Coloriage de graphe en programmation par contraintes

Nicolas Barnier, Pascal Brisset

Coloriage de graphe en programmation par contraintes

N. Barnier, P. Brisset

LOG-École Nationale de l'Aviation Civile
7, avenue Édouard Belin, 31 055 *Toulouse cedex 4*
(barnier,brisset)@recherche.enac.fr

Mots-clefs : coloriage de graphe, clique, programmation par contraintes

1 Introduction

Le coloriage optimal d'un graphe (GC) est un problème NP-difficile très classique et un sujet de recherche toujours actif [6], avec de nombreux domaines d'application : scheduling, allocation de ressources, reconnaissance de formes... Comme c'est un problème (NP-)difficile, les techniques approchées figurent en bonne place parmi les algorithmes les plus performants sur les graphes « aléatoires » peu structurés et de grande taille (algorithme glouton [4], recherche locale [5]). Cependant, les algorithmes énumératifs complets de Branch & Bound (B&B) peuvent également être efficaces [2] pour des problèmes plus structurés, par exemple issus d'applications « réelles », ou de taille moindre. Les deux méthodes principales utilisées pour le GC soit déterminent des classes de couleurs en partitionnant le graphe en stables (*independent sets*), soit colorie les nœuds séquentiellement (SC, *Sequential Coloring*).

Pour implémenter des algorithmes de SC exact, la Programmation Par Contraintes (PPC) est un outil efficace et élégant [7] : d'une part le problème se modélise naturellement en associant des variables à domaine fini à chaque nœud et des contraintes de diséquation à chaque arête, et d'autre part le B&B est l'algorithme de recherche standard dans les systèmes de PPC. Nous combinons cette modélisation basique avec diverses techniques visant à réduire la taille de l'espace de recherche et guider son exploration. Nous proposons également d'exploiter plus avant la structure des graphes en recherchant un *ensemble de cliques* avec un algorithme *glouton* pour améliorer l'inférence et l'ordre d'instanciation des variables [1].

L'algorithme ainsi obtenu permet de résoudre un problème de conception d'un réseau de routes aériennes directes et présente de bonnes performances sur les instances applicatives du banc de tests proposé dans [6].

2 Cliques, inférence et heuristiques en PPC

Recherche gloutonne de cliques Un ensemble de cliques (sous-optimales) est pré-calculé par un algorithme glouton très simple qui tente de construire des cliques *maximales* (au sens de l'inclusion) à partir de chaque nœud du graphe. Ces cliques sont utilisées pour ajouter au modèle des *contraintes globales* « tous distincts » (*all-different*) qui assurent la satifiabilité des sous-problèmes correspondants, ce qui permet d'inférer des échecs plus précocement.

Clique maximale Nous utilisons le B&B (exact) décrit dans [3] pour rechercher la clique *maximum* et obtenir une borne inférieure (LB) du GC. [3] conjecture que cette borne coïncide avec le nombre chromatique pour les graphes applicatifs, ce qui permet de prouver l'optimalité d'un coloriage de cardinal LB. Même si cette conjecture est fausse en général pour notre application, les instances qui présentent

cette propriété sont résolus très efficacement avec cette technique et la clique maximale est obtenue pour tous les exemples testés.

Nous définissons une contrainte pour implémenter en PPC les règles d'élagage proposées par [3] sur des variables booléennes associées à chaque nœud et spécifiant s'il fait partie de la clique. Un B&B standard est ensuite invoqué pour calculer la clique maximum, puis sa taille est utilisée comme borne inférieure du coloriage. Toutes les cliques découvertes lors de cette étape sont également ajoutées à celles calculées par l'algorithme glouton pour poser des contraintes globales et guider l'exploration.

Bien que la recherche de la clique maximum soit en lui-même un problème NP-dur, les performances de notre algorithme et celles rapportées par [3] témoignent de l'intérêt et de l'efficacité de cette approche.

Heuristiques Diverses heuristiques classiques d'instanciation des variables ont été essayées avec notre algorithme : BRÉLAZ, `dom/deg` etc. ainsi que leur combinaison avec une stratégie guidée par les cliques découvertes lors de la première phase, ordonnées par taille décroissante. L'effet escompté est de susciter les puissantes propagations des contraintes globales « tous différents » associées aux cliques les plus grandes.

3 Applications et résultats

Notre algorithme a été appliqué avec succès à la conception d'un réseau de routes aériennes directes : une route correspond à chaque flux d'avions partageant même origine et destination, et des niveaux de vols (couleurs) différents sont alloués aux flux qui s'intersectent. L'algorithme est très efficace sur ce problème et permet d'établir la faisabilité de ce réseau pour structurer l'espace français. Notre algorithme présente également de bonnes performances sur les instances applicatives des bancs de tests classiques de GC [6].

Bien que la PPC soit peu utilisée dans la littérature pour le GC, nous montrons qu'elle peut permettre de résoudre efficacement des problèmes industriels et servir de plateforme d'intégration de haut niveau pour les sophistications des algorithmes d'optimisation combinatoire.

Références

[1] N. Barnier, P. Brisset (2002). Graph Coloring for Air Traffic Flow Management. *CP-AI-OR*.

[2] D. Brélaz (1979). New Methods to Color the Vertices of a Graph. *CACM*.

[3] O. Coudert (1997). Exact Coloring of Real-Life Graphs is Easy. *DAC*.

[4] F. Leighton (1979). A Graph Colouring Algorithm for Large Scheduling Problems. *Journal of Research of the National Bureau of Standards*.

[5] C. Morgenstern (1986). Chromatic Number Approximation using Simulated Annealing. *ACM Mountain Regional Meeting*.

[6] M. Trick (2002). COLOR02/03 : Graph Coloring and its Generalizations. *mat.gsia.cmu.edu/COLOR02*.

[7] P. Van Hentenryck (1990). A Logic Language for Combinatorial Optimization. *Annals of Operations Research*.