Reconstruire un graphe en une ronde

Florent Becker, Martin Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, Ioan Todinca

Reconstruire un graphe en une ronde †

F. Becker[1], M. Matamala[2,3], N. Nisse[4], I. Rapaport[2,3], K. Suchan[5,6], I. Todinca[1]

[1] *LIFO, Université d'Orléans, France*
[2] *Departamento de Ingeniería Matemática, Universidad de Chile, Santiago, Chile*
[3] *Centro de Modelamiento Matemático (UMI 2807 CNRS), Univ. de Chile, Santiago, Chile*
[4] MASCOTTE*, INRIA, I3S (CNRS/Univ. Nice Sophia Antipolis), France*
[5] *Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Santiago, Chile*
[6] *Faculty of Applied Mathematics, AGH - Univ. of Science and Technology, Cracow, Poland*

Nous étudions quelles propriétés d'un réseau peuvent être calculées à partir d'une petite quantité d'informations locales fournie par ses noeuds. Notre modèle est une restriction de $\mathcal{CONGEST}$, un modèle distribué classique. Il est proche du modèle de complexité de communication avec messages simultanés de Babai *et al.* [BGKL04]. Chacun des n noeuds – qui ne connaissent que leur identifiant, ceux de leurs voisins et la taille du graphe– envoie un message de taille $O(\log(n))$ bits à une entité centrale, le superviseur. Celui-ci doit alors déterminer une certaine propriété du réseau. Nous montrons que des questions telles que : "Est-ce que le graphe contient un triangle ? un carré ? Quel est son diamètre ?" ne peuvent pas être résolues dans ce modèle. En revanche, pour de nombreuses classes de graphes : celles de dégénérescence bornée (incluant les graphes planaires, ceux de largeur arborescente bornée...), les sommets peuvent succinctement donner une description complète du graphe au superviseur. Nous laissons ouverte la question de décider la connexité.

Keywords: systèmes distribués, théorie des graphes, complexité de communication.

1 Calcul local de propriétés dans les graphes

Prologue. Lorsque, au cours du banquet d'AlgoTel, les organisateurs lancent un "Comment ça va ?", il est aisé de se faire une idée globale de l'ambiance juste en entendant les réponses simultanées des participants. Au contraire, si avant de rentrer, les organisateurs demandaient aux participants de se compter, il serait difficile d'obtenir une réponse exacte des décomptes locaux proposés. En effet, si un chercheur Sophipolitain répond "nous sommes 5 par ici" tandis qu'un autre lance "6 par là !", comment savoir si certains membres de l'équipe ont été comptés plusieurs fois ?

Le sujet de ce travail est de déterminer quelles propriétés globales d'un graphe peuvent être déterminées à partir de peu d'information locale. Grossièrement, supposez qu'une question est posée aux sommets d'un réseau, ceux-ci doivent donner des informations succinctes fonction uniquement de leur voisinage. Une autorité centrale, le superviseur, collecte ces réponses et doit en déduire la réponse à la question. Les principales difficultés viennent de la taille limitée des réponses ainsi que de leur simultanéité, i.e., les sommets ne savent pas ce que vont répondre leurs voisins et il est donc *a priori* difficile de partager l'information pertinente.

Modèles de communications. Lorsque l'on se réfère à un "réseau", on y pense habituellement comme à un système distribué où les nœuds correspondent à des agents ou des processeurs. Ces nœuds peuvent uniquement interagir localement, et, du fait du manque de connaissance et d'informations globales, de nouvelles notions algorithmiques et de complexité apparaissent. Au contraire de l'algorithmique "classique" où la machine de Turing fait office de consensus comme modèle formel à l'algorithme, dans le cas des systèmes distribués, il existe de nombreux modèles pour les protocoles de communications. Des modèles

† Une version étendue de ces travaux a été acceptée à IPDPS'11 [BMN+11]. Partially supported by ANR projects AGAPE (N. Nisse, I. Todinca) and DIMAGREEN (N. Nisse) and by the ECOS-Conicyt project C09E04 (M. Matamala, I. Rapaport, I. Todinca).

théoriques simples ont été conçus pour étudier divers aspects des protocoles tels que la tolérance aux pannes, le synchronisme, la localité, la congestion, etc.

Dans le modèle $\mathcal{CONGEST}$ (voir le livre [Pel00]), un réseau est représenté par un graphe dont les n sommets correspondent aux processeurs du réseau, et les arêtes aux liens entre ces processeurs. La communication est synchrone : à chaque ronde, chaque processeur peut envoyer un message de taille $O(\log n)$ bits par chacun des liens sortants auquels il est incident. Différentes variations au modèle $\mathcal{CONGEST}$ ont été proposées. L'idée générale est de supprimer certaines restrictions, rendant le modèle plus puissant, dans le but de se concentrer sur des problèmes particuliers. Dans ce contexte, Linial [Lin92] introduit le modèle appelé $\mathcal{LOCAL}$ (voir aussi [Pel00]) où la contrainte sur la taille des messages est levée afin de se concentrer sur le problème de la localité dans les systèmes distribués. Répondre à la question "*Qu'est ce qui peut être calculé localement ?*" dans le modèle $\mathcal{LOCAL}$ apparaît comme un problème crucial. Kuhn *et al.* montrent que des problèmes difficiles comme la couverture minimum ou l'ensemble dominant minimum ne peuvent être approchés lorsque les processeurs échangent des messages arbitrairement longs pendant un nombre borné de rondes [KMW04]. Grumbach et Wu s'intéressent au *calcul frugal* dans le modèle $\mathcal{CONGEST}$ [GW09], i.e., les calculs pour lesquels la quantité totale d'information traversant chaque arête est bornée par $O(\log n)$. Dans ce cadre, ils ont montré que pour tout réseau planaire ou de degré borné, toute formule logique du premier ordre peut être calculée frugalement.

D'un point de vue centralisé, tester les propriétés de graphes à l'aide de connaissances locales a aussi été étudié [Fis01, GR02]. Étant donnée la matrice d'adjacence d'un graphe, il s'agit de déterminer le nombre minimum de questions élémentaires, comme "quelle est le i^e voisin du sommet v ?", nécessaires pour décider si le graphe satisfait une certaine propriété. Par exemple, [AKKR08] propose des bornes inférieures et supérieures sur le nombre de questions nécessaires pour décider si un graphe contient un triangle. Goldreich et Ron proposent un algorithme pour tester la connexité des graphes de degré borné [GR02]. D'autres compromis entre la taille des structures de données et la complexité des algorithmes, en terme de nombre de bits à tester dans la structure de donnée, ont été prouvés en utilisant divers modèles de compléxité de communication, comme celui de cellule de test (*probe cell model*) [Yao81, MNSW95].

Résultats. Dans cet article, nous proposons une alternative au modèle de calcul frugal de Grumbach et Wu [GW09]. Dans notre modèle, le nombre de rondes de communication est borné, mais à chaque ronde, chaque sommet peut envoyer/recevoir un message de taille $O(\log n)$ bits à/depuis une autorité centrale, appelée *le superviseur*, qui communique avec tous les nœuds du réseau. La notion de superviseur a été utilisée, par exemple, dans le modèle $\mathcal{SIMULTANEOUS\ MESSAGES}$ [BGKL04]. Grâce à notre modèle, il est facile pour le superviseur de reconstruire, i.e., déterminer la matrice d'adjacence, n'importe quel graphe de degré borné, en une ronde. Nous montrons cependant que des questions comme décider si le graphe contient un triangle ou un carré, ou déterminer si le graphe a diamètre au plus 3 sont impossibles à résoudre en une ronde. Étonnament, une ronde suffit pour reconstruire n'importe quel graphe de dégénérescence bornée.

2 Communication frugale en une ronde

Un réseau de communication est représenté par un graphe $G = (V,E)$ simple connexe et non orienté à n sommets. Chaque nœud $x \in V$ possède un identifiant unique $ID(x) \in \{1,\dots,n\}$. En d'autres termes, nous considérons des graphes étiquetés. Dans la séquence de sommets $(v_1,\dots,v_n)$, v_i représente le sommet $x \in V$ dont l'identifiant est $ID(x) = i$. En chaque sommet v, une unité locale de calcul connaît son propre identifiant, l'ensemble $\{ID(y) \mid y \in N_G(v)\}$ des identifiants des voisins de v et le nombre total de sommets (comme d'habitude, on note $N_G(v)$, l'ensemble des voisins de v). Nous considérons le cas où il y a une autorité centrale, le *superviseur*. C'est-à-dire que le réseau d'interconnexion $\mathcal{G}$ consiste en un graphe $G = (V,E)$, avec $V = \{v_1,\dots,v_n\}$, plus un sommet universel v_0 représentant le superviseur, i.e., v_0 est adjacent à tous les sommets de G.

A chaque ronde du processus de communication, tous les nœuds réalisent un calcul local, basé uniquement sur leur propre connaissance locale, et envoient/reçoivent un message à/de chacun de leurs voisins (y compris le superviseur). Le protocole de communication est dit *frugal* si la taille de chaque message est limitée à $O(\log n)$ bits. Nous distinguons la complexité de communication, qui correspond au nombre de rondes du processus de communication, et la complexité locale de calcul qui correspond au maximum du temps pour effectuer chacun des calculs locaux. Après la dernière ronde, le superviseur doit être capable de

répondre à une question initialement posée sur la topologie du graphe G.

Dans cet article, nous étudions les problèmes relatifs au graphe G qui peuvent, ou non, être résolus au travers du réseau d'interconnexion $\mathcal{G}$ par un protocole frugal en une ronde. En d'autres termes, le superviseur, n'ayant aucune connaissance de la topologie du réseau, excepté sa taille, doit résoudre des problèmes en utilisant n messages, chacun de taille $O(\log n)$ bits, reçus des n processeurs. Notez que, puisque nous ne considérons qu'une seule ronde de communication, le réseau peut être asynchrone. En effet, connaissant la taille du réseau, le superviseur peut attendre jusqu'à avoir reçu un message de chacun des sommets.

Un protocole en une ronde consiste donc en une phase locale, où chaque sommet de G calcule un message en fonction de son voisinage, et en une phase globale, où le superviseur calcule le résultat attendu en fonction des messages reçus.

Plus formellement, un *protocole en une ronde* Γ est une famille $(\Gamma^l_n, \Gamma^g_n)_{n\in\mathbb{N}}$, où :

– $\Gamma^l_n : \{1,\dots,n\} \times \mathcal{P}(\{1,\dots,n\}) \to \{0,1\}^*$ est la *fonction locale* de Γ pour des graphes de tailles n,

– $\Gamma^g_n : (\{0,1\}^*)^n \to \{0,1\}^*$ est la *fonction globale* de Γ pour des graphes de tailles n.

Etant donné $G = (V = \{1,\dots,n\}, E)$, le vecteur de messages Γ de G est : $\Gamma^l(G) = (\Gamma^l_n(1, N_G(1)), \dots, \Gamma^l_n(n, N_G(n)))$. Le résultat de Γ pour G est $\Gamma(G) = \Gamma^g_n(\Gamma^l(G))$. Soit $|\Gamma^l(G)| = \max_{1\le i\le n} |\Gamma^l_n(i, N_G(i))|$.

Γ est dit *frugal* si : $\max_{G \text{ graphe de } n \text{ sommets}}(|\Gamma^l(G)|) = O(\log n)$. Notons que, comme habituellement dans le cadre de la complexité de communication, nous ne tenons pas compte de la complexité de Γ^l_n et Γ^g_n. Notons aussi que la fonction Γ^l_n peut être évaluée pour toute paire (i, N) où $i \in \{1,\dots,n\}$ et $N \subseteq \{1,\dots,n\}$. Ainsi, $\Gamma^l_n(i, N)$ correspond au message envoyé au superviseur par le sommet i avec N pour voisinage, dans un graphe à n sommets.

Notons que, définir $\Gamma^l_n(i, N) = N_G(i)$ [‡] $(i \le n)$ permet au superviseur de reconstruire tout graphe G. De plus, dans la classe des graphes de degré borné, une telle fonction est frugale puisque $|N_G(i)| = O(\log n)$ bits.

3 Des questions difficiles

Il apparaît que, étant donné un graphe S, il est souvent difficile de répondre à la question "*Est-ce que le graphe G contient S comme sous-graphe (induit ou non) ?*" en une ronde. Si S n'est pas réduit à une arête, cela s'explique par le fait que les sommets de G ne peuvent deviner *a priori* lesquels de leurs voisins sont susceptibles d'être impliqués dans l'instance de S recherchée : pour un sommet, tous ses voisins se ressemblent ! Pour cette raison, il leur faudrait envoyer leur liste d'adjacence complète ce qui représente des messages trop long de $O(n \log n)$ bits. Plus formellement, nous prouvons que :

Theorem 1 *Il n'existe pas de protocole frugal en une ronde qui permette au superviseur de décider si un graphe : contient un triangle comme sous-graphe, contient un carré (induit ou non), a diamètre au plus* 3.

Pour obtenir nos résultats d'impossibilité, nous utilisons deux ingrédients principaux. Tout d'abord, il est facile de remarquer que, s'il existe un protocole frugal en une ronde permettant au superviseur de reconstruire n'importe quel graphe d'au plus n sommets dans une classe $\mathcal{C}$, alors $|\mathcal{C}| = 2^{O(n \log n)}$ (autrement, au moins 2 instances différentes ne pourraient être distinguées à l'aide de $O(n \log n)$ bits). Dans un second temps, nous introduisons une technique de réduction : si il existe un protocole frugal en une ronde permettant de décider une propriété considérée, alors, il existe un tel protocole pour reconstruire tout graphe d'une famille trop grande, i.e., de taille $2^{\omega(n \log n)}$. Pour la question de décider si un graphe contient un triangle, la famille considérée est celle des graphes (étiquetés) bipartis avec des parties de taille $n/2$, pour décider si un graphe contient un carré, nous utilisons la classe des graphes de n sommets sans carré, et pour la question du diamètre, il s'agit de la classe des graphes à n sommets. Plus de détails sont disponibles ici [BMN+11].

4 Un protocole frugal pour reconstruire des graphes en une ronde

$k \in \mathbb{N}$ étant fixé, nous proposons un protocol frugal en une ronde pour décider si la dégénérescence d'un graphe est au plus k et reconstruire ce graphe, i.e., calculer sa matrice d'adjacence, si c'est le cas. Un graphe est de *dégénérescence* k si il existe un sommet r de degré au plus k dans G tel que $G \setminus \{r\}$ est de

‡. Ici, $N_G(i)$ signifie la concaténation de la representations binaires des identifiants des sommets de $N_G(i)$ plus une chaîne de bits de même taille pour indiquer le premier bit de chacun des identifiants

dégénérescence au plus k. Par exemple, les forêts sont les graphes de dégénérescence 1, les graphes planaires ont dégénérescence au plus 5 et la dégénérescence d'un graphe est bornée par sa largeur arborescente. Notre protocole peut donc être utilisé pour tout graphe de ces classes. Notons que tout graphe de dégénérescence au plus k peut être codé à l'aide de $O(n \log n)$ bits. La question est ici de savoir si cette quantité d'information *locale* est suffisante pour coder un tel graphe de façon à être décodée par le superviseur.

Theorem 2 *Il existe un protocole frugal en une ronde pour décider si un graphe est de dégénérescence $\leq k$ et reconstruire ce graphe si c'est le cas ($k \in \mathbb{N}$ fixé).*

Pour donner une idée de notre protocole, nous commençons par les graphes de dégénérescence $k = 1$: les forêts. Soit T une forêt et notons $N_S(v)$ le voisinage de $v \in V(S)$ dans un sous-graphe S de T. Le protocole fonctionne comme suit. Tout sommet v envoie au superviseur le triplet constitué de son identifiant $ID(v)$, de son degré $\deg_T(v)$ dans T et de la somme des identifiants de ses voisins $\sum_{w \in N_T(v)} ID(w)$, ce qui peut être encodé avec $8 \log n$ bits (formellement, $\Gamma^l_n(v, N_T(v))$ code $(ID(v), |N_T(v)|, \sum_{w \in N_T(v)} ID(w))$). Pour décoder ces messages, le superviseur choisit une feuille v, i.e., un des sommets de degré au plus 1, et "élague" cette feuille de T. Exécutant cette procédure récursivement, le superviseur peut reconstruire T.

Plus précisemment, le troisième élément du triplet envoyé par v contient l'identifiant de son unique voisin w dans T. Le superviseur peut alors modifier le triplet envoyé par w en le remplaçant par $(ID(w), \deg_T(w) - 1, (\sum_{z \in N_T(w)} ID(z)) - ID(v))$ ce qui est exactement $(ID(w), \deg_{T \setminus v}(w), \sum_{z \in N_{T \setminus v}(w)} ID(z))$ le triplet qu'aurait envoyé w dans $T \setminus \{v\}$. En combinant ce nouveau triplet avec ceux envoyés par tous les sommets sauf v, le superviseur peut reconstruire la forêt $T \setminus \{v\}$. Ainsi, par induction sur n, le superviseur peut reconstruire T ou décider si T contient un cycle.

Pour le cas $k \geq 1$, nous généralisons l'idée de "l'élagage" d'un sommet v de degré au plus k du graphe G de telle sorte que la connaissance du graphe $G \setminus v$ peut être obtenue grâce aux informations des sommets dans G en modifiant un peu celles des voisins de v. Plus formellement, le message envoyé au superviseur par tout sommet v est le $k+2$-uplet $\Gamma^l_n(v, N_G(v))$ codant $(ID(v), |N_T(v)|, \{\sum_{w \in N_T(v)} ID(w)^p\}_{p \leq k})$. Nous prouvons qu'à partir de ces n messages de taille $O(\log n)$ bits, le superviseur peut reconstruire la matrice d'adjacence du graphe G ou décider que G a dégénérescence $> k$ [BMN+11]. En particulier, pour tout sommet v de degré $\leq k$, il est possible de déterminer les identifiants de ses voisins à partir de $\{\sum_{w \in N_T(v)} ID(w)^p\}_{p \leq k}$.

5 Perspectives

Nous avons défini un nouveau modèle de communication distribuée et locale, et étudié certaines des questions auxquelles il peut répondre ou non. Les limites de ce modèle ne sont pas encore connues : en particulier, peut-on décider si un graphe est connexe ? D'autre part, les extentions naturelles de ce modèle –quand les messages ne sont pas simultanés, lorsque l'on permet plusieurs rondes– restent à étudier.

Références

[AKKR08] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron. Testing triangle-freeness in general graphs. *SIAM J. Discrete Math.*, 22(2) :786–819, 2008.

[BGKL04] L. Babai, A. Gál, P. G. Kimmel, and S. V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1) :137–166, 2004.

[BMN+11] F. Becker, M. Matamala, N. Nisse, I. Rapaport, K. Suchan, and I. Todinca. Adding a referee to an interconnection network : What can(not) be computed in one round. In *25th IEEE Int. Parallel & Dist. Processing Symp. (IPDPS)*, 2011. http ://arxiv.org/abs/1009.4447.

[Fis01] E. Fischer. The art of uninformed decisions : A primer to property testing. *Bulletin of the EATCS*, 75 :97–126, 2001.

[GR02] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2) :302–343, 2002.

[GW09] S. Grumbach and Z. Wu. Logical locality entails frugal distributed computation over graphs (extended abstract). In *35th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 5911 of *LNCS*, pages 154–165, 2009.

[KMW04] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally ! In *23rd ACM Symp. on Principles of Distributed Computing (PODC)*, pages 300–309, 2004.

[Lin92] N. Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1) :193–201, 1992.

[MNSW95] P. Bro Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC)*, pages 103–111, 1995.

[Pel00] D. Peleg. *Distributed computing : a locality-sensitive approach.* SIAM Monographs on Discrete Maths. and Applications, 2000.

[Yao81] A. Chi-Chih Yao. Should tables be sorted ? *J. ACM*, 28(3) :615–628, 1981.