COMPLEX - Partie 2 — Énoncé

2022-2023

Nom ou numéro d'anonymat :

Les deux exercices sont indépendants.
Une rédaction claire et concise sera appréciée. Toute affirmation devra être justifiée.
Une question non résolue n'empêche pas de faire les suivantes
(dans ce cas indiquez clairement que vous admettez le(s) résultat(s) de la question non faite).

Exercice 1 : Algorithme de Hirsch pour MAX-3-SAT

Algorithme 1 : Algorithme de Hirsch pour MAX-3-SAT

Entrée : Formule Φ de m clauses $C_1, \dots, C_m$ en n variables $x_1, \dots, x_n$; $T \in \mathbb{N}$
Sortie : $(y, \mu) \in \{0,1\}^n \times \{0, \dots, m\}$ tel que $\Phi(y)$ satisfait μ clauses

$\mu^\star \leftarrow -1$; $y^\star \leftarrow (0, \dots, 0)$ ▷ `solution optimale`
pour j *de* 1 *à* T **faire**
  $y = (y_1, \dots, y_n) \xleftarrow{} \{0,1\}^n$
  $\mu \leftarrow \#\{i \in \{1, \dots, m\} | C_i(y) = 1\}$ ▷ `nombre de clauses satisfaites par` y
  si $\mu > \mu^\star$ **alors**
    $\mu^\star \leftarrow \mu$; $y^\star \leftarrow y$ ▷ `mise à jour de la solution optimale`
  pour i *de* 1 *à* n **faire**
    si $\Phi(y) = 0$ **alors**
      considérer une clause C_t (arbitraire) non satisfaite de Φ
      $j \xleftarrow{} \text{VARIABLES}(C_t)$ ▷ `tirage d'une variable dans` C_t
      $y_j \leftarrow 1 - y_j$ ▷ `changement de la valeur de` y_j
      $\mu \leftarrow \#\{i \in \{1, \dots, m\} | C_i(y) = 1\}$ ▷ `nombre de clauses satisfaites par` y
      si $\mu > \mu^\star$ **alors**
        $\mu^\star \leftarrow \mu$; $y^\star \leftarrow y$ ▷ `mise à jour de la solution optimale`
    sinon
      retourner $(y^\star, \mu^\star)$
retourner $(y^\star, \mu^\star)$

Dans cet exercice, nous considérons un algorithme probabiliste du à E. A. Hirsch pour résoudre le problème MAX-3-SAT de façon approchée. L'algorithme est inspiré de l'algorithme de Schöning pour le problème 3-SAT vu en cours.

Soit Φ une formule booléenne en forme normale conjonctive formée de m clauses contenant toutes au plus 3 littéraux. Soit $\ell \leq m$ le nombre maximal de clauses simultanément satisfiables pour Φ et soit $z = (z_1, \dots, z_n)$ une assignation (optimale) des variables telle que $\Phi(z_1, \dots, z_n)$ satisfait ℓ clauses.

1.a] Montrer que $\ell \geq m/2$.

Soit $\epsilon > 0$. Nous disons qu'une assignation des n variables de Φ est *admissible* si elle satisfait au moins $(1-\varepsilon)\ell$ clauses. Notons que si lors de l'exécution de l'algorithme, nous trouvons une assignation des n variables qui est admissible, alors l'algorithme retourne une assignation admissible.

1.b] Considérons une assignation qui ne satisfait pas strictement plus de $m-(1-\varepsilon)\ell$ clauses de Φ dans une itération de la boucle **pour** … **faire** des lignes 7 à 16. Notons probabilité p_ε que l'algorithme change la valeur d'une variable qui a une valeur différente dans cette assignation et dans z. Montrer que

$$p_\varepsilon \geq \frac{\varepsilon\ell}{3(m-(1-\varepsilon)\ell)}$$

1.c] En déduire que

$$p_\varepsilon \geq \frac{\varepsilon}{3(1+\varepsilon)}$$

1.d] Supposons que, lors de l'initialisation de la boucle **pour** … **faire** des lignes 2 à 16, l'assignation y diffère de celle de z en exactement t variables (ce qui se produit avec probabilité $\binom{n}{t}/2^{-n}$). Montrer que pour une telle initialisation, l'algorithme obtient une assignation admissible avec probabilité supérieure ou égale à p_ε^t.

1.e] En sommant pour toutes les valeurs possibles de t, montrer que lors de chaque itération de la boucle **pour** ... **faire** des lignes 2 à 16, l'algorithme obtient une assignation admissible avec probabilité supérieure ou égale à $q = \left(\frac{1}{2}\left(1 + \frac{\varepsilon}{3(1+\varepsilon)}\right)\right)^n$.

1.f] Montrer que pour $T = 1/q$, l'algorithme obtient une assignation admissible avec probabilité supérieure ou égale à $\exp(-1) \simeq 0.368$ (et dans ce cas $T = c^n$ avec $c = (2 - 2\varepsilon/(3 + 4\varepsilon)) < 2$).

Exercice 2 : Algorithme de Karger en contractant les sommets

Nous considérons une variante de l'algorithme de Karger où à chaque étape, au lieu de choisir une arête aléatoire, nous choisissons deux sommets au hasard, identifions ces deux sommets en un seul et supprimons les “boucles” qui apparaissent lors de l'identification.

2.a] Donner une famille infinie de graphes pour lesquels la probabilité que cet algorithme modifié trouve une coupe minimale décroît exponentiellement avec le nombre de sommets du graphe.

Exercice 3 :

Soit $\mathcal{M}$ une machine de Turing probabiliste polynomiale. Nous disons que $\mathcal{M}$ a une probabilité d'erreur au plus $1/3$ si et seulement si pour tout mot $x \in \Sigma^*$, nous avons

$$\Pr(\mathcal{M}(x) \text{ accepte}) \leq 1/3 \text{ ou } \Pr(\mathcal{M}(x) \text{ accepte}) \geq 2/3$$

3.a] Montrer qu'il est indécidable de savoir si une machine de Turing probabiliste polynomiale (dont le temps d'exécution est connu) a une probabilité d'erreur au plus $1/3$.

Indication : On pourra construire une réduction au problème de l'arrêt pour les machines de Turing déterministes.