IFT800 - Série d'exercices #3

Manuel Lafond

Algorithmes probabilistes

Exercice 1. Considérez le problème MAX-k-SAT dans lequel chaque clause a k variable ou plus. Quel ratio d'approximation probabiliste pouvez-vous atteindre? Et quel ratio d'approximation déterministe pouvez-vous atteindre?

Exercice 2. Considérez le problème MAX-k-CUT, où il faut séparer V en k ensembles et maximiser le nombre d'arêtes traversantes. Quel ratio d'approximation probabiliste pouvez-vous atteindre?

Exercice 3. Considérez l'algorithme probabiliste suivant pour VERTEX-COVER. On construit notre ensemble X itérativement. Pour chaque arête $uv \in E$, on inclut u dans X avec probabilité $1/2$ s'il n'y est pas déjà, ou bien v dans X avec probabilité $1/2$ s'il n'y est pas déjà.

Il n'est pas évident d'analyser $\mathbb{E}[APP]$ par rapport à OPT.

On va simplifier et supposer que dans G, chaque sommet a exactement 2 voisins. Dans ce cas, que vaut $\mathbb{E}[APP]$? Et quel est le ratio d'approximation probabiliste? Ensuite, posez les même questions, mais généralisé au cas où chaque sommet a exactement k voisins.

LP et ILP

Exercice 4. Cet exercice sert à entraîner nos conversions problème-ILP. Le but n'est pas de développer un algorithme d'approximation.

1. Exprimez le problème de trouver un matching de poids maximum avec un ILP (le poids d'un matching est la somme des poids des arêtes dans le matching).

2. Exprimez le problème du sac à dos avec un ILP (voir notes de cours, le sac à dos est dans la table des matières).

3. Revenons au problème de supprimer tous les triangles d'un graphe en supprimant un minimum d'arêtes. Exprimez ce problème avec un ILP.

4. Dans le problème CLUSTER-DELETION, on reçoit un graphe G et on doit supprimer un minimum d'arêtes afin que chaque composante connexe de G soit une clique. Exprimez ce problème avec un ILP.

   Notez que dans un graphe, chaque composante connexe est une clique si et seulement si $\forall u, v, w$, l'affirmation $uv \in E \wedge vw \in E \Rightarrow uw \in E$ est vraie. Vous pouvez utiliser ce fait en boîte noire, mais rien ne vous empêche de le démontrer pour vos apprentissages.

Exercice 5. Considérez la version avec poids de VERTEX-COVER. C'est-à-dire, chaque sommet v a un poids $f(v)$ et on cherche un ensemble $X \subseteq V$ qui couvre chaque arête et qui minimise $\sum_{x \in X} f(x)$. Donnez une 2-approximation pour ce problème.

Je recommande de formuler un LP et d'arrondir comme en cours.

Exercice 6. Considérez le LP pour VERTEX-COVER vu en classe. On peut développer un algorithme probabiliste qui retourne $X \subseteq V$ tel que $\mathbb{E}[|X|] = OPT_{LP}$. L'idée est la suivante : pour chaque i de 1 à n, ajouter v_i à X avec probabilité x_i^*.

Argumentez d'abord que $\mathbb{E}[|X|] = OPT_{LP}$.

Ceci porte à croire que l'algorithme est une 1-approximation probabiliste. Il y a toutefois un problème. Quel est-il?

Exercice 7. Dans le problème SET-COVER-k-OCC, on a un univers U à couvrir avec un nombre minimum d'ensembles parmi $S = \{S_1, \ldots, S_m\}$. De plus, pour chaque $u \in U$, on peut supposer qu'il y a au plus k ensembles de S qui contiennent U.

Formulez ce problème avec un ILP. Donnez ensuite une k-approximation en relaxant ce ILP.

Indice. Si vous commencez avec $k = 2$, vous pourrez constater que c'est comme VERTEX-COVER. Généralisez ensuite à d'autres k.

Exercice 8. Montrez que l'algorithme sacadosPoly des notes de cours (section 6.3) n'est pas toujours optimal. (sacadosPoly est l'algorithme qui divise toutes les valeurs par K et appelle la prog dynamique)