École normale supérieure **2008-2009**
Département d'informatique

Algorithmique et Programmation
TD n° 9 : Programmation Linéaire
Avec Solutions

Exercices théoriques

Rappel : Dual d'un programme linéaire cf. le poly `http://www.di.ens.fr/notes-lp.eps` pages 14-17.

Lemme de Farkas

Exercice 1. Dans cet exercice, on considère des programmes linéaires de la forme $Ax = b$ avec $x \geq 0$ et on ne s'intéresse qu'à savoir s'il existe une solution réalisable, i.e. un tel x sans se préoccuper de maximiser une fonction objectif.

Soient A une matrice $m \times n$ et b un vecteur de taille m. Pour le système $Ax = b,\ x \geq 0$, démontrer le lemme de Farkas qui dit qu'une seule des affirmations suivantes est vraie :

1. il existe un x tel que $Ax = b,\ x \geq 0$.
2. il existe y tel que $A^T y \geq 0$ mais $b^T y < 0$.

Solution : cf. le poly précédent page 10-13.

Adéquation linéaire

Exercice 2. Un physicien prend des mesures d'une fonction $y(x)$ qu'il sait linéaire. Les résultats sont donnés sous la forme de couples (x_i, y_i). Il souhaite trouver la droite qui "correspond le mieux" à ces données, en ce sens que la distance verticale entre un point (x_i, y_i) et la droite soit la plus petite possible.

Modéliser ce problème comme un programme linéaire. Écrire le programme linéaire dual. Pourquoi peut-il être préférable de résoudre le dual ?

Solution : Supposons que les mesures n'ont pas toutes la même valeur x_i, sans quoi la droite d'adéquation est simplement la verticale $x = x_i$. Dans le cas contraire, on introduit trois variables a, b et m, m étant le maximum des distances entre les points et la droite $y = ax + b$.

Cette hypothèse sur m se traduit par les constraintes $-ax_i - b + m \geq -y_i$ et $ax_i + b + m \geq y_i$ (on veut $|x| \leq N$ ce qui est équivalent à $x \leq N$ et $-x \leq N$). La fonction linéaire à minimiser est tout simplement m. La seule contrainte de signe est $m \geq 0$. On notera C_i la i-ème contrainte du premier type et C_{i+n} la i-ème contrainte du second type.

Notons que ce problème linéaire a peu de variables mais énormément de contraintes. Le dual s'écrit :

$$\left\{\begin{array}{c} \max \sum_i y_i(z_{i+n} - z_i) \\ \sum_i x_i(z_{i+n} - z_i) = 0, \\ \sum_i z_{i+n} - z_i = 0, \\ \sum_i z_{i+n} - z_i) \leq 1. \end{array}\right.$$

Il est possible, quand le nombre de points est grand que sa géométrie soit plus simple, et qu'il soit par conséquent plus facile à résoudre par la méthode du simplexe.

Plus court chemin

Exercice 3. Étant donné un graphe orienté G avec arêtes valuées par une fonction positive c, et deux sommets s et t de G, on souhaite trouver un plus court chemin de s à t dans G.

En associant à chaque sommet v une variable d_v, écrire un programme linéaire dont la solution est une borne inférieure à la longueur du plus court chemin de s à t dans G. Montrer que la solution optimale du programme linéaire est en fait exactement égale à la longueur du plus court chemin de s à t. Écrire le dual du programme linéaire, et l'interpréter.

Solution : Soit $G = (V, E)$. On associe à chaque sommet v une variable $d(v)$. On considère le programme linéaire suivant :

$$\max(d(s) - d(t))$$

sous les contraintes : $\forall (u, v) \in E$, $d(v) - d(u) \leq c(u, v)$.

Considérons un chemin $\gamma = (x_i)_{1 \leq i \leq n}$ avec $x_1 = s$, $x_n = t$, et $(x_i, x_{i+1}) \in E$ pour tout $i < n$. Si $d_0(v)$ est une solution aux contraintes du programme linéaire ci-dessus, alors on a :

$$d(t) - d(s) = \sum_{i=1}^{n-1} d(x_{i+1}) - d(x_i) \leq \sum_{i=1}^{n-1} c(x_i - c_{i+1}),$$

qui est exactement la longueur du chemin γ. Il s'ensuit que $d(t) - d(s)$ minore en particulier la longueur du plus court chemin de s à t dans G.

Inversement, si l'on prend comme valeur des valeurs $d(v)$ les longueurs $\delta(v)$ des plus courts chemins de s à v, on a manifestement $d(v) - d(u) \leq c(u, v)$, sans quoi on obtient un chemin plus court pour aller en v en suivant un plus court chemin vers u puis en prenant l'arête (u, v) ; les valeurs $d(v) = \delta(v)$ vérifient donc le systèmes de contraintes. Par conséquent, $\max(d(t) - d(s)) \geq \delta(t) - \delta(s) = \delta(t)$. Mais on a vu que $\max(d(t) - d(s)) \leq \delta(t)$ dans la première partie, d'où l'inégalité (dans ces deux dernières inégalités, le maximum est bien entendu pris sur l'ensemble des solutions au système de contraintes).

Le dual du programme linéaire s'écrit quant à lui, en associant à chaque arête une variable $f(u, v)$,

$$\min \sum_{(u,v) \in E} c(u, v) f(u, v)$$

sous les contraintes :

$$\begin{aligned}
\sum_{(v,u) \in E} f(u, v) - \sum_{(u,v) \in E} f(u, v) &= 0, \forall u \in V \setminus \{s, t\}, \\
\sum_{(v,s) \in E} f(v, s) - \sum_{(s,v) \in E} f(s, v) &= -1, \\
\sum_{(v,t) \in E} f(v, t) - \sum_{(t,v) \in E} f(t, v) &= 1, \\
f(u, v) &\geq 0.
\end{aligned}$$

Les contraintes sont des contraintes de type flot, hormis le fait qu'il peut exister un flot sortant de t ou entrant en s, et que toutes les arêtes sont de capacité infinie.

La première contrainte traduit la conservation du flot, les deux suivantes indiquent que le flot total sortant de s vaut 1 et que le flot total entrant en t vaut 1. Il ne s'agit donc pas d'un problème de flot maximal (cf. exercice suivant), mais d'un problème où la quantité de flot est fixée (ici unitaire), et où le flot est pondéré par des coefficients $c(u, v)$, et on veut que son coût soit minimal ce type de problème est appelé *min-cost flow*. L'exercice montre que ce problème se réduit, par dualité, à un problème de plus court chemin. Noter le parallèle avec Ford-Fulkerson, où l'augmentation du flot se réduisait à un problème d'atteignabilité.

Flot maximum

Exercice 4. Étant donné un graphe orienté G avec capacités sur les arcs données par une fonction positive c, un sommet source s et un sommet destination t, on souhaite trouver un flot de s à t de valeur maximale.

Pour simplifier la présentation, on ajoute un arc de t à s de capacité infinie.

Écrire un programme linéaire pour ce problème. Écrire le dual du programme linéaire et l'interpréter.

Solution : Soit $G = (V, E)$. Pour $(u, v) \in E$, notons $f(u, v)$ le flot traversant l'arête (u, v) et $c(u, v)$ la capacité de l'arête (u, v). Le problème du flot maximal s'écrit alors

$$\max \sum_{(s,u)\in E} f(s, u)$$

sous les contraintes

$$\begin{array}{rcl}
\displaystyle\sum_{(v,u)\in E} f(v, u) - \sum_{(u,v)\in E} f(u, v) & = & 0 \ \forall u \in E \setminus \{s, t\}, \\
\displaystyle\sum_{(s,u)\in E} f(s, u) - \sum_{(u,t)\in E} f(u, t) & = & 0, \\
f(u, v) & \leq & c(u, v), \ \forall (u, v) \in E, \\
f(u, v) & \geq & 0.
\end{array}$$

Les deux premières contraintes sont des contraintes de conservation ; la dernière est une contrainte de capacité. On peut rendre le système plus symétrique en ajoutant une arête (t, s) de capacité suffisante pour absorber tout le flot (par exemple $> \sum_{(u,v)\in E} c(u, v)$) ; la seconde contrainte devient alors un cas particulier de la première, et il suffit de maximiser $f(t, s)$:

$$\begin{array}{rcl}
 & \max f(t, s) & \\
\displaystyle\sum_{(v,u)\in E} f(v, u) - \sum_{(u,v)\in E} f(u, v) & = & 0 \\
f(u, v) & \leq & c(u, v), \ \forall (u, v) \in E, \\
f(u, v) & \geq & 0.
\end{array}$$

Le dual de ce programme linéaire s'écrit alors, en associant à chaque sommet du graphe une variable $e(u)$ (correspondant à la contrainte de conservation associée au sommet u et à chaque arête une variable $x(u, v)$ (correspondant à la contrainte de capacité associée à l'arête (u, v) :

$$\begin{array}{rcl}
\min \sum c(u, v)x(u, v) & & \\
e(u) - e(v) + x(u, v) & \geq & 0, \ (u, v) \neq (t, s) \\
e(t) - e(s) + x(t, s) & \geq & 1, \\
x(u, v) & \geq & 0.
\end{array}$$

À toute coupe de G sous la forme $S \cup T$, on peut associer une solution à ce programme linéaire, en posant $e(u) = 0$ si $u \in S$, $e(v) = 1$ si $v \in T$, $x(u, v) = 1$ ssi $(u, v) \in S \times T$. La valeur de la fonction objectif ainsi obtenue vaut alors exactement la capacité de la coupe. Il s'ensuit que la solution optimale vaut au plus la valeur de la coupe minimale.

Par ailleurs, toute solution minimale doit satisfaire, pour $(u,v) \neq (t,s)$, l'équation $x(u,v) = e(v) - e(u)$ si $e(v) \geq e(u)$ et $= 0$ sinon. Choisissons maintenant un réel $r \in [0,1]$ uniformément au hasard, et définissons $e'(u) = 0$ si $e(u) \leq r$ et $e'(u) = 1$ sinon. Ceci induit une coupe aléatoire, dont la valeur moyenne M est :

$$\begin{aligned} M &= \sum_{u,v} c(u,v) \Pr[e'(u) = 0 \text{ et } e'(v) = 1] \\ &= \sum_{u,v: e(u) \leq e(v)} c(u,v) \Pr[r \in [e(u), e(v)]] \\ &= \sum_{u,v: e(u) \leq e(v)} c(u,v)(e(v) - e(u)) \\ &= \sum_{u,v} c(u,v) x(u,v). \end{aligned}$$

La valeur de la solution optimale est donc une moyenne de valeur de diverses coupes, donc vaut au moins la valeur de la coupe minimale.

Par conséquent, la solution optimale est exactement égale à la valeur de la coupe minimale, et le théorème de dualité de la programmation linéaire nous donne le théorème max-flow min-cut.

Sac-à-dos

Exercice 5. On considère le problème du sac-à-dos : les objets à emporter, numérotés de 1 à n, ont pour poids $p_1, p_2, \ldots, p_n$ et pour cœfficients d'intérêt $a_1, a_2, \ldots, a_n$. La capacité du sac (ou plutôt celle du porteur) est p.

1. Exprimer le problème d'optimisation du remplissage du sac sous forme d'un problème linéaire à $n+1$ contraintes faisant intervenir des variables à valeurs entières.
2. Exécuter la méthode du simplexe dans le cas suivant avec $p = 18$:

i	1	2	3	4	5
p_i	8	6	6	4	3
a_i	7	5	4	3	2

3. Un ensemble d'objets est dit *bloquant* si la somme des poids des objets qui le composent est supérieure à p, et *caractéristique* s'il est bloquant minimal. Montrer que l'on peut associer à chaque ensemble caractéristique une contrainte où les cœfficients des variables sont 0 ou 1. Montrer que l'ajout de ces contraintes permet de supprimer la contrainte du premier item qui contenait des cœfficients supérieurs à 1, et que les deux formulations sont équivalentes (c-à-d donnent les mêmes solutions optimales).
4. Peut-on être assuré que l'optimum trouvé a toutes ses coordonnées égales à 0 ou 1 ?

Solution :

(a) Introduisons une variable ϵ_i pour chaque objet à emporter, qui indiquera la présence de l'objet correspondant dans le sac : si $\epsilon_i = 1$ l'objet est dans le sac, sinon $\epsilon_i = 0$.

La fonction que l'on va chercher à maximiser est l'intérêt total, à savoir la somme des intérêts des objets emportés, ce qui s'écrit en termes des ϵ_i comme $\sum_i a_i \epsilon_i$.

Les contraintes sur la composition du sac sont de deux types : d'une part la contrainte originale disant que le poids total des objets emportés ne doit pas excéder p, qui s'écrit alors $\sum_{1 \leq i \leq n} p_i \epsilon_i \leq p$; d'autre part une contrainte par variable ϵ_i introduite, pour traduire le fait que cette variable ne peut pas prendre de valeur supérieure à 1. Le fait que ces variables soit positives est automatique dans le formalisme du simplexe. On obtient bien de cette façon $n+1$ contraintes.

En résumé, le programmme linéaire s'écrit :

$$\max \sum_i a_i \epsilon_i$$

$$\begin{array}{rcl}
\sum_{1\leq i\leq n} p_i\epsilon_i & \leq & p, \\
\epsilon_i & \leq & 1, 1 \leq i \leq n, \\
\epsilon_i & \geq & 0, 1 \leq i \leq n, \\
\epsilon_i \in \mathbb{Z} & &
\end{array}$$

(b) La méthode du simplexe travaille sur des réels et non sur des entiers. On résout donc par l'algorithme du simplexe, un problème différent, où il est possible de prendre des fractions des objets.

Voici l'exécution du simplexe sur cet exemple.

Initialisation.

Solution initiale : $x_1 = \ldots = x_5 = 0$.

$$\max 7x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5$$

$$\left\{\begin{array}{rcl}
t & = & 18 - 8x_1 - 6x_2 - 6x_3 - 4x_4 - 3x_5 \\
u_i & = & 1 - x_i \quad (1 \leq i \leq 5) \\
x_i, u_i, t & \geq & 0
\end{array}\right.$$

Première étape. On augmente x_1 autant que possible jusqu'à annuler u_1 lorsque $x_1 = 1$.

$$\max 7 - 7u_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5$$

$$\left\{\begin{array}{rcl}
t & = & 18 - 8x_1 - 6x_2 - 6x_3 - 4x_4 - 3x_5 \\
x_1 & = & 1 - u_1 \\
u_i & = & 1 - x_i \quad (2 \leq i \leq 5) \\
x_i, u_i, t & \geq & 0
\end{array}\right.$$

Deuxième étape. On augmente x_2 jusqu'à annuler u_2 lorsque $x_2 = 1$.

$$\max 12 - 7u_1 - 5u_2 + 4x_3 + 3x_4 + 2x_5$$

$$\left\{\begin{array}{rcl}
t & = & 18 - 8x_1 - 6x_2 - 6x_3 - 4x_4 - 3x_5 \\
x_1 & = & 1 - u_1 \\
x_2 & = & 1 - u_2 \\
u_i & = & 1 - x_i \quad (3 \leq i \leq 5) \\
x_i, u_i, t & \geq & 0
\end{array}\right.$$

Troisième étape. On augmente x_3 autant que possible jusqu'à annuler t quand $x_3 = 2/3$.

$$\max 12 - 7u_1 - 5u_2 + 4(\frac{4 + 8u_1 + 6u_2 - 4x_4 - 3x_5 - t}{6}) + 3x_4 + 2x_5$$

$$\begin{cases} x_1 &= 1 - u_1 \\ x_2 &= 1 - u_2 \\ x_3 &= (4 + 8u_1 + 6u_2 - 4x_4 - 3x_5 - t)/6 \\ u_3 &= (2 - 8u_1 - 6u_2 + 4x_4 + 3x_5 + t)/6 \\ u_4 &= 1 - x_4 \\ u_5 &= 1 - x_5 \\ x_i, u_i, t &\geq 0 \end{cases}$$

Quatrième étape. On augmente x_4 autant que possible : jusqu'à annuler u_4 lorsque $x_4 = 1$. Dans la fonction objectif, tous les coefficients des variables sont négatifs, donc on est arrivé à l'optimum qui vaut 15. Cet optimum est obtenu en annulant les variables qui interviennent dans la fonction objectif avec un coefficient strictement négatif, soit u_1, u_2, u_4, t ce qui fait $x_1 = x_2 = x_4 = 1$, et $6x_3 + 3x_5 = 0$, d'où $x_3 = x_5 = 0$.

(c) Dans cette question, on cherche à remplacer la contrainte d'origine sur le poids total des objets emportés par des contraintes dont les coefficients sont 0 ou 1. Soit E l'ensemble des indices des objets appartenant à un ensemble caractéristique C. Par définition, toute composition valide du sac ne saurait contenir tous les éléments de E. Comme le nombre d'éléments de C figurant dans le sac est simplement $\sum_{i \in E} \epsilon_i$, la contrainte issue de E s'écrit $\sum_{i \in E} \epsilon_i \leq \#E - 1$, où $\#E$ désigne le cardinal de E. Notons que cette contrainte est effectivement à coefficients 0 ou 1.

Montrons alors que le fait de remplacer la contrainte $\sum_{1 \leq i \leq n} p_i \epsilon_i \leq p$ transforme le système en un système équivalent.

Supposons dans un premier temps qu'une contrainte correspondant à un ensemble caractéristique dont l'ensemble d'indices est E n'est pas vérifiée ; on a alors la chaîne d'inégalités :

$$\sum_{1 \leq i \leq n} p_i \epsilon_i \geq \sum_{i \in E} p_i \epsilon_i \geq \sum_{i \in E} p_i > p,$$

où la deuxième inégalité vient du fait que $\sum_{i \in E} \epsilon_i = \#E$ et la troisième traduit le fait qu'un ensemble caractéristique est en particulier bloquant. Par la suite, la contrainte $\sum_{1 \leq i \leq n} p_i \epsilon_i \leq p$ est violée.

Réciproquement, supposons que $\sum_{1 \leq i \leq n} p_i \epsilon_i > p$. On construit alors une suite d'ensemble d'indices de la façon suivante :

$$E_0 = \{i / \epsilon_i = 1\}, \ E_{k+1} = E_k \setminus \{\min\{i / p_i = \min_{\ell \in E_k} p_\ell\}\}.$$

En d'autres termes, on enlève à chaque ensemble l'indice d'un des éléments de plus petit poids (en l'occurrence celui d'indice le plus petit) pour obtenir le suivant. Clairement, E_n est vide, donc $j = \max_k \{k / \sum_{i \in E_k} p_i > p\}$ existe bien. Montrons que l'ensemble d'objets associé à E_j est caractéristique : il est bloquant par définition, et si l'on enlève un de ses éléments de plus petit poids, on obtient un ensemble non bloquant (a fortiori si l'on enlève un élément qui n'est pas de poids minimal). Comme $E_j \subset E_0$, $\epsilon_i = 1$ pour tout $i \in E_j$ et on a donc $\sum_{i \in E_j} \epsilon_i > \#E - 1$. On a donc bien construit une contrainte d'ensemble caractéristique violée par la composition invalide du sac.

(d) Il y a plusieurs bonnes raisons de penser que non : par exemple si le maximum est atteint en un unique sommet du simplexe, la formulation donnée fournira la même solution que la formulation originale, à laquelle elle est équivalente.