TD 4

Exercice 1. Max-Sat via *la programmation linéaire*

Le problème Max-Sat prend en entrée une formule booléenne ϕ sous forme CNF : ϕ est une conjonction de clauses, chaque clause étant une disjonction de littéraux. Par exemple, on peut avoir la formule $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_4)$. L'objectif est de trouver une affectation des variables qui satisfait le plus de clauses possibles.

On réécrit le problème Max-Sat sous forme d'équation. On associe à chaque variable booléenne x_i la variable entière $v_i \in \{0,1\}$. Une clause C_j de la formule est transformée en une somme $S_j = \sum_{x_i \in C_j} v_i + \sum_{\neg x_i \in C_j}(1 - v_i)$. Par exemple, la formule précédente est transformée en deux sommes $v_1 + (1 - v_2) + v_3$ et $(1 - v_1) + v_4$.

1. On définit une bijection entre les affectations des variables booléennes x_i et les variables entières v_i par $v_i = 1$ si et seulement si x_i est VRAI. Montrer que C_j est satisfaite par une affectation booléenne des x_i si et seulement si $S_j \geq 1$ pour l'affectation correspondante des v_i.
2. On associe à chaque clause C_j une nouvelle variable $z_j \in \{0,1\}$. Montrer que Max-Sat est équivalent à maximiser la somme $\sum_j z_j$, sous la contrainte que $S_j \geq z_j$ pour tout j.

On relâche les contraintes sur les variables aléatoires : on demande maintenant simplement que les v_i et z_j soient des réels entre 0 et 1. On admet qu'on peut alors résoudre le problème de la question précédente en temps polynomial. Dans la suite, on note $v_i^\star$ et $z_j^\star$ les solutions réelles (optimales) de ce problème.

3. Montrer que pour toute formule, le nombre maximal de clauses satisfiables par une affectation des variables est borné par $\sum_j z_j^\star$.
4. Proposer une manière d'arrondir de manière probabiliste le résultat pour revenir à des variables booléennes.
5. Soit $\beta_k = 1 - (1 - 1/k)^k$. Montrer qu'avec l'arrondi proposé, une clause C_j ayant k littéraux est satisfaite avec probabilité $\geq \beta_k z_j^\star$. *On admettra que* $1 - (1 - z/k)^k \geq \beta_k z$ *pour tout* $z \in [0,1]$.
6. Montrer que $\beta_k \geq 1 - 1/e$ pour tout $k \geq 1$.
7. En déduire un algorithme de type Las Vegas fournissant une $(1 - 1/e)$-approximation de Max-Sat en temps polynomial espéré.