Des réfutations SAT aux réfutations Max-SAT *

Matthieu Py, Mohamed Sami Cherif, Djamal Habet

Des réfutations SAT aux réfutations Max-SAT*

Matthieu Py† **Mohamed Sami Cherif** **Djamal Habet**

Aix-Marseille Université, Université de Toulon, CNRS, LIS, Marseille, France
{matthieu.py, mohamed-sami.cherif, djamal.habet}@univ-amu.fr

Résumé

Adapter une preuve SAT par résolution en une preuve valide pour Max-SAT sans augmenter considérablement sa taille est une question longtemps restée ouverte. Cet article, qui résume les travaux publiés dans [3], contribue à y répondre en présentant des adaptations linéaires de réfutations SAT en réfutations Max-SAT, dans les cas *tree-like regular*, *tree-like* et *semi-tree-like*. On étend également ces résultats en proposant une adaptation complète pour toute réfutation SAT qui est exponentielle dans le pire des cas.

1 Introduction

Étant donnée une formule sous Forme Normale Conjonctive, le problème Max-SAT consiste à déterminer le nombre maximum de clauses qu'il est possible de satisfaire par une affectation des variables alors que le problème SAT consiste simplement à déterminer si la formule est satisfiable. Dans le contexte du problème SAT, une formule peut être démontrée insatisfiable à l'aide d'une séquence de résolutions [4], appelée réfutation par résolution, qui déduit de la formule initiale de nouvelles clauses jusqu'à en déduire la clause vide qui, par définition, est impossible à satisfaire. De même, pour Max-SAT, on utilise un système de preuve bien connu basé sur la règle de la max-résolution [2], qui étend la règle de résolution utilisée dans SAT. Les séquences de max-résolutions sont plus contraintes que les séquences de résolutions car la max-résolution remplace les prémisses par les conclusions, ce qui consomme les prémisses et empêche de les utiliser à nouveau. Adapter une réfutation SAT (par résolution) en une réfutation valide pour Max-SAT est par conséquent simple si la formule est *read-once*, c'est à dire si chaque clause est utilisée une seule fois comme prémisse d'une résolution. Dans ce cas, il suffit de remplacer chaque étape de résolution par une max-résolution pour obtenir une réfutation valide pour Max-SAT dont la taille (exprimée en nombres d'étapes de transformation) est linéaire par rapport à la taille de la réfutation SAT [1]. En revanche, adapter une réfutation SAT en une réfutation Max-SAT dans le cas général et sans augmenter considérablement sa taille reste une question ouverte.

Dans cet article, on propose d'apporter une contribution pour répondre à cette question. Pour cela, on augmente la règle de la max-résolution avec la règle du *split*, permettant de remplacer une clause par deux clauses en y ajoutant un littéral supplémentaire. Ainsi, on est désormais capable d'adapter n'importe quelle réfutation SAT *tree-like regular* en une réfutation Max-SAT de taille linéaire. On étend ensuite ce résultat aux cas *tree-like* et *semi-tree-like*. Enfin, on propose une adaptation des réfutations SAT dans le cas général mais dont la taille de la réfutation Max-SAT obtenue peut être exponentielle dans le pire des cas.

2 Des réfutations tree-like regular aux réfutations Max-SAT

Considérons tout d'abord le cas des réfutations par résolution *tree-like regular*. Une réfutation est dite *tree-like* si aucune clause intermédiaire (déduite par une résolution) n'est utilisée plusieurs fois en tant que prémisse d'une étape de résolution. Une réfutation est dite *regular* si chacune de ses branches (séquences de résolutions commençant depuis des clauses de la formule et se terminant par la déduction de la clause vide) contient au plus une résolution par variable.

Exemple 1. $\phi = (\overline{x_1} \vee x_3) \wedge (x_1) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee \overline{x_3})$. *La réfutation par résolution de* ϕ *représentée dans la figure 1 n'est pas read-once mais elle est tree-like regular.*

*Cet article est un résumé de [3].
†Papier doctorant : Matthieu Py est auteur principal.

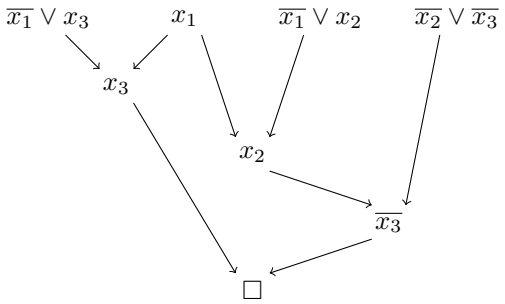

FIGURE 1 – Réfutation par résolution

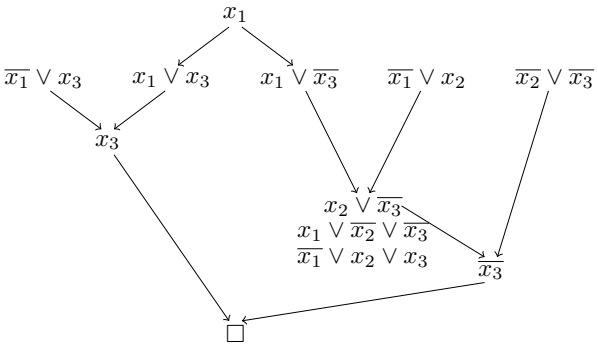

FIGURE 2 – Application de la règle *split* pour générer une réfutation Max-SAT

Théorème 1. *Étant donnée une formule insatisfiable ϕ et une réfutation par résolution tree-like regular P de ϕ, il existe une réfutation Max-SAT de ϕ contenant $O(|P|)$ étapes d'inférence.*

Comme énoncé dans le théorème 1, il est possible d'adapter une réfutation *tree-like regular* en une réfutation Max-SAT de taille linéaire par rapport à la taille de la réfutation SAT. Pour cela, si une clause est utilisée plusieurs fois, on cherche le point de jonction de toutes les branches partant de cette clause. Ce point de jonction est une étape de résolution sur une variable telle que l'application de la règle du *split* sur la clause initiale et cette variable génère deux clauses qui permettent de la remplacer sans impacter la validité de la preuve. En répétant cette opération autant de fois que nécessaire, on retombe sur une réfutation que l'on peut qualifier de *read-once* et on peut remplacer chaque résolution par une max-résolution pour obtenir une réfutation Max-SAT de la formule de départ.

Exemple 2. *Dans la réfutation de l'exemple 1, la clause (x_1) est utilisée deux fois et le point de jonction des deux branches qui partent de (x_1) est une résolution sur la variable x_3. On applique donc une étape de split sur la clause (x_1) et la variable x_3 pour obtenir deux nouvelles clauses permettant de remplacer (x_1). On remplace enfin chaque résolution par une max-résolution et on obtient la réfutation Max-SAT de la figure 2.*

3 Extension aux cas tree-like, semi-tree-like et général

Le résultat linéaire sur les réfutations *tree-like regular* s'étend aux réfutations *tree-like* en utilisant un résultat connu permettant de transformer n'importe quelle réfutation *tree-like* en une réfutation *tree-like regular* plus petite [5]. Pour cela, à chaque fois que l'on détecte deux résolutions sur la même variable dans une branche, on supprime la première résolution ainsi que toutes les autres résolutions devenues inapplicables. Ce résultat s'étend également au cas *semi-tree-like*, où chaque branche de la réfutation contient au plus une clause utilisée plusieurs fois. Dans ce cas-là, on découpe la réfutation en deux morceaux indépendants, le premier *read-once* et le second *tree-like* et on les traite de manière indépendante avant de les refusionner.

Théorème 2. *Étant donnée une formule insatisfiable ϕ et une réfutation par résolution semi-tree-like P de ϕ, il existe une réfutation Max-SAT de ϕ contenant $O(|P|)$ étapes d'inférence.*

Enfin, dans le cas général, il est possible de rendre la réfutation semi-tree-like moyennant une augmentation au pire exponentielle de sa taille. Pour cela, on duplique chaque portion de la preuve conduisant à une clause intermédiaire utilisée plusieurs fois afin d'en créer autant de copies que nécessaire, ce qui permet de rendre la preuve semi-tree-like et de revenir au cas précédent.

Théorème 3. *Étant donnée une formule insatisfiable ϕ et une réfutation par résolution P de ϕ, il existe une réfutation Max-SAT de ϕ contenant $O(2^{\mu(P)} \times |P|)$ étapes d'inférence, où $\mu(P)$ est le nombre de multi-utilisation des clauses intermédiaires de P.*

Références

[1] Federico HERAS et Joao MARQUES-SILVA : Read-Once resolution for Unsatisfiability-Based Max-SAT Algorithms. *In IJCAI*, 2011.

[2] María Luisa BONET, Jordi LEVY et Felip MANYÀ : Resolution for Max-SAT. *Artificial Intelligence*, 2007.

[3] M. PY, M. S. CHERIF et D. HABET : Towards Bridging the Gap Between SAT and Max-SAT Refutations. *In 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020.

[4] J. A. ROBINSON : A machine-oriented logic based on the resolution principle. *In Journal of the Association for Computing Machinery*, 1965.

[5] Alasdair URQUHART : The Complexity of Propositional Proof. *Bulletin of Symbolic Logic*, 1995.