Examen final - Graphe et recherche opérationnelle

Télécom Nancy

Décembre 2019

Durée : 2 H

Barème : exercice 1 : 7 points, exercice 2 : 4 points, exercice 3 : 4 points, exercice 4 : 7 points.

Remarques :

— documents (sauf antisèche à rendre avec vos copies), calculatrices et téléphones portables interdits ;

— il sera tenu compte de la rigueur et de la clareté de vos réponses aux questions posées.

Exercice 1 : *Résolution d'un P.L. par l'algorithme du simplexe* (7 points)

Dans cet exercice l'algorithme du simplexe utilisera la stratégie suivante :

— la variable entrante correspond au coefficient réduit le plus négatif ;

— en cas d'ambiguité pour choisir une variable entrante ou sortante, on choisira la variable d'indice le plus petit ; l'ordre entre différents types de variables étant (1) variables initiales, (2) variables d'écart (ce qui veut dire que si on a le choix entre une variable initiale et une variable d'écart, c'est la variable initiale qui prime).

Ces règles **devront être impérativement respectées** pour obtenir les points correspondants.

On considère le programme linéaire suivant :

$$(P)\left\{\begin{array}{lll} \min & f(x) = -4x_1 - 8x_2 & \\ & -x_1 + 3x_2 & \leq \; 9 \\ & x_1 + x_2 & \leq \; 7 \\ & 2x_1 + x_2 & \leq \; 12 \\ & x \geq 0 & \end{array}\right.$$

1. Résoudre (P) graphiquement : donner les explications nécessaires sur votre copie mais faire le graphique sur le canevas donné à la dernière page de l'énoncé.
2. Mettre (P) sous forme standard (on notera e_1, e_2, e_3 les variables d'écart). Quelle solution de base réalisable de départ évidente peut-on utiliser pour résoudre (P) avec l'algorithme du simplexe ? (préciser quelles sont les variables de base et les variables hors base et donner le dictionnaire initial).
3. Dérouler l'algorithme du simplexe (**avec les règles mentionnées ci-avant**) pour trouver la solution de (P). Remarque : pour éviter les erreurs de calcul, écrire la trajectoire de l'algorithme dans le plan (x_1, x_2) sur le dessin de la première question : les différentes solutions de base réalisables visitées par l'algorithme $(x^{(0)}, e^{(0)})$, $(x^{(1)}, e^{(1)})$, $(x^{(2)}, e^{(2)}) = (x^{(*)}, e^{(*)})$ devant correspondre à des sommets du polygône...
4. A partir du dictionnaire final (on note B^* la base optimale et H^* son complémentaire) en choisissant un ordre pour les variables de base et hors-base (que l'on précisera) donner (la valeur) des matrices et vecteurs suivants :

$$A_{B^*}, \quad A_{H^*}, \quad \tilde{A}_*(= (A_{B^*})^{-1}A_{H^*}), \quad c_{B^*}, \quad c_{H^*}, \quad \tilde{b}_* = (A_{B^*})^{-1}b, \quad \tilde{c}_*,$$

où $\tilde{c}_*$ est le vecteur des coûts réduits. Il n'y a AUCUN calcul à faire !

Exercice 2 : *Flot max avec bornes inférieures* (4 points)

On considère un graphe $G = (E, \Gamma)$, munis de capacités min et max, $\forall (i,j) \in \Gamma\,;\ \alpha_{i,j} \leq \beta_{i,j}$, le graphe comporte un nœud source s ainsi qu'un nœud puits t et on cherche un flot réalisable $f : \Gamma \mapsto \mathbb{R}$ vérifiant donc $\alpha_{i,j} \leq f_{i,j} \leq \beta_{i,j}$.

1. Montrer qu'une condition nécessaire pour qu'il existe un tel flot f est :

$$\forall j \in E \setminus \{s,t\} \quad \sum_{i \in P(j)} \alpha_{i,j} \leq \sum_{k \in S(j)} \beta_{j,k} \;\text{ et }\; \sum_{k \in S(j)} \alpha_{j,k} \leq \sum_{i \in P(j)} \beta_{i,j}$$

2. Une fois vérifiée les conditions précédentes, on peut chercher un flot réalisable f en cherchant un flot maximum f' sur le graphe auxiliaire G' construit de la manière suivante :
   — on ajoute deux sommets s' et t' : $E' = E \cup \{s', t'\}$;
   — on ajoute des arêtes reliant s' aux sommets $j \neq s$ et des arêtes reliant les sommets $j \neq t$ à t' :

$$\Gamma' = \Gamma \cup \{(s', j), \forall j \in E, j \neq s\} \cup \{(j, t'), \forall j \in E, j \neq t\}$$

   — on relie t à s par un arc de capacité infinie (s et t sont confondus pour G') ;
   — les capacités sont données par :

$$c'_{ij} = \beta_{i,j} - \alpha_{i,j}, \quad \forall (i,j) \in \Gamma$$
$$c'_{s'j} = \sum_{i \in P(j)} \alpha_{i,j}, \quad \forall j \neq s$$
$$c'_{jt'} = \sum_{k \in S(j)} \alpha_{j,k}, \quad \forall j \neq t$$

   Pour que ce flot maximum puisse être un flot réalisable sur G (via $f_{i,j} = f'_{i,j} + \alpha_{i,j}$) il faut tester la saturation du flot f' sur toutes les arêtes (s', j) et toutes les arêtes (i, t'). Montrer qu'il suffit de tester la saturation de f' uniquement sur l'un ou l'autre de ces deux jeux d'arêtes, c'est à dire montrer l'équivalence suivante :

$$f'_{s',j} = c'_{s',j}, \forall j \in E \setminus \{s\} \iff f'_{i,t'} = c'_{i,t'}, \forall i \in E \setminus \{t\}$$

Exercice 3 : *Programmation linéaire en variables binaires* (4 points)

On considère le problème de programmation linéaire en variables $\{0, 1\}$ (problème de "sac-à-dos") :

$$\begin{cases} \max\,[F(\mathrm{x}) = 12x_1 + 3x_2 + 5x_3 + 6x_4] \\ \quad 7x_1 + 3x_2 + 2x_3 + 4x_4 \leq 9 \\ \quad 6x_1 + 3x_2 + 6x_3 + x_4 \leq 11 \\ \quad x_1, x_2, x_3, x_4 \in \{0, 1\} \end{cases}$$

Résoudre ce problème par une procédure de séparation et évaluation (Branch-and-bound), en respectant les consignes suivantes :

— Pour déterminer une solution réalisable initiale, examiner les variables par ordre **décroissant** des coefficients dans F.

— Dans la procédure d'évaluation, examiner les variables dans l'ordre x_1, x_2, x_3 puis x_4.

— Séparer toujours en premier le sous-ensemble qui a l'estimation principale la plus élevée.

Exercice 4 : *Sur l'algorithme A^** (7 points)

Soit $G = (E, \Gamma, \ell)$ un graphe valué orienté, la valuation $\ell(x, y)$ de chaque arc (x, y) correspondant à une longueur pour aller de x à y. Si x et x' sont deux sommets du graphe, on dit qu'il existe un chemin de x à x' si une suite d'arcs permet d'aller de x vers x'. Un tel chemin sera noté :

$$< x\ (= x^{(0)}),\ x^{(1)},\ \ldots,\ x^{(p-1)},\ x'\ (= x^{(p)}) >, \quad (x^{(k-1)}, x^{(k)}) \in \Gamma, \forall k \in \{1, \ldots, p\}$$

sa longueur étant $\sum_{k=1}^{p} \ell(x^{(k-1)}, x^{(k)})$. Si un tel chemin existe, on définit $\mathcal{L}(x, x')$ comme la longueur du plus court chemin de x vers x', sinon $\mathcal{L}(x, x') := +\infty$.

Pour trouver un plus court chemin d'un sommet de départ x_0 vers un sommet d'arrivée x_F, l'algorithme A^*, arrivé à un sommet[1] x utilise une estimation $\phi(x)$ de la distance du chemin $< x_0, \ldots, x, \ldots, x_F >$

1. L'algorithme a déjà trouvé un chemin de x_0 à x de longueur $L(x)$.

de la forme $\phi(x) = L(x) + H(x)$ où $H(x)$ est [2] une estimation de $\mathcal{L}(x, x_F)$, i.e. $H(x) \simeq \mathcal{L}(x, x_F)$. Le chemin déjà trouvé jusqu'à x peut être optimal (i.e. $L(x) = \mathcal{L}(x_0, x)$) ou pas ($L(x) > \mathcal{L}(x_0, x)$). On dit que l'heuristique H est admissible si :

$$H(x) \leq \mathcal{L}(x, x_F),\ \forall x \in E$$

et consistante si :

$$H(x) \leq H(y) + \ell(x, y),\ \forall (x, y) \in \Gamma$$

1. Montrer qu'une heuristique consistante est admissible.
2. On suppose que chaque sommet du graphe est associé à des coordonnées (en dimension 2,3,.., on note d la dimension). Pour alléger les notations, on confond un sommet x du graphe avec le point x de $\mathbb{R}^d$ de ses coordonnées. Soit $\|.\|$ une norme dans $\mathbb{R}^d$ (norme 1 (de Manhattan), norme 2 (euclidienne),...), on suppose aussi que $\ell(x, y) = \|x - y\|,\ \forall (x, y) \in \Gamma$. Montrer que H définie par $H(x) = \|x - x_F\|, \forall x \in E$ est une heuristique consistante.
3. La question suivante a pour but de montrer que si l'heuristique est consistante alors A^* donne effectivement un plus court chemin de x_0 à x_F (s'il existe). On rappelle l'algorithme générique :

> $S = \{x_0\}\,;\ \overline{S} = \emptyset\,;\ L(x_0) = 0\,;\ \phi(x_0) = H(x_0)$.
> **tant que** $S \neq \emptyset$ **:**
>   **Déterminer** $x \in S$ tq $\phi(x) = L(x) + H(x) = \min_{x' \in S} (\phi(x'))$
> $S = S \setminus \{x\}\,;\ \overline{S} = \overline{S} \cup \{x\}$
>   **si** $x = x_F$ **:** arrêt (solution trouvée)
>   **pour** $x' \in Succ(x)$ **:**
>   **si** $x' \notin S \cup \overline{S}$ **:**
> $S = S \cup \{x'\};\ L(x') = L(x) + \ell(x, x');\ \phi(x') = L(x') + H(x')$
>   **sinon si** $L(x) + l(x, x') + H(x') < \phi(x')$ **:**
> $L(x') = L(x) + l(x, x')\,;\quad \phi(x') = L(x') + H(x')$
>   **si** $x' \in \overline{S}$ **:**
> $S = S \cup \{x'\}$ et $\overline{S} = \overline{S} \setminus \{x'\}$

qui trouve un plus court chemin sous l'hypothèse moins forte d'admissibilité de l'heuristique [3]. Avec l'hypothèse de consistance (et en supposant aussi qu'il existe un chemin $< x_0, \dots, x_F >$) vous allez montrer qu'à toute itération k de l'algorithme, on a la propriété [4] :

$$\mathcal{P}(k) : \forall x \in \overline{S}, L(x) = \mathcal{L}(x_0, x).$$

(a) Montrer que $\mathcal{P}(1)$ est vraie.

(b) Montrer pour $k \geq 2$ (et tant qu'on n'est pas sorti de l'algorithme) que $\mathcal{P}(k)$ est vraie. *Aide : la preuve, par l'absurde, est similaire à celle de Dijkstra : on se place à la première itération k^* pour laquelle $\mathcal{P}(k^*)$ est fausse ce qui revient à dire que le nouveau sommet y intégré à $\overline{S}$ est tel que $L(y) > \mathcal{L}(x_0, y)$. Comme il existe un chemin de x_0 vers y, il existe un chemin optimal $< x_0, \dots, z, \dots, y >$ (différent de celui trouvé) où on note z le premier sommet n'appartenant pas à $\overline{S}$.*

(c) En déduire que l'algorithme trouve effectivement un plus court chemin $< x_0, \dots, x_F >$.

(d) Comment utiliser cette information dans l'algorithme générique pour le rendre un peu plus performant ?

---

2. Formellement H est une application de $E \to \mathbb{R}^+$ avec $H(x_F) = 0$.
3. La preuve est cependant plus difficile.
4. Dans cette propriété, on considère l'ensemble dynamique $\overline{S}$ **après** l'inclusion du sommet x de S vérifiant $\min_{x' \in S} \phi(x')$.