Recherche opérationnelle

DUT Info 2e année, parcours A

Algorithmes gloutons

Florent Foucaud

IUT CLERMONT AUVERGNE

Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

Algorithme glouton

Définition (algorithme glouton / greedy algorithm)

C’est un type d’algorithme qui construit une solution pas à pas, en prenant la meilleure décision à chaque étape

→ (optimum local mais pas forcémeent global).

Le Glouton ou Carcajou, un animal du grand nord

(“Wolverine” en anglais)

Exemple : arbre couvrant de poids minimum

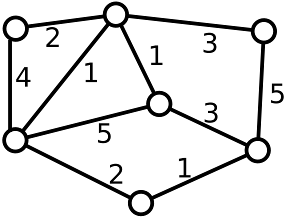
Problème : Construire un arbre couvrant T d'un graphe G (T touche tous les sommets)

Objectif : minimiser le poids total des arêtes de l'arbre

Exemple : arbre couvrant de poids minimum

Problème : Construire un arbre couvrant T d'un graphe G (T touche tous les sommets)

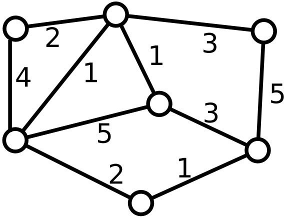
Objectif : minimiser le poids total des arêtes de l'arbre



Exemple : arbre couvrant de poids minimum

Problème : Construire un arbre couvrant T d'un graphe G (T touche tous les sommets)

Objectif : minimiser le poids total des arêtes de l'arbre



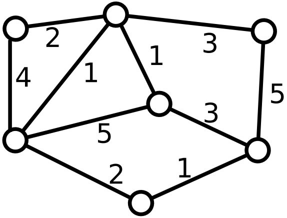
Algorithme de Jarník-Prim :

- Choisir un sommet v_0 de départ ; $T = (\{v_0\}, \{\})$
- Tant que tous les sommets de G ne sont pas dans l'arbre T
  - Choisir un sommet v dans $V(G) \setminus V(T)$ qui a le plus petit coût de rattachement à T
  - $T = T \cup \{v\}$
- Renvoyer T

Exemple : arbre couvrant de poids minimum

Problème : Construire un arbre couvrant T d'un graphe G (T touche tous les sommets)

Objectif : minimiser le poids total des arêtes de l'arbre



Algorithme de Jarník-Prim :

- Choisir un sommet v_0 de départ ; $T = (\{v_0\}, \{\})$
- Tant que tous les sommets de G ne sont pas dans l'arbre T
  - Choisir un sommet v dans $V(G) \setminus V(T)$ qui a le plus petit coût de rattachement à T
  - $T = T \cup \{v\}$
- Renvoyer T

Vojtěch Jarník
(1897-1970)

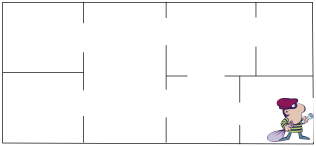
Robert C. Prim
(1921-)

Edsger W. Dijkstra
(1930-2002)

Surveillance d'un bâtiment : couverture par sommets

Problème : couvrir les couloirs d'un bâtiment avec des détecteurs de mouvement. Pour un sommet v, poser un détecteur coûte $c(v)$.

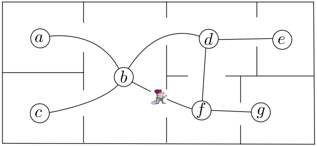
Objectif : minimiser le coût total de la pose des détecteurs



Surveillance d'un bâtiment : couverture par sommets

Problème : couvrir les couloirs d'un bâtiment avec des détecteurs de mouvement.
Pour un sommet v, poser un détecteur coûte $c(v)$.

Objectif : minimiser le coût total de la pose des détecteurs

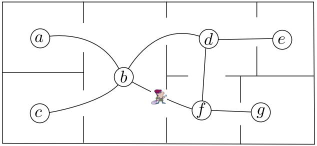


Le bâtiment est un graphe non-orienté $G = (V, E)$.

Surveillance d'un bâtiment : couverture par sommets

Problème : couvrir les couloirs d'un bâtiment avec des détecteurs de mouvement.
Pour un sommet v, poser un détecteur coûte $c(v)$.

Objectif : minimiser le coût total de la pose des détecteurs



Le bâtiment est un graphe non-orienté $G = (V, E)$.

On écrit le PL en nombres entiers suivant :

Une variable x_v pour chaque sommet v : $x_v = 1$ si on a un détecteur sur v, 0 sinon.
Chaque sommet v a un poids $c(v)$ (le coût de la pose d'un détecteur sur v).

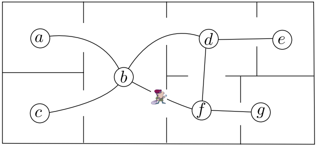
$$\begin{array}{lrcll}
\text{minimiser :} & \sum_{v \in V} x_v c(v) & & & \\
\text{tel que :} & x_u + x_v & \geq & 1 & \forall uv \in E \\
& x_v & \leq & 1 & \forall v \in V \\
& x_v & \geq & 0 & \forall v \in V \\
& x_v & \in & \mathbb{N} & \forall v \in V
\end{array}$$

On obtient une solution approchée (au pire deux fois moins bonne que l'optimum) en passant par la relaxation linéaire de ce PL. (Voir le cours précédent.)

Surveillance d'un bâtiment : couverture par sommets

Problème : couvrir les couloirs d'un bâtiment avec des détecteurs de mouvement.
Pour un sommet v, poser un détecteur coûte $c(v)$.

Objectif : minimiser le coût total de la pose des détecteurs



Le bâtiment est un graphe non-orienté $G = (V, E)$.

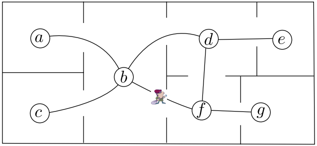
Algorithme glouton n°1 :

- $S = \{\}$
- Tant que toutes les arêtes ne sont pas couvertes :
  - Choisir un sommet v qui touche le plus grand nombre d'arêtes non couvertes par S
  - $S = S \cup \{v\}$
- Renvoyer S

Surveillance d’un bâtiment : couverture par sommets

Problème : couvrir les couloirs d’un bâtiment avec des détecteurs de mouvement.
Pour un sommet v, poser un détecteur coûte $c(v)$.

Objectif : minimiser le coût total de la pose des détecteurs



Le bâtiment est un graphe non-orienté $G = (V, E)$.

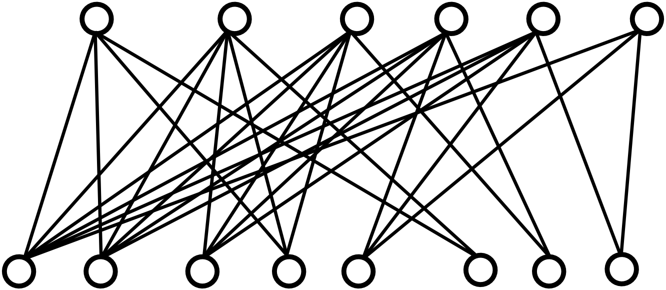
Algorithme glouton n°1 :

- $S = \{\}$
- Tant que toutes les arêtes ne sont pas couvertes :
  - Choisir un sommet v qui touche le plus grand nombre d’arêtes non couvertes par S
  - $S = S \cup \{v\}$
- Renvoyer S

Algorithme glouton n°2 :

- $S = \{\}$
- Tant que toutes les arêtes ne sont pas couvertes :
  - Choisir un sommet v qui maximise le ratio “nombre d’arêtes non couvertes” / $c(v)$
  - $S = S \cup \{v\}$
- Renvoyer S

Aucun des deux algos gloutons n'est bon



(tous les sommets ont le même poids)

Les algorithmes gloutons du transparent précédent n'ont

aucune garantie de performances...