Routage multi-contraint

Correction

Question 1 On peut faire cette preuve par induction.

Notons $2-MCP$ le problème MCP avec $m = 2$. $2-MCP \in NP$. Montrons que le problème PARTITION se réduit au problème $2-MCP$.

Considérons une instance de PARTITION $(a_1, \ldots, a_n)$ et construisons un réseau avec $n+1$ nœuds et $2n$ liens de telle sorte qu'il y ait 2 liens entre i et $i+1$ $(1 \leq i < n)$ pondérés par 2 métriques d_1 et d_2 et tels que :

— un lien pondéré avec $d_1 = M$ et $d_2 = 0$,

— l'autre lien pondéré avec $d_1 = M - a_i$ et $d_2 = a_i$.

avec $M = 2nS$ et $S = \sum_{i=1}^{n} a_i$.

Considérons une instance de $2-MCP$ telle que $d_1(p) \leq nM - \frac{S}{2}$ et $d_2(p) \leq \frac{S}{2}$ où p est un chemin entre les nœuds 1 et n. On peut remarquer que pour chaque lien entre les nœuds i et $i+1$, $d_1 + d_2 = M$. Donc pour n'importe quel chemin p, on a $d_1(p) + d_2(p) = nM$.

On sait que que $d_1(p) \leq nM - \frac{S}{2}$. Donc $d_2(p) \geq \frac{S}{2}$. Comme on a $d_2(p) \leq \frac{S}{2}$, on a donc $d_2(p) = \frac{S}{2}$ (et donc $d_1(p) = nM - \frac{S}{2}$). Comme entre chaque nœud i et $i+1$, d_2 est égal à 0 ou a_i, on a donc un sous-ensemble de $(a_1, \ldots, a_n)$ dont la somme est égal à $\frac{S}{2}$. On a donc trouvé une solution au problème PARTITION.

Inversement, considérons une instance de PARTITION et un des sous-ensembles de cette partition. On construit le chemin p en choisissant, entre les nœuds i et $i+1$, le lien qui a le poids a_i si a_i est dans le sous-ensemble, sinon on choisit l'autre lien. Ainsi on a $d_2(p) = \frac{S}{2}$ et $d_1(p) = nM - \frac{S}{2}$. On a donc une solution au problème $2-MCP$.

Supposons maintenant que $n-MCP$ est NP-complet. Montrons que $n-MCP$ se réduit au problème $(n+1)-MCP$. Pour cela, considérons une instance p à $(n+1)-MCP$. On a

$$d_i(p) \leq L_i (1 \leq i \leq n+1) \tag{1}$$

Choisissons $L_{n+1} = \sum_{e \in E} d_{n+1}(e)$. Ainsi on sait que pour n'importe quel chemin p, on a $d_{n+1}(p) \leq L_{n+1}$. Enfin pour tout i $(1 \leq i \leq n)$, $d_i(p) \leq L_i$ si l'équation (1) est vraie.

Question 2 Il suffit d'enlever, dans le graphe, les arcs dont au moins un des poids concaves ne respecte pas la contrainte concave associée.

Question 3 Soit p une solution à MCP$(G = (V,E), w_1, w_2', L_1, x, s, k)$. On a $w_1(p) \leq L_1$ et $w_2'(p) \leq x$.

On sait que $w_2'^e = \lceil \frac{w_2^e . x}{L_2} \rceil$. Donc $w_2^e \leq \frac{L_2}{x} w_2'^e$.

Comme $w_2(p) = \sum_{e \in p} w_2^e$, on a $w_2(p) \leq \sum_{e \in p} \frac{L_2}{x} w_2'^e = \frac{L_2}{x} w_2'(p) \leq L_2$. Donc p est une solution à MCP$(G = (V,E), w_1, w_2, L_1, L_2, s, k)$.

Question 4 Considérons le graphe de gauche de la figure 1 avec les contraintes $L_1 = 8$ et $L_2 = 20$. Le chemin (s, u, v, t) est une solution à MCP$(G, w_1, w_2, 8, 20, s, t)$. Si on prend $x = 5$, le nouveau graphe avec les poids w_2' est le graphe de droite de la figure 1 . Le chemin (s, u, v, t) n'est pas une solution à MCP$(G, w_1, w_2', 8, 20, s, t)$.

Question 5 Si p est une solution à MCP$(G = (V,E), w_1, w_2, L_1, L_2, s, k)$, on a $w_1(p) \leq L_1$.

$w_2'(p) = \sum_{e \in p} w_2'^e = \sum_{e \in p} \lceil \frac{w_2^e . x}{L_2} \rceil < \sum_{e \in p} (\frac{w_2^e . x}{L_2} + 1) = \frac{x}{L_2} \sum_{e \in p} w_2^e + \sum_{e \in p} 1 = \frac{x}{L_2} w_2(p) + l < \frac{x}{L_2}(1 - \frac{l-1}{x}).L_2 + l < x + 1$.

Comme $w_2'(p)$ et x sont des entiers, on a donc $w_2'(p) \leq x$. Donc p est une solution à MCP$(G = (V,E), w_1, w_2', L_1, x, s, k)$.

Question 6 L'algorithme MCP trouve une solution à MCP$(G = (V,E), w_1, w_2, L_1, L_2, s, k)$ si l'ensemble des solutions à ce problème est non vide et s'il existe un chemin p dans cet ensemble tel que $w_2(p) \leq (1 - \frac{l-1}{x}).L_2$.

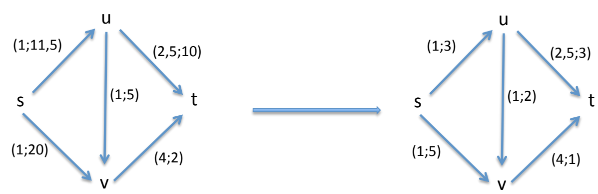

FIGURE 1 – Exemples pour la question 4

Algorithm 1 EDSP($G = (V, E), w_1, w'_2, x, s$)

for chaque sommet $v \in V$ **do**
  for $i = 0 \to x$ **do**
    $d[v, i] \leftarrow \infty$
    $\Pi[v, i] \leftarrow NIL$
  end for
end for
for $i = 0 \to x$ **do**
  $d[s, i] \leftarrow 0$
end for
$S \leftarrow \emptyset$
$Q \leftarrow \{(u, j) | u \in V, j \in \{0, \ldots, x\}\}$
while $Q \neq \emptyset$ **do**
  $(u, j) \leftarrow$ élément de Q ayant la plus petite valeur de d
  $S \leftarrow S \cup \{(u, j)\}$
  $Q \leftarrow Q - \{(u, j)\}$
  for chaque voisin v de u **do**
    $j' \leftarrow j + w'_2(u, v)$
    if $j' \leq x$ **then**
      if $d[v, j'] > d[u, j] + w_1(u, v)$ **then**
        $d[v, j'] \leftarrow d[u, j] + w_1(u, v)$
        $\Pi[v, j'] \leftarrow u$
      end if
    end if
  end for
end while

Question 7 L'algorithme 1 est dérivé de l'algorithme de Dijkstra.

Soient $u \in V$ et $k \in [1; x]$ entier. $d[u, k]$ correspond, à la fin de l'algorithme 1, au plus petit poids w_1 parmi les chemins allant de s à u et dont le poids $w'_2 = k$.