Devoir à la maison (pour le 6 novembre 2017)

Chaque réponse doit être soigneusement justifiée. La qualité de la rédaction est un critère primordial dans la notation.

Défintions.
L'objectif de ce devoir est de donner des algorithmes (probabilistes) d'approximation pour le problème Max-Sat et ses variantes Max-k-Sat pour $k \geq 1$.
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.
Le problème Max-k-Sat est la restriction du problème Max-Sat dans laquelle chaque clause doit avoir exactement k-littéraux. Un exemple d'entrée pour Max-3-Sat est la formule $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$.
Un algorithme est une *α-approximation de* Max-Sat (ou Max-k-Sat) si pour toute entrée ϕ, l'algorithme renvoie une affectation des variables qui satisfait αn_{opt} clauses, où n_{opt} est le nombre maximal de clauses satisfiables simultanément dans ϕ. Le réel α est appelé le *facteur d'approximation* de l'algorithme.

Exercice 1. *Approximation simple de* Max-3-Sat

Dans cet exercice, on considère le problème Max-3-Sat.

1. Supposons qu'on tire aléatoirement des valeurs pour les variables x_i. Pour toute clause de la formule, avec quelle probabilité est-elle satisfaite ?
2. Donner l'espérance du nombre de clauses satisfaites lors d'une affectation aléatoire des variables.
3. En déduire un algorithme de type Las Vegas qui satisfait une fraction constante (à déterminer) des clauses avec une espérance de temps de calcul polynomiale.
4. Que peut-on dire du facteur d'approximation de l'algorithme précédent ?
5. Généraliser le résultat au problème Max-k-Sat dans lequel chaque clause contient k littéraux.

Exercice 2. Max-Sat via *la programmation linéaire*

Dans cet exercice, on considère le problème Max-Sat. On le réécrit sous forme d'équations. 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.
8. Que peut-on dire du facteur d'approximation de l'algorithme précédent lorsqu'on l'applique à Max-k- !Sat, en fonction de k ?

Exercice 3. *Le meilleur des deux mondes*

L'objectif de cet exercice est d'utiliser les deux algorithmes précédents pour obtenir un meilleur facteur d'approximation.

1. On considère le problème Max-k-Sat, pour $1 \leq k \leq 5$. Donner les valeurs (approchées) des facteurs d'approximation des algorithmes des deux exercices précédents, pour chacune de ces 5 valeurs de k. Pour quelle valeur les deux algorithmes sont équivalents ?

On s'intéresse maintenant au problème général Max-Sat, dans lequel les clauses ont un nombre quelconque de littéraux. Pour obtenir le meilleur des deux mondes, on tire aléatoirement un bit et on utilise le premier algorithme si $b = 0$ et le deuxième sinon.

Soit n_1 l'espérance du nombre de clauses satisfaites par le premier algorithme et n_2 l'espérance du nombre de clauses satisfaites par le second. On note S_k l'ensemble des clauses de taille k de la formule ϕ en entrée de l'agorithme. On garde les notations de l'exercice précédent, les $z_j^\star$ désignant une solution optimale du problème relaché.

2. Montrer que $n_1 = \displaystyle\sum_{k=1}^{\infty} \sum_{C_j \in S_k} (1 - 1/2^k)$.
3. Montrer que $n_2 \geq \displaystyle\sum_{k=1}^{\infty} \sum_{C_j \in S_k} \beta_j z_j^\star$.
4. Montrer que pour tout k, $(1 - 2^{-k}) + \beta_k \geq 3/2$.
5. En déduire que $n_1 + n_2 \geq \frac{3}{2} \sum_j z_j^\star$.
6. Conclure que l'algorithme proposé dans cet exercice est un algorithme probabiliste d'approximation de Max-Sat. Quel est son facteur d'approximation ? Donner les garanties de l'algorithme : type Monte Carlo ou Las Vegas, complexité, probabilité de succès, etc.
7. On considère une variante de l'algorithme dans laquelle on applique les deux algorithmes sur l'entrée et on garde la meilleure des deux réponses (l'affectation qui satisfait le plus de clauses). Que peut-on dire de ce nouvel algorithme ?