TD : Algorithmes gloutons

Olivier Raynaud

raynaud@isima.fr
http ://www.isima.fr/raynaud

Exercice 1 (Un routier prudent)**.**
*Un routier conduit son camion d'Amsterdam à Lisbonne sur l'Européenne E*10*. Son réservoir, plein, lui permet de faire n kilomètres. Il dispose d'une carte routière indiquant les distances entre les différentes stations services qui jalonnent son parcourt. Le routier pressé désire effectuer le moins d'arrêts possibles.*

Question 1. *Proposer une méthode efficace pour choisir les stations services.*

Question 2. *Démontrer que votre stratégie aboutit à une solution optimale.*

Exercice 2 (Ensemble de points sur une droite)**.**
Etant donné un ensemble $\{x_1, x_2, ..., x_n\}$ *de points sur une droite, le problème consiste à déterminer le plus petit ensemble d'intervalles fermès de longueur* 1 *qui contient tous les points donnés.*

Question 1. *Proposer une méthode efficace pour choisir votre ensemble d'intervalles.*

Question 2. *Démontrer la validité de votre algorithme.*

Exercice 3 (Gardien de musée)**.**
Considérons une galerie de peinture rectiligne où les tableaux sont placés aux positions $\{x_1, x_2, ..., x_n\}$ *avec* $\forall i, x_i$ *réel. Un gardien peut surveiller des tableaux qui se trouvent à une distance au plus égale à* 1 *de lui-même (à gauche et à droite).*

Question 1. *Concevez un algorithme qui calcule le nombre minimal de gardiens requis pour surveiller tous les tableaux. Analyser la complexité et justifier l'optimalité de la solution.*

Exercice 4 (Le problème du Havresac)**.**

Un voleur dévalise un antiquaire et y trouve n objets, le ième valant v_i euros et pesant w_i kilogrammes (avec v_i et w_i tout deux entiers). L'individu, maigrichon ne pourra porter plus de w kilogrammes dans son havresac. Quels objets devra-t-il choisir pour maximiser son profit ?

Ce problème plus connu sous le nom de problème du sac à dos admet deux variantes.

- *variante du 'tout ou rien' : Chaque objet peut être choisi une fois et une seule fois et de manière indivisible.*
- *variante dite 'fractionnaire' : on suppose qu'un objet est sécable et donc que sa valeur peut être répartit uniformément sur son poids. Un lingot d'or peut être découpé en deux parties, la valeur de chacune d'elles dépendant de leur poids respectif.*

Question 2. *Expliquer que les deux variantes admettent la propriété de sous structure optimale.*

Question 3. *Proposer un algorithme glouton pour la variante dite 'fractionnaire'. Montrer ensuite que votre stratégie admet la propriété de choix glouton.*

Question 4. *Montrer sur un exemple simple que la stratégie gloutonne étudiée précédemment n'est pas adaptée à la variante du 'tout ou rien'.*

Question 5. *Supposons maintenant, dans le cadre de la variante du 'tout ou rien', que l'ordre des objets lorsqu'ils sont triés par ordre de poids croissant soit le même que lorsqu'ils sont triés par ordre de valeur décroissante. Donner un algorithme pour trouver une solution optimale et montrer pourquoi votre algorithme est correct.*

Exercice 5 (Affectations de tâches sur des machines)**.**
Etant donné un ensemble de n tâches de durées respectives $\{t_1, t_2, ..., t_n\}$, et m le nombre de machines. L'objectif est d'affecter les n tâches sur les m machines telle que le temps de complétion soit minimal.

On propose l'algorithme suivant : on liste les tâches dans un ordre quelconque, -et on assigne à chaque étape la tâche à la machine à laquelle on a donné le moins de travail.

Question 1. *Montrer que cet algorithme réalise un facteur d'approximation de 2.*

Question 2. *Considérer l'instance composée de m machines et $m^2 + 1$ tâches avec $t_i = 1$ pour $1 \leq i \leq m^2$ et $t_{m^2+1} = m$. Que peut-on en déduire ?*

Exercice 6 (Allocation de registres)**.**
Un programme informatique stocke ses variables dans des mémoires appelées registre. Un des objectifs d'un compilateur est de minimiser le nombre de registres utilisés lors de l'exécution d'un programme. Ainsi si deux variables ne sont pas utilisées en même temps, il est possible de les allouer à un même registre. Pour chacune des variables du programme, on calcule la première et dernière date où la variable est utilisée. On dit que la variable est active pendant l'intervalle de temps défini par ces deux dates. La figure 1 illustre les intervalles d'activité d'un programme comprenant sept variables $a, b, ..., g$.

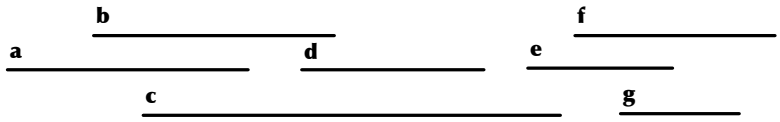

FIGURE 1 – Intervalles d'activités des variables

Un graphe d'intervalles $G = (S, A)$ est défini à partir d'une famille d'intervalles de la façon suivante : les sommets de S sont les intervalles d'activités associés à chaque variable ; il existe une arête $[a, b]$ de A si les intervalles correspondant s'intersectent.

Question 1. *Tracer le graphe d'intervalles correspondant à l'exemple de la figure 1*

Une $k-$coloration des sommets d'un graphe $G = (S, A)$ est une fonction C de $\{1, ..., k\}$ telle que pour toute arête $[a, b] \in A$ on a $C(a) \neq C(b)$. La plus petite valeur k telle qu'il existe une $k-$coloration de G est appelé nombre chromatique de G et est noté $\chi(G)$.

Question 2. *Montrer que minimiser le nombre de registres utilisés correspond à calculer $\chi(G)$ pour le graphe d'intervalles correspondant aux intervalles d'activités des variables du programme.*

Une clique est un sous ensemble de sommets $S' \subseteq S$ d'un graphe $G = (S, A)$ tel que pour toute paire de sommets a et b de la clique S' l'arête $[a, b]$ appartient à A. Pour un graphe G, la taille d'une plus grande clique est notée $w(G)$.

Question 3. *Montrer que pour tout graphe on a $w(G) \leq \chi(G)$.*

Question 4. *Proposer un algorithme glouton qui fournit une $k-$coloration d'un graphe d'intervalle $G = (S, A)$. Appliquer votre algorithme au graphe de la figure 1.*

Soit k le plus grand numéro de couleur obtenu après l'application de l'algorithme précédent à un graphe d'intervalles G et soit x un sommet de couleur k.

Question 5. *Montrer que x a au moins $k-1$ sommets voisins.*

Soit t la date de début de l'intervalle correspondant au sommet x.

Question 6. *Montrer que $w(G) \geq k$.*

Question 7. *Déduire des questions précédentes que pour les graphes d'intervalles, l'algorithme glouton fournit une coloration optimale.*

Exercice 7 (Pièces de monnaies)**.**

Considérons le problème consistant à rendre la monnaie de S euros avec le moins de pièces possibles à prendre dans un jeu de pièces $(a_1, a_2, ..., a_n)$.

Question 1. *Supposons que l'on dispose de pièces de 1, 2, 5 et 10 euros. Concevoir un algorithme glouton qui donne une solution optimale.*

Question 2. *Supposons que l'on dispose de pièces de $c^0, c^1, ..., c^k$ pour $c > 1$ et $k \geq 1$. Concevoir un algorithme glouton qui donne la solution optimale.*

Question 3. *Donner un ensemble de pièces tel que l'algorithme glouton proposé précédemment ne retourne pas la solution optimale et un autre pour lequel il ne donne pas de solution.*

Question 4. *Concevoir un algorithme qui retourne une solution optimale pour n'importe quel ensemble de pièces.*

Question 5. *Insérer dans votre algorithme les instructions pour afficher 'Rendre k pièce de type i' pour $k \geq 0$ et $0 \leq i \leq n$.*

Exercice 8 (Recouvrement par sous-ensembles)**.**

Afin de réaliser un produit une entreprise doit acquérir tous les objets d'un ensemble de n éléments $A = \{e_1, ..., e_n\}$, ces objets ne sont pas vendus séparément mais par paquets constituant un sous-ensemble S_i de E. Chacun de ces paquets à un coût c_i. La difficulté réside dans le calcule du coût minimal permettant d'acquérir tous les objets.

Question 1. *Proposer plusieurs solutions pour l'instance : $S_1 = \{1, 2\}$, $S_2 = \{3, 4\}$, $S_3 = \{1, 3, 4\}$, $S_4 = \{1, 4\}$ et $S_5 = \{2, 4\}$ avec les coûts $c_1 = 5, c_2 = 3, c_3 = 5, c_4 = 4$ et $c_5 = 2$.*

Question 2. *Proposer un algorithme glouton pour le problème décrit. Appliquer l'algorithme à l'instance donnée. Dire si l'algorithme calcule la solution optimale.*

Exercice 9 (Stockage)**.**
Soient n enregistrements de longueur variable stockés dans un fichier avec accès séquentiel. L'enregistrement e_i est de longueur s_i. On suppose que le temps d'accès au ième enregistrement est proportionnel à la somme des longueur des i premiers. Par exemple pour deux enregistrements e_1 et e_2, s'ils sont stockés dans l'ordre (e_2, e_1) le temps d'accès à l'enregistrement e_2 est $c.s_2$ et le temps d'accès pour e_1 est $c.(s_1 + s_2)$, c étant une constante. Pour chaque enregistrement i on connaît la fréquence d'accès f_i (on a $\sum_{i=1}^{n} f_i = 1$). Le temps moyen d'accès sera donc :

$$M = c.\sum_{i=1}^{n} (f_{\sigma(j)}.\sum_{k=1}^{j} s_{\sigma(k)})$$

Question 1. *Proposer une méthode gloutonne de stockage qui donnera toujours un temps d'accès optimal.*

Exercice 10 (Problème d'ordonnancement)**.**

On suppose que l'on doit réaliser des tâches T_1, T_2,...,T_n sur une seule machine. Chaque tâche T_i a une durée d_i et une priorité p_i. Une réalisation de ces tâches est donnée par une permutation T_{i1},T_{i2},...,T_{in} de celles-ci, représentant l'ordre dans lequel elles sont effectuées sur la machine.

La date de fin F_i d'une tâche T_i est donnée par la somme des durées des tâches qui la précèdent dans la permutation augmentée de sa durée propre d_i. On mesure la ***pénalité*** *d'une réalisation par la quantité $\sum_{i=1}^{n} p_i.F_i$ qui devra être rendue minimum. Ainsi les tâches de priorités plus grandes devraient être réalisées avant celles de priorité plus petite.*

Question 1. *On suppose qu'il y a 4 tâches ayant pour durée 3, 5, 7 et 4, et de priorités 6, 11, 9 et 5. Laquelle de ces deux réalisations est la meilleure : T_1,T_4,T_2,T_3 ou T_4,T_1,T_3,T_2 ?*

Question 2. *Soient deux tâches T_i et T_j de pénalité p_i et p_j. A quelle condition sur ces objets est-on sûr de l'ordre dans lequel il faut exécuter les tâches T_i et T_j ?*

Question 3. *Déduire de la question précédente un algorithme glouton et prouver votre algorithme.*

Question 4. *Exécuter votre algorithme sur l'exemple de la question 1.*

Exercice 11 (Des limites de la gloutonnerie)**.**

Cet exercice a pour but de montrer que de nombreux problèmes ne sont pas résolubles par un algorithme glouton. Considérons par exemple celui dit du sac à dos. On doit emporter en voyage un certain nombre d'objets parmi n, l'objet i a un poids p_i et un intérêt a_i. On est limité par le poids total P des objets à emporter et on cherche donc un sous-ensemble d'objets dont la somme des intérêts est maximale parmi ceux dont la somme des poids est inférieure ou égale à P.

1. *On applique l'algorithme consistant à classer les objets par intérêt décroissant et à considérer les objets itérativement dans cette liste ; on emporte ceux dont le poids ne fait pas dépasser le poids total P acceptable. Montrer en donnant un exemple que cet algorithme ne donne pas toujours l'optimum.*
2. *On applique ensuite l'algorithme consistant à classer les objets par poids croissant et à procéder ensuite comme pour l'algorithme précédent. Donner un nouvel exemple montrant que cet algorithme ne donne pas l'optimum.*
3. *On applique enfin l'algorithme consistant à classer les objets suivant le rapport (intérêt/ poids) et à poursuivre comme plus haut. Montrer à nouveau que cet algorithme ne donne pas l'optimum.*