Un algorithme polynomial pour reconnaître les graphes parfaits

M. Chudnovsky[1], G. Cornuéjols[2], X. Liu[2], P. Seymour[1] et K. Vušković[3]

1. Mathematics department, Princeton University, Princeton, NJ, USA
(mchudnov,pds)@Math.Princeton.EDU

2. GSIA, Carnegie Mellon University, Pittsburgh, PA 15213, USA
(gc0v,xinming)@andrew.cmu.edu

3. School of Computing, University of Leeds, Leeds LS2 9JT, Angleterre
vuskovi@comp.leeds.ac.uk

Mots-clefs : graphe parfait, algorithme de reconnaissance, trou impair.

1 Introduction

Récemment, deux algorithmes polynomiaux ont été découverts pour reconnaître les graphes parfaits. Les deux algorithmes utilisent la même première phase, dite de nettoyage. Cette phase fait l'objet d'un article co-signé par Chudnovsky, Cornuéjols, Liu, Seymour et Vušković [1]. La deuxième phase se fait de façons très différentes dans deux articles, l'un co-signé par Chudnovsky et Seymour [3] et l'autre par Cornuéjols, Liu et Vušković [6]. L'algorithme de Chudnovsky et Seymour recherche directement un trou impair dans un graphe nettoyé. L'algorithme de Cornuéjols, Liu et Vušković utilise un théorème de décomposition de Conforti, Cornuéjols et Vušković [4] pour les graphes sans trou impair.

2 Idée du Nettoyage

Un graphe G est parfait si, pour tout sous-graphe induit H de G, le nombre chromatique de H est égal à la taille d'une clique maximum dans H. Un *trou* est un cycle de longueur au moins quatre. Un trou est *impair* s'il a un nombre impair d'arêtes. Sinon, il est pair. Le complement du graphe G est dénoté par $\bar{G}$. Récemment, Chudnovsky, Robertson, Seymour et Thomas [2] ont démontré qu'un graphe G est parfait si et seulement si ni G ni $\bar{G}$ ne contient un trou impair. C'était la célèbre conjecture forte des graphes parfaits proposée par Berge en 1960. Par conséquent, pour reconnaître si un graphe G est parfait, il suffit de reconnaître si G ou $\bar{G}$ contient un trou impair.

Nous ne savons pas reconnaître en temps polynomial si un graphe G général contient un trou impair, mais nous savons le faire pour une classe spéciale de graphes que nous appelons graphes "nettoyés". Si H est un trou impair dans le graphe G, nous dirons que le sommet $u \in V(G) \setminus V(H)$ est *majeur* pour H si aucun chemin de longueur deux de H ne contient tous les voisins de u dans H. Nous dirons que le trou H est *nettoyé* si G ne contient aucun sommet majeur pour H. Un graphe est "nettoyé" s'il ne contient pas de trou impair ou si tous les trous impairs d'une certaine famille (définie de façon précise dans les articles [1] et [6]) sont nettoyés. Notre approche pour reconnaître si un graphe G est parfait consiste en deux étapes :

(i) construire en temps polynomial un graphe nettoyé G' qui contient un trou impair si et seulement si G en contient un ou, dans certains cas, terminer avec la conclusion que G est imparfait, et

(ii) verifier en temps polynomial si le graphe G' contient un trou impair.

L'étape (i) est résolue dans [1], où l'on démontre le théorème suivant. Soit H un trou impair dans un graphe G. Un ensemble $X \in V(G)$ est un ensemble *nettoyant* pour H s'il contient tous les somments majeurs de H et au plus trois sommets de H.

Théorème 1

Il y a un algorithme de complexité $O(|V(G)|^6$ qui, étant donné un graphe G, ou bien détermine que G est imparfait, ou bien trouve une famille $\mathcal{X}$ de $O(|V(G)|^5$ sous-ensembles de $V(G)$ tels que, si H est un plus petit trou impair dans G, au moins un des $X \in \mathcal{X}$ est un ensemble nettoyant pour H.

L'étape (i) se résoud en utilisant le Théorème 1 : pour tout ensemble $X \in \mathcal{X}$, on construit $O(|V(G)|^3$ sous-graphes de G en enlevant tous les sommets de X sauf trois. Le graphe G' est l'union de ces $O(|V(G)|^8$ sous-graphes de G. Dans la section suivante, nous présentons l'idée de l'algorithme décrit dans [6] pour l'étape (ii). Cet algorithme est basé sur un théorème de décomposition [4], contrairement à l'algorithme de Chudnovsky et Seymour [3].

3 Théorème de Décomposition

Une approche possible pour l'étape (ii) est de décomposer le graphe G donné en un nombre polynomial de graphes plus simples, $G_1, \ldots, G_m$ de telle façon que les deux propriétés suivantes soient satisfaites :

(1) G est un graphe sans trou impair si et seulement si les graphes G_i sont sans trou impair pour tout $i = 1, \ldots, m$, et

(2) pour tout $i = 1, \ldots, m$, on peut vérifier directement, en temps polynomial, si le graphe G_i contient un trou impair.

Les graphes nettoyés sont très utile pour satisfaire (1). L'algorithme de reconnaissance se base sur le théorème suivant de Conforti, Cornuéjols et Vušković [4] pour la décomposition des graphes sans trou impair.

Un ensemble S de sommets est une *étoile double* si S contient deux sommets adjacents u et v tels que tout sommet de S est adjacent à u ou v. Un ensemble de sommets S est un ensemble d'articulation de G si $G \setminus S$ contient plus de composantes connexes que G.

Un graphe G a un 2-join si ses sommets peuvent être partitionnés en ensembles V_1 et V_2 tels que, pour $i = 1, 2$, V_i contient des ensembles disjoints non vides de sommets A_i et B_i tels que tous les sommets de A_1 sont adjacents à tous les sommets de A_2, tous les sommets de B_1 sont adjacents à tous les sommets de B_2, et il n'y a pas d'autres adjacences entre V_1 et V_2. De plus, pour $i = 1, 2$, $|V_i| > 2$ et le graphe induit par V_i n'est pas un chemin.

Un graphe *de base* est un graphe biparti ou le graphe ligne d'un graphe biparti ou le complément d'un graphe ligne d'un graphe biparti.

Théorème 2 [4]

Si G est un graphe sans trou impair alors, ou bien G est un graphe de base, ou bien G a une étoile double d'articulation ou un 2-join.

On peut facilement vérifier en temps polynomial si un graphe est un graphe de base. La propriété (2) mentionée plus haut est donc satisfaite. On peut trouver un 2-join en temps polynomial

[5]. Il n'est pas difficile de trouver une étoile double d'articulation en temps polynomial : Soient u, v deux sommets adjacents et x, y deux sommets nonadjacents. On peut vérifier s'il existe une étoile double d'articulation centrée à u et v qui disconnecte x et y en enlevant tous les voisins de u et tous les voisins de v à l'exception de x et y. Par conséquent, la seule difficulté dans l'application du Théorème 2 est de décomposer un graphe G qui a une étoile double d'articulation ou un 2-join en graphes $G_1, \ldots, G_m$ qui satisfassent la propriété (2) mentionnée plus haut. Lorsque G a un 2-join, cela se fait assez aisément, mais lorsque G a une étoile double d'articulation, c'est nettement plus compliqué et c'est là que l'on a besoin que le graphe G soit nettoyé [6].

Références

[1] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour et K. Vušković (2003). Cleaning for Bergeness. Submitted to *Combinatorica*.

[2] M. Chudnovsky, N. Robertson, P. Seymour et R. Thomas (2003). The Strong Perfect Graph Theorem. Submitted to *Annals of Mathematics*.

[3] M. Chudnovsky et P. Seymour (2003). Recognizing Berge Graphs. Submitted to *Combinatorica*.

[4] M. Conforti, G. Cornuéjols et K. Vušković (2003). Decomposition of Odd-hole-free Graphs by Double Star Cutsets and 2-joins, à paraitre dans le numéro spécial de *Discrete Applied Mathematics* dédié au Symposium Brésilien sur les Graphes, Algorithmes et Combinatoire, Fortaleza, 17-19 Mars 2001.

[5] G. Cornuéjols et W.H. Cunningham, Compositions for Perfect Graphs, *Discrete Mathematics 55* (1985) 245-254.

[6] G. Cornuéjols, X. Liu et K. Vušković (2003). A Polynomial Algorithm for Recognizing Perfect Graphs. Submitted to *Combinatorica*.