Problèmes d'optimisation

Questions préliminaires

1°) Qu'est-ce qu'un algorithme glouton ?

2°) Quels sont les principes de la programmation dynamique ?

Exercice 1 : *Le problème du ravitaillement*

Le problème du ravitaillement est le suivant :

— On s'apprête à faire un trajet dans le désert, pour lequel on a besoin, pour chaque jour de trajet, d'une certaine quantité d'eau et on dispose d'une outre de capacité C (exprimée en jours de trajet).

— Tout au long de ce trajet se trouvent des oasis, qui se trouvent situées à des distances respectives $d_1, \ldots, d_n$ du point de départ, exprimées en jours de trajet — on suppose que $\forall i (d_{i+1} - d_i \leqslant C)$.

— Dans chaque oasis, on a la possibilité de faire le plein d'eau, auquel cas on doit vider son outre, ce qui, pour décourager le gaspillage, est facturé à un prix $P(k)$, où k est la quantité d'eau (exprimée en jours de trajet) restant dans l'outre. On peut ensuite remplir l'outre gratuitement.

Le but du problème est de déterminer dans quelles oasis s'arrêter pour minimiser le prix total payé en arrivant à la dernière oasis.

Dans cet exercice, on suppose que $P(k) = k$.

1°) Proposer un algorithme glouton pour déterminer où s'arrêter.

2°) On dit qu'un ensemble d'arrêts $\{i_1, i_2, \ldots, i_k\}$ est *compatible* si, en s'arrêtant à ces arrêts, on ne tombe jamais en panne d'eau. Donner une définition formelle de cette propriété de compatibilité. On notera également P la fonction qui, à un ensemble compatible, associe le coût du trajet. Pourquoi P est-elle bien définie ?

3°) Soient $S = \{i_1, i_2, \ldots, i_k\}$ un ensemble d'arrêts compatibles, et soit g_1 le premier arrêt proposé par l'algorithme glouton. Montrer que $S' = \{g_1, i_2, \ldots, i_k\}$ est compatible, et que $P(S') \leqslant P(S)$.

4°) En déduire que l'algorithme glouton est optimal.

Exercice 2 : *Le problème du ravitaillement (suite)*

Dans cet exercice, on suppose que les oasis ont changé leur politique tarifaire, et que maintenant $P(k) = k^2$ (toutes choses égales par ailleurs).

1°) L'algorithme glouton est-il encore optimal ?

2°) Proposer une décomposition du problème en sous-problèmes et exprimer une relation de dépendance entre sous-problèmes.

3°) En déduire un algorithme résolvant ce problème.

4°) En traçant sur un graphique la relation de dépendance entre sous-problèmes, évaluer le nombre de sous-problèmes qu'il sera nécessaire d'évaluer pour calculer une solution pour le problème global.

Sous-arbres d'un graphe

Questions préliminaires

1°) Qu'est-ce qu'un graphe ? et un arbre ?

2°) Quelle relation y a-t-il entre le nombre de sommets et le nombre d'arêtes d'un arbre ?

Exercice 1 : ***Arbre couvrant minimal***

Étant donné un graphe $G = (V, E)$ connexe, un arbre couvrant est un arbre $T = (V, E')$ avec $E' \subseteq E$ (i.e. un arbre ayant les mêmes sommets que le graphe et des arêtes prises parmi celles du graphe). Si le graphe est pondéré par une fonction de poids $w : E \to \mathbb{R}^+$, on peut chercher un arbre couvrant de poids minimal, i.e. minimisant $\sum_{e \in E'} w(e)$.

L'algorithme de Kruskal est un algorithme résolvant ce problème :

(*i*) On trie les arêtes par poids croissant et on commence avec l'ensemble vide pour E'.

(*ii*) On ajoute une arête de poids minimal à E'.

(*iii*) Tant que l'on n'a pas un arbre couvrant, on supprime toutes les arêtes qui relient deux sommets déjà reliés l'un à l'autre par les arêtes de E' et on reprend à l'étape (*ii*).

1°) De quel type d'algorithme s'agit-il ? Montrer qu'il produit bien un arbre couvrant.

2°) Montrer que le poids de l'arbre obtenu est minimal.

Indication : on pourra montrer par récurrence que si E'_k est l'ensemble des arêtes retenues à la k^{e} étape de l'algorithme, alors il existe un arbre couvrant minimal T_k contenant les arêtes d'E'_k.

Exercice 2 : ***Arbres couvrants et arbres de chemins***

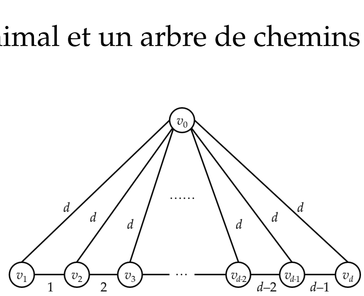
Étant donné un graphe pondéré $G = (V, E, w)$ connexe, et un sommet $r \in V$, un arbre de chemins minimaux enraciné en r est un arbre $T = (V, E')$ avec $E' \subseteq E$ (i.e. un arbre ayant les mêmes sommets que le graphe et des arêtes prises parmi celles du graphe) vérifiant que, pour tout sommet $u \in V$, le chemin (unique) de r à u dans l'arbre T est le chemin de poids minimal reliant r à u dans le graphe G

1°) Montrer qu'en effet, dans un arbre, étant donnés deux sommets quelconques u et v, il existe un unique chemin de u à v.

2°) Y a-t-il un rapport entre le fait d'être un arbre couvrant minimal et un arbre de chemins minimaux ?

3°) On considère T un arbre couvrant du graphe G suivant :

$$G = \left(\begin{array}{l} \{v_0, v_1, v_2, \ldots, v_d\}, \\ \{\{v_i, v_{i+1}\} \mid i \in [\![1, d[\![\} \cup \{\{v_0, v_i\} \mid i \in [\![1, d]\!]\}, \\ w : \left| \begin{array}{rcll} \{v_0, v_i\} & \mapsto & d & (\text{pour } i \in [\![1, d]\!]) \\ \{v_i, v_{i+1}\} & \mapsto & i & (\text{pour } i \in [\![1, d[\![) \end{array} \right. \end{array} \right)$$



Montrer que si v_0 a deux voisins dans T, alors T n'est pas minimal.

4°) Appliquer l'algorithme de Kruskal à G. Quel est le résultat ? Combien G a-t-il d'arbres couvrants minimaux ?

5°) Pour $i, j \in [\![0, d]\!]$, quel est le plus court chemin entre v_i et v_j ?

6°) On suppose $d > 9$. Montrer que, pour tout T arbre couvrant minimal de G et pour tout $r \in V$, T n'est pas un arbre de chemins minimaux enraciné en r.