Méthodes et modélisation pour l'optimisation
M1 informatique

Examen – lundi 08 janvier 2018 14h00-16h00

Une feuille recto-verso manuscrite autorisée. Calculatrices, ordinateurs et téléphones portables interdits. Ce sujet comporte 5 questions. La notation prendra en compte le soin apporté à la rédaction.

Exercice 1. La ville de Jötunheim produit 4500 tonnes de déchets par jour. Il y a deux modes d'élimination de ces déchets : par incinération ou par mise en décharge. La ville dispose de trois incinérateurs (A, B et C) avec les caractéristiques du tableau ci-dessous.

		émissions par tonne incinérée	
incinérateur	capacité tonnes / jour	unités de dioxyde de soufre	unités de particules
A	1 200	250	20
B	800	150	30
C	1 000	220	24

La mise en décharge est beaucoup plus chère que l'incinération, donc la ville aimerait minimiser la quantité (en tonnes / jour) de déchets mis en décharge. Malheureusement, l'état a imposé des restrictions pour réduire les polluants des trois incinérateurs. Les émissions totales de dioxyde de soufre sont limitées à 400 000 unités par jour et les émissions totales de particules sont limitées à 50 000 unités par jour.

a) Formuler ce problème comme un programme linéaire.

b) Les techniciens ont réussi à augmenter la capacité de l'incinérateur B de 5%. En revanche, les tonnes de déchets au-delà de la capacité normale imposent des émissions de dioxyde qui sont 10% plus importantes que la normale. Modifier votre programme pour prendre en compte cette nouvelle donnée.

Correction :

(a)

Les variables sont x_A, x_B, x_C : le nombre de tonnes / jour de déchets incinérés dans chaque incinérateur.

$$\begin{array}{rrl}
\text{Minimiser} & 4500 - (x_A + x_B + x_C) & \\
\text{sous les contraintes} & 250x_A + 150x_B + 220x_C & \leq 400000 \\
& 20x_A + 30x_B + 24x_C & \leq 50000 \\
& x_A & \leq 1200 \\
& x_B & \leq 800 \\
& x_C & \leq 1000 \\
& x_A, x_B, x_C & \geq 0
\end{array}$$

(b)

Introduire une variable y_B : le nombre de tonnes de déchets au-delà de la capacité normale incinérés.

$$\begin{array}{rrl} \text{Minimiser} & 4500 - (x_A + x_B + y_B + x_C) & \\ \text{sous les contraintes} & 250x_A + 150x_B + 165y_B + 220x_C & \leq 400000 \\ & 20x_A + 30x_B + 30y_B + 24x_C & \leq 50000 \\ & x_A & \leq 1200 \\ & x_B & \leq 800 \\ & y_B & \leq 40 \\ & x_C & \leq 1000 \\ & x_A, x_B, y_B, x_C & \geq 0 \end{array}$$

On ne peut pas imposer la contrainte (non-linéaire) $y_B > 0 \implies x_B = 800$. Ce n'est pas non plus nécessaire, parce que dans une solution optimale, il est toujours plus favorable d'utiliser toute la capacité normale de l'incinérateur B avant de mettre y_B à non-zéro.

Exercice 2. On considère le programme linéaire suivant :

$$\begin{array}{rrl} \text{Maximiser} & 3x_1 + 2x_2 + 4x_3 & \\ \text{sous les contraintes} & x_1 + x_2 + 2x_3 & \leq 4 \\ & 2x_1 + 3x_3 & \leq 5 \\ & 2x_1 + x_2 + 3x_3 & \leq 7 \\ & x_1, x_2, x_3 \geq 0 & \end{array}$$

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.

$$\begin{array}{rll} \text{Maximiser} & 3x_1 + 2x_2 + 4x_3 & \\ \text{sous les contraintes} & x_1 + x_2 + 2x_3 + x_4 & = 4 \\ & 2x_1 + 3x_3 + x_5 & = 5 \\ & 2x_1 + x_2 + 3x_3 + x_6 & = 7 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 & \end{array}$$

(b)

Solution initiale

- La base $\{x_4, x_5, x_6\}$ correspond à la solution $(0, 0, 0, 4, 5, 7)$.

$$\begin{array}{rclrrr} x_4 & = & 4 & -x_1 & -x_2 & -2x_3 \\ x_5 & = & 5 & -2x_1 & & -3x_3 \\ x_6 & = & 7 & -2x_1 & -x_2 & -3x_3 \\ \hline z & = & 0 & +3x_1 & +2x_2 & +4x_3 \end{array}$$

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 $\{x_1, x_4, x_6\}$ correspond à la solution basique admissible $(\frac{5}{2}, 0, 0, \frac{3}{2}, 0, 2)$.

$$\begin{array}{rcrrrr} x_1 & = & \frac{5}{2} & & -\frac{3}{2}x_3 & -\frac{1}{2}x_5 \\ x_4 & = & \frac{3}{2} & -x_2 & -\frac{1}{2}x_3 & +\frac{1}{2}x_5 \\ x_6 & = & 2 & -x_2 & & +x_5 \\ \hline z & = & \frac{15}{2} & +2x_2 & -\frac{1}{2}x_3 & -\frac{3}{2}x_5 \end{array}$$

Pivot 2

- x_2 entre dans la base. x_4 sort de la base.
- La base $\{x_1, x_2, x_6\}$ correspond à la solution faisable basique $(\frac{5}{2}, \frac{3}{2}, 0, 0, 0, \frac{1}{2})$.

$$\begin{array}{rcrrrr} x_1 & = & \frac{5}{2} & -\frac{3}{2}x_3 & & -\frac{1}{2}x_5 \\ x_2 & = & \frac{3}{2} & -\frac{1}{2}x_3 & -x_4 & +\frac{1}{2}x_5 \\ x_6 & = & \frac{1}{2} & +\frac{1}{2}x_3 & +x_4 & +\frac{1}{2}x_5 \\ \hline z & = & \frac{21}{2} & -\frac{3}{2}x_3 & -2x_4 & -\frac{1}{2}x_5 \end{array}$$

- Les coefficients des variables hors base (x_3, x_4, x_5) sont négatifs, donc c'est fini.
- **Une solution optimale au programme initiale est :**
  $(x_1, x_2, x_3) = (\frac{5}{2}, \frac{3}{2}, 0)$ avec l'optimum 21/2.

Exercice 3. Résoudre par l'algorithme DPLL l'ensemble de clauses ci-dessous. Donner 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.

$$\begin{array}{llllll} \{ & (x_1 \vee \neg x_2), & (\neg x_1 \vee \neg x_3 \vee x_5), & (x_2 \vee x_5), & (\neg x_2 \vee \neg x_7), & (\neg x_2 \vee x_4 \vee x_6), \\ & (x_3 \vee x_4), & (\neg x_4 \vee x_6), & (\neg x_5 \vee x_2), & (\neg x_5 \vee \neg x_6), & (\neg x_6 \vee \neg x_4) \quad \} \end{array}$$

Correction :

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

- 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 : $F_2 = PU(F_1 \cup \{(x_1)\}) = \{(\neg x_3 \vee x_5), (x_2 \vee x_5), (\neg x_2 \vee \neg x_7), (\neg x_2 \vee x_4 \vee x_6), (x_3 \vee x_4), (\neg x_4 \vee x_6), (\neg x_5 \vee x_2), (\neg x_5 \vee \neg x_6), (\neg x_6 \vee \neg x_4)\}$
  Choix de variable : x_2

- Pile : $x_1 = 1$, $x_2 = 1$
  On applique la PU : $F_3 = PU(F_2 \cup \{(x_2)\}) = \{(\neg x_3 \vee x_5), (x_4 \vee x_6), (x_3 \vee x_4),$ $(\neg x_4 \vee x_6), (\neg x_5 \vee \neg x_6), (\neg x_6 \vee \neg x_4)\}$
  PU affecte $x_7 = 0$
  Choix de variable : x_3

- Pile : $x_1 = 1$, $x_2 = 1$, $[x_7 = 0]$, $x_3 = 1$
  On applique la PU : $F_4 = PU(F_3 \cup \{(x_3)\}) = \{()\}$
  F_4 contient la clause vide, donc on remonte la pile.

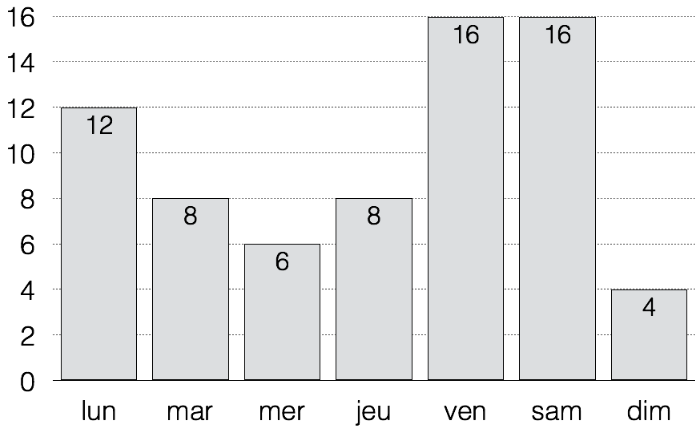
- Pile : $x_1 = 1$, $x_2 = 1$, $[x_7 = 0]$, $x_3 = 0$
  On applique la PU : $F_5 = PU(F_3 \cup \{(\neg x_3)\}) = \{()\}$
  F_5 contient la clause vide, donc on remonte la pile.

- Pile : $x_1 = 1$, $x_2 = 0$
  On applique la PU : $F_6 = PU(F_2 \cup \{(\neg x_2)\}) = \{()\}$
  F_6 contient la clause vide, donc on remonte la pile.

- Pile : $x_1 = 0$
  On applique la PU : $F_7 = PU(F_1 \cup \{(\neg x_1)\}) = \{()\}$
  F_7 contient la clause vide, donc on remonte la pile.

- Pile : vide
  La pile est devenue vide, donc F **n'est pas satisfaisable**.

Exercice 4. Une compagnie de bus doit affecter un nombre de chauffeurs aux jours de la semaine. Le nombre de bus simultanément en circulation chaque jour est représenté dans le diagramme ci-dessous. Un chauffeur doit travailler quatre jours consécutifs par semaine. Par exemple, si un chauffeur commence sa semaine vendredi, alors il travaille vendredi, samedi, dimanche et lundi. Si une journée, il y a plus de chauffeurs que de bus en circulation, alors un ou plusieurs chauffeurs resteront stationnaires. On suppose que l'emploi du temps reste le même d'une semaine sur l'autre.



▷ Proposer une modélisation en SAT pour établir un emploi du temps qui couvre les bus en circulation chaque jour et utilise au plus 20 chauffeurs. Décrire soigneusement les variables de décision et les clauses introduites.

Vous avez droit d'utiliser les 'macros' AtLeastOne/AtLeast-k *et* AtMostOne/AtMost-k.

Correction : Introduire les variables $x_{i,j}$ pour $i = 1, \ldots, 20$ (les chauffeurs) et $j = 1, \ldots, 7$ (les jours de la semaine). La variable $x_{i,j} = 1$ si le chauffeur i *commence ses quatre jours de travail consécutifs* le jour j.

Contraintes pour assurer que $x_{i,j}$ définit une affectation $f : \{1, \ldots, 20\} \to \{1, \ldots 7\}$:

- AtLeastOne$(x_{i,1}, x_{i,2}, \ldots, x_{i,7})$ pour tout $i = 1, \ldots, 20$
- AtMostOne$(x_{i,1}, x_{i,2}, \ldots, x_{i,7})$ pour tout $i = 1, \ldots, 20$

Contraintes pour assurer un nombre suffisant de chauffeurs chaque jour. Il faut qu'au moins le nombre minimum de chauffeurs a commencé ses semaines de travail pendant les quatre derniers jours.

- AtLeast-$12(x_{1,5}, x_{1,6}, x_{1,7}, x_{1,1}, x_{2,5}, x_{2,6}, x_{2,7}, x_{2,1}, \ldots, x_{20,5}, x_{20,6}, x_{20,7}, x_{20,1})$ (lundi)
- AtLeast-$8(x_{1,6}, x_{1,7}, x_{1,1}, x_{1,2}, x_{2,6}, x_{2,7}, x_{2,1}, x_{2,2}, \ldots, x_{20,6}, x_{20,7}, x_{20,1}, x_{20,2})$ (mardi)

  ...
- AtLeast-$4(x_{1,4}, x_{1,5}, x_{1,6}, x_{1,7}, x_{2,4}, x_{2,5}, x_{2,6}, x_{2,7}, \ldots, x_{20,4}, x_{20,5}, x_{20,6}, x_{20,7})$ (dimanche)

Solution alternative : si on veut plutôt utiliser des variables $y_{i,j}$ qui valent 1 si le chauffeur i *travaille* le jour j, alors il faut trouver un moyen d'exprimer qu'il s'agit de quatre jours consécutifs. Par exemple :

- AtLeast-$4(y_{i,1}, y_{i,2}, \ldots, y_{i,7})$ pour tout $i = 1, \ldots, 20$
- AtMost-$4(y_{i,1}, y_{i,2}, \ldots, y_{i,7})$ pour tout $i = 1, \ldots, 20$
- $(y_{i,j} \wedge y_{i,j+2}) \to y_{i,j+1} \iff \neg y_{i,j} \vee y_{i,j+1} \vee \neg y_{i,j+2}$
  pour tout $i = 1, \ldots, 20$ et $j = 1, \ldots 7$ (indices modulo 7)

Pour voir que ces contraintes suffisent, on argumente comme suit. Si le chauffer i ne travaille pas quatre jours consécutifs, alors il n'a pas non plus trois jours de repos consécutifs. Cela implique qu'un jour de repos doit être coincé entre deux jours de travail. Mais les dernières contraintes interdisent cette possibilité. Donc, si toutes les contraintes sont satisfaites, alors le chauffeur travaille quatre jours consécutifs.

Ensuite, les contraintes pour assurer le nombre minimum de chauffeurs se simplifient.

- AtLeast-$12(y_{1,1}, y_{2,1}, \ldots, y_{20,1})$ (lundi)
- AtLeast-$8(y_{1,2}, y_{2,2}, \ldots, y_{20,2})$ (mardi)

  ...
- AtLeast-$4(y_{1,7}, y_{2,7}, \ldots, y_{20,7})$ (dimanche)

Exercice 5. Considérer un marché avec I fournisseurs, J acheteurs et K produits qui sont achetés et vendus. Le fournisseur i possède une quantité de S_{ik} du produit k et le vend pour un prix de A_{ik} euros par unité. L'acheteur j demande au plus D_{jk} unités du produit k et accepte d'acheter pour un prix de B_{jk} euros par unité.

Dans ce marché, vous êtes en charge de coupler les fournisseurs et les vendeurs : vous collectez toutes les informations de prix et d'offres et ensuite, vous affectez les fournisseurs aux acheteurs. Votre bénéfice est l'écart entre le prix d'achat et le prix de vente pour chaque unité du produit k que le fournisseur i vend à l'acheteur j.

a) Formuler le problème de maximiser le bénéfice comme un programme linéaire.

b) Supposer désormais que les fournisseurs et les vendeurs ne sont pas au même endroit. La capacité de transport entre le fournisseur i et le vendeur j est au plus U_{ij} unités. De plus, vous êtes obligé de payer le transport de tous les produits vendus par le fournisseur i au vendeur j, pour un coût de C_{ij} euros par unité. Modifier votre programme pour prendre en compte ces nouvelles conditions.

Correction : Introduire les variables x_{ijk} pour $i = 1, \dots, I$, $j = 1, \dots, J$, $k = 1, \dots, K$. La variable x_{ijk} indique la quantité du produit k vendue par le fournisseur i à l'acheteur j.

(a)

$$\text{Max} \quad \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J}(A_{ik} - B_{jk})x_{ijk}$$

$$\begin{aligned}
\sum_{j=1}^{J} x_{ijk} &\leq S_{ik} \quad \text{pour tout } i = 1, \dots, I, k = 1 \dots, K \\
\sum_{i=1}^{I} x_{ijk} &\leq D_{jk} \quad \text{pour tout } j = 1, \dots, J, k = 1 \dots, K \\
x_{ijk} &\geq 0
\end{aligned}$$

(b)

$$\text{Max} \quad \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J}(A_{ik} - B_{jk} - C_{ij})x_{ijk}$$

$$\begin{aligned}
\sum_{j=1}^{J} x_{ijk} &\leq S_{ik} \quad \text{pour tout } i = 1, \dots, I, k = 1 \dots, K \\
\sum_{i=1}^{I} x_{ijk} &\leq D_{jk} \quad \text{pour tout } j = 1, \dots, J, k = 1 \dots, K \\
\sum_{k=1}^{K} x_{ijk} &\leq U_{ij} \quad \text{pour tout } i = 1, \dots, I, j = 1 \dots, J \\
x_{ijk} &\geq 0
\end{aligned}$$

Fin du sujet