

Apprentissage de clauses nobetters dans les solveurs séparation et évaluation pour Max-SAT

André Abramé, Djamal Habet

▶ **To cite this version:**

André Abramé, Djamal Habet. Apprentissage de clauses nobetters dans les solveurs séparation et évaluation pour Max-SAT. Actes des Treizièmes Journées Francophones de Programmation par Contraintes (JFPC 2017)), Jun 2017, Montreuil-sur-Mer, France. hal-01786548

HAL Id: hal-01786548
https://amu.hal.science/hal-01786548

Submitted on 7 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Apprentissage de clauses nobetters dans les solveurs séparation et évaluation pour Max-SAT

André Abramé **Djamal Habet**

Aix-Marseille Université, Université de Touloun, CNRS, ENSAM, LSIS, Marseille, France
{andre.abrame,djamal.habet}@lsis.org

Résumé

Nous introduisons une nouvelle méthode d'apprentissage de clauses dites *nobetters* pour les solveurs séparation et évaluation pour Max-SAT. Elle s'inspire de l'apprentissage de clauses *nogoods* utilisé par les solveurs SAT basés sur l'analyse de conflits (CDCL). Elle a pour objectif de permettre une meilleure résolution des instances industrielles par une meilleure prise en compte de leurs structures. Ce travail est un résumé d'un article que nous avons publié à la conférence ICTAI 2016 [1].

1 Introduction

Le problème Max-SAT consiste à déterminer, pour une formule propositionnelle en forme normale conjonctive, une interprétation des variables de cette formule maximisant le nombre de clauses satisfaites.

Les solveurs de type séparation et évaluation (S&E) pour Max-SAT ont montré leur efficacité sur les instances aléatoires et *crafted*. Cependant, sur les instances fortement structurées (comme celles issues de problèmes industriels), ils sont sensiblement moins performants que d'autres types de solveurs (notamment basés sur des appels itératifs à un solveur SAT moderne). Cela peut s'expliquer par leur incapacité à exploiter les informations structurelles de ces instances. À l'inverse, les solveurs modernes CDCL (Conflict-Driven Clause Learning) pour SAT sont très efficaces sur les instances industrielles. Une des raisons expliquant cette efficacité est le mécanisme d'apprentissage des clauses *nogoods*.

Dans ce papier, nous proposons un nouveau mécanisme d'apprentissage pour les solveurs S&E inspiré de celui des solveurs modernes. Il est basé sur la notion de clauses *nobetters*. Leur but est d'interdire les interprétations partielles dont les extensions ne permettront pas d'améliorer la meilleure solution trouvée jusqu'à présent. L'apprentissage des clauses *nobetters* permet de transposer aux solveurs S&E pour Max-SAT des techniques associées aux *nogoods* dans les solveurs modernes pour SAT.

2 Apprentissage de clauses nogoods dans les solveurs SAT

L'idée sous-jacente à l'apprentissage de clauses *nogoods* consiste à identifier les causes des conflits (i.e. un sous-ensemble de l'interprétation courante ayant mené au conflit) en analysant les étapes d'applications des règles d'inférences sémantiques utilisées pour étendre l'interprétation courante. Pour éviter la répétition de cette même interprétation, qui mènerait au même conflit, une clause (la clause *nogood*) est ajoutée à la formule. Ainsi, cela permet de limiter la redondance dans l'exploration de l'arbre de recherche tout en détectant plus tôt les conflits. De plus, ces clauses et les informations déduites durant leur construction sont utilisées par les solveurs modernes pour guider l'exploration de l'espace de recherche.

3 Apprentissage de clauses nobetters dans les solveurs Max-SAT

Comme les solveurs SAT modernes, les solveurs S&E pour Max-SAT explorent l'espace de recherche en construisant un arbre de recherche. Ils maintiennent deux valeurs : la borne supérieure (UB), qui est la meilleure solution trouvée jusqu'à présent et la borne inférieure (LB) qui correspond à une (sous-)estimation de la meilleure solution accessible dans la branche courante de l'arbre de recherche. À un nœud donné de l'arbre, si LB $\geq$ UB alors ils effectuent un retour-arrière (*backtrack*) jusqu'à atteindre une branche in-

explorée de l'arbre de recherche.

Lorsqu'un conflit est détecté (i.e. que LB $\geq$ UB), on sait que le sous-ensemble de l'interprétation courante qui apparaît dans les clauses falsifiées ne peut être étendu à une interprétation complète améliorant la meilleure solution connue. Comme dans le cadre des *nogoods* pour SAT, on peut alors analyser la séquence d'application des règles d'inférence sémantiques pour identifier un sous-ensemble de l'interprétation courante ayant mené au conflit, puis ajouter une nouvelle clause dure interdisant cette interprétation : la clause *nobetter*.

Pour mettre en place un tel système, il est nécessaire de redéfinir la notion de conflit. En effet, contrairement à SAT où il suffit d'une clause falsifiée pour rendre une interprétation conflictuelle, dans le cadre de Max-SAT il faut considérer toutes les clauses falsifiées ayant participé au calcul de la borne inférieure. De plus, les règles d'inférence sémantiques utilisées par les solveurs S&E pour Max-SAT ne sont pas les mêmes que dans le cadre de SAT. Nous avons donc défini, pour chacune d'entre elles (propagation unitaire réelle, règle des littéraux purs et règle des clauses unitaires dominantes), les causes provoquant leur application. Ces deux éléments sont suffisants pour construire un graphe d'implications décrivant la séquence d'application des règles d'inférence sémantiques ayant mené à l'interprétation courante. L'analyse de ce graphe, en partant des clauses falsifiées, permet de construire la clause *nobetter*.

Comme indiqué ci-dessus, plusieurs clauses falsifiées doivent être analysées pour produire une clause *nobetter*. Par conséquent, les clauses *nobetters* apprises seront potentiellement plus grandes que dans le cadre des solveurs SAT. Pour y remédier, nous avons modifié la procédure de construction des clauses *nobetters* en poursuivant l'analyse des causes des affectations jusqu'à atteindre le point d'implication unique de chaque niveau de décision apparaissant dans la clause apprise.

Enfin, il est intéressant de noter que la plupart des techniques utilisées dans les solveurs SAT modernes en combinaison avec l'apprentissage des clauses *nogoods* peuvent être adaptées aux solveurs S&E pour Max-SAT utilisant l'apprentissage des *nobetters*. En particulier, on peut effectuer des retours-arrières non-chronologique (*backjump*), des redémarrages ou encore mettre en place des heuristiques de branchement basées sur l'apparition des variables dans les conflits. L'intégration de ces techniques n'est cependant pas triviale et nécessitera une étude approfondie des interactions avec les autres composants des solveurs S&E.

4 Résultats expérimentaux

Nous avons évalué expérimentalement l'impact de l'apprentissage des clauses *nobetters* dans notre solveur AHMAXSAT. Les tests sont réalisés sur les instances et en suivant le protocole de la Max-SAT Evaluation 2015.

D'une part, les résultats obtenus montrent que le calcul des clauses *nobetters* entraîne une augmentation de 20% du temps moyen de résolution à cause du maintien des informations nécessaires à leur calcul. Par ailleurs, sur les instances industrielles, l'apprentissage des clauses *nobetters* permet de réduire sensiblement le nombre de décisions nécessaires pour résoudre les instances (-15% sur les instances industrielles Max-SAT partielles et -54% sur le même type d'instances mais pondérées). Par conséquent, plus d'instances sont résolues (231 contre 217 pour la version originale d'AHMAXSAT). Cependant, le temps moyen de résolution sur ces instances est dans l'ensemble inchangé, le surcoût induit par le calcul des clauses *nobetters* réduisant le gain obtenu par la diminution du nombre de décisions.

Enfin, malgré cette amélioration des performances sur les instances industrielles, les résultats obtenus restent encore en deçà de ceux des solveurs Max-SAT les plus performants sur ces catégories d'instances.

5 Conclusion

Nous avons introduit un nouveau mécanisme d'apprentissage de clauses inspiré de celui utilisé par les solveurs SAT modernes. Nous avons posé les bases théoriques de ce mécanisme et présenté les algorithmes permettant sa mise en œuvre. Les résultats expérimentaux que nous avons obtenus montrent que l'apprentissage des clauses *nobetters* améliore sensiblement les performances de notre solveur sur les instances industrielles. Ce travail est une étape et mérite d'être poursuivi. En effet, notre implémentation de l'apprentissage des clauses *nobetters* remplit seulement un de ses objectifs : limiter la redondance lors de la recherche. Il serait intéressant d'implémenter d'autres techniques utilisées dans les solveurs SAT modernes, telles que les redémarrages ou des heuristiques de branchement basées sur l'activité des variables.

Références

[1] A. Abramé and D. Habet. Learning Nobetter Clauses in Max-SAT Branch and Bound Solvers. *Proceedings of the 28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2016)*, pages 452–459, 2016.