Chapitre 1 : Exemples d'approximations polynomiales

MPRO - Complexité : algorithmes approchés

Dimitri Watel (dimitri.watel@ensiie.fr)

2023

Le problème de l'arbre de Steiner

Entrées

- Un graphe $G = (V, E)$ non orienté ;
- des poids $\omega : E \to \mathbb{R}^+$;
- un sous-ensemble de *terminaux* $X \subset V$

Sortie

Un arbre T inclu dans G couvrant X de poids minimum :
$\min \bigcup_{e \in T} \omega(e)$.

Quelques idées

Soit $k = |X|$ et $n = |V|$

- Que se passe-t-il si $k = 2$?
- Que se passe-t-il si $k = n$?

Une 2-approximation

Algorithme de Choukkmane (1978) et de Kou *et al.* (1981), PLesnìk (1981) et Iwainsky *et al.* (1986)

- Pour tout $x, x' \in X$, calculer un plus court chemin $p(x, x')$ de x à x' de poids $d(x, x')$.
- Créer un graphe $H = (X, E')$ complet pondéré avec d.
- Calculer un arbre couvrant T_H de H.
- Calculer $T = \bigcup_{(x,x') \in T_H} p(x, x')$.
- Nettoyer T (supprimer les cycles, les feuilles non terminales, ...).
- Renvoyer T.

Rapport d'approximation : $2 \cdot (1 - \frac{1}{k})$

Le problème du voyageur de commerce

Entrées

- Un graphe $G = (V, E)$ non orienté complet ;
- des poids $\omega : E \to \mathbb{R}^+$.

Sortie

Un cycle hamiltonien C de G de poids minimum : $\min \bigcup_{e \in C} \omega(e)$.

Quelques idées

Soit $k = |X|$ et $n = |V|$

- Qu'obtient-on si on retire une arête d'une solution optimale ?

Une 2-approximation

Algorithme de Christofides (simplifié) (1976)

- Calculer un arbre couvrant T de poids minimum de G
- Renvoyer le cycle visitant les nœuds de G dans l'ordre d'un parcours en profondeur de T.

Rapport d'approximation : ∞
Rapport d'approximation : 2 si inégalité triangulaire.

Une $\frac{3}{2}$-approximation

Algorithme de Christofides (1976)

- Calculer un arbre couvrant T de poids minimum de G
- Soient V' les nœuds de T de degré impairs, calculer dans G un couplage parfait M de V' de poids minimum.
- Calculer un cycle eulérien C' de $T \cup M$ et renvoyer le cycle visitant les nœuds de G dans le même ordre que C'.

Rapport d'approximation : ∞
Rapport d'approximation : $\frac{3}{2}$ si inégalité triangulaire.

Le problème de couverture par ensembles

Entrées

- Un ensemble X ;
- un ensemble de parties de X : $S \subset 2^X$;
- des poids $\omega : S \to \mathbb{R}^+$;

Sortie

Un sous-ensemble $C \subset S$ couvrant X (*i.e.* $X \subset \bigcup_{s \in C} s$) de poids minimum : $\min \sum_{s \in C} \omega(s)$.

Version non pondérée

Entrées

- Un ensemble X ;
- un ensemble de parties de X : $S \in 2^X$;

Sortie

Un sous-ensemble $C \subset S$ couvrant X de taille minimum : $\min |C|$.

Quels ensembles choisisseriez-vous instinctivement ?

Une ln k-approximation (version non pondérée)

Algorithme de Johnson (1974), Lovàsz (1975) et Chvàtal (1979)

- Tant que des éléments de X sont non couverts, sélectionner l'ensemble contenant le plus d'éléments non couverts.

Rapport d'approximation : $\ln(k/OPT) + 1$

Une ln k-approximation (version pondérée)

Algorithme de Johnson (1974), Lovàsz (1975) et Chvàtal (1979)

- Tant que des éléments X' de X sont non couverts, sélectionner l'ensemble s minimisant $\omega(s)/|X' \cap s|$.

Rapport d'approximation : $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k} \sim \ln(k)$

Le problème du sac à dos

Entrées

- Un entier V (le volume du sac) ;
- des entiers $u_1, u_2, \dots, u_n$ (la valeur des objets) ;
- des entiers $v_1, v_2, \dots, v_n$ (le volume des objets).

Sortie

Des objets que l'on peut mettre dans le sac et de valeur maximum : un sous-ensemble $I \subset [\![1; n]\!]$ tels que $\sum\limits_{i \in I} v_i \leq V$ et maximisant $\sum\limits_{i \in I} u_i$.

Quelques idées

Le problème du sac-à-dos est pseudo-polynomial.

- si le volume du sac est polynomial, le problème est polynomial
- si les volumes des objets sont polynomiaux, le problème est polynomial
- si les profits des objets sont polynomiaux, le problème est polynomial

Un FPTAS

Algorithme de Ibarra et Kim (1975)

- Soit $\varepsilon > 0$ et $K = \frac{\varepsilon \max(u_i)}{n}$
- Soit $u'_i = \lfloor \frac{u_i}{K} \rfloor$
- Résoudre l'instance où l'utilité u_i est remplacée par u'_i et renvoyer la solution trouvée

Rapport d'approximation : $1 - \varepsilon$
Complexité en temps : $O(n^2 \lfloor \frac{n}{\varepsilon} \rfloor)$

Le problème du bin packing

Entrées

- Un entier V (le volume d'un *bin*) ;
- des entiers $v_1, v_2, \ldots, v_n$ (le volume des objets).

Sortie

Le nombre minimum de bins nécessaires pour remplir tous les bin : n entiers $a_1, a_2, \ldots, a_n$ tels que $\forall i, \sum_{j : a_j = i} v_j \leq V$ minimisant $\max(a_i)$.

Remarque : on peut supposer $V = 1$

Quelques idées

- Et si ... je reçois les objets dans un ordre quelconque et qu'il fallait prendre une décision immédiatement. Laquelle prendre ?

Une 2-approximation

Algorithme First fit

Pour tout $i \in [\![1; n]\!]$, mettre i dans le premier bin où il peut aller.

Rapport d'approximation : 2

Quelques autres idées

- S'il existe deux constantes ε et K tels que tous les objets sont de taille au moins ε et le nombre d'objets de taille différente est K, le problème est polynomial.

Lemme

Si $v_i \geq \varepsilon$, il existe un algorithme d'approximation de rapport $1 + \varepsilon$:

- Trier les objets
- Les regrouper dans $\lceil \frac{1}{\varepsilon^2} \rceil$ groupes de taille au plus $\lfloor n\varepsilon^2 \rfloor$.
- Remplacer chaque objet x par x', le plus grand de son groupe
- Trouver une solution optimale S dans cette 2e instance
- Renvoyer la même solution dans la 1e instance (où on reremplace x' par x).

Une PTAS asymptotique

Algorithme de Vega et Lueker

- Soit $\varepsilon > 0$
- Soit A tous les objets de taille au plus ε, supprimer A
- Trouver une $1 + \varepsilon$ solution approchée avec l'algorithme du lemme précédent
- Ajouter tous les objets de A avec l'algorithme First Fit.

Rapport d'approximation : $(1 + 2\varepsilon + \frac{1}{OPT})$

Complexité : $O(n^{\binom{M+K}{M}})$, $M = \lfloor \frac{1}{\varepsilon} \rfloor$, $K = \lceil \frac{1}{\varepsilon^2} \rceil$

Le problème de coloration planaire

Entrées

- Un graphe $G = (V, E)$ planaire ;

Sortie

Le nombre minimum de couleurs nécessaire pour colorier G

Quelques idées

- Un graphe est 1 coloriable si et seulement s'il est déconnecté
- Un graphe est 2 coloriable si et seulement s'il est biparti
- Un graphe planaire est toujours 4 coloriable [Appel, Haken (1976)]

Une approximation de rapport absolu 1

Algorithme de coloration de graphe

- Si le graphe est déconnecté, renvoyer 1
- Si le graphe est biparti, renvoyer 2
- Sinon renvoyer 4

Rapport d'approximation (absolu) : 1