Prof. F. Eisenbrand EPFL - DISOPT

Optimisation Discrète

Adrian Bock Semestre de printemps 2011

Série 2 3 mars 2011

Remarque générale :
Pour obtenir un bonus pour l'évaluation finale, vous pouvez rendre des solutions écrites à l'exercice noté avant (le début de) la séance de la semaine prochaine. On peut obtenir deux points bonus cette semaine. L'exercice 2 est une question de programmation. Les exercices à rendre sont les mêmes pour tous les étudiants.
Le rendu peut être fait en groupe de trois personnes au plus.

Exercice 1 $(*), (\Delta)$

Résoudre le programme linéaire

$$\max\{c^T x \colon x \in \mathbb{R}^n,\ Ax = b\}$$

avec

(i) $A = \begin{pmatrix} 4 & -2 & 6 & 1 & -2 \\ -1 & 3 & -2 & 1 & 2 \\ 4 & 8 & 4 & 6 & 4 \end{pmatrix}$, $b = \begin{pmatrix} 7 \\ -4 \\ 2 \end{pmatrix}$ et $c^T = (2, 4, 5, 1, 3)$

(ii) $A = \begin{pmatrix} 4 & 2 & 0 & 3 & 1 \\ 6 & 4 & 2 & 3 & 5 \\ 3 & 5 & 2 & 1 & 1 \end{pmatrix}$, $b = \begin{pmatrix} 7 \\ 11 \\ 6 \end{pmatrix}$ et $c^T = (1, 1, 1, 0, 2)$

(iii) A, b comme (ii), mais $c^T = (1, 1, 1, 2, 0)$

Dans chaque cas, indiquer si le PL est admissible et s'il est borné.

Remarque : Considérer le noyau de la matrice A.

Solution

(i) Le programme linéaire n'est pas admissible, donc il n'y a pas de solutions. On peut le voir facilement comme expliqué dans l'exercice 3 avec $\lambda^T = (\frac{1}{2}, 1, -\frac{1}{4})$.

(ii) Le programme linéaire est admissible et borné. Donc il y a des solutions optimales.
D'abord on trouve que le noyau de la matrice A est engendré par $u^T = (-10, 0, 8, 13, 1)$ et $v^T = (0, 10, 24, -9, 7)$. Une solution admissible du programme linéaire est par exemple $x^T = (1, 0, 1, 1, 0)$. On peut alors écrire toutes les solutions admissibles sous la forme $x + \alpha u + \mu v$. Or, l'objectif

c est orthogonal à u et v. Donc on ne peut pas augmenter la valeur de l'objectif $c^T x = 2$ en ajoutant des multiples de u ou v. Toutes les solutions admissibles sont aussi optimales et le programme linéaire est borné.

(iii) Le programme linéaire est admissible, mais non-borné.
Comme au point précédent, on trouve le noyau et la description de toutes les solutions admissibles. Par contre, l'objectif c n'est pas orthogonal à u (ou v). Donc on peut augmenter l'objectif jusqu'à une valeur arbitraire si on considère une solution $x+\alpha u$ avec $\alpha > 0$ et on obtient $c^T(x+\alpha u) = 2+24\alpha$.

Exercice 2 $(*, \Delta)$

Modifier l'élimination de Gauss-Jordan vue en cours de telle sorte que la matrice obtenue soit sous la forme échelonnée réduite en lignes et que l'inverse U^{-1} de la matrice U de transvection soit aussi calculée par l'algorithme.
Si vous voulez, vous pouvez utiliser l'exemple `gauss2.py` disponible sur la site web du cours. Noter que Python n'admet qu'une seule valeur de retour. Donc on peut donner une liste des deux matrices U, U^{-1}, par exemple.

Vous devez rendre votre solution non seulement sur papier, mais aussi de façon électronique (un programme SAGE), par mail à adrianaloysius.bock [at] epfl.ch avant le début du cours du 10 mars.

Solution

```
def gauss2(A):
    U = matrix(QQ,A.nrows(),A.nrows())
    for i in range(A.nrows()):
        U[i,i]=1

    U1 = deepcopy(U)

    k = 1
    for j in range(1, A.ncols()+1):
        i=k
        while(i <= A.nrows() and A[i-1,j-1] == 0):
            i = i+1
        if (i <= A.nrows() ):
            U.swap_rows(k-1,i-1)
            U1.swap_columns(k-1,i-1)
            A.swap_rows(k-1,i-1)
            U.rescale_row(k-1, 1/A[k-1,j-1])
            U1.rescale_col(k-1, A[k-1,j-1])
            A.rescale_row(k-1, 1/A[k-1,j-1])


            for i in range(k+1,A.nrows()+1):
                U1.add_multiple_of_column(k-1,i-1,A[i-1,j-1])
                U.add_multiple_of_row(i-1,k-1,-A[i-1,j-1])
                A.add_multiple_of_row(i-1,k-1,-A[i-1,j-1])


            for i in range(1,k):
                U1.add_multiple_of_column(k-1,i-1,A[i-1,j-1])
```

```
            U.add_multiple_of_row(i-1,k-1,-A[i-1,j-1])
            A.add_multiple_of_row(i-1,k-1,-A[i-1,j-1])

        k = k+1

    return (U,U1)
```

Exercice 3

Démontrer que le système linéaire suivant n'admet pas de solution.

$$\begin{pmatrix} 2 & 0 & 1 & 3 \\ -1 & 3 & 1 & 1 \\ 3 & 3 & 3 & 7 \end{pmatrix} x = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix}$$

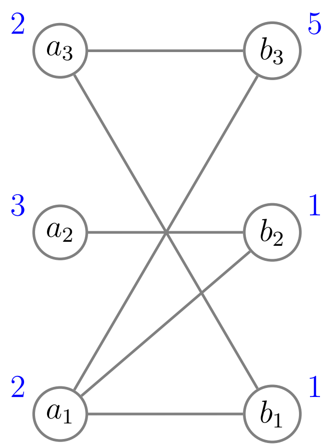
En autres termes, donner une combinaison des lignes pour obtenir une contradiction comme montré pendant le cours.

Solution

Le vecteur $\lambda^T = (2, 1, -1)$ donne la contradiction voulue, car $\lambda^T A = (0, 0, 0, 0)$ et $\lambda^T b = -1$.

Exercice 4

Considérons le graphe biparti $G = (\mathcal{A} \cup \mathcal{B}, E)$ avec $E \subset \mathcal{A} \times \mathcal{B}$ représenté à droite. On donne des poids c positifs sur les nœuds. Nous cherchons un transversal[1] dans G tel que la somme des poids des nœuds choisis soit minimisée.



Démontrer que $C = \{a_1, b_2, a_3\}$ est une solution optimale.

Indication :

Le programme linéaire suivant décrit le problème
(à voir plus tard dans le Cours)

$$\begin{array}{rcll} \min & \displaystyle\sum_{v \in \mathcal{A} \cup \mathcal{B}} c_v x_v & & \\ x_v + x_w & \geq & 1 & \forall e \in E \\ x_v & \geq & 0 & \forall v \in \mathcal{A} \cup \mathcal{B} \end{array}$$

1. Rappelons qu'un transversal (ou couverture de sommets) d'un graphe $G = (V, E)$ est un sous-ensemble de sommets $S \subseteq V$ contenant au moins une extrémité de chaque arête.

Utiliser l'idée expliquée dans l'exercice 2 de la série 1 !
$\Big(\lambda = (1, 0, 1, 1, 0, 2, 0, 2, 0, 0, 0, 2)\Big)$

Solution

On réécrit le programme linéaire avec la notation matricielle et on obtient une matrice A de taille 12×6, où les dernières 6 lignes constituent la matrice identité $\mathbf{I_6}$, et un vecteur b tel que les premières 6 composantes sont 1 et les autres sont zéro.
On trouve facilement

$$\lambda^T Ax = c^T x \geq \lambda^T b = 5,$$

donc la fonction de l'objectif est minorée par 5. Or, la solution donnée (est admissible et) obtient une valeur 5. Donc elle est optimale.