Projet Algorithmes et Complexité
Partie 2

Remplissage d'un réservoir

Dans cette partie on s'intéresse au problème de remplissage d'un réservoir d'essence à différents arrêts aux stations-service lors d'un long voyage. On dispose d'un véhicule dont le réservoir a une capacité donnée U. Pour éviter la transformation de la capacité du réservoir exprimée en litre en distance à parcourir, on suppose que U exprime directement la distance que le véhicule peut parcourir avec un réservoir d'essence plein (on peut facilement l'obtenir par le produit de la taille du réservoir et du kilométrage par unité d'essence du véhicule). Le voyage commence à une ville de départ *s* et se termine à une ville d'arrivée *t* en parcourant des milliers de km. Le voyage commence avec une certaine quantité donnée d'essence μ ($\leq$ U) dans le réservoir. On suppose qu'à chaque station-service *v*, l'essence peut être acheté à un prix de *c(v)*. Ce prix est le coût de l'essence par kilomètre. Par exemple, si l'essence coûte 2.40€ le litre et que le véhicule peut parcourir 15 kilomètres par litre, alors le coût par kilomètre est de 16 cents. A chaque station-service, le remplissage du réservoir prolonge l'autonomie du véhicule d'une centaine de km. Par ailleurs, il existe une variation importante du prix de l'essence entre les stations-service de différentes régions, ainsi le coût du voyage dépend de la station-service où le plein est fait.

On s'intéresse alors au problème de station-service définit commet suit : Étant donné un nœud de départ *s* et un nœud cible *t*, comment y aller de *s* à *t* de la manière la moins chère possible avec au plus *K* arrêts aux stations-service si le trajet commence à *s* avec μ_s quantité d'essence ?

Plus formellement les données du problème sont : un graphe complet G = (V,E) avec des longueurs d'arêtes d : E → R^+, d(u,v) est la distance en km entre le sommet *u* et le sommet *v* et des coûts d'essence c : V → R^+ *c(u)* est le coût de l'essence (par kilomètre) au sommet *v*, et une capacité de réservoir U (distance totale que peut parcourir le véhicule avec un réservoir plein). L'objectif est d'aller d'une source *s* à une destination *t* de la manière la moins chère possible en utilisant au maximum *K* arrêts pour remplir le réservoir. On suppose que toutes les données du problème sont des entiers.

Pour faciliter la résolution du problème, on se concentre sur le cas où au départ de *s*, le réservoir est vide ce qui oblige un remplissage à *s*.

Q1. Montrer que le problème (P1) : un départ de *s* avec μ_s unités dans le réservoir, et le problème (P2) : un départ de *s* avec un réservoir vide sont équivalents. Pour la démonstration, on vous propose d'ajouter un nouveau nœud *s'* relié directement à *s*. Précisez le coût de l'essence au sommet *s'* et la distance *d(s', s)*, puis trouver, à l'optimum, le lien entre le nombre d'arrêts dans (P1) et le nombre d'arrêts dans (P2).

Q2. Montrer que la stratégie optimale consiste à arriver à destination *t* avec un réservoir vide.

Pour résoudre ce problème on utilise la programmation dynamique. Pour cela, on utilise la fonction coût suivante.

C[u, q, g] = Coût minimum du voyage pour partir de *u* à *t* en utilisant *q* arrêts pour remplir le réservoir, sachant qu'on arrive à u avec g unités d'essence (km) et *u* est l'un des *q* arrêts.

Q3. Exprimer le coût de la solution optimale en fonction de l'état final du programme dynamique en fixant les valeurs de u, q et g dans la fonction coût C[u, q, g]

Q4. Donner l'expression de la récurrence de C[u, q, g] en fonction de C[v,q-1, h]

Q5. Proposer une initialisation de la fonction coût C[u, q, g]

Q6. Donner la complexité de ce programme dynamique

La principale difficulté pour utiliser l'état C[u, q, g] provient du fait qu'en principe, on doit considérer chaque valeur de g ∈ [0,U]. Une façon d'éviter cela est de discrétiser les valeurs que g peut prendre.

Q7. Expliquer pourquoi même avec la discrétisation des valeurs de g l'utilisation de L'état C[u, q, g] mène à un programme dynamique de complexité pseudo-polynomial.

Pour éviter l'utilisation de toutes les valeurs discrètes de g dans l'intervalle [0 ; U], on regarde de près les caractéristiques de problème.

Soit s = $u_1, u_2, \ldots, u_l$ les arrêts successifs de remplissage du réservoir d'une solution optimale utilisant au plus k arrêts ($l \leq k$).

Q8. Montrer que la stratégie suivante est optimale pour décider de la quantité d'essence à remplir à chaque arrêt.

1. A l'arrêt u_l, remplir juste assez d'essence pour atteindre *t* avec un réservoir vide ;
2. Pour $j < l$
   i) Si $c(u_j) < c(u_{j+1})$, alors à u_j faire le plein
   ii) Si $c(u_j) \geq c(u_{j+1})$, alors à u_j mettre juste assez d'essence pour atteindre u_{j+1}

Cette stratégie optimale est utilisée comme suit :
Considérons un arrêt de remplissage u (avec u ≠ s) dans la solution optimale, et soit *w* l'arrêt juste avant *u*.
La stratégie optimale ci-dessus implique que si c(w) > c(u), alors on atteint *u* à partir de *w* avec un réservoir vide en empruntant le plus court chemin entre *w* et *u*, sinon on atteint *u* avec U - d(w, u) dans le réservoir. Par conséquent, dans la formulation du programme dynamique, on doit garder la trace d'au plus *n* valeurs différentes de l'essence pour *u*. Soit GV(u) l'ensemble de ces valeurs, à savoir,

GV (u) = {U − d(w, u) |w ∈ V et c(w) < c(u) et d(w, u) ≤ U} ∪ {0} avec d(w,u) est la distance du plus court chemin entre w et u.

Ainsi la récurrence suivante nous permet de calculer C[u, q, g] pour tout g ∈ GV(u) :

- <u>Initialisation</u>

$$C[u, 1, g] = \begin{cases} (d(u,t) - g).c(u) & \text{si } g \leq d(u,t) \leq U \\ \infty & \text{sinon} \end{cases}$$

- <u>Récurrence</u>

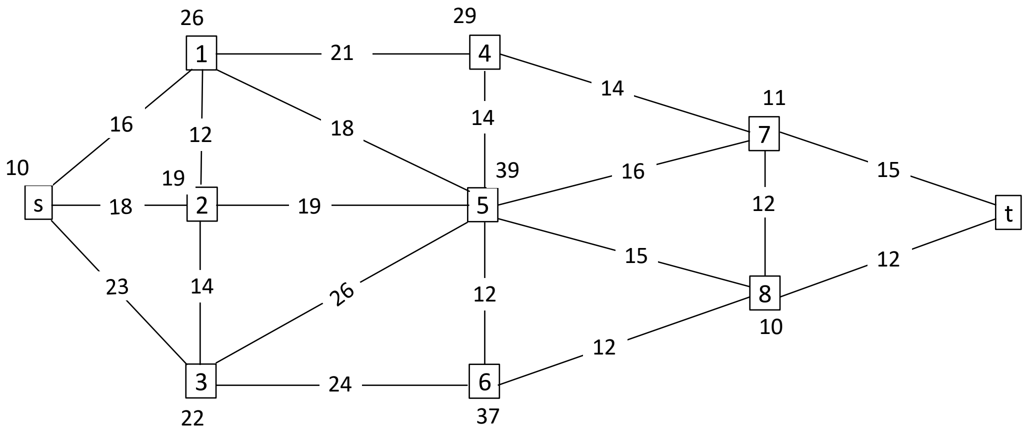
$$C[u, q, g] = min_{v \; tq \; d(u,v) \leq U} \begin{cases} C[v, q-1, 0] + (d(u,v) - g).c(u) & si \;\; c(v) \leq c(u) \text{ et } g \leq d(u,v) \\ C[v, q-1, U - d(u,v)] + (U - g).c(u) & \text{si} \;\; c(v) > c(u) \end{cases}$$

- <u>Solution optimale</u>

$$min_{1 \leq l \leq K} C[s, l, 0]$$

Q9. Expliquer votre compréhension des formules de récurrence (initialisation, récurrence, solution optimale)

Pour bien comprendre les formules du programme dynamique, soit l'exemple suivant d'un graphe où s est la ville de départ et t est la ville cible. Les valeurs sur les sommets sont le prix de l'essence au km (en centimes d'euro) et les valeurs sur les arrêts sont les distances (en km) entre les villes. Le véhicule utilisé dispose d'un réservoir de capacité U = 40km. On souhaite partir de s à t avec au plus 3 arrêts.



Q10. Pour l'exemple ci-dessus, donner l'ensemble GV(u) pour chaque $u \in V$. Pour le calcul des d(w,u) on utilise l'algorithme de Dijksra (plus court chemin).
Q11. Donner toutes les formules de calcul de C[5,2, g]
Q12. Écrire l'algorithme détaillé du programme dynamique
Q13. Proposer une implémentation de votre algorithme dans un langage de votre choix.
Q14. Exécuter votre programme sur l'exemple de graphe ci-dessus et fournir le coût de la solution optimale ainsi que l'itinéraire, les arrêts aux sommets (stations-service) et les quantités remplies à chaque arrêt.
Q15. Quelle est la complexité de ce programme dynamique (bien justifier votre réponse).