Chapitre 8
La programmation linéaire

Claude-Guy Quimper

Introduction

- Dans ce chapitre, nous allons étudier ce qu'est un programme linéaire.
- Ce problème d'optimisation peut encoder un grand nombre de problèmes industriels.
- Nous allons étudier la méthode du simplexe qui permet de résoudre ce problème d'optimisation.
- Nous allons étudier le théorème de dualité qui permet de mieux saisir la structure des problèmes d'optimisation.
- Nous allons terminer en montrant comment on peut résoudre des programmes à nombres entiers.

Programme linéaire

Définitions de ***programme***, nom masculin (aussi verbe)

- Liste d'émissions, de films à venir.
- Déroulement prévu; ensemble d'activités prévues pour une période déterminée. Voici le programme pour la prochaine semaine.
  - Document décrivant le déroulement prévu d'un spectacle, d'une cérémonie, d'une fête, etc.
- Ligne d'action ou suite d'actions proposées en vue d'un résultat. Programme électoral. Programme politique.
- Ensemble des connaissances et des matières prévues dans une classe ou sur lesquelles doit porter un examen. Programme scolaire. Ce n'est pas inscrit au programme.
- [INFORMATIQUE] Ensemble ordonné d'instructions codées dans un langage donné et décrivant les étapes menant à la solution d'un problème. Écrire un programme pour accomplir une tâche.

Flexion

m. s.	programme
m. pl.	programmes

Étymologie

IXe — XVIIe — XXIe s.

Du grec *programma*, 'ordre du jour'.

À proscrire

Anglicisme : faux ami dans le sens de « émission ».

Indice de fréquence : 67

rare — fréquent

Autre ressource

W Wikipédia

Source: Antidote HD

Programme linéaire

- À l'origine, un programme était une suite d'actions qu'il fallait planifier. Puisque ces actions étaient sujettes à des contraintes linéaires, le terme « programme linéaire » a été adopté.
- Nous allons définir plus formellement ce qu'est un programme linéaire.

Programme linéaire

- Entrées: un vecteur c, un vecteur b et une matrice A.
- Sortie: un vecteur x.

$$
\begin{array}{lcccccccc}
\text{maximiser} & c_1x_1 & + & c_2x_2 & +\ldots+ & c_nx_n & & \\
\\
\text{sujet à} & a_{11}x_1 & + & a_{12}x_2 & +\ldots+ & a_{1n}x_n & \leq & b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & +\ldots+ & a_{2n}x_n & \leq & b_2 \\
& & & \vdots & & & & \\
& a_{m1}x_1 & + & a_{m2}x_2 & +\ldots+ & a_{mn}x_n & \leq & b_m \\
& & & & & x_i & \geq & 0
\end{array}
$$

Forme normale

- Un programme linéaire peut être écrit dans sa forme normale de la façon suivante.

$$\begin{array}{rrcl} \text{maximiser} & c^T x & & \\ \text{sujet à} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

- L'expression x ≥ 0 est vraie si toutes les composantes du vecteur x sont non négatives.

- De même, l'expression Ax ≤ b est vraie si le produit scalaire entre la $i^{\text{ème}}$ ligne de la matrice A et le vecteur x est plus petit ou égal à b_i.

Équivalences de problèmes

- Un des défis de la programmation linéaire est de modéliser un problème et de l'écrire sous sa forme normale.
- En effet, les problèmes n'apparaissent pas tous naturellement sous cette forme. C'est la tâche de la personne qui modélise d'exprimer le problème dans ce gabarit pouvant parfois sembler très rigide.
- Heureusement, il existe des manipulations algébriques qui permettent de transformer des problèmes à la forme normale.

Minimisation vs Maximisation

$$\text{minimiser} \quad c^T x$$
$$\iff$$
$$\text{maximiser} \quad (-c^T)x$$

- Minimiser une expression mathématique est équivalent à maximiser sa négation.
- Il suffit donc d'inverser le signe des composantes du vecteur c pour passer d'un problème de minimisation à un problème de maximisation.

Plus grand vs plus petit

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq k$$
$$\iff$$
$$-a_1x_1 - a_2x_2 - \ldots - a_nx_n \leq -k$$

- Pour changer l'opérateur « plus grand que » en opérateur « plus petit que » il suffit de multiplier par -1 des deux côtés de l'inégalité.

Équation vs inégalité

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = k$$

$$\iff$$

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq k$$

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq k$$

- Une équation linéaire peut être exprimée à l'aide deux inégalités.
- Si une quantité n'est ni plus grande ni plus petite qu'une constante, alors cette quantité est égale à cette constante.

Variables libres vs variables contraintes

- Les variables d'un programme linéaire doivent être non négatives.
- Qu'arrive-t-il si, dans un problème, nous avons des variables libres, c'est-à-dire des variables pouvant être positives, nulles ou négatives?
- Une variable libre x_i peut être substituée par la différence de deux variables non négatives.

$$x_i = x_i^+ - x_i^-$$
$$x_i^+ \geq 0$$
$$x_i^- \geq 0$$

Variables libres vs variables contraintes

- Ces deux systèmes de contraintes sont équivalents.

$$\left.\begin{array}{rl} 3x + 2y + 4z & \leq 10 \\ x, y & \geq 0 \end{array}\right\} \iff \left\{\begin{array}{rl} 3x + 2y + 4z^+ - 4z^- & \leq 10 \\ x, y, z^+, z^- & \geq 0 \end{array}\right.$$

Exemple de transformation

$$
\begin{array}{lccccccc}
\text{minimiser} & 2x & + & 3y & - & 4z & & \\
\\
\text{sujet à} & 4x & + & 2y & & & \geq & -2 \\
 & & & 3y & + & 2z & = & 4 \\
 & & & & & x, y & \geq & 0
\end{array}
$$

$$
\begin{array}{lccccc}
\text{maximiser} & -2x & -3y & +4z^{+} & -4z^{-} & & \\
\\
\text{sujet à} & -4x & -2y & & & \leq & 2 \\
 & & 3y & +2z^{+} & -2z^{-} & \leq & 4 \\
 & & -3y & -2z^{+} & +2z^{-} & \leq & -4 \\
 & & \multicolumn{3}{c}{x, y, z^{+}, z^{-}} & \geq & 0
\end{array}
$$

Polytope convexe

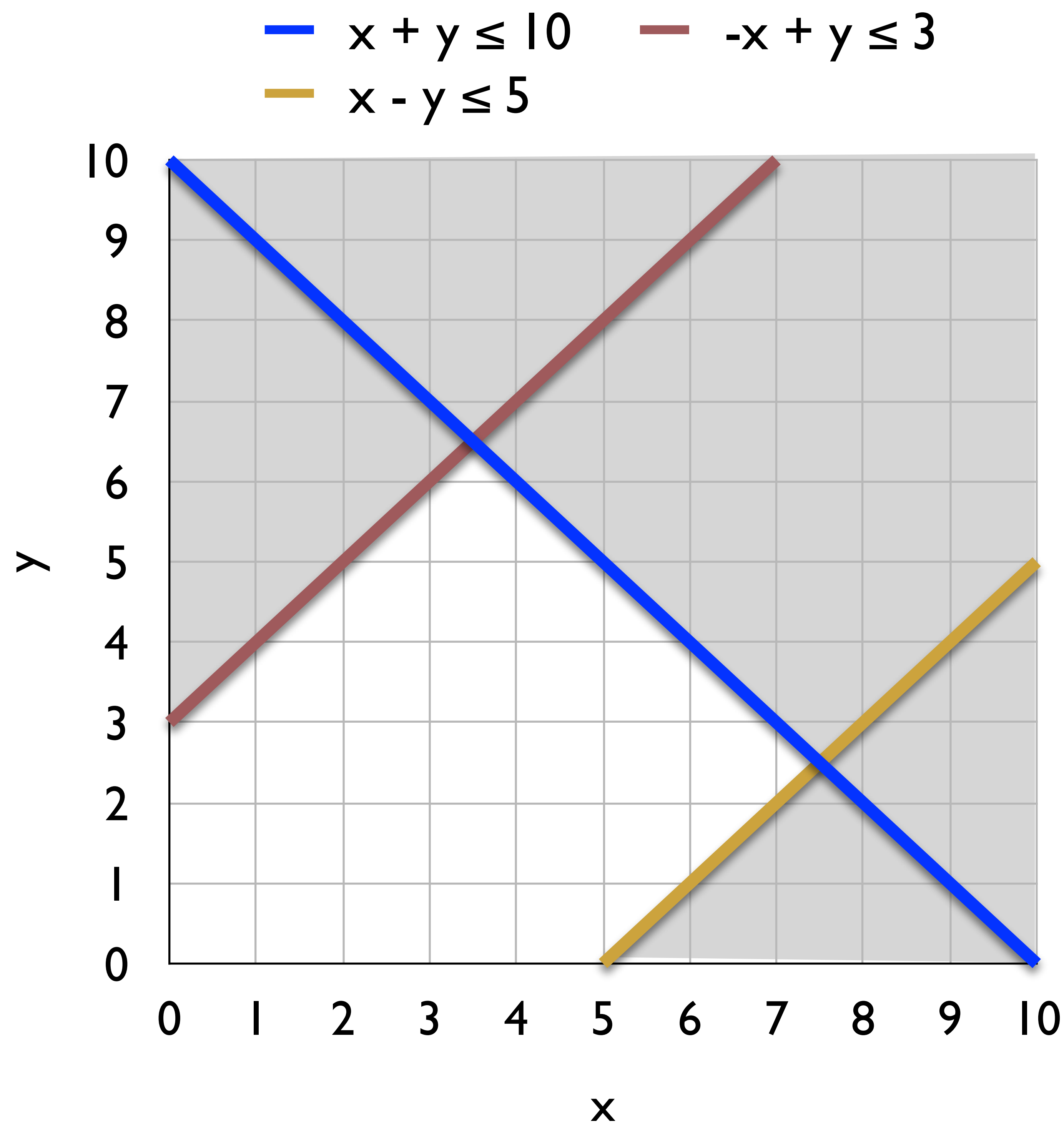
- Un polytope convexe est une forme géométrique convexe dont les bornes sont définies par des hyperplans.
- En 2 dimensions, un polytope convexe est un polygone convexe.
- En 3 dimensions, un polytope convexe est un polyèdre convexe.
- L'espace des solutions d'un système d'inégalités linéaires forme un polytope convexe.

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

Polytope

- Ce système d'inégalités forme le polytope en blanc.

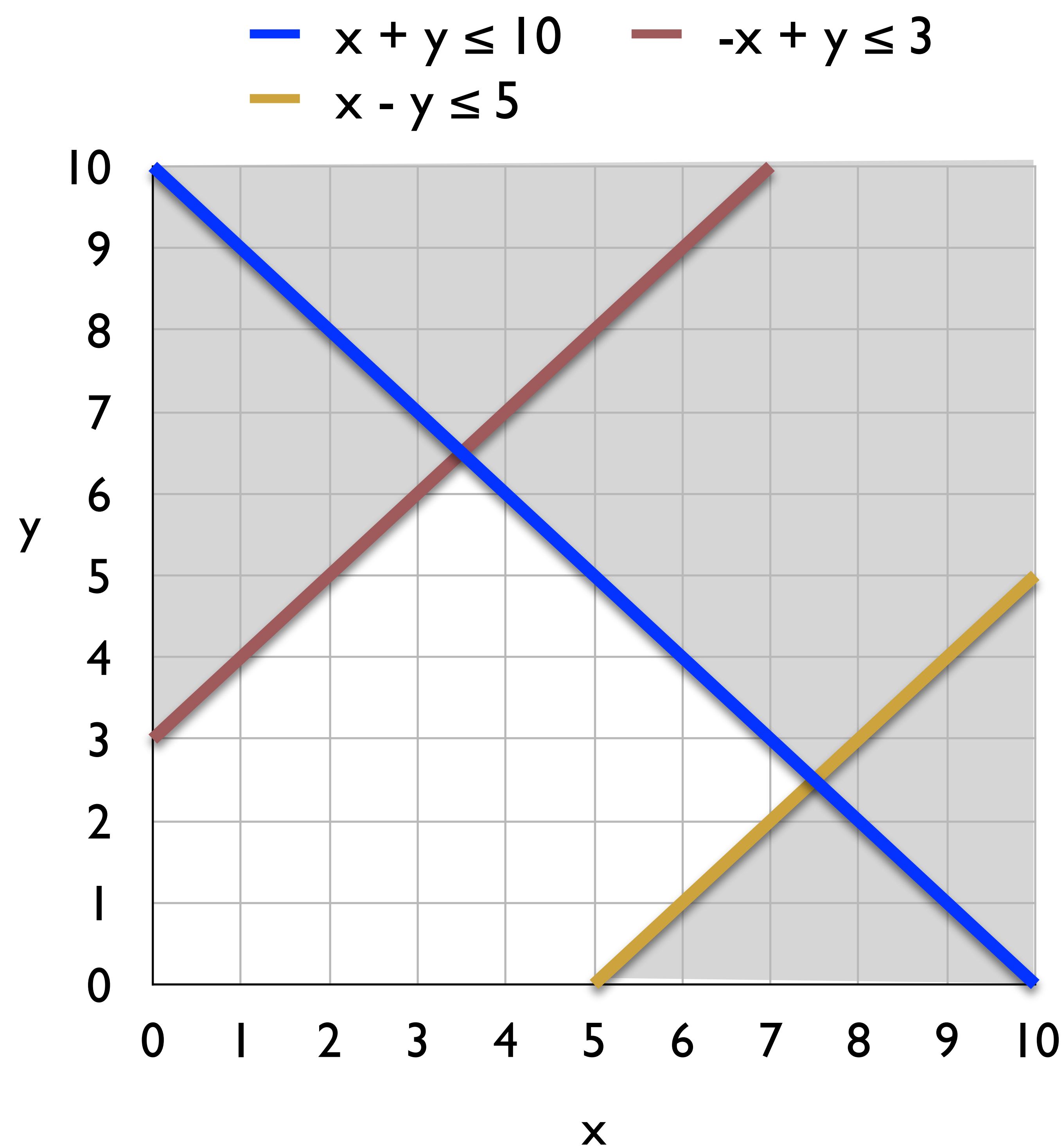
$$
\begin{array}{rcrcr}
x & + & y & \leq & 10 \\
-x & + & y & \leq & 3 \\
x & - & y & \leq & 5 \\
x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

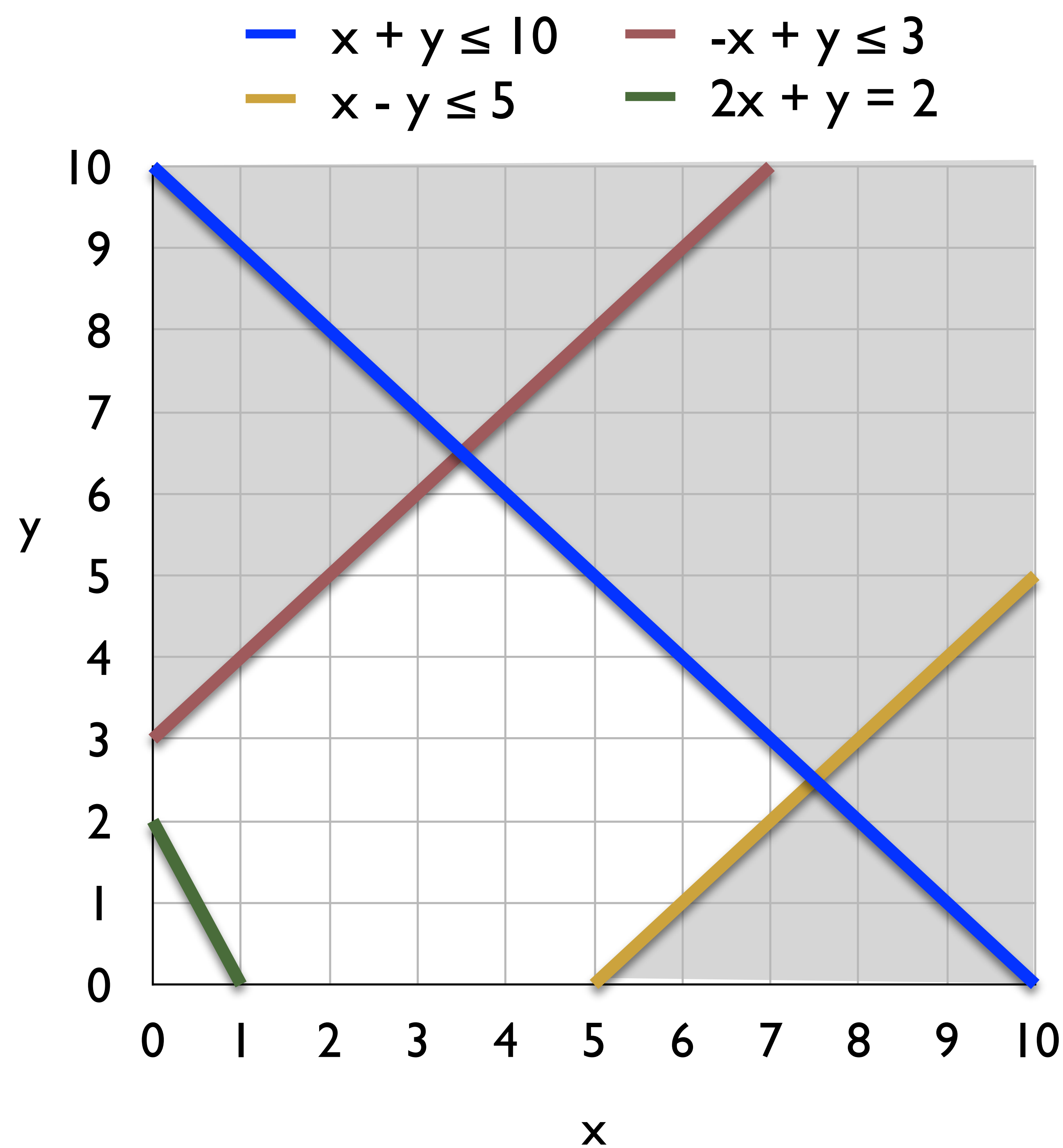
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

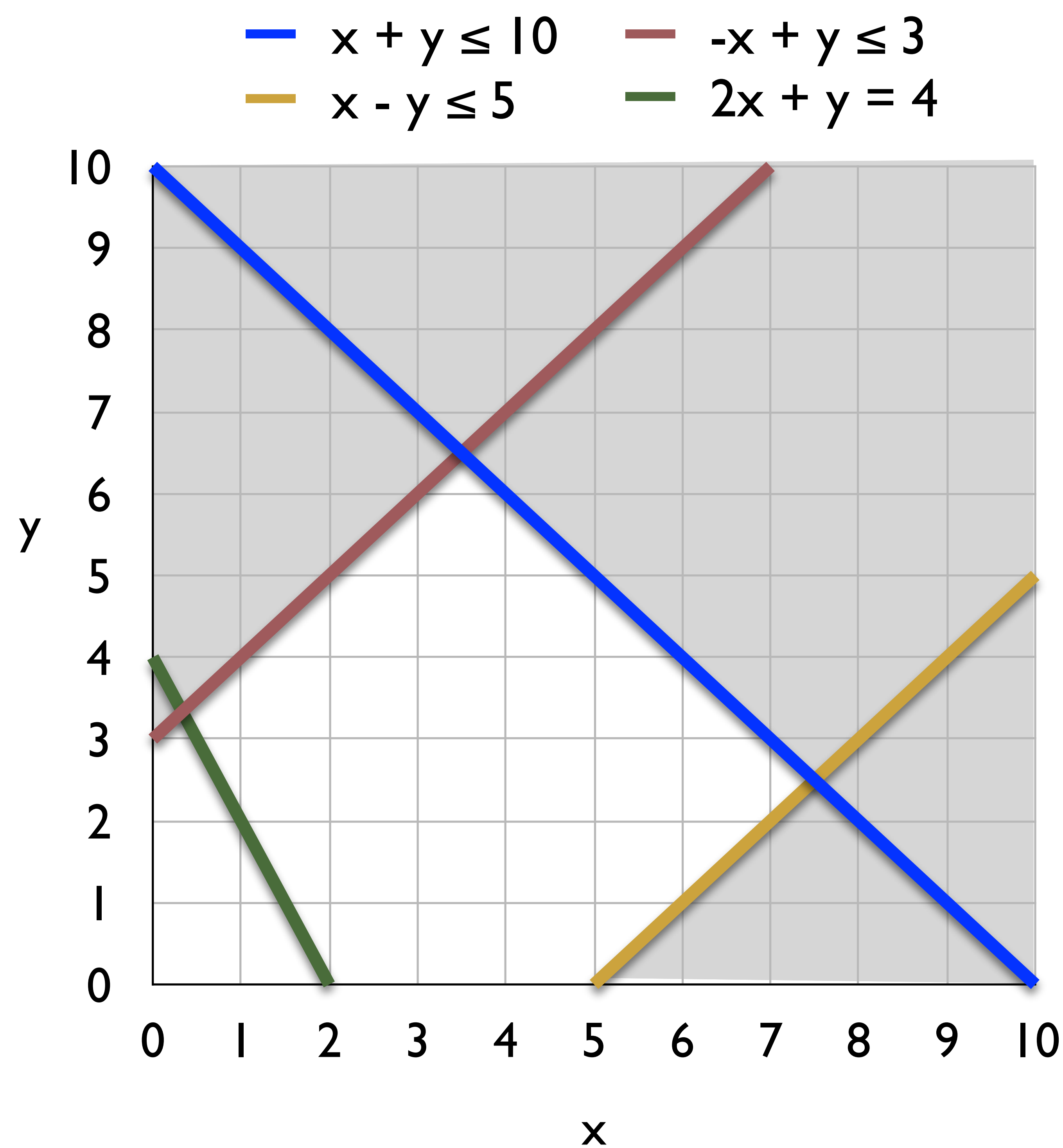
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

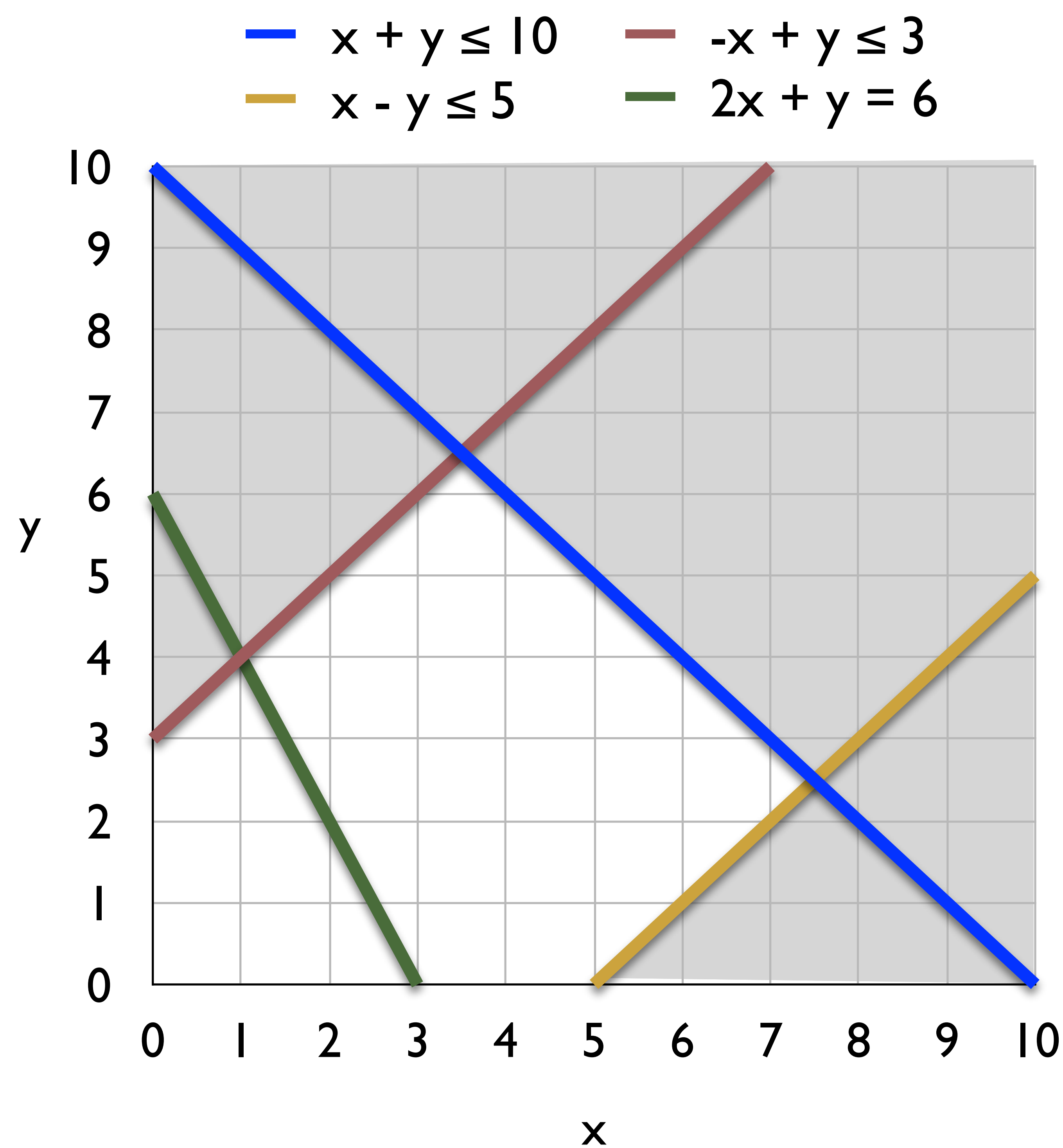
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

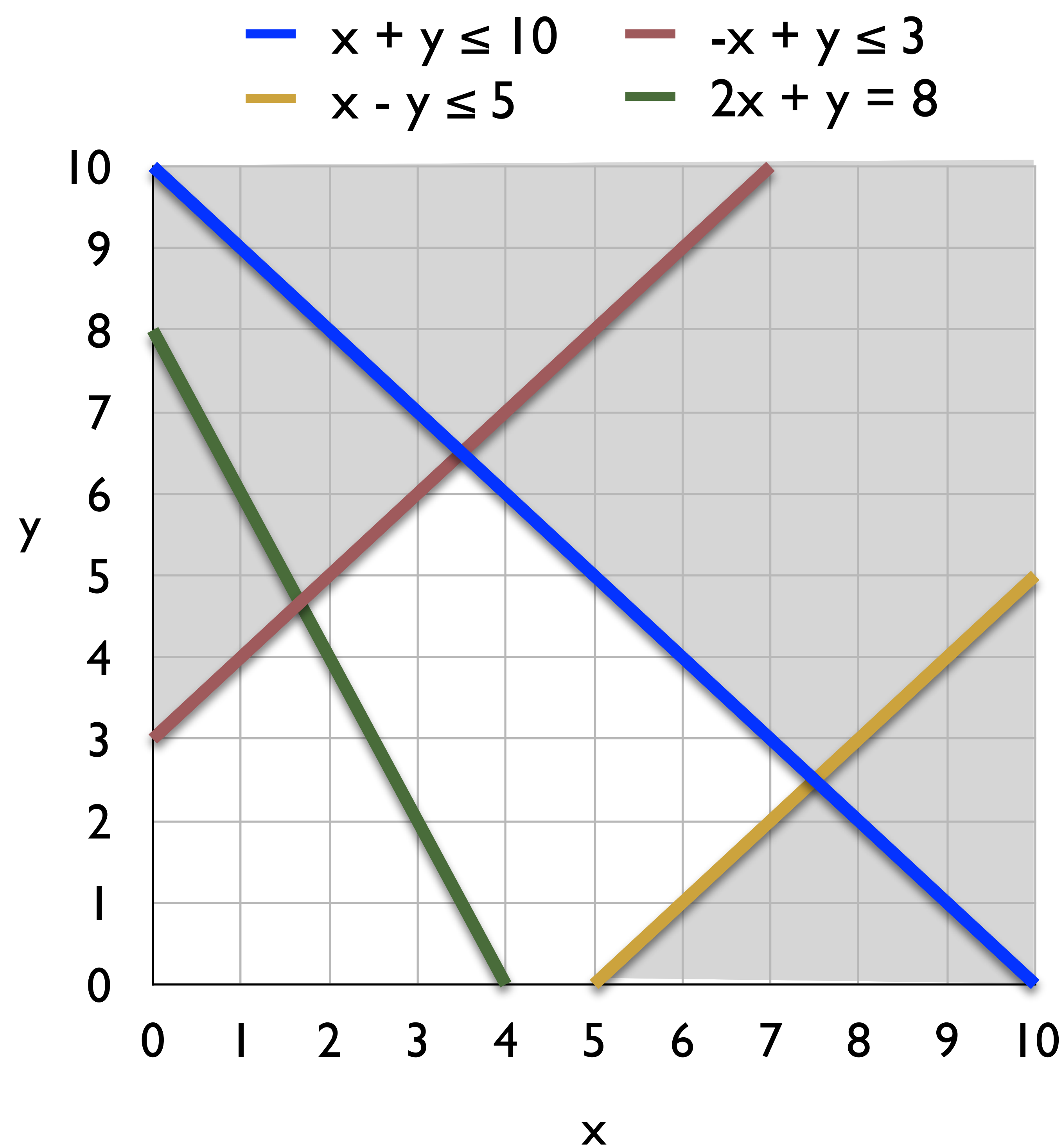
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

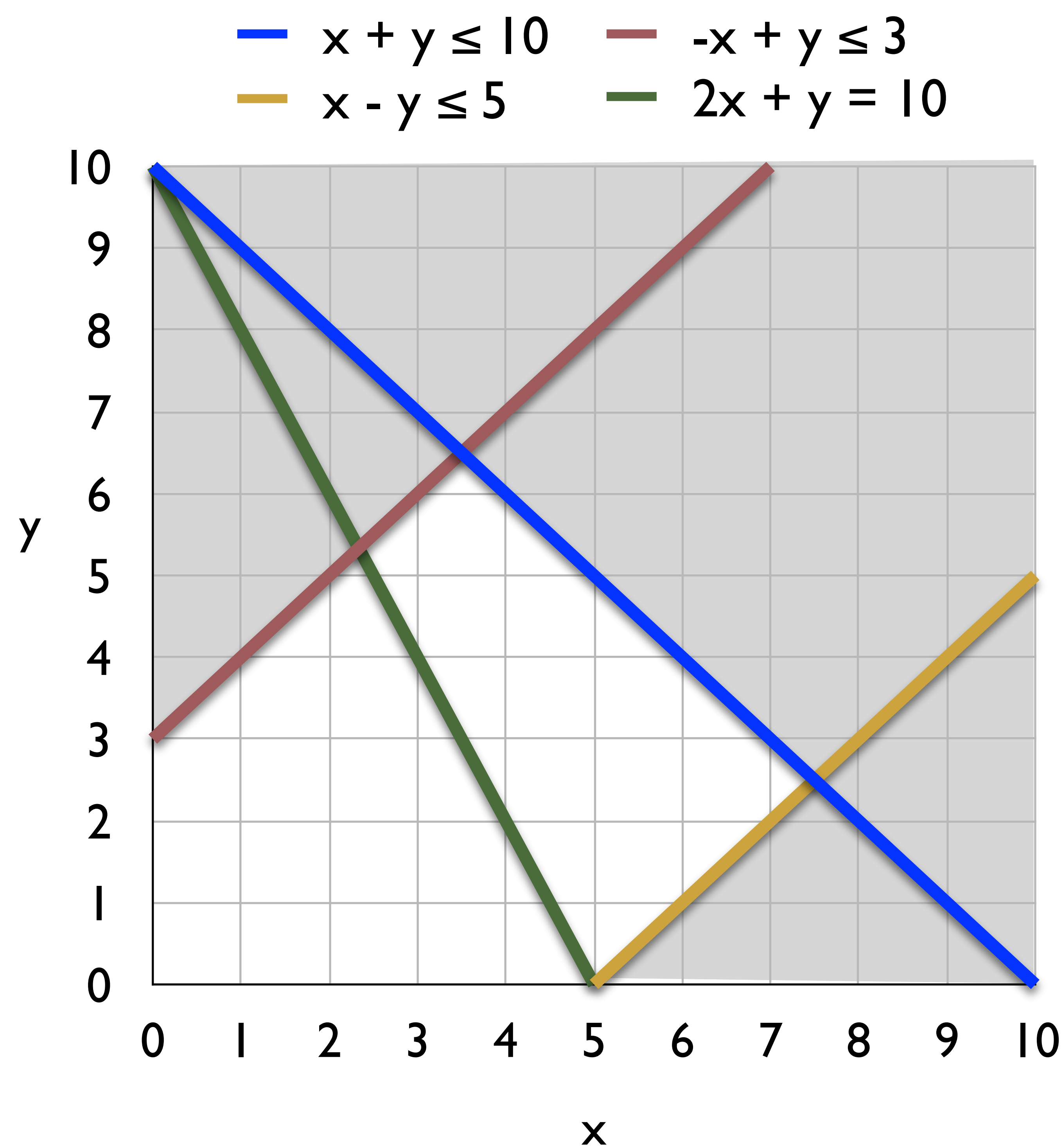
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

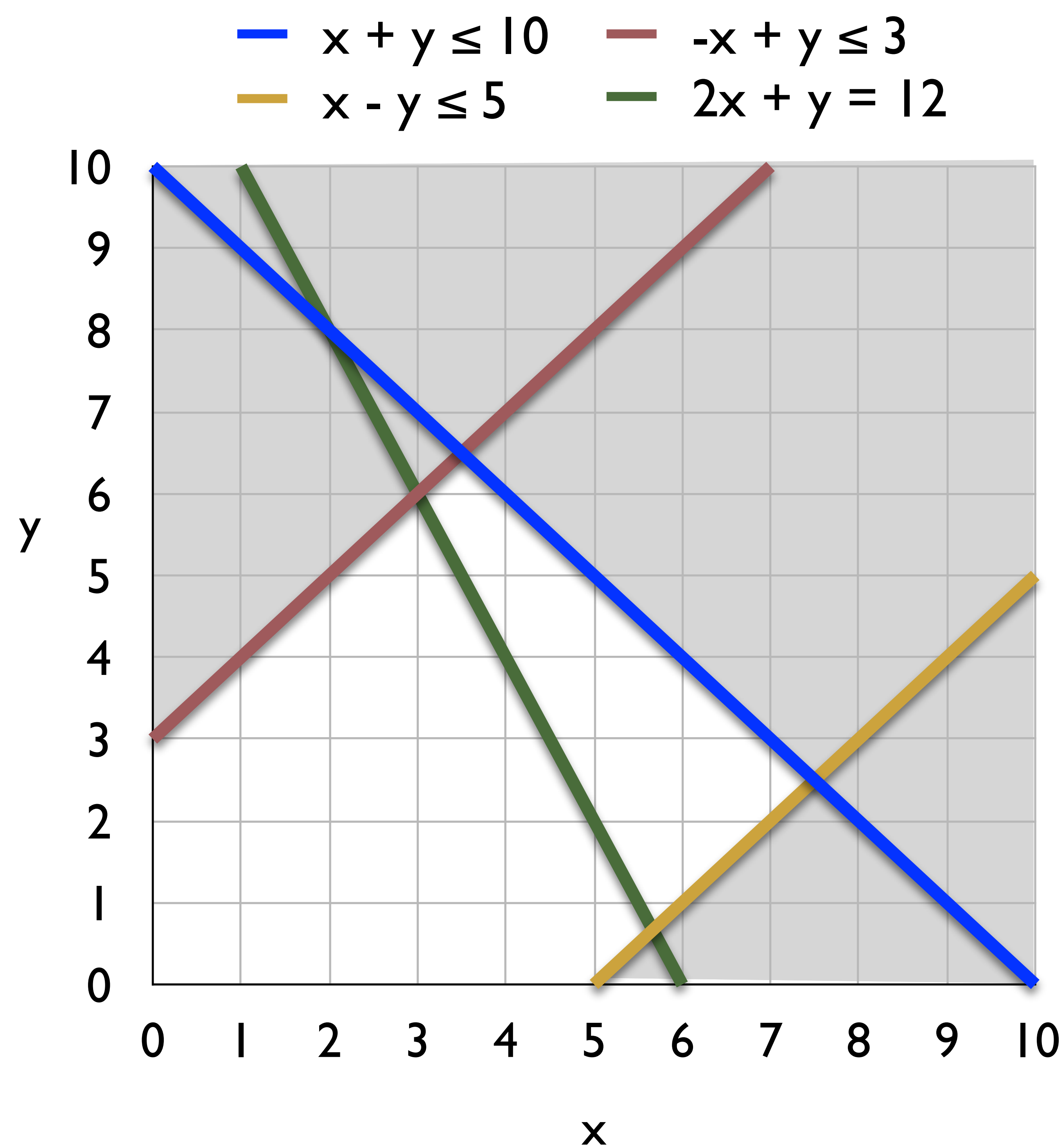
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

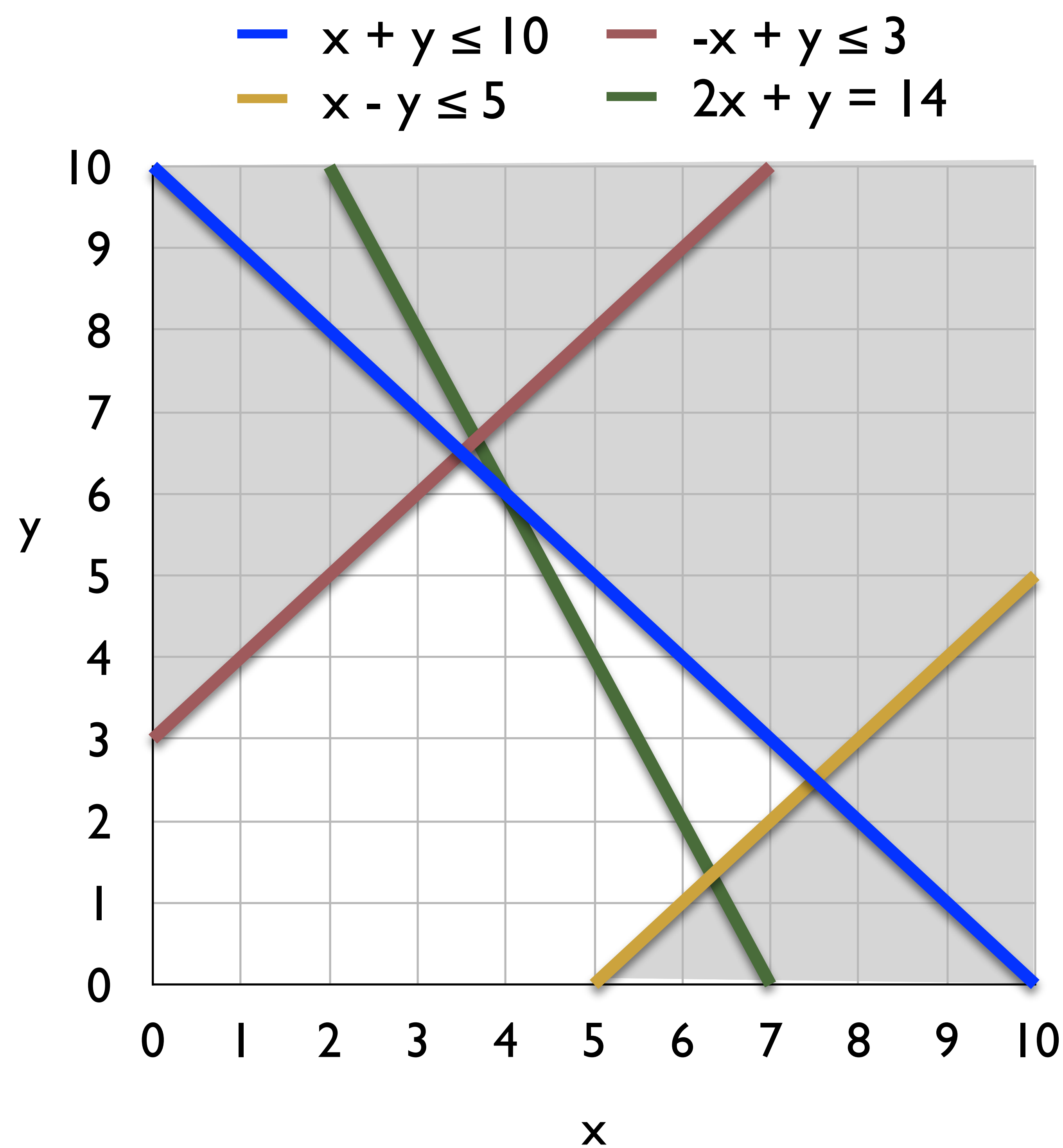
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

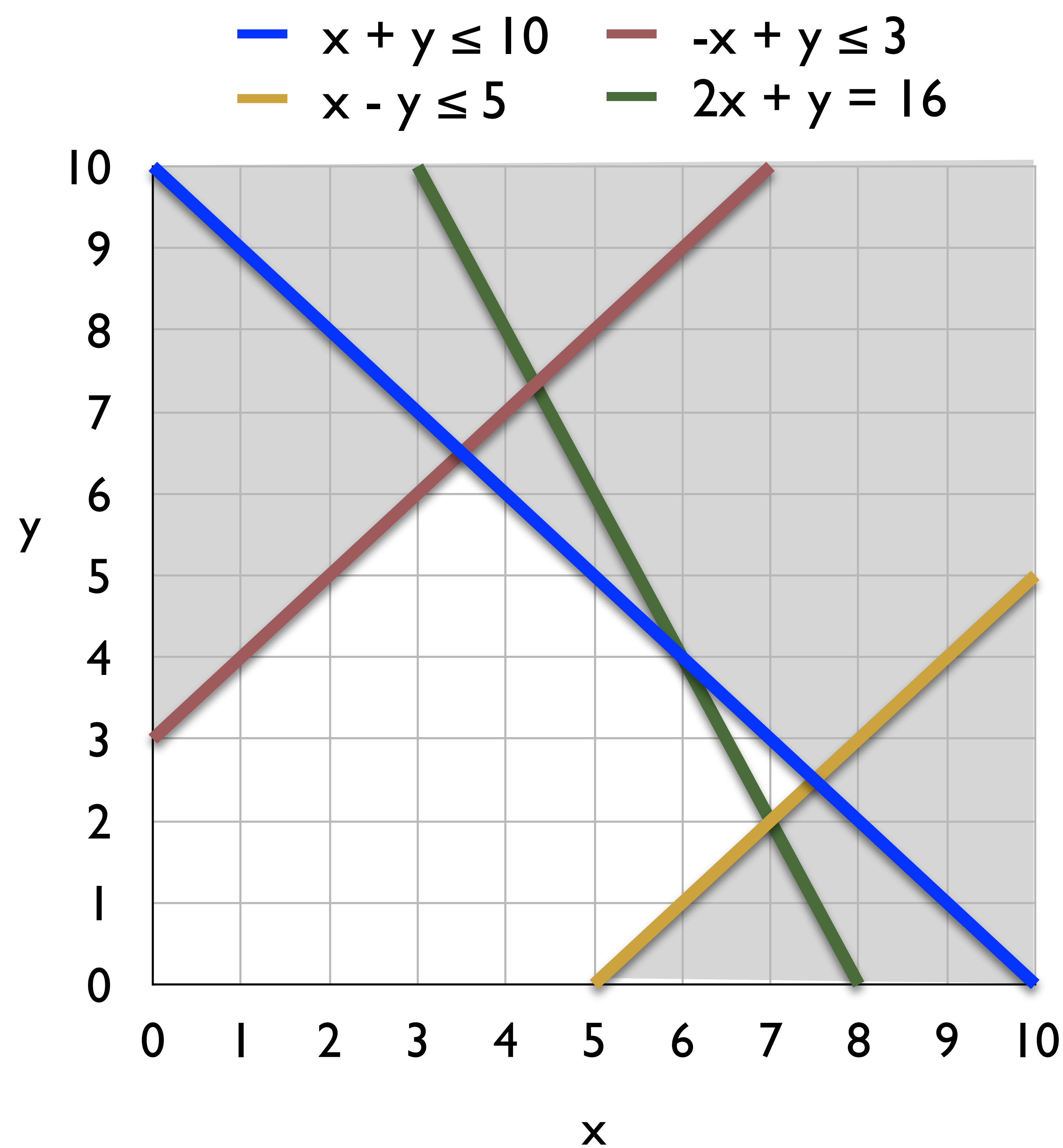
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^T x = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

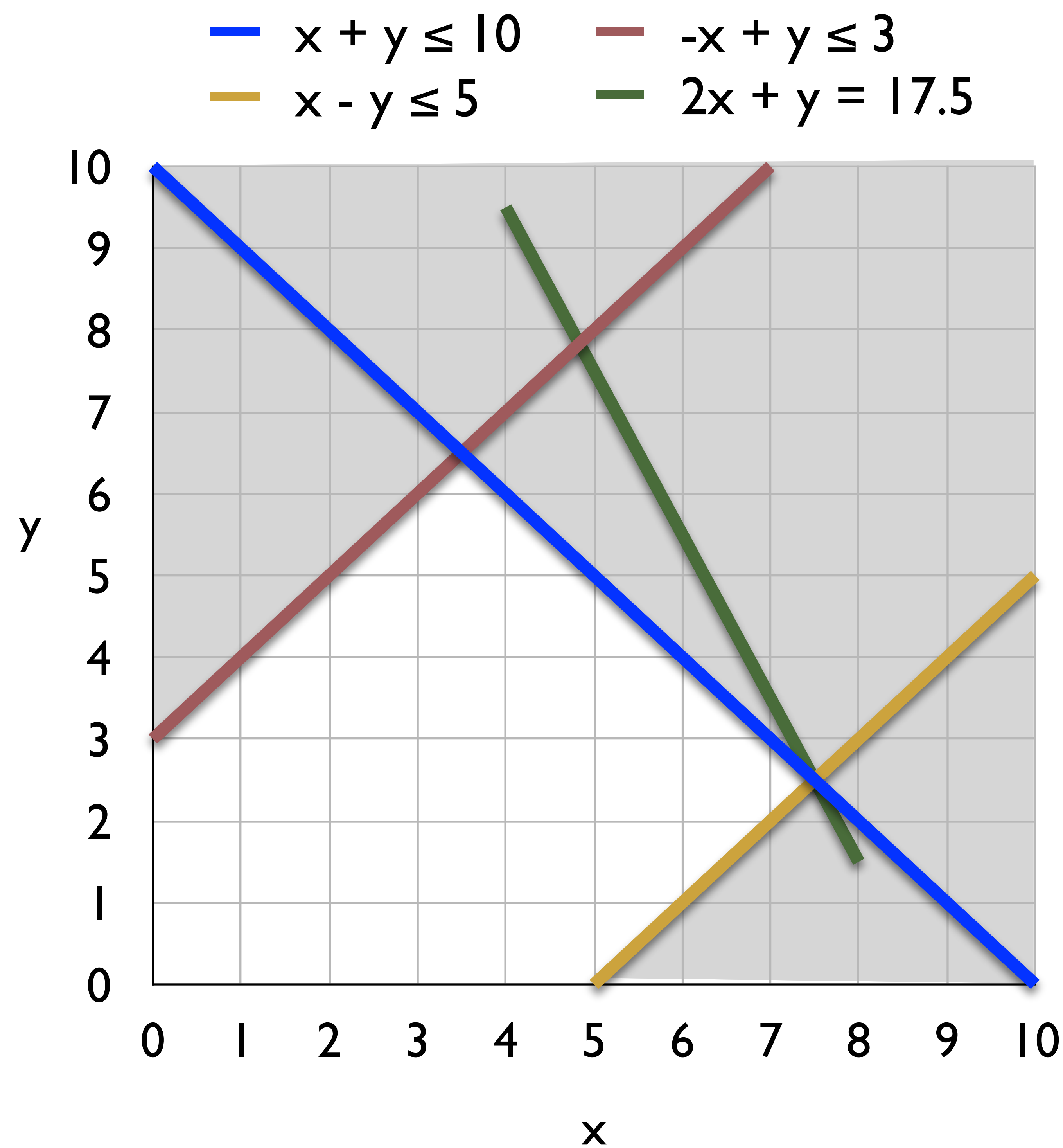
$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$



Résolution graphique

- Il est possible de résoudre graphiquement un programme linéaire.
- Il suffit de tracer la droite $c^Tx = k$ pour des valeurs croissantes de k et d'arrêter lorsque la droite atteint la frontière du polytope.

$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
 & -x & + & y & \leq & 3 \\
 & x & - & y & \leq & 5 \\
 & x & , & y & \geq & 0
\end{array}
$$

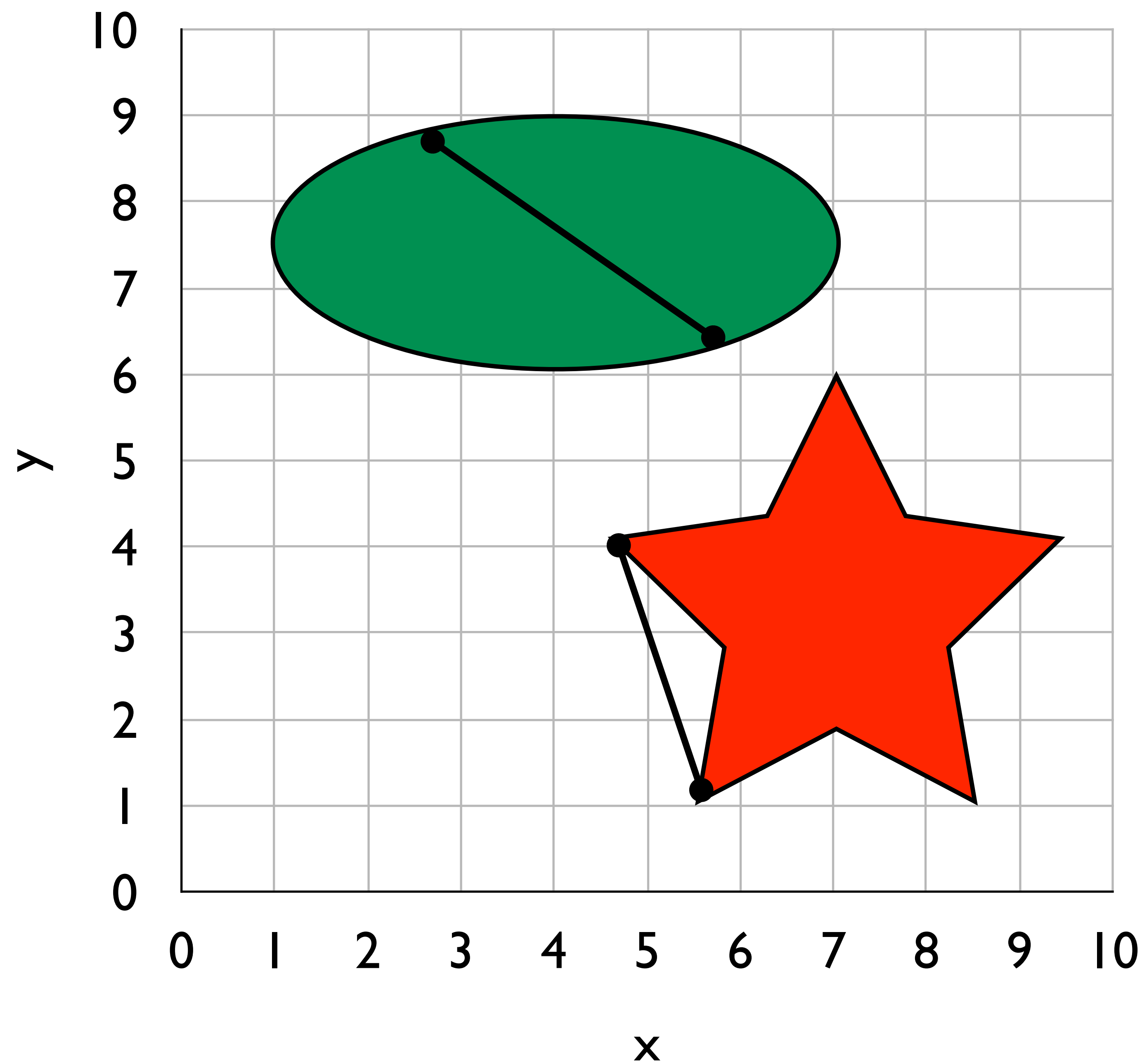


Propriétés des sommets

- La résolution graphique nous apprend que la solution (ou une solution) est nécessairement un sommet du polytope.
- Si le problème a plus d'une solution, alors l'espace des solutions est défini par une face du polytope ce qui inclut les sommets de cette face.
- La recherche d'une solution peut donc se limiter à l'ensemble des sommets du polytope.

Convexité

- Un espace est convexe si et seulement si pour tout point x et tout point y appartenant à cet espace, tous les points sur le segment reliant x à y font aussi partie de cet espace.
- Par exemple, une ellipse est une forme *convexe*, mais une étoile est *concave*.



Convexité

- De façon formelle, soit S un ensemble de points dans l'espace.
- S est convexe si et seulement si pour tous points x et y dans S et pour toute valeur α dans l'intervalle [0, 1], nous avons

$$\alpha\vec{x} + (1-\alpha)\vec{y} \in S$$

- Cette expression est appelée *combinaison convexe*.

Convexité

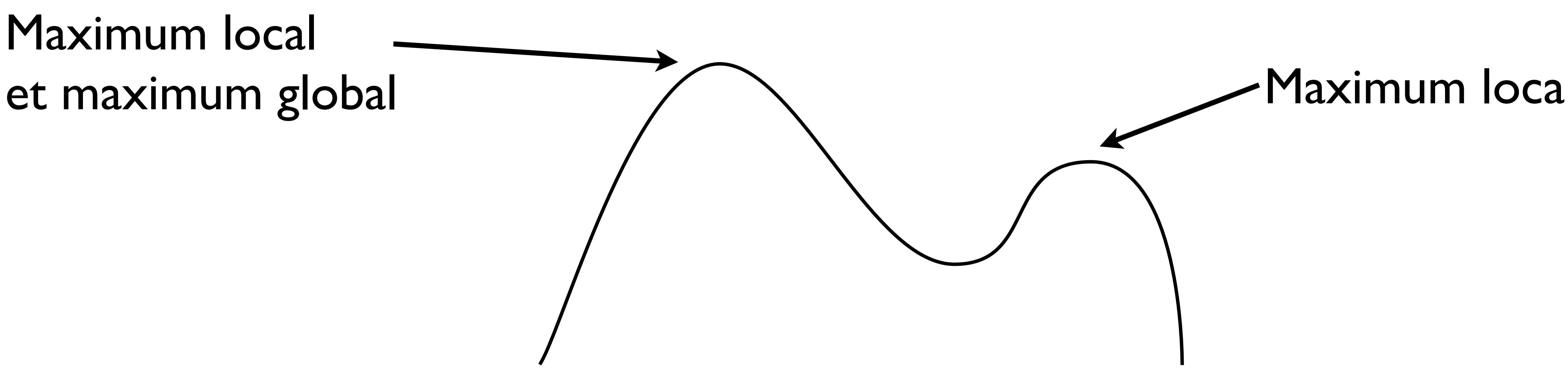
- Considérez ce programme linéaire.

$$\begin{array}{rrcl} \text{maximiser} & c^T x & & \\ \text{sujet à} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

- Soit y et z deux solutions valides (pas nécessairement optimales). Nous démontrons que αy + (1 - α)z est aussi une solution valide pour tout α dans [0, 1].

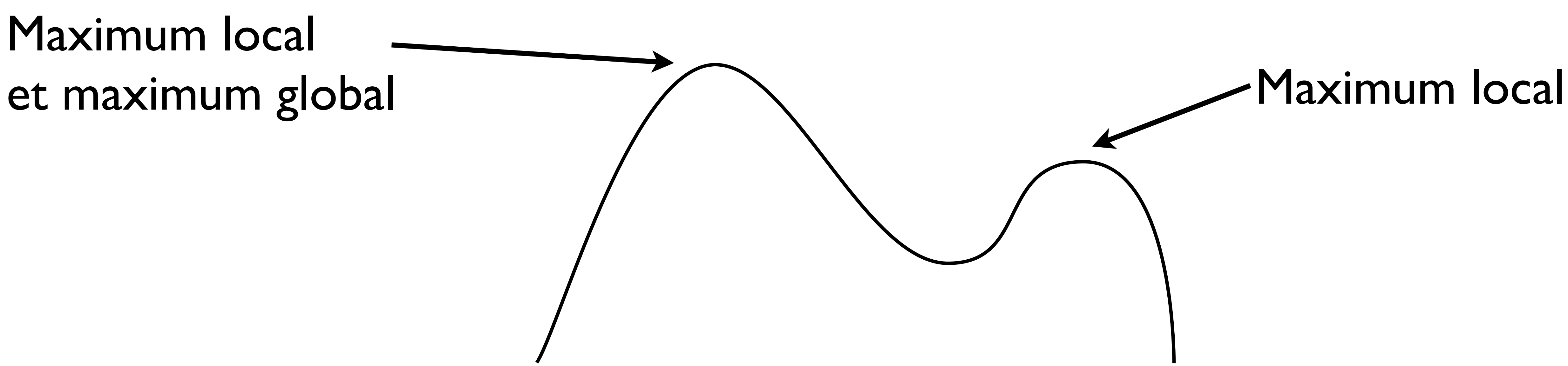
$$\begin{aligned} A(\alpha y + (1-\alpha)z) &= \alpha Ay + (1-\alpha)Az \\ &\leq \alpha b + (1-\alpha)b \\ &= b \\ \alpha y + (1-\alpha)z &\geq 0 + (1-\alpha)z \\ &\geq 0 \end{aligned}$$

Maximum local vs maximum global



- Un **maximum local** d'une fonction est un point qui est le plus haut parmi ses voisins (les points qui précèdent et qui suivent).
- Le **maximum global** est le point où une fonction prend sa plus grande valeur.

Maximum et convexité



- Pour trouver un maximum, il suffit de monter le long de la fonction jusqu'au point maximum. Ce maximum peut être local ou global. Cette méthode s'appelle la **montée du gradient** ou **descente de gradient** dans les problèmes de minimisation.
- Dans un espace convexe, il n'y a qu'un maximum. Ce maximum est donc global ce qui permet à la méthode de la montée du gradient de retourner une solution optimale.

Algorithme du simplexe

- George Dantzig proposa en 1947 le premier algorithme pour résoudre un programme linéaire.
- **L’algorithme du simplexe** exploite deux propriétés d’un programme linéaire.
  - La solution se retrouve à un sommet du polytope.
  - L’espace de recherche est convexe.
- L’algorithme trouve d’abord une solution réalisable, c'est-à-dire un sommet du polytope.
- L’algorithme parcourt ensuite les sommets du polytope le long des arêtes jusqu’à ce qu’aucun sommet adjacent n’ait une plus grande valeur objective. Grâce à la convexité du polytope, on sait que ce sommet est optimal.

Variables d'écart

- L'algorithme du simplexe résout un programme linéaire dans la forme suivante. Notez la présence de l'égalité.

$$\begin{array}{rrcl} \text{maximiser} & c^T x & & \\ \text{sujet à} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

- Une inégalité peut être transformée en égalité par l'ajout d'une **variable d'écart**.

$$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \leq k$$
$$\Longleftrightarrow$$
$$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n + s = k$$
$$s \geq 0$$

Variables d'écart

- Ce programme linéaire...

$$
\begin{array}{lccccccc}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\
& 2x_1 & + & x_2 & + & 3x_3 & \leq & 24 \\
& 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\
& x_1 & , & x_2 & , & x_3 & \geq & 0
\end{array}
$$

- ... est transformé en celui-ci.

$$
\begin{array}{lccccccccc}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & + & s_1 & = & 30 \\
& 2x_1 & + & x_2 & + & 3x_3 & + & s_2 & = & 24 \\
& 4x_1 & + & x_2 & + & 2x_3 & + & s_3 & = & 36 \\
& & & \multicolumn{5}{r}{x_1, s_1, x_2, s_2, x_3, s_3} & \geq & 0
\end{array}
$$

Solution initiale

- L'algorithme du simplexe démarre avec une solution réalisable (mais pas nécessairement optimale).
- **Question:** Trouvez une solution réalisable au programme linéaire suivant.

$$
\begin{array}{lrcrcrcrcr}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & + & s_1 & = & 30 \\
 & 2x_1 & + & x_2 & + & 3x_3 & + & s_2 & = & 24 \\
 & 4x_1 & + & x_2 & + & 2x_3 & + & s_3 & = & 36 \\
 & & & \multicolumn{5}{r}{x_1, s_1, x_2, s_2, x_3, s_3} & \geq & 0
\end{array}
$$

Solution initiale

- L'algorithme du simplexe démarre avec une solution réalisable (mais pas nécessairement optimale).
- **Question:** Trouvez une solution réalisable au programme linéaire suivant.

$$
\begin{array}{lccccccccc}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & + & s_1 & = & 30 \\
 & 2x_1 & + & x_2 & + & 3x_3 & + & s_2 & = & 24 \\
 & 4x_1 & + & x_2 & + & 2x_3 & + & s_3 & = & 36 \\
 & & & & & & x_1, s_1, x_2, s_2, & x_3, s_3 & \geq & 0
\end{array}
$$

- **Réponse:** On fixe les variables x_i à zéro. Les variables d'écart s_i prennent alors la seule valeur possible, celle à droite de l'équation.

Solutions réalisables de base

- Si la solution de départ a été si facile à obtenir, c'est qu'il y avait suffisamment de variables appartenant à la portée d'une seule contrainte.
- En fait, il y avait autant de variables appartenant à la portée d'une seule contrainte qu'il y a de contraintes.
- De plus, dans sa représentation matricielle, les colonnes associées aux variables non-nulles sont linéairement indépendantes. Elle forment donc une base.

$$\begin{bmatrix} 1 & 1 & 3 & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 2 & 1 & 3 & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 4 & 1 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \mathbf{s_1} \\ \mathbf{s_2} \\ \mathbf{s_3} \end{bmatrix} = \begin{bmatrix} 30 \\ 24 \\ 36 \end{bmatrix}$$

Solutions réalisables de base

- Considérons un programme linéaire de n variables et m contraintes.
- Une **solution de base** est une solution avec m variables non nulles de sorte que les colonnes associées à ces variables forment une base.
- Une **solution réalisable** est une solution qui satisfait toutes les équations et dont les valeurs sont non négatives.
- Une **solution réalisable de base** est à la fois une solution réalisable et une solution de base.
- L'ensemble des solutions réalisables de base forme l'ensemble des **sommets** du polytope.

Changement de base

- Pour visiter un sommet, il faut trouver une nouvelle base.
- Cette opération se fait en augmentant la valeur d'une variable nulle.
- En changeant la valeur d'une variable nulle, elle devient positive et entre dans la base.
- Une variable entre donc dans la base en devenant positive alors qu'une autre variable quitte la base en devenant nulle.

Changement de base

- Nous avons ici la solution de base $x_1 = x_2 = x_3 = 0$, $s_1 = 30$, $s_2 = 24$ et $s_3 = 36$.
- C'est x_1 qui augmentera le plus la valeur objective.
- Si on augmente la valeur de x_1, la première contrainte fait réduire la valeur de s_1. On peut augmenter x_1 jusqu'à 30 avant que s_1 ne devienne négative.
- De même, la deuxième contrainte permet d'augmenter x_1 jusqu'à 12 et la troisième contrainte jusqu'à 9. C'est donc la troisième contrainte la plus contraignante.

$$
\begin{array}{lrcrcrcrcr}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & + & s_1 & = & 30 \\
 & 2x_1 & + & x_2 & + & 3x_3 & + & s_2 & = & 24 \\
 & 4x_1 & + & x_2 & + & 2x_3 & + & s_3 & = & 36 \\
 & & & \multicolumn{5}{r}{x_1, s_1, x_2, s_2, x_3, s_3} & \geq & 0
\end{array}
$$

Changement de base

- On augmente donc x_1 à 9 et réduit s_3 à 0 pour obtenir la solution $x_1 = 9$, $x_2 = x_3 = 0$, $s_1 = 21$ $s_2 = 6$ et $s_3 = 0$.
- Les trois variables qui formaient une base ont été diminuées. La variable s_3 a été réduite à zéro et est donc sortie de la base alors que x_1 y est entrée.
- La valeur objective est passée de 0 à 27.

$$
\begin{array}{lrcrcrcrcr}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & + & s_1 & = & 30 \\
 & 2x_1 & + & x_2 & + & 3x_3 & + & s_2 & = & 24 \\
 & 4x_1 & + & x_2 & + & 2x_3 & + & s_3 & = & 36 \\
 & & & & & \multicolumn{3}{r}{x_1, s_1, x_2, s_2, x_3, s_3} & \geq & 0
\end{array}
$$

Changement de base

- On désire ensuite que la variable x_1 n'apparaisse qu'une seule fois dans le système d'équations.
- On divise donc la troisième équation par 4.

$$
\begin{array}{lrcrcrcrcr}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & \\
\text{sujet à} & x_1 & + & x_2 & + & 3x_3 & + & s_1 & = & 30 \\
 & 2x_1 & + & x_2 & + & 3x_3 & + & s_2 & = & 24 \\
 & x_1 & + & \frac{1}{4}x_2 & + & \frac{1}{2}x_3 & + & \frac{1}{4}s_3 & = & 9 \\
 & & & & & & x_1, s_1, x_2, & s_2, x_3, s_3 & \geq & 0
\end{array}
$$

Changement de base

- On élimine ensuite la variable x_1 des autres contraintes en soustrayant la troisième contrainte des deux premières.
- Remarquez que les valeurs des variables de la base apparaissent du côté droit des équations.

$$
\begin{array}{lrcrcrcrcrcr}
\text{maximiser} & 3x_1 & + & x_2 & + & 2x_3 & & & & & & \\
\text{sujet à} & & & \frac{3}{4}x_2 & + & \frac{5}{2}x_3 & + & s_1 & - & \frac{1}{4}s_3 & = & 21 \\
& & & \frac{1}{2}x_2 & + & 2x_3 & + & s_2 & - & \frac{1}{2}s_3 & = & 6 \\
& x_1 & + & \frac{1}{4}x_2 & + & \frac{1}{2}x_3 & + & & & \frac{1}{4}s_3 & = & 9 \\
& & & & & x_1, s_1, x_2, s_2, x_3, s_3 & & & \geq & 0 & &
\end{array}
$$

Tableau

- Pour mieux comprendre et structurer les opérations de changement de base, l'algorithme du simplexe encode le problème sous forme de tableau.
- Les opérations de changement de base se font ensuite sur ce tableau.
- L'état du tableau indique entre autres si la solution est optimale.

Tableau

- On dispose dans un tableau de la façon suivante les éléments de la matrice A, du vecteur c et du vecteur b ainsi que la valeur objective v de la solution de base courante.

$$\begin{array}{rrcl} \text{maximiser} & c^T x & & \\ \text{sujet à} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

$-c_1$	$-c_2$	$-c_3$	...	$-c_n$	v
a_{11}	a_{12}	a_{13}	...	a_{1n}	b_1
a_{21}	a_{22}	a_{23}	...	a_{2n}	b_2
...	...	...	...	...	...
a_{m1}	a_{m2}	a_{m3}	...	a_{mn}	b_m

Exemple

$$\begin{array}{lrcrcccccl}
\text{maximiser} & 2x & + & y & & & & & & \\
\text{sujet à} & x & + & y & + & s_1 & & & = & 10 \\
 & -x & + & y & + & & s_2 & & = & 3 \\
 & x & - & y & + & & & s_3 & = & 5 \\
 & x & , & y & \geq & 0 & & & &
\end{array}$$

-2	-1	0	0	0	0
1	1	1	0	0	10
-1	1	0	1	0	3
1	-1	0	0	1	5

Invariants du tableau

- Ce tableau a plusieurs invariants que nous maintiendrons durant la recherche.
- Les colonnes de la base ont des valeurs nulles à l'exception d'une composante qui est égale à un.
- Le coût associé à une variable de la base est 0.
- La valeur objective de la solution courante apparaît dans le coin supérieur droit.
- Les valeurs des variables non nulles apparaissent à la dernière colonne du tableau, sous la valeur objective. Puisque la solution est réalisable, ces valeurs sont non négatives.

Opération de pivot

- L'opération de pivot est le remplacement d'une variable de la base par une autre variable.
- On choisit d'abord la variable qui entre dans la base. Ce choix est généralement la variable dont le coût est négatif et dont la valeur absolue est la plus grande. Soit x_j la variable choisie.
- La variable x_i qui sort de la base est celle dont l'élément $a_{ij} > 0$ et qui minimise le ratio b_i / a_{ij}. L'élément a_{ij} est appelé le **pivot**.
- Une fois le pivot choisi, on divise la rangée i par a_{ij} afin que l'élément a_{ij} devienne 1.
- On soustrait ensuite a_{kj} fois la rangée i de la rangée k afin que l'élément a_{kj} devienne nul.
- On soustrait ensuite c_j fois la rangée i de la première rangée du tableau.

Opération de pivot

- La variable x_j qui entre dans la base est celle qui a un coût négatif avec la plus grande valeur absolue.

-2	-1	0	0	0	0
1	1	1	0	0	10
-1	1	0	1	0	3
1	-1	0	0	1	5

Opération de pivot

- La variable x_j qui entre dans la base est celle qui a un coût négatif avec la plus grande valeur absolue.
- La variable x_i qui sort de la base est celle dont l'élément $a_{ij} > 0$ et qui minimise b_i / a_{ij}.

-2	-1	0	0	0	0
1	1	1	0	0	10
-1	1	0	1	0	3
1	-1	0	0	1	5

Opération de pivot

- On divise la quatrième rangée du tableau par 1 (sans effet).
- On met les entrées de la colonne du pivot (autre que le pivot) à zéro en additionnant ou soustrayant la rangée du pivot aux autres rangées.

-2	-1	0	0	0	0
1	1	1	0	0	10
-1	1	0	1	0	3
1	-1	0	0	1	5

Opération de pivot

- On divise la quatrième rangée du tableau par 1 (sans effet).
- On met les entrées de la colonne du pivot (autre que le pivot) à zéro en soustrayant la rangée du pivot aux autres rangées.

0	-3	0	0	2	10
0	2	1	0	-1	5
0	0	0	1	1	8
1	-1	0	0	1	5

Solution optimale

- Lorsque l'on construit le tableau, on y inscrit le vecteur -c. La négation transforme donc le problème de maximisation en minimisation. C'est pourquoi on fait entrer dans la base les valeurs dont le coût est négatif. Cette négation sert seulement à rendre la valeur dans le coin supérieur droit égale à la valeur objective.

- Minimiser une fonction $-f(x)$ est équivalent à minimiser la fonction $-f(x) + k$ où k est une constante quelconque.

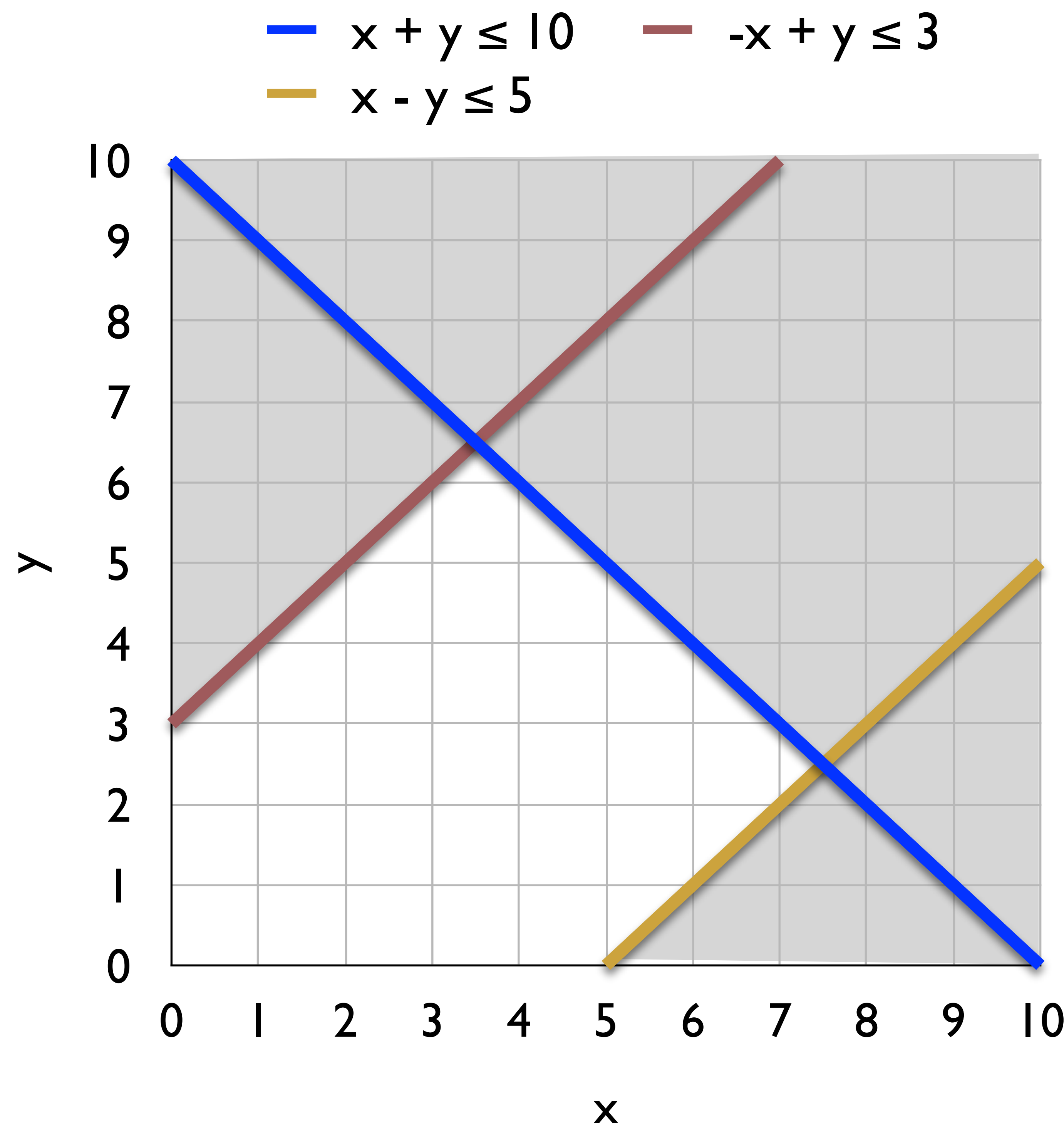
- Lorsque l'on additionne une ligne du tableau, on additionne en fait une combinaison linéaire de variables égale à une constante. Cette opération change donc le problème en minimisant $-f(x) + b_i$ plutôt que $-f(x)$. La solution, elle, demeure inchangée.

Solution optimale

- Lorsque tous les coefficients des variables dans $-f(x) + k$ sont positifs, il n'est plus possible d'augmenter une variable pour faire diminuer la fonction objectif.
- Nous avons donc atteint un minimum local et, par convexité du polytope, un minimum global.
- La solution réalisable de base est donc optimale.

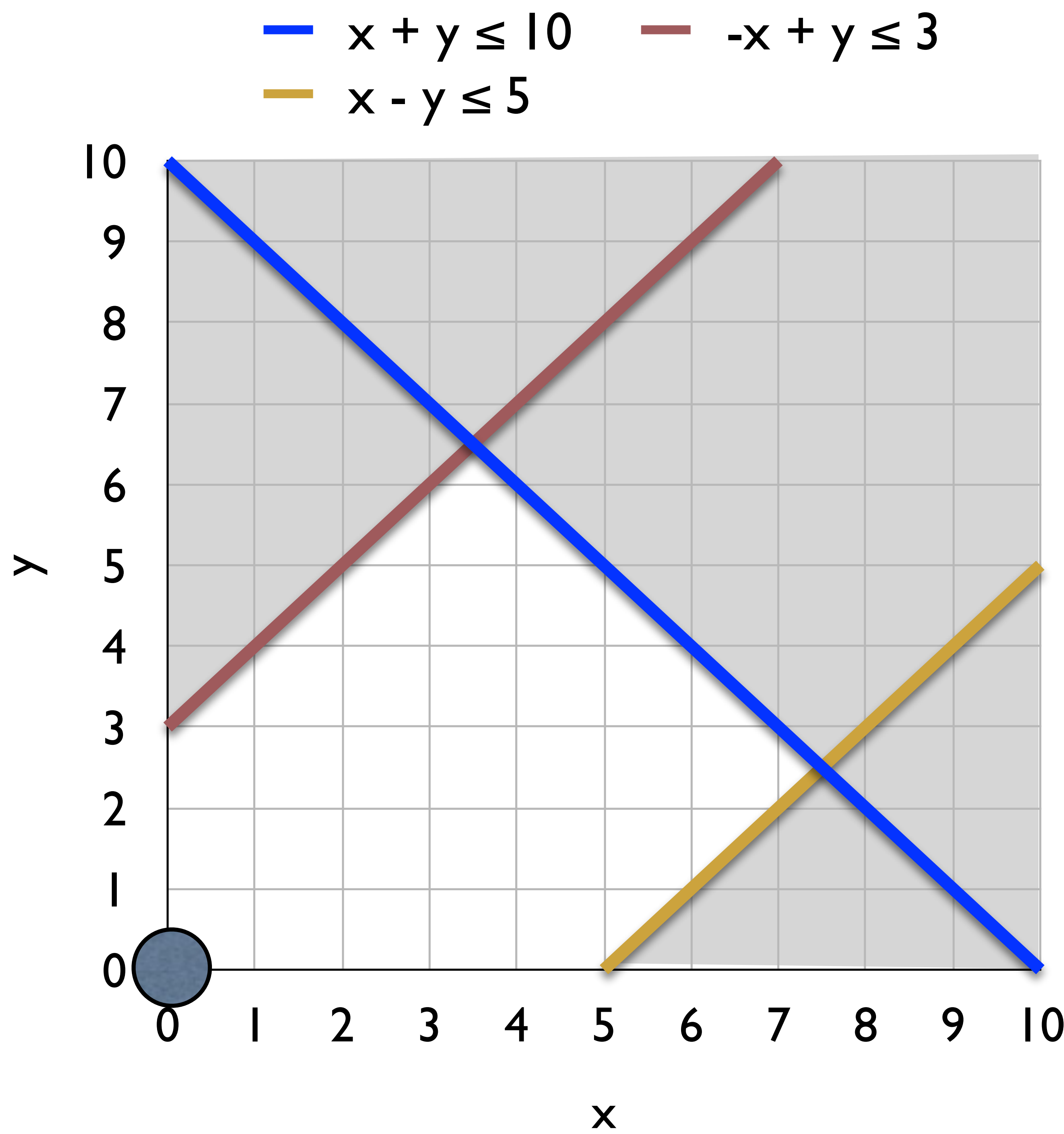
Exemple complet

$$\begin{array}{lrcrcr} \text{maximiser} & 2x & + & y & & \\ \text{sujet à} & x & + & y & \leq & 10 \\ & -x & + & y & \leq & 3 \\ & x & - & y & \leq & 5 \\ & x & , & y & \geq & 0 \end{array}$$



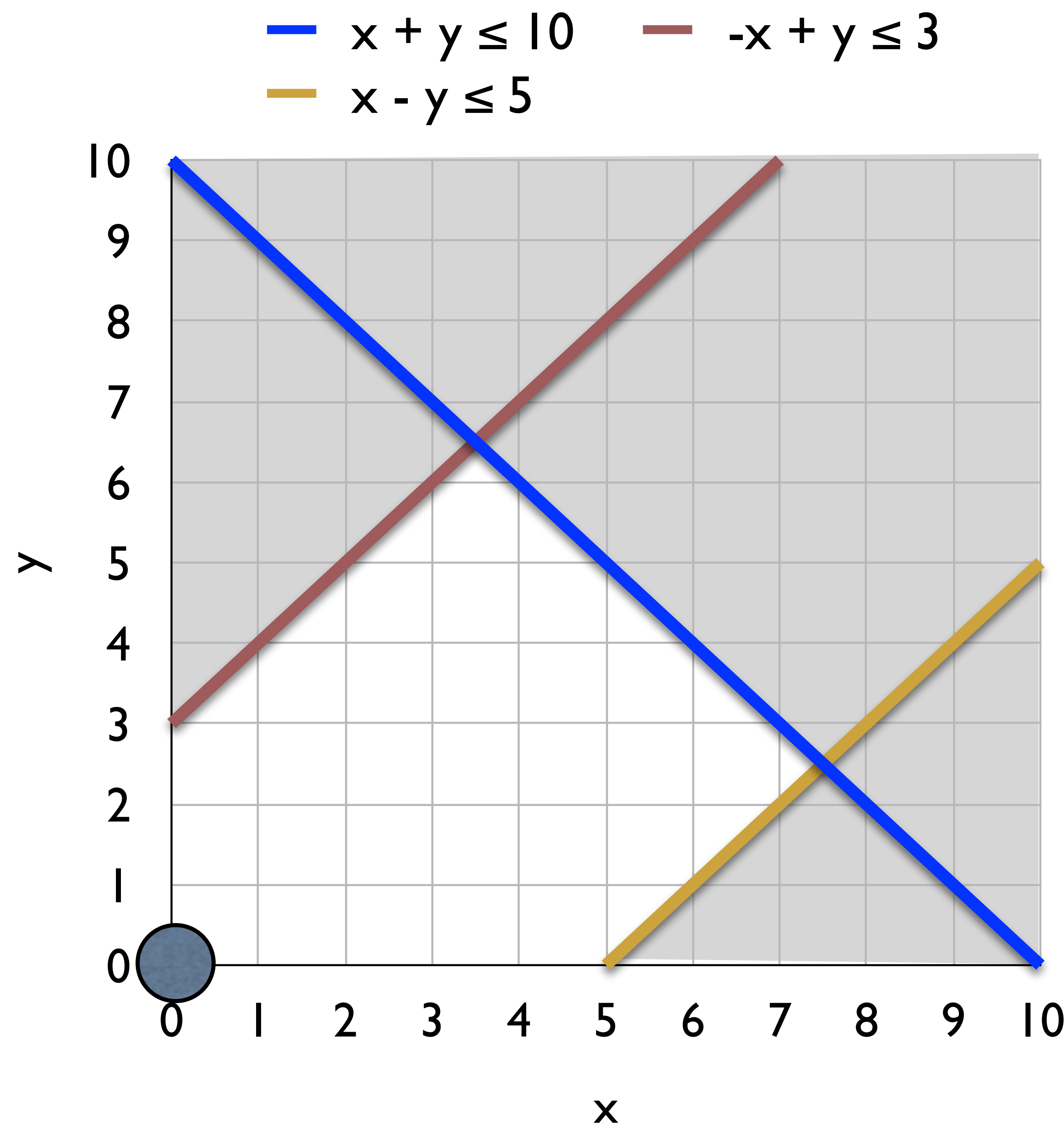
Exemple complet

-2	-1	0	0	0	0
1	1	1	0	0	10
-1	1	0	1	0	3
1	-1	0	0	1	5



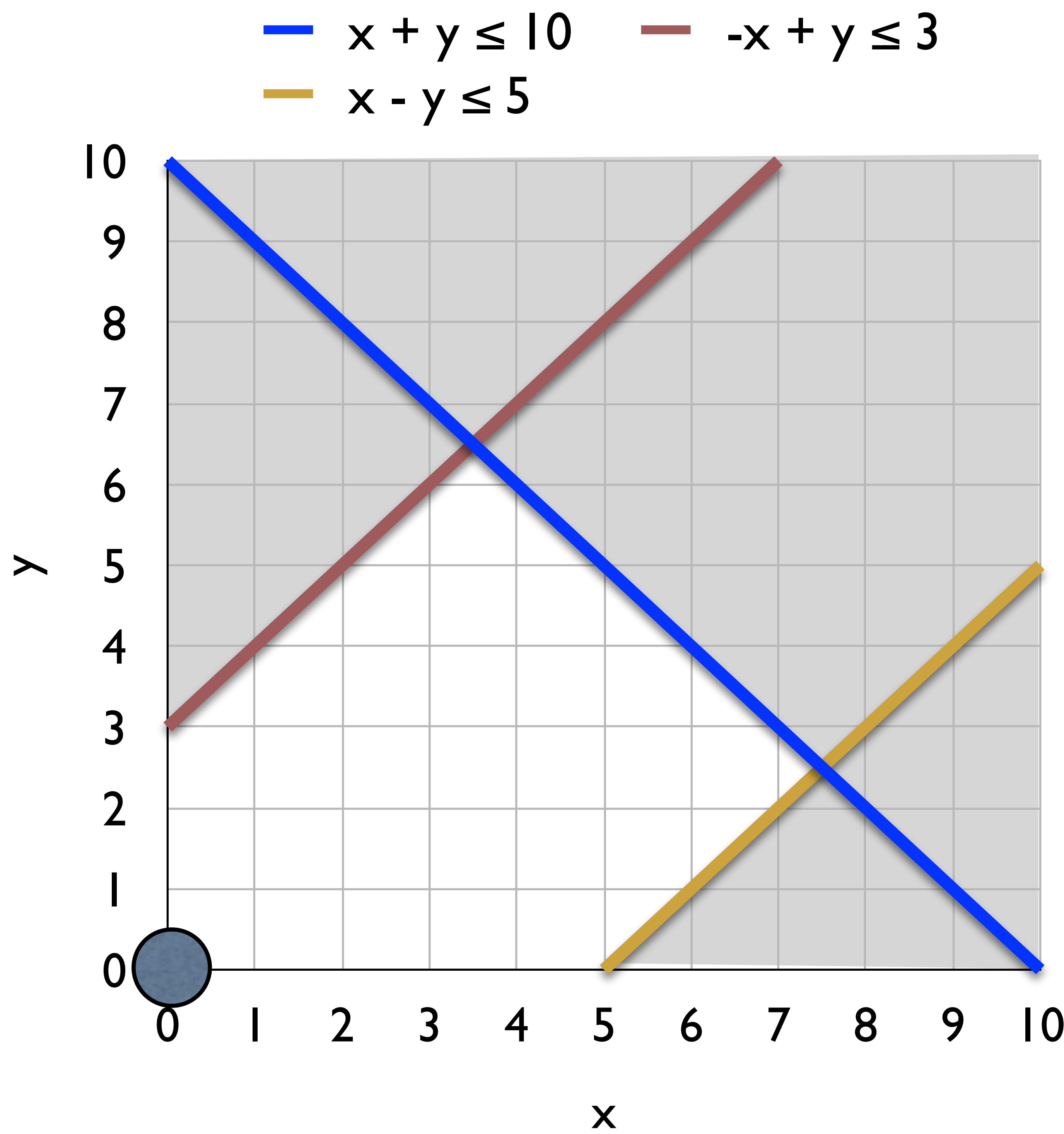
Exemple complet

-2	-1	0	0	0	0
1	1	1	0	0	10
-1	1	0	1	0	3
1	-1	0	0	1	5



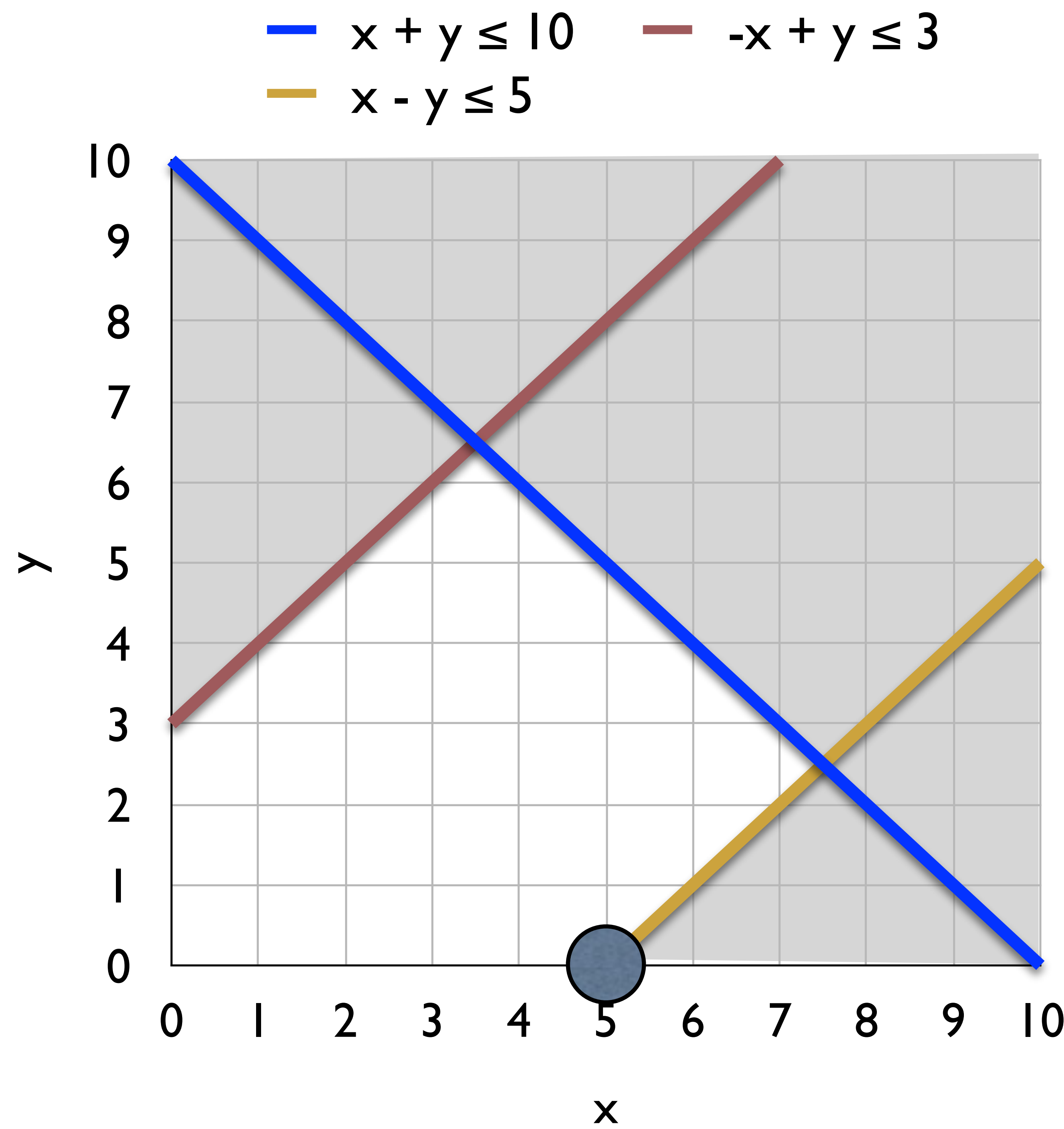
Exemple complet

0	-3	0	0	2	10
0	2	1	0	-1	5
0	0	0	1	1	8
1	-1	0	0	1	5



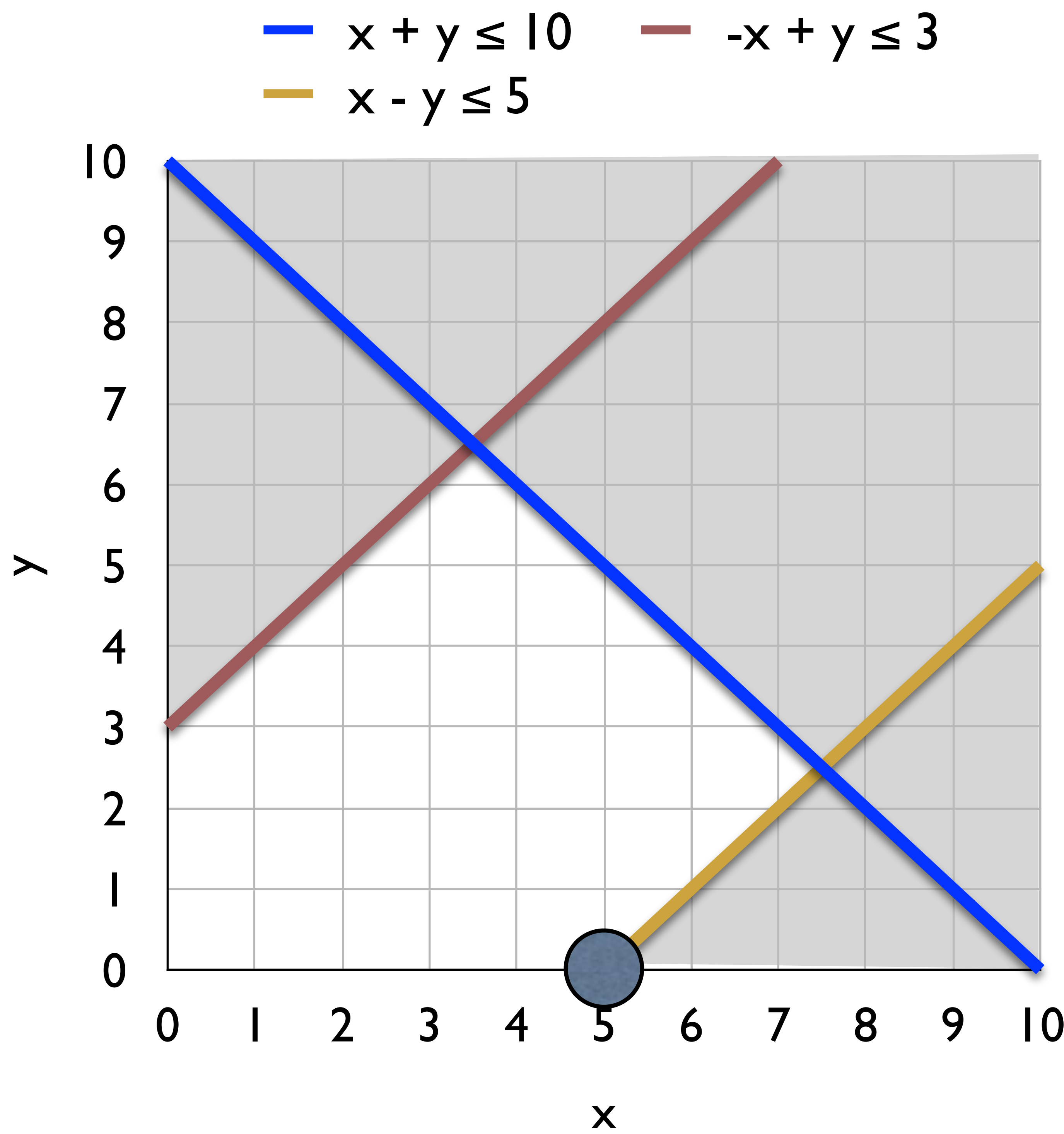
Exemple complet

0	-3	0	0	2	10
0	2	1	0	-1	5
0	0	0	1	1	8
1	-1	0	0	1	5



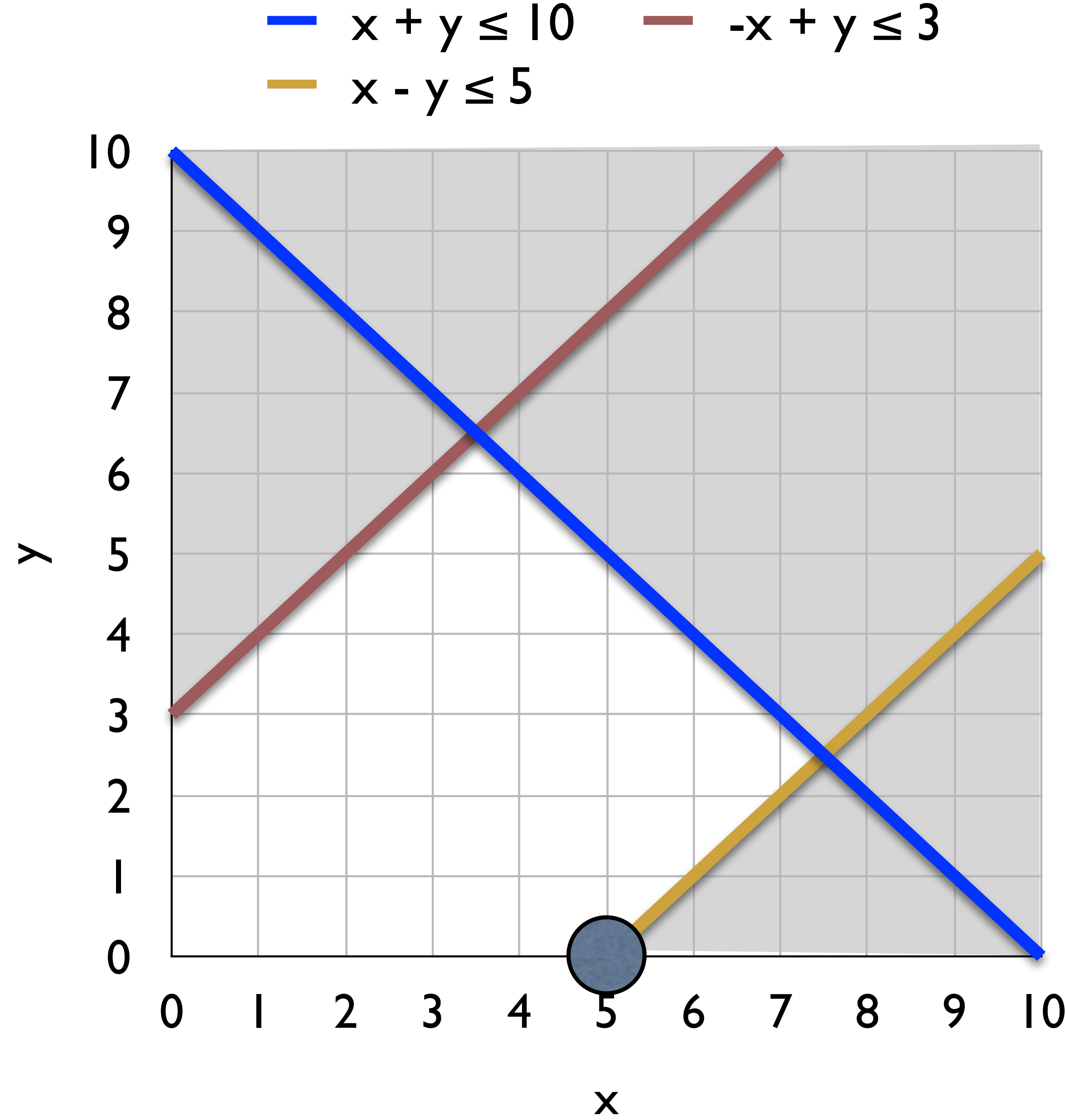
Exemple complet

0	-3	0	0	2	10
0	2	1	0	-1	5
0	0	0	1	1	8
1	-1	0	0	1	5



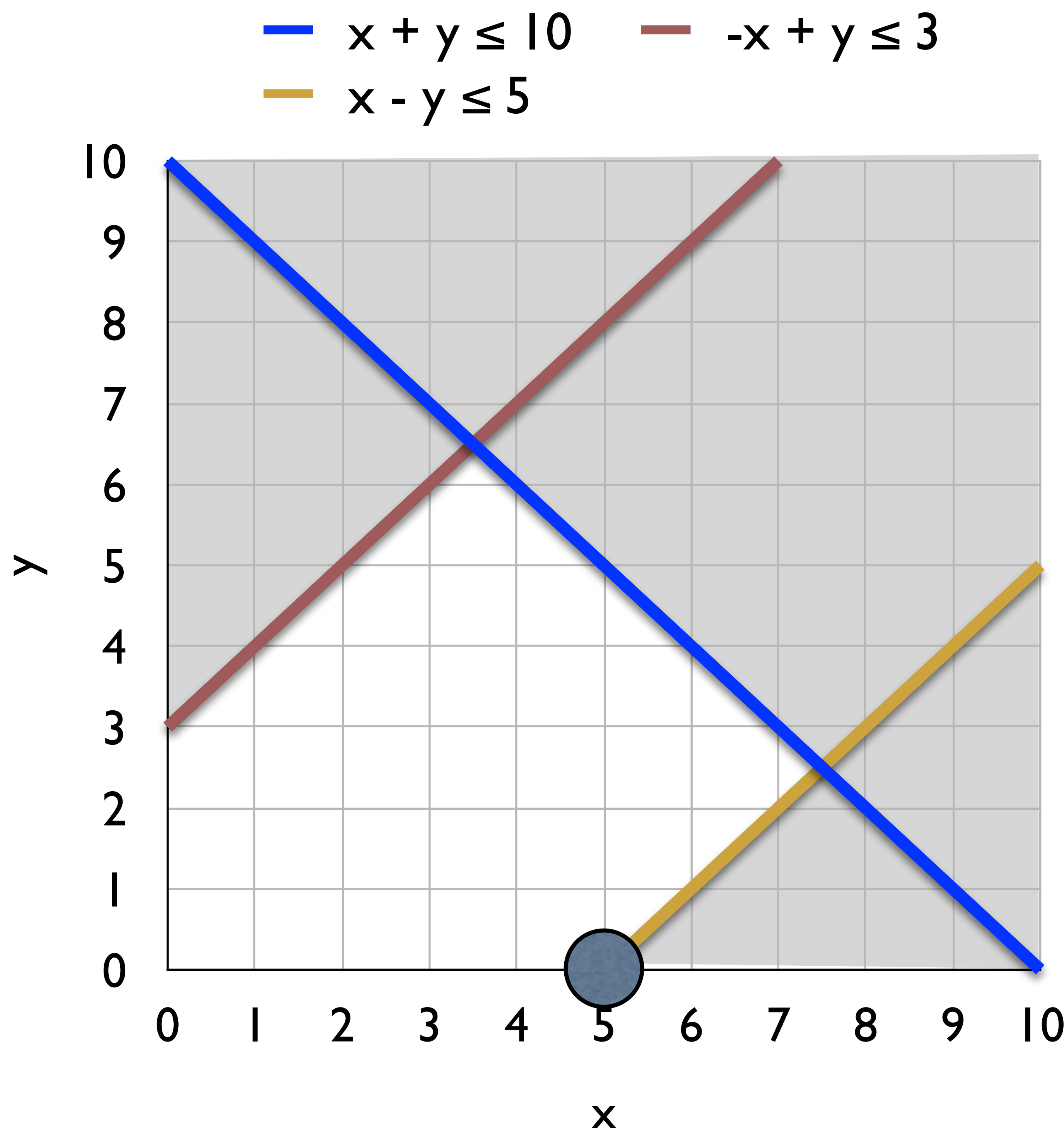
Exemple complet

0	-3	0	0	2	10
0	2	1	0	-1	5
0	0	0	1	1	8
1	-1	0	0	1	5



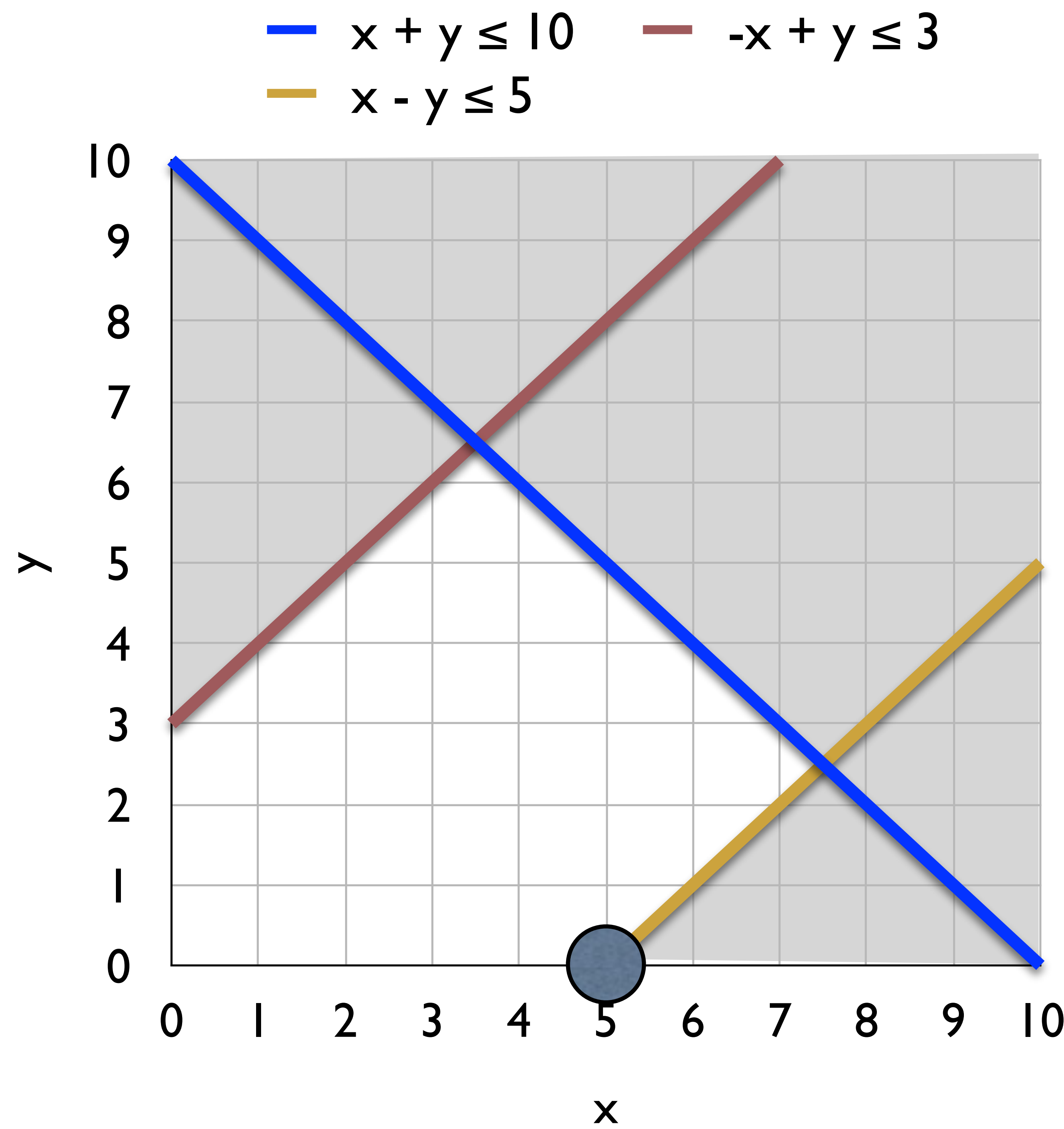
Exemple complet

0	-3	0	0	2	10
0	1	1/2	0	-1/2	5/2
0	0	0	1	1	8
1	-1	0	0	1	5



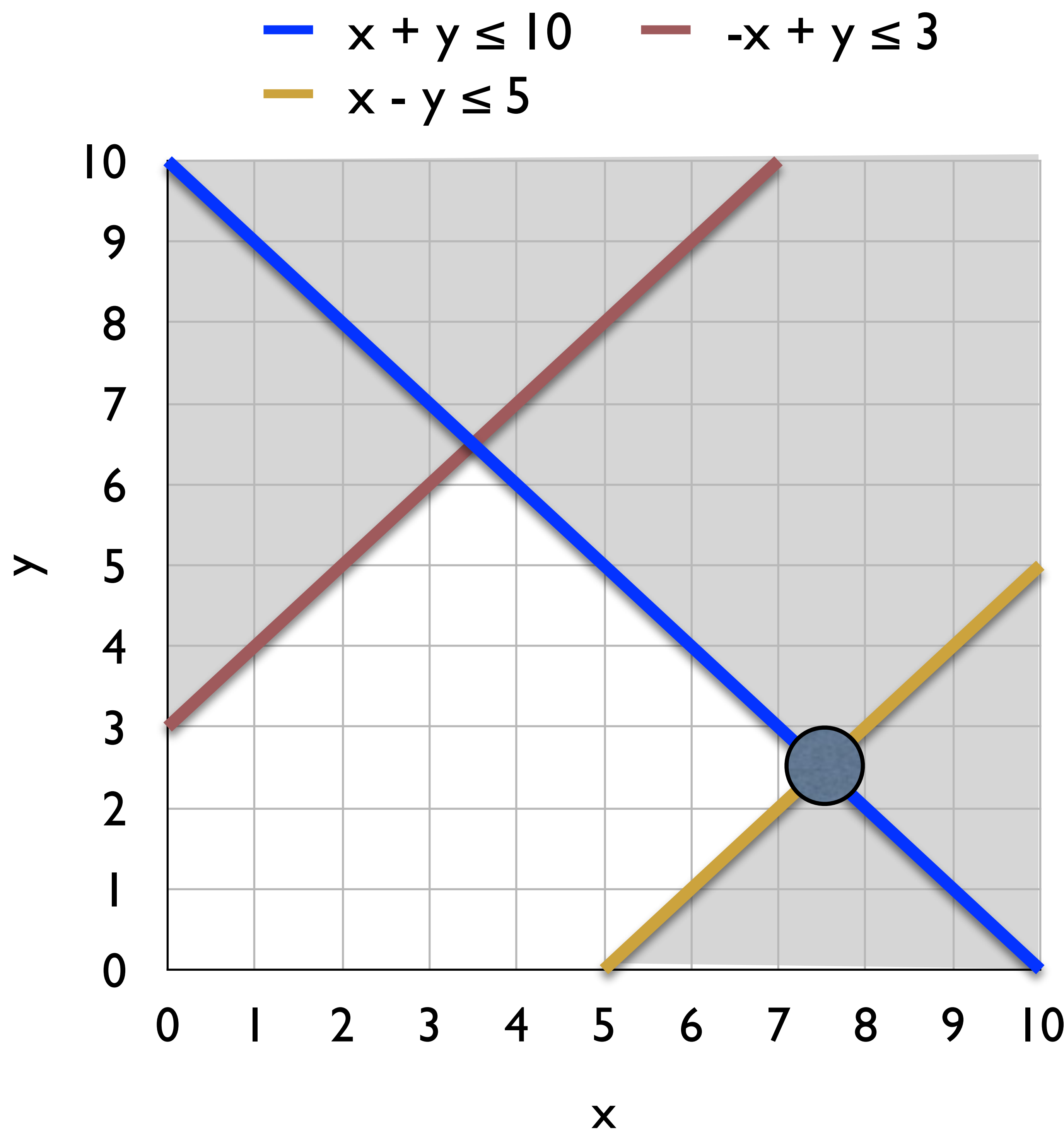
Exemple complet

0	0	3/2	0	1/2	17.5
0	1	1/2	0	-1/2	5/2
0	0	0	1	1	8
1	0	1/2	0	1/2	15/2



Exemple complet

0	0	3/2	0	1/2	17.5
0	1	1/2	0	-1/2	5/2
0	0	0	1	1	8
1	0	1/2	0	1/2	15/2



Solution initiale

- Il peut parfois être plus difficile de trouver une première solution réalisable de base.
- Par exemple, lorsque la valeur b_i apparaissant au côté droit d'une inégalité est négative, la solution de base formée par les variables d'écarts n'est pas réalisable.
- Dans l'exemple ci-dessous, la solution $x = 0$, $y = 0$, $s_1 = 10$, $s_2 = -3$ n'est pas réalisable, car $s_2 < 0$.

$$
\begin{array}{lrcrcr}
\text{maximiser} & 2x & + & y & & \\
\text{sujet à} & x & + & y & \leq & 10 \\
& -x & + & y & \leq & -3 \\
& x & , & y & \geq & 0
\end{array}
\Longrightarrow
\begin{array}{lrcrcrrcr}
\text{maximiser} & 2x & + & y & & & & & \\
\text{sujet à} & x & + & y & + & s_1 & & = & 10 \\
& -x & + & y & + & & s_2 & = & -3 \\
& x, & & y, & & s_1, & s_2 & \geq & 0
\end{array}
$$

Solution initiale

- Pour obtenir une solution initiale, on multiplie par -1 les deux côtés des équations dont le terme de droite est négatif.
- On ajoute ensuite une variable temporaire à chaque équation linéaire.
- On remplace la fonction objectif par la minimisation de la somme des variables temporaires.
- Une solution réalisable de base optimale pour ce nouveau problème est une solution réalisable de base (mais non optimale) pour le problème original.

Exemple

$$\begin{array}{lrcrcrcrcr}
\text{maximiser} & 2x & + & y & & & & & & \\
\text{sujet à} & x & + & y & + & s_1 & & & = & 10 \\
& -x & + & y & + & & & s_2 & = & -3 \\
& x, & & y, & & s_1, & & s_2 & \geq & 0
\end{array}
\iff
\begin{array}{lrcrcrcrcr}
\text{maximiser} & 2x & + & y & & & & & & \\
\text{sujet à} & x & + & y & + & s_1 & & & = & 10 \\
& x & - & y & - & & & s_2 & = & 3 \\
& x, & & y, & & s_1, & & s_2 & \geq & 0
\end{array}$$

$$\begin{array}{lrcrcrrcrcrcr}
\text{maximiser} & & & & & & & - & t_1 & - & t_2 & & \\
\text{sujet à} & x & + & y & + & s_1 & & + & t_1 & & & = & 10 \\
& x & - & y & - & & s_2 & + & & & t_2 & = & 3 \\
& \multicolumn{10}{c}{x, y, s_1, s_2, t_1, t_2} & \geq & 0
\end{array}$$

Exemple

$$
\begin{array}{lcccccccccccccc}
\text{maximiser} & & & & & & & & - & t_1 & - & t_2 & & \\
\text{sujet à} & x & + & y & + & s_1 & & & + & t_1 & & & = & 10 \\
 & x & - & y & - & & & s_2 & + & & & t_2 & = & 3 \\
 & & & & x, y, s_1, s_2, t_1, t_2 & & & & & & & & \geq & 0
\end{array}
$$

Tableau

0	0	0	0	1	1	0
1	1	1	0	1	0	10
1	-1	0	-1	0	1	3

Exemple

0	0	0	0	1	1	0
1	1	1	0	1	0	10
1	-1	0	-1	0	1	3

On choisit les deux variables temporaires pour la base et on rétablit l'invariant sur la fonction objectif.

-2	0	-1	1	0	0	-13
1	1	1	0	1	0	10
1	-1	0	-1	0	1	3

Exemple

-2	0	-1	1	0	0	-13
1	1	1	0	1	0	10
1	-1	0	-1	0	1	3

Opération de pivot

0	-2	-1	-1	0	2	-7
0	2	1	1	1	-1	7
1	-1	0	-1	0	1	3

Exemple

0	-2	-1	-1	0	2	-7
0	2	1	1	1	-1	7
1	-1	0	-1	0	1	3

Opération de pivot

0	0	0	0	1	1	0
0	1	1/2	1/2	1/2	-1/2	7/2
1	0	1/2	-1/2	1/2	1/2	13/2

Exemple

0	0	0	0	1	1	0
0	1	1/2	1/2	1/2	-1/2	7/2
1	0	1/2	-1/2	1/2	1/2	13/2

- Ce dernier tableau présente une solution optimale. Notez que les variables t_1 et t_2 sont nulles.
- La solution $x = 6.5$, $y = 3.5$, $s_1 = 0$, $s_2 = 0$ est donc une solution réalisable de base à la fois pour le problème avec les variables temporaires, mais aussi pour le problème original.
- On retire donc les deux variables temporaires et réinsère la fonction objectif originale.

Exemple

0	0	0	0	1	1	0
0	1	1/2	1/2	1/2	-1/2	7/2
1	0	1/2	-1/2	1/2	1/2	13/2

On retire les deux variables temporaires et réinsère la fonction objectif qui est de maximiser 2x + y

-2	-1	0	0	0
0	1	1/2	1/2	7/2
1	0	1/2	-1/2	13/2

Exemple

-2	-1	0	0	0
0	1	1/2	1/2	7/2
1	0	1/2	-1/2	13/2

On rétablit l'invariant sur la première rangée

0	0	3/2	-1/2	33/2
0	1	1/2	1/2	7/2
1	0	1/2	-1/2	13/2

Exemple

0	0	3/2	-1/2	33/2
0	1	1/2	1/2	7/2
1	0	1/2	-1/2	13/2

Opération de pivot. La solution x = 10 et y = 0 est optimale.

0	1	2	0	20
0	2	1	1	7
1	1	1	0	10

Infaisabilité

- **Question**: Qu'arrive-t-il si on ne réussit pas à trouver une solution initiale? C'est-à-dire qu'après avoir ajouté des variables temporaires et de les avoir minimisées, nous ne trouvons pas une solution où ces variables temporaires sont nulles.
- **Réponse**: Il n'y a pas de solution initiale, car il n'y a pas de solutions au problème. On dit alors que le problème n'est pas réalisable.

Polytope non borné

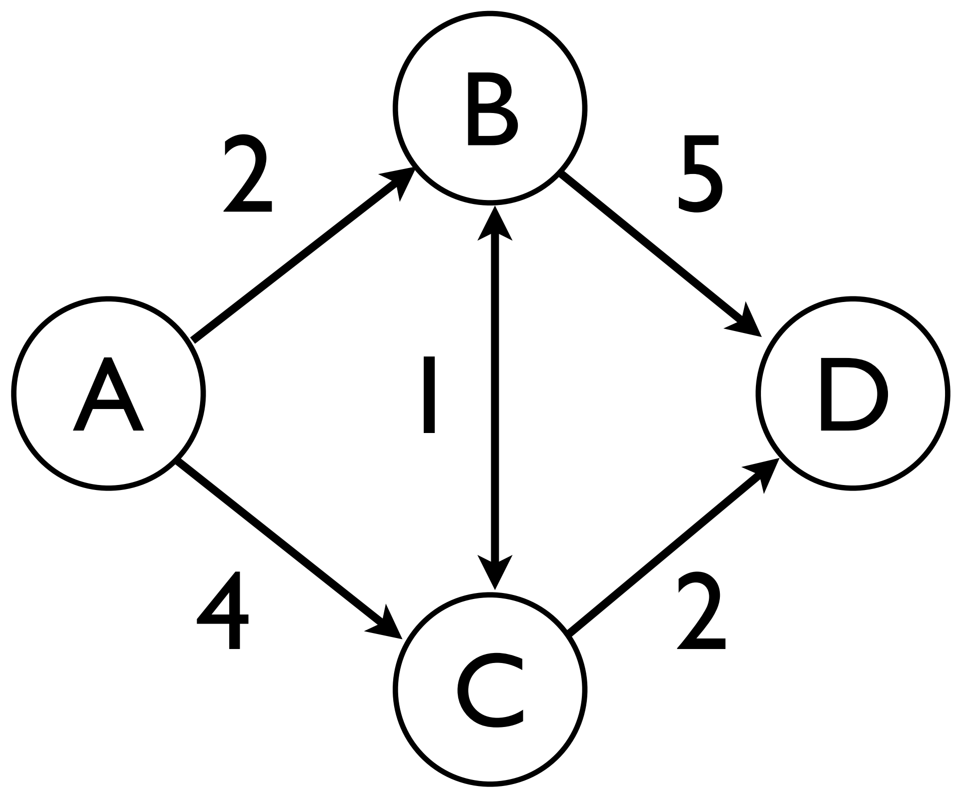
- **Question:** Qu'arrive-t-il si, lors du choix d'un pivot, toutes les valeurs de la colonne du pivot sont négatives ou nulles?
- **Réponse:** Il est possible d'augmenter la variable du pivot à l'infini sans violer aucune contrainte. La valeur objective tend donc vers l'infini. On dit alors que le problème n'est pas borné.

Complexité

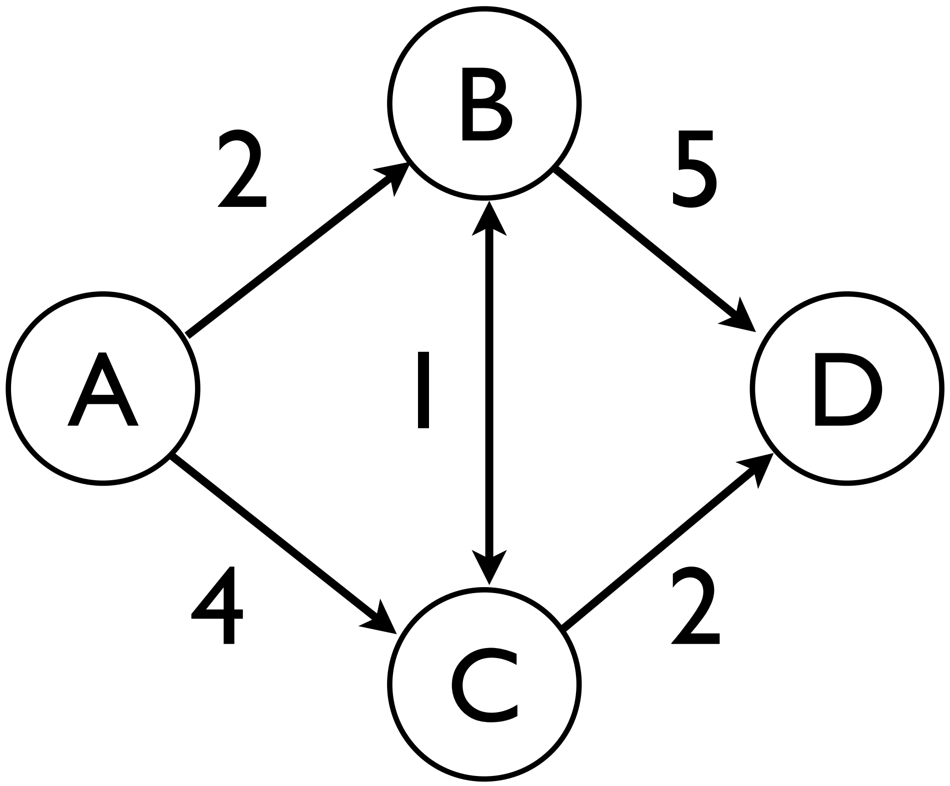
- Le nombre de sommets d'un polytope peut être exponentiel dans le nombre de contraintes.
- Malheureusement, il existe des programmes linéaires dont la résolution nécessite la visite d'un nombre exponentiel de sommets. La complexité, en pire cas, est donc exponentielle.
- Cependant, l'algorithme du simplexe est très efficace en pratique. Sa complexité en temps moyen est en fait polynomiale.
- Généralement, l'algorithme converge vers une solution entre 2m et 3m itérations où m est le nombre d'égalités dans le programme linéaire.

Le chemin le plus court

- **Question**: Comment peut-on modéliser le problème de trouver le chemin le plus court entre le noeud A et le noeud D sous la forme d'un programme linéaire?

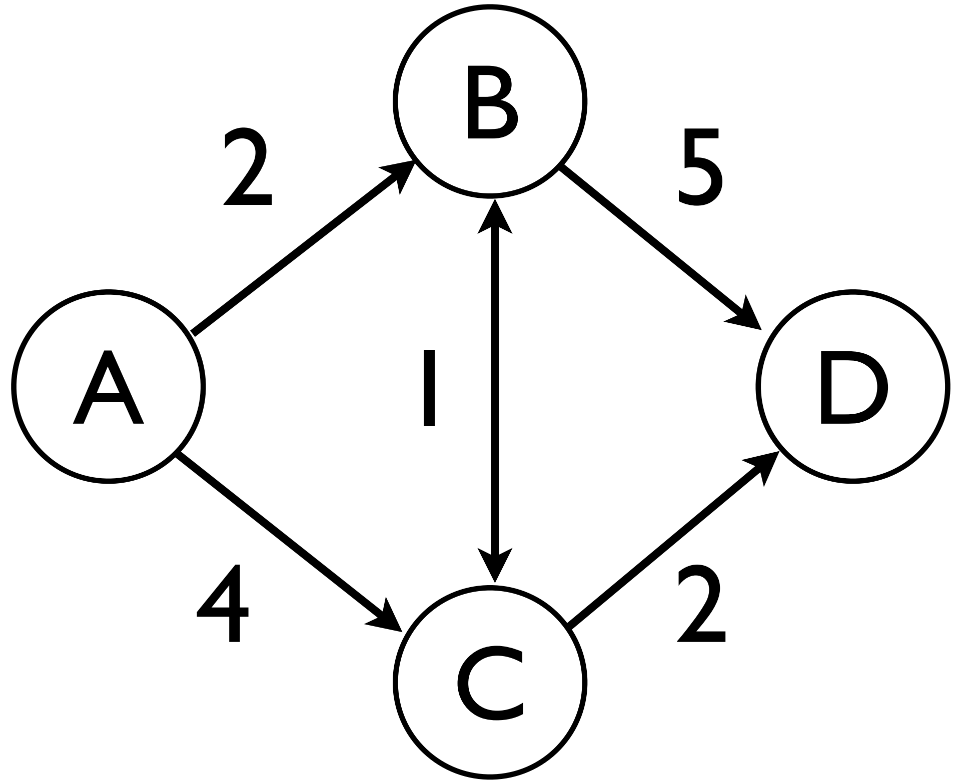


Premier modèle



- Un chemin peut être perçu comme un flot de valeur un.
- On peut donc déclarer une variable par arête et imposer les contraintes de conservation de flot sur ces variables.

Premier modèle



$$
\begin{array}{lrcrcrcrcrcrcr}
\text{minimiser} & 2x_{ab} & + & 4x_{ac} & + & x_{bc} & + & 5x_{bd} & + & x_{cb} & + & 2x_{cd} & & \\
\text{sujet à} & -x_{ab} & - & x_{ac} & & & & & & & & & = & -1 \\
 & x_{ab} & & & - & x_{bc} & - & x_{bd} & + & x_{cb} & & & = & 0 \\
 & & & x_{ac} & + & x_{bc} & & & - & x_{cb} & - & x_{cd} & = & 0 \\
 & & & & & & & x_{bd} & & & + & x_{cd} & = & 1 \\
 & \multicolumn{11}{c}{x_{ab}, x_{ac}, x_{bc}, x_{bd}, x_{cb}, x_{cd}} & \geq & 0
\end{array}
$$

Premier modèle

- Notez que les quatre équations obtenues sont linéairement dépendantes. En effet, la première équation peut être obtenue en additionnant les trois autres équations ensembles et en multipliant cette somme par -1.
- Puisque la première équation est redondante, on simplifie le modèle en l'omettant.

$$
\begin{array}{lrcrcrcrcrcrcrr}
\text{minimiser} & 2x_{ab} & + & 4x_{ac} & + & x_{bc} & + & 5x_{bd} & + & x_{cb} & + & 2x_{cd} & & \\
\text{sujet à} & -x_{ab} & - & x_{ac} & & & & & & & & & = & -1 \\
 & x_{ab} & & & - & x_{bc} & - & x_{bd} & + & x_{cb} & & & = & 0 \\
 & & & x_{ac} & + & x_{bc} & & & - & x_{cb} & - & x_{cd} & = & 0 \\
 & & & & & & & x_{bd} & & & + & x_{cd} & = & 1 \\
 & \multicolumn{11}{c}{x_{ab}, x_{ac}, x_{bc}, x_{bd}, x_{cb}, x_{cd}} & \geq & 0
\end{array}
$$

Premier modèle

- Notez que les quatre équations obtenues sont linéairement dépendantes. En effet, la première équation peut être obtenue en additionnant les trois autres équations ensembles et en multipliant cette somme par -1.
- Puisque la première équation est redondante, on simplifie le modèle en l'omettant.

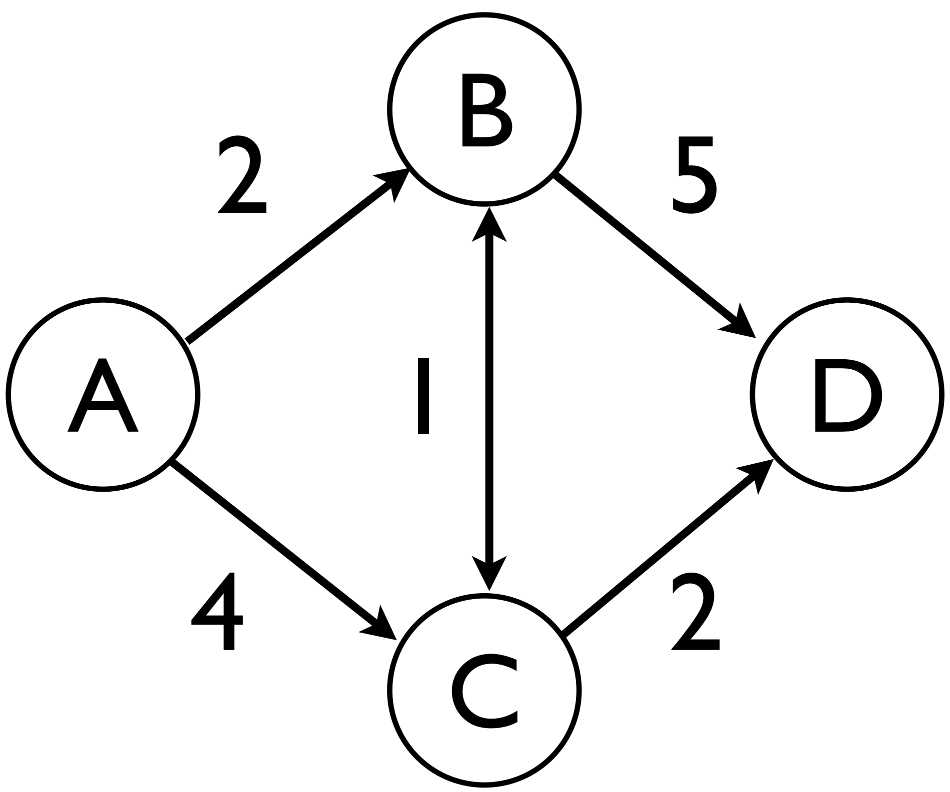
$$
\begin{array}{lcccccccccccccc}
\text{minimiser} & 2x_{ab} & + & 4x_{ac} & + & x_{bc} & + & 5x_{bd} & + & x_{cb} & + & 2x_{cd} & & \\
\text{sujet à} & x_{ab} & & & - & x_{bc} & - & x_{bd} & + & x_{cb} & & & = & 0 \\
 & & & x_{ac} & + & x_{bc} & & & - & x_{cb} & - & x_{cd} & = & 0 \\
 & & & & & & & x_{bd} & & & + & x_{cd} & = & 1 \\
 & & & & \multicolumn{7}{c}{x_{ab}, x_{ac}, x_{bc}, x_{bd}, x_{cb}, x_{cd}} & & \geq & 0
\end{array}
$$

Premier modèle

- Sous sa forme matricielle, nous obtenons ce programme linéaire.

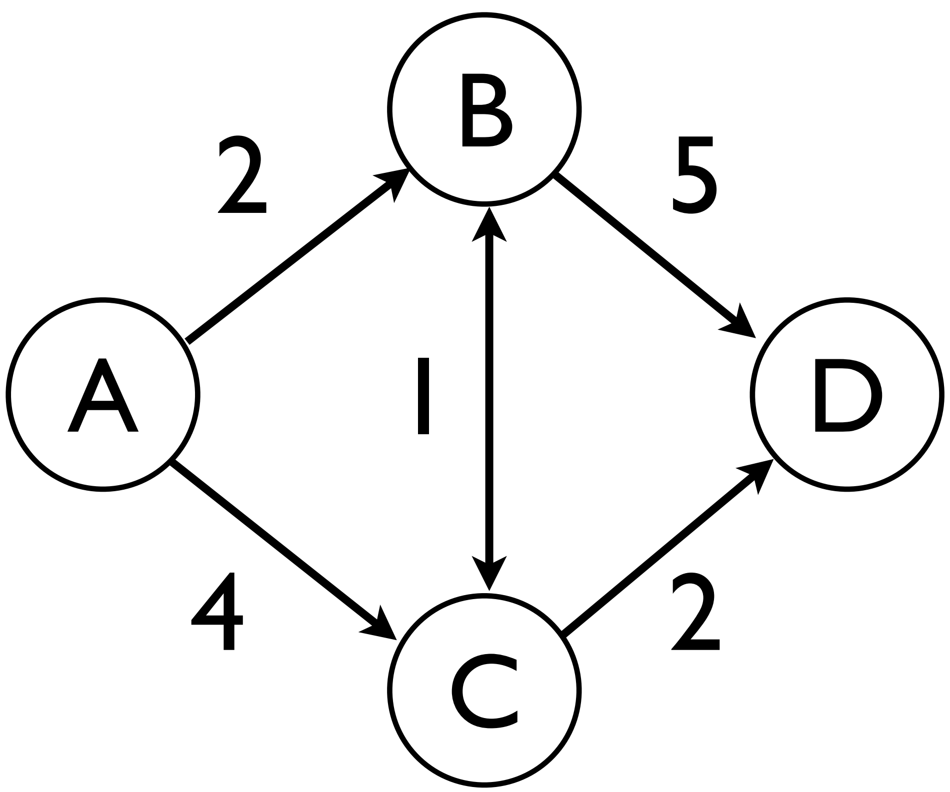
$$\begin{array}{ll}
\text{minimiser} & \begin{bmatrix} 2 & 4 & 1 & 5 & 1 & 2 \end{bmatrix} \vec{x} \\
\text{sujet à} & \begin{bmatrix} 1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ab} \\ x_{ac} \\ x_{bc} \\ x_{bd} \\ x_{cb} \\ x_{cd} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
& \vec{x} \geq \vec{0}
\end{array}$$

Deuxième modèle



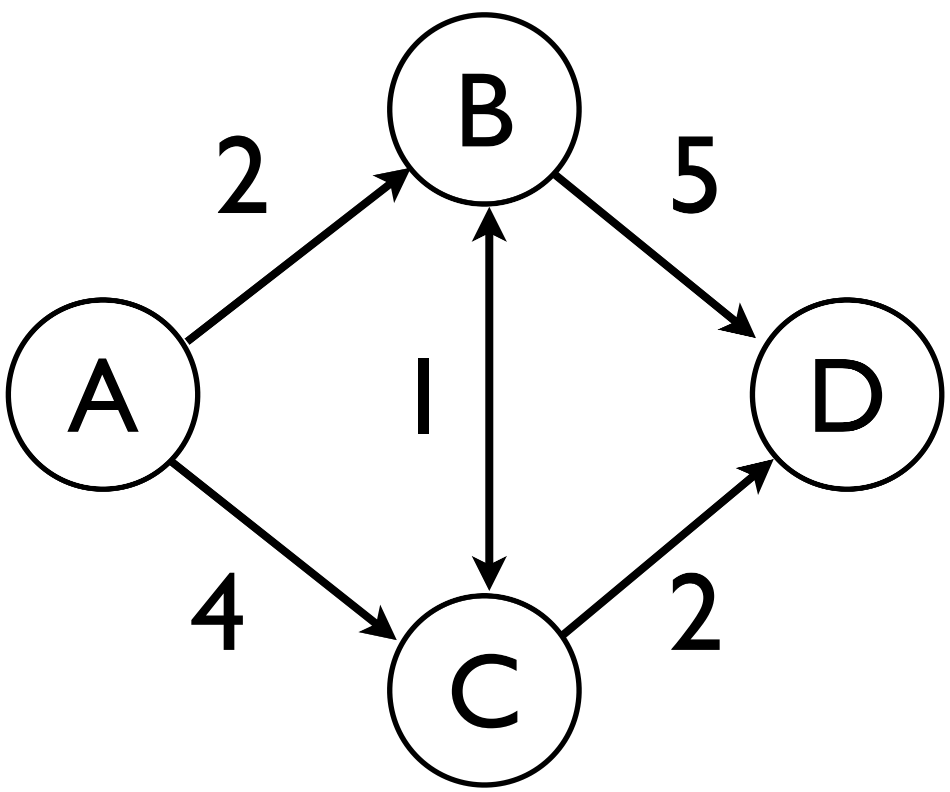
- On peut fabriquer ce graphe avec de la ficèle pour représenter les arêtes.
- Un bout de ficèle de 2 cm est attaché à un bout de ficèle de 4 cm par un noeud étiqueté A. Ces deux ficèles représentent les arêtes (A, B) et (A, C).
- Ces mêmes bouts de ficèles seraient reliés à leur autre extrémité par un troisième bout de ficèle de 1 cm formant le cycle (A, B, C, A). On procède ainsi pour les arêtes (B, D) et (C, D).

Deuxième modèle



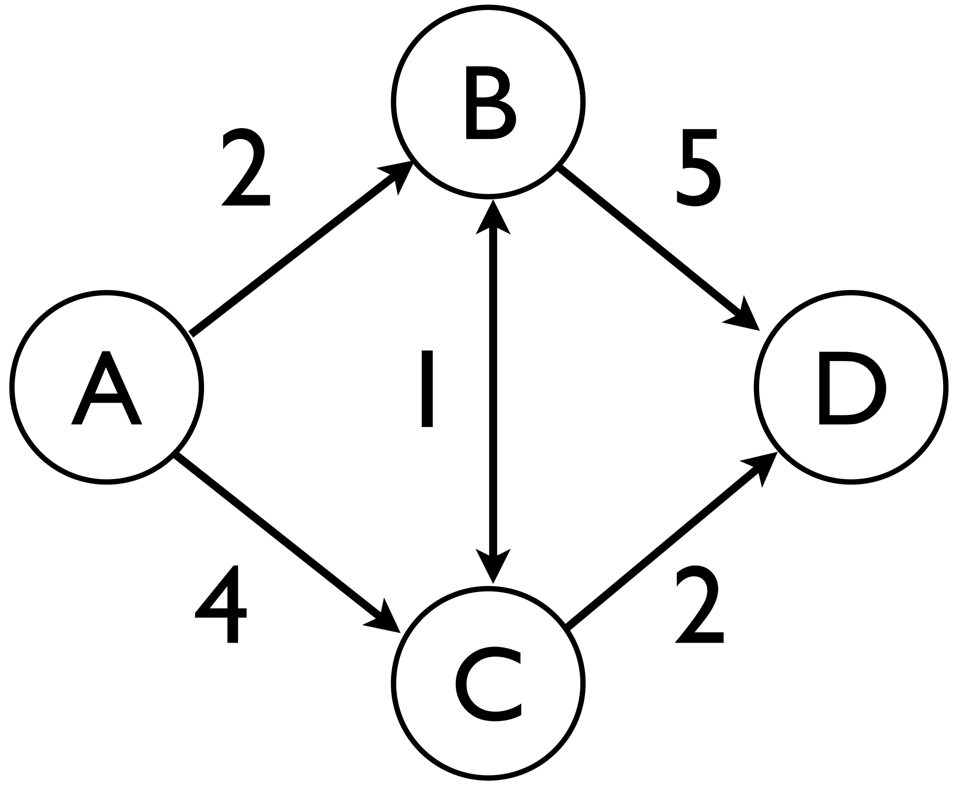
- Pour trouver le plus court chemin entre A et D, on pince le noeud A avec la main gauche et le noeud D avec la main droite.
- On tire ensuite sur les noeuds A et D en directions opposées.
- À un certain point, il ne sera plus possible d'éloigner les noeuds A et D car les ficèles seront tendues.
- L'ensemble des ficèles tendues forme le chemin le plus court entre A et D.

Deuxième modèle



- Pour modéliser ce problème, on utilise une variable par noeud qui représente la position du noeud.
- On a ensuite une contrainte par arête afin d'empêcher les deux noeuds d'êtres à des positions plus éloignées de ce que leur permet l'arête qui les relie.

Deuxième modèle



$$
\begin{array}{lrrrrrrrrr}
\text{maximiser} & -y_a & + & & & & + & y_d & & \\
\text{sujet à} & -y_a & + & y_b & & & & & \leq & 2 \\
& -y_a & & & + & y_c & & & \leq & 4 \\
& & & -y_b & + & y_c & & & \leq & 1 \\
& & & -y_b & & & + & y_d & \leq & 5 \\
& & & y_b & - & y_c & & & \leq & 1 \\
& & & & & -y_c & + & y_d & \leq & 2
\end{array}
$$

Deuxième modèle

- Notez que si le vecteur $[y_a, y_b, y_c, y_d]$ est une solution, le vecteur $[y_a + k, y_b + k, y_c + k, y_d + k]$ est aussi une solution. Cette deuxième solution correspond à un déplacement de tous les noeuds de k cm vers la droite.

- Afin d'éviter cette infinité de solutions, on peut fixer la position du noeud A à $y_a = 0$. Nous obtenons alors ce modèle.

$$
\begin{array}{lrcrcrcl}
\text{maximiser} & & & & & y_d & & \\
\text{sujet à} & y_b & & & & & \leq & 2 \\
 & & + & y_c & & & \leq & 4 \\
 & -y_b & + & y_c & & & \leq & 1 \\
 & -y_b & & & + & y_d & \leq & 5 \\
 & y_b & - & y_c & & & \leq & 1 \\
 & & & -y_c & + & y_d & \leq & 2
\end{array}
$$

Deuxième modèle

- Sous sa forme matricielle, nous obtenons ce programme linéaire.

$$
\begin{array}{lccc}
\text{maximiser} & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \vec{y} & \\
\text{sujet à} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} & \begin{bmatrix} y_b \\ y_c \\ y_d \end{bmatrix} & \leq \begin{bmatrix} 2 \\ 4 \\ 1 \\ 5 \\ 1 \\ 2 \end{bmatrix}
\end{array}
$$

Comparaison

- **Question**: Quelles sont les ressemblances entre les deux modèles?

$$
\begin{array}{lll}
\text{minimiser} & \begin{bmatrix}2 & 4 & 1 & 5 & 1 & 2\end{bmatrix}\vec{x} & \\
\text{sujet à} & \begin{bmatrix}1 & 0 & -1 & -1 & 1 & 0\\ 0 & 1 & 1 & 0 & -1 & -1\\ 0 & 0 & 0 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}x_{ab}\\ x_{ac}\\ x_{bc}\\ x_{bd}\\ x_{cb}\\ x_{cd}\end{bmatrix} = \begin{bmatrix}0\\0\\1\end{bmatrix} & \\
 & \vec{x} \geq \vec{0} &
\end{array}
$$

$$
\begin{array}{ll}
\text{maximiser} & \begin{bmatrix}0 & 0 & 1\end{bmatrix}\vec{y} \\
\text{sujet à} & \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1 & 1 & 0\\ -1 & 0 & 1\\ 1 & -1 & 0\\ 0 & -1 & 1\end{bmatrix}\begin{bmatrix}y_b\\ y_c\\ y_d\end{bmatrix} \leq \begin{bmatrix}2\\4\\1\\5\\1\\2\end{bmatrix}
\end{array}
$$

Comparaison

- **Réponses**: Pour obtenir la matrice de coefficients du deuxième problème, il suffit de transposer la matrice de coefficients du premier problème.
- Il y a autant de variables dans un problème qu'il y a de contraintes dans l'autre problème.
- Le vecteur de la fonction objectif du premier problème est le vecteur à droite des inégalité du deuxième problème.
- Le vecteur de la fonction objectif du deuxième problème est le vecteur à droite des égalités du premier problème.
- Le premier problème est une minimisation alors que le deuxième est une maximisation.
- Le premier problème a des égalités alors que le deuxième a des inégalités.
- Les variables du premier problème doivent être positives ou nulles alors que celles du deuxième problème sont libres.

Dualité

- Ces deux programmes linéaires sont équivalents.

$$\begin{array}{lrcl} \text{minimiser} & c^T x & & \\ \text{sujet à} & Ax & = & b \\ & x & \geq & 0 \end{array} \iff \begin{array}{lrcl} \text{maximiser} & b^T y & & \\ \text{sujet à} & A^T y & \leq & c \\ & y & \gtrless & 0 \end{array}$$

- Le premier problème s'appelle le problème **primal** alors que le deuxième s'appelle le problème **dual**.
- Nous allons faire la démonstration de l'équivalence entre le primal et le dual.

Comparaison des solutions réalisables

- Soit x une solution réalisable du primal et soit y une solution réalisable du dual. Ces solutions n'ont pas à être optimales mais seulement réalisables.
- La solution du primal satisfait Ax = b. La solution du dual satisfait $A^T y \leqslant c$.
- Par manipulation algébrique, nous prouvons que la valeur objective du primal est toujours plus grande que celle du dual.

$$
\begin{array}{rll}
A^T y & \leq c & \text{Solution du dual} \\
y^T A & \leq c^T & \text{Application de la transposée} \\
y^T A x & \leq c^T x & \text{Multiplication par } x \text{ des deux côtés} \\
y^T b & \leq c^T x & \text{Solution du primal} \\
b^T y & \leq c^T x & \text{Propriété du produit scalaire}
\end{array}
$$

Conséquence

- **Question:** Qu'arrive-t-il si le problème primal n'est pas borné et que $c^T x$ tend vers $-\infty$?
- **Réponse:** Le problème dual n'a pas de solution.
- L'inverse est aussi vrai. Si le problème dual n'est pas borné, alors le primal n'a pas de solution.
- **Exemple:** Dans le problème du chemin le plus court, si on peut éloigner les noeuds A et D à une distance infinie sans qu'aucune ficelle ne devienne tendue, c'est qu'il n'y a pas de chemin entre A et D. Le problème primal n'est donc pas réalisable.

La matrice B

- Soit x une solution réalisable de base qui est optimale.
- Soit B une matrice formée des colonnes de la matrice A associées aux variables de la base.
- Le produit $B^{-1}A$ est la matrice des coefficients du tableau dans son état final.

La matrice B (exemple)

$$\text{minimiser} \quad \begin{bmatrix}2 & 4 & 1 & 5 & 1 & 2\end{bmatrix} \vec{x}$$

$$\text{sujet à} \quad \begin{bmatrix}1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1\end{bmatrix} \begin{bmatrix}x_{ab} \\ x_{ac} \\ x_{bc} \\ x_{bd} \\ x_{cb} \\ x_{cd}\end{bmatrix} = \begin{bmatrix}0 \\ 0 \\ 1\end{bmatrix}$$

$$\vec{x} \geq \vec{0}$$

0	1	**0**	2	2	**0**	5
0	1	**1**	1	-1	**0**	1
1	1	**0**	0	0	**0**	1
0	0	**0**	1	0	**1**	1

- À gauche, le programme linéaire. À droite, le tableau à l'état optimal.

$$B = \begin{bmatrix}-1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 1\end{bmatrix}$$

$$B^{-1}A = \begin{bmatrix}0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1\end{bmatrix} \begin{bmatrix}1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}0 & 1 & 1 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1\end{bmatrix}$$

Le vecteur C_B

- Soit C_B le vecteur contenant le coût associé aux variables de la base.

$$\begin{aligned} \text{minimiser} \quad & \begin{bmatrix} 2 & 4 & 1 & 5 & 1 & 2 \end{bmatrix} \vec{x} \\ \text{sujet à} \quad & \begin{bmatrix} 1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ab} \\ x_{ac} \\ x_{bc} \\ x_{bd} \\ x_{cb} \\ x_{cd} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ & \vec{x} \geq \vec{0} \end{aligned}$$

0	1	**0**	2	2	**0**	5
0	1	**1**	1	-1	**0**	1
1	1	**0**	0	0	**0**	1
0	0	**0**	1	0	**1**	1

- À gauche, le programme linéaire. À droite, le tableau à l'état optimal.
- Nous avons C_B = [1, 2, 2].
- Le vecteur de coût dans le tableau est donné par $c^T - c_B^T B^{-1} A$

Le vecteur C_B

- Dans notre exemple, nous avons les coûts suivants dans le tableau.

$$c^T - c_B^T B^{-1} A = \begin{bmatrix} 2 & 4 & 1 & 5 & 1 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 4 & 1 & 5 & 1 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 2 & 2 & 0 \end{bmatrix}$$

Solution du dual

- Nous démontrons que si le problème primal a une solution *x*, alors le problème dual a aussi une solution.

- Nous avons vu que les coûts dans le tableau optimal sont donnés par le vecteur $c^T - c_B^T B^{-1} A$. Or, à l'optimalité, ces coûts sont non négatifs.

$$c^T - c_B^T B^{-1} A \geq 0$$

- Soit $y^T = c_B^T B^{-1}$. Alors *y* est une solution réalisable du dual.

$$c^T - y^T A \geq 0$$

$$y^T A \leq c^T$$

$$A^T y \leq c$$

Comparaison des solutions optimales

- Nous démontrons que si *x* est une solution optimale du primal, il existe une solution optimale au dual qui a la même valeur objective.
- La valeur objective du primal $c^T x$ n'est définie que par les variables non nulles, c'est-à-dire les variables de la base: $c^T x = c_B^T x_B$.
- Les variables non nulles ont leur valeur sur la dernière colonne du tableau optimal: $x_B = B^{-1} b$.
- Lorsque x est optimal, nous avons vu que $y^T = c_B^T B^{-1}$ est une solution réalisable pour le dual.
- Par substitutions, nous démontrons que les deux solutions ont la même valeur objective.

$$c^T x = c_B^T x_B = c_B^T B^{-1} b = y^T b = b^T y$$

Comparaison des solutions (résumé)

- Soit x une solution réalisable du primal et y une solution réalisable du dual.
- Nous avons l'inégalité suivante.

$$b^T y \leq c^T x$$

- Lorsque les deux solutions sont optimales, nous avons l'égalité.

$$b^T y = c^T x$$

Généralisation

- Nous avons vu que si le primal est un problème de minimisation sous contraintes d'égalité avec des variables non négatives, le dual est un problème de maximisation sous contraintes d'inégalités avec des variables libres.
- Ce résultat se généralise pour d'autres combinaisons.
- De façon générale:
    - Une équation dans le primal se traduit par une variable libre dans le dual.
    - Une inégalité dans le primal devient une variable non négative dans le dual.
    - Une variable libre dans le primal devient une équation dans le dual.
    - Une variable non négative dans le primal devient une inégalité dans le dual.
    - Une minimisation dans le primal devient une maximisation dans le dual.

Généralisation

- Soit E l'ensemble des indices de contraintes d'égalité dans le primal.
- Soit L l'ensemble des indices des variables libres dans le primal.
- Soit A[i] la ième rangée de la matrice A.
- Le primal (à gauche) est équivalent au dual (à droite).

$$
\begin{array}{rrclc}
 & \text{minimiser } c^T x & & & \\
\text{sujet à} & A[i]x & = & b_i & i \in E \\
 & A[i]x & \geq & b_i & i \notin E \\
 & x_j & \gtrless & 0 & j \in L \\
 & x_j & \geq & 0 & j \notin L
\end{array}
\qquad
\begin{array}{rrcl}
 & \text{maximiser } b^T y & & \\
\text{sujet à} & y_i & \gtrless & 0 \\
 & y_i & \geq & 0 \\
 & A^T[j]y & = & c_j \\
 & A^T[j]y & \leq & c_j
\end{array}
$$

Exercice

- Trouvez le dual de ce programme linéaire.

$$
\begin{array}{lrcrcrcl}
\text{minimiser} & x_1 & - & 3x_2 & + & 2x_3 & & \\
\text{sujet à} & 2x_1 & + & x_2 & - & x_3 & \geq & 5 \\
 & x_1 & - & 2x_2 & - & 4x_3 & = & 8 \\
 & & & x_1, x_3 & & & \geq & 0
\end{array}
$$

Exercice

- Trouvez le dual de ce programme linéaire.

$$
\begin{array}{lccccccc}
\text{minimiser} & x_1 & - & 3x_2 & + & 2x_3 & & \\
\text{sujet à} & 2x_1 & + & x_2 & - & x_3 & \geq & 5 \\
 & x_1 & - & 2x_2 & - & 4x_3 & = & 8 \\
 & & & x_1, x_3 & & & \geq & 0
\end{array}
$$

- Solution:

$$
\begin{array}{lcccccc}
\text{maximiser} & 5y_1 & + & 8y_2 & & \\
\text{sujet à} & 2y_1 & + & y_2 & \leq & 1 \\
 & y_1 & - & 2y_2 & = & -3 \\
 & -y_1 & - & 4y_2 & \leq & 2 \\
 & y_1 & & & \geq & 0
\end{array}
$$

Programmation en nombres entiers

- Pour plusieurs problèmes, nous voulons que les variables prennent des valeurs entières.
- La programmation en nombres entiers (Integer Programming ou IP) permet de résoudre les problèmes modélisés sous cette forme.

$$\begin{array}{rrcl} \text{maximiser} & c^T x & & \\ \text{sujet à} & Ax & \leq & b \\ & x & \in & \mathbb{N}^n \end{array}$$

- Cette formulation ne diffère d'un programme linéaire que par l'appartenance de chaque composante du vecteur x à l'ensemble des nombres naturels.

Résoudre un programme à nombres entiers

- Il existe plusieurs façons de résoudre un programme en nombres entiers.
- Nous verrons deux techniques largement utilisées.
  - La fouille dans un arbre de recherche.
  - L'ajout de coupes de Gomory.
- Ces deux approches ont un point en commun. Elles utilisent la méthode du simplexe en espérant obtenir, par chance, une solution entière. Si ce n'est pas le cas, la recherche de solution se poursuit autrement.

Branch & Bound

- On relaxe la contrainte $x \in \mathbb{N}^n$ par la contrainte $x \geq 0$.
- Si la méthode du simplexe retourne une solution entière, alors nous avons trouvé la solution optimale.
- S'il existe une variable $x_i = v$ qui n'est pas entière, alors nous créons deux branches dans l'arbre de recherche.
  - Une branche où l'on ajoute la contrainte: $x_i \leq \lfloor v \rfloor$
  - Une deuxième branche où l'on ajoute la contrainte: $x_i \geq \lceil v \rceil$

Branch & Bound

- La fouille dans l'arbre de recherche mènera à plusieurs solutions entières. Il faut évidemment trouver celle qui optimise la fonction objectif.
- Soit v^* la valeur objective de la meilleure solution trouvée (pour un problème de maximisation).
- Cette valeur v^* est donc une borne minimale pour la solution optimale puisque toute solution au problème devra avoir une valeur objective supérieure ou égale à v^*.
- Supposons qu'en résolvant le programme linéaire, nous trouvons une solution (pas nécessairement entière) dont la valeur objective est inférieure à v^*.
  - Alors il est inutile de poursuivre la recherche dans cette portion de l'arbre de recherche.
  - En effet, même en permettant des solutions réelles, on n'arrive pas à améliorer la dernière solution trouvée. Il y a donc un retour arrière.

Branch & Bound

- Une remarque:
  - En ajoutant les contraintes $x_i \leq \lfloor v \rfloor$ et $x_i \geq \lceil v \rceil$, la valeur objective du programme linéaire ne peut que diminuer.
  - En plongeant dans l'arbre de recherche, la valeur objective diminue.
  - Lorsqu'elle atteind v*, il y a un retour arrière.

Coupes de Gomory

- Une coupe est une contrainte que l'on ajoute à un programme linéaire pour éliminer des solutions non-entières tout en conservant les solutions entières du problème.
- Il existe plusieurs types de coupe. Certaines sont spécifiques à un problème.
- Les coupes de Gomory fonctionnent pour tous les programmes en nombres entiers et sont largement utilisées par les solveurs linéaires.

Coupes de Gomory

- Soit B la base de la solution trouvée par la méthode du simplexe.
- Soit x_i une variable de la base dont la valeur b_i n'est pas entière.
- Dans le tableau du simplexe, il est possible d'extraire une équation de cette forme.

$$x_i + \sum_{j \notin B} a_{i,j} x_j = b_i$$

- Dans la solution courante, les variables x_j ne sont pas dans la base et sont donc nulles.

Coupes de Gomory

- Ces deux équations sont équivalentes.

$$x_i + \sum_{j \notin B} a_{i,j} x_j = b_i$$

$$\Longleftrightarrow$$

$$x_i + \sum_{j \notin B} \lfloor a_{i,j} \rfloor x_j - \lfloor b_i \rfloor = b_i - \lfloor b_i \rfloor - \sum_{j \notin B} (a_{i,j} - \lfloor a_{i,j} \rfloor) x_j$$

- Le côté gauche de la deuxième équation est un entier pour toute solution entière.

- Le côté droit de la deuxième équation est strictement plus petit qu'un pour toute solution réalisable.

- Un entier strictement plus petit qu'un est plus petit ou égal à zéro.

$$b_i - \lfloor b_i \rfloor - \sum_{i \notin B} (a_{i,j} - \lfloor a_{i,j} \rfloor) x_j \leq 0$$

Coupes de Gomory

- L'inégalité

$$b_i - \lfloor b_i \rfloor - \sum_{i \notin B} (a_{i,j} - \lfloor a_{i,j} \rfloor) x_j \leq 0$$

- peut être réécrite sous la forme d'une égalité par l'ajout d'une variable d'écart.

$$\sum_{j \notin B} (a_{i,j} - \lfloor a_{i,j} \rfloor) x_j - s = b_i - \lfloor b_i \rfloor$$

- Cette égalité est appelée « coupe de Gomory ».

Exemple

- Lorsque nous avions voulu résoudre ce programme linéaire

$$
\begin{array}{lrcrcrrrcr}
\text{maximiser} & 2x & + & y & & & & & & \\
\text{sujet à} & x & + & y & + & s_1 & & & = & 10 \\
 & -x & + & y & + & & s_2 & & = & 3 \\
 & x & - & y & + & & & s_3 & = & 5 \\
 & x, & & y, & & s_1, & s_2, & s_3 & \geq & 0
\end{array}
$$

- nous avions obtenu ce tableau optimal.

0	0	3/2	0	1/2	17.5
0	1	1/2	0	-1/2	5/2
0	0	0	1	1	8
1	0	1/2	0	1/2	15/2

Exemple

- La première ligne du tableau indique l'équation.

$$y + 0.5s_1 - 0.5s_3 = 2.5$$

- Cette équation produit cette coupe de Gomory.

$$0.5s_1 + 0.5s_3 - s_4 = 0.5$$

Exemple

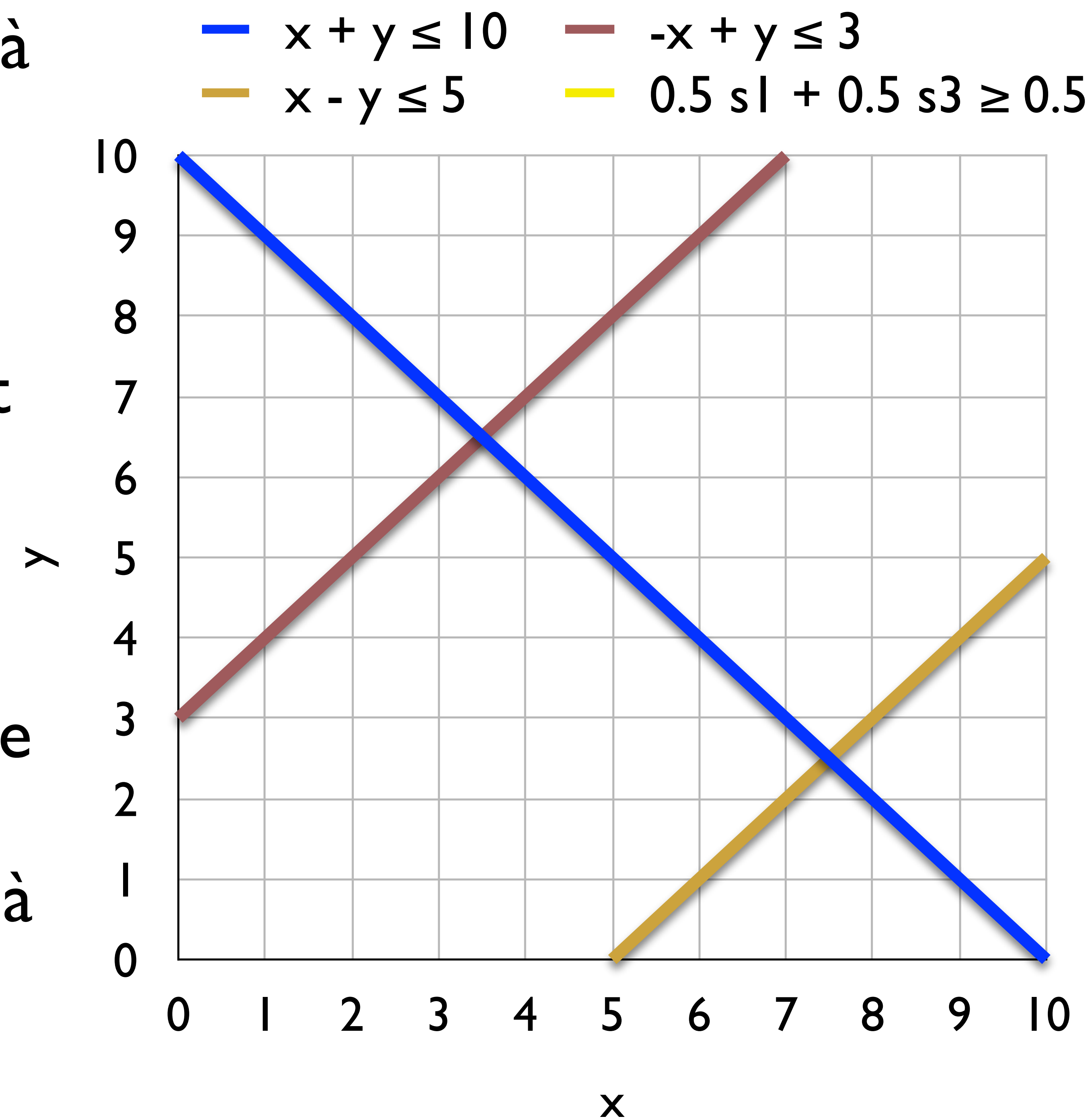
- La coupe est équivalente à cette inégalité.

$$0.5s_1 + 0.5s_3 \geq 0.5$$

- Lorsque projetée sur x et y, nous obtenons.

$$x \leq 7$$

- L'exécution de la méthode du simplexe sur le nouveau problème mène à la solution entière x = 7 et y = 3.



Exemple

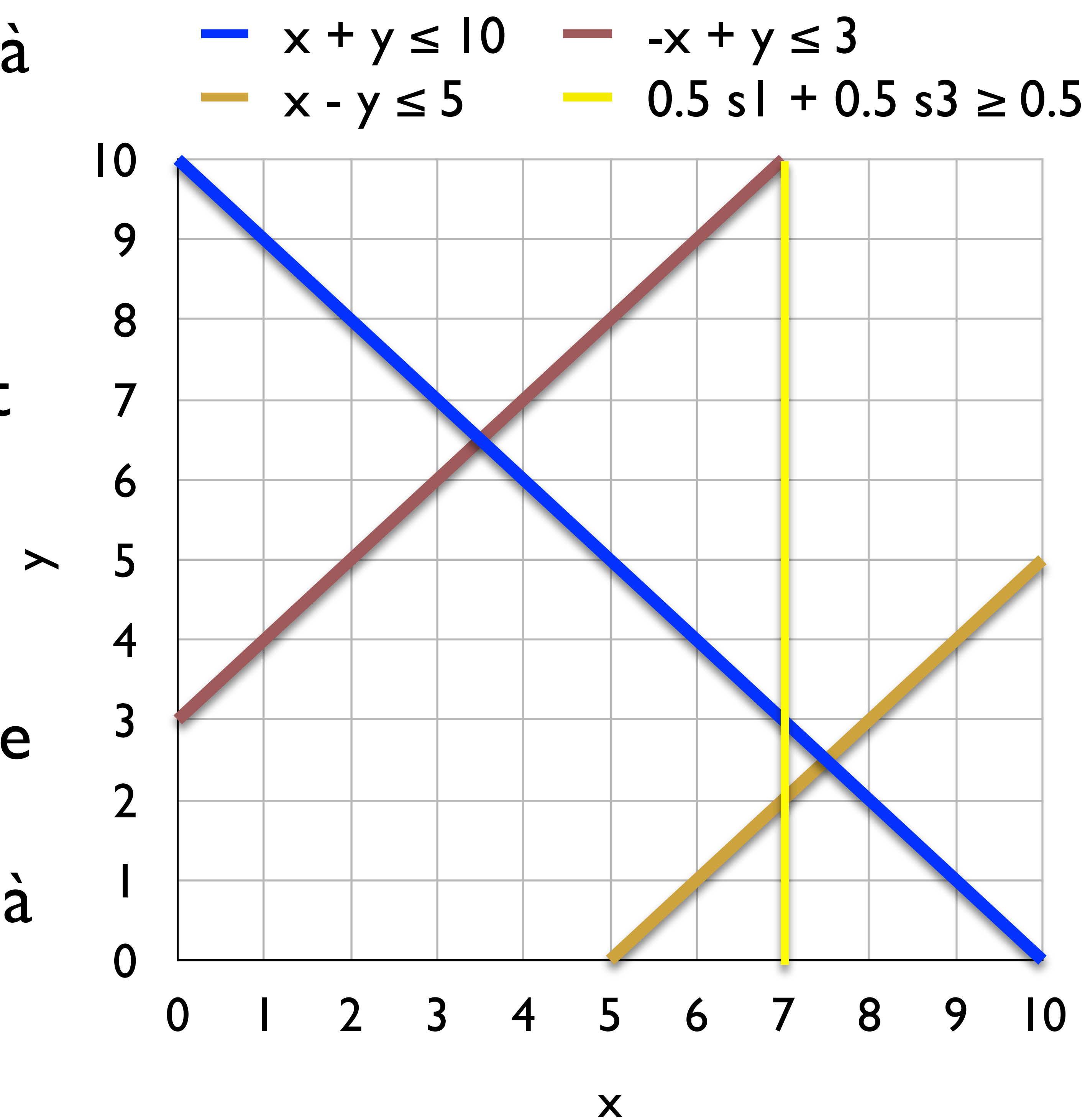
- La coupe est équivalente à cette inégalité.

$$0.5 s_1 + 0.5 s_3 \geq 0.5$$

- Lorsque projetée sur x et y, nous obtenons.

$$x \leq 7$$

- L'exécution de la méthode du simplexe sur le nouveau problème mène à la solution entière x = 7 et y = 3.



Matrices totalement unimodulaires

- Il existe des matrices A et des vecteurs b pour lesquels l'algorithme du simplexe retourne nécessairement une solution entière.
- C'est le cas lorsque A est une matrice totalement unimodulaire et b est un vecteur d'entiers.
- Une **matrice totalement unimodulaire** est une matrice dont chaque sous-matrice carrée a un déterminant de -1, 0, ou 1.

Exemple d'une matrice totalement unimodulaire

$$\begin{bmatrix} 1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

- En choisissant k lignes et k colonnes de cette matrice, on obtient une matrice de dimensions k x k dont la déterminant est -1, 0 ou 1.

L'inverse d'une matrice

- Il existe plusieurs façons de calculer l'inverse d'une matrice.
- Une de ces façons est de multiplier l'adjointe d'une matrice par l'inverse de son déterminant.
- La **comatrice** de la matrice carrée A, dénotée com(A), est une matrice de mêmes dimensions que A. L'élément à la position (i, j) est le déterminant de A auquel on a retiré a ligne i et la colonne j. Ce déterminant est multiplié par -1 si i + j est impair.
- L'**adjointe** d'une matrice carrée A est la transposée de la comatrice.

$$A^{-1} = \frac{1}{|A|}\mathrm{adj}(A)$$

Exemple

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 1 \\ 3 & 2 & 1 \end{bmatrix}, |A| = 8$$

$$\mathrm{com}(A) = \begin{bmatrix} \begin{vmatrix} 0 & 1 \\ 2 & 1 \end{vmatrix} & -\begin{vmatrix} 1 & 1 \\ 3 & 1 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 3 & 2 \end{vmatrix} \\ -\begin{vmatrix} 2 & 3 \\ 2 & 1 \end{vmatrix} & \begin{vmatrix} 1 & 3 \\ 3 & 1 \end{vmatrix} & -\begin{vmatrix} 1 & 2 \\ 3 & 2 \end{vmatrix} \\ \begin{vmatrix} 2 & 3 \\ 0 & 1 \end{vmatrix} & -\begin{vmatrix} 1 & 3 \\ 1 & 1 \end{vmatrix} & \begin{vmatrix} 1 & 2 \\ 1 & 0 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} -2 & 2 & 2 \\ 4 & -8 & 4 \\ 2 & 2 & -2 \end{bmatrix}$$

$$\mathrm{adj}(A) = \begin{bmatrix} -2 & 4 & 2 \\ 2 & -8 & 2 \\ 2 & 4 & -2 \end{bmatrix}$$

$$A^{-1} = \frac{1}{8} \begin{bmatrix} -2 & 4 & 2 \\ 2 & -8 & 2 \\ 2 & 4 & -2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & -1 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix}$$

Observations

- Si A est une matrice totalement unimodulaire, alors son adjointe ne contient que des -1, des 0 et des 1.
- Si A est inversible, alors son inverse ne contient que des -1, des 0 et de 1.
- L'inverse d'une matrice totalement unimodulaire est donc une matrice entière.

Programmes linéaires et matrices totalement unimodulaires

- Les valeurs des variables de base sont données par l'équation.

$$x_B = B^{-1}b$$

- La matrice B étant une sous-matrice de A. Si A est totalement unimodulaire, alors B est totalement unimodulaire et B^{-1} est entière.

- Donc si b est un vecteur d'entiers, le vecteur solution x sera aussi un vecteur d'entiers.

La matrice d'incidence d'un graphe

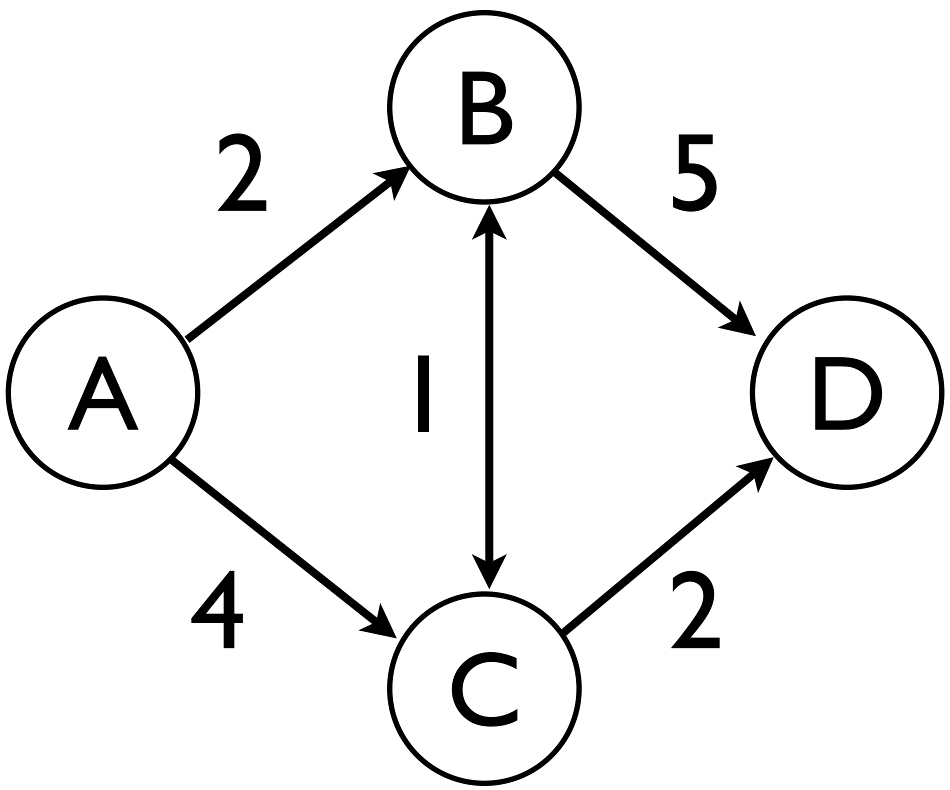
- Considérons un graphe G.
- La matrice d'incidence A du graphe G est une matrice de dimensions |V| x |E| telle que

$$A_{ij} = \begin{cases} 1 & \text{si la destination de l'arête } j \text{ est le noeud } i \\ -1 & \text{si l'origine de l'arête } j \text{ est le noeud } i \\ 0 & \text{sinon} \end{cases}$$

- **Théorème**: La matrice d'incidence d'un graphe est totalement unimodulaire.

Exemple

- Un graphe et sa matrice d'incidence



$$
\begin{array}{c c}
 & \begin{array}{cccccc} (a,b) & (a,c) & (b,c) & (b,d) & (c,b) & (c,d) \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}
\end{array}
$$

- Il s'agit de la matrice que nous avions utilisée pour modéliser le problème du chemin le plus court.
- Même en retirant la première ligne de la matrice qui est linéairement dépendante, la matrice demeure totalement unimodulaire.

L'importance des matrices totalement unimodulaires

- Les matrices totalement unimodulaires nous garantissent de trouver facilement une solution entière à notre problème.
- C'est le cas pour les problèmes de flot dont la matrice est une matrice d'incidence de graphe.
- Il est possible que la matrice d'un sous-problème ne soit pas totalement unimodulaire, mais qu'une sous-matrice le soit.

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq b$$

$$x_i, y_i \in \mathbb{N}$$

- Si A est totalement unimodulaire, mais pas B. On peut utiliser le *branch and bound* pour trouver les valeurs de y_i. Une fois ces variables fixées, il en résulte un programme à nombres entiers pouvant être résolu avec une seule exécution du simplexe.

Conclusion

- Un programme linéaire est un problème d'optimisation sous contraintes d'inégalités linéaires.
- Plusieurs problèmes industriels peuvent s'exprimer sous la forme d'un programme linéaire.
- La méthode du simplexe est un algorithme très efficace pour résoudre ces problèmes.
- La programmation linéaire offre une foule de résultats théoriques qui nous en apprend sur la nature des problèmes d'optimisation.
- Le théorème de dualité nous montre que pour tout problème de maximisation, il existe un problème équivalent de minimisation.
- Le problème dual est souvent une deuxième façon de percevoir le problème original.

Conclusion

- Les programmes en nombres entiers sont des problèmes NP-Complets.
- Nous pouvons utiliser un programme linéaire pour calculer une borne optimiste d'un programme à nombres entiers.
- Les coupes de Gomory permettent de converger vers une solution entière.
- La méthode du simplexe peut résoudre un programme à nombres entiers si la matrice des coefficients est totalement unimodulaire.

Références

- Introduction to algorithms par Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest et Clifford Stein. Chapitre 29.
- Combinatorial optimization, Algorithms and Complexity. Christos H. Papadimitriou and Kenneth Steiglitz. Chapitres 3 t 13.