  $X_t = \{12, 7, 17, 10, 3, 16, 13, 11, 9\}$ .

Les figures 12, 13 et 14 présentent respectivement l'arbre couvrant partiel obtenu par l'algorithme d'origine de Kruskal, celui obtenu par notre heuristique et l'arbre minimum de Steiner. Dans cet exemple, l'algorithme de Kruskal fournit un résultat très proche de la longueur de l'arbre minimum de Steiner. Néanmoins, l'heuristique proposée est capable d'améliorer la performance de l'arbre.



**Figure 12 :** Arbre construit par l'algorithme de Kruskal : 12 arêtes

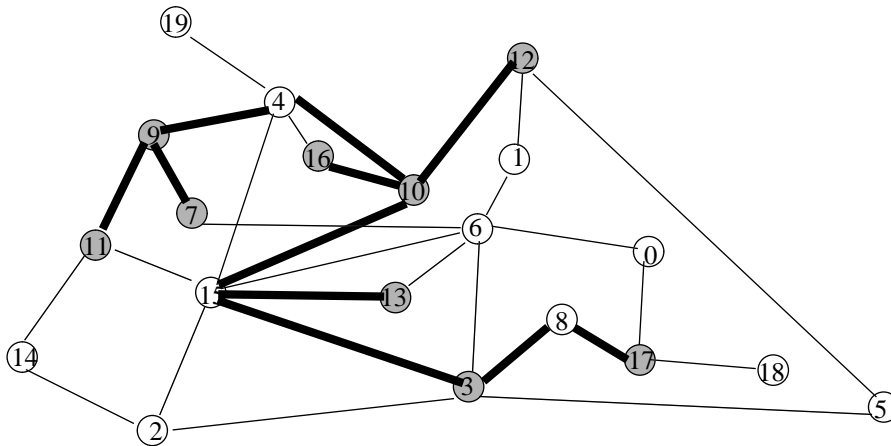


Figure 13 : Arbre construit par notre heuristique : 11 arêtes

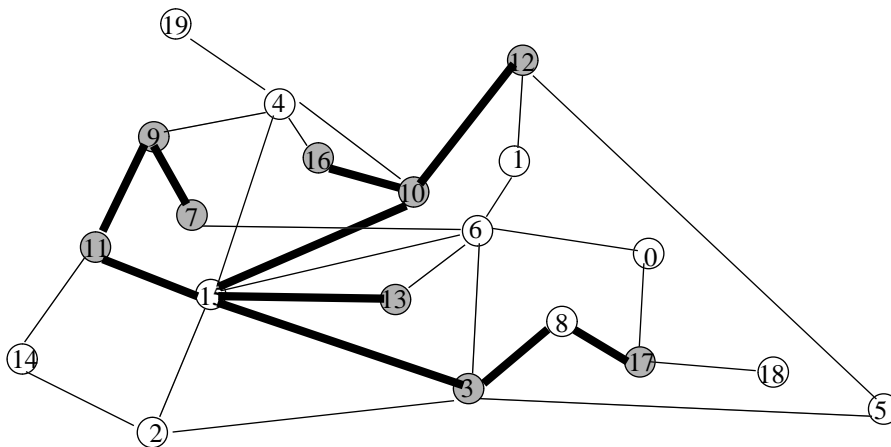


Figure 14 : Arbre de Steiner : 10 arêtes

## 5 Conclusion

Dans cette étude, nous avons considéré un sous-ensemble des heuristiques qui permettent de construire en temps polynomial des arbres couvrants partiels quasi-optimaux (la construction exacte d'un arbre minimum de Steiner est NP-complète). Parce qu'il existe des groupes de noeuds dont l'arbre couvrant obtenu par les plus courts chemins est éloigné de l'arbre minimum de Steiner du groupe, nous avons constaté dans certains cas que la performance des arbres construits sur la base des plus courts chemins ne peut pas être améliorée. Pour trouver des solutions plus favorables, nous avons proposé de modifier certaines heuristiques pour construire des ar-

bres dits “de Steiner”. La modification consiste essentiellement à introduire des connexions sous forme d’arbres de Steiner.

Nous avons basé notre démarche sur l’heuristique de Kruskal dans la mesure où l’algorithme correspondant peut être distribué. En limitant la structure des liaisons possibles entre composants (i.e., entre les arbres d’une forêt dans l’algorithme de Kruskal), le temps de recherche est polynomial. La complexité des arbres de liaison et la profondeur de la recherche des liaisons possibles entre les composants sont des paramètres qui peuvent fournir un compromis entre performance et temps d’exécution. La comparaison des résultats et de ces temps d’exécution (i.e., la comparaison de la performance des différents algorithmes) fera l’objet d’une étude ultérieure.

## Remerciements

Je tiens à exprimer mes sincères remerciements à MM. Raymond MARIE et Patrice LEGUESDRON qui m’ont beaucoup aidé à réaliser ce rapport.

## Bibliographie

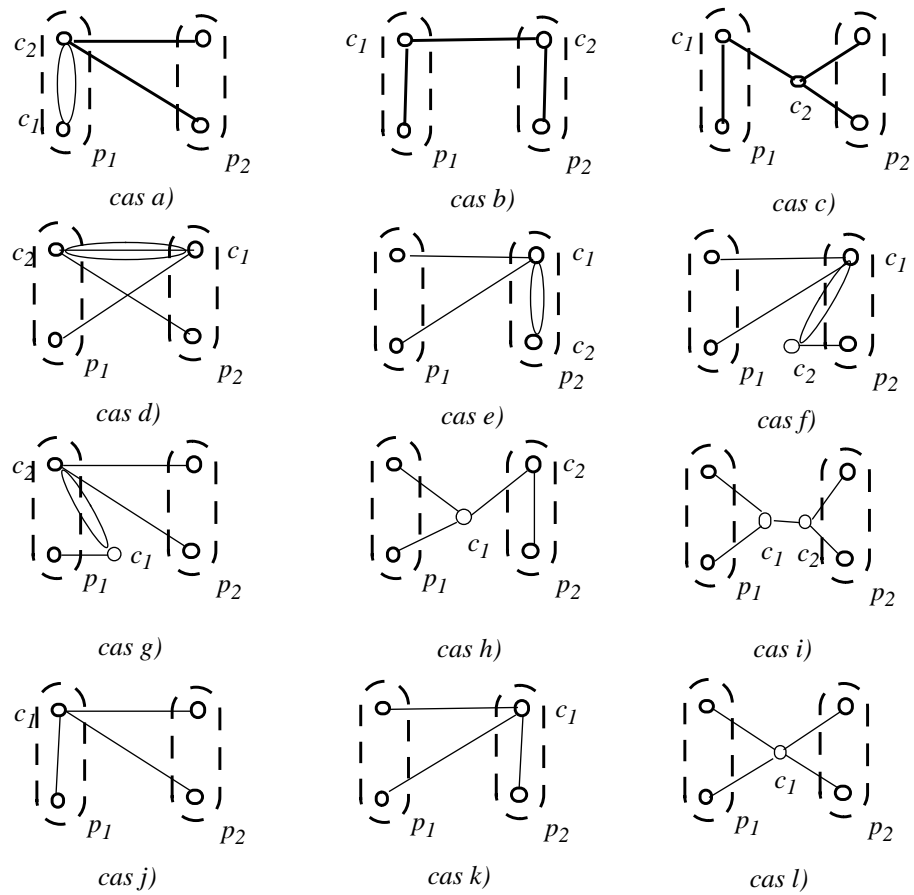
- [1] **P. Winter**, “*Steiner problem in networks: A survey*”, Networks, vol. 17. pp. 129-167, 1987.
- [2] **S. Ramanathan**, “*Multicast Tree Generation in Networks With Asymmetric Links*”, IEEE/ACM Transactions on Networking, vol. 4. no. 4., 1996.
- [3] **H. Takahashi, A. Matsuyama**, “*An approximate solution for the Steiner problem in graphs*”, Math. Japonica, vol. 24., 1981.
- [4] **J. Kruskal**, “*On the shortest spanning subtree of a graph and the traveling salesman problem*”, Proc. Amer. Math. Soc., vol. 16. pp. 48-50, 1956.
- [5] **F. Bauer, A. Varma**, “*Distributed Algorithms for Multicast Path Setup in Data Networks*”, IEEE/ACM Transactions on Networking, vol. 4. no. 2. 1996.
- [6] **L. Kou, G. Markowsky, L. Berman**, “*A fast algorithm for Steiner trees*”, Acta Informatica, vol. 15. pp. 141-145, 1981.
- [7] **V. P. Kompella, J. C. Pasquale, G. C. Polyzos**, “*Multicast routing for multimedia communications*”, IEEE/ACM Transactions on Networking, vol. 1. no. 3, 1993.
- [8] **F.K. Hwang, D. S. Richards**, “*Steiner Tree Problems*”, Networks, vol. 22. pp. 55-89, 1992.
- [9] **I. Charon, A. Germa, O. Hudry**, “*Méthode d’optimisation combinatoire*”, Masson, pp. 50-51, 1996.
- [10] **A.Z. Zelikovsky**, “*A 11/6-Approximation Algorithm for the Network Steiner Problem*”, Algorithmica, vol. 9. pp. 463-470, 1993.

## Annexe : Énumération de topologies des arbre de Steiner d'un groupe de quatre noeuds

La topologie d'un arbre minimum de Steiner couvrant quatre noeuds ne peut avoir au plus que deux S-noeuds. Pour relier les quatre noeuds et les deux S-noeuds, on utilise cinq plus courts chemins au plus (cf. figure 11). Les différents arbres de Steiner qui couvrent les quatre noeuds sont des cas particuliers de cette topologie générale comme l'énumération ci-dessous l'indique. En reprenant les notations de la figure 11,  $p_1$  désigne la paire de noeuds reliée à  $c_1'$  directement par deux plus courts chemins et  $p_2$  la paire de noeuds reliée directement à  $c_2'$ . Pour simplifier, introduisons l'ensemble  $p_3$  de noeuds qui désigne les noeuds du graphe  $G$  n'appartenant ni à  $p_1$  ni à  $p_2$ . Dans la suite, on énumère toutes les combinaisons possibles en ce qui concerne la relation entre les S-noeuds  $c_1'$ ,  $c_2'$  et leurs relations vis-à-vis des ensembles  $p_1$ ,  $p_2$  et  $p_3$  afin de ne retenir que celles qui correspondent à des arbres de Steiner à 0,1 ou 2 S-noeuds (cf. figure 15) :

- a. )  $c_1' \in p_1$  et  $c_2' \in p_1$  et  $c_1' \neq c_2'$ . Cette configuration introduit des boucles (cf. figure 15), elle ne correspond donc pas à un arbre de Steiner
- b. )  $c_1' \in p_1$  et  $c_2' \in p_2$  et  $c_1' \neq c_2'$
- c. )  $c_1' \in p_1$  et  $c_2' \in p_3$  et  $c_1' \neq c_2'$
- d. )  $c_1' \in p_2$  et  $c_2' \in p_1$  et  $c_1' \neq c_2'$ . Cette configuration est impossible (cf. figure 15)
- e. )  $c_1' \in p_2$  et  $c_2' \in p_2$  et  $c_1' \neq c_2'$ . Cette configuration est impossible (cf. figure 15)
- f. )  $c_1' \in p_2$  et  $c_2' \in p_3$  et  $c_1' \neq c_2'$ . Cette configuration est impossible (cf. figure 15)
- g. )  $c_1' \in p_3$  et  $c_2' \in p_1$  et  $c_1' \neq c_2'$ . Cette configuration est impossible (cf. figure 15)
- h. )  $c_1' \in p_3$  et  $c_2' \in p_2$  et  $c_1' \neq c_2'$
- i. )  $c_1' \in p_3$  et  $c_2' \in p_3$  et  $c_1' \neq c_2'$
- j. )  $c_1' = c_2' \in p_1$
- k. )  $c_1' = c_2' \in p_2$
- l. )  $c_1' = c_2' \in p_3$ .





**Figure 15 :** Topologies correspondant aux différents cas particuliers des arbres de Steiner de quatre noeuds



---

Unité de recherche INRIA Lorraine, technopôle de Nancy-Brabois, 615 rue du jardin botanique, BP 101, 54600 VILLERS-LÈS-NANCY  
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN  
Unité de recherche INRIA Rocquencourt, domaine de Voluceau, Rocquencourt, BP 105, LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur

INRIA, Domaine de Voluceau, Rocquencourt, BP 105 LE CHESNAY Cedex (France)  
<http://www.irisa.fr>

ISSN 0249-6399