

Un algorithme hybride pour le problème SAT

Frédéric Lardeux, Frédéric Saubion et Jin-Kao Hao

LERIA, Université d'Angers
2 Bd Lavoisier, F-49045 Angers Cedex 01
{lardeux,saubion,hao}@info.univ-angers.fr

Mots-clefs : SAT, algorithme hybride, recherche locale, algorithme évolutionnaire

1 Introduction

Le problème de satisfiabilité (SAT) [3] consiste à trouver une affectation booléenne validant une formule en logique propositionnelle. SAT a beaucoup d'applications possibles telles que les tests et vérifications de circuits intégrés (VLSI), le diagnostic de pannes, les emplois du temps... Le problème SAT est généralement considéré comme un problème de décision mais peut aussi être abordé comme :

- recherche de modèle : trouver une ou plusieurs solutions
- MAX-SAT : trouver une solution qui rende vraie le plus de clauses possibles

Les algorithmes dédiés au problème SAT se séparent en deux classes : les algorithmes complets et les algorithmes incomplets.

Les algorithmes complets répondent au problème de décision et sont basés sur la procédure Davis-Putnam-Loveland [1]. Leur principale différence réside dans la règle de branchement pour explorer l'arbre de recherche.

Les algorithmes incomplets traitent le problème de recherche de modèle et MAX-SAT. Ils reposent souvent sur la recherche locale et les algorithmes évolutionnaires. Ils représentent une approche permettant de trouver rapidement une solution pour les instances de grande taille.

Dans ce papier nous présentons GASAT, un nouvel algorithme hybride incomplet où une recherche tabou est insérée dans un mécanisme évolutionnaire. D'une certaine manière, GASAT présente des similitudes avec l'algorithme hybride proposé par [2]. Cependant, GASAT se distingue par l'originalité de son opérateur de croisement, la puissance de sa recherche tabou et par l'interaction de ces deux opérateurs.

2 L'Algorithme Génétique Hybride : GASAT

2.1 L'Algorithme GASAT

L'algorithme est obtenu par combinaison d'un croisement et d'une recherche locale. Pour une population donnée où chaque individu représente une affectation des variables de la formule booléenne étudiée, la première étape consiste à sélectionner les individus engendrant le moins de clauses fausses. Ensuite, deux parents sont choisis parmi ces individus et l'opérateur de croisement leur est appliqué. Le fils ainsi obtenu est amélioré par la recherche tabou puis intégré à la population initiale sous certaines conditions. Ce processus est réitéré jusqu'à ce qu'une solution soit trouvée ou que le nombre de croisements autorisés soit atteint.

2.2 L'Opérateur de Croisement et la Recherche Tabou

L'idée de notre croisement est de travailler sur les clauses plutôt que sur les variables ce qui permet de conserver une certaine structure au cours des générations. Ce croisement contient trois étapes principales :

- pour les clauses fausses simultanément chez les deux parents, le fils est construit de manière à les rendre vraies,
- pour les autres clauses, on recopie la valeur des variables identiques chez les deux parents,
- les variables du fils qui n'ont pas encore de valeur sont valuées aléatoirement.

La recherche tabou utilisée est celle décrite dans [4] sur laquelle nous avons ajouté un mécanisme de diversification. Lors de la recherche tabou, si les affectations proposées engendrent régulièrement la même clause fausse, alors cette clause est forcée à vrai en modifiant une de ses variables. Les clauses devenant fausses par cette modification sont elles aussi forcées à vrai. La recherche tabou reprend ensuite son travail en s'interdisant de modifier les variables traitées lors de la diversification.

3 Résultats Expérimentaux

Nous avons comparé GASAT à différents algorithmes incomplets [5]. Les tests ont été effectués sur des instances aléatoires et structurées, satisfiables ou non. Aucun algorithme ne domine clairement les autres. Les résultats montrent que GASAT est généralement le plus efficace sur les instances structurées en particulier, celles issues d'un problème de coloration.

4 Conclusion

L'algorithme GASAT présenté ici n'est encore qu'une version préliminaire. Il donne déjà des résultats compétitifs par rapport à deux algorithmes de l'état de l'art. Nous étudions son comportement par rapport aux différentes classes de problèmes (aléatoire, coloration, ...) et essayons d'améliorer ces performances en travaillant sur de nouveaux croisements.

Références

- [1] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, Jul 1962.
- [2] Charles Fleurent and Jacques A. Ferland. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63:437–461, 1997.
- [3] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman & Company, San Francisco, 1978.
- [4] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, 1998.
- [5] Jin-Kao Hao, Frédéric Lardeux, and Frédéric Saubion. Evolutionary computing for the satisfiability problem. *soumis à 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2003)*.