

Spanner et routage compact : similarités et différences

Cyril Gavoille[†]

LaBRI, Université de Bordeaux, gavoille@labri.fr

Un *spanner* est un sous-graphe H couvrant les sommets d'un graphe G et qui approxime les distances de G . L'*étirement* de H est borné par une fonction s , on parle alors de s -spanner, si $d_H(u, v) \leq s(d_G(u, v))$ pour tous sommets u, v de G . De nombreux travaux concernent l'étude du compromis entre la taille du spanner (c'est-à-dire son nombre d'arêtes) et son étirement. En parallèle, le routage compact s'intéresse à la construction de schémas de routage réalisant un compromis entre la taille des tables de routage et l'étirement de la longueur des routes générées.

Il se trouve que le compromis taille-étirement pour les spanners et pour les schémas de routage coïncide³ pour les étirements *multiplicatifs*, c'est-à-dire lorsque $s(d) = \alpha \cdot d$ pour une constante $\alpha \geq 1$. Des travaux récents montrent qu'il est possible de construire des s -spanners de taille comparables aux précédents mais avec un étirement *additif*, $s(d) = d + \beta$ pour une constante $\beta \geq 0$.

Nous montrons que les résultats concernant les spanners d'étirement additifs ne peuvent pas être étendus au routage compact. Plus précisément nous montrons que tout schéma de routage garantissant pour tout graphe à n sommets des tables de routage de taille en $o(\sqrt{n})$ possède un étirement additif non borné. Cette borne inférieure prouve pour la première fois une séparation entre les deux théories.

Keywords: routage compact, étirement additif, spanner

1 Introduction

L'idée de calculer le « résumé » ou le « squelette » d'un graphe est largement utilisée en informatique dès lors que la taille du graphe devient importante (cf. la notion d'arbre couvrant par exemple). Dans de nombreux problèmes il devient acceptable de supprimer un grand nombre d'arêtes, passant d'une complexité et d'une taille de $O(n^2)$ arêtes à $O(n^{1.5})$ par exemple, si les distances peuvent être modérément distordues, d'un facteur 3 par exemple. Dans le domaine des réseaux, les auteurs de [5] ont montrés comment un squelette de la topologie sous-jacente pouvait être utilisé efficacement pour le stockage des liens actifs des réseaux ad-hoc à la vue de la standardisation IETF du protocole de routage OLSR [2].

Spanners peu denses. Étant donné un graphe non orienté G , un sous-graphe couvrant H est un s -spanner pour G si $d_H(u, v) \leq s(d_G(u, v))$ pour tous sommets u, v de G , où d_X est la fonction de distance du graphe X , les graphes possédant *a priori* une arête-valuation. La fonction s est appelée *étirement* de H , et la *taille* est le nombre d'arêtes de H . Introduit par Peleg *et al.* [7], les spanners sont l'ingrédient clef de plusieurs applications du domaine du calcul distribué, comme les synchroniseurs, le calcul distribué des presque plus courts chemins, les oracles de distances. Les spanners apparaissent aussi dans le domaine des algorithmes d'approximation pour les espaces géométriques, et aussi pour la résolution de système linéaire. À chaque fois, la qualité des spanners (en terme de taille et d'étirement) impacte directement la qualité des solutions des problèmes précités.

La littérature sur les spanners est maintenant très abondante (cf. le survol de Pettie [8]). Il est bien connu que tout graphe à n sommets possède un s -spanner de $O(n^{1+1/k})$ arêtes avec $s(d) = (2k - 1) \cdot d$, pour tout

[†]Supporté par le projet "ALADDIN" de l'ANR et l'équipe-projet "CÉPAGE" de l'INRIA Sud-Ouest.

³ Pour être plus précis c'est le degré moyen des spanners qui coïncide avec la taille des tables.

entier $k > 0$. Une conjecture d'Erdős en théorie des graphes implique, si elle est vraie⁴, que tout s -spanner d'un graphe même non-valué doit posséder $\Omega(n^{1+1/k})$ arêtes dans le pire des cas dès que l'étirement vérifie $s(d) \leq \alpha \cdot d + \beta$ avec $\alpha + \beta < 2k + 1$. Compatibles avec cette conjecture, des s -spanners de $O(n^{1+1/k})$ arêtes avec $s(d) = d + \beta$ pourraient exister à condition⁵ que $\beta \geq 2k - 2$ et que les arêtes ne soient pas valuées. (Dans la suite de la discussion, nous supposons que les graphes ne sont pas valués pour donner pleinement un sens à la notion d'étirement additif). Pour des étirements additifs de la forme $s(d) = d + \beta$, en effet il existe pour tout graphe des s -spanners de $O(n^{1+1/2})$ arêtes pour $\beta = 2$ et même des s -spanners avec $O(n^{1+1/3})$ arêtes pour $\beta = 6$. (Notons que l'on pourrait espérer dans ce dernier cas $\beta = 4$). On ne sait pas si pour d'autres valeurs de β des s -spanners de $O(n^{1+1/3})$ arêtes existent pour tout graphe.

Routage compact. Dans ce problème il s'agit de construire pour chaque graphe G un *schéma de routage* pour G , c'est-à-dire un algorithme de routage indiquant pour chaque sommet u et chaque message arrivant en u vers quel voisin v transmettre le message. Les structures de données utilisées par l'algorithme et associées à chaque sommet sont communément appelées *tables de routage*. Pour une description plus précise du modèle voir par exemple [4].

On mesure la performance d'un schéma par la taille maximale (en bits) des tables de routage qu'il produit, ainsi que l'étirement de la longueur des routes qu'il induit. L'étirement est s si la longueur de la route $\rho(u, v)$ entre toute paire source-destination u, v de G vérifie $\rho(u, v) \leq s(d_G(u, v))$. Intuitivement, plus les tables sont « compactes » plus l'étirement est grand. La littérature sur ce sujet est très abondante et commence dès les débuts d'Internet avec les travaux de Kleinroch et al. [6] à la fin des années 70.

Comme pour le problème des spanners, il existe un compromis entre la taille des tables de routage et l'étirement. Thorup et Zwick [10], puis dans un modèle plus général Abraham *et al.* [1], ont montré que pour tout entier $k > 0$, tout graphe (valués) à n sommets possède un schéma de routage avec des tables⁶ de $\tilde{O}(n^{1/k})$ et un étirement $s(d) = O(k \cdot d)$. L'étirement, pour une taille en $n^{1/k}$, est asymptotiquement optimal, car il a été montré que si $s(d) < (2k + 1) \cdot d$, alors des tables de $\Omega((n \log n)^{1/k})$ bits sont nécessaires dans le pire des cas. Le meilleur étirement multiplicatif que l'on puisse espérer, toujours pour une taille en $n^{1/k}$, est donc⁷ $s(d) = (2k - 1) \cdot d$. Le meilleur schéma connu réalise $s(d) \leq (4k - 5) \cdot d$, ce qui coïncide avec la borne inférieure pour $k = 2$ seulement.

Résultat. Les constructions des schémas de routage sont techniquement plus difficile à mettre au point que celles de spanners, la partie « structures de données » par exemple étant absente de ces dernières. Plusieurs auteurs [9, 3] pensent ainsi qu'avec un effort particulier, les techniques développées pour les spanners d'étirement additif pourront être adaptées au routage compact. En particulier, nous avons vu que des s -spanners de $O(n^{1+1/3})$ arêtes avec $s(d) \leq d + 6$ existent. On peut naturellement se demander si un schéma de routage avec des tables de taille $\tilde{O}(n^{1/3})$ et un étirement additif $s(d) \leq d + \beta$ n'existe pas aussi pour une certaine constante β éventuellement proche de 6.

Nous allons voir que ce rêve est vain. Il existe des graphes ayant des spanners peu denses et un étirement additif, où pourtant il n'est pas possible de router avec des tables compactes et un étirement additif. Cette propriété combinatoire, l'existence de spanner peu dense, et l'aptitude au routage compact sont bien deux caractéristiques différentes des graphes.

Plus précisément, nous montrons que, si pour tout graphe à n sommets on peut construire un schéma de routage utilisant des tables d'au plus $\mu = \mu(n)$ bits, alors un de ces schémas produit pour au moins un graphe un étirement additif d'au moins $\Omega(n^{1/3}/\mu^{2/3})$. En particulier, pour des tables de routage de taille bornée par $\mu = o(\sqrt{n})$, l'étirement additif ne peut pas être borné par une constante. Plus concrètement, nous pouvons montrer que le schéma de routage de Thorup et Zwick [10] avec $k = 3$, qui produit des tables de taille $\tilde{O}(n^{1/3})$ pour un étirement $s(d) = (4k - 5) \cdot d = 7d$, possède inévitablement un étirement additif d'au moins $n^{\Omega(1)}$.

⁴ Elle est cependant déjà prouvée pour $k = 1, 2, 3, 5$.

⁵ Puisque si $\beta < 2k - 2$, alors $1 + \beta < 2k - 1$ et $\Omega(n^{1+1/(k-1)})$ arêtes sont requises.

⁶ La notation $\tilde{O}(f(n))$ est un raccourci pour $f(n)(\log n)^{O(1)}$.

⁷ Puisqu'un étirement $s(d) < (2k - 1) \cdot d$ impliquera des tables de $\Omega((n \log n)^{1/(k-1)})$ bits.

2 Un résultat d'impossibilité

Afin d'énoncer correctement notre résultat, nous devons préciser un certain nombre de points techniques. Un schéma de routage est dit *étiqueté* si les adresses utilisées dans le routage ont été optimisées en fonction de la topologie du réseau, par le concepteur du schéma, comme par exemple les coordonnées (x, y) des nœuds d'une grille le seraient pour un réseaux maillé 2D. Le schéma est dit de routage avec *indépendance des noms* si au contraire le routage d'une source u à une destination v se fait sur la base du nom original de v . En quelque sorte, le nom est décorrélé de la topologie. Il peut correspondre par exemple à la clef (le *hash*) d'un contenu que l'on souhaite atteindre dans un réseaux de partage de fichiers. De manière plus pessimiste, ce nom peut être choisi par un adversaire voulant compliquer la tâche du concepteur du schéma. Il va sans dire que les tables des schémas de routage avec indépendance des noms sont bien moins compactes que celles des schémas étiquetés, ou qu'à taille égale l'étirement est plus grand. Pour le cas étiqueté il convient de tenir compte de la longueur des adresses, puisque sinon n'importe quelle information pourrait être codée dans les adresses.

Pour tout schéma de routage R sur un graphe G , et tout sommet x de G , on note R_x l'algorithme⁸ de routage de x , qui à tout en-tête et port d'entrée de x fait correspondre un port de sortie de x et éventuellement d'un nouvel en-tête. La *taille* de R_x est définie comme la longueur (en bits) du plus petit programme implantant R_x . L'*espace mémoire* du schéma R est défini comme le maximum, sur tous les sommets x de G , de la taille de R_x . Notez bien que l'espace mémoire de R ainsi défini représente la quantité d'information nécessaire au schéma R (la taille des tables de routage), et pas la quantité de mémoire nécessaire pour évaluer R_x par exemple.

On appelle *stratégie de routage* une fonction qui à tout graphe G d'une famille donnée construit un schéma de routage pour G . En quelque sorte, une stratégie modélise l'action du concepteur d'un schéma de routage (qui détermine l'algorithme de routage, le format des en-têtes, éventuellement celui des adresses, etc.).

Comme il s'agit de donner une borne inférieure, nous supposons le modèle le plus fort qu'il soit, à savoir, les schémas de routage étiquetés où la taille des en-têtes n'est pas limitée⁹ et où les ports des arêtes sortantes de chaque sommet u peuvent être fixés librement¹⁰ par la stratégie de routage.

Théorème 1 *Toute stratégie de routage sur les graphes non-valués à n sommets construisant des schémas de routage étiquetés d'espace mémoire et d'adressage d'au plus μ bits génère pour au moins un graphe un schéma d'étirement additif $\Omega(n^{1/3}/\mu^{2/3})$.*

Le reste de cette section est consacrée au sketch de la preuve de ce résultat.

On considère une famille de graphes dépendant de trois paramètres entiers μ , p and δ , où μ est la borne sur l'espace mémoire définie par l'énoncé du théorème. Plus tard on fixera $\delta = \Theta(p/\mu)$. Chaque graphe G de la famille est entièrement déterminé par une matrice booléenne $p \times p$, disons M pour le graphe G . Le graphe $G = G(M)$ est composé des ensembles de sommets $S = \{s_1, \dots, s_p\}$, $T = \{t_1, \dots, t_p\}$, $A = \{a_1, \dots, a_p\}$, $B = \{b_1, \dots, b_p\}$ et d'un ensemble de sommets de degré deux. Pour tout i , le sommet s_i est connecté à a_i et à b_i par un chemin de longueur δ donc comprenant $\delta - 1$ sommets intermédiaires de degré deux. De plus, pour chaque couple (i, j) , un chemin de longueur δ connecte t_j à a_i si $M[i, j] = 1$ et à b_i si $M[i, j] = 0$ (voir l'exemple ci-après où $p = 3$ et $\delta = 1$).

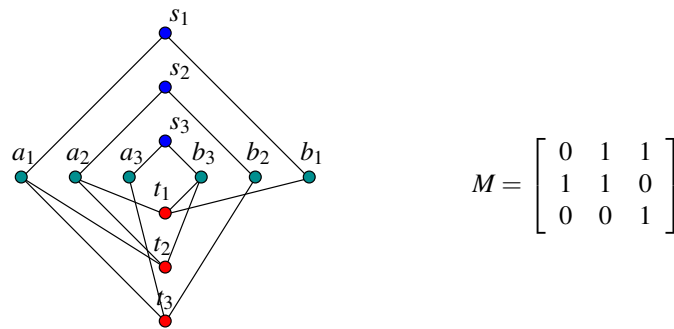
Le nombre de sommets de G est $n = \Theta(\delta p^2)$. Soit R le schéma de routage sur G produit par la stratégie de routage. On note β l'étirement additif du schéma R .

La remarque importante est que dans G , tout chemin (pas forcément simple) de s_i à t_j de longueur $< 4\delta$ contient a_i si et seulement si $M[i, j] = 1$. D'un autre côté, si $\beta < 2\delta$, la longueur de la route de s_i à t_j induite par R vérifie: $\rho_R(s_i, t_j) \leq d_G(s_i, t_j) + \beta < 4\delta$ car $d_G(s_i, t_j) = 2\delta$. Donc, si $\beta < 2\delta$, la remarque précédente s'applique non seulement aux chemins, mais aux routes du schéma R de s_i à t_j .

⁸ Incluant toutes les constantes et tables auxiliaires nécessaires.

⁹ Notez bien que lorsqu'une source u envoie un message à destination de v , le premier en-tête doit pouvoir être produit en u seulement à partir de l'adresse de la destination v et des tables stockées en u .

¹⁰ Cependant ils doivent être représentés par un entier unique de $\{1, \dots, \deg(u)\}$.



Le point technique consiste à montrer que $\delta\mu p = \Omega(K(M))$ où $K(M)$ est la longueur du plus petit programme¹¹ qui renvoie M . Puis on fixe M comme une matrice telle que $K(M) \geq p^2 - O(1)$, matrice qui existe nécessairement par argument de comptage¹². On a donc établi que, si $\beta < 2\delta$, alors $\delta\mu p > cp^2$ pour une constante $c > 0$ suffisamment petite. De manière équivalente : si $\delta\mu \leq cp$, alors $\beta \geq 2\delta$. En prenant $\delta = \lfloor cp/\mu \rfloor$, on a donc montré que l'étirement de R est $\beta \geq 2\delta = \Omega(p/\mu)$. On conclut en notant que $n = \Theta(\delta p^2) = \Theta(p^3/\mu)$, et donc l'étirement est $\beta = \Omega(p/\mu) = \Omega(n^{1/3}/\mu^{2/3})$.

On remarque que pour $\delta > 1$, les graphes de la famille considérée ont au plus $2n$ arêtes, et donc des spanners avec un compris taille-étirement évidemment très bon. Cependant, par le théorème précédent, un tel compromis pour le routage n'est pas possible.

References

- [1] I. ABRAHAM, C. GAVOILLE, AND D. MALKHI, *On space-stretch trade-offs: Upper bounds*, in 18th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), ACM Press, July 2006, pp. 207–216. 2
- [2] C. ADJIH, A. LAOUI, P. MUHLETHALER, A. QAYYUM, AND L. VIENNOT, *The optimised routing protocol for mobile ad-hoc networks: protocol specification*, Tech. Rep. 5145, INRIA, Mar. 2003. 1
- [3] M. ELKIN AND J. ZHANG, *Efficient algorithms for constructing $(1 + \epsilon, \beta)$ -spanners in the distributed and streaming models*, Distributed Computing, 18 (2006), pp. 375–385. 2
- [4] P. FRAIGNIAUD AND C. GAVOILLE, *Routing in trees*, in 28th International Colloquium on Automata, Languages and Programming (ICALP), F. Orejas, P. G. Spirakis, and J. v. Leeuwen, eds., vol. 2076 of Lecture Notes in Computer Science, Springer, July 2001, pp. 757–772. 2
- [5] P. JACQUET AND L. VIENNOT, *Remote spanners: what to know beyond neighbors*, in 23rd IEEE International Parallel & Distributed Processing Symposium (IPDPS), IEEE Computer Society Press, May 2009. 1
- [6] L. KLEINROCK AND F. KAMOUN, *Hierarchical routing for large networks; performance evaluation and optimization*, Computer Networks, 1 (1977), pp. 155–174. 2
- [7] D. PELEG AND J. D. ULLMAN, *An optimal synchronizer for the hypercube*, in 6th Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM Press, Aug. 1987, pp. 77–85. 1
- [8] S. PETTIE, *Low distortion spanners*, in 34th International Colloquium on Automata, Languages and Programming (ICALP), vol. 4596 of Lecture Notes in Computer Science, Springer, July 2007, pp. 78–89. 1
- [9] ———, *Distributed algorithms for ultrasparse spanners and linear size skeletons*, in 27th Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM Press, Aug. 2008, pp. 253–262. 2
- [10] M. THORUP AND U. ZWICK, *Compact routing schemes*, in 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), ACM Press, July 2001, pp. 1–10. 2

¹¹ En résumé, il faut déterminer un programme P de longueur $|P| = O(\delta\mu p)$ permettant de construire M , et donc $|P| \geq K(M)$ par définition. Ceci peut être fait sur la base des algorithmes de routage R_x (plus précisément des programmes de longueur $\leq \mu$ implantant ces algorithmes) de tous les sommets des chemins $s_i - a_i$ et $s_i - b_i$, ainsi que les adresses (de longueur $\leq \mu$) des sommets de T et d'un certain nombre de numéros de ports. Le programme P simule alors la route de s_i à t_j sur sa première partie, et teste le fait que la route induite par R utilise ou non a_i . Comme $\beta < 2\delta$, la propriété s'applique et l'on peut montrer que P construit M .

¹² Il faut compter le nombre de programmes de longueur au plus t bits, nombre qui doit être supérieur au nombre de matrices booléennes $p \times p$.