



HAL
open science

Interconnexion multi-groupes

Lélia Blin, Christian Laforest, Stephane Rovedakis, Nicolas Thibault

► **To cite this version:**

Lélia Blin, Christian Laforest, Stephane Rovedakis, Nicolas Thibault. Interconnexion multi-groupes. 9ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2007, Ile d'Oléron, France. pp.41-44. inria-00176947

HAL Id: inria-00176947

<https://inria.hal.science/inria-00176947>

Submitted on 5 Oct 2007

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.

Interconnexion multi-groupes

Lélia Blin, Christian Laforest, Stephane Rovedakis, Nicolas Thibault

Tour Evry 2, IBISC / CNRS, Université d'Evry, 523 place des Terrasses, 91000 Evry, France
Email: {lelia.blin, christian.laforest, stephane.rovedakis, nicolas.thibault}@ibisc.fr

Cet article est consacré au problème de l'interconnexion de plusieurs *groupes* dans un réseau. Nous cherchons ici à réserver des ressources (liens de communication) du réseau, afin d'établir des connexions efficaces entre les *membres* de chaque groupe. Ces connexions doivent fournir des garanties sur la *latence maximum* (ou temps de communication maximum) de chaque groupe dans la ou les structures formées par les ressources réservées. Nous allons montrer que les approches pour interconnecter des groupes qui conservent les diamètres donnent lieu à des problèmes algorithmiques difficiles. Ces résultats nous conduisent à proposer une autre approche consistant à construire un seul arbre pour interconnecter tous les groupes. Nous proposons ici le premier algorithme avec garantie de performance mettant en œuvre cette approche.

Keywords: Algorithmique, Approximation, Graphe, Réseau, Structure de connexion

Nous nous intéressons au problème de la qualité des communications au sein de *groupes* composés de machines (les *membres* du groupe) disséminées dans un réseau point-à-point. Dans le cadre d'applications réseaux comme les réunions virtuelles ou les video-conférences, il est important de garantir une bonne qualité de communications entre les membres d'un même groupe. Une façon de réaliser cela est de réserver et d'allouer des ressources pour interconnecter ces groupes afin d'éviter les engorgements dus au trafic extérieur ou une instabilité du réseau (changement de routes). Une approche algorithmique donnant des garanties de performance sur les communications au sein de chaque groupe et non de façon globale est plus réaliste, ce type d'approche n'a à notre connaissance pas été faite à ce jour, et fait l'objet de cet article. Notons que la réservation de ressources avec garantie de performance pour *un seul* groupe d'utilisateurs a déjà été abordée (en séquentiel [IKL00] et en réparti [BBL06]). La présence de plusieurs groupes entraîne des problématiques spécifiques : il faut gérer le *partage* des ressources entre les groupes et faire en sorte que le trafic d'un groupe ait le moins d'impact possible sur le trafic des autres.

Pour être plus précis sur ce que l'on entend ici par « communications performantes », nous introduisons les quelques notations suivantes. Soit $G = (V, E, w)$ un graphe quelconque, avec V l'ensemble de ses sommets (représentant l'ensemble des machines du réseau), E l'ensemble de ses arêtes (représentant l'ensemble des liens) et w une fonction de poids positive sur les arêtes ($\forall e \in E, w(e) > 0$ représente le temps de traversée du lien e). On notera $d_G(u, v)$ la *distance entre u et v dans G* , c'est-à-dire la somme des poids d'un plus court chemin entre u et v dans G , représentant la durée d'acheminement d'une information entre u et v . Un *groupe* M est un sous ensemble de machines du réseau ($M \subseteq V$). Une *structure* $S = (V_S, E_S)$ *couvrant* M dans G est un sous graphe de G vérifiant $M \subseteq V_S \subseteq V$ et $E_S \subseteq E$. Le *facteur d'élongation* induit par S sur M est le rapport $\frac{D_S(M)}{D_G(M)}$ où $D_S(M)$ est le diamètre du groupe M dans la structure de connexion S et $D_G(M)$ le diamètre du groupe M dans le graphe ; ce facteur traduit l'impact sur la durée maximale des communications entre les membres de M si elles ont lieu dans la structure S .

Dans cet article, nous montrons (dans la section 1) que minimiser le nombre d'arêtes d'une structure ou construire des structures disjointes pour chaque groupe, tout en conservant le même diamètre que dans le graphe (facteur d'élongation de 1) donne lieu à des problèmes algorithmiques difficiles. Ces résultats nous conduisent à envisager une autre approche consistant à construire un seul arbre pour interconnecter tous les groupes. Pour cela, nous proposons et analysons dans la section 2 les performances d'une méthode de construction qui permet de réaliser un certain compromis entre temps de communication (résultats sur l'élongation maximum et moyenne) des groupes, nombre de liens réservés et facilité du routage (grâce au fait que l'on alloue un arbre).

1 Deux approches qui conservent le diamètre

Dans cette section, nous nous intéressons spécifiquement aux structures de connexion qui conservent le diamètre (facteur d'élongation de 1) afin d'obtenir les meilleures performances possible dans chaque structure. Nous montrons que cette contrainte est trop forte et donne des problèmes algorithmiques difficiles.

Minimisation du nombre d'arêtes dans une structure préservant le diamètre. Lorsque plusieurs groupes doivent être interconnectés, il est naturel d'essayer de faire en sorte que chaque groupe utilise le moins de ressources possible. Nous allons donc essayer ici, pour un groupe M donné, de construire une structure préservant son diamètre et utilisant le moins d'arêtes possible. Formellement, cela donne le problème 1.

Problème 1

Instance : Un graphe $G = (V, E)$ non pondéré (ou avec $w(e) = 1, \forall e \in E$), un groupe $M \subseteq V$.

Question : Existe-t-il une structure S préservant le diamètre de M dans G (i.e. telle que $D_S(M) = D_G(M)$), tel que $|E_S| = \min\{|E_{S'}| : S' \text{ une structure couvrant } M \text{ et } D_{S'}(M) = D_G(M)\}$?

Théorème 1.1 \forall constante α , il n'existe pas d'algorithme α -approché pour le problème 1, sauf si $P = NP$.

On prouve ce résultat par contradiction en montrant que s'il existe un algorithme polynomial α -approché (α constant) pour le problème 1 alors on peut construire un algorithme polynomial β -approché (β constant) pour le problème *minimal set cover*; or, ce problème est non APX (c'est-à-dire non approximable par une constante).

Structures deux à deux arêtes disjointes préservant le diamètre. L'approche précédente conduit à un problème pour lequel il n'existe pas d'algorithme d'approximation avec un rapport constant, nous allons maintenant essayer d'allouer des structures préservant le diamètre et utilisant des liens totalement indépendants. Ainsi, la qualité en terme de durée maximum de communications peut être garantie : le diamètre pour chaque groupe connecté est minimum et les liens étant indépendants, le trafic d'un groupe ne vient pas perturber celui d'un autre groupe. Formellement, ceci s'exprime par le problème 2.

Problème 2

Instance : Un graphe $G = (V, E)$ non pondéré, $M_1, M_2 \subseteq V$.

Question : Existe-t-il deux structures arêtes disjointes préservant le diamètre de M_1 et M_2 ? C'est-à-dire existe-t-il $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ tel que : $M_i \subseteq V_i \subseteq V, E_1 \cap E_2 = \emptyset, E_i \subseteq E$, et $D_{G_i}(M_i) = D_G(M_i)$, avec $i = 1, 2$.

Théorème 1.2 Le problème 2 est NP-complet.

On prouve ce résultat en se réduisant au problème *set splitting* (voir [GJ79]). Ce résultat reste vrai même lorsque $M_1 \cap M_2 = \emptyset$, ou bien lorsque $M_1 = M_2 = V$, ou encore lorsque $D_G(M_i) = 3$ ou bien $D_G(M_i) \geq 6$.

2 Notre solution au problème

Les résultats de la section 1 montrent qu'il est difficile de trouver plusieurs structures disjointes pour interconnecter tous les groupes. Nous proposons donc de construire une unique structure connectant tous les groupes. De plus, nous avons également montré la difficulté (non approximable par une constante) de construire une structure préservant le diamètre de poids minimal.

Ces résultats négatifs nous amènent à nous concentrer sur le problème suivant : étant donné un ensemble de groupes, construire *un arbre* couvrant tous les groupes et minimisant le diamètre de chaque groupe dans l'arbre. La structure d'arbre présente l'avantage de simplifier les mécanismes de routage et de duplication des informations à faire circuler entre les membres d'un groupe du réseau. En effet, il n'y a qu'une seule route possible entre deux sommets d'un arbre, le routage est donc trivial. De plus, l'absence de cycle simplifie la diffusion de messages d'un membre vers tous les autres : cette diffusion peut être faite de manière naturelle, par inondation, sans contrôle complexe.

Notre Algorithme. Nous présentons maintenant les grandes lignes de notre algorithme. Celui-ci prend en entrée un graphe G , et l'ensemble des g groupes. En sortie, l'algorithme renvoie un arbre A_g couvrant chacun des g groupes.

1. Construire pour chaque groupe M_i ($1 \leq i \leq g$) un arbre de plus courts chemins, enracinés en un membre quelconque du groupe, couvrant M_i .
2. Soit T_1, \dots, T_g , la suite des arbres ainsi obtenus classés par diamètres croissants.
3. Soit $A_1 = T_1$. Pour tout i ($2 \leq i \leq g$), on construit A_i en fusionnant A_{i-1} et T_i de sorte que A_i soit sans cycle (on supprime les éventuels cycles en retirant des arêtes de T_i sans toucher à A_{i-1}) couvrant $M_1 \cup \dots \cup M_i$.

Remarque : L'algorithme décrit ci-dessus ne retourne pas toujours une structure connexe. Il suffit, pour obtenir un arbre, d'ajouter des chemins pour relier les composantes connexes de la structure A_g construite. Notons que l'ajout de ces chemins ne change en rien l'analyse de l'algorithme.

Nous allons donc maintenant nous intéresser à l'analyse des performances de notre algorithme.

Le facteur d'élongation maximum

Etant donné un arbre T couvrant g groupes M_1, \dots, M_g , la dégradation du diamètre subie par chaque groupe M_i dans T est quantifiée par le *facteur d'élongation* induit par T sur M_i : $\frac{D_T(M_i)}{D_G(M_i)}$. Ainsi, une mesure de performance simple est la plus grande valeur des facteurs d'élongation des groupes dans T , c'est-à-dire le *facteur d'élongation maximum* induit par T : $\max \left\{ \frac{D_T(M_i)}{D_G(M_i)} : i = 1, \dots, g \right\}$. Le but est alors de construire un arbre T couvrant $M_1 \cup \dots \cup M_g$ minimisant le facteur d'élongation maximum. On pose $\alpha = \min \left\{ \max \left\{ \frac{D_T(M_i)}{D_G(M_i)}, i = 1, \dots, g \right\} : T \text{ arbre couvrant } M_1 \cup \dots \cup M_g \right\}$.

Borne inférieure. On peut voir sur l'exemple de la figure 1(a) que tout algorithme connectant tous les groupes (ayant ici chacun deux membres) par un arbre, construit obligatoirement un arbre qui ne peut pas avoir un facteur d'élongation maximum plus petit que $g - 1$, ainsi : $\alpha \geq g - 1$. En effet, pour obtenir un arbre T , il faut supprimer une arête du cycle entre i et $i + 1$ (ou entre g et 1), mais cela éloigne du même coup les deux membres du groupe M_i (ou M_g), induisant un facteur d'élongation de T de $\frac{D_T(M_i)}{D_G(M_i)} = \frac{g-1+\epsilon}{1+\epsilon}$ qui tend vers $g - 1$ lorsque ϵ tend vers 0.

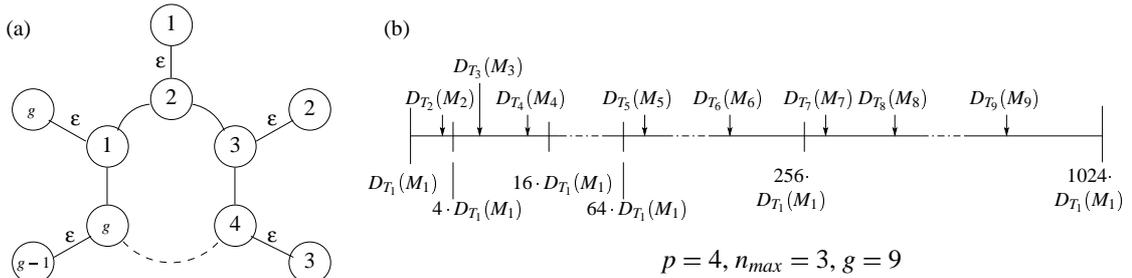


FIG. 1: (a) exemple exhibant la borne inférieure de tout algorithme pour le facteur d'élongation maximum, (b) exemple de partition en paliers des diamètres dans les structures T_i des groupes M_i .

Borne supérieure. On montre que notre algorithme construit des arbres dont le facteur d'élongation maximum est toujours inférieur ou égal à $2g$. En effet, comme les groupes M_i sont ajoutés en suivant l'ordre croissant des diamètres $D_{T_i}(M_i)$, on a : $D_{A_i}(M_i) \leq iD_{T_i}(M_i)$. Par ailleurs, il est facile de voir que $D_{T_i}(M_i) \leq 2D_G(M_i)$, d'où : $D_{A_i}(M_i) \leq 2iD_G(M_i)$; comme $A_i \subseteq A_g$ on a $D_{A_g}(M_i) \leq 2iD_G(M_i)$.

Ainsi, $\max \left\{ \frac{D_{A_g}(M_i)}{D_G(M_i)} : i = 1, \dots, g \right\} \leq 2g$. Par ailleurs, nous avons prouvé que la borne de $2g$ est atteinte, c'est-à-dire que, dans certains cas, l'algorithme construit des arbres qui induisent un facteur d'élongation maximum de $2g$.

Notons qu’une analyse similaire sur le diamètre de plusieurs groupes a été menée dans [KLPT06] mais dans un autre cadre, puisque les différents groupes sont construits de manière incrémentale. Cette méthode « classique » d’évaluation n’est cependant pas assez fine pour capturer *toutes* les dégradations subies par les groupes. En effet, dans la figure 1(a) *un seul* groupe subit une dégradation importante, alors que les autres groupes conservent leurs diamètres dans le graphe. Or ceci n’est pas reflété par le facteur d’élongation maximum. Pour cela nous allons utiliser un paramètre plus fin d’évaluation : *le facteur d’élongation moyen*.

Le facteur d’élongation moyen

Nous définissons maintenant le *facteur d’élongation moyen* induit par un arbre T couvrant les g groupes M_1, \dots, M_g par $\frac{1}{g} \cdot \sum_{i=1}^g \frac{D_T(M_i)}{D_G(M_i)}$. Pour représenter plus finement l’impact de la disparité des diamètres sur ce facteur, nous allons définir et utiliser la notion de *palier*. Comme dans l’algorithme, les groupes M_i sont triés par valeurs croissantes des diamètres $D_{T_i}(M_i) : D_{T_1}(M_1) \leq \dots \leq D_{T_g}(M_g)$. Etant donnée une valeur $p > 1$, le palier numéro $k \geq 0$ contient tous les groupes M_i dont le diamètre $D_{T_i}(M_i)$ vérifie : $p^k D_{T_1}(M_1) \leq D_{T_i}(M_i) < p^{k+1} D_{T_1}(M_1)$. La *taille d’un palier* est le nombre de groupes qu’il contient (N.B. certains paliers peuvent être vides). Etant donné p , on note n_{max} la taille maximum d’un palier. La figure 1(b) donne une illustration de partition de $g = 9$ groupes en paliers, avec la valeur $p = 4$. Sur cet exemple, pour $p = 4$, on a $n_{max} = 3$.

Théorème 2.1 *Notre algorithme construit des arbres dont le facteur d’élongation moyen est d’au plus :*
 $\min \left\{ g + 1, 2n_{max} \cdot \frac{p}{p-1} \right\}$, *pour tout* $p > 1$. *La borne* $g + 1$ *est atteinte.*

La borne $2n_{max} \cdot \frac{p}{p-1}$ est plus intéressante que $g + 1$ lorsque p est « grand » et n_{max} « petit ». Ceci est le cas lorsque les diamètres des groupes sont très différents les uns des autres. Par exemple, si le diamètre de chaque groupe est 3 fois plus grand que le précédent, alors, en prenant $n_{max} = 1$ et $p = 3$, on a $2n_{max} \cdot \frac{p}{p-1} = 3$, alors que $g + 1$ peut être arbitrairement grand.

Notons que l’analyse présentée ici s’intéresse aux pires instances que doit traiter notre approche. En effet, c’est dans le cas où les membres des groupes sont « entremêlés » que l’on constate les plus fortes dégradations. Par contre, si les groupes sont complètement « isolés » les uns des autres dans le réseau, la dégradation sera minimale puisque les arbres vont pouvoir se déployer sans interférer les uns sur les autres.

3 Bilan et perspectives

Nous avons présenté et analysé le premier algorithme avec garantie de performance pour interconnecter des groupes dispersés sur un réseau. Nous avons montré que notre algorithme induisait dans le pire cas un facteur d’élongation maximum optimal en ordre de grandeur. Une première analyse nous a permis de montrer que notre algorithme induisait un facteur d’élongation moyen constant dans certain cas et d’au plus $g + 1$ dans le cas général. Nous espérons dans un futur proche affiner notre algorithme et son analyse en exploitant de façon plus pertinente la localisation relative des groupes dans le graphe. Nous souhaitons aussi arriver à mieux maîtriser les engorgements induits par les divers groupes entre eux dans l’arbre. Enfin, pour mieux coller à la réalité des réseaux à grandes échelles nous travaillons sur une version répartie de notre travail.

Références

- [BBL06] Fabien Baille, Lélia Blin, and Christian Laforest. Distributed approximation allocation resources algorithm for connecting groups. In *Euro-Par*, pages 519–529, 2006.
- [GJ79] M. Garey and D. Johnson. Computers and intractability. In *Freeman and compagny*, 1979.
- [IKL00] Alexis Irlande, Jean-Claude König, and Christian Laforest. Construction of low-cost and low-diameter steiner trees for multipoint groups. In *SIROCCO*, pages 197–210, 2000.
- [KLPT06] Ralf Klasing, Christian Laforest, Joseph Peters, and Nicolas Thibault. Constructing incremental sequences in graphs. *Algorithmic Operations Research*, 1(2) :1–7, 2006.