Corrigé du TD 1

15 septembre 2010

1. Il suffit de considérer la suite d'objets suivants : (poids, intérêt)$= (2, 3), (1, 2), (1, 2)$.

Dans un sac de capacité 2, l'algorithme glouton reposant sur l'intérêt, choisira d'emporter l'objet 1, alors qu'il est préférable d'emporter les objets 2 et 3.

Dans un sac de capacité 3, l'algorithme glouton basé sur le rapport intérêt/poids choisira d'emporter les objets 2 et 3, tandis que 2 et 1 (ou 3 et 1) serait préférable ; il en est de même pour l'algorithme glouton basé sur le poids.

2. On vérifie les axiomes de matroïde. La partie $\mathcal{F}$ est non vide (elle contient $\emptyset \subset F$), et manifestement si $F \in \mathcal{F}$, alors pour tout $G \subset F$, $G \in \mathcal{F}$.

Vérifions l'axiome d'extension. Soit un ensemble de tâches $X = T_{i_1}, T_{i_2}, \ldots, T_{i_k}$ réalisable dans cet ordre et un autre ensemble $Y = T_{j_1}, T_{j_2}, \ldots, T_{j_l}$ réalisable dans cet ordre avec $l > k$. Soit T_{j_p} la tâche la plus tardive de Y qui n'est pas dans X. L'ensemble de tâche obtenu en ajoutant T_{j_p} à X est réalisable, dans l'ordre X'', T_{j_p}, Y' où $Y' = T_{j_{p+1}}, \ldots, T_{j_l}$ est l'ensemble des tâches qui se trouvent après T_{j_p} dans Y (et qui sont aussi des tâches de X), et où X'' est la suite des tâches restantes de X.

Pour vérifier que cet ordre est bien réalisable, on remarque que :

- les tâches de X" sont effectuées en un temps qui est inférieur ou égal à celui où elles l'étaient dans X,
- les tâches T_{j_p} et dans Y' sont effectuées en un temps qui est inférieur ou égal à celui où elles l'étaient dans Y,

On a donc bien prouvé que $\mathcal{F}$ définit un matroïde. Il suffit donc d'appliquer l'algorithme glouton pour les valuations auxquelles on s'intéresse, ici les pénalités : la fonction d'intérêt à maximiser est la somme des pénalités des tâches effectuées.

3. **i.** La stabilité par passage au sous-ensemble est claire. Soient F et G deux sous-ensembles constitués de colonnes linéairement indépendantes et tels que $|F| < |G|$. Comme $\dim \mathrm{Vect}(F) < \dim \mathrm{Vect}(G)$, il existe une colonne C de G n'appartenant pas à $\mathrm{Vect}(F)$. On obtient que $F \cup \{C\}$ est également un indépendant.

ii. Montrons que les ensembles formés de colonnes linéairement indépendantes correspondent aux ensembles d'arêtes du graphe ne comportant pas de cycles, i.e. aux forêts du graphe.

Si un ensemble d'arêtes forme un cycle simple alors la somme des vecteurs colonnes associés à ces arêtes dans la matrice d'incidence est nulle. Réciproquement supposons qu'il existe une relation de liaison $\sum a_k C_k = 0$ où $(C_k)_{k \in K}$ est une famille de colonnes qui correspond à un ensemble d'arêtes sans cycle. Soit i une des feuilles de cette forêt, il existe un unique $k_i \in K$ tel que $C_{k_i}(i) \neq 0$, ce qui implique $a_{k_i} = 0$. Par récurrence, on obtient $a_k = 0$ pour tout $k \in K$.

4. i. L'ensemble des sous-ensembles de X qui sont indépendants forme un matroide (le vérifier). Pour trouver un indépendant maximum il suffit donc d'appliquer l'algorithme glouton en donnant 1 comme valuation à chaque élément.

ii. Si X n'est pas un indépendant il n'existe pas de base le contenant. Sinon l'ensemble des parties Y du complément de X dans M telles que $X \cup Y$ soit un indépendant forme un matroïde (le vérifier). De nouveau il suffit donc d'appliquer l'algorithme glouton.

5. i. La suite T_1, T_4, T_2, T_3 donne $F_1 = 3$, $F_4 = 7$, $F_2 = 12$, $F_3 = 19$, soit une pénalité totale de $3 \cdot 6 + 12 \cdot 11 + 9 \cdot 19 + 5 \cdot 7 = 356$.

La suite T_4, T_1, T_3, T_2 donne $F_1 = 7$, $F_2 = 19$, $F_3 = 14$, $F_4 = 4$, soit une pénalité de $7 \cdot 6 + 19 \cdot 11 + 9 \cdot 14 + 5 \cdot 4 = 397$, le premier choix est donc le meilleur.

ii. Si ces deux tâches se suivent, soit T le temps écoulé avant la première des deux. Alors si i précède j, la pénalité sera $(T + d_i)p_i + (T + d_i + d_j)p_j$; dans le cas contraire la pénalité sera $(T + d_j)p_j + (T + d_j + d_i)p_i$. La différence de ces deux valeurs vaut $d_i p_j - d_j p_i$. Il s'ensuit que i doit précéder j si $d_i p_j - d_j p_i < 0$, soit encore $d_i/p_i < d_j/p_j$.

iii. Supposons que dans l'affectation optimale, les tâches ne soient pas effectuées par d/p croissant. Il existe alors deux tâches $t = (d, p)$ et $t' = (d', p')$ avec $d/p > d'/p'$ telle que t est effectuée juste avant t'. La question **ii.** montre alors qu'inverser t et t' améliore l'affectation, contradiction. Il suffit donc d'utiliser l'algorithme glouton en ordonnant les tâches par ordre d/p croissant. Sur l'exemple de l'énoncé on obtient T_2, T_1, T_3, T_4 pour une pénalité de 333.