MIM2 - Algorithmes d'approximation- TD 6
Emmanuelle Lebhar
elebhar@ens-lyon.fr - Bureau 317

Méthode de l'espérance conditionnelle
27 Octobre 2003

Exercice 1 (Sur l'algorithme MAX-CUT du cours)

1. On peut poursuivre la dérandomisation de l'algorithme du cours en remarquant que les sommets sont traités dans un ordre abitraire. Complétez-la pour obtenir un algorithme entièrment déterministe.
2. Proposez une instance critique pour chacun des trois algorithmes MAX-CUT : l'algorithme randomisé, le dérandomisé et le totalement déterministe.

Exercice 2 (k-coupes)
Considérons la généralisation suivante du problème de la coupe maximum.

Problème MAX k-CUT : Étant donnés un graphe non-orienté $G = (V, E)$ muni d'une fonction de coût positive sur les arêtes, et un entier k, trouver une partition de V en k sous-ensembles $S_1, \ldots, S_k$ tels que le coût total des arêtes entre ces ensembles soit maximal.

1. Donnez un algorithme randomisé pour ce problème qui soit une $(1-\frac{1}{k})$-approximation.
2. L'analyse de votre algorithme est-elle au plus près ?
3. Dérandomisez votre algorithme.

Exercice 3 (Satisfaction maximum : MAX-SAT)

Problème MAX-SAT : (en anglais, *Maximum satisfiability problem*) Étant donné une formule f sous forme normale conjonctive, de variables Booléennes x_1,..., x_n, et où un poids positif w_c est attribué à chaque clause c, trouver une instanciation des variables Booléennes qui maximise le poids total des clauses satisfaites.

Notons $\mathcal{C}$ l'ensemble des clauses de f, i.e., $f = \bigvee_{c \in \mathcal{C}} c$. Chaque clause est une disjonction de littéraux ; chaque littéral est une des variables Booléennes, ou sa négation. Notons taille(c) la *taille* d'une clause c, i.e., le nombre de ses littéraux. Nous considérons que les tailles des clauses de f sont arbitraires. Les clauses sont supposées non-triviales (e.g., $c = x \vee y \vee \overline{x}$), et les littéraux non-redondants à l'intérieur d'une même clause.

Pour tout entier positif k, nous appelons MAX-kSAT la restriction de MAX-SAT aux instances dont les clauses sont toutes de tailles $\leq k$. MAX-SAT est NP-difficile ; en fait, même MAX-2SAT est NP-difficile (contrairement à 2SAT qui est dans P).

1. Proposez un algorithme randomisé trivial pour le problème MAX-SAT.
2. Estimez le facteur d'approximation de votre algorithme pour le cas où toutes les clauses sont de taille k (MAX-kSAT). Que dire du cas général ?
3. Dérandomisez votre algorithme.