Dualité en programmation linéaire

François Schwarzentruber

18 janvier 2023

1 Certificat d'optimalité ?

Exemple 1 Considérons le programme linéaire suivant :

$$\text{maximiser } 3x_1 + 4x_2$$
$$\begin{cases} \frac{1}{2}x_1 + 2x_2 \leq 30 & (1) \\ 3x_1 + x_2 \leq 25 & (2) \\ x_1, x_2 \geq 0 \end{cases}$$

$$\begin{array}{ll} 6 \times (1): & 3x_1 + 4x_2 \leq 3x_1 + 12x_2 \leq 180 \\ 4 \times (2): & 3x_1 + 4x_2 \leq 12x_1 + 4x_2 \leq 100 \\ 2 \times (1) + 1 \times (2): & 3x_1 + 4x_2 \leq 4x_1 + 5x_2 \leq 85 \\ y_1 \times (1) + y_2 \times (2): & 3x_1 + 4x_2 \leq (\frac{1}{2}x_1 + 2x_2)y_1 + (3x_1 + x_2)y_2 = (\frac{1}{2}y_1 + 3y_2)x_1 + (2y_1 + y_2)x_2 \leq 30y_1 + 25y_2 \end{array}$$

$$\text{minimiser } 30y_1 + 25y_2$$
$$\begin{cases} \frac{1}{2}y_1 + 3y_2 \geq 3 \\ 2y_1 + y_2 \geq 4 \\ y_1, y_2 \geq 0 \end{cases}$$

Exemple 2 Considérons le programme linéaire suivant :

$$\text{maximiser } x_1 + 6x_2$$
$$\begin{cases} x_1 \leq 200 & (1) \\ x_2 \leq 300 & (2) \\ x_1 + x_2 \leq 400 & (3) \\ x_1 + x_2 \geq 0 \end{cases}$$

Question. $(x_1, x_2) = (100, 300)$ est une solution optimale avec un objectif de 1900. Me croyez-vous ?
Vous pouvez me croire que c'est une solution : on remplace et on vérifie que $(100, 300)$ vérifie les contraintes et que l'objectif vaut 1900. Mais comment être sûr que c'est optimal ? En fait, on peut des fois (on verra que... toujours !) en essayant de majorer l'objectif en sommant les contraintes.

Réponse.

- On a : (1) + 6(2) donne $x_1 + 6x_2 \leq 200 + 300 \times 6 = 2000$.
- Mieux : 0(1) + 5(2) + 1(3) donne $x_1 + 6x_2 \leq 5 \times 300 + 400 = 1900$. Cela me certifie que 1900 est l'optimum.

Idée de généralisation. S'occuper de minimiser $200y_1 + 300y_2 + 400y_3$ sous la contrainte

$$\underbrace{(y_1 + y_3)x_1 + (y_2 + y_3)x_2}_{\geq x_1 + 6x_2 ?} \leq x_1y_1 + x_2y_2 + (x_1 + x_2)y_3 \leq 200y_1 + 300y_2 + 400y_3$$

Quid de considérer

$$\text{minimiser } 200y_1 + 300y_2 + 400y_3$$
$$\begin{cases} y_1 + y_3 \geq 1 \\ y_2 + y_3 \geq 6 \\ y_1, y_2, y_3 \geq 0 \end{cases} \quad ?$$

2 Exemple d'une usine de meubles

On considère une usine de meubles.

	bureau	table	temps libre
menuiserie	1h	2h	20h
assemblage	2h	1h	22h
vernissage	1h	1h	12h
profit	300€	200€	

Le programme primal consiste à calculer combien va-t-on construire de bureaux et de tables. Soit x_1 le nombre de bureaux construits, et x_2 le nombre de tables construits. On souhaite maximiser le profit, mais en respectant les contraintes de temps libres.

$$\text{maximiser } 300x_1 + 200x_2$$
$$\begin{cases} x_1 + 2x_2 \leq 20 & (1) \\ 2x_1 + x_2 \leq 22 & (2) \\ x_1 + x_2 \leq 12 & (3) \\ x_1, x_2 \geq 0 \end{cases}$$

$$\begin{array}{ll} 300 \times (1): & 300x_1 + 200x_2 \leq 300x_1 + 400x_2 \leq 6000 \\ 200 \times (2): & 300x_1 + 200x_2 \leq 400x_1 + 200x_2 \leq 4400 \\ 300 \times (3) & 300x_1 + 200x_2 \leq 300x_1 + 300x_2 \leq 3600 \\ 100(2) + 100(3): & 300x_1 + 200x_2 \leq 300x_1 + 300x_2 \leq 3400 \\ y_1(1) + y_2(2) + y_3(3): & 300x_1 + 200x_2 \leq (y_1 + 2y_2 + y_3)x_1 + (2y_1 + y_2 + y_3)x_2 \leq 20y_1 + 22y_2 + 12y_3 \end{array}$$

On obtient donc :

$$\text{minimiser } 20y_1 + 22y_2 + 12y_3$$
$$\begin{cases} y_1 + 2y_2 + y_3 \geq 300 \\ 2y_1 + y_2 + y_3 \geq 200 \\ y_1, y_2, y_3 \geq 0 \end{cases}$$

Le programme dual s'explique aussi en changeant de point de vue. Supposons que l'on souhaite vendre à un client tous le temps libre. On note y_1 le prix d'1h de menuiserie, y_2 le prix d'1h d'assemblage, y_3 le prix d'1h de vernissage. Le client va vouloir minimiser le prix de vente qui est de $20y_1 + 22y_2 + 12y_3$. Mais nous, nous ne voulons pas être perdant par rapport à une utilisation directe des ressources. Nous souhaitons que le revenu gagné par exemple en vendant les ressources pour faire un bureau (à savoir $y_1 + 2y_2 + y_3$) soit plus grand que ce qu'on aurait gagné en vendant le bureau directement (300).

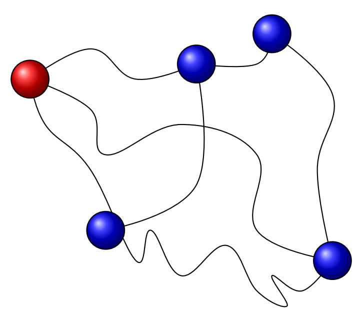
3 Motivation

1. Comprendre que le lien entre deux problèmes algorithmiques (par exemple : flot max et coupe min)
2. Construire des algorithmes spécialisés efficaces : transportation simplex method, l'algorithme hongrois (pour couplage pondéré dans un graphe biparti), network simplex method
3. Analyser comment un programme est sensible aux modifications (ajout d'une contrainte, etc.).
4. Parfois, trouver une solution faisable est plus facile dans le dual que dans le primal
5. Démontrer des résultats théoriques de manière élégante
6. Interprétation en économie etc.
7. Parfois le dual est plus simple à résoudre que le primal.

4 Problème dual

Définition 3 (programme dual) A tout programme linéaire, appelé *programme primal*, on associe un programme linéaire, appelé le *programme dual*, en utilisant cette recette de cuisine :

	Programme primal	Programme dual
Variables	$x_1, \ldots, x_n$	$y_1, \ldots, y_m$
Matrice	A	A^t
Vecteur de droite	b	c
Fonction objectif	**maximiser** $c^t x$	**minimiser** $b^t y$
Contraintes	la i-ème contrainte a $\leq$	$y_i \geq 0$
	la i-ème contrainte a $\geq$	$y_i \leq 0$
	la i-ème contrainte a $=$	$y_i \in \mathbb{R}$
	$x_j \geq 0$	j-ème contrainte a $\geq$
	$x_j \leq 0$	j-ème contrainte a $\leq$
	$x_j \in \mathbb{R}$	j-ème contrainte a $=$



Proposition 4 Le dual du dual de P est P.

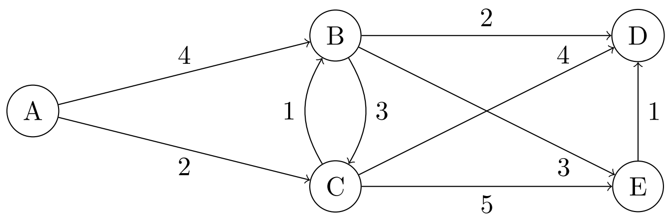
Exercice 1 Donner le programme dual de

$$\text{maximiser } 5x + 3y$$
$$\begin{cases} -5x + 2y \leq 0 \\ x + y \leq 7 \\ x \leq 5 \\ x, y \geq 0 \end{cases}$$

5 Plus court chemin : dualité et système physique

Définition 5 **Problème d'un plus court chemin d'une source à une destination**

entrée : graphe $G = (S, A, poids)$ pondéré positivement, deux sommets $s, t \in S$
sortie : le poids d'un plus court chemin depuis s vers t ; $+\infty$ sinon.



Proposition 6 Le problème d'un plus court chemin d'une source à une destination se réduit à la programmation linéaire réelle. Pour trouver un plus court chemin de s à t dans $G = (S, A, poids)$, on peut résoudre l'un des programmes suivants

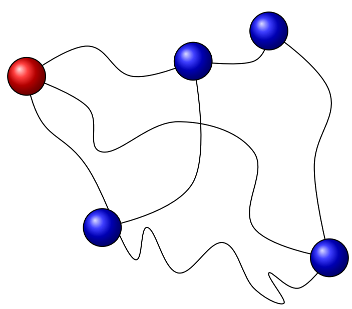
Programme primal

$$\text{minimiser } \sum_{e \in A} x_e poids(e)$$
$$\begin{cases} \sum_{e \text{ partant de } s} x_e - \sum_{e \text{ sortant de } s} x_e & = 1 \\ \sum_{e \text{ partant de } t} x_e - \sum_{e \text{ sortant de } t} x_e & = -1 \\ \sum_{e \text{ partant de } u} x_e - \sum_{e \text{ sortant de } u} x_e & = 0 \text{ pour tout sommet } u \notin \{s, t\} \\ x_e \geq 0 \text{ pour tout arc } e \end{cases}$$

Programme dual

$$\text{maximiser } d_t - d_s$$
$$\begin{cases} d_v - d_u \leqslant poids(u, v) \text{ pour tout arc } u \to v \\ d_u \in \mathbb{R} \text{ pour tout sommet } u \end{cases}$$

Démonstration. Ces programme sont duaux.On verra que le théorème de dualité forte s'y applique, mais on ne sait pas si les optimums sont atteints pour des solutions entières pour le programme primal. On verra la matrice de ces problèmes est totalement unimodulaire. Et donc les optimums sont atteints par des solutions entières. ■



6 Sac à dos fractionnaire

6.1 Aucune limite sur la quantité de matière

$c = (c_1, \ldots, c_n)$ où c_i = valeur d'une quantité de matière i
$a = (a_1, \ldots, a_n)$ où a_i = poids d'une quantité de matière i
b = poids maximal
On suppose $\frac{c_1}{a_1} \geq ... \geq \frac{c_n}{a_n}$.

Programme primal P
maximiser $c^t x$
$$\begin{cases} a^t x \leq b \\ x \geq 0 \end{cases}$$
$x = (x_1, \ldots, x_n)$ où x_i est la quantité de matière i prise

i.e

maximiser $c_1 x_1 + \cdots + c_n x_n$
$$\begin{cases} a_1 x_1 + \cdots + a_n x_n \leq b \\ x \geq 0 \end{cases}$$
Une solution du primal : $x_1 = \frac{b}{a_1}$, $x_2 = ... = x_n = 0$.

Programme dual D
minimiser by
$$\begin{cases} ay \geq c \\ y \geq 0 \end{cases}$$
y (€/ kg) = variation de la valeur maximale gagnée quand b augmente de 1kg

minimiser by
$$\begin{cases} a_1 y \geq c_1 \\ \vdots \\ a_n y \geq c_n \\ y \geq 0 \end{cases}$$
La solution du dual : $y = \frac{c_1}{a_1}$.

6.2 Limite sur la quantité de matière

Programme primal P
maximiser $c^t x$
$$\begin{cases} a^t x \leq b \\ x_1 \leq 1 \\ \vdots \\ x_n \leq 1 \\ x \geq 0 \end{cases}$$
Une solution optimale du primal est de la forme :
$x_1 = \ldots x_{k-1} = 1$
$x_k = \frac{b - \sum_{i=1}^{k-1} a_i}{a_k}$
$x_{k+1} = \cdots = x_n = 0$

Programme dual D
minimiser $by + z_1 \cdots + z_n$
$$\begin{cases} a_1 y + z_1 \geq c_1 \\ \vdots \\ a_n y + z_n \geq c_n \\ \\ y \geq 0, z \geq 0 \end{cases}$$

Une solution optimale du dual est :
$y = \frac{c_k}{a_k}$

$z_i = c_i - a_i \frac{c_k}{a_k}$ pour tout $i = 1..k-1$

$z_{k+1} = 0$
$\vdots$
$z_n = 0$

7 Théorèmes de dualité

Programme primal P
maximiser $c^t x$
$$\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

Programme dual D
minimiser $b^t y$
$$\begin{cases} A^t y \geq c \\ y \geq 0 \end{cases}$$

Théorème 7 (de dualité faible) Pour toute solution x de P, pour toute solution y de D, $c^t x \leq b^t y$.

Démonstration.
$$\begin{array}{lll} c^t x & = \sum_{j=1}^n c_j x_j & \\ & \leq \sum_{j=1}^n \left(\sum_{i=1}^m a_{ij} y_i\right) x_j & \text{par } A^t y \geq c \\ & = \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_j y_i & \\ & \leq \sum_{i=1}^m b_i y_i & \text{par } Ax \leq b \\ & = b^t y & \end{array}$$

■

Théorème 8 (de dualité forte) On est dans l'une des quatre situations suivantes :

1. P et D n'admettent pas de solutions ;
2. P est non borné et D n'admet pas de solutions ;
3. P n'admet pas de solutions et D est non borné ;
4. P possède une solution optimale x^*, D possède une solution optimale y^* et $c^t x^* = b^t y^*$.

Démonstration.

On se ramène à ces quatre cas via le théorème de dualité faible. Par exemple, le cas P non borné et D possède une solution optimale est impossible. Il ne reste plus qu'à montrer le quatrième cas. Supposons que P possède une solution optimale x^*, montrer que D possède une solution optimale y^* et que $c^t x^* = b^t y^*$.

Exemple 9 Considérons le programme primal suivant : $$\text{maximiser } 5x+3y \quad \begin{cases} -5x+2y \leq 0 \\ x+y \leq 7 \\ x \leq 5 \\ x, y \geq 0 \end{cases}$$

Considérons l'exécution de l'algorithme du simplexe sur le problème primal P. L'algorithme commence par mettre sous forme équationnelle :

$$\text{maximiser } \bar{c}^t \bar{x} \quad \begin{cases} \bar{A}\bar{x} = b \\ \bar{x} \geq 0 \end{cases} \qquad \text{où } \bar{c} = (c, 0, \ldots, 0),\ \bar{A} = (A\ Id_m).$$

Exemple 10 Voici le programme équationnel : $$\text{maximiser } 5x+3y \quad \begin{cases} -5x+2y+\alpha = 0 \\ x+y+\beta = 7 \\ x+\gamma = 5 \\ x, y, \alpha, \beta, \gamma \geq 0 \end{cases}$$

L'algorithme considère alors le tableau associé, puis exécute des pivotages. On considère une exécution qui termine, par exemple celle obtenue via règle de Bland. On atteint donc un tableau final de base B :

$$\text{maximiser } obj + c'^t \bar{x}_N \quad \begin{cases} \bar{x}_B = p - Q\bar{x}_N \\ \bar{x} \geq 0 \end{cases} \qquad \text{où les coordonnées du vecteur } c' \text{ sont négatives.}$$

Exemple 11 Le tableau final de base $B = \{x, y, \alpha\}$ est $$\text{maximiser } 31 - 2\gamma - 3\beta \quad \begin{cases} x = 5 - \gamma \\ y = 2 + \gamma - \beta \\ \alpha = 21 - 7\gamma + 2\beta \end{cases}$$

La solution basique $\bar{x}^*$ de ce dernier tableau est telle que les n premières coordonnées x^* de $\bar{x}^*$ forment une solution optimale du primal. On rappelle que $\bar{x}^*_B = p = \bar{A}_B^{-1} b$ et $\bar{x}^*_N = 0$; aussi $c' = \bar{c}_N - (\bar{c}^t_B \bar{A}_B^{-1} \bar{A}_N)^t$; (voir cours sur l'algorithme du simplexe).

Exemple 12 $x^* = (5, 2)$ et $\bar{x}^* = (5, 2, 21, 0, 0)$. $\bar{x}^*_B = (5, 2, 21)$.

$\boxed{\text{Posons } y^* = (\bar{c}^t_B \bar{A}_B^{-1})^t.}$

Exemple 13 Ici, $\bar{c}^t_B = (5, 3, 0)$. La matrice $\bar{A}_B = \begin{pmatrix} -5 & 2 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$. Son inverse est $\bar{A}_B^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -2 & 7 \end{pmatrix}$. On trouve $y^* = \begin{pmatrix} 0 \\ 3 \\ 2 \end{pmatrix}$.

Fait 14 $c^t x^* = b^t y^*$.

Démonstration. $c^t x^* = \bar{c}^t \bar{x}^* = \bar{c}^t_B \bar{x}^*_B + \bar{c}^t_N \underbrace{\bar{x}^*_N}_{0} = \bar{c}^t_B(\bar{A}_B^{-1} b) = (\bar{c}^t_B \bar{A}_B^{-1})b = (y^*)^t b = b^t y^*$. ■

Fait 15 y^* est une solution du dual (et donc est une solution minimale).

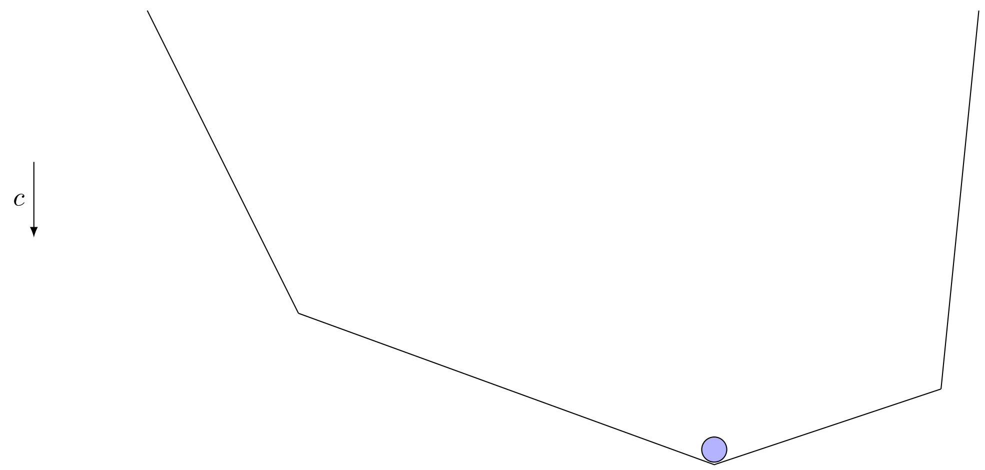
Démonstration. Il suffit de montrer les conditions de faisabilité $A^t y^* \geq c$ et $y^* \geq 0$, que l'on peut résumer par $\bar{A}^t y^* \geq \bar{c}$. Posons $w := \bar{A}^t y^* = \bar{A}^t(\bar{c}^t_B \bar{A}_B^{-1})^t = (\bar{c}^t_B \bar{A}_B^{-1} \bar{A})^t$. Regardons maintenant les coordonnées de w selon que c'est une coordonnée dans la base du tableau final ou non :
$w_B = (\bar{c}^t_B \bar{A}_B^{-1} \bar{A}_B)^t = \bar{c}_B$.
$w_N = (\bar{c}^t_B \bar{A}_B^{-1} \bar{A}_N)^t = \bar{c}_N - c' \geq \bar{c}_N$ car c' est un vecteur négatif. ■
■

8 Interprétation physique

Programme primal P
maximiser $c^t x$
$$\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

Programme dual D
minimiser $b^t y$
$$\begin{cases} A^t y \geq c \\ y \geq 0 \end{cases}$$

Afin d'être plus concret, considérons que x est de dimension 2 ou 3, et que x est la position d'une bille à l'intérieur du polyèdre défini par $Ax \leq b$. On suppose que c est le vecteur correspondant à la force de gravité.



Ainsi maximiser $c^t x$ revient à laisser agir la gravité. La bille tombe. Une solution optimale x^* est la position de la bille une fois tombée dans un creux (généralement, intersection de 3 faces, même si elle peut aussi rester en plein milieu d'une face si elle est horizontale).

Soit N les indices des faces que la bille touche. Soit $i \in N$. Comme la bille touche la face i, la variable d'écart correspondante est nulle. Dit autrement, voici la i-ème ligne-contrainte du programme primal : $(Ax^*)_i = b_i$ pour tout $i \in N$. La ligne $A_{i.}$ sont les coefficients de la face. Si on le considère comme un vecteur (colonne, autrement dit on transpose), ce vecteur est normal à la face. Autrement dit, le vecteur $A^t_{i.}$ est normal à la face i.

Faisons maintenant un peu de physique. La force de gravité F, proportionnelle à c. Elle se décompose sur les vecteurs normaux des faces touchées par la bille. De même pour c. Il existe donc y^* (la notation n'est pas hasardeuse, on verra que c'est un optimum du dual) telle que $c = \sum_{i=1}^m y^*_i (A^t_{i.})$ avec $y^*_i = 0$ pour $i \notin N$ (bah oui, une face non touchée ne participe pas à la force). Autrement dit, $c = A^t y^*$. Montrons que

$$c^t x^* = b^t y^*.$$

On a :

$$c^t x^* = (y^*)^t A x^* = \sum_{i=1}^m (y^*)_i (Ax^*)_i = \sum_{i \in N} (y^*)_i (Ax^*)_i + \sum_{i \notin N} \underbrace{(y^*)_i}_{0} (Ax^*)_i = \sum_{i \in N} (y^*)_i b_i + \sum_{i \notin N} \underbrace{(y^*)_i}_{0} b_i = b^t y^*.$$

9 Correspondance entre solutions primal/dual

Théorème 16 (coefficients magiques) Voici une solution optimale du problème dual :

exécuter l'algorithme du simplexe dans le problème primal
soit c' le vecteur des coefficients dans l'objectif du tableau final
y^* := opposés des coefficients dans c' des variables d'écart (coefficient nul si dans la base)

DÉMONSTRATION. Avant de lancer le simplexe, nous avons mis le problème primal sous forme équationnel, puis nous avons introduire une variable d'écart par contraintes. N'oublions pas qu'il y a une variable dans le problème dual pour chaque contrainte du problème primal. Donc il y a autant de variables d'écart que de variables dans le problème dual. Donc le typage dans $y^* := ...$ (dernière ligne de l'algo) est bon.

Considérons une variable d'écart d'indice i. La i-ème colonne de la matrice $\bar{A}$, la matrice dans le problème équationnel, que l'on note $\bar{A}_i$ est le i-ème vecteur unité, puisque la variable d'écart numéro i n'apparait que sur la contrainte numéro i avec un coefficient 1.

Considérons maintenant la base B du dernière tableau de l'exécution de l'algorithme du simplexe :

$$\text{maximiser } obj + c'^t \bar{x}_N \quad \begin{cases} \bar{x}_B = p - Q\bar{x}_N \\ \bar{x} \geq 0 \end{cases} \qquad \text{où les coordonnées du vecteur } c' \text{ sont négatives.}$$

Reprenons les notations de la démonstration du théorème de dualité forte : $w := \bar{A}^t y^* = \bar{A}^t(\bar{c}_B^t \bar{A}_B^{-1})^t = (\bar{c}_B^t \bar{A}_B^{-1} \bar{A})^t$. Comme $w := \bar{A}^t y^*$, on a $w_i = \bar{A}_i^t y^* = y_i^*$.

Comme i est une variable d'écart, $\bar{c}_i = 0$ (les variables d'écart n'apparaîssent pas dans l'objectif de départ). Ainsi, si on lit les équations de w_B et w_N donné en fin de la démonstration de la dualité on a :

- Si la variable d'écart i est dans B, alors $w_i = \bar{c}_i = 0$;
- Si une variable d'écart i est dans N, alors $w_i = 0 - c'_i = -c'_i$.

Conclusion :

- Si la variable d'écart i est dans B, alors $y_i^* = 0$;
- Si une variable d'écart i est dans N, alors $y_i^* = -c'_i$.

■

Théorème 17 (des écarts complémentaires) Soit x une solution du primal P et y une solution du dual D.

x et y sont solutions optimales de respectivement P et D	ssi	1. pour tout indice i de contraintes de P, $y_i \times (b_i - \sum_j a_{ij}x_j) = 0$; 2. et pour tout indice j de contraintes de D, $x_j \times (\sum_i a_{ij}y_i - c_j) = 0$.

DÉMONSTRATION.

$$\begin{gathered} x \text{ et } y \text{ solutions optimales de leurs programmes respectifs} \\ \Updownarrow \\ c^t x = b^t y \text{ (dualité forte)} \\ \Updownarrow \\ c^t x = y^t A x = b^t y \text{ (voir démo du th. de dualité faible)} \\ \Updownarrow \\ (c^t - y^t A)^t x = 0 \text{ et } y^t(b - Ax) = 0 \text{ (réécriture algébrique)} \\ \Updownarrow \\ \text{points 1 et 2} \end{gathered}$$

(car les coordonnées des vecteurs sont positives et $\sum$ nombres positifs = 0 implique que ces nombres = 0)

■

10 Matrices totalement unimodulaires

Cas où un programme linéaire réel à coefficients entiers admet une solution optimale entière

Définition 18 (matrice totalement unimodulaire) Une matrice est totalement unimodulaire si toute sous-matrice (en supprimant quelques lignes et/ou colonnes) carrée est de déterminant -1, 0 ou 1.

Proposition 19 Une matrice totalement unimodulaire ne contient que des -1, 0 ou 1.

Exemple 1 $\begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$ est totalement unimodulaire.

Exemple 2 $\begin{pmatrix} 42 & -1 \\ -1 & 0 \end{pmatrix}$ et $\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ ne sont pas totalement unimodulaires.

Proposition 20 Soit A une matrice totalement unimodulaire. La matrice $\bar{A}$ obtenue en ajoutant un vecteur unité e_i en dernière colonne est aussi totalement unimodulaire.

Démonstration. Calculer le déterminant d'une sous-matrice carrée en faisant un développement de Laplace selon la dernière colonne. ■

Exemple 3 $\begin{pmatrix} 1 & -1 & 0 & 1 \\ -1 & 1 & 1 & 0 \end{pmatrix}$ est totalement unimodulaire.

Théorème 21 Considérons un programme linéaire $\begin{array}{l} \text{maximiser } c^t x \\ \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{array}$ où $b \in \mathbb{Z}^m$ et A totalement unimodulaire. Si le programme admet une solution optimale, alors il admet une solution optimale entière $x^* \in \mathbb{Z}^n$.

Exemple 4 Si $\begin{array}{l} \text{maximiser } x + 3y - 5z \\ \begin{cases} x - y & \leq 30 \\ -x + y + z & \leq 42 \\ x, y, z \geq 0 \end{cases} \end{array}$ est borné, alors il admet une solution maximale entière.

Démonstration. Étudions l'exécution de l'algorithme du simplexe. Il travaille d'abord sur un programme équationnel où les contraintes sont $\bar{A}\bar{x} = b$ avec $\bar{A} = (A \mid Id_m)$ et $\bar{x} \in \mathbb{R}^{n+m}$. L'algorithme du simplexe trouve une base $B \in \{1, \dots, n+m\}$ telle que la solution basique associée $\bar{x}^*$ soit définie par :

$$\begin{aligned} \bar{x}^*_B &= \bar{A}_B^{-1} b \\ \bar{x}^*_N &= 0 \end{aligned}$$

Les coefficients de $\bar{A}_B^{-1}$ s'écrivent, via les **règles de Cramer**, comme des fractions d'entiers avec au dénominateur $det(\bar{A}_B)$. Comme A est totalement unimodulaire, $\bar{A}$ l'est aussi par la proposition 20. Ainsi, $det(\bar{A}_B) \in \{-1, 0, 1\}$ (la valeur 0 est impossible car la matrice est inversible). ■

Proposition 22 Soit A une matrice contenant seulement les éléments 0, 1 ou -1 et qui satisfait les 2 conditions suivantes :

1. Chaque colonne contient au plus 2 éléments non nuls ;
2. Les lignes de A peuvent être partitionnées en 2 sous-ensembles S_1 et S_2 tel que pour chaque colonne contenant 2 éléments non nuls :
   - Si les 2 éléments non nuls ont le même signe, alors l'un est dans S_1 et l'autre dans S_2 ;
   - Si les 2 éléments non nuls sont de signe différents, alors ils sont tous deux dans S_1, ou tous deux dans S_2.

Alors A est totalement unimodulaire.

Démonstration. On pose

$$\mathcal{P}(\ell) : \text{le déterminant de toute sous-matrice de taille } \ell \times \ell \text{ de } A \text{ est dans } \{-1, 0, 1\}.$$

que l'on va démontrer par récurrence. Le cas de base $\mathcal{P}(1)$ est ok par définition de A. Supposons que $\mathcal{P}(\ell - 1)$. Montrons $\mathcal{P}(\ell)$. Considérons une sous-matrice Q de taille $\ell \times \ell$. Par la condition 1, chaque colonne de A contient au plus deux éléments non nuls. Ainsi, il en est de même pour chaque colonne de Q. S'il y a une colonne avec que des 0, $\det(Q) = 0$. S'il y a une colonne avec un seul élément non nul, on se ramène, à signe près, au calcul du déterminant d'une sous-matrice de A de taille $(\ell - 1) \times (\ell - 1)$, qui est par récurrence de déterminant dans $\{-1, 0, 1\}$. Enfin, toutes les colonnes de Q contiennent deux éléments non nuls, alors on montre que $det(Q) = 0$. En effet, les lignes sont dépendantes.

1. En sommant toutes les lignes d'indices S_1, on obtient un certain vecteur ;
2. Et en sommant les lignes d'indices S_2 on obtient le vecteur opposé, vu la condition 2.

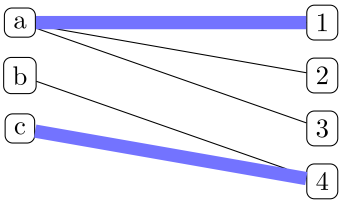
■

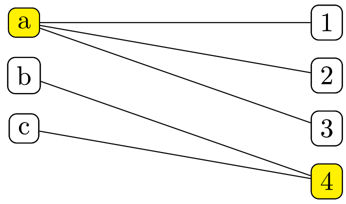
11 Théorème de König

TODO: mettre la motivation SWERC

Théorème 23 Dans un graphe biparti,

$$\text{cardinal d'un couplage maximal} \quad = \quad \text{cardinal d'une couverture minimal de sommets.}$$





DÉMONSTRATION. Quitte à renommer les objets, on note $\{1, \ldots, n\}$ l'ensemble des sommets, et $\{1, \ldots, m\}$ l'ensemble des arêtes.

1) Programme linéaire pour le couplage maximal

$$\text{maximiser } \sum_{j=1}^{m} x_j$$
$$\begin{cases} \sum_{j \text{ incident à } i} x_j \leq 1 \text{ pour tout sommet } i = 1..n \\ x \geq 0 \\ x \in \mathbb{Z}^m \end{cases}$$

En posant $A \in \mathbb{R}^{n \times m}$ par $a_{ij} = \begin{cases} 1 & \text{si l'arc } j \text{ incidente au sommet } i \\ 0 & \text{sinon.} \end{cases}$, ce programme se réécrit en :

$$\text{maximiser } \sum_{j=1}^{m} x_j$$
$$\begin{cases} Ax \leq 1 \\ x \geq 0 \\ x \in \mathbb{Z}^m \end{cases}$$

2) Programme linéaire pour la couverture minimal

$$\text{minimiser } \sum_{i=1}^{n} y_i$$
$$\begin{cases} \sum_{j \text{ incident à } i} y_i \geq 1 \text{ pour toute arête } j = 1..m \\ y \geq 0 \\ y \in \mathbb{Z}^n \end{cases}$$

que l'on peut réécrire en

$$\text{minimiser } \sum_{i=1}^{n} y_i$$
$$\begin{cases} A^t y \geq 1 \\ y \geq 0 \\ y \in \mathbb{Z}^n \end{cases}$$

Si on avait "$\in \mathbb{R}^m$" et "$\in \mathbb{R}^n$" à la place "$\in \mathbb{Z}^m$" et "$\in \mathbb{Z}^n$", les programmes seraient duaux l'un de l'autre et cela conclurait la démonstration. Ici, on ne peut pas conclure car il n'y a pas de théorème de dualité pour la programmation linéaire entière. Mais heureusement, la proposition 22 conclut la démonstration ! En effet :

1. Chaque colonne (arête) est incidente à deux sommets ;
2. Les paquets S_1 et S_2 sont pile poil les deux patates de sommets du graphe biparti.

■

Exercice 24 Montrer que la matrice d'incidence signée du problème du plus court chemin (et aussi du problème de flots) est totalement unimodulaire. On notera qu'il n'y a pas besoin ici que le graphe soit biparti.

Notes bibliographiques

La dualité est évoquée dans [DPV08] et [CLRS09]. L'explication avec le certificat provient de [DPV08]. Les démonstrations et les applications proviennent essentiellement de [GM07].

Références

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009.

[DPV08] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani. *Algorithms.* McGraw-Hill, 2008.

[GM07] Bernd Gärtner and Jirí Matousek. *Understanding and using linear programming.* Universitext. Springer, 2007.