Méthodes et modélisation pour l'optimisation
M1 informatique

Examen – jeudi 19 janvier 2017
CORRECTION

Les instruments électroniques sont interdits. Une feuille recto-verso manuscrite autorisée. La durée de l'examen est de 2h. Ce sujet comporte 5 questions. La notation prendra en compte le soin apporté à la rédaction (pour de vrai !)

Exercice 1. Un agriculteur possède 10 hectares de terre à utiliser entièrement pour trois parcelles de raisin, de pommes de terre et de salade. Chaque culture nécessite un certain temps de travail et un investissement en capital et donne à terme un bénéfice différent par hectare. L'agriculteur possède un capital de 540 euros et 48 heures de travail. De quelle taille doit être chaque parcelle ?

culture	heures de travail / hectare	investissement euros / hectare	bénéfice net euros / hectare
raisin	9	54	60
p.d.t.	9	26	45
salade	3	27	30

a) Formuler ce problème comme un programme linéaire pour maximiser le revenu de l'agriculteur.

b) L'agriculture peut embaucher du personnel à 12 euros/heure. Comment pouvez-vous modifier votre programme pour déterminer le nombre d'heures pour lesquelles il devrait embaucher afin d'optimiser son revenu ?

Correction :

(a)

Les variables sont x_r, x_p, x_s : la taille de parcelle en hectares de raisins, de pommes de terre et de salades. On veut maximiser le bénéfice net. La contrainte $x_r + x_p + x_s \leq 10$ peut également être une égalité.

$$\begin{array}{rrl} \text{Maximiser} & 60x_r + 45x_p + 30x_s & \\ \text{sous les contraintes} & 9x_r + 9x_p + 3x_s & \leq 48 \\ & 54x_r + 26x_p + 27x_s & \leq 540 \\ & x_r + x_p + x_s & \leq 10 \\ & x_r, x_p, x_s & \geq 0 \end{array}$$

(b)

Introduire une variable H : le nombre d'heures que l'agriculteur devrait embaucher. Modifier les deux égalités concernant le temps de travail et l'investissement en capital :

$$\begin{aligned} 9x_r + 9x_p + 3x_s &\leq 48 + H \\ 54x_r + 26x_p + 27x_s &\leq 540 - 12H \end{aligned}$$

Exercice 2. On considère le programme linéaire suivant :

$$\begin{aligned} \text{Maximiser} \quad & 4x_1 + 3x_2 + 5x_3 \\ \text{sous les contraintes} \quad & 2x_1 - x_2 + 4x_3 \leq 18 \\ & 4x_1 + 2x_2 + 5x_3 \leq 10 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

a) Donner une forme équationnelle en introduisant des variables d'écart.

b) Appliquer l'algorithme du simplexe. En cas de choix, la variable du plus petit indice entre dans la base. On prendra soin d'indiquer à chaque étape la base visitée par l'algorithme et la valeur de la fonction objectif.

Correction :

(a)

Introduire deux variables d'écart : x_4, x_5 pour les deux contraintes.

$$\begin{aligned} \text{Maximiser} \quad & 4x_1 + 3x_2 + 5x_3 \\ \text{sous les contraintes} \quad & 2x_1 - x_2 + 4x_3 + x_4 = 18 \\ & 4x_1 + 2x_2 + 5x_3 + x_5 = 10 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

(b)

Solution initiale

- La base $\{x_4, x_5\}$ correspond à la solution $(0, 0, 0, 18, 10)$.

$$\begin{array}{rcrrrr} x_4 & = & 18 & -2x_1 & +x_2 & -4x_3 \\ x_5 & = & 10 & -4x_1 & -2x_2 & -5x_3 \\ \hline z & = & 0 & +4x_1 & +3x_2 & +5x_3 \end{array}$$

Pivot 1

- x_1 a le plus petit indice parmi les variables hors base ayant un coefficient positif, donc x_1 entre dans la base. x_5 sort de la base.
- La base $\{x_1, x_4\}$ correspond à la solution faisable basique $(\frac{5}{2}, 0, 0, 13, 0)$.

$$\begin{array}{rcrrrr} x_1 & = & \frac{5}{2} & -\frac{1}{2}x_2 & -\frac{5}{4}x_3 & -\frac{1}{4}x_5 \\ x_4 & = & 13 & +2x_2 & -\frac{3}{2}x_3 & +\frac{1}{2}x_5 \\ \hline z & = & 10 & +x_2 & & -x_5 \end{array}$$

Pivot 2

- x_2 entre dans la base. x_1 sort de la base.
- La base $\{x_2, x_4\}$ correspond à la solution faisable basique $(0, 5, 0, 23, 0)$.

$$\begin{array}{rcrlll} x_2 & = & 5 & -2x_1 & -\frac{5}{2}x_3 & -\frac{1}{2}x_5 \\ x_4 & = & 23 & -4x_1 & -\frac{13}{2}x_3 & -\frac{1}{2}x_5 \\ \hline z & = & 15 & -2x_1 & -\frac{5}{2}x_3 & -\frac{3}{2}x_5 \end{array}$$

- Les coefficients des variables hors base (x_1, x_3, x_5) sont négatifs, donc c'est fini.
- **Une solution optimale au programme initiale est :**
  $(x_1, x_2, x_3) = (0, 5, 0)$ avec l'optimum 15.

Exercice 3. Résoudre par l'algorithme DPLL les deux ensembles de clauses ci-dessous. Donner, dans les deux cas, une affection satisfaisante, s'il y en a une. On fera les choix de variable dans l'ordre lexicographique, $x_1, x_2, \ldots$ et on testera d'abord l'affectation à 1, puis à 0.

a) $\{ \quad (\neg x_1, x_2, x_3), \quad (x_2, \neg x_3), \quad (\neg x_2, x_3), \quad (\neg x_2, \neg x_3) \quad \}$

b) $\{ \quad (\neg x_1, \neg x_7), \quad (x_1, x_7), \quad (x_1, \neg x_3), \quad (\neg x_1, x_3, x_9), \quad (\neg x_2, \neg x_4, x_9),$
$(\neg x_2, \neg x_5), \quad (x_2, x_5), \quad (x_3, x_6, x_8), \quad (\neg x_3, \neg x_9), \quad (\neg x_3, x_6),$
$(x_3, \neg x_6, x_7), \quad (x_3, \neg x_8), \quad (\neg x_3, x_8), \quad (x_4, \neg x_5), \quad (\neg x_4, \neg x_6, \neg x_9),$
$(x_4, \neg x_7), \quad (\neg x_4, x_7), \quad (x_4, x_9), \quad (x_6, \neg x_7), \quad (x_6, x_9) \quad \}$

Correction :

(a)

$$F = \{(\neg x_1, x_2, x_3), (x_2, \neg x_3), (\neg x_2, x_3), (\neg x_2, \neg x_3)\}$$

- Pile : vide
  On applique la PU : $F_1 = PU(F)$ ne modifie rien
  Choix de variable : x_1
- Pile : $x_1 = 1$
  On applique la PU : $F_2 = PU(F_1 \cup \{(x_1)\}) = \{(x_2, x_3), (x_2, \neg x_3), (\neg x_2, x_3), (\neg x_2, \neg x_3)\}$
  Choix de variable : x_2
- Pile : $x_1 = 1$, $x_2 = 1$
  On applique la PU : $F_3 = PU(F_2 \cup \{(x_2)\}) = \{()\}$
  PU affecte $x_3 = 1$
  F_3 contient la clause vide, donc on remonte la pile.

- Pile : $x_1 = 1$, $x_2 = 0$
  On applique la PU : $F_4 = PU(F_2 \cup \{(\neg x_2)\}) = \{()\}$
  PU affecte $x_3 = 1$
  F_4 contient la clause vide, donc on remonte la pile.

- Pile : $x_1 = 0$
  On applique la PU : $F_5 = PU(F_1 \cup \{(\neg x_1)\}) = \{(x_2, \neg x_3), (\neg x_2, x_3), (\neg x_2, \neg x_3)\}$
  Choix de variable : x_2

- Pile : $x_1 = 0$, $x_2 = 1$
  On applique la PU : $F_6 = PU(F_5 \cup \{(x_2)\}) = \{()\}$
  F_6 contient la clause vide, donc on remonte la pile.

- Pile : $x_1 = 0$, $x_2 = 0$
  On applique la PU : $F_7 = PU(F_5 \cup \{(\neg x_2)\}) = \{\}$
  PU affecte $x_3 = 0$
  F_7 est vide, donc on répond SAT. Une affectation satisfaisante :

$$x_1 = x_2 = x_3 = 0$$

(b)

$$F = \{ \text{ beaucoup de clauses } \}$$

- Pile : vide
  On applique la PU : $F_1 = PU(F) = F$ ne modifie rien
  Choix de variable : x_1

- Pile : $x_1 = 1$
  On applique la PU : $F_2 = PU(F_1 \cup \{(x_1)\}) = \{()\}$
  PU affecte (par ex.) : $x_7 = 0, x_4 = 0, x_5 = 0, x_2 = 1, x_9 = 1, x_3 = 0, x_6 = 0, x_8 = 1$
  F_2 contient la clause vide, donc on remonte la pile.

- Pile : $x_1 = 0$
  On applique la PU : $F_3 = PU(F_1 \cup \{(\neg x_1)\}) = \{\}$
  PU affecte $x_7 = 1, x_3 = 0, x_8 = 0, x_6 = 1, x_4 = 1, x_9 = 0, x_2 = 0, x_5 = 1$
  F_3 est vide, donc on répond SAT. Une affectation satisfaisante :

$$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 1, x_6 = 1, x_7 = 1, x_8 = 0, x_9 = 0$$

Exercice 4. Un technicien d'un centre de ressources informatiques est en train de planifier les tâches hebdomadaires qui doivent tourner sur les deux machines du centre. Le technicien connaît le nombre de tâches, n, et possède une liste des dépendances entre les tâches ; cette liste spécifie que certaines tâches ne peuvent pas être effectuées en même temps (sur des machines différentes). Une machine ne peut pas traiter plus d'une tâche à la fois et chaque tâche est prévue de tourner pendant une journée sur une machine. Donc, le technicien a 7 créneaux sur chaque machine auxquels il peut affecter les tâches.

a) Proposer une modélisation en SAT de ce problème. Décrire soigneusement les variables et les clauses introduites. Expliquer comment une affectation aux variables peut être interprétée comme un planning hebdomadaire des deux machines.

b) Maintenant on suppose que certaines tâches (marquées "multi-journée" sur la liste du technicien) doivent tourner 2 ou plusieurs jours sur une même machine (mais peuvent être interrompues par une autre tâche et redémarrées une ou plusieurs journées plus tard).

Exemple : une tâche "multi-journée" avec une durée de 2 jours peut être planifiée sur la machine 1 pour lundi et mercredi mais pas sur la machine 1 pour lundi et sur la machine 2 pour mercredi.

▷ Décrire comment modéliser la planification des tâches multi-journées.

Vous avez droit d'utiliser les 'macros' AtLeast *et* AtMost *introduites en cours.*

Correction :

(a)

Soit $1, 2, \ldots, n$ les tâches. Introduire une variable $x_{(m,c),t}$, $m = 1, 2$, $c = 1, \ldots, 7$, $t = 1, 2, \ldots, n$ avec l'interprétation : "la tâche t est affectée à la machine m au créneau c".

Contraintes :

- AtLeast-1$(x_{t,(1,1)}, \ldots, x_{t,(2,7)})$ et
  AtMost-1$(x_{t,(1,1)}, \ldots, x_{t,(2,7)})$ pour tout $t = 1, \ldots, n$
  "toute tâche est affectée à exactement une machine et un créneau"

- AtMost-1$(x_{1,(m,c)}, \ldots, x_{n,(m,c)})$ pour $m = 1, 2$, $c = 1, \ldots, 7$
  "au plus une tâche est affectée sur chaque machine à chaque créneau"

- Si la liste de dépendances spécifie que les tâches i et j ne peuvent pas être effectuées en même temps, alors
  AtMost-1$(x_{i,(1,c)}, x_{j,(2,c)})$ et AtMost-1$(x_{i,(2,c)}, x_{j,(1,c)})$ pour tout $c = 1, \ldots, 7$

(b)

On découpe toute multi-tâche T en k tâches standards $t_1, \ldots, t_k$ où k est la durée (en jours) de T. S'il y a une dépendance entre T et une autre tâche t, alors, on ajoute des dépendances entre t_1 et t, entre t_2 et t, ...et entre t_k et t sur la liste des dépendances. Ensuite, on introduit les mêmes variables et contraintes que pour la question (a).

Pour éviter que deux tâches t_i et t_j appartenant à une même multi-tâche T tournent sur deux machines différentes, on introduit une variable $y_{T,m}$, $m = 1, 2$ avec l'interprétation *"la multi-tâche T va tourner sur la machine m"* et on introduit les contraintes suivantes :

- $\neg x_{t_i,(m,c)} \vee y_{T,m}$ pour toute machine $m = 1, 2$, tout créneau $c = 1, \ldots, 7$ et toute multi-tâche T et tâche t_i appartenant à T
  "si t_i est affecté à la machine m au créneau c, alors la multi-tâche T va tourner sur la machine m"

- AtMost-1$(y_{T,1}, y_{T,2})$ pour toute multi-tâche T
  "une multi-tâche doit tourner sur une seule machine"

Exercice 5. L'entreprise Priceler fabrique des camionnettes et des camping-cars. La demande pour chaque type de véhicule pour les 12 mois qui viennent est connue : pour le mois t, la demande est de $d_{a,t}$ camionnettes et de $d_{b,t}$ camping-cars.

Supposer que l'entreprise veut répondre à la demande chaque mois. Le coût de fabrication d'une camionnette est de c_a euros et celle d'un camping-car est de c_b euros. Les véhicules qui ne sont pas vendus le mois de leur construction peuvent être stockés dans un entrepôt. Le coût de stockage d'un mois à l'autre est de s_a euros par camionnette et s_b euros par camping-car. Chaque mois, au plus K véhicules peuvent être construits. Au début du premier mois, il y a e_a camionnettes et e_b camping-cars dans l'entrepôt.

Formuler ce problème comme un programme linéaire qui minimise les coûts de Priceler pendant les 12 mois qui viennent.

Correction :

Soit $x_{a,t}$ et $y_{b,t}$ le nombre de véhicules à construire dans le mois $t = 1, \ldots, 12$.

Soit $y_{a,t}$ et $y_{b,t}$ le nombre de véhicules dans l'entrepôt au début du mois t.

$$\text{Minimiser} \sum_{t=1}^{12} c_a \cdot x_{a,t} + c_b \cdot x_{b,t} + s_a \cdot y_{a,t} + s_b \cdot y_{b,t}$$

$$\begin{aligned}
y_{a,t} + x_{a,t} &\geq d_{a,t} && t = 1, \ldots, 12 \\
y_{b,t} + x_{b,t} &\geq d_{b,t} && t = 1, \ldots, 12 \\
y_{a,t+1} &= y_{a,t} + x_{a,t} - d_{a,t} && t = 1, \ldots, 11 \\
y_{b,t+1} &= y_{b,t} + x_{b,t} - d_{b,t} && t = 1, \ldots, 11 \\
x_{a,t} + x_{b,t} &\leq K && t = 1, \ldots, 12 \\
y_{a,1} &= e_a \\
y_{b,1} &= e_b \\
x_{a,t}, x_{b,t}, y_{a,t}, y_{b,t} &\geq 0 && t = 1, \ldots, 12
\end{aligned}$$

Fin du sujet