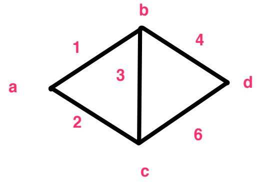
Matroïdes

Michel Habib
Algorithmique, L3

13 décembre 2017

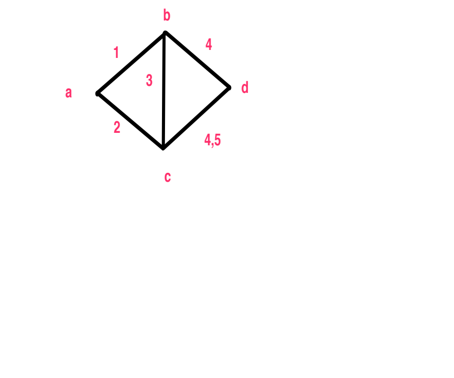
Plan

Le calcul du 2ème arbre



L'arbre optimum est constitué des arêtes ab, ac, bd et vaut 7. Le deuxième arbre lexicographique est ab, ac, cd et vaut 9. Par contre l'arbre ab, bc, bd vaut 8.

2eme arbre

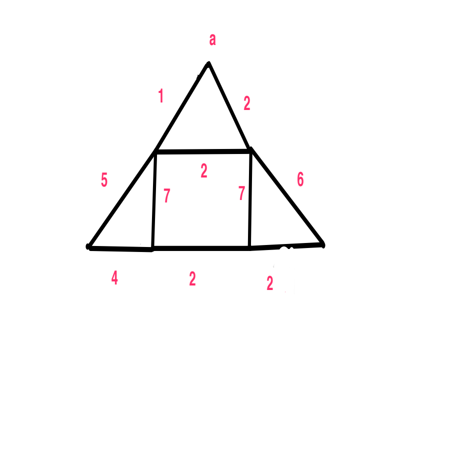


L'arbre optimum est constitué des arêtes ab, ac, bd et vaut 7. Le deuxième arbre lexicographique est ab, ac, cd et vaut 7,5 Par contre l'arbre ab, bc, bd vaut 8 et il n'y a pas d'autre arbre entre 7 et 7,5.

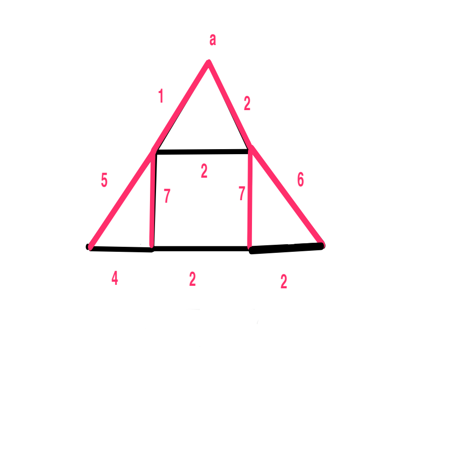
En conclusion

- Le deuxième arbre de poids minimum n'est pas toujours le deuxième arbre suivant l'ordre lexicographique sur les arbres.
- Peut-on caractériser les graphes pour lesquels cette propriété est vraie ?

Plus courts chemins

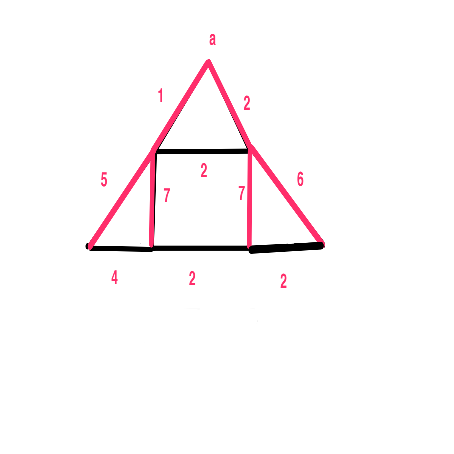


Plus courts chemins



L'arbre de poids minimum n'est pas identique à l'arbre des plus courtes chaînes issues de a.

Plus courts chemins



L'arbre de poids minimum n'est pas identique à l'arbre des plus courtes chaînes issues de a.

- Mais l'algorithme de Prim ressemble à un parcours de graphe.

Théorème

Un arbre de poids min est un arbre des distances min, si l'on choisit comme valuation d'une chaîne le maximum des valuations des arêtes de la chaîne.

Preuve

Supposons le contraire, on considère un arbre recouvrant minimal T_0 sur un graphe G et deux sommets $a, b \in V(G)$ pour lesquels il existe une meilleure chaîne μ dont la valuation max est meilleure que celle de chemin ν de a vers b dans T_0.
Soit $e = uv$ l'arête de valuation maximale de ν. On considère z, t les premières intersections respectivement avant u et après v entre μ et ν (éventuellement ce sont les sommets a et b). $[z, t]_\nu$ et $[z, t]_\mu$ permettent de construire un cycle μ' de G.

Considérons A, B les deux composantes connexes de $T_0 \setminus e$. Le cocyle $\theta(A)$ et le cycle μ' ont au moins deux arêtes en commun : e et au moins une autre f.
Comme $T_0(A)$ et $T_0(B)$ sont connexes $\theta(A)$ ne contient qu'une arête de T_0. Donc $f \notin T_0$.
Par hypothèse la chaîne μ étant meilleure que ν pour la valuation maximale : $\omega(f) < \omega(e)$ et donc
on peut construire l'arbre $T_0' = T_0 \setminus e + f$ vérifiant $\omega(T_0') < \omega(T_0)$ d'où la contradiction.

Lemme d'échange sur les arbres

On considère un graphe connexe $G = (X, E)$ et deux arbres recouvrants sur G : $T = (X, F)$ et $T' = (X, F')$ (F, F' sont inclus dans E).

Echange entre arbres

$\forall e \in F\text{-}F'$, $\exists f \in F'\text{-}F$ such that $T'' = (X, F\text{-}e + f)$ est un arbre recouvrant de G.

Conséquences

Etant donné deux arbres recouvrants T, T' de G, il existe une suite d'échanges de 2 arêtes qui permet de passer de T à T'.

Mais aussi

Une structure d'espace vectoriel sur le corps $\{0, 1\}$ pour les cycles et les cocyles.

Abstraction de la notion d'indépendance les matroïdes généralisent à la fois les espaces vectoriels ainsi que les forêts dans les graphes.

Une définition de matroïde par ses indépendants

Withney [1935] : $M = (X, \mathcal{I})$ constitué d'un ensemble fini X et d'une famille de parties $\mathcal{I}$ de X vérifiant les 3 axiomes suivants est un matroïde.

1. $\emptyset \in \mathcal{I}$
2. Si $I \in \mathcal{I}$ alors $\forall J \subseteq I,\ J \in \mathcal{I}$
3. Si $I, J \in \mathcal{I}$ et $|I| = |J| + 1$ alors $\exists x \in I$-J tel que : $J \cup \{x\} \in \mathcal{I}$.

Des variantes de l'axiome 3

- 3' Si $I, J \in \mathcal{I}$ et $|I| > |J|$ alors $\exists x \in I$-J tel que : $J \cup \{x\} \in \mathcal{I}$.
- 3'' $\forall A \subseteq X$ et $\forall I, J \subseteq A$, $I, J \in \mathcal{I}$ and maximal then $|I| = |J|$
- Il est facile de se convaincre que ses 3 axiomes sont équivalents.

1 et 2 = famille héréditaire de parties.
Les bases sont les indépendants de taille maximale et elles ont toutes le même cardinal (d'après 3).

Le matroïde défini par ses bases

1. $\mathcal{B} \neq \emptyset$
2. $\forall B, B' \in \mathcal{B}$, $\forall x \in B \setminus B'$ et $\exists y \in B' \setminus B$ tels que : $B\text{-}x{+}y \in \mathcal{B}$.

Rappel

Echange entre arbres, étant donné deux arbres recouvrants T, T' d'un graphe G :
$\forall e \in F\text{-}F'$, $\exists f \in F'\text{-}F$ such that $T'' = (X, F\text{-}e + f)$ est un arbre recouvrant de G.

Matroïdes Graphiques

A partir d'un graphe non orienté G on peut définir un matroïde comme suit :
$M = (E(G), \mathcal{I})$ où $\mathcal{I}$ est constitué des ensembles d'arêtes des forêts de G.

- $\mathcal{I}$ est trivialement une famille héréditaire.
- Concernant l'axiome 3. Considérons deux forêts $I, J \in \mathcal{I}$ telles que $|I| = |J| + 1$.
  Soient $X_1, \ldots, X_k$ les ensembles de sommets des composantes connexes de J, n.b. ces ccx contiennent les sommets isolés du graphe. S'il existe une arête e de I, entre deux de ces composantes connexes, alors $J \cup \{e\}$ n'a pas de cycle et donc $J \cup \{e\} \in \mathcal{I}$.
  Sinon toutes les arêtes de I sont internes aux composantes connexes, mais alors $|I| \leq |J|$, contradiction.

Un matroïde $M = (X, \mathcal{I})$ est appelé graphique, s'il existe un graphe G tel que
$E(G) = X$ et $\mathcal{I}$ est égal aux forêts de G (en tant qu'ensemble d'arêtes).
Tous les matroïdes ne sont pas graphiques.

Théorème principal Edmonds, Rado

1. Soit $M = (E, \mathcal{I})$ un matroïde et $\omega : E \to \mathcal{R}^+$ une fonction de poids, alors :
   un élément $J \in \mathcal{I}$ lexicographiquement maximum de M est de poids maximum (pour la somme).
2. Sorte de réciproque
   On considère une famille de parties J sur E qui vérifie les axiomes 1 et 2. Si pour toute fonction de poids ω un élément J lexicographiquement maximum de M est de poids maximum (pour la somme), alors $M = (E, \mathcal{J})$ est un matroïde.

Preuve du 1

Soit $A = (a_1, \dots a_k)$ t.q. $\omega(a_1) \geq \omega(a_2) \dots \omega(a_k)$, un indépendant lexicographiquement maximal. Vu que la valuation ω est positive, A est nécessairement une base de M. De même si l'on considère un indépendant de poids maximal $B = (b_1, \dots b_h)$ t.q. $\omega(b_1) \geq \omega(b_2) \dots \omega(b_h)$ c'est aussi une base. Donc posons h=k=n. Par définition de l'ordre lexicographique (qui est l'ordre de la première différence) et donc $A \gg B$ implique :
$\exists k \in [1, n]$ t.q. pour $i \leq k$, $\omega(a_i) = \omega(b_i)$ and $\omega(a_{k+1}) > \omega(b_{k+1})$. (On supposera ici $A \neq B$).

Considérons les deux ensembles suivants :
$A_{k+1} = (a_1, \dots a_{k+1})$ t.q. $\omega(a_1) \geq \omega(a_2) \dots \omega(a_{k+1})$ et
$B_k = (b_1, \dots b_k)$ t.q. $\omega(b_1) \geq \omega(b_2) \dots \omega(b_k)$
Comme un matroïde est une famille de parties héréditaires, A_{k+1}, B_k sont aussi indépendants.
D'après l'axiome 3 des matroïdes, comme $|A_{k+1}| = |B_k| + 1$, il existe $a \in \{a_1, \dots, a_{k+1}\}$ tel que $B'_{k+1} = B_k + a$ soit un indépendant. En outre
$\omega(B'_{k+1}) = \omega(B_k) + \omega(a) > \omega(B_k) + \omega(b_{k+1}) = \omega(B_{k+1})$.
En considérant B'_{k+1} et B on peut compléter en ajoutant que des b_j avec $j > k+1$ (en utilisant $n-k-1$ fois l'axiome 3) B'_k en une base B' qui vérifie :
$\omega(B') > \omega(B)$ ce qui contredit la maximalité de B.

Preuve du 2, l'espèce de réciproque

On considère $A \subseteq X$ et $I, J \in \mathcal{I}$ maximaux pour l'inclusion.
Supposons $|I| < |J|$.
Définissons une valuation $\omega : X \to R^+$ de la manière suivante :
$\omega(x) = 1 + \alpha$ si $x \in I$
$\omega(x) = 1$ si $x \in J$-I
$\omega(x) = 0$ si $x \in X$-I-J
Cela donne $\omega(I) = |I|.(1 + \alpha)$ et $\omega(J) = |I \cap J|.(1 + \alpha) + |J\text{-}I|$

- I est lexicographiquement plus grand que J.
- $|I| < |J|$ implique que $|I\text{-}J| < |J\text{-}I|$
- $\omega(I) < \omega(J)$ est équivalent à $(1+\alpha) < \frac{|J-I|}{|I-J|}$. Vu la remarque ci-dessus, il existe une telle valeur de $\alpha > 0$, d'où la contradiction.
  Donc si l'axiome 3" n'est pas vérifié, nous avons exhibé une fonction de poids pour laquelle lexicographiquement maximal n'implique pas maximum pour la somme.

Le glouton sur un matroïde

Glouton standard sur un matroïde $M = (E, \mathcal{F})$

$X \leftarrow \emptyset$;
tant que $|A| \neq \emptyset$ **faire**
    $A \leftarrow \{x \in E$ tel que $X \cup \{x\} \in \mathcal{F}\}$;
    Choisir $a \in A$ de poids maximum ;
    $X \leftarrow X \cup \{a\}$;
fin

Cette procédure calcule un indépendant lexicographiquement maximum et donc l'optimum (cf. le théorème d'Edmonds et Rado).
La complexité de l'algorithme dépend de celle du calcul de l'ensemble A.
Mais les choix ne sont jamais remis en cause (principal paradigme de la gloutonnerie).

Algorithmique sur les matroïdes

Lié à la représentation du matroïde $M = (E, \mathcal{I})$:

- Par un oracle qui répond sur une partie $I \subseteq E$ est un indépendant en $O(1)$
- Par un oracle qui répond sur une partie $I \subseteq E$ est un indépendant en $O(|I|)$
- Mais vérifier que $X \cup \{x\} \in \mathcal{F}$ peut demander beaucoup de ressources pour certains matroïdes. (i.e. demander de résoudre un problème NP-complet par exemple).

Cependant l'algorithme de Prim qui calcule aussi un arbre de poids min dans un graphe, ne s'exprime pas dans un matroïde général. Il utilise trop la structure du matroïde graphique sous-jacent.

Un matroïde défini par ses cycles

Les cycles sont les dépendants minimaux

1. $\forall C \in \mathcal{C}, \forall C' \subset C,\ C' \notin \mathcal{C}$
2. $\forall C, C' \in \mathcal{C}, \forall a \in C \cap C', \exists C" \in C$ tel que : $C" \subseteq C \cup C'$-a

Cycles d'un matroïde

Ce sont les dépendants minimaux.

Axiomatique des cycles d'un matroïde

(1) $C_1, C_2 \in \mathcal{C}$ si $C_1 \subseteq C_2$ alors $C_1 = C_2$.
(2) Pour $C_1 \neq C_2 \in \mathcal{C}$ et $x \in C_1 \cap C_2$, il existe $C_3 \in \mathcal{C}$, tel que $C_3 \subseteq C_1 \cup C_2 - x$

On peut donc écrire une variante de l'algorithme de Kruskal

La deuxième base de poids maximum

Soit $A = \{a_1, \dots, a_n\}$ une base de poids maximum d'un matroïde M et $B = \{b_1, \dots, b_n\}$ la deuxième base de poids maximum de ce matroïde, alors il existe un échange qui permet de passer de B à A.

Preuve

Si $B \neq A$, il existe $b_j \notin A$. D'après l'axiome d'échange sur les bases, il existe $a_i \in A$ tel que : $A' = A - a_i + b_j$ soit une autre base de M.
$\omega(A') = \omega(A) - \omega(a_i) + \omega(b_j)$ comme A est optimum nécessairement :
$\omega(b_j) \leq \omega(a_i)$ et donc : $\omega(B) \leq \omega(A') \leq \omega(A)$.
Comme $A' \neq A$, nécessairement $\omega(B) = \omega(A')$. CQFD

Fonction de rang d'un matroïde

Au matroïde $M = (E, \mathcal{I})$ on associe la fonction de rang :
$r_M : \mathcal{P}\ (E) \to \mathcal{N}$ telle que :
$\forall A \subseteq E$, $r_M(A) = max\{|I|$ avec $I \in \mathcal{I}$ et $I \subseteq A\}$
Cette fonction est bien définie car :
$M(A) = (A, \mathcal{J})$ avec $\mathcal{J} = \{\mathcal{I} \cap \mathcal{A} | \mathcal{I} \in \mathcal{I}\}$ est aussi un matroïde et $r_M(A)$ est égal au cardinal d'une base de ce matroïde.

Ou de manière équivalente :

$r : \mathcal{P}(\mathcal{X}) \rightarrow \mathcal{N}$ qui vérifie :

1. $r(\emptyset) = 0$
2. $\forall A \subseteq X$, $\forall A \subseteq X$, $r(A + x)$-$r(A) \in \{0, 1\}$
3. $\forall A \subseteq X$, $\forall x, y \in X$-A such that : $r(A + x) = r(A + y) = r(A)$ alors $r(A + x + y) = r(A)$.

Ou encore

$r : \mathcal{P}(\mathcal{X}) \to \mathcal{N}$ qui vérifie :

1. $\forall A \subseteq X,\ r(A) \leq |A|$
2. $\forall A, B \subseteq X,\ A \subseteq B$ alors $r(A) \leq r(B)$
3. submodularité
   $\forall A, B \subseteq X,\ r(A \cap B) + r(A \cup B) \leq r(A) + r(B)$

- Les indépendants vérifient $r(A) = |A|$.
  La fonction de rang peut servir d'oracle afin de manipuler un matroïde (cf. l'algorithme glouton).
- submodularité = sorte de convexité discrète
- Matroïdes par les fermés, treillis, géométries ...

Matroïde dual d'un matroïde

Au matroïde $M = (E, \mathcal{I})$ on associe le matroïde dual $M^* = (E, \mathcal{C}^*)$
$\forall A \subseteq E$, $r^*(A) = |A| + r(E\text{-}A) - r(E)$
B est une base de M ssi E-B est une base de M^*.

Les différentes définitions d'un matroïde : par ses indépendants, par ses bases (indépendants maximaux), sa fonction de rang, ses cycles (dépendants minimaux) définissent le même concept mathématique, mais hélas elles ne sont pas du tout équivalentes du point de vue de la complexité algorithmique.
Notion d'oracle pour l'indépendance

Exemple de matroïde sur E :
$M_k =$ l'ensemble des parties de E ayant $\leq k$ éléments.
appelé le matroïde uniforme.
Les axiomes sont trivialement vérifiés.

Le matroïde de partition

Soit $P = \{E_1, \ldots, E_k\}$ une partition de E et k entiers positifs $d_1, \ldots, d_k$ et $\mathcal{I}$=$\{$ $I \subseteq E$ tel que $\forall i \in [1, k]$, $|I \cap E_i| \leq d_i\}$ alors $M = (E, \mathcal{I})$ est un matroïde dit de partition.
Les axiomes 1 et 2 sont trivialement vrais.
Considérons 3. Considérons I, J tels que $|I| = |J| + 1$.
Si $\forall i$, $|E_i \cap I| = |E_i \cap J|$ alors $|I| = |J|$.
Donc il existe au moins un entier i tel que $d_i \geq |E_i \cap I| > |E_i \cap J|$ et donc on peut augmenter J d'au moins un élément.

Paradigmes algorithmiques et matroïdes

- Algorithmes gloutons :
  les prototypes des algorithmes gloutons sont les algorithmes pour l'arbre de poids maximum.
  Indépendant de poids maximum dans un matroïde.
- Algorithmes par échanges améliorants (principe des chaînes alternées dans les couplages ou flots)
  Indépendant commun à deux matroïdes de taille maximum
- Programmation linéaire et algorithmes de type simplexe qui examine tous les sommets d'un polyèdre.
  Matroïdes orientés

Exercices :

1. Comment se calcule la fonction de rang d'un matroïde graphique (on suppose disposer du graphe) ?
2. Pour le problème de l'arbre de Steiner avec $|S| = 2$, montrer que la solution est la plus courte chaîne reliant les deux sommets.

$G = (X, U)$ un graphe orienté avec $|X| = n$, une fonction de poids $\omega : U \to R_+$, considérons les 3 matroïdes suivants définis sur G :

$M_1 = (U, \mathcal{I})$

Le matroïde graphique des forêts de G

$M_2 = (U, \mathcal{J})$

avec $\mathcal{J}$=$\{J \subseteq U$ tel que $|J| \leq n - 1$ et $\forall x \in X$, J contient au plus un arc sortant de $x\}$

$M_3 = (U, \mathcal{K})$

avec $\mathcal{K} = \{K \subseteq U$ tel que $|K| \leq n - 1$ et $\forall x \in X$, K contient au plus un arc entrant en $x\}$

Tout d'abord

M_1, M_2 et M_3 sont bien des matroïdes.

Intersection de 2

$A \subseteq U$ est un arbre orienté recouvrant de poids maximal
ssi
A est une base de poids maximal commune à M_1 et M_2

Intersection de 3

$A \subseteq U$ est un chemin hamiltonien de poids maximal
ssi
A est une base de poids maximal commune à M_1, M_2 et M_3.

Représentations d'un matroïde

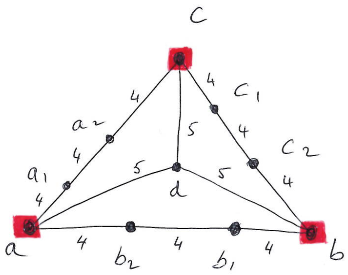
- Pour un matroïde donné on recherche s'il existe une représentation matricielle sur un corps F (car ceci implique des calculs polynomiaux en utilisant la notion de systèmes de vecteurs indépendants).
- Les matroïdes graphiques sont représentables en utilisant la matrice d'incidence sommet –arêtes.
- papier Krastch soda 2012
  transversal matroïdes et gammoids seulement représentables de manière randomisées.

Les limites de la gloutonnerie

Arbre de Steiner

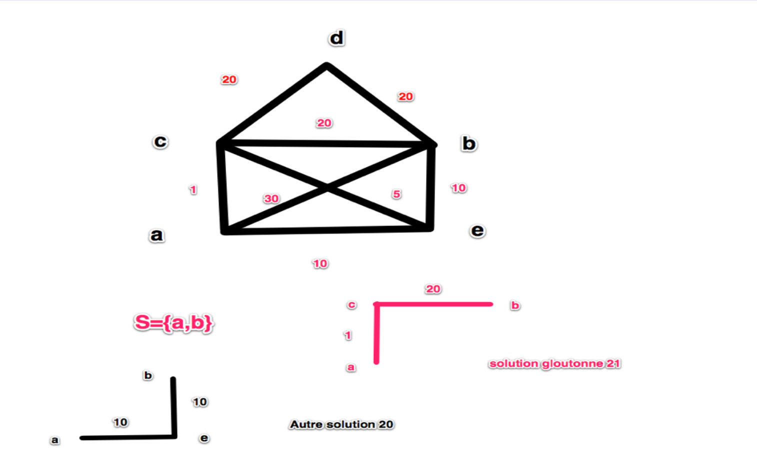
Données : un graphe $G = (X, E)$ connexe et $S \subseteq X$, une fonction de poids $\omega : E \to R_+$.
Résultat : un arbre $T = (Y, F)$ tel que $S \subseteq Y$ et T de poids minimum.

Un exemple I

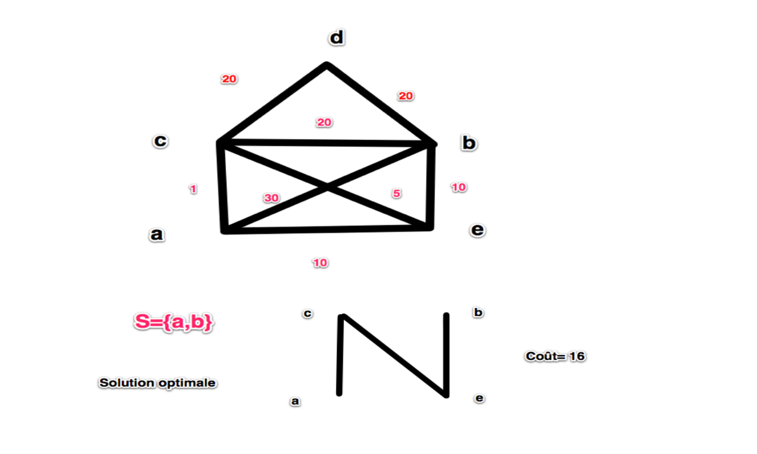


In this example : $S = \{a, b, c\}$. The optimum is the star (ad, bd, cd).
A Prim's inspired greedy solution would be :
$(aa_1, a_1a_2, a_2c, cc_1, c_1c_2, c_2b)$

Un exemple II



Un exemple III



Ce problème très utilisé et étudié en VLSI (nombreuses heuristiques ad-hoc disponibles).
Les solutions ne vérifient pas de lemme d'échange et n'ont pas le même nombre d'arêtes.
Le problème est NP-difficile, sauf pour certaines valeurs de S.
Par exemple : $S = X$ ou $|S| \leq 4 \ldots$

L'arbre de steiner est un problème important en recherche opérationnelle et optimisation :

http://corner.mimuw.edu.pl/?p=354

4-sous-séquence

Données : n entiers $a_1, \ldots, a_n$
Résultat : la sous-séquence constituée de 4 éléments consécutifs dont la somme est maximale.

Exemple

La donnée : $-5, -5, -5, 10, -5, -5, -5, 1, 1, 1, 1$
Si par gloutonnerie on choisit 10 (le plus grand élément), on obtient une solution de valeur -5
Alors qu'il existe une solution de valeur 4.

Sur ce problème du calcul de la sous-séquence maximale, c'est un autre paradigme différent de la gloutonnerie qu'il faut utiliser pour obtenir un bon algorithme (linéaire).

Le principe dit du balayage

Paradigme très utilisé dans le cadre des algorithmes géométriques.

A comparer avec

Données : n entiers $a_1, \ldots, a_n$
Résultat :le sous-ensemble de 4 éléments dont la somme est maximale.
On retrouve ici M_4 le matroïde uniforme sur lequel l'algorithme glouton donne l'optimum.

Bin packing

Données : n entiers $2, 3, 3, 4, 4, 4, \ldots$
Résultat : on doit les ranger par paquet de taille ≤ 10 en minimisant les pertes.

Le glouton donne : 2, 3, 3 puis 4, 4 puis 4 . . . perte de 4 dans les 2 premiers
une solution meilleure :
3, 3, 4 puis 2, 4, 4 puis 4 . . . perte de 0 dans les 2 premiers

Il existe des structures moins fortes que les matroïdes sur lesquelles les gloutons fonctionnent :
Les greedoïdes qui contiennent les matroïdes et les antimatroïdes.

Schéma glouton

On veut optimiser une fonction f sur n variables x_i
les valeurs $a_1, \dots a_i$ ayant été fixées, on cherche la valeur de x_{i+1} qui optimise localement la fonction f.
A défaut de toujours fournir l'optimum, les schémas gloutons permettent de construire de puissantes heuristiques.

Paires de ski

Un moniteur de ski doit distribuer des paires de ski à ses stagiaires. Les stagiaires et les skis sont munies d'une longueur. On supposera qu'il y a autant de paires de ski que de stagiaires.
Afin d'harmoniser la distribution, le moniteur veut trouver l'affectation "optimale", celle qui minimise la somme des valeurs absolues des différences entre la taille d'un stagiaire et celle de la paire de skis qui lui est affectée.

Colonie de vacances

Données : n stagiaires de taille $x_1 \leq \ldots, \leq x_n$, n paires de ski $s_1 \leq \ldots, \leq s_n$
Résultat : l'affectation σ qui minimise la somme des valeurs absolues des différences : $\Sigma_{i=1}^{i=n} |x_i - s_{\sigma(i)}|$

À quel schéma glouton correspond l'algorithme de Dijkstra ?

Un problème d'ordonnancement à une machine

On dispose de n tâches unitaires $T = \{t_1, \ldots, t_n\}$, toutes munies d'une date limite de démarrage $D = \{d_1, \ldots, d_n\}$ et d'une pénalité en cas d'exécution après la date limite $P = \{p_1, \ldots, p_n\}$.

Exemple avec 7 tâches, numérotées de 1 à 7

$T = \{t_1, \ldots, t_7\}$
$D = \{4, 2, 4, 3, 4, 1, 6\}$
$P = \{70, 60, 50, 40, 30, 20, 10\}$

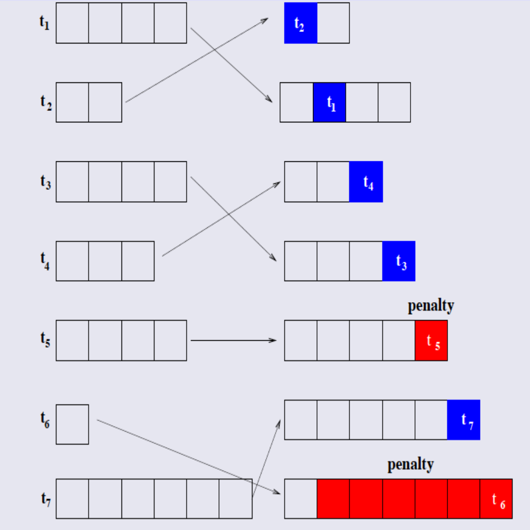
La question principale

Quel algorithme utiliser pour ordonnancer ces tâches à moindre pénalité ?

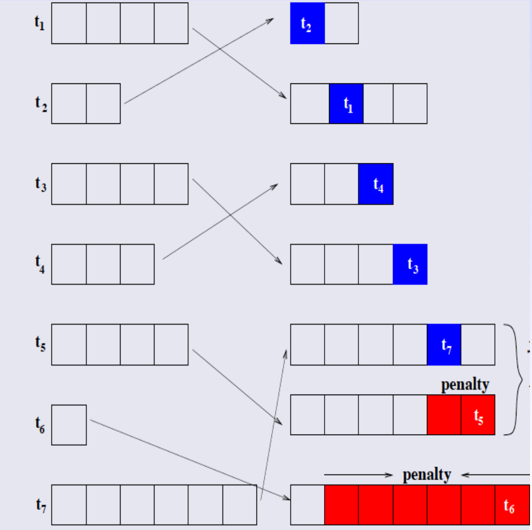
Premières remarques

- Ordonnancer revient exactement à affecter une date de début à chacune des tâches. Ou encore trouver une permutation sur $[1, n]$, car les tâches sont unitaires (elles ont toutes la même durée).
- Sur l'exemple la permutation identité à une pénalité de 100. Peut-on faire mieux ?
- Oui, car l'ordonnancement : $2, 1, 4, 3, 5, 6, 7$ déclenche une pénalité de 50.
- Minimiser les pénalités revient à maximiser les non-pénalités.

Un exemple



Un exemple suite



Forme canonique

- Comme les pénalités ne s'exercent qu'une fois, on peut transformer tout ordonnancement en un ordonnancement bipartitionné :
  en premier les tâches à-temps, puis les tâches hors-temps.
- Un ordonnancement est sous forme canonique lorsque ses tâches à-temps sont ordonnées suivant l'ordre croissant de leur date limite.

L'algorithme glouton 1

1. Choisir tant qu'il en existe au moins une, la tâche ayant la pénalité maximale encore excécutable dans les temps.
2. Quand il n'en existe plus, ajouter les tâches restantes à l'ordonnancement dans un ordre quelconque.

Hélas, il n'est pas correct, comme le montre l'exemple à deux tâches :
t_1, t_2
$d_1 = 2, d_2 = 1$
$p_1 = 60, p_2 = 50$
l'algorithme donnerait l'ordonnancement t_1 puis t_2 ayant une pénalité de 50.
Alors que l'ordonnancement t_2 puis t_1 est meilleur car il n'a pas de pénalité.

L'algorithme glouton 2

for $i \leftarrow 1$ **to** n **do**
  $T(i) \leftarrow \{t_j | d_j = i\}$;
  Choisir dans $T(i)$ une tâche de pénalité maximale.
  if $T(i) = \emptyset$ **then**
    choisir une tâche encore dans les délais et de pénalité maximale
  end
end

Hélas même s'il donne le bon ordonnancement sur l'exemple précédent à deux tâches.
Il est facile de faire un contre-exemple. Par exemple sur l'exemple à 7 tâches on doit nécessairement commencer par la tâche t_6, ce qui ne permet pas de construire un ordonnancement optimal.

Mais il y a peut-être un matroïde caché ici ?

Rappel : une définition de matroïdes par ses indépendants

$M = (X, \mathcal{I})$ est un matroïde :
un ensemble fini X, $\mathcal{I}$ une famille de parties de X qui vérifie :

1. $\emptyset \in \mathcal{I}$
2. Si $I \in \mathcal{I}$ alors $\forall J \subseteq I$, $J \in \mathcal{I}$
3. Si $I, J \in \mathcal{I}$ et $|I| = |J| + 1$ alors $\exists x \in I$-J tel que :
   $J \cup \{x\} \in \mathcal{I}$.

Le matroïde associé

Indépendants

Un ensemble A de tâches est un indépendant ss'il existe un ordonnancement dans lequel les tâches de A sont exécutées dans les délais.

Notation

Pour un ensemble de tâches A, notons $N_t(A) = \{$les tâches de A dont les temps limites sont $\leq t\}$. Par convention : $\forall A,\ N_0(A) = \emptyset$.

Conséquences

Un ensemble de tâches A est un indépendant ssi $\forall i,\ 1 \leq i \leq n$, $|N_i(A)| \leq i$.

1. Trivialement les indépendants sont des parties héréditaires.
2. On considère deux indépendants A, B tels que $|B| = |A| + 1$.
   Comme $|N_0(A)| = |N_0(B)| = 0$ et $|N_n(B)| = |N_n(A)| + 1$, il existe un k maximum tel que $|N_k(B)| = |N_k(A)|$.
   Donc $\forall j,\ k+1 \leq j \leq n$ nous avons $|N_j(B)| = |N_j(A)| + 1$.
   Ainsi B a une tâche de plus b que A ayant une date limite $k+1$.
   Prenons $A' = A \cup \{b\}$ et A' est bien un indépendant.
   En effet : $\forall i \leq k$, $|N_i(A')| = |N_i(A)| \leq i$ car A est un indépendant.
   et $\forall j,\ k+1 \leq j \leq n$, $|N_j(A')| \leq |N_j(A)| + 1 = |N_j(B)| \leq j$
   Donc $\forall i,\ 1 \leq i \leq n$, $|N_i(A')| \leq i$.

Conséquences les indépendants de tailles maximum (i.e. les bases) ont exactement le même nombre de tâches dans les délais (ainsi que de tâches hors-délais).

Rappel : le glouton sur un matroïde

Glouton standard sur un matroïde $M = (E, \mathcal{F})$

$X \leftarrow \emptyset$;

tant que $|A| \neq \emptyset$ **faire**

  $A \leftarrow \{x \in E$ tel que $X \cup \{x\} \in \mathcal{F}\}$;

  Choisir $a \in A$ de poids maximum ;

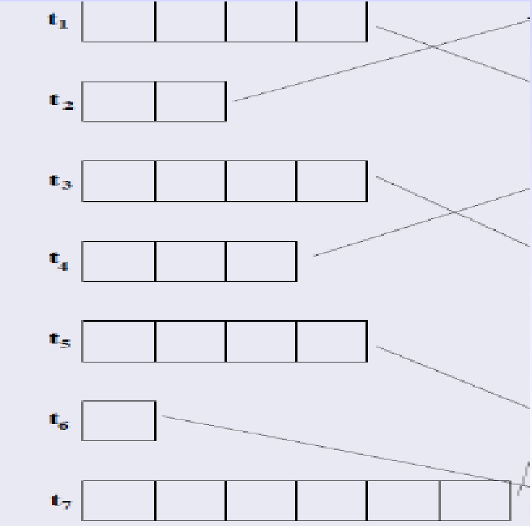
  $X \leftarrow X \cup \{a\}$;

fin

Appliquons le dans le cas de l'ordonnancement

$X \leftarrow \emptyset$;
tant que $|A| \neq \emptyset$ **faire**
    $A \leftarrow \{x \in E$ tel que $X \cup \{x\} \in \mathcal{F}\}$;
    Choisir $a \in A$ de pénalité maximale ;
    $X \leftarrow X \cup \{a\}$;
fin

Le glouton sur l'exemple initial



On peut ajouter t_1, t_2, t_3, t_4
mais on ne pas ajouter t_5 et t_6
par contre on peut ajouter t_7.
$\{t_1, t_2, t_3, t_4, t_7\}$ est un indépendant de taille maximale.
Un ordonnancement associée est $t_2, t_4, t_1, t_5, t_7, t_5, t_6$ optimal et de pénalité 50

Au contraire des deux gloutons précédemment présentés on n'affecte pas les tâches à des créneaux, on essaye tant que possible d'ajouter une nouvelle tâche quitte à réordonner l'ordonnancement.