

# Schéma d'adjacence compact pour les arbres de degré borné et applications

Nicolas Bonichon\*    Cyril Gavoille\*    Arnaud Labourel\*

15 septembre 2006

## Résumé

Nous proposons une représentation implicite ou schéma d'étiquetage d'adjacence compact pour les arbres de degré borné par  $d$ . Notre schéma, nommé *Parcours bondissant*, assigne à chaque sommet d'un tel arbre de  $n$  sommets des étiquettes binaires distinctes de taille  $\log_2 n + O(\log_2 d)$  bits telles que l'on puisse calculer l'adjacence entre 2 sommets en utilisant seulement leur étiquette. Nous utilisons ce résultat pour améliorer les bornes connues pour la représentation implicite des graphes planaires extérieurs (respectivement des graphes planaires) obtenant ainsi  $2 \log_2 n$  (respectivement  $3 \log_2 n$ ) comme borne pour la taille des étiquettes.

## 1 Introduction

Les deux manières les plus connues de représenter un graphe sont les matrices d'adjacence et les listes d'adjacence. Cette dernière représentation est efficace en terme d'information à stocker pour les graphes peu denses, mais les requêtes d'adjacence requiert une recherche dans la liste. A l'opposé, les matrices d'adjacence permettent des requêtes en temps constant au prix d'une information à stocker de taille super-linéaire.

Une autre technique appelée *representation implicite* ou *schéma d'étiquetage d'adjacence*, consiste à assigner une étiquette à chaque sommet d'un graphe telle que l'on puisse calculer les requêtes d'adjacences entre deux sommets avec pour seule entrée les deux étiquettes des sommets. Le but est minimiser la taille maximum des étiquettes associées aux sommets ainsi que de minimiser le temps d'exécution de la fonction d'adjacence.

Les schémas d'étiquetage d'adjacence, introduits par [Bre66, BF67], ont été étudiés par [KNR88, KNR92]. Ils ont donné des schémas d'étiquetage d'adjacence pour de nombreuses familles de graphes avec des étiquettes de tailles

---

\*Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, Bordeaux, France. {bonichon,gavoille,labourel}@labri.fr. Les trois auteurs sont financés par le projet "GeoComp" de l'ACI Masse de Données.

$O(\log n)$  bits. En particulier, pour les arbres leur schéma consiste à : 1) pré-étiqueter les sommets de l'arbres avec des étiquettes distinctes choisies de manière arbitraire et servant d'identifiant ; 2) choisir une racine ; et 3) donner aux sommets comme étiquette la paire composée de son identifiant et de l'identifiant de son père. Le test d'adjacence vérifie si l'identifiant d'un sommet est égal à l'identifiant du père de l'autre sommet. De telles étiquettes sont de taille  $2 \lceil \log n \rceil$  bits<sup>1</sup>, alors que  $\lceil \log n \rceil$  sont clairement nécessaires puisque les étiquettes doivent être distinctes.

Améliorer la borne supérieure sur la taille des étiquettes obtenu grâce à ce schéma simple n'est pas une tache facile. Cette borne a été améliorée de manière non-triviale par [AKM01] à  $1.5 \log n + O(\log \log n)$  bits, et plus récemment à  $\log n + O(\log^* n)$  bits<sup>2</sup> [AR02] laissant ouverte la question de savoir si on peut améliorer cette borne à  $\log n + O(1)$  bits qui est la meilleure borne inférieure connue.

Une importante observation de [KNR88] est que, chaque famille  $\mathcal{F}$  de graphes d'au plus  $n$  sommets supportant un schéma d'étiquetage avec des étiquettes de taille au plus  $c \log n$  bits, a un graphe  $\mathcal{U}_n(\mathcal{F})$  d'au plus  $n^c$  sommets tel que chaque graphe de  $\mathcal{F}$  avec au plus  $n$  sommets est un sous-graphe induit de  $\mathcal{U}_n(\mathcal{F})$ . Le graphe  $\mathcal{U}_n(\mathcal{F})$  est appelé graphe *universel-induit* pour  $\mathcal{F}$ .

Motivées par des application dans les moteurs de recherche XML, ainsi que dans les applications distribuées telles que les réseaux pair-à-pair ou le routage dans des réseaux, d'autres structures de données distribuées de taille d'étiquettes de  $O(\log n)$ -bit ont été développées.

Par exemple, le routage dans les arbres [FG01, TZ01], le routage par plus court chemin dans des réseaux spécifiques [BG05, DL02, DL04], des requêtes de distances pour les graphes d'intervalles, d'arc circulaires et de permutations [BG05, GP03a] ont des structures de données distribuées de taille  $O(\log n)$  bits. Plus spécifiquement pour les requêtes sur les arbres, il existe : le plus petit ancêtre commun [AGKR04] avec des étiquette de taille  $O(\log n)$ , l'ascendance [AAK<sup>+</sup>05] avec  $\log n + O(\sqrt{\log n})$  bits, et des requêtes sur des sommets proches avec  $\log n + \Theta(\log \log n)$  bits [KM01, ABR05]. Un "survey" sur les schémas d'étiquetage peut être trouvé dans [GP03b]. Tous ces schémas donnent des étiquetages de taille  $\log n + \omega(1)$ <sup>3</sup>.

À notre connaissance, pour des familles de graphe raisonnablement grande, il n'existe pas de structure de donnée distribuée optimale pour la taille des étiquettes à un facteur constant près. En particulier, pour les requêtes d'adjacence dans les arbres, la meilleure borne inférieure connue est  $\log n$  alors que la borne supérieure est  $\log n + O(\log^* n)$  [AR02]. Ce schéma, basé sur une décomposition récursive en  $\Theta(\log^* n)$  niveaux, a un temps de calcul pour les requêtes d'adjacence de  $\Omega(\log^* n)$ .

<sup>1</sup>Tous les logarithmes sont en base deux.

<sup>2</sup> $\log^* n$  dénote le nombre de fois le log doit être itéré afin d'obtenir une constante.

<sup>3</sup> $f(n) = \omega(g(n))$  si et seulement si  $g(n) = o(f(n))$ .

## 2 Description générale de la technique

De manière générale, la technique de parcours bondissant consiste à :

1. Sélectionner un parcours approprié pour le graphe ;
2. Associer à chaque sommet  $x$  une information  $C(x)$  ;
3. Réaliser le parcours du graphe et assigner des étiquettes aux sommets de manière croissante dans l'ordre du parcours du graphe mais pas forcément consécutives.

Intuitivement, le test d'adjacence entre  $x$  et  $y$  est fait sur la base de  $C(x)$  et  $C(y)$ . En fait, les sauts réalisés en étape 3 sont faits en sélectionnant un intervalle associé à chaque sommet dans lequel chaque étiquette doit être. Il est important de noter que les intervalles sont ordonnés dans le même ordre que les sommets correspondants dans le parcours. De plus, tous ces intervalles doivent être disjoints. La position de l'étiquette de  $x$  dans son intervalle est choisie de manière à ce que l'on puisse encoder  $C(x)$  dans l'étiquette et ensuite pouvoir extraire cette information. En général, l'information  $C(x)$  détermine un certain nombre d'intervalles contenant les voisins de  $x$  le suivant dans le parcours et seulement ceux-ci.

La principale difficulté de cette technique est de choisir l'information  $C(x)$  à coder et surtout la manière de la coder. En effet, généralement, on ne pourra pas coder les distances exactes qui déterminent les intervalles contenant les voisins, et en fait on code une approximation qui est supérieure à la distance initiale. Ensuite, pour que le schéma soit valide, on décale tous les étiquettes des sommets suivants pour que cette distance devienne la véritable distance. Intuitivement, il est clair que plus on approxime la distance, plus on doit décaler les étiquettes des sommets suivants et inversement.

Cette technique diffère fondamentalement des schémas précédents, dans lesquels une étiquette est essentiellement vue comme un identifiant de  $\lceil \log n \rceil$  bits plus des informations supplémentaires, obtenant ainsi inévitablement des étiquettes de taille  $\log n + \omega(1)$  bits.

## 3 Arbres de degré borné et applications

En utilisant le parcours bondissant, on peut obtenir un schéma pour les arbres de degré borné dont la taille des étiquettes dépend de  $d$  et qui est optimal à un facteur constant près si  $d$  est constant. Il est important de noter que le schéma de [AR02] pour les arbres quelconques ne se simplifie pour les arbres de degré borné.

**Théorème 1** *La famille des arbres d'au plus  $n$  sommets et de degré borné par  $d$  admet un schéma d'étiquetage d'adjacence avec des étiquettes de taille au plus  $\log n + O(1)$  bits. De plus, le test d'adjacence se fait en temps constant et l'étiquetage du graphe en temps  $O(n)$ .*

D'après [KNR88], cela implique que les arbres de degré borné par  $d$  ont un graphe universel induit de taille  $O(d^{O(1)}n)$ . De plus, en utilisant la

décomposition en arêtes de [Gon] pour les graphes planaires en trois arbres dont un est de degré borné par quatre, on peut améliorer la meilleure borne connue pour les graphes planaires.

**Théorème 2** *La famille des graphes planaires d'au plus  $n$  sommets admet un schéma d'étiquetage d'adjacence avec des étiquettes de taille au plus  $3 \log n + O(1)$  bits. De plus, le test d'adjacence se fait en temps constant et l'étiquetage du graphe en temps  $O(n)$ .*

De la même façon on peut améliorer la borne pour les graphes planaires extérieurs en utilisant le fait qu'on peut partitionner leurs arêtes en deux arbres dont un est de degré borné par trois.

**Théorème 3** *La famille des graphes planaires extérieurs d'au plus  $n$  sommets admet un schéma d'étiquetage d'adjacence avec des étiquettes de taille au plus  $2 \log n + O(1)$  bits. De plus, le test d'adjacence se fait en temps constant et l'étiquetage du graphe en temps  $O(n)$ .*

## 4 Conclusion

La conjecture de représentation implicite de graphe de [KNR88, KNR92], toujours irrésolue à ce jour pose le problème de savoir si chaque famille héréditaire<sup>4</sup> avec  $2^{O(n \log n)}$  graphes étiquetés de  $n$  sommets admet un schéma d'étiquetage d'adjacence avec des étiquettes de taille  $O(\log n)$ -bit. Cette conjecture est motivée par le fait que chaque famille avec au moins  $2^{cn \log n}$  graphes étiquetés de  $n$  sommets requiert des étiquettes de taille au moins  $c \log n$  bits.

Notre schéma suggère que pour les arbres, des étiquettes de taille  $\log n + O(1)$  bits sont peut-être possible. Nous proposons donc de prouver ou de réfuter la proposition suivante :

*Chaque famille héréditaire de graphe avec au plus  $n!2^{O(n)} = 2^{n \log n + O(n)}$  graphes étiquetés admet un schéma d'étiquetage d'adjacence avec des étiquettes de taille  $\log n + O(1)$  bits.*

De nombreuses familles de graphes sont concernées par cette proposition : arbres, graphes planaires, graphes d'arborescence bornée, graphes de genre borné, graphe excluant un mineur fixé (cf. [NRTW05] pour l'énumération de ces graphes). Prouver cette dernière conjecture semble difficile. En effet, la meilleure borne connue pour les planaires est seulement de  $3 \log n + O(\log^* n)$  qui est relativement éloignée de la borne de  $\log n + O(1)$  proposée par la conjecture. Inversement, réfuter cette conjecture est difficile car une simple énumération ne suffit pas.

---

<sup>4</sup>Les familles héréditaires sont les famille de graphes close en prenant un sous-graphe.

## Références

- [AAK<sup>+</sup>05] Serge Abiteboul, Stephen Alstrup, Haim Kaplan, Tova Milo, and Theis Rauhe. Compact labeling schemes for ancestor queries. *SIAM Journal on Computing*, 2005.
- [ABR05] Stephen Alstrup, Philip Bille, and Theis Rauhe. Labeling schemes for small distances in trees. *SIAM Journal on Discrete Mathematics*, 19(2) :448–462, 2005.
- [AGKR04] Stephen Alstrup, Cyril Gavoille, Haim Kaplan, and Theis Rauhe. Nearest common ancestors : A survey and a new algorithm for a distributed environment. *Theory of Computing Systems*, 37 :441–456, 2004.
- [AKM01] Serge Abiteboul, Haim Kaplan, and Tova Milo. Compact labeling schemes for ancestor queries. In *12<sup>th</sup> Symposium on Discrete Algorithms (SODA)*, pages 547–556. ACM-SIAM, January 2001.
- [AR02] Stephen Alstrup and Theis Rauhe. Small induced-universal graphs and compact implicit graph representations. In *43<sup>rd</sup> Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 53–62. IEEE Computer Society Press, November 2002.
- [BF67] Melvin A. Breuer and Jon Folkman. An unexpected result on coding the vertices of a graph. *Journal of Mathematical Analysis and Applications*, 20 :583–600, 1967.
- [BG05] Fabrice Bazzaro and Cyril Gavoille. Localized and compact data-structure for comparability graphs. In *16<sup>th</sup> Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3827 of Lecture Notes in Computer Science, pages 1122–1131. Springer, December 2005.
- [Bre66] Melvin A. Breuer. Coding the vertexes of a graph. *IEEE Transactions on Information Theory*, IT-12 :148–153, 1966.
- [DL02] Feodor F. Dragan and Irina Lomonosov. New routing schemes for interval graphs, circular-arc graphs, and permutation graphs. In *14<sup>th</sup> IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, pages 78–83, November 2002.
- [DL04] Feodor F. Dragan and Irina Lomonosov. On compact and efficient routing in certain graph classes. In *15<sup>th</sup> Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3341 of Lecture Notes in Computer Science, pages 402–414. Springer, December 2004.
- [FG01] Pierre Fraigniaud and Cyril Gavoille. Routing in trees. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *28<sup>th</sup> International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2076 of Lecture Notes in Computer Science, pages 757–772. Springer, July 2001.

- [Gon] D. Gonalves. Covering planar graphs with 3 forests, one being of maximum degree 4. submitted.
- [GP03a] Cyril Gavoille and Christophe Paul. Optimal distance labeling schemes for interval and circular-arc graphs. In G. Di Battista and U. Zwick, editors, *11<sup>th</sup> Annual European Symposium on Algorithms (ESA)*, volume 2832 of Lecture Notes in Computer Science, pages 254–265. Springer, September 2003.
- [GP03b] Cyril Gavoille and David Peleg. Compact and localized distributed data structures. *Journal of Distributed Computing*, 16 :111–120, May 2003. PODC 20-Year Special Issue.
- [KM01] Haim Kaplan and Tova Milo. Short and simple labels for small distances and other functions. In *7<sup>th</sup> International Workshop on Algorithms and Data Structures (WADS)*, volume 2125 of Lecture Notes in Computer Science, pages 32–40. Springer, August 2001.
- [KNR88] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. In *20<sup>th</sup> Annual ACM Symposium on Theory of Computing (STOC)*, pages 334–343. ACM Press, May 1988.
- [KNR92] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM Journal on Discrete Mathematics*, 5 :596–603, 1992.
- [NRTW05] Serguei Norine, Neil Robertson, Robin Thomas, and Paul Wollan. Proper minor-closed families are small. *Journal of Combinatorial Theory, Series B*, 2005. To appear.
- [TZ01] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *13<sup>th</sup> Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–10. ACM Press, July 2001.