

# Vers l'extension de l'apprentissage pour Max-SAT

André Abramé

Djamal Habet

Aix Marseille Université, CNRS, ENSAM, Université de Toulon,  
LSIS UMR 7296, 13397, Marseille, France.  
{andre.abrame,djamal.habet}@lsis.org

## Résumé

Un des composants les plus importants des solveurs Séparation et Évaluation (S&E) pour Max-SAT est l'estimation de la borne inférieure. À chaque nœud de l'arbre de recherche, elle consiste à détecter par des méthodes basées sur la propagation unitaire, puis à transformer les sous-ensembles inconsistants (SI) disjoints présents dans la formule simplifiée par l'interprétation partielle courante. Les solveurs récents appliquent une forme limitée d'apprentissage en mémorisant les transformations par max-résolution des SI qui correspondent à des motifs prédéfinis. Cependant, le calcul de la borne inférieure reste extrêmement redondant et les mêmes SI sont détectés et traités de nombreuses fois durant l'exploration de l'arbre de recherche. Nous nous intéressons dans cet article à l'augmentation de l'apprentissage réalisé par les solveurs S&E. Nous introduisons de nouveaux ensembles de motifs de clauses qui produisent des clauses résolventes unitaires lorsqu'ils sont transformés par max-résolution. Nous étudions expérimentalement l'impact de l'apprentissage des transformations de ces motifs dans notre solveur ahmaxsat et analysons leurs effets sur son comportement.

## Abstract

One of the most critical components of Branch & Bound (BnB) solvers for Max-SAT is the estimation of the lower bound. At each node of the search tree, they detect the inconsistent subsets (IS) of the formula by unit propagation based methods and apply a treatment to them. The currently best performing Max-SAT BnB solvers perform a very little amount of memorization, thus the same IS may be detected and treated several times during the exploration of the search tree. We address in this paper the problem of increasing the learning performed by BnB solvers. We present new sets of clause patterns which produce unit resolvent clauses when they are transformed by max-resolution. We study experimentally the impact of these transformation' memorization in our solver ahmaxsat and we discuss their effects on the solver behavior.

## 1 Introduction

Le problème Max-SAT consiste à trouver, pour une formule CNF en entrée, une interprétation des variables Booléennes de la formule qui maximise (minimise) le nombre de clauses satisfaites (falsifiées). Dans la version pondérée de Max-SAT, un poids positif est associé à chaque clause et l'objectif est de maximiser (minimiser) la somme des poids des clauses satisfaites (falsifiées). Dans la version partielle de Max-SAT, les clauses de la formule sont divisées en deux catégories : les clauses dures et les clauses souples. L'objectif est alors de trouver une interprétation satisfaisant toutes les clauses dures et maximisant (minimisant) le nombre de clauses souples satisfaites (falsifiées). Enfin, la version partielle pondérée de Max-SAT est la combinaison des deux variantes précédentes. Par soucis de clarté, nous utilisons dans ce papier les notations associées au problème Max-SAT (non-pondéré et non-partiel). Cependant, les résultats présentés peuvent être étendus aux autres variantes de Max-SAT.

Parmi les méthodes complètes pour résoudre le problème Max-SAT, les algorithmes de type Séparation et Évaluation (S&E) tels que WMAXSATZ [12, 15], AKMAXSAT [9] ou MINIMAXSAT [7, 8] ont prouvé leur efficacité, en particulier sur les instances aléatoires et *crafted*. Ces solveurs explorent l'ensemble de l'espace de recherche en construisant un arbre de recherche. À chaque nœud de cet arbre, ils calculent la Borne Inférieure (BI) en additionnant le nombre de clauses falsifiées par l'interprétation partielle courante et une (sous-)estimation de celles qui deviendront falsifiées si l'on étend cette interprétation. Puis, ils comparent BI à la meilleure solution trouvée jusqu'à présent (la Borne Supérieure, BS). Si  $BI \geq BS$ , alors il n'est pas possible de trouver une meilleure solution en étendant la branche courante de l'arbre de recherche et les solveurs réalisent un retour-arrière (*backtrack*). Le calcul de BI, et en particulier l'estimation des clauses qui deviendront falsifiées est un des composants les plus importants des sol-

veurs S&E : il est coûteux en temps de calcul et de sa qualité dépend grandement le nombre de nœuds qui seront explorés.

Les solveurs S&E récents estiment le nombre de clauses qui seront falsifiées en comptant les Sous-ensembles Inconsistants (SI) disjoints présents dans la formule à chaque nœud de l'arbre de recherche. Ils utilisent des méthodes basées sur la propagation unitaire pour détecter les inconsistencies. Les étapes de propagation correspondantes sont ensuite analysées pour construire les SI. Chaque SI détecté doit ensuite être transformé pour qu'il ne soit compté qu'une fois. Deux méthodes de transformation sont utilisées par les solveurs S&E. Si un SI correspond à certains motifs prédéfinis, alors il est transformé par une série d'opérations de max-résolution [5, 6, 10, 11, 15] entre ses clauses, et la transformation est conservée dans la sous-partie de l'arbre de recherche. Ce traitement agit comme une forme (limitée) d'apprentissage. Les clauses des SI qui ne correspondent à aucun motif sont simplement supprimées de la formule ou transformées par max-résolution, et dans les deux cas les changements sont locaux au nœud courant (i.e. ils sont défaites avant la prochaine décision).

Nous proposons dans cet article d'augmenter la part d'apprentissage réalisée par les solveurs S&E. Nous nous intéressons en particulier aux sous-ensembles de la formule qui produisent, une fois transformés par max-résolution, des clauses résolventes unitaires. L'apprentissage de ces transformations a un double avantage. Elle permet de réduire la redondance dans la détection et la transformation des SI et les clauses unitaires ainsi produites améliorent la capacité de la propagation unitaire à détecter d'autres SI. Nous définissons de nouveaux motifs correspondants à ces sous-ensembles, puis nous étudions expérimentalement l'impact de leurs apprentissages sur le comportement de notre solveur AHMAXSAT [1, 2, 3, 4] en faisant varier la taille de ces motifs et celle des clauses qu'ils contiennent. Les résultats obtenus montrent l'intérêt de cette approche et donnent des indices sur l'impact de l'apprentissage des transformations par max-résolution sur le comportement du solveur.

Ce papier est organisé comme suit. Nous donnons dans la section 2 les notations et définitions utilisées dans la suite de cet article et nous présentons la règle de la max-résolution. Dans la section 3, nous présentons les méthodes existantes pour la détection et la transformation des SI et nous rappelons les schémas d'apprentissage utilisés dans les solveurs S&E les plus récents. Nous introduisons de nouveaux ensembles de motifs dans la section 4 et discutons de leur détection. La section 5 est dédiée à l'étude empirique de l'inclusion de ces nouveaux motifs dans le schéma d'apprentissage. Nous concluons dans la section 6.

## 2 Préliminaires

Nous donnons dans cette section les principales définitions et notations utilisées dans la suite de cet article et nous présentons la règle de la max-résolution, qui est la pierre angulaire des méthodes d'apprentissage pour Max-SAT.

### 2.1 Définitions et Notations

Une formule  $\Phi$  en forme normale conjonctive (CNF) définie sur une ensemble de variables propositionnelles  $X = \{x_1, \dots, x_n\}$  est une conjonction de clauses. Une clause  $c_j$  est une disjonction de littéraux et un littéral  $l_j$  est une variable  $x_i$  ou sa négation  $\bar{x}_i$ . Nous représenterons dans cet article les formules sous forme de multi-ensembles de clauses  $\Phi = \{c_1, \dots, c_m\}$  et les clauses sous forme d'ensembles de littéraux  $c_j = \{l_1, \dots, l_k\}$ .

Une interprétation des variables de  $X$  peut être représentée comme un ensemble  $I$  de littéraux ne pouvant contenir à la fois un littéral et sa négation. Si  $x_i$  est affectée à *vrai* (resp. *faux*), alors  $x_i \in I$  (resp.  $\bar{x}_i \in I$ ). On dira qu'une interprétation  $I$  est complète si  $|I| = |X|$  et qu'elle est partielle sinon.

Un littéral  $l$  est satisfait par une interprétation  $I$  si  $l \in I$  et il est falsifié si  $\bar{l} \in I$ . Une variable qui n'apparaît ni positivement ni négativement dans  $I$  est dite non affectée. Une clause est satisfaite par  $I$  si au moins un de ses littéraux est satisfait et elle est falsifiée si tous ses littéraux sont falsifiés. Les clauses vides (notées  $\square$ ) sont toujours falsifiées.

Résoudre une instance  $\Phi$  du problème Max-SAT consiste à trouver une interprétation complète qui maximise (minimise) le nombre de clauses satisfaites (falsifiées) de  $\Phi$ .

### 2.2 Règle de la max-résolution

La règle de la max-résolution [5, 6, 10] est une adaptation pour Max-SAT de la règle de la résolution pour SAT. Elle est définie comme suit :

$$\frac{c_i = \{x, y_1, \dots, y_s\}, c_j = \{\bar{x}, z_1, \dots, z_t\}}{cr = \{y_1, \dots, y_s, z_1, \dots, z_t\}, cc_1, \dots, cc_t, cc_{t+1}, \dots, cc_{t+s}}$$

avec :

$$\begin{aligned} cc_1 &= \{x, y_1, \dots, y_s, \bar{z}_1, z_2, \dots, z_t\}, \\ cc_2 &= \{x, y_1, \dots, y_s, \bar{z}_2, z_3, \dots, z_t\}, \\ &\dots \\ cc_t &= \{x, y_1, \dots, y_s, \bar{z}_t\}, \\ cc_{t+1} &= \{\bar{x}, z_1, \dots, z_t, \bar{y}_1, y_2, \dots, y_s\}, \\ cc_{t+2} &= \{\bar{x}, z_1, \dots, z_t, \bar{y}_2, y_3, \dots, y_s\}, \\ &\dots \\ cc_{t+s} &= \{\bar{x}, z_1, \dots, z_t, \bar{y}_s\} \end{aligned}$$

Les prémisses de la règle sont les clauses originelles  $c_i$  et  $c_j$  qui sont supprimées de la formule, tandis que les conclusions sont la clause résolvente  $cr$  et les clauses de compensation  $cc_1, \dots, cc_{t+s}$  ajoutées pour préserver l'équivalence de la formule.

### 3 Apprentissage dans les solveurs S&E de l'état de l'art

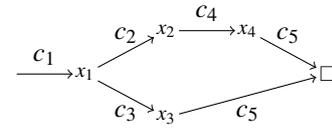
À chaque nœud de l'arbre de recherche, les solveurs S&E calculent la borne inférieure (BI) en estimant le nombre de sous-ensembles inconsistants (SI) disjoints présents dans la formule simplifiée par l'interprétation courante. Nous présentons dans cette section les principales techniques de détection et de transformation des SI, puis nous rappelons les schémas d'apprentissage utilisés par les solveurs S&E les plus performants.

Les solveurs S&E récents utilisent des méthodes basées sur la propagation unitaire (PU) pour détecter les SI (plus précisément la propagation unitaire simulée (PUS) [13] et la méthodes des littéraux contradictoires (LC) [14]). Pour rappel, la propagation unitaire consiste à satisfaire les littéraux apparaissant dans des clauses unitaires jusqu'à ce qu'une clause vide (un conflit) soit produite ou qu'il n'y ait plus de clauses unitaires. Quand une clause vide est générée par PU, un sous-ensemble inconsistant peut être construit en analysant les étapes de propagation ayant mené au conflit. Ces étapes de propagation peuvent être représentées par un graphe d'implications [16], où les nœuds sont les littéraux propagés et les arcs connectent les littéraux falsifiés des clauses unitaires aux littéraux qu'elles propagent.

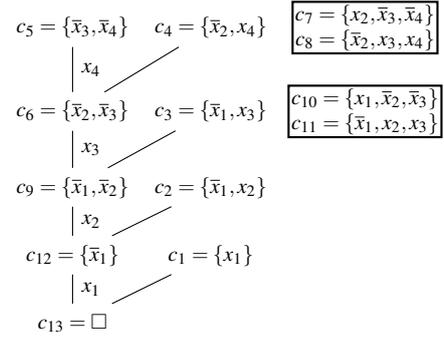
Une fois détecté, un SI peut être transformé en appliquant une série d'opérations de max-résolution entre ses clauses. Les clauses du SI sont supprimées de la formule tandis qu'une clause résolvente et un ensemble de clauses de compensation lui sont ajoutés. La clause résolvente est falsifiée par l'interprétation courante, donc PUS et LC ne sont plus nécessaires pour détecter l'inconsistance. En conservant ces transformations, les solveurs peuvent mémoriser les SI et éviter des détections et des transformations redondantes. L'exemple suivant illustre cette méthode de transformation.

**Exemple 1** *Considérons la formule  $\Phi = \{c_1, \dots, c_5\}$  avec  $c_1 = \{x_1\}$ ,  $c_2 = \{\bar{x}_1, x_2\}$ ,  $c_3 = \{\bar{x}_1, x_3\}$ ,  $c_4 = \{\bar{x}_2, x_4\}$  et  $c_5 = \{\bar{x}_3, \bar{x}_4\}$ . L'application de PUS sur  $\Phi$  conduit à la séquence de propagation  $\langle x_1 @ c_1, x_2 @ c_2, x_3 @ c_3, x_4 @ c_4 \rangle$  ( $x_1$  est propagée par la clause  $c_1$ , puis  $x_2$  par  $c_2$ , etc.). La figure 1a montre le graphe d'implications correspondant à cette séquence. La clause  $c_5$  est alors vide et  $\Phi$  est un sous-ensemble inconsistant. Sa transformation par max-résolution est réalisée comme suit. La max-résolution est d'abord appliquée entre les clauses  $c_5$  et  $c_4$  sur la variable  $x_4$ . La résolvente intermédiaire  $c_6 = \{\bar{x}_2, \bar{x}_3\}$  est*

*ajoutée à la formule ainsi que les clauses de compensation  $c_7 = \{x_2, \bar{x}_3, \bar{x}_4\}$  et  $c_8 = \{\bar{x}_2, x_3, x_4\}$ . Les clauses originelles  $c_4$  et  $c_5$  sont supprimées de la formule. Puis, la max-résolution est appliquée entre la résolvente intermédiaire  $c_6$  et la clause  $c_3$  sur la variable  $x_3$  et ainsi de suite. La figure 1b montre ces étapes de max-résolution (les cadres contiennent les clauses de compensation). On peut noter que cette méthode est proche de l'apprentissage de clause réalisé par les solveurs SAT modernes [16]. Après transformation complète, on obtient la formule  $\Phi' = \{\square, c_7, c_8, c_{10}, c_{11}\}$  avec  $c_{10} = \{x_1, \bar{x}_2, \bar{x}_3\}$  et  $c_{11} = \{\bar{x}_1, x_2, x_3\}$ .*



(a) Graphe d'implications



(b) Étapes de max-résolution

FIGURE 1 – Graphe d'implications et étapes de max-résolution appliquées sur la formule  $\Phi$  dans l'exemple 1.

Cette méthode de transformation a cependant des inconvénients qui empêchent sa généralisation. Tout d'abord, l'ajout des résolvents et des clauses de compensation peut conduire à une augmentation rapide la taille de la formule. Ensuite, il a été observé expérimentalement que l'apprentissage de ces transformations pouvait mener à une dégradation de la qualité de l'estimation de la BI, entraînant l'augmentation du nombre de nœuds explorés de l'arbre de recherche.

Pour ces raisons, les solveurs S&E récents limitent l'apprentissage à la sous-partie courante de l'arbre de recherche (i.e. les changements sont défaits lors des retours-arrières). De plus, l'apprentissage n'est appliqué que pour la transformation des ensembles de clauses qui correspondent à un (ou une combinaison) des motifs suivants (avec au dessus les clauses des motifs et en dessous les clauses obtenues après application de la transformation) :

$$\frac{\{\{l_1, l_2\}, \{l_1, \bar{l}_2\}\}}{\{\{l_1\}\}} \quad (1)$$

$$\frac{\{\{l_1, l_2\}, \{l_1, l_3\}, \{\bar{l}_2, \bar{l}_3\}\}}{\{\{l_1\}, \{l_1, l_2, l_3\}, \{\bar{l}_1, \bar{l}_2, \bar{l}_3\}\}} \quad (2)$$

$$\frac{\{\{l_1\}, \{\bar{l}_1, l_2\}, \{\bar{l}_2, l_3\}, \dots, \{\bar{l}_{k-1}, l_k\}, \{\bar{l}_k\}\}}{\{\square, \{l_1, \bar{l}_2\}, \{l_2, \bar{l}_3\}, \dots, \{l_{k-1}, \bar{l}_k\}\}} \quad (3)$$

Ces motifs ne couvrent pas nécessairement entièrement les SI. En particulier, l'application de la max-résolution sur les clauses des motifs 1 et 2 produit une clause résolvente unitaire. L'apprentissage de telles transformations réduit la redondance dans la détection et la transformation des SI. Les clauses unitaires ainsi produites peuvent également être utilisées dans la sous-partie de l'arbre de recherche pour réaliser davantage de propagations unitaires et donc de détecter potentiellement plus de SI.

#### 4 Apprentissage étendu

Le calcul de la borne inférieure, basé sur l'estimation du nombre de sous-ensembles inconsistants (SI) disjoints, est un composant déterminant pour l'efficacité des solveurs S&E. En effet, il est très coûteux en temps de calcul et la qualité de l'estimation guide les retours-arrières et donc impacte le nombre de nœuds de l'arbre de recherche qui seront explorés. À cet égard, l'apprentissage semble un moyen naturel pour limiter le temps de calcul consacré au calcul de la BI en rendant la détection et le traitement des SI plus incrémental.

Nous donnons dans cette section une définition générale des motifs produisant après transformation des clauses résolvantes unitaires (*unit clause subset*, UCS). Nous motivons cette généralisation et montrons que ces motifs peuvent être détectés efficacement. Nous discutons également de la fréquence d'apparitions de ces motifs dans les SI des instances du benchmark présenté en section 5.

##### 4.1 Unit Clause Subset (UCS)

Nous avons vu dans la section précédente que deux des motifs utilisés dans les schémas d'apprentissage produisent, après transformation, des clauses résolvantes unitaires. L'intérêt de la mémorisation de ces transformations est double. D'un côté, elle permet de limiter la redondance des étapes de propagation unitaire réalisées dans la sous-partie de l'arbre de recherche et d'un autre côté les clauses unitaires produites peuvent permettre de détecter plus de sous-ensembles inconsistants par propagation unitaire, et donc d'améliorer la qualité de la borne inférieure.

Nous proposons dans ce chapitre d'étendre l'apprentissage réalisé par les solveurs en prenant en compte de nou-

veaux motifs produisant, après transformation par max-résolution, des clauses résolvantes unitaires. Ces motifs peuvent être définis formellement comme suit.

**Definition 1 (unit clause subsets, UCS)** *Considérons une formule CNF  $\Phi$ . Un sous-ensemble produisant des clauses résolvantes unitaires est un ensemble  $\{c_1, \dots, c_k\} \subset \Phi$  avec  $\forall i \in \{1, \dots, k\}, |c_i| > 1$  et tel qu'il existe une séquence d'étapes de max-résolution sur les clauses  $c_1, \dots, c_k$  qui produit une clause résolvente unitaire. Nous notons  $k$ -UCS l'ensemble des motifs correspondant aux UCS de taille  $k$ .*

**Exemple 2** *Ci-dessous les motifs de l'ensemble 3-UCS :*

$$\frac{\{\{l_1, l_2\}, \{l_1, l_3\}, \{\bar{l}_2, \bar{l}_3\}\}}{\{\{l_1\}, \{l_1, l_2, l_3\}, \{\bar{l}_1, \bar{l}_2, \bar{l}_3\}\}} \quad (4)$$

$$\frac{\{\{l_1, l_2\}, \{\bar{l}_2, l_3\}, \{l_1, \bar{l}_2, \bar{l}_3\}\}}{\{\{l_1\}, \{\bar{l}_1, \bar{l}_2, l_3\}\}} \quad (5)$$

$$\frac{\{\{l_1, l_2\}, \{l_1, l_3\}, \{l_1, \bar{l}_2, \bar{l}_3\}\}}{\{\{l_1\}, \{l_1, l_2, l_3\}\}} \quad (6)$$

$$\frac{\{\{l_1, l_2\}, \{l_1, \bar{l}_2, l_3\}, \{l_1, \bar{l}_2, \bar{l}_3\}\}}{\{\{l_1\}\}} \quad (7)$$

Les graphes d'implications présentés dans la figure 2 décrivent des exemples d'étapes de propagation qui peuvent mener à la détection de ces motifs.

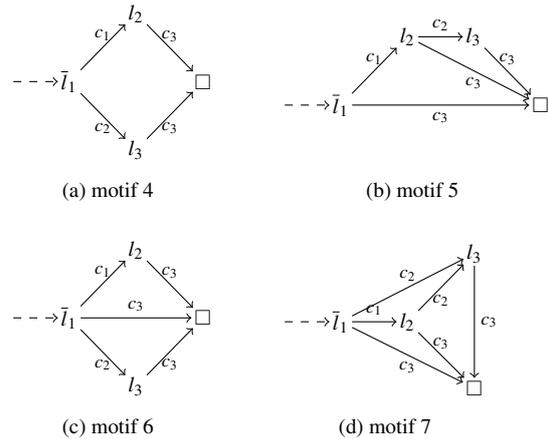


FIGURE 2 – Graphes d'implications décrivant des exemples de séquences de propagations unitaires permettant de détecter les motifs de l'ensemble des 3-UCS présentés dans l'exemple 2. On suppose que le littéral  $\bar{l}_1$  a déjà été propagé et que les clauses de chacun des motifs sont nommées  $c_1, c_2$  et  $c_3$ .

Puisque l'un des objectifs de l'apprentissage des transformations des UCS est d'accroître les affectations réalisées par propagation unitaire (PUS ou LC) lors du calcul de la borne inférieure, nous ne considérons pas les sous-ensembles contenant des clauses unitaires. Dans le meilleur des cas (s'ils contiennent une seule clause unitaire), la transformation de tels sous-ensembles laisserait inchangé le nombre de clauses unitaires présentes dans la formule. Dans le pire des cas (s'ils contiennent plus d'une clause unitaire), la formule transformée contiendrait moins de clauses unitaires que l'originel. Le nombre d'affectations faites par propagation unitaire, et par conséquent le nombre de SI détectés, peuvent alors être réduits.

Nous faisons une distinction entre les motifs des ensembles  $k$ -UCS selon la taille de leurs clauses. Nous notons par  $k^b$ -UCS le sous-ensemble de  $k$ -UCS composé des motifs qui ne contiennent que des clauses binaires et  $k'$ -UCS le sous-ensemble composé des motifs contenant au moins une clause ternaire. Pour tout  $j > 0$ , nous noterons par  $\{k_1, k_2, \dots, k_j\}$ -UCS l'union des ensembles  $k_1$ -UCS,  $k_2$ -UCS,  $\dots$ ,  $k_j$ -UCS.

On peut noter que les motifs 1 et 2 appartiennent respectivement aux ensembles 2-UCS et  $3^b$ -UCS.

#### 4.2 Détection et fréquence d'apparitions des $k$ -UCS

Les motifs des  $k$ -UCS sont facilement détectables lors de l'analyse du graphe d'implications réalisée pour construire les sous-ensembles inconsistants. En effet, les clauses situées entre le conflit et le premier point d'implication unique (*first unit implication point*, FUIP) [16] produisent lorsqu'elles sont transformées par max-résolution une résolution unitaire. Il suffit donc de compter le nombre et la taille de ces clauses pour savoir s'il on est en présence d'un UCS valide. Cela ne change pas la complexité de la procédure d'analyse des conflits et le surcoût en terme de temps de calcul est négligeable.

Nous avons mesuré le taux d'apparitions des motifs  $k$ -UCS dans les sous-ensembles inconsistants détectés par notre solveur AHMAXSAT sur les instances du benchmark présenté en section 5. Les résultats sont présentés dans la table 1. Il est intéressant d'observer qu'en moyenne, les motifs utilisés dans le schéma le plus répandu d'apprentissage (2-UCS et  $3^b$ -UCS) sont présents dans moins de 5% des SI. En moyenne, des UCS sont détectés dans plus 57% des sous-ensembles inconsistants trouvés par AHMAXSAT, ce qui montre que la prise en compte des UCS dans le schéma d'apprentissage peut potentiellement réduire sensiblement les redondances dans le calcul de la borne inférieure.

On peut remarquer que ces valeurs varient fortement selon les classes d'instances. Par exemple, le taux moyen d'apparitions des UCS est inférieur à 3,5% sur les instances

TABLE 1 – Pourcentages d'apparitions des motifs  $k$ -UCS dans les SI détectés par AHMAXSAT.

Classes d'instances		2	$3^b$	$3'$	$4^b$	$4'$	$5^b$	$5'$	> 5	
ms	crafted	bipartite	0,05	0,52	0,03	0,22	0,08	42,19	0,29	42,13
		maxcut	0,06	15,71	0,04	4,15	9,45	5,24	6,73	21,49
		set-covering	-	-	-	-	-	-	-	-
	random	highgirth	1,47	0,79	0,49	0,58	0,8	0,34	1,15	23,54
		max2sat	0,12	3	0,05	20,03	0,59	10,63	2,57	41,98
		max3sat	1,23	3,11	1,45	4,5	4,3	1,92	6,79	38,02
min2sat		0,39	2,76	0,11	16,93	0,29	12,18	0,9	43,2	
pms	crafted	frb	0,11	12,63	0,03	0,02	25,38	2,52	3,05	19,16
		job-shop	-	-	-	-	-	-	-	-
		maxclique	0,09	10,25	0,05	0,03	9,33	3,76	0,95	17,33
		maxone	1,89	0,32	0,56	0,25	0,82	0,15	0,47	5,3
		min-enc/kbtree	0,21	0,74	0,15	2,74	0,45	1,34	1,48	20,41
		pseudo/miplib	0,42	1,17	0,26	0,11	0,67	0,18	5	9,48
		reversi	0,51	2,62	0,37	0,43	2,3	1,45	1,34	29,39
	scheduling	-	-	-	-	-	-	-	-	
	random	min2sat	0,57	0,23	0,33	0,12	0,28	6,66	0,28	60,09
		min3sat	0,18	2,33	0,08	0,15	0,66	11,69	0,33	51,22
		pmax2sat	0,96	8,03	0,14	15,92	1,04	5,88	3,43	29,53
		pmax3sat/hi	1,11	1,9	0,65	3,12	1,78	1,85	3,17	37,25
		auction	0,35	8,73	0,24	0,03	15,82	0,16	3,26	3,79
		CSG	0,16	3,66	0,01	0,05	0,41	>0,01	0,16	1,53
frb		0,11	13,46	0,04	0,03	21,05	3,04	2,69	19,37	
wpms	crafted	min-enc	0,29	0,94	0,47	1,4	0,66	0,3	1,17	16,45
		pseudo/miplib	0,27	0,5	0,8	0,28	2,6	0,12	1,64	35,8
		ramsey	0,29	5,18	0,21	0,07	1,49	0,63	1,82	11,42
		random-net	0,07	1,4	0,2	>0,01	8,48	>0,01	18,1	4,02
		set-covering	0,89	0,27	0,7	0,02	0,58	0,02	0,35	0,6
	wmaxcut	0,08	13,74	0,05	0,44	12,65	6,21	8,56	22,48	
	random	wmax2sat	0,09	3,35	0,03	21,57	0,47	11,33	2,92	40,64
		wmax3sat	0,89	2,02	0,97	3,11	3,31	1,43	5,96	38,7
		wpmax2sat	0,44	5,21	0,08	12,91	0,67	6,76	2,71	32,77
		wpmax3sat/hi	0,81	1,12	0,48	1,88	1,28	1,17	2,38	33,9
Ensemble		0,5	4,43	0,29	6,84	3,24	8,13	2,54	31,09	

de la classe wpms/crafted/set-covering tandis qu'il est supérieur à 85% sur celles de la classe ms/crafted/bipartite. On peut aussi noter que les taux d'apparitions de chaque sous-ensemble des UCS varient également d'une classe d'instances à une autre. Cela met en exergue les différences structurelles entre les instances de ces classes.

### 5 Évaluation empirique de l'apprentissage des UCS

Nous présentons dans cette section une évaluation expérimentale de l'impact sur les performances de notre solveur AHMAXSAT de l'ajout des motifs des  $k$ -UCS à son schéma d'apprentissage. Tous les tests sont effectués sur les instances aléatoires et *crafted* du benchmark de la *Max-SAT Evaluation (MSE) 2014*<sup>1</sup>. Nous n'incluons aucune instance issue de problèmes industriels. Même si les résultats présentés ici peuvent être étendus à ces instances, les solveurs S&E (notre solveur AHMAXSAT y compris, et à l'except-

1. [www.maxsat.udl.cat/14/benchmarks/index.html](http://www.maxsat.udl.cat/14/benchmarks/index.html)

tion de MINIMAXSAT [8]) ne gèrent pas efficacement les instances industrielles de très grande taille<sup>2</sup>.

Nous avons implémenté le schéma d'apprentissage étendu grâce aux UCS dans notre solveur AHMAXSAT<sup>3</sup> [1, 2, 3, 4]. Les tests sont effectués sur des machines équipées de processeurs Intel Xeon 2.4 Ghz, de 24 Gb de mémoire vive et de systèmes d'exploitation GNU/Linux. Le temps de résolution est limité à 1800 secondes et la mémoire à 3.5 Gb pour chaque instance traitée (comme lors des MSE).

En partant d'une variante d'AHMAXSAT utilisant les motifs du schéma d'apprentissage classique ( $\{2, 3^b\}$ -UCS), nous ajoutons successivement les motifs de l'ensemble  $3^l$ -UCS, puis ceux de 4-UCS et ceux de 5-UCS. Des résultats préliminaires (non présentés dans cet article) suggèrent que l'ajout des motifs des ensembles  $k$ -UCS avec  $k > 5$  ont un impact négatif sur les performances du solveur. Toutes les variantes incluent également dans leur schéma d'apprentissage le motif 3 qui n'est pas un UCS. Les résultats sont présentés dans la table 2. Pour chaque variante d'AHMAXSAT, nous présentons le nombre d'instances résolues (colonnes S), le nombre moyen de nœuds explorés (colonnes D), le temps moyen de résolution (colonnes T), le nombre moyen de propagations effectuées à chaque nœud de l'arbre de recherche (colonnes P/D) et le nombre moyen de sous-ensembles inconsistants détectés à chaque nœud de l'arbre de recherche (colonnes  $\square$ /D). Le rôle de ces deux derniers indicateurs est de mesurer la réduction des redondances dans l'application de la propagation unitaire et dans la détection des SI. Ils peuvent également indiquer une dégradation de la précision de l'estimation de la borne inférieure.

**$\{2, 3^l\}$ -UCS vs.  $\{2, 3\}$ -UCS** Comme on pouvait s'y attendre, l'ajout des  $3^l$ -UCS aux motifs utilisés par les solveurs existants (les  $\{2, 3^b\}$ -UCS) ne change pas fondamentalement le comportement d'AHMAXSAT puisque les sous-ensembles inconsistants ne correspondent que très rarement à ces motifs (0,29% en moyenne, cf. table 1). Sur les classes d'instances où le taux d'apparitions des  $3^l$ -UCS n'est pas nul, on peut observer une légère réduction du nombre moyen de propagations et de SI détectés par nœud de l'arbre de recherche (colonnes P/D et  $\square$ /D) qui se traduit par une légère amélioration de la vitesse d'exploration de l'arbre de recherche (non représentée dans la table 2). Le nombre moyen de nœuds explorés et le temps moyen de résolution sont aussi réduits sur ces classes d'instances (colonnes D et T).

Par exemple, sur la classe ms/random/max3sat, qui a le plus fort pourcentage d'apparitions des  $3^l$ -UCS dans les SI (1,45% en moyenne), on peut voir que le nombre moyen de propagations et de SI détectés par nœud sont réduits

respectivement de 1% et 2%. La vitesse moyenne d'exploration de l'arbre de recherche passe de 1306,5 nœuds par secondes à 1326,5 (+1,1%). De plus, le nombre de nœuds explorés est réduit de 5,1%. La combinaison de ces deux effets produit une réduction du temps moyen de résolution de 6,1%.

Ces résultats confirment les bénéfices attendus de l'apprentissage des transformations des UCS :

1. l'augmentation de la vitesse d'exploration de l'arbre de recherche due à la réduction de la redondance dans le calcul de la borne inférieure, et
2. la réduction du nombre de nœuds explorés due à l'amélioration de la qualité de la borne inférieure induite par l'augmentation du nombre de clauses unitaires dans la formule.

On peut noter également que ce second effet masque en partie l'impact du premier dans les statistiques que nous présentons. La réduction de la redondance fait baisser les valeurs des indicateurs P/D et  $\square$ /D tandis qu'à l'inverse, l'augmentation du nombre de clauses unitaires les augmente.

Si l'on considère les résultats sur l'ensemble du benchmark, le nombre moyen de nœuds explorés et le temps moyen de résolution sont réduits respectivement de 0,8% et 1%. Comme nous l'avons dit précédemment, la faible ampleur de ces résultats est à mettre en rapport avec la faible fréquence d'apparitions des motifs de l'ensemble  $3^l$ -UCS dans les SI détectés par AHMAXSAT.

**$\{2, 3\}$ -UCS vs.  $\{2, 3, 4\}$ -UCS** Si l'on ajoute les motifs de l'ensemble 4-UCS au schéma d'apprentissage, les nombres moyens de propagations et de SI détectés par décisions sont réduits significativement (respectivement -5,3% et -17,1%). Comme la quantité d'apprentissage réalisée par le solveur augmente, on peut s'attendre à une légère réduction de ces deux valeurs, mais pas dans de telles proportions. Ces valeurs indiquent probablement une perte dans la capacité de détection des sous-ensembles inconsistants, et donc une réduction de la qualité de la borne inférieure. On peut remarquer que sur les classes d'instances présentant les plus fortes réductions de ces indicateurs, comme ms/random/max2sat ou wpms/random/wmax2sat, le nombre moyen de nœuds explorés augmente fortement (respectivement +472,1% et +210,7%) ce qui vient confirmer notre interprétation. Sur l'ensemble du benchmark, 10 instances de moins sont résolues et le temps moyen de résolution est augmenté de 19,1%.

Il est intéressant d'observer que les résultats varient fortement selon les classes d'instances. La perte de performances ne survient que sur les classes d'instances ayant de forts pourcentages d'apparitions des  $4^b$ -UCS. A l'inverse, sur les classes d'instances ayant de faibles pourcentages d'apparitions des  $4^b$ -UCS, comme pms/crafted/frb ou

2. [www.maxsat.udl.cat/14/results/index.html](http://www.maxsat.udl.cat/14/results/index.html)

3. [www.lsis.org/habetd/Djamal\\_Habet/MaxSAT.html](http://www.lsis.org/habetd/Djamal_Habet/MaxSAT.html)

TABLE 2 – Comparaison de l’impact des schémas d’apprentissage basés sur les motifs  $\{2, 3^b\}$ -UCS et  $\{2, 3\}$ -UCS. Les deux premières colonnes montrent les classes d’instances et le nombre d’instances par classe. Les colonnes S, D, T, P/D et  $\square/D$  donnent respectivement le nombre d’instances résolues, le nombre moyens de nœuds explorés, le temps moyen d’exécution, le nombre moyen de propagations faites par nœud de l’arbre de recherche et le nombre moyen de sous-ensembles inconsistants détectés par nœud de l’arbre de recherche. Les colonnes marquées avec une étoile ne prennent en compte que les instances résolues par toutes les variantes.

Classes d’instances	#	$\{2, 3^b\}$ -UCS					$\{2, 3\}$ -UCS					$\{2, 3, 4\}$ -UCS					$\{2, 3, 4'\}$ -UCS					$\{2, 3, 4', 5\}$ -UCS					$\{2, 3, 4', 5'\}$ -UCS						
		S	D*	T*	P/D*	$\square/D^*$	S	D*	T*	P/D*	$\square/D^*$	S	D*	T*	P/D*	$\square/D^*$	S	D*	T*	P/D*	$\square/D^*$	S	D*	T*	P/D*	$\square/D^*$	S	D*	T*	P/D*	$\square/D^*$		
ms crafted	bipartite	100	100	33658	88,3	1135,3	73,9	100	33726	88,4	1135,4	74	100	33485	87,9	1135,6	74	100	33518	90,7	1135,8	74	98	184518	318,9	687,8	30	100	32328	84	1132,4	73,4	
	maxcut	67	56	241129	63,1	146,1	22	56	241129	63	146,1	22	56	164939	42,1	109,2	15,5	56	167956	48,5	142,9	20,1	56	192076	49	127,8	17,3	56	171380	44,7	135,2	18,3	
	set-covering	10	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-		
	random	highgirth	82	7	4563195	1045,3	43,2	2,5	7	4649532	1064,2	43,1	2,5	7	4636028	1065,9	43,1	2,5	7	4647438	1121,2	43	2,5	6	4680189	1086,2	43,1	2,5	7	4650016	1072,9	43,3	2,5
		max2sat	100	100	30406	62,1	926,9	48,6	100	30484	62,4	927,5	48,7	93	174399	291,3	699,6	31,6	100	29710	62,4	926,3	48,6	90	206837	334,3	681	29,8	100	28292	55,6	890,8	44,4
		max3sat	100	100	372213	284,9	231,6	24,5	100	353221	267,5	229,9	24	100	355812	250,4	213,1	21,1	100	325667	242,3	221,3	22,3	100	367432	249,3	202,4	100	340968	231,2	205,8	20,1	
min2sat		96	96	1067	2,7	1097,5	35,9	96	1067	2,7	1097,9	35,9	96	1259	2,5	843	19,9	96	1065	2,7	1096,3	35,8	96	3689	7	844,6	20,9	96	1046	2,6	1096,1	35,7	
pms crafted	frb	25	5	1364465	333,5	49,8	6,9	5	1364465	333,8	49,8	6,9	5	801715	205,2	47,1	5,2	5	804408	209,3	47	5,2	5	687551	176,4	43,9	4,6	5	585949	154,2	46,1	4,8	
	job-shop	3	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-		
	maxclique	158	133	25474	28,7	88,3	7,5	133	25474	27,8	88,3	7,5	133	22426	30,5	86,2	6,4	133	22340	28,4	86,1	6,4	133	26735	33,9	231,7	5,8	133	23107	30,2	86,1	6,1	
	maxone	140	109	148070	31,2	718,8	23,2	109	148071	31,4	718,6	23,2	107	133744	28,5	717,2	23	109	133749	28,8	718,6	23,1	108	133742	28,3	1105,5	27,4	109	133747	28,7	718,7	23,1	
	min-enc/kbtree	42	34	202579	500,8	1035,3	63,7	34	208641	515,8	1035,1	63,7	34	202738	488,4	1027,5	62,8	34	209900	507,2	1032,2	63,5	35	205645	494	979,4	60,8	34	197998	476,2	1037	63,5	
	pseudo/miplib	4	2	268	0	11,8	1,8	2	268	0	11,8	1,8	2	268	0	11,8	1,8	2	268	0	11,8	1,8	2	288	0	4726,2	17	2	288	0	11,8	1,8	
	reversi	44	8	4679	129,9	2596,2	23,9	8	4679	129,3	2596,2	23,9	8	4719	130	2648,1	24,5	8	4736	128,1	2637,7	24,5	8	4708	132,2	5135,3	34,6	8	4555	129,2	2907,4	27	
	scheduling	5	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-	-	0	-	-	-		
	random	min2sat	60	58	5271	68,7	6055,5	51,9	58	5278	68,6	6055,5	51,9	58	5281	69,4	6056,5	51,9	58	5283	69,2	6056,5	51,9	51	12161	161,9	5658,3	42,1	58	5283	69,4	6056,4	51,9
		min3sat	60	58	67532	181,2	1199,2	27,6	58	67587	181,6	1199,3	27,6	58	66974	180,7	1195,6	27,3	58	66887	186,2	1195,9	27,3	52	187338	451,5	1018,6	23,1	58	66917	181,2	1195,8	27,3
		pmax2sat	60	60	1045	3,5	944,4	74,6	60	1066	3,6	946,5	74,5	60	2781	8,9	863,4	62	60	1037	3,5	931,4	72,4	60	2257	6,8	838,2	58,6	60	1021	3,2	878,4	65,8
pmax3sat/hi		30	30	89897	65	242,4	16,4	30	89236	64,6	242,2	16,4	30	86400	60,9	236,2	15,6	30	86932	62,2	240,4	16,1	30	86893	60,3	229,8	17,4	30	84944	60,1	235,8	15,6	
Ensemble		1776	1438	112580	92,7	1344,7	77,4	1438	111719	91,8	1344,6	77,4	1428	108233	109,3	1273,3	64,2	1438	96202	87,3	1340,7	76,5	1412	126692	144,3	1347,1	58,6	1440	94407	82,7	1328,8	73,3	
wpmms crafted	auktion	40	40	467409	123,5	143,7	66,4	40	467518	124,1	143,4	66,3	40	309981	111,1	127,1	73,9	40	310614	112	126,8	73,7	40	310322	118,4	111,8	72,3	40	310021	115,5	116,7	74,6	
	CSG	10	4	54378	432,4	16,9	0	4	54378	453	16,9	0	4	54648	432	15,4	0	4	54378	445,4	16,9	0	4	54813	431,7	14,2	1,2	4	54813	452,3	16,4	0	
	frb	34	14	492276	128,6	40,4	6	14	492276	125	40,4	6	14	290082	82,4	36,1	4,4	14	291044	78,7	36	4,4	14	248789	71,5	35,4	4,1	14	212369	58,7	36,5	4,3	
	min-enc	74	64	21340	109,1	8805,1	78,2	64	21340	109,2	8805,1	78,2	64	24009	117,8	8561,6	73,5	64	20522	107,6	8805,4	76,8	64	21554	111,6	10838,8	74,8	64	20101	108,8	8826,3	77,1	
	pseudo/miplib	12	4	1500	159,4	2859	162,8	4	1500	159,3	2859	162,8	4	1438	155,8	2602,5	166,4	4	1493	157,6	2906,2	169,2	4	1496	159,5	158,5	32,9	4	1505	158,8	2852,7	166,4	
	ramsey	15	4	158529	44,2	7	2,1	4	158529	44,3	7	2,1	4	158131	44,3	7	2,1	4	158161	44,6	7	2,1	4	157496	45	23,3	7,2	4	157559	44,8	7	2,2	
	random-net	32	1	1253990	1069,5	-	-	1	1253990	1078,6	-	-	1	1563328	1366,2	-	-	1	1563328	1360,2	-	-	3	335161	240	-	-	3	335161	237,5	-	-	
	set-covering	45	25	3754	42,3	453,7	46,8	25	3754	42,5	453,7	46,8	25	3754	42,7	453,7	46,8	25	3754	44,1	453,7	46,8	25	3764	42,9	501,7	51	25	3754	46,4	453,7	46,8	
	wmaxcut	48	46	53152	85,9	597,1	122,5	46	53214	86,6	597,1	122,5	46	28518	51,4	434,5	97,3	46	28679	53,7	599,9	109,6	46	28737	51,1	613,4	99,6	46	26985	49,4	582,4	97,9	
	random	wmax2sat	120	120	4192	48,1	2595,5	279,3	120	4188	47,8	2596,1	279,5	119	13014	131,6	2307,9	182	120	4182	48	2591,6	277,6	118	16188	170,2	2226,1	169,7	120	3810	43,2	2538,3	262
		wmax3sat	40	40	41332	115,2	612,4	106,7	40	40387	112,2	611,2	105,8	40	38980	107	598,2	100,1	40	37939	106,2	605,1	102,7	40	38221	103	579,6	95,3	40	36836	99,5	591,3	97,7
wpmx2sat		90	90	575	5,9	1172,3	209,2	90	573	5,7	1170,7	210	90	841	12	1322,3	209,3	90	575	5,7	1156,1	211,8	90	777	10,5	1270,6	197	90	575	5,5	1144,6	205,5	
wpmx3sat/hi		30	30	31591	88,4	656,6	64,6	30	31464	87,8	656,8	64,5	30	30751	85,6	653,7	63,3	30	31004	87,5	656,2	64,1	30	30705	85,1	660,9	64,4	30	30451	84,1	654	63,4	
Ensemble		1776	1438	112580	92,7	1344,7	77,4	1438	111719	91,8	1344,6	77,4	1428	108233	109,3	1273,3	64,2	1438	96202	87,3	1340,7	76,5	1412	126692	144,3	1347,1	58,6	1440	94407	82,7	1328,8	73,3	

wpms/crafted/frb, le temps moyen de résolution est réduit (-39,5% et -34,1% respectivement). Ces résultats suggèrent que la mémorisation des transformations des SI correspondant aux motifs  $4^b$ -UCS impacte négativement et dans des proportions importantes les performances de notre solveur.

**{2,3}-UCS vs. {2,3,4<sup>t</sup>}-UCS** Si l'on ignore les  $4^b$ -UCS et que l'on ne mémorise que les transformations des SI correspondant aux motifs  $\{2,3,4^t\}$ -UCS, le nombre moyen de nœuds explorés et le temps moyen de résolution sur l'ensemble du benchmark sont réduits respectivement de 13,9% et de 4,9%. On peut observer que les nombres moyens de propagation et de SI détectés par nœud sont légèrement réduits (respectivement -0,3% et -1,2%).

**{2,3,4<sup>t</sup>}-UCS vs. {2,3,4<sup>t</sup>,5}-UCS** Lorsqu'on ajoute les motifs 5-UCS au schéma d'apprentissage d'AHMAXSAT, on observe le même comportement qu'avec les motifs 4-UCS. Sur l'ensemble du benchmark, le nombre moyen de SI détectés par nœud est réduit significativement (-23,3%). En conséquence, le nombre moyen de nœuds explorés augmente (+31,7%), 26 instances de moins sont résolues et le temps moyen de résolution est plus élevé (+65,3%). Comme précédemment, la perte de performances est particulièrement élevée sur les classes d'instances ayant un fort pourcentage d'apparitions des motifs  $5^t$ -UCS dans les SI, comme les ms/crafted/bipartite, ms/random/max2sat, pms/random/min3sat ou wpms/random/wmax2sat où l'augmentation du temps moyen de résolution est respectivement de 251,5%, 435,7%, 142,5% et 254,6%.

**{2,3,4<sup>t</sup>}-UCS vs. {2,3,4<sup>t</sup>,5<sup>t</sup>}-UCS** Comme précédemment, si l'on ignore les motifs  $5^b$ -UCS et qu'on ne mémorise que les motifs  $\{2,3,4^t,5^t\}$ -UCS, 2 instances de plus sont résolues, le nombre moyen de nœuds et le temps moyen de résolution sont réduits respectivement de 1,9% et 5,3%.

**Bilan** L'ajout des motifs  $\{3^t,4^t,5^t\}$ -UCS au schéma d'apprentissage de notre solveur AHMAXSAT permet de résoudre 2 instances de plus. Le nombre moyen de nœuds explorés et le temps moyen de résolution sont réduits de 16,2% et 11,8% respectivement. Les figures 3a et 3b comparent instance par instance respectivement le temps de résolution et le nombre de nœuds explorés d'AHMAXSAT $\{2,3^b\}$ -UCS et d'AHMAXSAT $\{2,3,4^t,5^t\}$ -UCS. On peut observer que le gain est régulièrement réparti sur l'ensemble du benchmark.

Enfin, la figure 4 compare les temps de résolutions cumulés de AHMAXSAT $\{2,3^b\}$ -UCS et AHMAXSAT $\{2,3,4^t,5^t\}$ -UCS avec ceux de trois des solveurs S&E les plus performants : AKMAXSAT [9] et deux variantes de WMAXSATZ [15, 12]. On peut voir que le gain

apporté par l'ajout des motifs  $\{3^t,4^t,5^t\}$ -UCS au schéma d'apprentissage d'AHMAXSAT est significatif. Sur les instances considérées, les deux variantes de notre solveur sont nettement supérieures aux autres solveurs S&E de l'état de l'art. Ces résultats sont conformes à ceux de la dernière *Max-SAT Evaluation*, où AHMAXSAT (avec l'apprentissage étendu) a été classé premier dans trois des neuf catégories de l'évaluation.

**Discussion** Il est communément admis que l'apprentissage de la transformation des SI par max-résolution peut dans certains cas réduire la qualité de l'estimation de la BI. Cependant, à notre connaissance, les causes de ce comportement n'ont jamais été décrites clairement. L'étude empirique que nous avons réalisée montre que l'apprentissage de la transformation des clauses relatives à certains motifs (les  $4^b$ -UCS et les  $5^b$ -UCS) semble particulièrement affecté par ce phénomène. De plus, les statistiques détaillées présentées précédemment montrent une réduction du nombre de propagations et de SI détectés à chaque nœud de l'arbre de recherche et donc une augmentation du nombre de nœuds explorés. En effet, si le nombre d'étapes de propagation est réduit, alors moins de SI seront détectés et l'estimation de la BI sera moins précise. Par conséquent, les retours-arrières seront effectués plus bas dans l'arbre de recherche et davantage de nœuds seront explorés. Nous illustrons dans l'exemple ci-dessous l'impact de la transformation des clauses d'un motif  $4^b$ -UCS sur la propagation unitaire.

**Exemple 3** Considérons le sous-ensemble  $\Phi'' = \{c_2, c_3, c_4, c_5\}$  de la formule  $\Phi$  introduite dans l'exemple 1. Si l'on ajoute la clause  $c_{14} = \{\bar{x}_1, x_4\}$  à  $\Phi''$ , il y a deux UCS possibles :  $\psi_1 = \{c_3, c_5, c_{14}\}$  et  $\psi_2 = \{c_2, c_3, c_4, c_5\}$  qui sont respectivement un  $3^b$ -UCS et un  $4^b$ -UCS.

Si  $\psi_1$  est transformé par max-résolution, on obtient la formule  $\Phi^{(3)} = \{c_2, c_4, c_{15}, c_{16}, c_{17}\}$  avec  $c_{15} = \{\bar{x}_1\}$ ,  $c_{16} = \{\bar{x}_1, x_3, x_4\}$  et  $c_{17} = \{x_1, \bar{x}_3, \bar{x}_4\}$ . Dans  $\Phi^{(3)}$ , l'affectation de  $x_1$  à vrai conduit à la séquence de propagations  $\langle x_2 @ c_2, x_4 @ c_4 \rangle$ . La clause  $c_{15}$  est falsifiée tandis que  $c_{16}$  et  $c_{17}$  sont satisfaites.

Si  $\psi_2$  est transformé par max-résolution, on obtient la formule  $\Phi^{(4)} = \{c_{12}, c_7, c_8, c_{10}, c_{11}, c_{14}\}$  (cette transformation est décrite dans l'exemple 1). L'affectation de  $x_1$  à vrai conduit, dans  $\Phi^{(4)}$ , à la séquence de propagations unitaires  $\langle x_4 @ c_{14} \rangle$ . La clause  $c_{12}$  est falsifiée,  $c_8$  et  $c_{10}$  sont satisfaites et  $c_7$  et  $c_{11}$  sont réduites. Il n'y a plus de clauses unitaires et  $x_2$  ne peut pas être propagée. Il est intéressant d'observer que les deux clauses réduites  $c_7 = \{x_2, \bar{x}_3\}$  et  $c_{11} = \{x_2, x_3\}$  pourraient mener à la propagation de  $x_2$  si l'on appliquait la max-résolution entre elles sur la variable  $x_3$ . Mais la propagation unitaire seule ne permet pas de propager  $x_2$ .

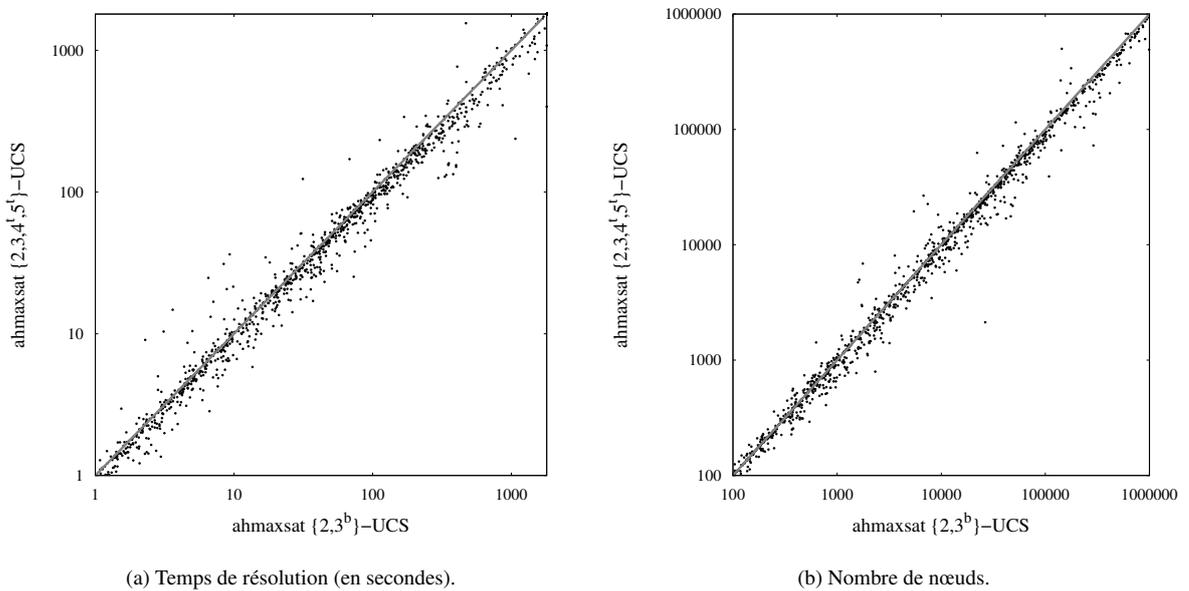


FIGURE 3 – Comparaison du nombre de nœuds explorés et du temps de résolution d’AHMAXSAT avec les ensembles de motifs  $\{2, 3^b\}$ -UCS et  $\{2, 3, 4^t, 5^t\}$ -UCS. Chaque point représente une instance. Tous les axes sont en échelle logarithmique.

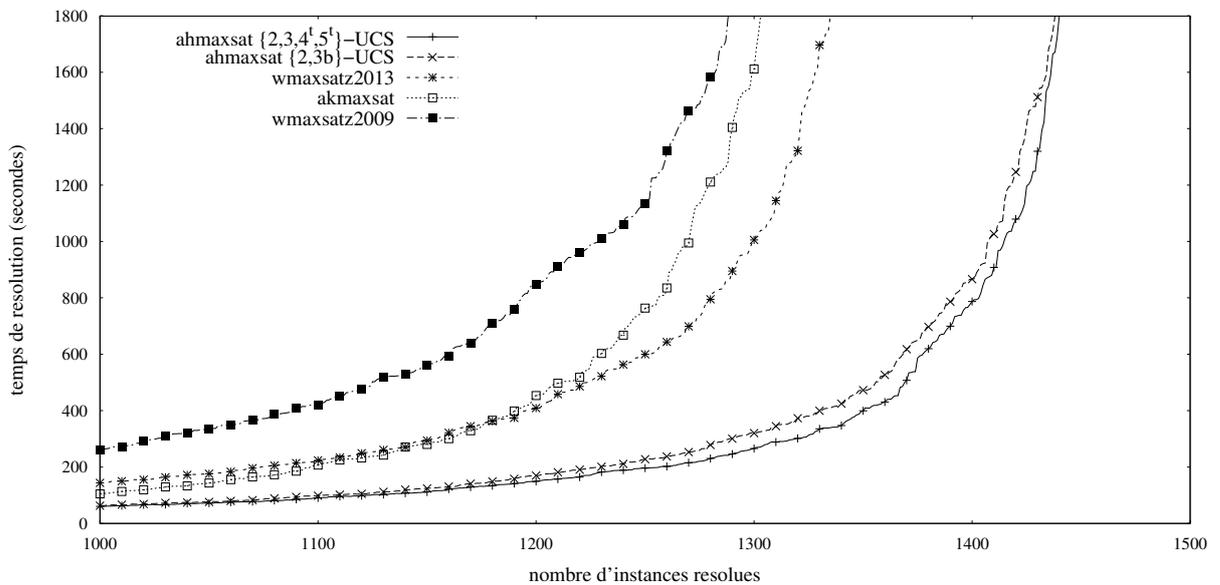


FIGURE 4 – Comparaison des variantes d’AHMAXSAT, avec en abscisse le nombre d’instances résolues et en ordonnée le temps cumulé de résolution.

Même si nous avons décrit ici au travers d'un exemple comment les transformations peuvent réduire le nombre de variables propagées et l'impact que ça a sur la qualité de l'estimation de la BI, les caractéristiques précises des sous-ensembles de clauses concernés par ce phénomène sont méconnues. Une étude approfondie de ces caractéristiques serait d'un grand intérêt dans la perspective d'améliorer les schémas d'apprentissages utilisés par les solveurs S&E.

## 6 Conclusion

Nous avons présenté dans ce papier de nouveaux ensembles de motifs permettant d'enrichir le schéma d'apprentissage des solveurs de type S&E. L'étude expérimentale que nous avons réalisé montre que l'apprentissage des transformations de certains de ces motifs (les  $\{2, 3, 4', 5'\}$ -UCS) réduit sensiblement le nombre de nœuds de l'arbre de recherche explorés et le temps de résolution de notre solveur AHMAXSAT.

Les résultats expérimentaux obtenus montrent également que l'apprentissage des transformations des  $\{4^b, 5^b\}$ -UCS réduit la capacité de notre solveur à détecter les sous-ensembles inconsistants, et par là même ses performances. Nous avons décrit ce phénomène, ce qui à notre connaissance n'avait jamais été fait auparavant. Nous étudierons ce phénomène dans le future avec pour objectif d'établir les fondements théoriques de l'apprentissage dans les solveurs S&E pour Max-SAT.

## Références

- [1] André Abramé and Djamel Habet. Efficient application of max-sat resolution on inconsistent subsets. In Barry O'Sullivan, editor, *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming (CP 2014)*, volume 8656 of *Lecture Notes in Computer Science*, pages 92–107. Springer International Publishing, 2014.
- [2] André Abramé and Djamel Habet. Local max-resolution in branch and bound solvers for max-sat. In *Proceedings of the 26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2014)*, pages 336–343. IEEE Computer Society, 2014.
- [3] André Abramé and Djamel Habet. Maintaining and handling all unit propagation reasons in exact max-sat solvers. In Stefan Edelkamp and Roman Barták, editors, *Proceedings of the 7th Annual Symposium on Combinatorial Search (SOCS 2014)*, pages 2–9. AAAI Press, 2014.
- [4] André Abramé and Djamel Habet. On the extension of learning for max-sat. In Ulle Endriss and João Leite, editors, *Proceedings of the 7th European Starting AI Researcher Symposium (STAIRS 2014)*, volume 241 of *Frontiers in Artificial Intelligence and Applications*, pages 1–10. IOS Press, 2014.
- [5] María Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution for max-sat. *Artificial Intelligence*, 171(8-9) :606–618, 2007.
- [6] Federico Heras and Javier Larrosa. New inference rules for efficient max-sat solving. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, pages 68–73. AAAI Press, 2006.
- [7] Federico Heras, Javier Larrosa, and Albert Oliveras. Minimaxsat : A new weighted max-sat solver. In João Marques-Silva and Karem Sakallah, editors, *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT 2007)*, volume 4501 of *Lecture Notes in Computer Science*, pages 41–55. Springer Berlin / Heidelberg, 2007.
- [8] Federico Heras, Javier Larrosa, and Albert Oliveras. Minimaxsat : An efficient weighted max-sat solver. *Journal of Artificial Intelligence Research*, 31 :1–32, 2008.
- [9] Adrian Kügel. Improved exact solver for the weighted max-sat problem. In Daniel Le Berre, editor, *Proceedings of the 1st Pragmatics of SAT Workshop (POS 2010)*, volume 8 of *EasyChair Proceedings in Computing*, pages 15–27. EasyChair, 2010.
- [10] Javier Larrosa and Federico Heras. Resolution in max-sat and its relation to local consistency in weighted csp. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 193–198. Professional Book Center, 2005.
- [11] Javier Larrosa, Federico Heras, and Simon de Givry. A logical approach to efficient max-sat solving. *Artificial Intelligence*, 172(2-3) :204–233, 2008.
- [12] Chu Min Li, Felip Manyà, Nouredine Ould Mohamedou, and Jordi Planes. Resolution-based lower bounds in maxsat. *Constraints*, 15(4) :456–484, 2010.
- [13] Chu Min Li, Felip Manyà, and Jordi Planes. Exploiting unit propagation to compute lower bounds in branch and bound max-sat solvers. In Peter van Beek, editor, *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP 2005)*, volume 3709 of *Lecture Notes in Computer Science*, pages 403–414. Springer Berlin / Heidelberg, 2005.
- [14] Chu Min Li, Felip Manyà, and Jordi Planes. Detecting disjoint inconsistent subformulas for computing lower bounds for max-sat. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, pages 86–91. AAAI Press, 2006.
- [15] Chu Min Li, Felip Manyà, and Jordi Planes. New inference rules for max-sat. *Journal of Artificial Intelligence Research*, 30 :321–359, 2007.
- [16] João Marques-Silva and Karem A. Sakallah. Grasp : A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5) :506–521, August 1999.