



HAL
open science

Des réfutations SAT aux réfutations Max-SAT *

Matthieu Py, Mohamed Sami Cherif, Djamal Habet

► **To cite this version:**

Matthieu Py, Mohamed Sami Cherif, Djamal Habet. Des réfutations SAT aux réfutations Max-SAT *. Journées Francophones de Programmation par Contraintes (JFPC 2021), 2021, Conférence virtuelle, France. hal-03343027

HAL Id: hal-03343027

<https://hal.science/hal-03343027v1>

Submitted on 13 Sep 2021

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.

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 prémisses 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 prémisses 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. $\phi = (\bar{x}_1 \vee x_3) \wedge (x_1) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee \bar{x}_3)$. 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.

