Exemple 1

Une compagnie fabrique deux types d'acier : l'un de type α et l'autre de type β. Le profit pour une tonne d'acier de type α (respectivement type β) est 6 (resp. 15). De plus, il faut 2 (resp. 3) tonnes de matières premières et 4 (resp. 2) unités de temps pour produire une tonne d'acier α (resp. β). Finalement, la compagnie dispose de 120 tonnes de matières premières et de 100 unités de temps.

- *Déterminier le plan de production optimal (c'est à dire la quantité qu'il faut fabriquer de chaque acier pour maximiser le profit tout en satisfaisant toutes les contraintes du problème).*

Modélisation

	Profit	matières 1ères	temps
Acier type α	6	2	4
Acier type β	15	3	2
Disponible		120	100

- Programme linéaire associé :

$$\left\{\begin{array}{llcrcl} \max & 6x_1 & + & 15x_2 & & \\ s.c. & 2x_1 & + & 3x_2 & \leq & 120 \\ & 4x_1 & + & 2x_2 & \leq & 100 \\ & x_1 & , & x_2 & \geq & 0 \end{array}\right.$$

Programme linéaire sous forme standard

$$
\begin{cases}
\max & 6x_1 & + & 15x_2 & & & & & & \\
s.c. & 2x_1 & + & 3x_2 & + & x_3 & & & = & 120 \\
 & 4x_1 & + & 2x_2 & & & + & x_4 & = & 100 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq & 0
\end{cases}
$$

Algorithme du simplexe (méthode des tableaux)

- Tableau initial

	Base	$\overline{A}^1$	$\overline{A}^2$	$\overline{A}^3$	$\overline{A}^4$	$\overline{b}$
$\rightarrow$	A^3	2	3	1	0	120
	A^4	4	2	0	1	100
	$(\overline{c}, -c_I^t\overline{a})$	6	15	0	0	0
		$\uparrow$				

- Ligne du pivot normalisée

Base	$\overline{A}^1$	$\overline{A}^2$	$\overline{A}^3$	$\overline{A}^4$	$\overline{b}$
A^3	2/3	1	1/3	0	40
A^4	4	2	0	1	100
$(\overline{c}, -c_I^t\overline{a})$	6	15	0	0	0

- Changement de base

Base	$\overline{A}^1$	$\overline{A}^2$	$\overline{A}^3$	$\overline{A}^4$	$\overline{b}$
A^2	2/3	1	1/3	0	40
A^4	8/3	0	−2/3	1	20
$(\overline{c}, -c_I^t\overline{a})$	−4	0	−5	0	−600

Variables duales

- $\pi^t = c_I^t (A^I)^{-1}$

  - $I = \{2, 4\}$,
  - $c^t = \begin{pmatrix} 6 & 15 & 0 & 0 \end{pmatrix}$,
  - $A = \begin{pmatrix} 2 & 3 & 1 & 0 \\ 4 & 2 & 0 & 1 \end{pmatrix}$,
  - $a^t = \begin{pmatrix} 120 & 100 \end{pmatrix}$,

  - $\begin{pmatrix} 15 & 0 \end{pmatrix} \begin{pmatrix} 1/3 & 0 \\ -2/3 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 0 \end{pmatrix}$

- $\pi = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$

Coûts réduits

- $\overline{c} = c - \pi A$

  - $\overline{c}_1 = c_1 - \pi A^1 = 6 - \begin{pmatrix} 5 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \end{pmatrix} = -4$
  - $\overline{c}_2 = c_2 - \pi A^2 = 15 - \begin{pmatrix} 5 & 0 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 0$
  - $\overline{c}_3 = c_3 - \pi A^3 = 0 - \begin{pmatrix} 5 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -5$
  - $\overline{c}_4 = c_4 - \pi A^4 = 0 - \begin{pmatrix} 5 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0$

- $\overline{c} = \begin{pmatrix} -4 & 0 & -5 & 0 \end{pmatrix}^t$

Exemple 1 (suite)

La compagnie souhaite mettre sur le marché un nouveau type d'acier (type γ). Pour en produire une tonne, il faut 2 tonnes de matières premières et 3 unités de temps.

- *Quelle est la valeur minimale c_γ du profit d'une tonne d'acier de type γ pour que sa production soit rentable ?*

Coûts réduits

- $\overline{c} = c - \pi A$

  - $$\begin{aligned} \overline{c}_\gamma &= c_\gamma - \pi A^\gamma = c_\gamma - \begin{pmatrix} 5 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} \\ &= c_\gamma - 10 \end{aligned}$$

- Production de l'acier $\gamma \Longleftrightarrow c_\gamma \geq 10$

Principe de la génération de colonnes

Considérons le programme linéaire :

$$(P)\quad \begin{cases} \min & \sum_{j=1}^{n} c_j x_j & & \\ s.c. & \sum_{j=1}^{n} A^j x_j & = & a \\ & x_j & \geq & 0, \qquad j = 1, \ldots, n \end{cases}$$

où A^j et a sont des m-vecteurs ($m < n$).

- x_I solution de base réalisable associée à la base I
- c_I le coût associé
- $\pi = c_I \left(A^I\right)^{-1}$ le multiplicateur du simplexe
  où chaque composante π_i est la variable duale associée à la contrainte $A_i x = a_i$, $i = 1, \ldots, m$.

Améliorer cette solution de base ?

Coûts réduits des variables hors-base

$$\overline{c}_j = c_j - \pi A^j = c_j - c_I \left(A^I\right)^{-1} A^j$$

Si

$$\min\left\{\overline{c}_j \mid j \in \overline{I}\right\} = \overline{c}_s < 0$$

Alors la solution courante peut-être améliorée en faisant entrer x_s dans la base via un pivotage

- Plus généralement,
  supposons que les colonnes A^j sont dans un ensemble S contenant les m-vecteurs satisfaisant les contraintes, le **sous-problème** générateur de colonnes s'écrit :

$$\min\left\{c_j - \pi A^j \mid A^j \in S\right\} = v(\pi)$$

Problème de découpe (Gilmore et Gomory)

À partir de rouleaux (de papier, d'aluminium, ...) de longueur standard L, le problème consiste à découper une commande caractérisée par la donnée de a_i pièces de longueurs l_i pour $i = 1, \ldots, m$.

Le but est de satisfaire la commande $b = (b_1, \ldots, b_m)$ tout en minimisant le nombre total de rouleaux à découper. Chaque découpe de rouleau induit des chutes (pertes). Il s'agit donc de minimiser ces chutes.

→ Première Modélisation.

Définissons un schéma (plan) de découpe comme une manière de découper un rouleau de longueur L en a_1 pièces de longueurs l_1, a_2 pièces de longueurs l_2, etc.

Un schéma de découpe est réalisable si

$$a_1 l_1 + a_2 l_2 + \cdots + a_m l_m \leq L.$$

Problème : *Spécifier le nombre de fois qu'un plan de découpe doit être utilisé pour satisfaire la demande tout en minimisant le nombre de rouleaux.*

Un plan de découpe est déterminé par la donnée du vecteur

$$(a_1, \dots, a_m)$$

Il existe un nombre fini de plans de découpe mais ce nombre peut-être très grand.

Associons à chaque schéma de découpe un vecteur colonne

$$A^j = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}$$

- A^j_i = nbre de pièces de longueurs l_i découpées par le plan j
- x_j = nombre de fois que le schéma j est utilisé
- $b = (b_1, \dots, b_m)$ vecteur demandes pour les longueurs $l_1, \dots, l_m$.

La contrainte de satisfaction des demandes s'écrit :

$$\sum_j A^j x_j \geq b$$

La fonction objectif est :

$$\min \sum_j x_j$$

qui minimise le nombre total de rouleaux de longueur L à découper.

Exemple.

À partir d'un nombre minimum de rouleaux de 100m, on veut obtenir :

- 97 pièces de 45m
- 610 pièces de 36m
- 395 pièces de 31m
- 211 pièces de 14m

Quelques exemples de schémas de découpe réalisables :

$$A = \begin{pmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 3 & 2 & 1 & 0 & 2 & 1 & 0 & 0 & 2 & 1 & 0 & 4 & 3 \end{pmatrix}$$

Si x_j est entier, alors le problème est très difficile.

En fait, pour les problèmes avec une grande demande, on peut considérer que l'arrondi d'une fraction à l'optimum est négligeable.

On relache alors cette contrainte dans ce qui suit.

Cependant, le problème demeure difficile car le nombre de colonnes A^j est très important.

Par exemple, si les rouleaux à découper sont de longueur $L = 200$ et s'il y a des demandes pour 40 longueurs différentes de 20 à 80, alors le nombre possible de plans de découpe peut dépasser 10 à 100 millions.

Étant donné le problème de découpe écrit sous la forme suivante :

$$\begin{cases} \min & \sum_j x_j \\ s.c. & \sum_j A^j x_j \geq b \\ & x_j \geq 0 \end{cases} \tag{1}$$

Les coûts réduits sont donnés par

$$\overline{c}_j = 1 - \pi A^j = 1 - \sum_i \pi_i A_i^j$$

où π est le m-vecteur multiplicateur du simplexe associé aux contraintes (1).

Une colonne A^j (ou un vecteur $(a_1, \cdots, a_m)^t$) décrit un schéma réalisable, si

$$\begin{cases} A_i^j = a_i \text{ entier non négatif} \\ \sum_{i=1}^{m} A_i^j l_i = \sum_{i=1}^{m} a_i l_i \leq L \end{cases}$$

Le problème de minimisation du coût réduit

$$\overline{c}_j = 1 - \pi A^j$$

sur l'ensemble de toutes les colonnes

(les schémas de découpes possibles) devient :

$$(K) \quad \begin{cases} \max & \sum_i \pi_i a_i \\ s.c. & \sum_{i=1}^{m} l_i a_i \leq L \\ & a_i \in \mathbb{N} \qquad i = 1, \ldots, m \end{cases}$$

(problème de sac à dos)

Soit $v(K(\pi))$ la valeur optimale du programme (K),

le critère de candidature du simplexe devient

$$\overline{c}_s = \min \overline{c}_j < 0 \qquad \Longleftrightarrow \qquad v(K(\pi)) = \sum_{i=1}^{m} \pi_i a_i^* > 1.$$

Le schéma de découpe

$$A^s = (a_1^*, \cdots, a_m^*)^t$$

correspondant est formé puis entre dans la base.

Sinon, et si les coûts réduits $\overline{c}_j$ des variables de surplus verifient

$$\overline{c}_j = \pi_j \geq 0$$

la solution courante est optimale.

Rappels

Théorème 1 *Soit X un ensemble convexe compact de $\mathbb{R}^n$, et soit $E(X)$ l'ensemble des points extrêmes avec $Co(E(X))$ l'enveloppe convexe de $E(X)$, alors $Co(E(X)) = X$.*

Théorème 2 *Soit $X = \{x \mid Bx = b,\ x \geq 0\}$ un ensemble borné non vide. Soient y^i, $i = 1, \ldots, r$, les points extrêmes. Alors $\forall x \in X$, $\exists \lambda_i \geq 0$, $i = 1, \ldots, r$ tels que $x = \sum_{i=1}^{r} \lambda_i y^i$ et $\sum_{i=1}^{r} \lambda_i = 1$.*

Le résultat du Théorème précédent peut-être étendu au cas non borné.

Théorème 3 *Soit $X = \{x \mid Bx = b,\ x \geq 0\}$ un ensemble borné non vide. $x \in X$ si et seulement si il peut s'écrire comme une combinaison convexe des points extrêmes de X plus une combinaison linéaire non négative des rayons extrêmes de X, i.e.*

$$x = \sum_i \lambda_i y^i$$

où $\sum_i \lambda_i \delta_i = 1$, *et* $\delta_i = \begin{cases} 1 & \text{si } y^i \text{ est un point extrême} \\ 0 & \text{si } y^i \text{ est un rayon extrême de } X \end{cases}$

Décomposition de Dantzig-Wolfe (1960)

Soit (P) un porogramme linéaire où les contraintes sont partitionnées en deux sous-ensembles :

$$(P) \quad \left\{ \begin{array}{llcll} \min & cx & & & \\ s.c. & Ax & = & a & (m_1 \text{ contraintes}) \\ & Bx & = & b & (m_2 \text{ contraintes}) \\ & x & \geq & 0 & \end{array} \right.$$

et supposons que le polyèdre

$$X = \{x \mid Bx = b, \ x \geq 0\}$$

est borné ;

Il contient donc un nombre fini de points extrêmes

(indicés par $j \in \Omega$).

On sait que

$$\forall x \in X, \qquad x = \sum_{j \in \Omega} \lambda_j y^j$$

avec y^j les points extrêmes de X

et $\sum_{j \in \Omega} \lambda_j = 1, \ \lambda_j \geq 0, \ \forall j \in \Omega.$

En substituant dans (P)

$$Ax = a \quad \Longrightarrow \quad \sum_{j\in\Omega} (Ay^j)\lambda_j = a$$

et

$$cx \quad \Longrightarrow \quad \sum_{j\in\Omega} (cy^j)\lambda_j.$$

le programme (P) devient :

$$\left\{\begin{array}{llcll} \min & \sum_{j\in\Omega}(cy^j)\lambda_j & & & \\ s.c. & \sum_{j\in\Omega}(Ay^j)\lambda_j & = & a & (m_1 \text{ contraintes}) \\ & \sum \lambda_{j\in\Omega} & = & 1 & (1 \text{ contrainte}) \\ & \lambda_j & \geq & 0 & \forall j \in \Omega \end{array}\right.$$

En dénotant par

$$\hat{c}_j = cy^j$$

le coût du point extrême y^j

$$\hat{A}^j = Ay^j$$

la contribution du point extrême y^j aux contraintes $Ax = a$,

on obtient le modèle suivant :

$$(PM) \quad \begin{cases} \min & \sum_{j\in\Omega} \hat{c}_j \lambda_j & & \\ s.c. & \sum_{j\in\Omega} \hat{A}^j \lambda_j & = & a \\ & \sum_{j\in\Omega} \lambda_j & = & 1 \\ & \lambda_j & \geq & 0\,, \qquad \forall j \in \Omega \end{cases}$$

appelé **problème maître** équivalent au problème d'origine.

Ce nouveau modèle possède $m_1 + 1$ contraintes comparativement à $m_1 + m_2$ contraintes.

Par contre, il contient beaucoup plus de variables.

Si $m_2 \gg 0$ le gain est de l'ordre de m_2, mais en contrepartie, le problème maître contient beaucoup plus de colonnes, chacune décrivant un point extrême de X

En pratique, il n'est pas facile d'énumérer tous les points extrêmes d'un polyèdre. La technique de résolution consiste à ne générer qu'un sous-ensemble restreint de points extrêmes de X afin de déterminer une solution optimale de (PM).

Soit $\Omega' \subset \Omega$, un sous-ensemble des indices de Ω.

On définit le **problème maître restreint** de la décomposition de Dantzig-Wolfe par :

$$(PM')\quad \left\{ \begin{array}{llll} \min & \sum_{j\in\Omega'} \widehat{c}_j \lambda_j & & \\ s.c. & \sum_{j\in\Omega'} \widehat{A}^j \lambda_j & = & a \\ & \sum_{j\in\Omega'} \lambda_j & = & 1 \\ & \lambda_j & \geq & 0\ , \quad \forall j \in \Omega' \end{array} \right.$$

Résolution de (PM')

$\rightarrow$ les variables duales $\pi \in \mathbb{R}^{m}1$ associées aux contraintes couplantes

$$\sum_{j\in\Omega'} \widehat{A}^j \lambda_j = a$$

$\rightarrow$ la variable duale $\pi_0 \in \mathbb{R}$ associée à la contrainte

$$\sum_{j\in\Omega'} \lambda_j = 1$$

Pour effectuer une itération l'algorithme du simplexe,

il suffit d'évaluer les coûts réduits des variables λ_j, pour $j \in \Omega \setminus \Omega'$,

$$\overline{c}_j = \widehat{c}_j - \pi \widehat{A}^j - \pi_0 = cy^j - \pi A y^j - \pi_0 , \qquad j \in \Omega \setminus \Omega'$$

Si ces coût réduits sont non-négatifs,

Alors la solution de (PM') est optimale pour (PM),

et par conséquent pour le problème original (P).

Sinon, on peut adjoindre à Ω' une nouvelle variable de coût réduit négatif et résoudre à nouveau le problème maître restreint.

Ce processus se termine lorsqu'il n' y a plus de variables λ_j, $j \in \Omega \setminus \Omega'$, ayant un couût réduit négatif.

Une itération de la Génération de Colonnes :

Il faut donc résoudre un **sous-problème** permettant de déterminer si oui ou non il existe des variables ayant un coût réduit négatif.

Il s'agit de résoudre le sous-problème :

$$\overline{c}_j(\pi, \pi_0) = \min_{j \in \Omega \setminus \Omega'} \overline{c}_j = \min_{j \in \Omega \setminus \Omega'} cy^j - \pi A y^j - \pi_0 .$$

Il s'agit de résoudre le sous-problème :

$$\overline{c}_j(\pi, \pi_0) = \min_{j \in \Omega \setminus \Omega'} \overline{c}_j = \min_{j \in \Omega \setminus \Omega'} cy^j - \pi A y^j - \pi_0.$$

$$= \min_{j \in \Omega \setminus \Omega'} (c - \pi A)\, y^j - \pi_0 .$$

Puisque le problème restreint courant a été résolu à l'optimalité,

$$\overline{c}_j \geq 0 \;, \forall j \in \Omega'$$

On peut donc déterminer le minimum sur Ω, i.e. l'ensemble des points extrêmes de X.

Or pour choisir λ_s, il suffit de résoudre

$$(SP) \quad \begin{cases} \min & (c - \pi A)x - \pi_0 \\ s.c. & Bx = b \\ & x \geq 0 \end{cases}$$

puisque la solution optimale est atteinte en un point extrême.

Soit x^s une solution optimale de (SP), la colonne entrante dans la base du problème maître restreint est donnée par

$$\begin{bmatrix} \widehat{A}^s \\ 1 \end{bmatrix} = \begin{bmatrix} Ay^s \\ 1 \end{bmatrix}$$

avec le coût

$$\hat{c}_s = cy^s.$$

Algorithme de décomposition Dantzig-Wolfe :

1. *Initialisation* : Introduire un ensemble de variables correspondant à des points extrêmes connus et compléter au besoin la base du problème maître restreint par des variables artificielles.

2. *Étape 1* : Résoudre le problème maître restreint pour déterminer une solution optimale et les multiplicateurs du simplexe associés : $\pi \in \mathbb{R}^m$ et $\pi_0 \in \mathbb{R}$.

3. *Étape 2* : Résoudre le sous- problème (SP). Si $v(SP) \geq 0$, alors la solution courante du problème maître restreint est optimale pour le problème maître. Sinon, un point point extrême est candidat pour augmenter la taille de Ω'.

4. *Étape 3* : Construire le nouveau problème maître restreint. Retourner à l'étape 2.

En pratique, il est avantageux d'introduire dans le problème maître restreint plusieurs variables de coût réduit négatif.

En fait, on cherche un ensemble de candidats qui satisfont les contraintes

$\sum_{j\in\Omega'} \hat{A}^j \lambda_j = a.$

Cas d'une matrice bloc-diagonale

Considérons une matrice de contraintes à p blocs, avec $p \geq 1$:

$$\begin{pmatrix} A \\ \\ B \end{pmatrix} = \begin{pmatrix} A_1 & A_2 & \cdots & A_p \\ B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & B_p \end{pmatrix}$$

Soit alors le programme linéaire (P) réécrit sous la forme

$$\left\{ \begin{array}{lllllll} \min z = & c_1x_1 & c_2x_2 & \cdots & c_px_p & & \\ s.c. & A_1x_1 & A_2x_2 & \cdots & A_px_p & = & a \\ & B_1x_1 & & & & = & b_1 \\ & & B_2x_2 & & & = & b_2 \\ & & & \cdots & & = & \vdots \\ & & & & B_px_p & = & b_p \\ & x_1 & x_2 & \cdots & x_p & \geq & 0 \end{array} \right.$$

Le sous-problème s'écrit

$$\begin{cases} \min & \sum_{i=1}^{p}(c_i - \pi_i A_i)x_i \\ s.c. & B_i x_i = b_i, \quad i = 1, \dots, p \\ & x_i \geq 0, \quad i = 1, \dots, p \end{cases}$$

L'objectif étant séparable, on obtient p problèmes indépendants :

$$\begin{cases} \min & (c_i - \pi_i A_i)x_i \\ s.c. & B_i x_i = b_i \\ & x_i \geq 0 \end{cases}$$

$i = 1, \dots, p.$

L'approche décrite plus haut devient interéssante si le nombre de blocs indépendants est grand.