Module D108 : Programmation mathématique et optimisation
Programmation Linéaire

Auteur(s) :
Nasser Saheb (IUP Miage, Bordeaux)
Mike Robson (IUP Miage, Bordeaux)

Contact :
saheb@labri.fr
mike.robson@labri.fr

Date de dernière modification : 30 Janvier 2006

Programmation linéaire

La programmation linéaire est un cas simple mais important du problème général d'optimisation mathématique de trouver la valeur optimale (maximum ou minimum selon le cas) d'une fonction mathématique étant donné un nombre de restrictions (*contraintes*) sur les valeurs des paramètres. En général, les problèmes de ce type peuvent être très compliqués à résoudre mais dans le cas de la programmation linéaire des algorithmes efficaces sont connus mais un nombre énorme de problèmes réels et importants peuvent être exprimés par des programmes linéaires.

Définition: un programme linéaire sur n variables (par convention nous écrivons les variables $x_1, \ldots x_n$) est d'optimiser (minimiser ou maximiser) une fonction linéaire dans les variables étant donné des contraintes linéaires dans les variables.

[Une fonction linéaire est une fonction qui peut être exprimée comme $\sum_{i=1}^{n} c_i x_i + c_0$; on ignore le c_0 de la fonction à optimiser (*la fonction objective ou économique*) parce qu'elle ne change en rien la question. Une contrainte linéaire est qu'une fonction linéaire doit être supérieure ou égale à une autre; par une simple manipulation algébrique on peut toujours supposer que la première a le terme constante *0* et la deuxième est une fonction triviale constante On obtient donc une condition de la forme de: $\sum_{i=1} a_i x_i \leq b$.]

Exemple

Une usine de verre peut produire de la verre de qualité basse, moyenne ou supérieure; une tonne de la basse qualité nécessite *3* tonnes de sables, *500*gms d'additifs et 1 heure du travail d'un ouvrier; pour les deux autres qualités, les chiffres correspondants sont (moyenne) *3.2*

tonnes, *1000*gms et *1,5* heures; (supérieure) *3.5* tonnes, *2500* gms et *4* heures. Les bénéfices de la vente d'une tonne sont de *100*, *200* et *300* Euros. Chaque semaine *200* tonnes de sable et *120000* gms d'additif sont disponibles et il y a *6* ouvriers chacun travaillant au maximum *35* heures/semaine. Quelles quantités doit-on produire chaque semaine pour maximiser les bénéfices?

On écrit $x_1,\ x_2,\ x_3$ pour les quantités produites de chaque qualité.
On veut maximiser $100x_1 + 200x_2 + 300x_3$ avec les contraintes:

$$\begin{cases} 3x_1 + 3.2x_2 + 3.5x_3 \leq 200 \ (sable) \\ 500x_1 + 1000x_2 + 2500x_3 \leq 120000 \ (additifs) \\ x_1 + 1.5x_2 + 4x_3 \leq 210 \ (main\ d'oeuvre). \end{cases}$$

Forme canonique

Un programme linéaire en forme canonique est à maximiser une fonction $\sum_{i=1}^{n} c_i x_i$ avec m contraintes dont la i-ème est: $\sum_{j=1} a_{i,j} x_j \leq b_i$ et aussi les contraintes implicites que chaque $x_i \geq 0$.

Il est claire que tout programme linéaire peut être transformé en un programme équivalent en forme canonique; un tel programme est résumé par la matrice A des $a_{i,j}$ et les deux vecteurs b et c.

On appelle *solution* du programme n'importe quel vecteur de valeurs x_i; les solutions qui respectent toutes les contraintes sont les *solutions réalisables*; un programme qui admet au moins une solution réalisable est un *programme faisable*. Pour un programme faisable, on appelle *solution optimale* une solution réalisable où la fonction objective prend sa valeur maximale parmi toutes les solutions réalisables. (On dit aussi de façon inexacte que cette valeur maximale est la ``solution optimale".)

Cas Spéciaux

Il y a des cas où la recherche de ``la solution optimale" ne peut pas réussir; il est important de savoir quels sont ces cas parce qu'un programme doit pouvoir les traiter:

- le programme n'est pas faisable; impossible de trouver la meilleure solution réalisable s'il n'y en a pas;
- il existe plusieurs solutions optimales; laquelle sera considérée ``la solution optimale"?
- solution infinie: si la fonction économique peut prendre des valeurs indéfiniment grandes parmi les solutions réalisables, il n'est pas question de trouver une solution optimale parce qu'il n'y pas de valeur maximale de la fonction.

Exemples

- le programme n'est pas faisable: $x_1 \leq 10,\ x_2 \leq 5,\ -x1 - x2 \leq -20$
- il existe plusieurs solutions optimales: $x_1 + x_2 \leq 20$, maximiser $2x_1 + 2x_2$
- solution infinie: $x_1 - 2x_2 \leq 10,\ -2x_1 + x_2 \leq 20$, maximiser $x_1 + x_2$
- (pour tout nombre positif x, $(x/2, x/2)$ est une solution réalisable avec valeur x de la fonction économique.)
- Les problèmes suivants, sont-ils des programmes linéaires? (dans tous les exercices sauf si explicité, les contraintes implicites que toutes les variables sont positives ou nulles sont valables.)
  - Minimiser $100x_1 - 200x_2 + 300x_3$ avec les contraintes :
    $$\begin{cases} -3x_1 + 3.2x_2 - 3.5x_3 \leq -200 \\ 500x_1 + 1000x_2 + 2500x_3 \leq 120000 \\ x_1 + 1.5x_2 + 4x_3 \leq 210 \end{cases}$$
  - Maximiser $100x_1 + 200x_2 + 300x_3$ avec les contraintes :
    $$\begin{cases} 3x_1 + 3.2x_2 + 3.5x_3 \leq 200 \\ 500x_1 + 1000x_2 + 2500x_3 \leq 120000 + 50x_1 \\ x_1 + 1.5x_2 + 4x_3 + 2x_3 \leq 210 \end{cases}$$
  - Minimiser $100x_1 + 200x_2 + 300x_3$ avec les contraintes :
    $$\begin{cases} 3x_1 + 3.2x_2 - 3.5x_3 < 200 \\ 500x_1 + 1000x_2 + 2500x_3 < 120000 \\ x_1 + 1.5x_2 + 4x_3 < 210 \end{cases}$$
  - Maximiser $100x_1 - 200x_2 + 300x_3$ avec les contraintes :
    $$\begin{cases} x_1 + 3.2x_2 + 3.5x_3 \leq 200 \\ 500x_1 + 1000x_2 + 2500x_3 \leq 120000 \\ x_1 * x_2 + 4x_3 \leq 210 \end{cases}$$
  - Maximiser $100x_1 - 200x_2 + 300x_3$ avec les contraintes :
    $$\begin{cases} x_1 + 3.2x_2 + 3.5x_3 \leq 200 \\ 500x_1 + 1000x_2 + 2500x_3 \leq 120000 \ ou \\ x_1 + 1.5x_2 + 4x_3 + 2x_3 \leq 210 \end{cases}$$
- Lesquelles des solutions suivantes du programme de l'usine de verre sont réalisables?
  - $x_1 = 40,\ x_2 = 10, x_3 = 10$
  - $x_1 = 50,\ x_2 = -20, x_3 = 20$
  - $x_1 = 40,\ x_2 = 20, x_3 = 10$
  -
- La solution $(x_1 = 20,\ x_2 = 20,\ x_3 = 20)$ est-elle optimale?

Exercice 1 :

Exprimer le problème suivant en forme de programme linéaire
On veut acheter des aliments au plus bas prix possible qui fourniront les besoins journaliers de calories (au moins *2000*), protéines (*30* unités) et fibre (*20* unités); les aliments disponibles sont de la viande (un kilo à un prix de *5* fournit *800* calories, *10* unités de protéine et *2* de fibre), des fruits (*200* cal, *6* et *4* unités à un prix de *2*) et du riz (*100* cal, *5* et *4* unités à un prix de *1,5*).

Consulter la solution...

Solution de l'exercice 1

Solution

Soit x_1, x_2 et x_3 les quantités de viande, fruits et riz:
Minimiser $5x_1 + 2x_2 + 1.5x_3$ avec les contraintes
$800x_1 + 200x_2 + 100x_3 \geq 2000$
$10x_1 + 6x_2 + 5x_3 \geq 30$
$2x_1 + 4x_2 + 4x_3 \geq 20$
$x_1, x_2, x_3 \geq 0$

Exercice 2 :

Donner un programme linéaire en forme canonique équivalent à:

Minimiser $2x_1 + 3x_2 - x_3$ avec les contraintes :

$$\begin{cases} 3.2x_2 + 3.5x_3 \leq 200 + 5x_1 \\ 500x_1 + 1000x_2 + 2500x_3 \geq 120000 \\ x_1 + 1.5x_2 + 4x_3 \geq x_3 + 210 \end{cases}$$

Consulter la solution...

Solution de l'exercice 2

Solution

Maximiser $-2x_1 - 3x_2 + 3x_3$ avec les contraintes

$$5x_1 + 3.2x_2 + 3.5x_3 \leq 200$$
$$-500x_1 - 1000x_2 - 2500x_3 \leq -120000$$
$$-x_1 - 1.5x_2 - 3x_3 \leq 210$$

Exercice 3 :

Donner un programme linéaire en forme canonique équivalent à:

Maximiser $100x_1 - 200x_2 + 300x_3$ avec les contraintes :

$$1x_1 + 3.2x_2 + 3.5x_3 \leq 200$$
$$500x_1 + 1000x_2 + 2500x_3 \leq 120000$$
$$x_1 + x_2 + 4x_3 \leq 210$$

sans les contraintes implicites $x_1 \geq 0$ etc.

Solution de l'exercice 3

Solution

Maximiser $100(x_1 - y_1) - 200(x_2 - y_2) + 300(x_3 - y_3)$ avec les contraintes:

$$1(x_1 - y_1) + 3.2(x_2 - y_2) + 3.5(x_3 - y_3) \leq 200$$
$$500(x_1 - y_1) + 1000(x_2 - y_2) + 2500(x_3 - y_3) \leq 120000$$
$$(x_1 - y_1) + (x_2 - y_2) + 4(x_3 - y_3) \leq 210$$

Module D108 : Programmation mathématique et optimisation
Programmation linéaire
Programmation linéaire en nombres entiers

Auteur(s) : Nasser SAHEB;- IUP Miage Bordeaux
Date de dernière modification : 30 Janvier 2006

Pourquoi les contraintes d'intégrité ?

La méthode du simple permet de trouver la solution optimale d'une fonction linéaire économique face à des contraintes du type linéaire. Elle autorise des variables de signes quelconques et peut aussi les astreindre à être positives (ou négatives). Dans beaucoup de circonstances réelles, les variables ne peuvent être que de type entiers. Il arrive en effet fréquemment que le choix imposé doit respecter la règle d'intégrité. Malheureusement ce type de contraintes ne peut pas être traité directement par l'algorithme de simplexe.

Exemple illustratif. : Un produit es disponible dans des paquets de *20kg* de prix de *50€* ou des paquets de *40kg* de prix de *80€*. Un client, qui a d'au moins de *100kg* du produit cherche à en acheter au moindre coût possible. La contrainte d'intégrité qui s'oppose tout naturellement ici, implique qu'il ne peut pas en acheter la moitié ou deux tiers de paquet. Soient x_1 et x_2 les nombres respectifsde paquets que le client va acheter. Le programme linéaire suivant formalise le problème :

$$\text{minimiser } 50x_1 + 80x_2$$

$$\text{sous : } \begin{cases} 20x_1 + 40x_2 \geq 100 \\ x_1, x_2 \geq 0 \text{ et } x_1, x_2 \in IN \end{cases}$$

Effet des contraintes d'intégrité

Dans le problème illustratif, si l'on enlève la contrainte d'intégrité de $x_1, x_2 \in IN$, le problème devient très simple. En effet puisque le prix du produit dans le paquet de *20kg* est de *2,5 €/kg* et dans le paquet de 50kg est de *2€/kg*, le client pourrait acheter 2,5 paquets de contenance de 80kg et cela lui coûterait $2{,}5 \times 80 = 200€$. Malheureusement, la contrainte d'intégrité lui impose de remplacer le nombre *2,5* par *3*, et cela lui coûterait $3 \times 80 = 240€$. Or, il existe une solution moins coûteuse : 2 paquets de *40kg* et un paquet de *20kg*, coûtant au total de *130€*.

Cet exemple simple montre que l'arrondissement de la solution optimale d'un programme linéaire et nombres réels n'aboutit pas nécessairement à une solution optimale pour le même problème avec la contrainte d'intégrité.

On peut être tenté de confirmer maintenant que du moment où la recherche d'un solutions optimale pour un programme linéaire en nombre d'entrées se fait dans un ensemble plus restreint, la méthode de résolution doit être plus simple que ce même problème sans la contrainte d'intégrité. C'est une confirmation tout à fait erronée. En effet, alors que l'algorithme du simplexe fournit une solution satisfaisante pour les programmes linéaire en nombres réels, on démontre que le problème de programmation linéaire en nombres entiers s'apparente à une classe de problème, appelés problèmes NP-complets, qui sont considérés comme des problèmes algorithmiquement difficiles. Il s'agit d'un ensemble de problèmes de décision qui ont été démontrés être polynomialement équivalents. On ne connaît pas d'algorithmes de décision en temps polynomial pour ces problèmes, sans avoir prouvé que la difficulté est de nature intrinsèque. Quoiqu'il en soit on admet en général que ces problèmes et, donc, celui de programmation linéaire en nombres entiers est de nature difficile.

Forme standard d'un programme linéaire en nombres entiers.

Tout programme linéaire en nombres entiers s'écrit sous la forme :

$$\text{maximiser } c, x$$

$$\text{sous : } \begin{cases} Ax \leq b \\ x \geq 0 \\ x \in IN^n \end{cases}$$

Où c est un vecteur de IR^n, A une matrice à m lignes et n colonnes et $b \in IR^n$. On peut transformer un problème de maximisation ou un problème avec contraintes de type $=$ (au lieu de $\leq$) en un problème sous cette forme, en appliquant les méthodes utilisées en programmation linéaire.

Applications

Les problèmes d'optimisation, où des contraintes d'intégrité interviennent, ont un champ d'application très vaste. On peut en énumérer, à titre d'exemple, production en séries, allocations de ressources, localisation des sites, exploitations des mimerais , investissement, et programme des vols d'avions. D'autres applications de programmes linéaires se trouvent en calculs scientifiques tels que les problèmes d'optimisation combinatoires dans un contexte d'ingénierie ou mathématico-physique.

Problème du sac-à-dos

Une instance très particulière des programmes linéaire qui, malgré sa simplicité apparente, se classifie dans les problèmes difficiles est celui du sac-à-dos (*knapsack en anglais*).

Ce problème s'énonce ainsi :

Un randonneur peut faire un certain nombre de choix parmi les *n* nourritures disponibles numérotées $1, \cdots, n.$ Chaque nourriture *i* possède un volume v_i et sa valeur nutritive u_i. Le sac à dos du randonneur a une capacité V à ne pas dépasser. Le randonneur souhaite alors à faire un menu qui maximise la valeur nutritive totale. Désignons par $x_i = 0 \; ou \; 1,$ la quantité de la nourriture *i* que le randonneur va prendre (si $x_i = 0,$ la nourriture *i* n'est pas choisie, si $x_i = 1,$ elle est choisie). Le problème du sac-à-dos s'écrit alors :

$$\text{maximiser} \sum_{j=1} u_j x_j$$

$$\text{sous :} \begin{cases} \sum_{j=1}^{n} v_j x_j \le V \\ x_j \in \{0,1\}, \; pour \; 1 \le j \le n. \end{cases}$$

Il existe une variante de ce problème légèrement plus générale : on suppose que toute nourriture j est disponible en quantité b_j (b_j entier). Les contraintes $x_j \in \{0,1\}$ dans cette invariante deviennent alors : $x_i \in IN \; et \;\; x_i \le b_j \;\; pour \;\; 1 \le j \le n.$

La version initiale est appelée par certains auteurs ‘‘programme linéaire en variables bivalentes’’. Ces deux variantes de problème du sac-à-dos et d'autres variables sont liées à un bon nombre de problèmes d'optimisation combinatoire. Etant tous liés au problèmes difficiles. A bondonnant une recherche entièrement algébrique, les informaticiens se limitent à des méthodes de résolutions ‘‘approximatives’’, appelées heuristiques pour aborder ces problèmes dans toute leur généralité. Ces recours ne sont pas tout à fait satisfaisants, dans la mesure où l'heuristique ne fournit pas toujours la solution exacte et elle n'est pas non plus à l'abri d'une complexité exponentielle.

Il existe de nombreuses méthodes, qui avec de divers degré de succès, s'appliquent aux programmes linéaires en variables aléatoires. Les techniques les plus populaires sont fondées sur séparation et évaluation (*branch and bound en anglais*).

Méthode par séparation et évaluation

Soit à résoudre le programme :

$$\text{maximiser} \; z = cx$$

$$\text{sous :} \begin{cases} Ax \le b \\ x \in IN^n \end{cases}$$

Soit z la valeur maximum de la fonction objective cx .

<u>Etape 1.</u> On considère le programme linéaire relaxé (i.e. sans contrainte d'intégrité).

La méthode simplexe s'applique et l'on peut trouver une solution optimale ($x_1', \cdots, x_2', z'$).

z' est évidemment un majorant de $z = cx$ dans une initial. Il n'est pas nécessairement réalisable, puisque la solution trouvée peut ne pas respecter les contraintes d'intégrité.

Etape 2. Si $x_1' = n + \varepsilon, \; avec \; 0 \leq \varepsilon \leq 1,$ on considère deux programmes linéaires relaxés, l'un avec la contrainte supplémentaire $x_1 \leq n$ et l'autre avec $x_1 \geq n+1$ (appelée branche gauche et branche droite respectivement). La résolution éventuelle de chacune des branches (s'elle admet une solution) permet de trouver deux solutions plus restrictives (mais pas nécessairement réalisables).

Etape 3. On continue l'étape 2 pour une des variables qui n'a pas reçu une valeur entière et cela pour chacune des branches.

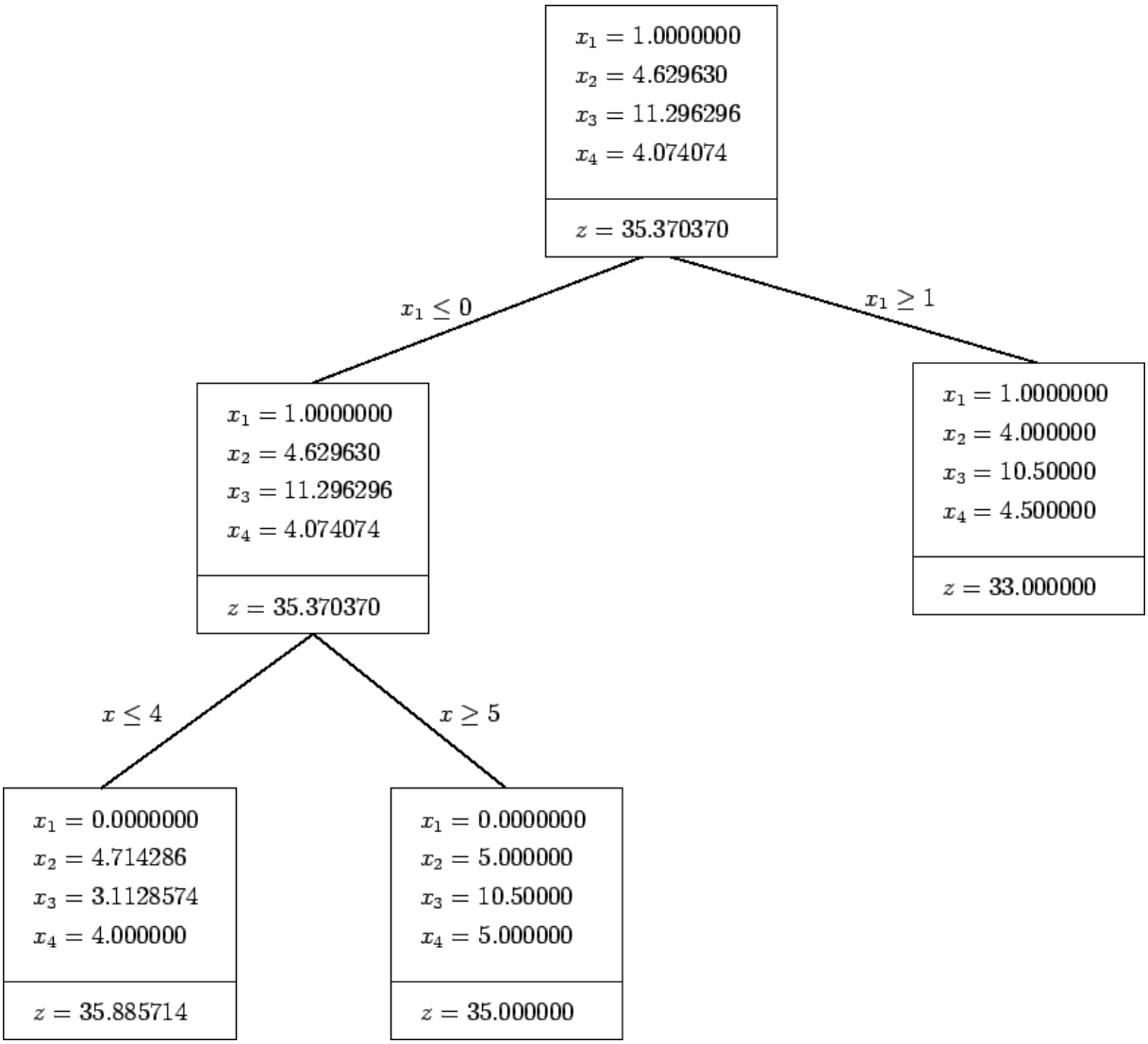
Il est évident qu'à chaque branchement les valeurs trouvées pour *z* seront plus petites et donc, en descendant une branche de l'arborescence, on finira soit par l'irréalisablité du programme linéaire relaxé avec des contraintes associées au branchement soit par une solution entière optimale.

Une comparaison finale des branches abouties permet de trouver la solution optimale entière recherchée.

Exemple 1. Soit :

$$\text{maximiser } -x_1 + x_2 + 2x_3 + 2x_4$$

$$\text{sous : } \begin{cases} 3x_1 - 3x_2 + 4x_3 - 4x_4 \leq 15 \\ 4x_1 - 5x_2 - 2x_3 - 2x_4 \leq 16 \\ -2x_1 - 3x_2 + 2x_3 + 4x_4 \leq 25 \\ x_1, x_2, x_3, x_4 \in IN \end{cases}$$



Solution optimale

Exemple 2. (Une application de la programmation dynamique)

L'exemple suivant est emprunté à D. Sills. Soit le problème du sac-à-dos :

$$\text{maximiser } z = \sum_{j=1}^{n} u_j x_j$$

$$\text{sous : } \begin{cases} \sum_{j=1}^{n} a_j x_j \leq b \\ x_j \in IN, pour\ tout 1 \leq j \leq n. \end{cases}$$

On suppose c_i et a_j sont positifs et $b > 0.$ Nous utilisons la méthode de programmation dynamique pour aborder ce problème.

Cherchons dans un premier temps la solution optimale du problème relaxé. Soit $x^* = (x_1^*, \cdots, x_n^*)$ cette solution et soit z^* la valeur de la fonction objective pour cette solution.

Désignons par $\lfloor x^* \rfloor$ le vecteur formé des valeurs entières par défaut de $x_1^*, \cdots, x_n^*$. $\lfloor x^* \rfloor$ est alors une solution réalisable du problème. Donc la valeur optimale recherchée se trouve dans l'intervalle $\left[\sum_{j=1}^{n} c_j \lfloor x_j^* \rfloor, z^*\right]$.

Soit maintenant $F(s) = \max\left\{\sum c_j x_j \middle| \sum a_j x_j \le s, x_j \in IN\right\}$. $F(s)$ est donc la valeur maximum de la fonction objective, lorsqu'on remplace b et s. Calculons $F(s)$ pour $s = 0,1,2,\cdots,b,$ tenant compte de $F(0) = 0$ et $F(s) = max(max_j\{c_j + F(s - a_j)\}, F(s-1))$

On peut poser $F(s) = -\infty \; pour \; s < 0.$ La formule pour $F(s)$ correspond au fait que le problème du sac-à-dos pour un volume s à ne pas dépasser consiste à remplir le sac d'abord d'un volume $s - a_j$ et ajouter ensuite un article de volume a_j ; on choisit donc l'indice j qui maximise $c_j + F(s - a_j)$. La deuxième solution pour $s = b$ est la solution optimale.

Exemple numérique.

Soit

$$\text{maximiser } 11x_1 + 7x_2 + 5x_3 + x_4$$
$$\text{sous : } \begin{cases} 6x_1 + 4x_2 + 3x_3 + x_4 \le 25 \\ x_1, x_2, x_3, x_4 \in IN \end{cases}$$

Solution : Dans le problème relaxé le rapport ‘‘quantité-prix’’, montre que le randonneur a l'intérêt de prendre la 1ère nourriture au maximum et cela fournit la solution $x^* = \left(\frac{25}{6}, 0, 0, 0\right)$, donnant la valeur $11.\frac{25}{6}$ à la fonction objective. La solution arrondie vaut alors $\left(\left\lfloor \frac{25}{6} \right\rfloor, 0, 0, 0\right) = (4,0,0,0)$ donnant la valeur 44 à la fonction objective. Donc la valeur optimale de z doit satisfaire $44 \le z \le 45.$

Appliquons maintenant l'algorithme de programmation dynamique à cette instance du problème du sac-à-dos.

$$
\begin{aligned}
F(0) &= 0 \\
F(1) &= max\{11 + F(1-6),\ 7 + F(1-4),\ 5 + F(1-3),\ 1 + F(1-1),\ F(1-1)\} = 1 \\
F(2) &= max\{11 + F(2-6),\ 7 + F(2-4),\ 5 + F(2-3),\ 1 + F(2-1),\ F(2-1)\} = 2 \\
F(3) &= max\{11 + F(3-6),\ 7 + F(3-4),\ 5 + F(3-3),\ 1 + F(3-1),\ F(3-1)\} = 5 \\
F(4) &= max\{11 + F(4-6),\ 7 + F(4-4),\ 5 + F(4-3),\ 1 + F(4-1),\ F(4-1)\} = 5 \\
&\ldots \\
F(25) &= max\{11 + F(25-6),\ 7 + F(25-4),\ 5 + F(25-3),\ 1 + F(25-1),\ F(25-1)\} = 45
\end{aligned}
$$

On en déduit alors que la valeur nutritive maximale que la randonneur peut placer dans son sac-à-dos est de 45, obtenu en choisissant le menu $(4, 0, 0, 1)$

Module D108 : Programmation mathématique et optimsation
Programmation linéaire
Interprétation Géométrique

Auteur(s) : Mike Robson - IUP Miage Bordeaux

Date de dernière modification : 30 Janvier 2006

Point dans un espace

Il est très utile de considérer les n variables $x_1, \ldots, x_n$ comme les coordonnées d'un point dans un espace de dimension n. Ce point de vue permet souvent de comprendre mieux les problèmes et les algorithmes.

Interprétation de tous les concepts

Une *solution* n'est qu'un point dans l'espace.

Une *contrainte* (comme $x_1 + 2x_2 - x_3 \leq 0$) représente la division de l'espace par un plan (dans l'exemple le plan $x_1 + 2x_2 - x_3 = 0$); les points respectant la contrainte sont tous ceux sur le ``bon'' coté du plan plus ceux situés sur le plan (parce que la contrainte est du type $\leq$).

Dans le cas $n = 2$, une contrainte n'est donnée par un plan mais par une ligne droite; dans les cas $n > 3$ c'est un *hyperplan*.

L'espace de toutes les solutions réalisables est l'intersection de tous les demi-espaces donnés par les contraintes; si cette intersection est vide le programme n'est pas faisable; s'il est fini l'intersection est un polygone (en dimension 2), un polyèdre (en dimension 3) ou un *polytope* (en dimension supérieure à 3); s'il est infini le programme risque de ne pas avoir une solution optimale parce qu'il n'y a pas de borne sur la fonction objective.

Maximiser la fonction objective correspond à aller le plus loin possible dans une *direction préférée* avec déplacements orthogonaux à cette direction permises; pour prendre un exemple simple en deux dimensions, maximiser $x_1 + x_2$ équivaut à se déplacer tant que possible dans le sens nord-est; toutes les directions plus à l'est que nord-ouest ou plus au nord que sud-est sont bonnes parce que leur composant dans le sens nord-est est strictement positif. Maintenant c'est clair que si l'espace de solutions faisables est fini, la fonction économique prend son maximum sur un bord de cet espace; on va voir davantage dans un instant.

Convexité

Quand l'espace de solutions réalisables n'est pas vide, il a une propriété importante, la *convexité*. Tout point situé sur le segment de ligne droite reliant deux solutions réalisables est lui-même aussi une solution réalisable. De la perspective géométrique ce fait est tout à fait évident; algébriquement, si on note qu'un point sur ce segment entre deux solutions $x_1, \ldots, x_n$ et $y_1, \ldots, y_n$ est donné par $c * x_1 + (1 - c) * y_1, \ldots, c * x_n + (1 - c) * y_n$ pour une constante c avec $0 \leq c \leq 1$, on peut démontrer facilement que si une contrainte linéaire est vérifiée à $x_1, \ldots, x_n$ et à $y_1, \ldots, y_n$, elle est vérifiée aussi aux solutions sur la ligne entre les deux.

Solutions de base

n équations linéaires en n variables, en général, déterminent une solution unique. (D'autres cas sont possibles quand les équations ne sont pas linéairement indépendantes; soit aucune solution, soit plusieures.)
Une solution est *de base* si elle est la solution unique déterminée par n des contraintes, en remplaçant le $\leq$ de la contrainte par $=$. Autrement dit une solution de base est l'intersection de n des plans (hyperplans) des contraintes; si les (hyper)plans n'ont pas un seul point comme intersection, ils ne donnent pas de solution de base.

Quand le programme est donné en forme canonique, les équations $x_j = 0$ des contraintes implicites $x_j \geq 0$ peuvent entrer dans les n équations déterminant une solution de base.

Les solutions de base réalisables ne sont que les sommets (angles) du polytope de solutions réalisables.

Une propriété très importante pour la solution de programmes linéaires est que, (si le programme a une solution réalisable optimale), une des solutions réalisables de base est optimale. De la perspective géométrique cette propriété semble évidente.

On considère le programme suivant:

Maximiser $x_1 + x_2 + x_3$ avec les contraintes:

$$\begin{cases} 2x_1 + x_2 - x_3 \leq 18, \\ -4x_1 + x_2 - x_3 \leq -12, \\ -8x_1 - x_2 + 5x_3 \leq 0, \\ -2x_1 + 2x_2 - 2x_3 \leq 6, \\ -x_2 \leq -5 \\ x_1,\ x_2,\ x_3 \geq 0. \end{cases}$$

Pour chaque solution donnée, est-elle de base, est-elle réalisable?

- $(20, 5, 27)$

  Consulter la solution...

- $(5, 20, 12)$

  Consulter la solution...

- $(5, 5, -3)$

  Consulter la solution...

- $(35, 5, 57)$

  Consulter la solution...

- $(0, 22.5, 4.5)$

  Consulter la solution...

- $(40, 2, 64)$

  Consulter la solution...

- $(5, 10, 2)$

  Consulter la solution...

Exercice 1 :

Dans chaque cas est-ce que la troisième solution est dans le segment de ligne droite reliant les deux premières?

- $(5, 8, 0),\ (14, 2, 6) :\quad (11, 4, 4)$

  Consulter la solution...

- $(3, 5, 0),\quad (9, 2, 9) :\quad (7, 4, 6)$

  Consulter la solution...

- $(4, 4, 10),\quad (-4, -4, 18) :\quad (-6, -6, 20)$

  Consulter la solution...

Exercice 2 :

Donner une solution réalisable de base d'un programme dont les contraintes sont:

$$\begin{cases} -x_1 - \quad x_2 \leq -4, \\ -3x_1 - 10x_2 \leq -5, \\ -3x_1 + \ 4x_2 \leq 23, \\ x_1, x_2 \geq 0. \end{cases}$$

Consulter la solution...

Exercice 3 :

Donner la solution optimale de
Maximiser $x_1 + 2x_2$ avec les contraintes:

$$\begin{cases} x_1 \leq 10, \\ x_2 \leq 10, \\ x_1 + x_2 \leq 15, \\ x_1, x_2 \geq 0. \end{cases}$$

Consulter la solution...

Module D108 : Programmation mathématique et optimisation
Programmation Linéaire
L'algorithme du simplexe

Auteur(s) : Mike Robson - IUP Miage Bordeaux
Date de dernière modification : 30 Janvier 2006

Dans cette section nous allons regarder un algorithme performant conçu par G. Dantzig en 1947. Il est performant d'une part parce qu'il effectue assez peu d'itérations de son opération de base (avancer d'une solution réalisable de base à une autre meilleure) et d'autre part parce que chacune de ces opérations est effectuée par des calculs algébriques simples.

L'idée géométrique

A chaque itération l'algorithme a une solution réalisable de base (nous allons voir comment la première est trouvée) et essaie de trouver une autre solution réalisable de base qui est adjacente en tant que sommet du polytope de solutions réalisables et meilleure (possède une

meilleure valeure de la fonction économique). S'il n'existe pas de telle solution meilleure adjacente, la solution actuelle est forcément optimale.

Bien que théoriquement ce processus puisse parcourir un nombre exponentiel de sommets du polytope, en pratique on trouve que le nombre réellement parcourus est petit, ce qui permet à l'algorithme de résoudre facilement des programmes avec des centaines de variables.

Une solution de base étant donnée par l'intersection de n des (hyper)plans définis par les contraintes, une arête du polytope passant par cette solution est donnée par l'intersection de $n-1$ parmi ces hyperplans et, donc, une solution adjacente est donnée par l'intersection de $n-1$ des hyperplans plus un nouveau. Si on choisit l'hyperplan à supprimer de telle façon que l'arête va dans une bonne direction (valeur de la fonction économique augmentante), une bonne solution adjacente est donnée par la prochaine intersection entre cette arête et un autre hyperplan donné par une nouvelle contrainte.

Calcul efficace d'une solution réalisable adjacente

Le calcul d'une bonne solution adjacente s'avère très simple dans le cas où la solution actuelle a $x_j = 0$ pour tout j. Notons d'abord que dans ce cas $(0, 0, \cdots, 0)$ étant une solution réalisable, chaque contrainte $\sum_{j=1} a_{i,j} x_j \leq b_i$ a forcément $b_i \geq 0$.

Les arêtes du polytope passant par cette solution sont les n droites sur lesquelles $n-1$ des variables restent nulles. Prenons par exemple celle sur laquelle toutes les variables sauf x_j sont nulles; si la fonction économique est $\sum_{i=1}^{n} c_i x_i$, sa valeur sur cette ligne est donnée par $c_j x_j$ et, donc, devient positive si $c_j > 0$. C'est-à-dire qu'il faut choisir un c_j positif; parmi les c_j positifs, on peut choisir arbitrairement. Toujours sur cette ligne, une contrainte comme $\sum_{j=1} a_{i,j} x_j \leq b_i$ devient simplement $a_{i,j} x_j \leq b_i$ et, si $a_{i,j} \leq 0$, x_j n'est pas contrainte; si $a_{i,j} > 0$, on sait que x_j ne peut pas dépasser $b_i / a_{i,j}$ sans sortir du polytope. Enfin le minimum parmi ces rapports $b_i / a_{i,j}$ pour les $a_{i,j}$ positifs doit donner la valeur de x_j à la solution adjacente.

Ainsi, si la solution actuelle est $(0, 0, \cdots, 0)$, trouver une bonne solution adjacente est facile et rapide. L'astuce de l'algorithme du simplexe est d'ajouter de nouvelles variables de telle façon que toute solution de base est $(0, 0, \cdots, 0)$ par rapport à un sous-ensemble bien choisi de n variables.

Concrètement, pour chaque contrainte $\sum_{j=1} a_{i,j} x_j \leq b_i$ on ajoute une *variable d'écart* x_{n+i} définie par $\sum_{j=1} a_{i,j} x_j + x_{n+i} = b_i$; x_{n+i} sera forcément nulle à une solution de base située sur l'hyperplan $\sum_{j=1} a_{i,j} x_j = b_i$ et sera positive à chaque autre solution réalisable.

Maintenant on a $m+n$ variables et m équations (contraintes d'égalité) et les seules autres contraintes sont que toutes les variables sont ≥ 0; à une solution réalisable de base, on a toujours n des variables nulles et les autres (*les variables de base*), en général, sont positives. A chaque itération on va choisir une nouvelle base afin de garder les deux conditions qui rendaient facile le calcul d'une solution adjacente:

- la fonction économique est donnée par une expression linéaire dans les variables hors base,
- chaque variable de base est donnée par une une expression linéaire dans les variables hors base.

Un calcul assez simple calcule ces expressions linéaires pour la nouvelle base permettant de commencer la prochaine itération.

Les détails du calcul

On a une matrice A de m lignes et $m+n$ colonnes un vecteur colonne b et un vecteur ligne c; les contraintes d'égalité sont $Ax = b$ et la fonction économique est $cx + constante$; les colonnes de A correspondant aux variables de la base constituent une matrice de permutation, c'est-à-dire une matrice carrée (m par m) où tous les éléments sont nuls sauf un élément égal à 1 en chaque ligne et chaque colonne; les éléments de c correspondant à ces mêmes variables sont tous nuls.

- On choisit un élément positif c_j de c; la variable x_j va entrer dans la base; s'il n'y en a pas, la solution actuelle est optimale;
- parmi les $A_{i,j}$ qui sont positifs, on choisit celui pour lequel le rapport $A_{i,j}/b_i$ est maximal; la ligne i a un seul élément $A_{i,k} = 1$ pour x_k une variable de la base; c'est cette variable x_k qui va sortir de la base;
  $A_{i,j}$ est le *pivot* et on va procéder à une opération de *pivotage*:
- on divise la ligne i et b_i par $A_{i,j}$;
- pour chaque i' différent de i on ajoute à la ligne i' et à $b_{i'}$ le multiple de la ligne i et b_i par $-A_{i',j}/A_{i,j}$, ainsi réduisant $A_{i',j}$ à zéro; de la même façon on ajoute à c le multiple $-c_j/A_{i,j}$ de la ligne i, réduisant c_j à zéro.

Et on a réproduit les conditions énoncées pour la nouvelle base.

On peut noter que si, quand on cherche les $A_{i,j}$ positifs, il n'y en a pas, on a trouvé un cas ou la fonction objective n'admette aucune borne supérieure finie.

Trouver une solution réalisable initiale

Si $(0, 0, \cdots, 0)$ n'ést pas une solution réalisable, une astuce permet de trouver une solution de base réalisable (s'il y'en a) avec une première application du simplexe à un programme *auxiliaire* et ensuite de continuer directement à la solution souhaitée avec une deuxième application (méthode en deux *phases*.)

$(0, 0, \cdots, 0)$ n'ést pas une solution réalisable parce que quelques uns des b_j sont strictement négatifs; on ajoute une nouvelle variable artificielle (disons x_a) avec sa contrainte implicite $x_a >= 0$, et pour chaque j avec $b_j < 0$ on modifie la contrainte numéro j en ajoutant $-x_a$ à gauche, les autres contraintes restant inchangées; maintenant les contraintes sont faisables avec (par exemple) $x_a = -min(b_j)$ et toutes les autres variables nulles.

Avec ces contraintes et la solution initiale réalisable de base $x_a = -min(b_j),\ x_i = 0\ (1 \leq i \leq n)$ on maximise la fonction $-x_a$; ceci est le programme auxiliaire. Si on trouve une solution optimale avec $x_a = 0$, on a trouvé une solution réalisable du programme initial; sinon il n'existe pas de telle solution.

Qui plus est, quand on a trouvé une solution avec $x_a = 0$, en supprimant la variable artificielle x_a et en modifiant la fonction objective de départ (avec les mêmes pivotages utilisés dans la première phase), on a directement une solution réalisable de base du programme initial et le tableau qu'il faut pour y appliquer la méthode du simplexe.
(On va voir un exemple au lieu de démonstration.)

Un problème : les solutions dégénérées

Un petit problème embêtant est celui des solutions dégénérées: quand une solution de base est l'intersection d'un nombre strictement supérieur à n de contraintes, il est possible que l'opération de pivotage remplace une de ces contraintes par une autre, produisant une amélioration nulle dans la fonction objective, et même que celui-ci se réproduit dans une boucle de façon que l'algorithme ne termine pas. Parmi plusieurs solutions possibles à ce problème est la règle de Bland: parmi les candidats pour la colonne ou la ligne du pivot, quand il y'en a plusieurs, choisir toujours celui de l'indice minimum. Cette règle est opposée à l'intuition qui dit qu'il faudrait choisir le coefficient maximum dans la fonction objective mais le risque de non terminaison est sérieux; en effet des programmes réels sont souvant de type dégénéré.

Module D108 : Programmation mathématique et optimisation
Programmation Linéaire
Le programme dual d'un programme linéaire
La relation entre les solutions des deux programmes
L'algorithme du simplexe trouve les deux solutions !
Interprétation économique de la solution du dual
Exercices

Auteur(s) : Mike Robson - IUP Miage Bordeaux
Date de dernière modification : 30 Janvier 2006

A un programme linéaire exprimé dans la forme

$$\textit{maximiser } cx \textit{ sous les contraintes } Ax <= b,\ x >= 0$$

on associe son programme *dual* qui est

$$\textit{minimiser } by \textit{ sous les contraintes } A^T y >= c, y >= 0\ ;$$

ici A^T est la transposée de la matrice A. Par exemple le dual du programme: (des exercices de la section 3)

maximiser $2x_1 + 3x_2 + x_3$ avec les contraintes

$$\begin{cases} x_1 & +2x_2 & +5x_3 & \leq 10 \\ 4x_1 & -2x_2 & +x_3 & \leq 20 \\ -4x_1 & & +3x_3 & \leq 15 \\ 3x_1 & +6x_2 & -5x_3 & \leq 20 \end{cases}$$

serait:

minimiser $10y_1 + 20y_2 + 15y_3 + 20y_4$ avec les contraintes

$$\begin{cases} y_1 & +4y_2 & -4y_3 & +3y_4 & \geq 2 \\ 2y_1 & -2y_2 & & +6y_4 & \geq 3 \\ 5y_1 & +y_2 & +3y_3 & -5y_4 & \geq 1 \end{cases}$$

On peut transformer le dual dans la forme familière d'une maximisation sous contraintes de la forme $<=$ par deux multiplications par -1 :

$$\textit{maximiser } -by \textit{ sous les contraintes } -A^T y <= -c, y >= 0$$

Pour notre exemple cette transformation donne:

maximiser $-10y_1 - 20y_2 - 15y_3 - 20y_4$ **avec les contraintes**

$$\begin{cases} -y_1 & -4y_2 & +4y_3 & -3y_4 & \leq -2 \\ -2y_1 & +2y_2 & & -6y_4 & \leq -3 \\ -5y_1 & -y_2 & -3y_3 & +5y_4 & \leq -1 \end{cases}$$

On voit clairement qu'effectuer cette transformation deux fois revient au programme de départ: la relation de dualité est une relation réciproque.

La relation entre les solutions des deux programmes

On retrouve le programme dual en cherchant à démontrer algébriquement un majorant sur les solutions réalisables du programme initial. Par exemple dans notre programme exemple nous pouvons facilement démontrer le majorant grossier de 31 sur la fonction objective en choisissant des multiplicateurs $m_1 = 1,\ m_2 = 0.5,\ m_3 = 0.2,\ m_4 = 0.4$ pour les lignes, ce qui donne les nouvelles inéquations:

$$\begin{cases} x_1 & +2x_2 & +5x_3 & \leq 10 \\ 2x_1 & -x_2 & +0.5x_3 & \leq 10 \\ -0.8x_1 & & +0.6x_3 & \leq 3 \\ 1.2x_1 & +2.4x_2 & -2x_3 & \leq 8 \end{cases}$$

et ensuite ajoutant ces inéquations: $3.4x_1 + 3.4x_2 + 4.1x_3 <= 31$ d'où on déduit que $2x_1 + 3x_2 + x_3 <= 31$ puisque les x_j ne peuvent pas être négatifs; c'est-à-dire que 31 est un majorant sur la fonction objective.

Si on pose la question pour quels multiplicateurs ce processus est valable, on voit qu'on doit forcément avoir que:

- $m_j >= 0$; si on multiplie une inéquation par une valeur négative, elle ne reste pas vraie;
- $\sum m_j a_{i,j} >= c_j$; par exemple dans notre exemple on a utilisé le fait que $3.4 >= 2,\ 3.4 >= 2,\ 4.1 >= 1$.

C'est-à-dire que les m_j constituent une solution réalisable du dual. Qui plus est, le majorant calculé par ce processus est exactement $\sum m_j c_j$, la fonction objective du dual (dans sa première forme avant les multiplications par -1). Donc les majorants qu'on peut démontrer de cette façon sont exactement les valeurs de la fonction objective des solutions réalisables du dual et, en particulier, la solution optimale du dual donne le meilleur majorant qu'on puisse obtenir de cette façon.

Reste la question: existe-t-il de meilleurs majorants qu'on ne peut pas démontrer de cette façon simple? La réponse est que non mais, au lieu de la démontrer ici, nous allons la traiter dans la section suivante.

Si on avait commencé avec un programme qui n'avait pas de solution optimale finie (donc, programme non faisable ou sans majorant sur la fonction objective), le dual aurait aussi eu ce caractère. Il n'est pas possible que les deux soient sans majorant mais les trois autres cas (infaisable-infaisable, infaisable-sans majorant ou sans majorant-infaisable) sont possibles.

L'algorithme du simplexe trouve les deux solutions !

Dans notre programme exemple, le simplexe donne le tableau final:

```
 1.00    0.00    0.37    0.00    -0.02    -0.11    -
0.15    0.53
 0.00    1.00   -0.80    0.00    -
0.20     0.10     0.00    0.10
 0.00    0.00   -0.39    1.00    -0.06    -
0.09     0.05    0.35
 0.00    0.00   35.10    0.00    -5.40    -1.05    -
0.50
```

solution
variable 1= 0.53
variable 2= 0.10
variable 3= 0.00
variable 4= 0.35
pour un resultat de 14.45

ou pour le dual:

```
 0.00   0.00   1.00    0.15    0.00   0.00   -
0.05    0.50

1.00   0.00   0.00    0.02    0.20   0.00    0.06    5.
40
```

```
 0.00    0.00    0.00    -
0.37     0.80    1.00     0.39    35.10
 0.00    1.00    0.00     0.11    -
0.10    0.00     0.09     1.05
 0.00    0.00    0.00    -0.53    -0.10    0.00    -0.35
```

solution
variable 1= 5.40
variable 2= 1.05
variable 3= 0.50
pour un resultat de 14.45

On voit que sauf les multiplications par -1, la solution trouvée dans l'un cas se trouve dans le tableau final de l'autre cas dans la ligne ``fonction économique'' dans les colonnes des variables d'écart. L'explication de ce phénomène en général n'est pas trop compliquée.

- à chaque itération chacune des lignes $1..m$ du tableau contient une combinaison linéaire des valeurs initiales de ces lignes; donc la dernière ligne (celle de la fonction) contient sa valeur initiale moins une combinaison linéaire des autres lignes; mettons m_i pour le coefficient de la ligne i dans cette combinaison linéaire après la dernière itération, c'est-à-dire
$$c(final)_j = c(initial)_j - \sum_{i=1}^{m} m_i a_{i,j}$$
.
- donc, en regardant la colonne $n+j$ on obtient $m_j = -c(final)_{n+j}$.
- donc, parce que l'algorithme ne halte que quand chaque $c(final)_j <= 0$, on a que chaque $m_j >= 0$.
- et, en regardant les autres colonnes,
$$c(initial)_i <= \sum_{j=1}^{m} m_j a_{i,j}$$
- alors, les m_j constituent une solution réalisable du programme dual.

- la valeur de z trouvée pour le programme initial est de $\sum_{j=1}^{m} m_j b_j$ qui est exactement la valeur de son z de la solution réalisable du dual ($m_1, ..m_j$); et parce que les solutions du dual donnent des majorants sur les solutions du primal, cette valeur de z est forcément la solution optimale commune des deux programmes et les deux solutions trouvées sont chacune optimale pour son programme.

Interprétation économique de la solution du dual

La solution du programme dual peut nous donner des informations utiles sur comment la solution du programme initial va changer si les contraintes changent. Rappelons que si m_i est

une valeur de la variable y_i dans la solution optimale du dual, $-m_i$ est le coefficient final dans z de la variable d'écart numéro i dans le tableau du primal. Donc, si cette variable est incrémentée par ϵ, la valeur de z sera décrémentée par $m_i\epsilon$; donc, si on pouvait décrémenter la variable d'écart, z serait incrémentée par $m_i\epsilon$. Cela veut dire que si la $i$ème contrainte est relachée par ϵ (remplacant b_i par $b_i + \epsilon$), la valeur de z peut être augmentée par $m_i\epsilon$. Si cette contrainte exprime une quantité limité d'une ressource (matière de base, main d'oeuvre etc.) on peut déduire que m_i est le prix maximum qu'on puisse raisonnablement payer pour avoir plus de cette ressource. Cette conclusion n'est valable que pour les ϵ suffisamment petits que déplacer la contrainte par ϵ ne change pas la topologie du polytope. L'intervalle de ces ϵ peut être étroite ou même nulle!

Module : Modélisation des problèmes économiques
Optimisation mathématique
Exercices

Préambule

Le paradigme d'optimisation mathématique permet une modélisation adéquate d'un bon nombre de problèmes physiques ou économiques, en conséquence, une étude systématique du problème qui peut aboutir à une solution complète. Dans cette UE, nous nous sommes occupé des cas particuliers d'optimisation d'une fonction linéaire face à des contraintes linéaires. L'algorithme du simplexe permet de calculer efficacement la solution optimale. Mieux encore, des techniques complémentaires en programmation linéaire rendent possible une étude approfondie de la sensibilité de la solution trouvée face aux problèmes possibles.

Dans le cas général des problèmes modélisés, suivant leur complexité de résolution, ceux-ci peuvent être divisés en deux catégories : Celle des problèmes faciles et celle de problèmes difficiles. Un problème de programmation linéaire fait partie de la première catégorie. La mise en équation d'un problème d'optimisation peut aussi poser des difficultés supplémentaires. En effet, le problème initial est décrit en langage usuel et sa mathématisation sous une forme simple peut s'avérer ardue.

L'objectif de cette étude n'est pas d'aborder ces difficultés dans leurs généralités, pour lesquelles il n'existe pas de recette universelle, mais de faciliter quelques exemples isotypes. Les exemples suivants sont choisis parmi les plus simples et ont pour but de donner quelque entraînement au lecteur. Ils sont empruntés aux ouvrages suivants :

1. Roseaux, « Exercices et Problèmes Résolus de Recherche opérationnelle », Tome 3, Masson 1985.
2. M. Sakarovitch, « Optimisation Combinatoire, Graphes et Programmation Linéaires », Hermann, 1984.

Exemple 1 : Problème de Production

1. Une usine fabrique deux produits P1 et P2.

   Chacun de ces produits demande, pour son usinage, des heures de fabrication unitaires sur les machines (ou dans les ateliers) A B C D E comme indiqué dans le tableau suivant :

	A	B	C	D	E
P1	0	1h,5	2	3	3
P2	3	4	3	2	0
Disponibilité totale de chaque machine	39h	60h	57h	70h	57h

   Les marges brutes de chaque produit sont respectivement :
   M1 = 1 700 F
   M2 = 3 200 F

   Ecrire un programme linéaire correspondant.

2. Les produits utilisent trois fournitures F1, F2 et F3 dans les conditions indiquées ci-dessous :

	F1	F2	F3
P1	0	12	8
P2	5	36	0
Unités	Kg	M^3	M^2
Stock disponible	55	432	126

3.
4. Réécrire sous forme algébrique seulement, le nouveau programme linéaire ainsi créé. Eliminer les contraintes redondantes.

Cliquez pour afficher la réponse correspondante (réponse).

Exemple 2 :Composition d'Aliments pour le Bétail.

On désire déterminer la composition, à coût minimal, d'un aliment pour bétail qui est obtenu en mélangeant au plus trois produits bruts : orge, arachide, sésame. L'aliment ainsi conditionné devra comporter au moins 22% de protéines et 3,6% de graisses,

pour se conformer aux exigences de la clientèle. On a indiqué ci-dessous les pourcentages de protéines et de graisses contenus, respectivement, dans l'orge, les arachides et le sésame, ainsi que le coût par tonne de chacun des produits bruts :

Produit brut	1	2	3	Pourcentage requis
	ORGE	ARACHIDES	SESAME	
Pourcentage de protéines	12%	52%	42%	22%
Pourcentage de graisses	2%	2%	10%	3,6%
coût par tonne	25	41	39	

1. On notera Xj (j = 1, 2, 3) la fraction de tonne de produit brut j contenu dans une tonne d'aliment. Formuler le problème algébriquement.
2. Montrer qu'il est possible de réduire la dimension du problème.

Cliquez pour afficher la réponse correspondante (réponse).

Exemple 3 : Crème Glacée

Un fabricant désire produire 100 Kg d'une préparation de base pour crème glacée. Cette préparation doit contenir 21,5 Kg de matières grasses, 21 Kg de sucre, 1,2 Kg d'œuf et 53 Kg d'eau. Les ingrédients dont il dispose figurent en tête des colonnes du tableau ci-dessous ; les constituants figurent en ligne. Ce tableau précise également les pourcentages (en poids) de chaque constituant dans chaque ingrédient ainsi que le coût, au Kg, de chaque ingrédient.

		Ingrédients					
		Crème	Jaune d'oeuf frais	Lait entier en poudre	Jaune d'oeuf surgelé et sucré	Sirop de sucre de canne	eau
Constituants	Matières grasses	40	50	12	30		
	Sucre				14	70	
	Oeuf		40		40		
	Eau	60	10	88	16	30	100
	Coût en Kg	3 F	4 F	1 F	2 F	0,80 F	0,00 F

3. Le fabricant désire déterminer la composition du mélange de coût minimal. Ecrire le programme linéaire correspondant à ce problème, sans le résoudre.

4. Jusqu'à présent, le fabricant produisait le mélange suivant :

CREME	50 Kg
JAUNE D'OEUF FRAIS	3 Kg
SIROP	30 Kg
EAU	17 Kg

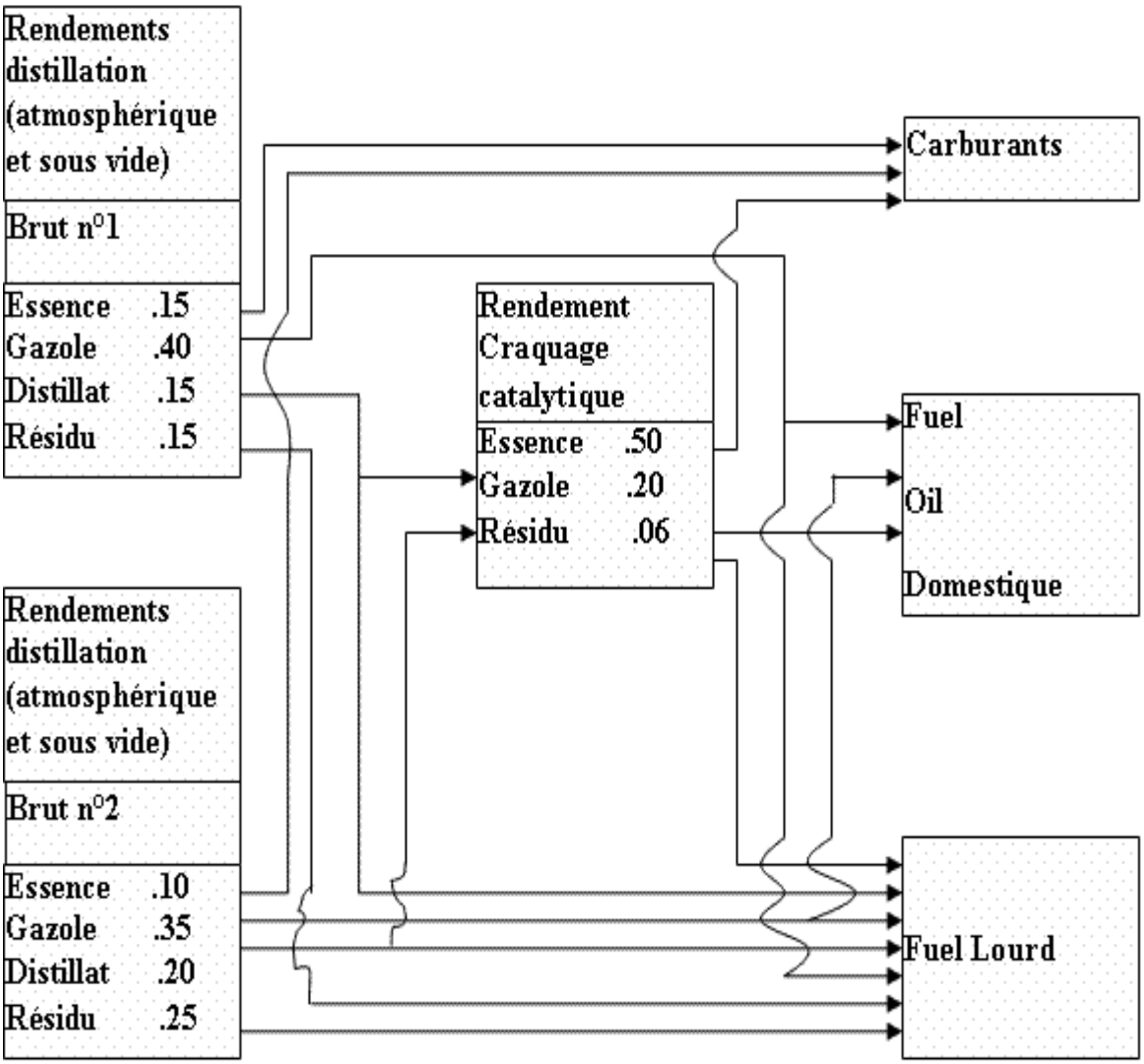
5. Quel est le coût de ce mélange ? Peut-on dresser le tableau du simplexe associé à cette solution ? Pourquoi ? Si oui, le faire.
6. Quelle est la composition du mélange de coût minimal ?
7. Par suite de fluctuations économiques, les prix de la crème et du jaune d'œuf frais passent respectivement à 4 et 7 F.
   Le mélange déterminé à la question 3 est-il toujours de coût minimal ?

Cliquez pour afficher la réponse correspondante (réponse).

Exemple 4 :Schématisation d'une Raffinerie

Une raffinerie, représentée sur le schéma ci-dessous, dispose d'une unité de distillation (permettant d'obtenir quatre produits : carburants, gazole, distillat lourd, résidu), une unité de reformage et une unité de craquage catalytique (pouvant traiter les distillats lourds).

Pendant une période donnée, la raffinerie peut traiter deux pétroles bruts B1 et B2 dont les rendements sont indiqués sur le schéma (pour les carburants de rendements indiqués correspondent aux produits finis après passage par l'unité de reformage). Les prix de B1 et B2 sont respectivement de 1 300 F/t et de 1 500 F/t. Les frais de distillation sont estimés à 10 F/t. Le coût de traitement du distillat sous vide au craquage catalytique est de 20 F/t. On distingue seulement 3 types de produits : la raffinerie doit fabriquer pendant la période considérée, au moins, 200 000 t de carburants ; 400 000 t de fuel-oil domestique et 250 000 t de fuel lourd ; elle cherche à minimiser son coût de production.



Par ailleurs, les produits finis doivent satisfaire certaines contraintes de qualités :
- la teneur en soufre du fuel-oil domestique doit être inférieure à 0,5%, la teneur en soufre du gasoil issu de la distillation du brut n°1 étant de 0,2% en poids et de 1,2% pour le brut n°2, la teneur de gasoil de craquage étant de 0,3% en poids pour le brut n°1 et de 2,5% en poids pour le brut n°2.
Enfin le raffineur doit traiter au moins 550 000 t de brut n°1 et la capacité du craquage est de 200 000 t pour la période considérée.

Formuler le problème en terme de programmation linéaire ; on pourra donner un nom mnémonique aux variables.

Cliquez pour afficher la réponse correspondante (réponse).

Exemple 5 :

Un physicien ayant fait n expériences, s'est rendu compte qu'une certaine quantité Q est fonction de la variable t. Il a de bonnes raisons de penser que la loi reliant Q à t est

de la forme :
Q(t) = a sin(t) + b tg(t) + c
, où les paramètres a et b sont à choisir numériquement « au ieux » à partir des n expériences réalisées : celles –ci, pour n valeurs différentes de t: t_1, t_2,..., et t_n ont donné pour Q, les valeurs Q_1, Q_2,..., Q_n.
Tenant compte du fait que ces n expériences sont entachées d'erreur et que la loi prévue n'est peut-être qu'approximative, le physicien n'espère pas trouver un ajustement parfait (le système d'équations a sin(t_i) + b tg(t_i) + c = Q_i (i = 1, 2, …, n) aux inconnues a, b et c n'admet pas de solution). Le physicien a deux idées différentes de ce qu'un meilleur ajustement peut signifier :
(1) la première idée consiste à chercher a, b, c minimisant
$\sum_{i=1}^{n} |a\ sin(t_i) + b\ tg(t_i) + c - Q_i|$

(2) La seconde idée consiste à chercher a, b, c minimisant :
$Max_{i=1,2,\ldots,n}[|a\ sin(t_i) + b\ tg(t_i) + c - Q_i|]$

Par ailleurs, et pour des raisons physiques, les coefficients a, b, c doivent être supérieurs ou égaux à zéro.
Montrer que dans chacun des cas (1) et (2) on peut formuler le problème comme un programme linéaire qu'on explicitera.

Cliquez pour afficher la réponse correspondante (réponse).