



**HAL**  
open science

## Certificats d'optimalité pour Max-SAT

Matthieu Py, Mohamed Sami Cherif, Djamel Habet

► **To cite this version:**

Matthieu Py, Mohamed Sami Cherif, Djamel Habet. Certificats d'optimalité pour Max-SAT. JFPC 2022, Jun 2022, Saint-Etienne, France. hal-03737729

**HAL Id: hal-03737729**

**<https://hal-amu.archives-ouvertes.fr/hal-03737729>**

Submitted on 25 Jul 2022

**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.

# Certificats d’optimalité pour Max-SAT

Matthieu Py, Mohamed Sami Cherif, Djamel Habet

Aix-Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

{ matthieu.py, mohamedsami.cherif, djamel.habet }@univ-amu.fr

## Résumé

Dans ce papier, on présente un outil, MS-Builder, qui génère des certificats pour le problème Max-SAT, en appelant itérativement un oracle SAT pour générer une réfutation par résolution de la formule qui est ensuite rendue valide pour Max-SAT et appliquée sur la formule. Ce procédé est répété jusqu’à ce que la formule restante devienne satisfiable. On propose également un outil, MS-Checker, capable de vérifier les certificats Max-SAT basés sur l’application de règles d’inférence Max-SAT. Cet article résume les travaux publiés à la conférence SAT 2021 [4].

## Mots-clés

Optimisation Combinatoire, Problème Max-SAT, Problème SAT, Certificats d’Optimalité

## 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 [5], 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. Ainsi, les solveurs SAT modernes, en plus de proposer la résolution du problème SAT, fournissent aussi la preuve de la réponse affirmée, soit sous forme de modèle (si la formule est satisfiable), soit sous forme de réfutation (si elle ne l’est pas). En revanche, si les solveurs Max-SAT sont de plus en plus efficaces chaque année, ils ne produisent pas encore de certificats d’optimalité sur la solution trouvée. Les systèmes de preuve pour Max-SAT, ont juste été étudiés théoriquement ou pour leur aptitude à aider à la résolution du problème Max-SAT.

Dans cet article, on propose un outil indépendant, MS-Builder, capable de produire des certificats d’optimalité pour le problème Max-SAT. Cet outil se base sur l’adaptation des réfutations par résolution pour Max-SAT [3]. Itérativement, l’outil fait appel à un oracle SAT pour obtenir une réfutation par résolution de la formule courante. Cette réfutation est adaptée pour Max-SAT en max-réfutation (qui

utilise les règles de la max-résolution [2] et du split [3]), puis appliquée sur la formule. Ce procédé est répété jusqu’à ce que l’oracle SAT détecte que la formule est devenue satisfiable. L’ensemble des transformations calculées, plus le modèle pour la formule résiduelle, forme un certificat d’optimalité pour le problème Max-SAT. On propose aussi un outil associé, MS-Checker, pour vérifier les certificats d’optimalité basés sur les règles d’inférence Max-SAT. Ces deux outils ont été testés expérimentalement sur les instances partielles non pondérées de la Compétition Max-SAT 2020 [1].

## 2 Fonctionnement de MS-Builder

**Exemple 1.** Soit  $\phi = (\overline{x_1} \vee x_3) \wedge (x_1) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2}) \wedge (\overline{x_3}) \wedge (x_2 \vee x_3)$ .

**Lecture du fichier :** MS-Builder récupère la formule en lisant le fichier représenté dans la Figure 1.

```
c formule avec optimum = 2
1 -1 3 0
1 1 0
1 -1 2 0
1 -2 -3 0
1 -2 0
1 -3 0
1 2 3 0
```

FIGURE 1 – Fichier avec formule

**Première itération :** MS-Builder fait appel à un oracle SAT, qui retourne que la formule initiale est insatisfiable, avec la réfutation par résolution de la Figure 2, qui est adaptée en la max-réfutation de la Figure 3 puis appliquée sur la formule. La formule courante est maintenant  $\phi = (\overline{x_1} \vee x_3) \wedge (x_1) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge \square$ .

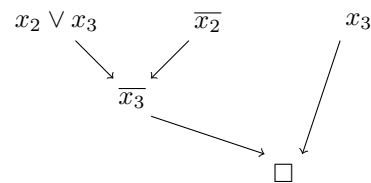


FIGURE 2 – Première réfutation par résolution

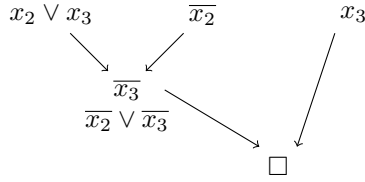


FIGURE 3 – Adaptation de la première réfutation

**Deuxième itération :** MS-Builder fait appel à un oracle SAT (en laissant de côté la clause vide), qui retourne que la formule initiale est insatisfiable, avec la réfutation par résolution de la Figure 4, qui est adaptée en la max-réfutation de la Figure 5 puis appliquée sur la formule. La formule courante est maintenant  $\phi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge \square \wedge \square$ .

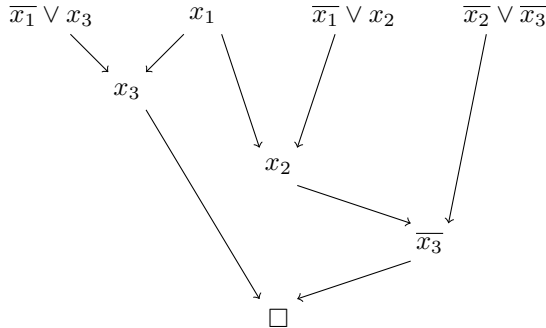


FIGURE 4 – Deuxième réfutation par résolution

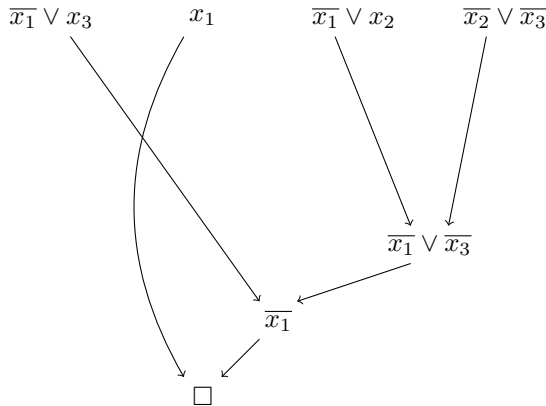


FIGURE 5 – Adaptation de la deuxième max-réfutation

**Troisième itération :** MS-Builder fait appel à un oracle SAT (en laissant de côté les deux clauses vides), qui retourne que la formule initiale est satisfiable, avec l'affectation  $x_1 = x_2 = x_3 = 0$ . Il retourne donc le certificat Max-SAT représenté dans la Figure 6, constitué de cinq étapes de max-résolution, plus l'affectation qui permet de satisfaire les clauses non vides restantes après transformation de la formule.

```

c read-once refutation
t msres < 1 2 3 | 1 -2 >
t msres < 1 3 | 1 -3 >
c semi-read-once refutation
t msres < 1 -1 2 | 1 -2 -3 >
t msres < 1 -1 3 | 1 -1 -3 >
t msres < 1 1 | 1 -1 >
o 2
v 000

```

FIGURE 6 – Certificat d'optimalité Max-SAT

### 3 Expérimentations

MS-Builder et MS-Checker sont mis à disposition de la communauté scientifique (<https://pageperso.lis-lab.fr/matthieu.py/en/software.html>) et ont été testés expérimentalement sur les instances partielles non pondérées de la Compétition Max-SAT 2020 [1]. En particulier, MS-Builder a été capable de construire des certificats d'optimalité complets pour 163 instances sur 576 instances (Figure 7).

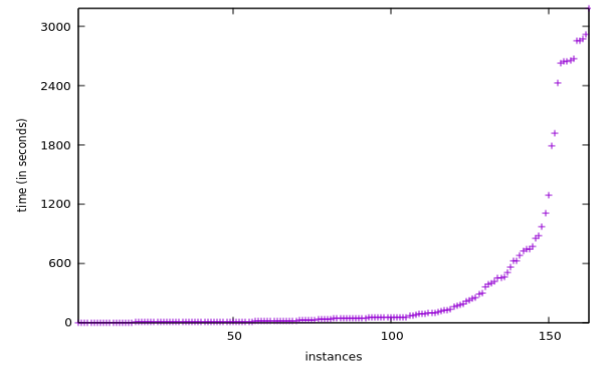


FIGURE 7 – Temps d'exécution (en secondes) pour la construction de certificats complets

### Références

- [1] Fahiem Bacchus, Matti Järvisalo, and Ruben Martins. MaxSAT Evaluation, 2020.
- [2] María Luisa Bonet, Jordi Levy, and Felip Manyà. A complete calculus for MAX-SAT. In *Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121, pages 240–251, 08 2006.
- [3] Matthieu Py, Mohamed Sami Cherif, and Djamel Habet. Towards Bridging the Gap Between SAT and Max-SAT Refutations. In *32nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2020*, pages 137–144. IEEE, 2020.
- [4] Matthieu Py, Mohamed Sami Cherif, and Djamel Habet. A Proof Builder for Max-SAT. In *Proceedings of the 24th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2021.
- [5] John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machinery*, 12 :23–41, 1965.