Une technique d'approximation.

Plan

1. Présentation du problème

2. Algorithmes d'approximations

3. Heuristiques

1. The set-covering problème.

L'énnoncé du problème

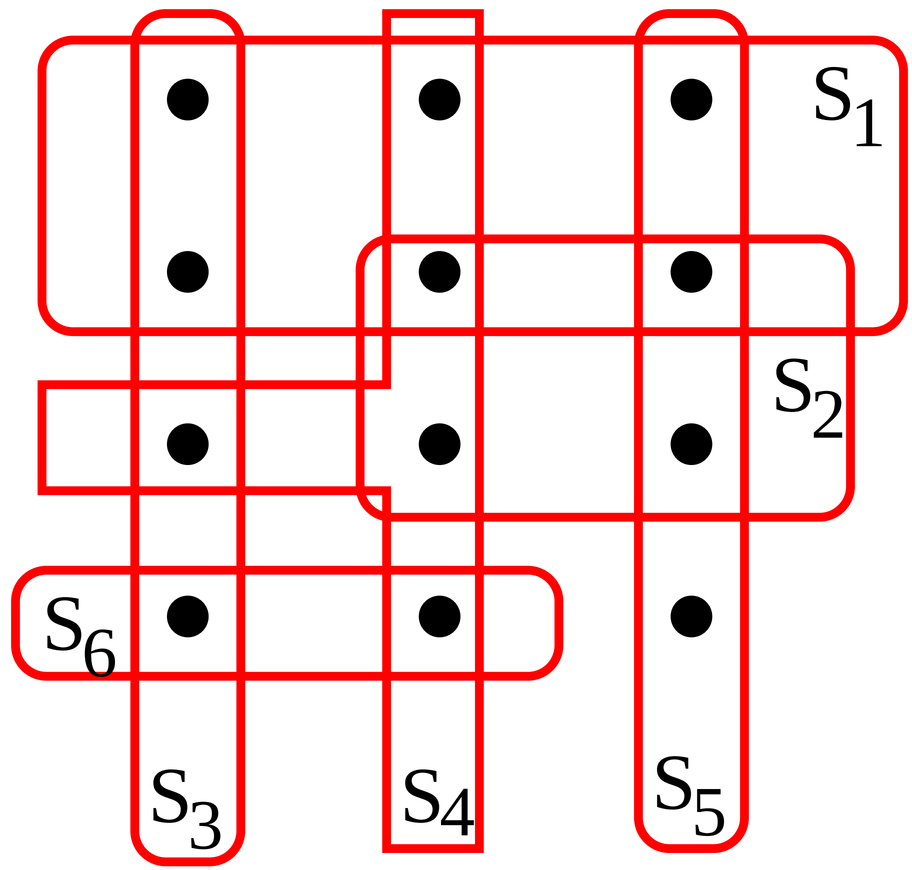
The set-covering Problem

Instance: Un ensemble fini X, une famille $\mathcal{F}$ de X tel que chaque élement de X appartient at au moins un élément de $\mathcal{F}$, un entier k

Question: Existe-t-il une couverture C de $\mathcal{F}$ de taille $\leq k$?

Exemple

X : ens. des 12 points.
$\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$
couverture : $\{S_3, S_4, S_5\}$



Algorithme glouton

Algorithme glouton:

Entrée: X et $\mathcal{F}$

Sortie: un sous ensemble de $\mathcal{F}$

1. $U \longleftarrow X$
2. $\mathcal{C} \longleftarrow \emptyset$
3. tant que $(U \neq \emptyset)$ faire
   (a) Choisir $S \in \mathcal{F}$ tel que $|S \cap U| = max_{S' \in \mathcal{F}}(|S' \cap U|)$.
   (b) $U \longleftarrow U - S$
   (c) $\mathcal{C} \longleftarrow \mathcal{C} \cup \{S\}$
4. retourner $\mathcal{C}$

Algorithme glouton

- **Complexité de l'algorithme :** $\mathcal{O}(|X| \times |\mathcal{F}| \times min(|X|, |\mathcal{F}|))$
- **Solution obtenue dans l'exemple:** $\{S_1, S_4, S_5, S_3\}$
- l'algorithme glouton est un algorithme polynomial avec un rapport de $ln(|X| + 1)$.

Calcul de la qualité de l'algorithme

- cout $\mathcal{C}$ de la solution $\mathcal{C}$ de l'algo. $=$ nombre d'élement dans la solution.
- 1 ajout d'un élément $=$ incremente le cout de la solution DE 1.
- S_i le i^{ieme} ensemble selectionner par l'algorithme.
- Que se passe-t-il si on reporte le cout sur les éléments de X?
- c_x le cout associé à l'élément x: affecté lors que x est couvert pour la premiere fois.
- Si x est couvert pour la première fois par S_i, alors

$$c_x = \frac{1}{|S_i - (S_1 \cup \ldots \cup S_{i-1})|}$$

Calcul de la qualité de l'algorithme

- $|\mathcal{C}| = \sum_{x \in X} c_x$.
- Soit $\mathcal{C}^*$ la solution optimale.
- le cout de la solution optimale $=\sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x$.
- $\sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x \geq \sum_{x \in X} c_x$. ($x$ couvert au moins 1 fois.)
- $|\mathcal{C}| \leq \sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x$

- Supposons que $\forall S \in \mathcal{F}$, $\sum_{x \in S} c_x \leq H(|S|)$
  - le d^{ieme} nombre harmonique $H(d) = \sum_{i=1}^{d} 1/i$
  - $ln(d+1) \leq H(d) \leq ln(d) + 1$

- $|\mathcal{C}| \leq \sum_{S \in \mathcal{C}^*} H(|S|) \leq |\mathcal{C}^*| H(max(|S| : S \in \mathcal{F}))$

Calcul de la qualité de l'algorithme

Prouvons que $\forall S \in \mathcal{F}$, $\sum_{x \in S} c_x \leq H(|S|)$

- Soit $S \in \mathcal{F}$, soit $i = 1, \ldots, |\mathcal{C}|$, $u_i = |S - (S_1 \cup \ldots \cup S_i)|$
- u_i: nombre d'éléments de S non couvert apres la selection$S_1, \ldots, S_i$ $(u_0 = |S|)$
- Soit k le $+$ plus petit indice tel que $u_k = 0$.
- $u_{i-1} - u_i$: nombre d'éléments de S couvert pour la premiere fois par S_i.
- $\sum_{x \in S} c_x = \sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{|S_i - (S_1 \cup \ldots \cup S_{i-1})|}$
- Notons $|S_i - (S_1 \cup \ldots \cup S_{i-1})| \geq |S - (S_1 \cup \ldots \cup S_{i-1})| = u_{i-1}$
- $\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{u_{i-1}}$

Calcul de la qualité de l'algorithme

- $\sum_{x \in S} c_x = \sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{|S_i - (S_1 \cup \ldots \cup S_{i-1})|} \leq \sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{u_{i-1}}$
- $\sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{u_{i-1}} = \sum_{i=1}^{k} (\sum_{j=u_i+1}^{u_{i-1}} frac1u_{i-1})$
- $\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (\sum_{j=u_i+1}^{u_{i-1}} 1/j)$ car $(j \leq u_{i-1})$
- $\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (\sum_{j=u_i+1}^{u_{i-1}} 1/j - \sum_{j=u_i+1}^{u_i} 1/j)$
- $\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (H(u_{i-1} - H(u_i))$
- $\sum_{x \in S} c_x \leq H(u_0) - H(u_k)$
- $\sum_{x \in S} c_x \leq H(u_0) \leq H(|S|)$

2. Le problème des Multipoints relais.

Présentation

- $N(x)$: ensemble de tous voisins de x
- $N^2(x)$: ensemble de tous voisins de x à distance 2.
- $MPR(x)$: ensemble de voisins de x tel que $N^2(x) = \cup_{v \in MPR(x)} N(v)$

Algorithme glouton

Algorithme glouton:

Entrée: un sommet x , $N(x)$, $N^2(x)$

Sortie: un sous ensemble de $N(x)$

1. $U \longleftarrow N^2(x)$
2. $MPR(x) \longleftarrow \emptyset$
3. tant que $\exists v : v \in U \wedge \exists ! w \in N(x) : v \in N(w)$ faire
   (a) $U \longleftarrow U - N(w)$ et $MPR(x) \longleftarrow MPR(x) \cup \{w\}$.
4. tant que $(U \neq \emptyset)$ faire
   (a) Choisir $w \in N(x)$ tel que $|N(w) \cap U| = max(|w' \cap U| : w' \in N(x))$..
   (b) $U \longleftarrow U - N(w)$
   (c) $MPR(x) \longleftarrow MPR(x) \cup \{w\}$
5. retourner $MPR(x)$

Calcul de la qualité de l'algorithme

- Soit S^* la solution optimale et soit S la solution de l'algo.
- S_1: les noeuds choisis à l'étape 3a.$\Rightarrow S_1 \subseteq S^*$.
- $x_1, \ldots, x_k$ les noeuds choisis à l'étape 4a.

- Montrons que $|S - S_1| \leq log(n)|S^* - S_1|$ la solution optimale.
- N_1^2: l'ensemble de noeuds appartenant à $N^2(x)$ et voisins de S_1.
- $N^{2'} = N^2(x) - N_1^2$,$S' = S - S_1$ et $S^{*\prime} = S^* - S_1$
- $N'(v) = N(v) \cap N^{2'}$, $v \in N(x)$.

- 1 ajout d'un élément $x_i =$ incremente le cout de la solution DE 1.

- Que se passe-t-il si on reporte le cout sur les sommets de $N^2(x)$?
- Si x_i est le premier voisin de y mis dans S,

$$c_y = \frac{1}{|N'(x_i) - (N'_1 \cup \ldots \cup N'_{i-1})|}$$

- $S' = \sum_{y \in N'} c_y$
- Supposons que $\forall z \in S^{*\prime}$, $\sum_{y \in N'(z)} c_y \leq log(|N'(z)|)$.
- $|S'| = \sum_{y \in N'} c_y \leq \sum_{z \in S^{*\prime}} \sum_{y \in N'(z)} c_y \leq \sum_{z \in S^{*\prime}} log(|N'(z)|)$
- $|S'| \leq |S^{*\prime}| log(n)$