Recherche Tabou renforcée pour la coloration de graphe

Daniel Porumbel[1,2], Jin Kao Hao[1], et Pascale Kuntz[2]

[1] LERIA, Université d'Angers, 2 Bd Lavoisier, Angers 49045 Cedex 01, France
daniel.porumbel@univ-nantes.fr,hao@info.univ-angers.fr
[2] LINA, Ecole Polytechnique de l'Université de Nantes, 44306 Nantes, France
pascale.kuntz@univ-nantes.fr

Mots-Clefs. optimisation combinatoire, coloration de graphes, méta-heuristiques, recharche tabou, fonction d'évaluation.

1 Introduction

Le problème de la coloration de graphe est probablement un des problèmes combinatoires les plus étudiés en mathématiques discrètes. Etant donné un graphe G, le problème de coloration (COL) consiste à déterminer le plus petit nombre de couleurs (le nombre chromatique χ) pour pouvoir construire une coloration sans conflits. Une coloration en K couleurs est sans conflits (K-coloration légale) s'il n'y a pas deux sommets adjacents avec la même couleur.

On trouve dans la littérature un très grand nombre d'algorithmes heuristiques pour la résolution approchée de COL [2]. Divers algorithmes exacts existent également, mais leur application dans le cas général est limitée aux graphes de taille réduite (les graphes aléatoires de densité 50% et d'ordre 100 au maximum selon [7] par exemple).

La recherche Tabou a été appliquée pour la première fois à COL en 1987 par Hertz et de Werra [1] et a permis d'obtenir de bons résultats. Depuis, d'autres algorithmes Tabou ont été développés avec des performances supérieures [1, 3, 4, 6]. Encore aujourd'hui, ces algorithmes font partie des principaux algorithmes de référence pour COL, bien que les meilleurs résultats soient détenus par des algorithmes hybrides plus complexes [5, 6, 8]. Dans ce papier, nous présentons un algorithme Tabou "renforcé" (nommé RTS, pour "Reinforced Tabu Search") pour COL et décrivons des résultats expérimentaux sur l'ensemble complet du célèbre benchmark Dimacs [2].

2 Algorithme

Etant donné un graphe $G(V, E)$, la procédure suivante est utilisée pour trouver une valeur approximative de χ. Le nombre de couleurs est initialisé à une valeur élevée, e.g. $K = |V|$. Ensuite, une K-coloration est recherchée pour cette valeur de K. Chaque fois qu'une K-coloration est trouvée (sans conflit), K est décrémenté de 1 ($K = K-1$). L'algorithme s'arrête lorsqu'il n'est plus possible de résoudre le problème. Le dernier K pour lequel l'algorithme a trouvé une coloration est le nombre chromatique approximatif de G.

Pour le problème de K-coloration (donc pour un G et un K fixés), l'algorithme Tabou générique commence avec une K-coloration initiale conflictuelle qui peut être engendrée de manière aléatoire. Puis, à chaque itération, il choisit un sommet i en conflit pour remplacer sa couleur c par une nouvelle couleur c' pour engendrer une K-coloration voisine. Selon le principe de la recherche Tabou, ce choix s'opère parmi les meilleurs voisins autorisés. Dans notre cas, il s'agit d'un couple (i, c') qui conduit au gain le plus important en terme de nombre de conflits.

Si plusieurs choix sont possibles, l'un d'eux est pris au hasard. Pour que ce choix du meilleur voisin soit réalisé rapidement, des structures de données spécifiques sont utilisées pour permettre une évaluation incrémentale du voisinage. La paire (i, c) est ainsi mise dans la liste tabou pour interdire de réaffecter la couleur c au sommet i pendant les prochaines tl itérations. La valeur tl change pendant la recherche en fonction du nombre des sommets en conflit et en fonction de deux variables aléatoires [4,5,6]. La recherche locale se termine soit lorsque l'on trouve une solution (une coloration sans conflits), soit lorsque l'on arrive au nombre d'itérations maximal fixé a priori.

3 Nouvelles fonctions d'évaluation

La fonction d'évaluation f_c, tout comme le voisinage, est un des éléments clés de tout algorithme de recherche locale ; elle sert de "guide" pour orienter la recherche dans un espace combinatoire complexe. Une grande partie des algorithmes de recherche locale pour la coloration (y compris Tabou) utilise simplement comme fonction d'évaluation le cardinal de l'ensemble CE des arêtes en conflit : $f_c = |CE|$. Cette fonction a néanmoins un inconvénient majeur : elle ne peut pas faire la distinction entre plusieurs colorations ayant le même nombre de conflits. Or, certains conflits peuvent s'avérer plus faciles, ou difficiles, que d'autres à résoudre.

Pour prendre en compte cette information supplémentaire dans la recherche, nous avons recours à une nouvelle fonction d'évaluation $\widetilde{f}$ qui associe à chaque conflit un poids en relation avec sa difficulté potentielle :

$$\widetilde{f} = f_c - h = \sum_{\{i,j\} \in CE} (1 - h_{ij}) \tag{1}$$

Pour définir le terme h_{ij}, nous proposons deux approches. La première est basée sur le degré des sommets i et j d'une arête $\{i, j\}$. On affecte un poids plus important aux arêtes incidentes à des sommets de forts degrés ; ces derniers sont considérés comme plus difficiles pour la résolution de conflit. La deuxième approche utilise un mécanisme d'apprentissage simple basé sur la distribution de la fréquence des changements de couleur dans une première phase de recherche : on considère que les sommets dont la couleur varie le plus sont plus difficiles et on utilise cette information dans une deuxième phase de la recherche.

4 Résultats et conclusions

Pour évaluer la performance de l'algorithme RTS, nous avons mené des expérimentations sur tous les benchmarks du challenge Dimacs [2] et des comparaisons avec plus de 10 algorithmes parmi les plus performants.

Par rapport aux quatre autres algorithmes Tabou de la littérature, nous avons observé que RTS obtient des meilleures colorations (K inférieurs) dans la moitié des cas, et des résultats similaires (même K) pour l'autre moitié. Au niveau de la recherche locale, RTS égalise ou améliore strictement les résultats de trois méthodes locales (ILS, VNS, ALS) sur quatre. Enfin, par rapport aux algorithmes distribués et hybrides (les plus performants sur les graphes difficiles de très grande taille), RTS trouve également la meilleure coloration connue pour 24 graphes Dimacs (sur 36).

Références

1. A. Hertz et D. de Werra. Using tabu search techniques for graph coloring, *Computing*, 39(4) :345-351, 1987.
2. D.S. Johnson et M. Trick (Eds). Cliques, coloring and satisfiability : Second Dimacs implementation challenge, American Mathematical Society, 1996.
3. C. Fleurent et J. A. Ferland. Object-oriented implementation of heuristic search methods for graph coloring, maximum clique and satisfiability, in [2], pages 619-652, 1996.
4. R. Dorne et J.-K. Hao. Tabu search for graph coloring, T-coloring and set T-colorings. *Meta-Heuristics : Advances and Trends in Local Search Paradigms for Optimization*, pages 77-92, 1998.
5. R. Dorne et J.-K. Hao. A new genetic local search algorithm for graph coloring. *Lecture Notes in Computer Science*, 1498 : 745-754, Springer-Verlag, 1998.
6. P. Galinier et J.-K. Hao, Hybrid evolutionary algorithms for graph coloring, *Journal of Combinatorial Optimization*, 3(4) : 379-397, 1999.
7. D.S. Johnson, C.R. Aragon, L.A. McGeoch et C. Schevon. Optimization by simulated annealing : An experimental evaluation ; Part II, Graph coloring and number partitioning, *Operations Research*, 39(3) :378-406, 1991.
8. C.A. Morgenstern, Distributed coloration neighborhood search, in [2], pages 335-357, 1996.