Complexité et approximation de MAX (k-)SAT

Benjamin Hellouin

Définition 1. *Les problèmes MAX (k-)SAT sont définis ainsi :*
Entrée : un ensemble de formules en FNC (sur au plus k littéraux).
Sortie : une valuation satisfaisant le maximum de formules.

1 Résultats de complexité

Théorème 1. *MAX SAT et les MAX k-SAT, $k \geq 2$ sont NP-difficiles.*

$$\begin{aligned} \text{SAT} &\leq_T \text{MAX SAT} \\ \text{k-SAT} &\leq_T \text{MAX k-SAT} \end{aligned}$$

nous permettent de conclure pour $k \geq 3$.

Proposition 1. *3-SAT $\leq_T$ MAX 2-SAT*

Soit $\Phi = \{C_1 \dots C_m\}$ une instance de 3-SAT contenant les variables $x_1 \dots x_n$. Si $C_i = a_i \vee b_i \vee c_i$, on définit :

$$C_i' = \{a_i, b_i, c_i, y_i, (\overline{a_i} \vee \overline{b_i}), (\overline{a_i} \vee \overline{c_i}), (\overline{b_i} \vee \overline{c_i}), (a_i \vee \overline{y_i}), (b_i \vee \overline{y_i}), (c_i \vee \overline{y_i})\}$$

On remarque que :

- Si $\delta(a_i) = \delta b_i = \delta c_i = 0$, au plus six clauses de C_i' peuvent être satisfaites.
- Sinon, sept clauses et pas plus peuvent être satisfaites.

On en déduit que, si $\Phi' = \bigcup_{i=1}^{m} C_i'$, alors Φ est satisfaisable si et seulement si au moins $7m$ clauses de Φ' sont satisfaisables.

2 Approximation de MAX SAT

Théorème 2. *MAX SAT et les MAX k-SAT sont $\frac{1}{2}$ approximables en temps polynômial. Autrement dit, il existe un algorithme en temps polynômial qui, étant donné une instance Φ pour laquelle une valuation optimale satisfait m_0 clauses, renvoie une valuation satisfaisant au moins $\lfloor \frac{m_0}{2} \rfloor$ clauses.*

On commence par définir la fonction de poids w qui à toute clause c associe $w(c) = \frac{1}{2^{|c|}}$ (1 si la clause est vide). On présente maintenant un algorithme qui est une $\frac{1}{2}$-approximation de MAX SAT et des MAX k-SAT.

Algorithme de Johnson :

Entrée : un ensemble de clauses C sur les variables V

Pour $v \in V$ **faire**

$P \leftarrow \{c \in C | v \text{ apparaît dans } C\}$

$N \leftarrow \{c \in C | \overline{v} \text{ apparaît dans } C\}$

Si $\sum_{c \in P} w(c) \geq \sum_{c \in N} w(c)$ **alors**

$\delta(v) \leftarrow 1$

$C \leftarrow \{c \backslash \{v\} | c \in C \backslash P\}$

Sinon

$\delta(v) \leftarrow 0$ et de même en inversant P et N

Sortie : la valuation δ

Correction. Soit C_0 et C_1 les états de la variable C avant et après une étape de la boucle. On sépare C_0 en P, N et $C_0 \backslash P \cup N$. Si on suppose que la variable a été mise à 1, les clauses de P one été satisfaites et les clauses de N ont été raccourcies d'une variable. Alors

$$\sum_{C_1} w(c) = 0 + 2 \sum_{N} w(c) + \sum_{C_0 \backslash P \cup N} w(c) \leq \sum_{P} w(c) + \sum_{N} w(c) + \sum_{C_0 \backslash P \cup N} w(c)$$

L'autre cas étant symétrique, $w(C)$ est décroissante à chaque étape de l'algorithme. Si, initialement, $C = \{c_1 \dots c_m\}$, alors $w_{init}(C) = \sum_{i=1}^{m} \frac{1}{2^{|c_i|}}$. Si on note C' les clauses non satisfaites à la fin de l'exécution de l'algorirthme, on a $w(C') = |C'| \leq w_{init}(C) \leq \frac{m}{2}$. La moitié des clauses sont satisfaites, et a fortiori il s'agit d'une $\frac{1}{2}$-approximation.