Partie III

Optimisation différentiable avec contraintes linéaires

Le traitement des contraintes est simplifié si elles sont supposées linéaires. Les contraintes linéaires, d'égalité et/ou d'inégalité décrivent des ensemble convexes. La théorie de l'optimisation sous contraintes linéaires s'appuie sur l'algèbre linéaire et l'analyse convexe.

L'ère moderne d'optimisation mathématique origine des travaux de George Bernard Dantzig sur la programmation linéaire à la fin des années 1940. Le chapitre 4 en présente les résultats principaux.

Parmi les premiers développement, plusieurs travaux ont consisté à modéliser une fonction objectif non linéaire en conservant la structure de contraintes linéaires.

Chapitre 4

Programmation linéaire

Sujets du chapitre

- Observations sur la géométrie du problème.
- Condition d'optimalité.
- Déduction de l'algorithme du simplexe.
- Théorie de la dualité linéaire.
- Dégénérescence.
- Aspects numériques.

Introduction

La programmation linéaire constitue l'origine de l'optimisation mathématique moderne. Son étude a été menée par George Bernard Dantzig à partir de 1947. L'algorithme du simplexe, que nous présentons dans ce chapitre, est considéré comme un des dix algorithmes les plus importants du vingtième siècle. C'est la première fois qu'un problème avec contraintes d'inégalité a été résolu.

Dans ce chapitre, nous abordons l'étude *directe* des problèmes de minimisation de fonctions linéaires dans un domaine défini par des contraintes linéaires. Par une suite d'observations de nature géométriques ou encore algébriques, nous déduirons les conditions d'optimalité de la programmation linéaire, l'algorithme du simplexe révisé, les notions de dualité, et les variantes duales et primales-duales de l'algorithme du simplexe.

4.1 Formulation du problème

Pour simplifier l'exposé, nous considérons que le problème est formulé sous la forme dite *standard*, c'est-à-dire

$$\begin{aligned} \min\ & cx \\ \text{sujet à } & Ax = b, \\ & x \geqslant 0 \end{aligned} \tag{4.1}$$

où c et x sont des vecteurs de $\mathbb{R}^n$, b un vecteur de $\mathbb{R}^m$ et A une matrice $m \times n$. Les lignes de la matrice A sont notées $a_i, i = 1, 2, \ldots, m$ alors que ses colonnes sont notées $A_j, j = 1, 2, \ldots, n$. De plus, nous faisons l'hypothèse que la matrice A est de rang m, c'est-à-dire que ses lignes sont linéairement indépendantes. Ainsi, les contraintes définissent l'intersection d'un sous-espace de $\mathbb{R}^n$ de dimension $n - m$ avec l'orthant positif. Le vecteur c constitue le gradient de la fonction linéaire cx, et donc est un vecteur-ligne.

Nous verrons plus tard que l'hypothèse que les lignes de A sont linéairement indépendantes n'est pas très restrictive.

4.2 Solutions de base réalisables

Attardons-nous un instant sur la structure du domaine réalisable, cette intersection d'un sous-espace avec l'orthant positif. Plus particulièrement, essayons de caractériser les points d'intersection avec la frontière de l'orthant. Dans $\mathbb{R}^3$, par exemple, s'il n'y a qu'une seule contrainte (en plus des bornes qui définissent l'orthant), le domaine réalisable est l'intersection d'un plan avec l'orthant. Selon la position de ce plan, l'intersection peut être vide, ou encore constituée seulement de l'origine, ou encore d'un demi-axe, ou d'un cône, ou finalement d'un triangle. Dans tous les cas où le domaine réalisable n'est pas vide, au moins un point de la partie positive des axes canoniques en fait partie. C'est une propriété importante. Nous verrons qu'un tel point appartenant à un axe canonique est un *point extrême* du domaine réalisable, correspondant à ce que nous nommerons une *solution de base*. Complétons notre observation de $\mathbb{R}^3$ en examinant l'intersection d'un sous-espace de dimension deux (une droite) avec l'orthant. Dans ce cas, si l'intersection n'est pas vide, elle est constituée de l'origine, ou encore d'une demi-droite, ou encore d'un segment de droite. Dans ce cas, les points extrêmes appartiennent à un plan canonique (extrémités de la demi-droite ou du segment).

Ces illustrations géométriques sont à la base de la caractérisation des solutions de base en programmation linéaire. En effet, en supposant que le domaine n'est pas vide, un sous-espace de $\mathbb{R}^n$ de dimension $n-m$ intersecte au moins un hyperplan canonique de dimension m. Dans l'exemple de $\mathbb{R}^3$, le plan intersecte au moins un axe canonique (hyperplan de dimension un) alors que la droite intersecte au moins un des plans canoniques (xy, yz ou zx). Supposons donc que les composantes d'un vecteur de $\mathbb{R}^n$ sont ordonnées de telle sorte que l'hyperplan canonique qui contient un tel point d'intersection est formé des m premières composantes. Ce point est donc de la forme $(x_1, x_2, \ldots, x_m, 0, 0, \ldots, 0)^t$. Ordonnons conséquemment les colonnes de la matrice A ainsi que les composantes du vecteur c, et écrivons :

$$\begin{aligned}
\min cx &= [c_B, c_N] \begin{pmatrix} x_B \\ x_N \end{pmatrix} \\
\text{sujet à } Ax &= [B, N] \begin{pmatrix} x_B \\ x_N \end{pmatrix} = Bx_B + Nx_N = b \\
x_B &\geqslant 0 \\
x_N &= 0.
\end{aligned}$$

Maintenant, supposons que la matrice B est inversible et donc $x_B = B^{-1}b$. Les colonnes de la matrice B forment une base du sous-espace canonique de dimension m alors que le vecteur x_B constitue les coordonnées du vecteur b dans cette base. La partition $[\![B \mid N]\!]$, ou encore le vecteur $x = [x_B, 0]$ est nommé *solution de base réalisable.*

Théorème 4.2.1 *Si le problème 4.1 possède une solution réalisable, alors il possède une solution de base réalisable*

Cette notation est universelle, malgré l'ambigüité de la signification des symboles B et N, qui réfèrent à la partition des composantes de $\mathbb{R}^n$ mais qui réfèrent aussi aux sous-matrices correspondantes de la matrice A.

Avant de terminer cette introduction, revenons sur l'hypothèse que la matrice B soit inversible. Puisque nous supposons que $Bx_B = b$, même si la matrice B est singulière, le système d'équations $Bx = b$ possède tout de même une solution. En fait, si la matrice B est singulière, c'est que le système $Bx = b$ possède plusieurs solutions, dont certaines comportent des composantes nulles. C'est une manifestation du phénomène de dégénérescence, que nous traiterons plus tard.

Définition 4.2.1 *Un ensemble E est dit* affine *si* $\forall x, y \in E$, $\alpha x + (1-\alpha)y \in E$, $\forall \alpha \in \mathbb{R}$ *; l'ensemble est dit* convexe *si la même propriété est satisfaite pour* $\alpha \in [0,1]$.

Exercice 4.2.1 *[Systèmes d'équations linéaires]*

a) Montrez que si B est une matrice $m \times m$ singulière, et si le système $Bx = b$ possède une solution, alors l'ensemble des solutions constitue un ensemble affine.

b) Dans les conditions de a), montrez que si le système possède une solution telle que $x \geqslant 0$, alors il possède une solution avec une ou des composantes nulles.

La forme des contraintes (4.1) est commode pour présenter l'approche issue de l'algorithme du simplexe. Cette forme n'est aucunement restrictive, tout ensemble défini par des inégalités et éventuellement égalités linéaires pouvant se ramener à cette forme. Par exemple, supposons que l'ensemble des contraintes soit $E = \{x \in \mathbb{R}^n : \bar{A}\bar{x} \leqslant \bar{b}\}$; alors, en dédoublant $\bar{x} = \bar{x}^+ - \bar{x}^-$ avec $\bar{x}^+, \bar{x}^- \geqslant 0$ et en ajoutant des variables d'écart $s \geqslant 0$, on constate que le programme

$$\begin{aligned} &\min \bar{c}\bar{x} \\ &\text{sujet à } \bar{A}\bar{x} \leqslant \bar{b} \end{aligned} \tag{4.2}$$

peut se récrire

$$\begin{aligned} &\min \bar{c}\bar{x}^+ - \bar{c}\bar{x}^- \\ \text{sujet à } &\begin{pmatrix} \bar{A} & -\bar{A} & I \end{pmatrix} \begin{pmatrix} \bar{x}^+ \\ \bar{x}^- \\ s \end{pmatrix} = \bar{b} \\ &\bar{x}^+, \bar{x}^-, s \geqslant 0 \end{aligned}$$

qui est exactement de la forme (4.1) avec $c = (\bar{c} \;\; -\bar{c} \;\; 0)$, $A = \begin{pmatrix} \bar{A} & -\bar{A} & I \end{pmatrix}$, $b = \bar{b}$ et $x = \begin{pmatrix} \bar{x}^+ & \bar{x}^- & s \end{pmatrix}^t$.

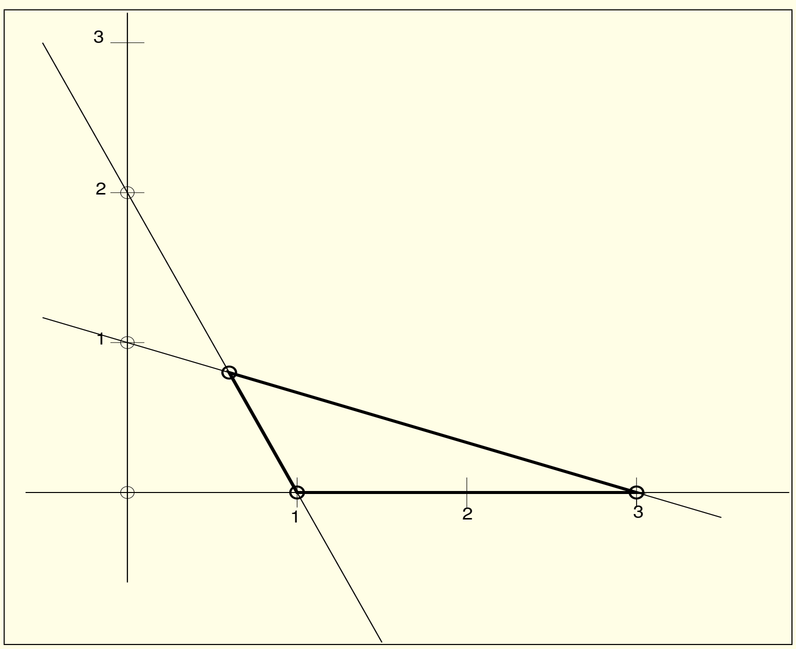
Examinons un petit exemple. Considérons le problème

$$\begin{array}{lrcrcl} \min z = & x_1 & - & x_2 & & \\ \text{sujet à} & 2x_1 & + & x_2 & \geqslant & 2 \\ & x_1 & + & 3x_2 & \leqslant & 3 \\ & x_1 & , & x_2 & \geqslant & 0 \end{array} \tag{4.3}$$

que l'on transforme sous la forme standard en

$$\begin{array}{lrcrcrcrcl} \min z = & x_1 & - & x_2 & & & & & & \\ \text{sujet à} & 2x_1 & + & x_2 & - & x_3 & & & = & 2 \\ & x_1 & + & 3x_2 & & & + & x_4 & = & 3 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0 \end{array} \tag{4.4}$$

Les solutions de base du problème (5.14) peuvent être identifiées comme les points d'intersection entre deux droites sur un graphique bi-dimensionnel représentant le problème (4.3). En effet, chacune des droites correspondant aux deux contraintes correspond à $x_3 = 0$ ou $x_4 = 0$. Le point d'intersection des deux droites possède donc deux composantes nulles, les composantes de N. L'intersection d'une des deux droites avec un axe canonique possède aussi deux composantes nulles, celle de l'autre axe (l'intersection avec l'axe x_1 est telle que $x_2 = 0$) et sa propre variable d'écart nulle. Finalement, l'origine est telle que $x_1 = x_2 = 0$ et les variables d'écart des deux droites sont non nulles puisque l'origine n'appartient pas aux droites en question.



Les intersections des droites encerclées en gras sont les points extrêmes réalisables correspondant aux solutions de base réalisables de (5.14). Les intersections encerclées en mince ne sont pas réalisables, mais sont tout de même des solutions de base de (5.14).

FIGURE 4.1 – Exemple simple bi-dimensionnel

Exercice 4.2.2 *[Équivalence de formulations]* Considérez le programme linéaire le plus général comportant des contraintes d'égalité et d'inégalité ainsi que des variables bornées et des variables libres.

$$\begin{aligned} \min\ & c_1x_1 + c_2x_2 \\ \text{sujet à}\ & A_{11}x_1 + A_{12}x_2 \leqslant b_1 \\ & A_{21}x_1 + A_{22}x_2 = b_2 \\ & x_1 \geqslant 0, \end{aligned}$$

où A est une matrice $(m_1 + m_2) \times (n_1 + n_2)$ et $c, x \in \mathbb{R}^{n_1+n_2}$ et $b \in \mathbb{R}^{m_1+m_2}$.

a) Démontrez que l'ensemble E des contraintes est un ensemble convexe.

b) Ramenez ce programme sous la forme standard du simplexe (4.1).

c) Ramenez ce programme sous la forme standard utilisant uniquement des contraintes d'inégalité (4.2)

4.3 Modèles pour la compression d'images

La programmation linéaire est un outil de modélisation devenu incontournable dans le traitement de signaux en général eten particulier d'images. L'article *Decoding by linear programming* [5] est consacré à de nouveaux résultats très influents. Nous nous limiterons à examiner le type de modèle de programmation linéaire mis en cause dans ces travaux.

En fait, un outil clef pour ces modèles est la norme ℓ_1 (voir section 1.2.1). Le contexte est de retrouver une donnée (codage d'une image selon une certaine base) x à partir de mesures y (image) sachant que les mesures et l'image originale obéissent à l'équation $y = Ax + e$ où A décrit l'encodage de l'image. Nous avons discuté de trois situations similaires à la section 1.7.1.

Voilà pour le contexte, examinons maintenant différentes variantes de modèles et les équivalences entre elles.

Le résultat remarquable de [5] est que la vraie représentation $\bar{x}$ peut-être retrouvée exactement en minimisant

$$\min_x \|Ax - y\|_1 \tag{4.5}$$

lorsque A et e satisfont à certaines hypothèses, en particulier que e comporte peu de composantes non-nulles.

Exercice 4.3.1 *[PL équivalent]* Considérez le problème $\min_x \|Ax - y\|_1$.

a) Reformulez-le sous la forme d'un programme linéaire.

b) Transformez le programme linéaire résultant sous la forme standard (4.1).

c) Transformez le programme linéaire résultant sous la forme standard (4.2).

Si l'on suppose que $A \in \mathbb{R}^{m \times n}$ avec $m > n$, on peut constituer une matrice $Z \in \mathbb{R}^{(n-m) \times m}$ dont les lignes forment une base du noyau à gauche de A défini comme $\ker_g A \overset{\text{def}}{=} \{z : zA = 0\}$, donc $ZA = 0$. Par conséquent, $\tilde{y} = Zy = Z(Ax + e) = Ze$. Après analyse, on en arrive à la formulation suivante, connue en anglais sous le nom *basis pursuit*.

$$\begin{aligned} &\min \|e\|_1 \\ &\text{sujet à } Ze = \tilde{y} \end{aligned} \tag{4.6}$$

Exercice 4.3.2 *[PL équivalent]* Considérez le problème $\min_{Ze=\tilde{y}} \|e\|_1$.

a) Reformulez-le sous la forme d'un programme linéaire.

b) Transformez le programme linéaire résultant sous la forme standard (4.1).

c) Transformez le programme linéaire résultant sous la forme standard (4.2).

4.4 Condition d'optimalité

Il est remarquable que la solution d'un programme linéaire puisse se limiter aux solutions de base réalisables. C'est l'objet du théorème dit *théorème fondamental de la programmation linéaire*.

Théorème 4.4.1 (Théorème fondamental de la programmation linéaire) *Si le problème 4.1 possède une solution réalisable optimale, alors il possède une solution de base réalisable optimale*

Nous examinons dans cette section les conditions sous lesquelles une solution $[x_B, 0]$ est optimale pour le programme linéaire (4.1). Il semble clair que pour qu'une telle solution soit optimale, il suffit de s'assurer qu'il est impossible de réduire la fonction objectif $c_B x_B + c_N 0$ en augmentant les composantes de x_N. Profitons du fait que B est inversible pour écrire

$x_B = B^{-1}(b - Nx_N)$. Cette dernière relation permet de mettre x_B en fonction de x_N. De même, la fonction objectif peut se récrire comme

$$\begin{aligned} c_B x_B + c_N x_N &= c_B B^{-1}(b - Nx_N) + c_N x_N \\ &= (c_N - c_B B^{-1}N)x_N + c_B B^{-1}b. \end{aligned}$$

En fait, il est maintenant possible de reformuler le programme (4.1) en utilisant seulement les composantes x_N, donnant le programme réduit :

$$\begin{aligned} &\min\ (c_N - c_B B^{-1}N)x_N \\ &\text{sujet à } B^{-1}(b - Nx_N) \geqslant 0 \\ &x_N \geqslant 0. \end{aligned} \tag{4.7}$$

On voit donc que si $(c_N - c_B B^{-1}N) \geqslant 0$, alors $x_N = 0$ est solution du programme (4.7), et donc x_B, x_N est solution du programme (4.1). Si de plus $x_B > 0$ et $x_N = 0$ est solution du programme (4.7), alors $(c_N - c_B B^{-1}N) \geqslant 0$. En effet, autrement, il serait possible d'augmenter légèrement une composante de x_N associée à une composante négative du vecteur $(c_N - c_B B^{-1}N)$ tout en assurant que $x_B (= B^{-1}(n - Nx_N))$ demeure non-négatif, ce qui ferait décroître la fonction objectif, contredisant l'optimalité de la solution.

Cependant, si une composante de x_B est nulle, il est concevable qu'aucune augmentation d'une certaine composante de x_N ne maintienne x_B non-négatif. Ceci est une manifestation du phénomène de dégénérescence, que nous étudierons en détail plus tard. Mentionnons seulement que dans un tel cas, la décomposition $[\![B \mid N]\!]$ des composantes positives n'est pas unique (puisqu'une valeur de x_B est nulle), et il est possible que le vecteur $(c_N - c_B B^{-1}N)$ soit non-négatif pour une autre décomposition.

En terminant cette section, présentons la notation usuelle en programmation linéaire. Le vecteur ligne $\pi \stackrel{\text{def}}{=} c_B B^{-1}$ est nommé *multiplicateurs* alors que le vecteur $c_N - \pi N$ est nommé vecteur des *coûts réduits*. Souvent, on dénote $\pi N = z$, de telle sorte que le $j^{\text{ième}}$ coût réduit s'exprime comme $c_j - z_j$.

Exercice 4.4.1 *[Programmes non bornés]* Est-il possible que le programme (4.1) ainsi que le programme $\max cx$ sous les mêmes contraintes soient tous deux non-bornés ? Justifiez.

4.5 Algorithme du simplexe

Encore une fois, nous allons déduire l'algorithme du simplexe en supposant que la condition d'optimalité énoncée plus haut n'est pas satisfaite en un certain point qui s'exprime comme $[x_B, 0] \geqslant 0$.

Examinons donc un cas où la décomposition $[x_B, 0]$ donne lieu à des coûts réduits négatifs. En d'autres termes, examinons la situation lorsque les conditions d'optimalité discutées plus haut ne sont pas satisfaites. Nous allons proposer une autre décomposition $[x_{B'}, 0]$ pour laquelle la valeur de la fonction objectif $c_{B'}x_{B'} < c_B x_B$. Tout comme dans la discussion au sujet des conditions d'optimalité, supposons que $x_B > 0$. Nous étudierons en détail le cas dégénéré plus loin.

Comme nous l'avions discuté plus haut, il est alors possible d'augmenter une composante de x_N tout en diminuant la fonction objectif. Supposons que nous ayons une composante $\bar{\jmath}$ pour laquelle $c_{\bar{\jmath}} - z_{\bar{\jmath}} < 0$. Alors, nous allons augmenter la composante $x_{\bar{\jmath}}$. S'il est possible d'augmenter $x_{\bar{\jmath}}$ sans qu'aucune composante de x_B ne diminue, alors il est possible de produire une solution réalisable de coût aussi petit que l'on veut, et le problème est non-borné inférieurement.

Autrement, il nous faut déterminer l'augmentation maximale que nous pouvons réaliser à la composante $x_{\bar{\jmath}}$ tout en assurant $x_B \geqslant 0$. Examinons l'expression que nous connaîssons, $x_B = B^{-1}(b - Nx_N)$. Puisque nous augmentons seulement la composante $x_{\bar{\jmath}}$, écrivons $x_B = B^{-1}(b - N_{\bar{\jmath}}x_{\bar{\jmath}})$. C'est donc le vecteur $B^{-1}N_{\bar{\jmath}}$ qui est responsable de l'augmentation ou de la diminution des composantes de x_B en fonction de $x_{\bar{\jmath}}$. Si $B^{-1}N_{\bar{\jmath}} \leqslant 0$, alors aucune composante de x_B ne diminue lorsque $x_{\bar{\jmath}}$ augmente, et le problème n'est pas borné. Autrement, une augmentation unitaire de $x_{\bar{\jmath}}$ provoque un changement de $-\delta x_B = -B^{-1}N_{\bar{\jmath}}$, ou encore, pour une composante i donnée, $\delta x_i = (B^{-1}N_{\bar{\jmath}})_i$. Puisque pour x_N=0 on a $x_B = B^{-1}b$, on a que la valeur de $x_{\bar{\jmath}}$ qui annule x_i est précisément $\frac{x_i}{\delta x_i}$. On garanti donc que x_B demeure non-négatif en posant

$$x_{\bar{\jmath}} = \min_{i:\delta x_i > 0}\left(\frac{x_i}{\delta x_i}\right). \tag{4.8}$$

Après ce changement, la partition $[\![B \mid N]\!]$ est modifiée en ajoutant la variable $\bar{\jmath}$ à B et en enlevant de B (pour l'ajouter dans N) un des x_i qui devient zéro, c'est-à-dire un qui atteint le $\min_{i:\delta x_i>0}(\frac{x_i}{\delta x_i})$. Alors, on passe de la partition $[\![B \mid N]\!]$ à la nouvelle partition $[\![B' \mid N']\!]$ définie comme suit : $B' = B \cup \{\bar{\jmath}\}\backslash\{\bar{\imath}\}$ et $N' = N \cup \{\bar{\imath}\}\backslash\{\bar{\jmath}\}$, où $\bar{\imath}$ est un des indices qui atteignent le $\min_{i:\delta x_i>0}(\frac{x_i}{\delta x_i})$. Nous supposons pour l'instant que ce min n'est atteint que pour un seul indice $\bar{\imath}$ et de plus que $x_{\bar{\imath}} > 0$. Lorsque ce n'est pas le cas, c'est que le problème comporte de la dégénérescence, et ce phénomène sera étudié plus tard.

Nous examinerons plus tard diverses techniques permettant de *mettre-à-jour* l'information que représente la matrice B^{-1} plutôt que de recalculer celle-ci à chaque modification de la partition $[\![B \mid N]\!]$. Pour l'instant, contentons-nous d'observer que si on sous la main une représentation de B^{-1} alors $B^{-1}(B, N_{\bar{\jmath}}) = (I, \delta x)$, donc, dans l'ordre des variables de la nouvelle partition, cette dernière matrice devient

$$B^{-1}(B_1, \ldots, B_{i-1}, N_{\bar{\jmath}}, B_{i+1}, \ldots, B_m, B_i) = (e_1, \ldots, e_{i-1}, \delta x, e_{i+1}, \ldots, e_m, e_i)$$

et en utilisant des opérations élémentaires sur les matrices, il est possible de transformer cette dernière en $(I, N'_{\bar{\imath}})$.

Sans préciser davantage comment s'effectuent les calculs, nous pouvons énoncer l'algorithme dit du simplexe révisé. Si le choix de la variable sortante est unique de même que $x_B > 0$, il est clair que cet algorithme termine en un nombre fini de calculs car le nombre de décomposition $[\![B \mid N]\!]$ est grand, mais fini et à chaque nouvelle décomposition examinée, la fonction objectif décroît, empêchant d'en examiner aucune plus d'une fois. Cependant, nous verrons plus tard qu'il faut préciser davantage l'algorithme pour garantir sa terminaison pour des problèmes dégénérés.

Remarquons ici que l'algorithme du simplexe original (non révisé) consiste à manipuler un *tableau* donné par la matrice $\bar{A} = B^{-1}[b, B, N] = [x_B, I, B^{-1}N]$ en calculant explicitement toute la matrice transformée. Le tableau comporte également une "zéroième" ligne $c - \pi A$ qui vaut zéro dans les colonnes B. En traitant la dernière colonne comme une colonne de A, on peut utiliser l'élément de la ligne zéro en dernière colonne pour y inscrire $-\pi b$.

La version révisée de l'algorithme présentée ici consiste à calculer au besoin seulement la colonne $B^{-1}N_{\bar{\jmath}}$ à partir d'une représentation de la matrice B^{-1}.

4.5.1 Exemple simple

À partir du problème (5.14) que nous répétons ici

$$\begin{array}{lrcrcrcrcl} \min z = & x_1 & - & x_2 & & & & & & \\ \text{sujet à} & 2x_1 & + & x_2 & - & x_3 & & & = & 2 \\ & x_1 & + & 3x_2 & & & + & x_4 & = & 3 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0 \end{array}$$

on obtient un tableau de départ

x_1	x_2	x_3	x_4	
1	−1	0	0	0
2	1	−1	0	2
1	3	0	1	3

(4.9)

Maintenant, rien d'évident pour obtenir une partition $[\![B \mid N]\!]$ telle que $x_B = B^{-1}b \geqslant 0$. Nous verrons aux sections 4.8 et 4.9 des techniques systématique pour y parvenir. Pour l'instant, remarquons par inspection de l'exemple très simple que $x = (1\ 0\ 0\ 2)^t$ est une solution réalisable et $B = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}$. Calculons donc $B^{-1}A = \begin{pmatrix} 1 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{5}{2} & \frac{1}{2} & 1 \end{pmatrix}$. $x_B = B^{-1}b = \binom{1}{2}$ et $\pi = c_B B^{-1} = (\frac{1}{2}\ 0)$ ce qui nous donne $c - \pi A = (0\ \ -\frac{3}{2}\ \frac{1}{2}\ 0)$. Enfin, en traitant la colonne b comme une colonne de A, on retrouve $0 - \pi b = -1$, ce qui nous donne sous forme de tableau réalisable

x_1	x_2	x_3	x_4	
0	−3/2	1/2	0	−1
1	1/2	−1/2	0	1
0	5/2	1/2	1	2

```
Simplexe révisé
  { Données : une partition [[B | N]] de ℝ^n, une représentation }
  { de la matrice B^{-1} ainsi qu'une solution réalisable [x_B, 0]. }
  optimal ← faux
  non_borné ← faux
  répéter
    { Production d'un vecteur de coûts réduits }
        π ← SOLUTION(πB = c_B)
        z_N ← πN
        d ← c_N − z_N
    { Choix d'une variable entrante }
    si (d > 0 ) alors optimal ← vrai
    sinon
      j̄ ← Calcule_j̄
      δx ← SOLUTION(Bδx = N_j̄)
      { Choix d'une variable sortante }
      si (δx < 0 ) alors non_borné ← vrai
      sinon
        ī ← arg min_{i:δx_i>0}(x_i / δx_i)
        { Mises-à-jour diverses }
            [[B | N]] ← [[B ∪ {j̄}\{ī} | N ∪ {ī}\{j̄}]]
            B^{-1} ← Mise_à_jour(B^{-1}, [[B | N]], ī, j̄)
  jusqu'à ( optimal ∨ non_borné)
```

En pratique, les deux résolutions de systèmes linéaires sont efficaces car la matrice B est représentées par une décomposition appropriée à la solution. Cette décomposition est donc mise à jour en même temps que la partition des composantes ; dans l'algorithme, nous avons nommé cette représentation B^{-1}, mais nous verrons qu'il s'agit plutôt d'une décomposition de la matrice $B = LU$

Algorithme 4.1: Simplexe révisé.

Ce que nous recherchons, c'est une partition $[\![B \mid N]\!]$ telle que le tableau ainsi calculé ait sa zéroième ligne et sa dernière colonne non-négatives, exception faite du coin commun à la zéroième ligne et première colonne. L'algorithme du simplexe recherche itérativement une telle partition en maintenant la dernière colonne non-négative et en progressant pour amener la première ligne non-négative également.

Appliquons-le. On remarque que la seconde colonne a un élément négatif. Nous allons échanger $x_2 \in N$ avec soit x_1, soit x_4. Comment choisir ? Utilisons le critère développé (4.8). On calcule le quotient de la dernière colonne divisé par la deuxième colonne pour les éléments positifs de la deuxième colonne. Ça donne 2 et $\frac{4}{5}$, donc on choisit la seconde ligne, correspondant à x_4. Bref, notre itération va échanger x_2 et x_4 de partition. Selon l'algorithme, il suffit de constituer la nouvelle matrice $B = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$ et refaire les calculs qui nous ont permis de construire le premier tableau. Cependant, remarquons qu'une seule colonne de B a changé. Donc, inutile de recalculer *toute* la nouvelle matrice B^{-1}, il suffit de mettre-à-jour celle qu'on a déjà. Sur le tableau, ceci peut se calculer mécaniquement comme suit. L'élément pivot est à la seconde ligne et seconde colonne. Il suffit d'effectuer des opérations élémentaires pour transformer la seconde colonne en une portion d'identité, donc que l'élément pivot vaille un et le reste de la colonne zéro. On divise la deuxième ligne par l'élément pivot et ajoute un multiple approprié de celle-ci aux autres lignes pour obtenir :

x_1	x_2	x_3	x_4	
0	0	4/5	3/5	1/5
1	0	−3/5	−1/5	3/5
0	1	1/5	2/5	4/5

(4.10)

Ce tableau est réalisable (dernière colonne non-négative) et optimal (première ligne non-négative) et la valeur optimale est $-\frac{1}{5}$ (élément première ligne et dernière colonne multiplié par -1).

Évidemment, sur des problèmes sérieux, beaucoup plus d'une itération seront nécessaires.

Exercice 4.5.1 *[Illustration graphique]* Illustrez graphiquement l'itération de l'algorithme

L'algorithme du simplexe primal passe d'un point extrême réalisable à l'autre en réduisant la fonction objectif.

Exercice 4.5.2 *[Pivots]* Pour s'exercer avec l'opération de pivot du simplexe, en commençant avec l'origine, effectuez 5 pivots pour passer successivement à tous les sommets de la figure 4.1.

Remarque 5 *Notre exemple simple bi-dimensionnel permet facilement d'énumérer tous les points extrêmes, réalisables ou non. En général, c'est une tâche très difficile de procéder à l'énumération. D'abord, le nombre de points extrêmes est très grand,* $\binom{n}{m}$ *; par exemple,* $n = 100$ *et* $m = 50$ *nous fournissent* 10^{29} *points. Ensuite, il est très difficile de trouver un ordre pour parcourir tous les points extrêmes.*

4.6 Dualité

Examinons maintenant les relations que nous avons obtenues jusqu'à date pour en déduire une propriété importante à laquelle satisfont les multiplicateurs π lorsque la base est optimale. Soit donc $[x_B, 0]$ une solution réalisable optimale associée à la décomposition $[\![B \mid N]\!]$. Nous avons la définition $\pi B = c_B$. Par ailleurs, l'optimalité de la base nous assure que $c_N - z_N \geqslant 0$, ou encore $\pi N \leqslant c_N$. Donc, les multiplicateurs associés à une base optimale satisfont à la relation $\binom{B^t}{N^t}\pi = \pi A \leqslant c$. Ceci justifie la proposition suivante.

Proposition 4.6.1 *Si* $x_B > 0$*, alors* $\pi = c_B B^{-1}$ *est une solution réalisable au programme (nommé dual)*

$$\begin{aligned} &\max \pi b \\ &\textit{sujet à } \pi A \leqslant c \end{aligned} \tag{4.11}$$

Cette proposition constitue la première moitié du théorème de dualité. En fait, puisque nous supposons qu'il existe une solution optimale $[x_B, 0]$, nous avons également

Proposition 4.6.2 π *est une solution optimale au programme (4.11)*

Preuve Nous allons démontrer cette proposition en ramenant ce dernier programme à la forme standard. Cet exercice nous permet d'utiliser des manipulations classiques et d'en introduire la notation habituelle. Pour transformer en égalités les contraintes d'inégalité, nous introduisons des *variables d'écart* $\alpha \geqslant 0$ et récrivons $\pi A \leqslant c$ sous la forme $\pi A + \alpha = c$. Pour que toutes les variables qui définissent le programme soient non-négatives, nous remplaçons les variables *libres* π par une différence de nouvelles variables non-négatives : $\pi = \beta^+ - \beta^-$ avec $\beta^+ \geqslant 0$ et $\beta^- \geqslant 0$. Ainsi, le programme (4.11) se récrit sous forme standard

$$\begin{aligned} \max\ & (\beta^+ - \beta^-)b \\ \text{sujet à }\ & (\beta^+ - \beta^-)A + \alpha = c \\ & \beta^+ \geqslant 0 \\ & \beta^- \geqslant 0 \\ & \alpha \geqslant 0. \end{aligned} \tag{4.12}$$

Si l'on décompose α selon la partition $[\![B \mid N]\!]$, on retrouve $\alpha_B = 0$ et $\alpha_N \geqslant 0$. En conséquence, une base du programme (4.12) est donnée par $\begin{pmatrix} B_1^t & -B_2^t & 0 \\ N_1^t & -N_2^t & I \end{pmatrix}$. En écrivant la définition des multiplicateurs pour le problème (4.12), on constate qu'il s'agit précisément du vecteur $[x_B, 0]$ et en écrivant les conditions d'optimalité pour le programme (4.12), on constate qu'il ne s'agit de rien d'autre que la condition $x_B \geqslant 0$, ce qui confirme que π est solution optimale pour le problème (4.12). $\square$

Corollaire 4.6.1 *[Écarts complémentaires] Ajoutons des variables d'écart s de telle sorte que les contraintes du programme (4.11) deviennent $yA + s = c$ avec $s \geqslant 0$; alors, si x et $[y, s]$ sont des solutions optimales pour les programmes (4.1) et (4.11) respectivement, $sx = 0$.*

Nous sommes maintenant en mesure de démontrer le théorème de *dualité faible*. Nous aborderons le théorème de dualité forte plus loin, où nous traiterons de la dégénérescence. La dualité faible affirme que si les programmes primal et dual ont la même valeur de fonction objectif pour un x et un π réalisables, alors le x et π en question sont des solutions optimales. Le théorème de dualité forte se contente de l'existence d'une solution optimale à l'un des deux programmes pour garantir l'existence d'une solution optimale de même valeur pour l'autre.

Théorème 4.6.3 (Dualité faible) *Considérons les problèmes (4.1) et (4.11).*

a) si les deux programmes possèdent une solution réalisable x et π, alors $cx \geqslant \pi b$; de plus, si $cx = \pi b$, alors x et π sont optimaux pour les programmes (4.1) et (4.11) respectivement.

b) si un des problèmes n'est pas borné, alors l'autre n'est pas réalisable.

c) Si un des programmes n'est pas réalisable, alors l'autre est soit non-borné, soit non-réalisable.

Preuve La partie a) découle des propositions précédentes. De même, pour la partie c), si un programme n'est pas réalisable, l'autre ne peut pas posséder de solution optimale car sinon, de la contredirait la partie a). On laisse en exercice le soin de trouver un exemple où ni le primal, ni le dual ne possèdent de solution réalisable.

Pour justifier la parties b), examinons le problème (4.11). Ses contraintes sont $\pi A \leqslant c$, et donc $\pi Ax \leqslant cx$ pour tout vecteur π réalisable pour le problème (4.11). Maintenant, en supposant que x est réalisable pour le problème (4.1), on retrouve $Ax = b$ de sorte que l'on obtient $\pi b \leqslant cx$, c'est-à-dire que toute solution réalisable pour le dual possède une valeur de fonction objectif à la valeur de l'objectif primal pour une solution réalisable quelconque. Si le primal n'est pas borné, c'est qu'aucun π ne satisfait $\pi b > -\infty$. $\square$

Les problèmes (4.1) et (4.11) sont liés entre eux par une relation de dualité. D'autres formulations de programmes linéaires conduisent à d'autres paires de problèmes duaux.

Exercice 4.6.1 *[Programmes irréalisables]* Fournissez un exemple de programme pour lequel ni le primal, ni le dual ne possède de solution réalisable.

Exercice 4.6.2 *[Dualité]* Considérez le programme linéaire suivant

$$\begin{aligned} \min\ & cx \\ \text{sujet à } & Ax \leqslant b \\ & x \geqslant 0, \end{aligned}$$

et vérifiez que son dual est

$$\begin{aligned} \max\ & yb \\ \text{sujet à } & yA \leqslant c \\ & y \leqslant 0 \end{aligned}$$

en le ramenant à la forme standard, puis en simplifiant le dual ainsi obtenu.

En fait, on peut montrer que le dual peut être obtenu selon des règles sans passer à chaque fois par la transformation en forme standard. Le tableau suivant énumère les règles. Par exemple, la forme (4.1) comporte des contraintes de type $=$ et des variables de type $\geqslant 0$. Selon la table 4.6, les variables du dual sont libres et ses contraintes sont de type $\leqslant 0$, exactement le problème (4.11). Un autre exemple, l'exercice 4.6.2 : contraintes de type $\leqslant$, donc variables duales de type $\leqslant 0$; variables de type $\geqslant 0$, donc contraintes duales de type $\leqslant$.

primal	dual
min	max
c : coût	b : coût
b : terme de droite	c : terme de droite
contrainte i du type $\geqslant$	variable i du type $\geqslant 0$
contrainte i du type $=$	variable i du type *libre*
contrainte i du type $\leqslant$	variable i du type $\leqslant 0$
variable i du type $\geqslant$	contrainte i du type $\leqslant 0$
variable i du type *libre*	contrainte i du type $=$
variable i du type $\leqslant$	contrainte i du type $\geqslant 0$

TABLE 4.1 – Règle reliant le primal au dual

Exercice 4.6.3 *[Dualité]* Considérez le programme linéaire le plus général envisageable

$$\begin{aligned} \min\ & c_1x_1 + c_2x_2 \\ \text{sujet à } & A_{11}x_1 + A_{12}x_2 \leqslant b_1 \\ & A_{21}x_1 + A_{22}x_2 = b_2 \\ & x_1 \geqslant 0, \end{aligned}$$

où A est une matrice $(m_1 + m_2) \times (n_1 + n_2)$ et $c, x \in \mathbb{R}^{n_1+n_2}$ et $b \in \mathbb{R}^{m_1+m_2}$. Caractérisez le dual de ce problème.

4.7 Algorithme dual du simplexe

Comme nous venons de le voir, avec les mêmes données, c'est-à-dire le vecteur c, la matrice A et le vecteur b, nous pouvons définir deux problèmes duaux l'un par rapport à l'autre, soit les problèmes (4.1) et (4.11). Le théorème 4.10.2 ()que nous verrons plus tard à la page 207) de dualité forte garanti que nous pouvons résoudre l'un ou l'autre de ces problèmes et en déduire la solution de l'autre. On pourrait donc appliquer l'algorithme du simplexe au problème (4.11) pour résoudre (4.1). L'algorithme dual du simplexe fait exactement ça, mais en utilisant les données telles que formulées pour le primal.

Simplexe dual révisé

{ Données : une partition $[\![B \mid N]\!]$ de $\mathbb{R}^n$, une représentation }
{ de la matrice B^{-1} ainsi qu'une solution dual-réalisable $[x_B, 0]$. }
{ telle que $\pi B = c_B$ et $c_N - \pi N \geqslant 0$. }
réalisable $\leftarrow$ faux
vide $\leftarrow$ faux
répéter
    { Choix d'une variable sortante }
    $\bar{\imath} \leftarrow \arg\min_{i \in B}(x_i)$
    si $(x_{\bar{\imath}} \geqslant 0)$ **alors** réalisable $\leftarrow$ vrai
    sinon
        { Production d'un vecteur de coûts réduits }
            $\pi \leftarrow$ SOLUTION$(\pi B = c_B)$
            $z_N \leftarrow \pi N$
            $\bar{c}_N \leftarrow c_N - z_N$
        { Production de la ligne $\bar{\imath}$ }
            $\bar{a}_{\bar{\imath}} \leftarrow (e_{\bar{\imath}}^T B^{-1}) N$
        { Choix d'une variable entrante }
        si $(\bar{a}_{\bar{\imath}} \geqslant 0)$ **alors** vide $\leftarrow$ vrai
        sinon
            $\bar{\jmath} \leftarrow \arg\min_{j \in N, \bar{c}_j > 0}(\frac{\bar{a}_{\bar{\imath}j}}{\bar{c}_j})$
            $\delta x \leftarrow$ SOLUTION$(B\delta x = N_{\bar{\jmath}})$
            $x_{\bar{\jmath}} \leftarrow \frac{x_{\bar{\imath}}}{\delta x_{\bar{\imath}}}$
            $x_B \leftarrow x_B - x_{\bar{\jmath}} \delta_x$
            { Mises-à-jour diverses }
                $[\![B \mid N]\!] \leftarrow [\![B \cup \{\bar{\jmath}\} \setminus \{\bar{\imath}\} \mid N \cup \{\bar{\imath}\} \setminus \{\bar{\jmath}\}]\!]$
                $B^{-1} \leftarrow$ Mise_à_jour$(B^{-1}, [\![B \mid N]\!], \bar{\imath}, \bar{\jmath})$
jusqu'à (réalisable $\vee$ vide)

Ici encore, les solutions et mises-à-jour de B^{-1} sont efficaces. Il est également possible de mettre à jour les vecteurs $\bar{c}_N$ plutôt que de les recalculer à partir d'un π réobtenu par le système d'équations.

Algorithme 4.2: Simplexe dual révisé.

4.7.1 Exemple simple

À partir du problème (5.14) on obtient un tableau de départ, multipliant la première ligne du tableau (4.9) par -1,

x_1	x_2	x_3	x_4	
1	-1	0	0	0
-2	-1	1	0	-2
1	3	0	1	3

Maintenant, rien d'évident pour obtenir une partition $[\![B \mid N]\!]$ telle que, posant $\pi = c_B B^{-1}$, la zéroième ligne $c - \pi A$ soit non-négative. Rappelons que pour l'algorithme dual, on ne se soucie pas que $x_B \geqslant 0$, l'algorithme maintiendra $c - \pi A \geqslant 0$ et progressivement atteindra que $x_B \geqslant 0$.

Comme dans le cas de l'algorithme primal, en inspectant le problème, on découvre que $x = (0\ 1\ \ -1\ 0)^t$ est dual-réalisable. Pour ce x, on a $B = \{2, 3\}$, $N = \{1, 4\}$, la matrice $B = \begin{pmatrix} -1 & 1 \\ 3 & 0 \end{pmatrix}$ et $B^{-1}A = \begin{pmatrix} \frac{1}{3} & 1 & 0 & \frac{1}{3} \\ -\frac{5}{3} & 0 & 1 & \frac{1}{3} \end{pmatrix}$. $B^{-1}b = (1\ \ -1)^t$. $\pi = c_B B^{-1}) = (0\ \ -\frac{1}{3})$ et $c - \pi A = (\frac{4}{3}\ 0\ 0\ \frac{1}{3})$. Finalement $-\pi b = 1$ et nous avons toutes les quantités pour écrire le tableau optimal mais irréalisable

x_1	x_2	x_3	x_4	
4/3	0	0	1/3	1
1/3	1	0	1/3	1
$-5/3$	0	1	1/3	-1

Il faut rendre la dernière colonne non-négative. Il faudrait retirer x_3 de la partition B. En ce faisant, nous allons remplacer x_3 par soit x_1, soit x_4. Comment choisir ? Utilisons le critère $\arg\min_{j \in N, \bar{c}_j > 0}(\frac{\bar{a}_{\bar{\imath}j}}{\bar{c}_j})$. Il nous faut calculer le quotient de la seconde ligne sur la zéorième ligne pour les composantes de N, c'est-à-dire les colonnes un et quatre. C'est la première colonne qui gagne et notre nouvelle partition sera $B = \{1, 2\}$ et $N = \{3, 4\}$. On effectue un pivot sur l'élément de la première ligne et première colonne pour mettre-à-jour le tableau afin qu'il représente l'information reliée à la nouvelle partition. On retrouve un tableau équivalent au tableau optimal (4.10).

x_1	x_2	x_3	x_4	
0	0	4/5	3/5	1/5
0	1	1/5	2/5	4/5
1	0	$-3/5$	$-1/5$	3/5

Exercice 4.7.1 *[Illustration graphique]* Illustrez graphiquement l'itération de l'algorithme. De plus, sur la figure 4.1, les points extrêmes primal-réalisables sont identifiés ; identifiez les points extrêmes dual-réalisables.

Exercice 4.7.2 *[Dual]* Nous allons refaire le tout directement sur le dual.

a) Écrivez le dual du problème (4.3).
b) Ramenez le dual sous la forme standard de la forme (4.1).
c) Faites un graphique du dual et établissez la correspondance entre les intersections des droites du primal (figure 4.1) et celles du dual.
d) Établissez la correspondance entre les solutions de base du primal et du dual, tous deux standartisés.
e) Appliquez l'algorithme primal du simplexe à la formulation du dual standardisée.
f) Constatez que l'algorithme primal appliqué au dual est équivalent l'algorithme dual appliqué au primal.

Ces observations simples sont cependant trompeuses. Nous avons choisi un exemple bidimensionnel pour pouvoir en faire des illustrations. En général, le nombre de variables du dual n'est pas égal au nombre de variables du primal. Il faut toujours garder en tête que des illustrations nourrissent notre intuition, mais que la situation générale de dimension plus grande que deux est considérablement plus complexe et que beaucoup d'aspects ne se manifestent simplement pas en deux dimension. C'est le cas par exemple du phénomène de cyclage de l'algorithme dont nous présenterons un exemple à la section 4.10.1 qui n'apparaît pas en dimension inférieure à six.

4.8 Comment obtenir la première solution de base réalisable

Parmi les techniques usuelles pour initialiser l'algorithme du simplexe, la plus répandue consiste à résoudre un problème en *deux phases*. La première phase résout le problème

$$\begin{aligned} &\min ez \\ \text{sujet à } &Iz + Ax = b \\ &x, z \geqslant 0, \end{aligned} \tag{4.13}$$

où les variables z sont nommées variables artificielles et le vecteur $e \in \mathbb{R}^m$ est tel que $e_i = 1, i = 1, 2, \ldots, m$. En prenant le soin de multiplier par -1 les contraintes associées à un membre de droite b_i négatif, le vecteur $z = b, x = 0$ constitue une solution de base réalisable pour le problème (4.13).

Si le programme (4.1) possède une solution réalisable, alors le programme (4.13) possède une solution optimale de valeur nulle, et $z = 0$. La phase I consiste donc à appliquer l'algorithme du simplexe au problème (4.13). À sa terminaison, trois cas peuvent se produire :

1. $z = 0$, et toutes les composantes de z sont dans la partition N ;
2. $z = 0$, mais certaines composantes de z sont dans B ;
3. $z \neq 0$.

Dans le premier cas, il est possible de résoudre le programme (4.1) directement. Dans le troisième cas, le programme (4.1) ne possède pas de solution réalisable. Enfin, dans le second cas, il faut effectuer certaines opérations de pivotage pour éliminer les composantes de z de B. Cependant, pour ce faire, les critères de pivot sont simplifiées puisque notre unique but est de compléter la base partielle (éléments de B correspondant à des variables x) avec d'autres composantes x, tout en les laissant nulles. Soit donc une colonne associée à une variable de base $z_i = 0$. Il est alors possible d'effectuer un pivot sur tout élément non-nul de la ligne $\bar{a}_i$ correspondant à une variable x. Ceci a pour objet de changer la partition sans changer le point (z, x), qui constitue une solution dégénérée. Si jamais une ligne de la matrice $\bar{A}$ devenait nulle, c'est le signe que la matrice A n'est pas de rang m, et il est alors possible d'éliminer cette ligne.

4.9 Algorithme auto-dual paramétrique

Maintenant que nous avons étudié la dualité, l'algorithme dual du simplexe, ainsi que l'analyse de sensibilité, nous avons les outils pour définir une variante très intéressante de l'algorithme du simplexe. Cette variante ne nécessite pas que la solution de base initiale soit primal ou dual réalisable. L'algorithme ne nécessite donc jamais deux phases.

4.9.1 Idée générale

L'idée est la suivante. Considérons que nous avons sous la main une partition base/hors-base $[\![B \mid N]\!]$. Alors, la solution de base $\tilde{x}_B = B^{-1}b$ et $\tilde{c}_N = \pi N - c_N$ où $\pi B = c_B$. Si la solution était primal réalisable, on aurait $\tilde{x}_B \geqslant 0$ et si elle était dual-réalisable, on aurait $\tilde{c}_N \geqslant 0$. Puisque ce n'est possiblement pas le cas, ajoutons une perturbation positive $\hat{x}_B > 0$ et $\hat{c}_N > 0$ de sorte que dans le problème perturbé que nous considérons, $x_B = \tilde{x}_B + \mu\hat{x}_B$ et $\bar{c}_N = \tilde{c}_N + \mu\hat{c}_N$. Il est clair que nous pouvons choisir μ assez grand pour que la base courante soit optimale (primal-réalisable et dual-réalisable) pour le problème perturbé. L'algorithme consiste donc à ramener à zéro le paramètre μ par une suite de pivots appropriés.

Pour décrire une itération de l'algorithme, on débute en calculant

$$\mu_0^* = \min\{\mu : \tilde{c}_N + \mu\hat{c}_N \geqslant 0 \text{ et } \tilde{x}_B + \mu\hat{x}_B \geqslant 0\}. \tag{4.14}$$

Alors, la base courante est primal et dual-réalisable pour toute valeur de $\mu \geqslant \mu_0^*$. Pour $\mu = \mu^*$, soit on a un indice $j \in N$ tel que $\tilde{c}_j + \mu_0^*\hat{c}_j = 0$, soit on a un indice $i \in B$ tel que $x_i + \mu_0^*\hat{x}_i = 0$. Dans le premier cas, on sélectionne j comme variable d'entrée, et on effectue un pivot primal. Autrement, on sélectionne i comme variable de sortie, et effectue un pivot dual. Après le pivot, on aura de nouvelles valeurs de $\tilde{x}_B$, $\tilde{c}_N$, $\hat{x}_B$, $\hat{c}_N$, et la relation (4.14) nous donnera une nouvelle valeur de μ_1^*, et l'algorithme continue ainsi jusqu'à ce que $\mu_k^* < 0$. Comme alors la base courante sera optimale et réalisable pour toute valeur de $\mu_{k-1} \geqslant \mu \geqslant \mu_k^*$, en particulier, la base sera optimale pour $\mu = 0$, ce que nous voulions obtenir.

4.9.2 Exemple détaillé

Voyons cet algorithme sur un exemple.
Considérons le problème

$$\begin{array}{lrcrcl}
\min z = & x_1 & - & x_2 & & \\
\text{sujet à} & 2x_1 & + & x_2 & \geqslant & 2 \\
& x_1 & + & 3x_2 & \leqslant & 3 \\
& x_1 & , & x_2 & \geqslant & 0
\end{array}$$

On obtient un tableau de départ

x_1	x_2	x_3	x_4	
1	−1	0	0	0
−2	−1	1	0	−2
1	3	0	1	3

auquel on ajoute un paramètre μ perturbant le problème ainsi en choisissant $\hat{x}_B = e_m$ et $\hat{c}_n = e_{n-m}^t$, où e_m dénote le vecteur $(1, 1, ..., 1)^t$ de dimension m. Ici, ça adonne que c_N et x_B sont tous deux de dimension 2.

x_1	x_2	x_3	x_4	
$1+\mu$	$-1+\mu$	0	0	0
−2	−1	1	0	$-2+\mu$
1	3	0	1	$3+\mu$

La plus petite valeur de μ pour laquelle ce tableau est optimal et réalisable est $\mu^* = 2$, et le tableau est réalisable pout toute valeur de $\mu \geqslant 2$. Pour cette valeur, la variable sortante est x_3, et la variable entrante est x_2 car $\frac{-1}{-1+\mu_0^*} < \frac{-2}{1+\mu_0^*}$. On effectue le pivot, et obtient le nouveau tableau

x_1	x_2	x_3	x_4	
$3-\mu$	0	$\mu-1$	0	$-z$
2	1	-1	0	$2-\mu$
-5	0	3	1	$4\mu-3$

Pour ce nouveau tableau, la plus petite valeur de μ pour laquelle ce tableau est optimal et réalisable est $\mu_1^* = 1$; en fait, les conditions sont $2-\mu \geqslant 0$, $4\mu-3 \geqslant 0$, $3-\mu \geqslant 0$ et $\mu-1 \geqslant 0$, d'où on tire que si $\mu_1^* = 1 \leqslant \mu \leqslant 2 = \mu_0^*$, alors le tableau est optimal et réalisable. Pour cette valeur, la variable entrante est x_3, et la variable sortante est x_4. On effectue le pivot, et obtient le nouveau tableau

x_1	x_2	x_3	x_4	
$4/3+2\mu/3$	0	0	$1/3-\mu/3$	$-z$
$1/3$	1	0	$1/3$	$1+\mu/3$
$-5/3$	0	1	$1/3$	$4\mu/3-1$

Pour ce nouveau tableau, la plus petite valeur de μ pour laquelle ce tableau est optimal et réalisable est $\mu_2^* = 3/4$, car l'intervalle de μ est $\mu_2^* = 3/4 \leqslant \mu \leqslant 1 = \mu_1^*$. Pour cette valeur, la variable sortante est x_3, et la variable entrante est x_1. On effectue le pivot, et obtient le nouveau tableau

x_1	x_2	x_3	x_4	
0	0	$4/5+2\mu/5$	$3/5-\mu/5$	$-z$
0	1	$1/5$	$2/5$	$4/5+3\mu/5$
1	0	$-3/5$	$-1/5$	$3/5-4\mu/5$

Maintenant, la valeur de $\mu = 0$ rend ce tableau optimal et réalisable, et le tableau final de notre problème est donc

x_1	x_2	x_3	x_4	
0	0	$4/5$	$3/5$	$-z$
0	1	$1/5$	$2/5$	$4/5$
1	0	$-3/5$	$-1/5$	$3/5$

Exercice 4.9.1 *[Valeur de z]* Complétez le tableau pour avoir une expression de la valeur de $-z$ à chaque itération, et donc pour la solution optimale.

Exercice 4.9.2 *[Intervalle final]* Quel est l'intervalle de μ qui rend le tableau final réalisable et optimal ?

Exercice 4.9.3 *[Illustration graphique]* Illustrez graphiquement les itérations de l'algorithme

4.10 Dégénérescence

Jusqu'à maintenant, nous avons escamoté les difficultés causées par la dégénérescence en supposant systématiquement que $x_B > 0$, et que le choix de la variable sortante était unique dans l'algorithme 1.1. Penchons-nous maintenant sur le problème de la dégénérescence.

Définition 4.10.1 *[Solution de base dégénérée]*

- *Une solution de base x_B est dite* primal dégénérée *lorsque une ou plusieurs composantes de x_B sont nulles.*
- *Une solution de base x_B est dite* dual dégénérée *lorsque une ou plusieurs composantes de $c_N - \pi N$ sont nulles.*
- *Une solution de base x_B est dite* dégénérée *si elle est primal ou dual dégénérée.*

Ce phénomène peut se manifester pour le programme (4.1) tout comme pour son dual (4.11), où il s'énonce comme suit : le vecteur de coût réduit $c_N - \pi N$ associé à une solution réalisable de base $[x_B, 0]$ possède une composante nulle.

Dans le cas de dégénérescence du programme primal (4.1), il est probable que l'on puisse trouver une autre décomposition $[\![B \mid N]\!]$ décrivant la même solution réalisable x_B puisque sa composante nulle n'appartient pas naturellement aux composantes B. Dans le second cas, il est possible de modifier la partition $[\![B \mid N]\!]$ en changeant la solution réalisable, mais sans modifier la valeur de la fonction objectif. Dans chacun de ces cas, le fait de changer de partition $[\![B \mid N]\!]$ n'entraîne pas de diminution de la fonction objectif, et il faut raffiner l'algorithme pour pouvoir démontrer qu'il ne produit pas de cycle de partitions. Nous illustrerons un tel cycle à la section prochaine, et proposerons une analyse d'une règle permettant d'éviter ce phénomène de cyclage.

***Exercice 4.10.1** *[Dégénérescence et unicité]*

a) Démontrez que si aucune solution optimale du programme (4.1) n'est dégénérée, alors le programme (4.11) possède une solution unique.

b) Est-il vrai que le programme (4.11) possède toujours plus d'une solution lorsque le primal (4.1) possède une solution dégénérée ?

Simplexe auto dual paramétrique

{ Données : une partition $[\![B \mid N]\!]$ de $\mathbb{R}^n$, une représentation }
{ de la matrice B^{-1}. }
$\mu^* \leftarrow \max\left(\max_{j\in N, \hat{c}_j>0} -\frac{\tilde{c}_j}{\hat{c}_j}, \max_{i\in B, \hat{x}_i>0} -\frac{\tilde{x}_i}{\hat{x}_i}\right)$
tantque ($\mu^* > 0 \;\vee$ non_borné $\;\vee$ vide)
  si (Primal) **alors**
    $\delta x \leftarrow$ SOLUTION$(B\delta x = N_{\bar{\jmath}})$
    { Choix d'une variable sortante }
    si ($\delta x \leqslant 0$) **alors** non_borné $\leftarrow$ vrai
    sinon $\bar{\imath} \leftarrow \arg\max_{i\in B}(\frac{\delta x_i}{\tilde{x}_i + \mu^* \hat{x}_i})$
  sinon
    { Dual }
    { Production de la ligne $\bar{\imath}$ }
      $\bar{a}_{\bar{\imath}} \leftarrow (e_{\bar{\imath}}^T B^{-1})N$
    { Choix d'une variable entrante }
    si ($\bar{a}_{\bar{\imath}} \geqslant 0$) **alors** vide $\leftarrow$ vrai
    sinon
      $\bar{\jmath} \leftarrow \arg\min_{j\in N}(\frac{\bar{a}_{\bar{\imath}j}}{\tilde{c}_j + \mu^* \hat{c}_j})$
      $\delta x \leftarrow$ SOLUTION$(B\delta x = N_{\bar{\jmath}})$
  $\hat{x}_{\bar{\jmath}} \leftarrow \frac{\hat{x}_{\bar{\imath}}}{\delta x_{\bar{\imath}}};\quad \hat{x}_B \leftarrow \hat{x}_B - \hat{x}_{\bar{\jmath}}\delta_x$
  $\tilde{x}_{\bar{\jmath}} \leftarrow \frac{\tilde{x}_{\bar{\imath}}}{\delta x_{\bar{\imath}}};\quad \tilde{x}_B \leftarrow \tilde{x}_B - \tilde{x}_{\bar{\jmath}}\delta_x$
  { Mises-à-jour diverses }
    $[\![B \mid N]\!] \leftarrow [\![B \cup \{\bar{\jmath}\}\backslash\{\bar{\imath}\} \mid N \cup \{\bar{\imath}\}\backslash\{\bar{\jmath}\}]\!]$
    $B^{-1} \leftarrow$ Mise_à_jour$(B^{-1}, [\![B \mid N]\!], \bar{\imath}, \bar{\jmath})$
  { Production d'un vecteur de coûts réduits }
    $\tilde{\pi} \leftarrow$ SOLUTION$(\pi B = \tilde{c}_B)$; $\quad \hat{\pi} \leftarrow$ SOLUTION$(\pi B = \hat{c}_B)$
    $\tilde{z}_N \leftarrow \tilde{\pi}N;\quad \hat{z}_N \leftarrow \hat{\pi}N$
    $\tilde{c}_N \leftarrow c_N - \tilde{z}_N;\quad \hat{c}_N \leftarrow c_N - \hat{z}_N$
  $\mu^* \leftarrow \max\left(\max_{j\in N, \tilde{c}_j>0} -\frac{\tilde{c}_j}{\hat{c}_j}, \max_{i\in B, \hat{x}_i>0} -\frac{\tilde{x}_i}{\hat{x}_i}\right)$
$x_B \leftarrow \tilde{x}$

Ici encore, on peut mettre à jour les $\hat{c}_N$ et $\tilde{c}_N$, et bien entendu, les solutions et mises-à-jour de B^{-1} sont efficaces.

Algorithme 4.3: Simplexe auto-dual paramétrique.

Exercice 4.10.2 *[Dégénérescence]* Considérez les contraintes linéaires

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, \dots n$$
$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \dots n$$
$$x_{ij} \geqslant 0, i = 1, 2, \dots n, j = 1, 2, \dots n.$$

Trouvez un vecteur c pour lequel ce problème possède des bases dégénérée simultanément pour les programmes (4.1) et (4.11).

4.10.1 Un exemple de cyclage dans l'algorithme du simplexe

Considérons le programme

$$\begin{array}{rrrrrl} \max & 2.3x_1 & +2.15x_2 & -13.55x_3 & -0.4x_4 & \\ \text{sujet à} & 0.4x_1 & +0.2x_2 & -1.4x_3 & -0.2x_4 & \leqslant 0 \\ & -7.8x_1 & -1.4x_2 & +7.8x_3 & +0.4x_4 & \leqslant 0 \end{array} \tag{4.15}$$

En utilisant les tableaux pour représenter les itérations, après avoir ajouté des variables x_5 et x_6 qui constituent nos variables de base de départ, nous obtenons un premier tableau.

x_1	x_2	x_3	x_4	x_5	x_6	
-2.3	-2.15	13.55	0.4			0
0.4	0.2	-1.4	-0.2	1		0
-7.8	-1.4	7.8	0.4		1	0

La colonne x_1 possède le coût le plus faible (-2.3) et l'élément positif de la colonne est sur la ligne un. On effectue une opération pivot sur cet élément (0.4) et obtient le second tableau.

x_1	x_2	x_3	x_4	x_5	x_6	
	-1	5.5	-0.75	5.75		0
1	0.5	-3.5	-0.5	2.5		0
	2.5	-19.5	-3.5	19.5	1	0

Ici, la seconde colonne a le coût le plus faible et on choisit la première ligne dont le quotient est minimal (les deux quotients sont nuls) pour obtenir le troisième tableau.

x_1	x_2	x_3	x_4	x_5	x_6	
		-2.3	-2.15	13.55	0.4	0
1		0.4	0.2	-1.4	-0.2	0
	1	-7.8	-1.4	7.8	0.4	0

Maintenant, on peut observer que ce troisième tableau est un simple décalage circulaire (de deux colonnes) du premier. Par conséquent, après deux autres itérations, nous retrouverons un tableau encore décalé de deux colonnes, et finalement après un total de six itération, nous retronverons le tableau initial.

Il devrait être clair que les règles utilisées pour déterminer les variables entrantes et sortantes influencent le comportement de l'algorithme du simplexe, et que l'exemple de cyclage précédant ne vaut que pour notre règle d'entrer la variable de coût réduit minimum et de sortir la première variable (si plusieurs peuvent sortir).

Cet exemple produit un cycle parce que deux variables de base sont nulles, et selon les règles d'entrée et de sorties, aucun pivot ne change de valeur de x, mais les pivots changent uniquement la décomposition $[\![B \mid N]\!]$.

4.10.2 Règles d'anti-cyclage

Pour remédier au phénomène de cyclage de l'algorithme du simplexe, plusieurs auteurs ont présenté des règles pour choisir les variables entrantes et sortantes.

Nous présentons deux telles règles. La première conserve la même stratégie pour choisir la variable sortante, mais impose une stratégie nouvelle pour choisir la variable entrante (règle de Bland) ; son expression est remarquablement simple et elle s'applique très bien à l'algorithme du simplexe révisé tel que présenté. La seconde permet n'importe quelle stratégie pour choisir la variable entrante, mais impose un choix précis pour choisir la variable sortante (règle lexicographique) ; d'expression plus complexe, cette règle semble plus utile en pratique, malgré qu'elle s'applique plus naturellement au simplexe dit non-révisé.

Définition 4.10.2 *[Règle de Bland] La variable entrante est la première pour laquelle $c_j - z_j < 0$: $\bar{\jmath} = \min\{j : c_j - z_j < 0\}$. La variable sortante est la première à atteindre le $\min_{i:\delta x_i>0}(\frac{x_i}{\delta x_i})$: $\bar{\imath} = \min\{i : (\frac{x_i}{\delta x_i}) \leqslant (\frac{x_k}{\delta x_k}) \forall k : \delta x_k > 0\}$.*

L'autre règle que nous présentons fait intervenir la notion d'ordre lexicographique de vecteurs.

Définition 4.10.3 *[Ordre lexicographique] Un vecteur $v \in \mathbb{R}^n$ est dit* lex-positif*, noté $v \overset{L}{\geqslant} 0$ si sa première composante non-nulle est positive ; $v = 0$ est* lex-zéro*, noté $v \overset{L}{=} 0$.*

Utilisant cette définition, on peut comparer deux vecteurs par l'équivalence $v \overset{L}{\geqslant} u \iff v - u \overset{L}{\geqslant} 0$, et définir le lex-min d'un ensemble de vecteurs $\{v_i\}$ comme celui (ou ceux) pour lesquels $v_i \overset{L}{\geqslant} v_{\bar{\imath}}, \forall i$.

La règle de pivot lexicographique consiste à choisir parmi les indices i qui atteignent le $\min_{i:\delta x_i>0}(\frac{x_i}{\delta x_i})$ celle dont la ligne complète $(B^{-1}N)_i$ est lex-min de toutes les lignes telles que $\delta x_i = (B^{-1}N_{\bar{\jmath}})_i > 0$. Remarquons que l'ordre des composantes de $\mathbb{R}^n$ est important, et que malgré que nous regroupions en deux ensembles les variables, les comparaisons lexicographiques *doivent s'effectuer* selon l'ordre des variables du programme original (4.1).

Définition 4.10.4 *[Règle lexicographique d'anti-cyclage] Supposons que toutes les lignes de la matrice A soient lex-positives : $a_i \overset{L}{\geqslant} 0$. On peut choisir toute colonne $\bar{\jmath}$ telle que $c_{\bar{\jmath}} - z_{\bar{\jmath}} < 0$; la variable sortante est choisie selon le critère lex-min comme suit : soit $\bar{A} = B^{-1}[b : A]$, la matrice transformée dont les lignes sont $\bar{a}_i$. $\bar{\imath}$ satisfait*

$$\frac{\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}} \overset{L}{<} \frac{\bar{a}_i}{\bar{a}_{i\bar{\jmath}}}, \forall i : \bar{a}_{i\bar{\jmath}} > 0.$$

Remarquons que sous l'hypothèse que A est de rang m, aucune paire de lignes de A n'est proportionnelle, et donc l'indice $\bar{\imath}$ ainsi défini est unique. Remarquons également qu'il est toujours possible de démarrer l'algorithme du simplexe de sorte que les lignes de la matrice $\bar{A}$ soient lex-positives.

Exercice 4.10.3 *[Anti-cyclage]* Considérez le tableau initial de l'exemple de la section 4.10.1 ; avec chacune des deux règles d'anti-cyclage présentées à la section 4.10.2, effectuez suffisamment d'itérations de l'algorithme du simplexe pour changer de point réalisable.

Exercice 4.10.4 *[Valeur optimale]* Quelle est la valeur optimale du programme linéaire sur l'exemple de cyclage ?

Exercice 4.10.5 *[Deux phases]* Proposez une méthode, utilisant deux phases, permettant que pour chacune des deux phases, le tableau initial comporte des lignes lex-positives.

4.10.3 Convergence finie de l'algorithme du simplexe

Nous allons montrer que l'algorithme du simplexe modifié avec la règle lexicographique ne peut pas cycler.

Théorème 4.10.1 *Supposons que l'on démarre l'algorithme du simplexe avec un tableau dont les lignes sont lex-positives. En appliquant la règle lexicographique, (i) les lignes de la matrice $\bar{A} = B^{-1}[b, A]$ demeurent lex-positives et (ii) la première ligne du tableau est strictement lex-croissante.*

Preuve Montrons d'abord que les lignes demeurent lex-positives. Après un pivot, la $\bar{\imath}^{\text{ème}}$ ligne devient

$$\bar{a}'_{\bar{\imath}} = \frac{\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}}$$

avec $\bar{a}_{\bar{\imath}\bar{\jmath}} > 0$ de sorte que $\bar{a}_{\bar{\imath}}$ demeure lex-positive. Lorsque $i \neq \bar{\imath}$, distinguons deux cas.

- $\bar{a}_{i\bar{\jmath}} > 0$; dans ce cas,

$$\begin{aligned}\bar{a}'_i &= \frac{\bar{a}_i - \bar{a}_{i\bar{\jmath}}\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}} \\ &= \bar{a}_{i\bar{\jmath}}\left(\frac{\bar{a}_i}{\bar{a}_{i\bar{\jmath}}} - \frac{\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}}\right) \geqslant 0\end{aligned}$$

  par choix lexicographique.
- $\bar{a}_{i\bar{\jmath}} \leqslant 0$; dans ce cas,

$$\begin{aligned}\bar{a}'_i &= \frac{\bar{a}_i - \bar{a}_{i\bar{\jmath}}\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}} \\ &= \bar{a}_i + \frac{|\bar{a}_{i\bar{\jmath}}|\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}} \overset{L}{\geqslant} \bar{a}_i\end{aligned}$$

Enfin, examinons l'effet d'un pivot sur la "ligne 0" d'un tableau. Cette ligne est composée de $[-c_B x_B, c - c_b B^{-1}A]$. Notons cette ligne par $\bar{a}_0$. On peut montrer qu'un pivot transforme la ligne 0 en

$$\begin{aligned}\bar{a}'_0 &= \bar{a}_0 - \frac{\bar{a}_{0\bar{\jmath}}\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}} \\ &= \bar{a}_0 + \frac{|\bar{a}_{0\bar{\jmath}}|\bar{a}_{\bar{\imath}}}{\bar{a}_{\bar{\imath}\bar{\jmath}}} \overset{L}{>} \bar{a}_0\end{aligned}$$

puisque par hypothèse $\bar{a}_{0\bar{\jmath}} < 0$ (critère d'entrée) et toutes les lignes sont lex-positives. Par conséquent l'algorithme ne peut pas cycler, car il ne peut pas réexaminer deux fois une même base. □

***Exercice 4.10.6** *[Théorie]* Démontrez la formule utilisée dans la preuve du théorème 4.10.1 donnant la mise à jour de la ligne 0 après un pivot.

Le résultat précédent permet de garantir que l'algorithme du simplexe ne termine pas avec des coûts relatifs négatifs : en effet, la règle d'entrée choisit un indice de coût relatif négatif, et la règle d'anti cyclage assure que la décomposition $[\![B \mid N]\!]$ ne sera jamais revisitée, donc la seule possibilité est de terminer avec des coûts relatifs non-négatifs. Ceci démontre le théorème de dualité forte.

Théorème 4.10.2 (Dualité forte) *Si le programme (4.1) possède une solution optimale x^*, alors le programme (4.11) possède une solution optimale π^* telle que $cx^* = \pi^* b$.*

Preuve Soit x^* la solution optimale obtenue par l'algorithme du simplexe utilisant une règle d'anti-cyclage, $[\![B \mid N]\!]$ la partition optimale et $\pi^* = c_B B^{-1}$ les multiplicateurs associés. Alors, les coûts relatifs $c - \pi^* N$ sont non-négatifs. Par conséquent, π^* est une solution réalisable pour le dual. De plus, $\pi^* b = c_B B^{-1} b = c_B x^*_B = cx^*$. $\square$

4.10.4 Anti-cyclage probabiliste

Une manière simple d'éviter l'anti-cyclage est la suivante. Considérons l'algorithme primal du simplexe. Le cyclage survient dès que deux variables de base ont des quotients $\frac{\bar{b}_i}{\bar{A}_{is}}$ égaux, et que l'une d'elle est choisie comme variable sortante. Alors, les 2 deviennent nulles en même temps, une sort de la base, et l'autre y demeure, provoquant une solution dégénérée. Il suffit donc de s'assurer que jamais deux tels quotients ne se présentent pour assurer la terminaison de l'algorithme. Une manière d'y parvenir est de remplacer le vecteur b par une perturbation $b + r$, où r est un vecteur de dimension m dont les composantes sont aléatoires et *très* petites. La probabilité d'avoir alors le choix entre deux variables de sortie est nulle. Un point faible de cette technique est que le problème résolu est en fait une perturbation du problème original.

Une alternative est rendue possible par l'algorithme auto dual paramétrique. Rappelons que cet algorithme consiste à ajouter à un tableau initial un multiple μe au vecteur c et au vecteur b ; pour que le tableau initial soit optimal, il suffit que $\mu \geqslant \max(-\min_i c_i, -\min_j b_j, 0)$. Si on veut éviter que lorsque l'on réduit μ, deux composantes de $\bar{c}$ ou $\bar{b}$ deviennent nulles, il suffit de remplacer $e = (1, 1, \ldots, 1)^t$ par un vecteur dont les composantes sont aléatoires entre $1/2$ et $3/2$, disons. Remarquons que cette fois-ci, la perturbation devient nulle dès que l'algorithme atteint $\mu = 0$, et donc c'est bel et bien le problème original qui est alors résolu.

Rappelons que ainsi, l'algorithme auto dual paramétrique consistera à démarrer avec un tableau initial sous forme canonique, dans lequel on remplace le vecteur $\bar{c} = c$ par un vecteur paramétrique $\bar{c} + \mu\epsilon_c$ et le vecteur $\bar{b} = b$ par un vecteur paramétrique $\bar{b} + \mu\epsilon_b$. ϵ_c et ϵ_b sont des vecteurs de même dimension que c et b respectivement, dont les composantes sont aléatoires, uniformes entre $1/2$ et $3/2$, disons. Alors, la plus petite valeur de μ pour laquelle le tableau initial est optimal et réalisable est $\mu_0 \geqslant \max(-\min_i \frac{\bar{c}_i}{\epsilon_{c_i}}, -\min_j \frac{\bar{b}_j}{\epsilon_{b_j}}, 0)$. L'utilisation des vecteurs ϵ_c et ϵ_b assurent qu'une seule valeur de $\bar{b} + \mu_0\epsilon_b$ ou $\bar{c} + \mu_0\epsilon_c$ ne vaut 0 (avec probabilité un). Si c'est une valeur de $\bar{c} + \mu_0\epsilon_c$, on effectue un changement de base avec cette composante comme variable entrante à l'aide d'une itération de l'algorithme primal du simplexe. Autrement, c'est une valeur de $\bar{b} + \mu_0\epsilon_b$ qui est nulle, et on fait sortir la composante correspondante de la base à l'aide d'une itération de l'algorithme dual du simplexe. On trouve alors une valeur μ_1 au delà de laquelle ce nouveau tableau est réalisable et optimal. On calcule μ_1 de la même manière qu'on avait calculé μ_0, sauf que $\bar{c}$ et $\bar{b}$ ont été modifiés par le pivot. On continue jusqu'à ce que $\mu_{k+1} \leqslant 0 < \mu_k$. Alors, le tableau est réalisable et optimal pour tout $\mu \in [\mu_{k+1}, \mu_k]$, et cet intervalle contient $\mu = 0$, donc le tableau est optimal pour le problème original.

4.11 Un algorithme de décomposition

Nous présentons dans cette section un algorithme de décomposition, dû à Dantzig et Wolfe, qui introduit une technique qui s'est avérée importante et efficace nommée *génération de colonnes*. Considérons un programme de la forme

$$\begin{array}{llcll} \min & c_1x_1 & + & c_2x_2 & \\ \text{sujet à} & A_1x_1 & + & A_2x_2 & = b_0 \\ & B_1x_1 & & & = b_1 \\ & & & B_2x_2 & = b_2 \\ & x_1 & , & x_2 & \geqslant 0. \end{array} \tag{4.16}$$

Si ce n'était pas de la contrainte *couplante* $A_1x_1 + A_2x_2 = b_1$, on serait en présence de deux problèmes indépendants,

$$\begin{array}{lll} \min & c_1x_1 & \\ \text{sujet à} & B_1x_1 & = b_1 \\ & x_1 & \geqslant 0 \end{array} \tag{4.17}$$

et

$$\begin{array}{lll} \min & c_2x_2 & \\ \text{sujet à} & B_2x_2 & = b_2. \\ & x_2 & \geqslant 0 \end{array} \tag{4.18}$$

Supposons pour cette introduction que chacun des sous-problèmes (4.17) et (4.18) ont des domaines réalisables bornés. Alors, il n'est pas difficile de se convaincre intuitivement que le domaine réalisable est en fait une *combinaison convexe* des sommets du polyèdre, des solutions de base réalisables des contraintes. En effet, tout point réalisable qui ne peut pas s'exprimer comme combinaison de deux autres points réalisables est un point extrême, solution de base. Donc, un point quelconque est soit un point extrême, soit une combinaison de points réalisables. Les combinaisons de 2 points extrêmes nous donnent les arêtes du polyèdre, de trois points des faces, et ainsi de suite, engendrant le polyèdre en entier. La démonstration rigoureuse de cette propriété est délicate, aussi nous nous contenterons d'utiliser le résultat suivant.

Théorème 4.11.1 *Soit un ensemble borné décrit par des contraintes $Ax = b, x \geqslant 0$. Soient $p_1, p_2, \ldots, p_k$ l'ensemble des points extrêmes (solutions de base réalisables) de l'ensemble. Alors, $\{x : Ax = b, x \geqslant 0\} = \{x : x = \sum \alpha_i p_i, \sum \alpha_i = 1, \alpha \geqslant 0\}$.*

Par exemple, un triangle est l'ensemble des combinaisons convexes de ses sommets, et un point d'un carré peut s'exprimer comme combinaison convexe des 2 (parfois plus) triangles qui le contiennent. On voit alors que les multiples α_i qui décrivent un point d'un carré ne sont pas uniques. Ces valeurs α_i sont nommées *coordonnées barycentriques* du point. Sous certaines conditions (par exemple si le point est dans un triangle 2D, donc 3 points extrêmes), cette représentation est unique, mais en général, plusieurs (en fait , une infinité) combinaisons convexes de points extrêmes pourront décrire le même point.

Remarque 4.11.1 *Le nombre de solutions de base réalisables est énorme, beaucoup plus grand que n ou m. Par exemple, les contraintes simples $0 \leqslant x \leqslant 1$ comportent n variables, $2n$ contraintes, et 2^n sommets. pour $n = 20$, on a 40 contraintes, mais 1 048 576 sommets.*

Avec cette représentation, nous allons décrire les domaines réalisables des problèmes (4.17) et (4.18) en utilisant des combinaisons convexes de leurs points extrêmes.

$$\begin{array}{ll} x_1 & = \sum \alpha_i p_i \\ \sum \alpha_i & = 1 \\ \alpha & \geqslant 0 \end{array} \tag{4.19}$$

et

$$\begin{array}{ll} x_2 & = \sum \beta_j q_j \\ \sum \beta_j & = 1 \\ \beta & \geqslant 0. \end{array} \tag{4.20}$$

Sans les dénombrer explicitement, on suppose que les sommations sur i et j s'appliquent à tous les points extrêmes de leur polyèdre respectifs.

Maintenant, récrivons le problème original (4.16) en utilisant les variables α et β, obtenant ainsi le problème nommé *problème maître*.

$$\begin{array}{llcll} \min_{\alpha,\beta} & c_1(\sum \alpha_i p_i) & + & c_2(\sum \beta_j q_j) & \\ \text{sujet à} & A_1(\sum \alpha_i p_i) & + & A_2(\sum \beta_j q_j) & = b_0 \\ & \sum \alpha_i & & & = 1 \\ & & & \sum \beta_j & = 1 \\ & \alpha & & & \geqslant 0 \\ & & & \beta & \geqslant 0. \end{array} \tag{4.21}$$

Notre nouveau problème (4.21) a quelques $(m_1 + m_2 - 2)$ contraintes de moins que (4.16), mais comporte *beaucoup* plus de variables. L'algorithme du simplexe doit trouver une variable parmi les α_i et β_j profitable pour entrer dans la base. L'opération de pivot fera intervenir $m_0 + 2$ lignes. Pour sélectionner une variable d'entrée, il faut obtenir les coûts relatifs, donc les multiplicateurs (variables duales) du problème maître.

Les multiplicateurs seront partitionnés entre m_0 composantes associées au premier bloc de contraintes (y_0), et deux composantes scalaires (y_1, y_2) associées aux deux contraintes de somme des α et β. Introduisons les notations $\xi_i = c_1 p_i$, ξ est le vecteur de coût des variables α. Similairement, $\mu_j = c_2 q_j$, μ est le vecteur de coût des variables β. Pour les contraintes, introduisons $\Psi_i = A_1 p_i$ et $\Phi_j = A_2 q_j$. Les coûts relatifs de la colonne α_i obéit donc à la relation

$$\bar{\xi_i} = \xi_i - (y_0, y_1, y_2) \begin{pmatrix} \Psi_i \\ 1 \\ 0 \end{pmatrix} = \xi_i - y_0 \Psi_i - y_1, \tag{4.22}$$

et les coûts relatifs de la colonne β_jà la relation

$$\bar{\mu}_j = \mu_j - (y_0, y_1, y_2) \begin{pmatrix} \Phi_i \\ 0 \\ 1 \end{pmatrix} = \mu_j - y_0 \Phi_j - y_2. \tag{4.23}$$

Il faut donc trouver le minimum des $\bar{\xi_i}$, des $\bar{\mu}_j$ et prendre le minimum des 2. S'il est positif, la solution est optimale, autrement, nous avons déterminé la variable d'entrée.

Maintenant, pour trouver le minimum des $\bar{\xi_i}$, utilisons (4.22) et remarquons que y_1 et y_2 ne dépendent pas de i (ni de j), et donc il suffit de minimiser $\xi_i - y_0 \Psi_i$ et $\mu_j - y_0 \Phi_j$. En se souvenant que $\xi_i = c_1 p_i$ et $\Psi_i = A_1 p_i$, on constate qu'il faut minimiser $(c_1 - y_0 A_1)u$ pour tous les points extrêmes u du domaine réalisable $\{x_1 : B_1 x_1 = b_1, x_1 \geqslant 0\}$. Or, on peut obtenir cette minimisation sans connaître les p_i car il suffit de résoudre

$$\begin{array}{lll} \min & (c_1 - y_0 A_1) x_1 & \\ \text{sujet à} & B_1 x_1 & = b_1 \\ & x_1 & \geqslant 0, \end{array} \tag{4.24}$$

et similairement pour les variables β avec le sous-problème

$$\begin{array}{lll} \min & (c_2 - y_0 A_2) x_2 & \\ \text{sujet à} & B_2 x_2 & = b_2 \\ & x_2 & \geqslant 0. \end{array} \tag{4.25}$$

On a donc réduit le problème de choisir une variable d'entrée à la résolution de 2 sous-problèmes. Comme les points extrêmes ne sont pas explicitement énumérés, mais générés par la solution des sous-problèmes (4.24) et (4.25), on parle de génération de colonne. L'algorithme génère les colonnes pertinentes au fur et à mesure des itérations du simplexe.

4.12 Un lien avec la théorie des jeux

Dans cette section, nous examinons une situation simpliste de ce qu'on appelle un jeu matriciel. Ces objets mathématiques peuvent décrire des situations conflictuelle dans lesquelles deux instances (joueurs) choisissent des stratégies à tour de rôle. Le théorème fondamental de la théorie des jeux est une conséquence du théorème de dualité en programmation linéaire.

4.12.1 Un exemple simpliste

Nous utiliserons le jeu “roche, papier, ciseaux” (RPC ci-après) pour introduire les notions de jeux matriciels. Cette présentation est inspirée du texte de Robert Vanderbei.

Dans le jeu RPC, deux joueurs présentent leur choix de roche, papier ou ciseaux simultanément ; le papier enveloppe la roche, les ciseaux coupent le papier, et la roche émèche les ciseaux. Si les choix des joueurs sont égaux, on annule. Autrement, un des joueurs gagne $(+1)$ alors que l’autre perd (-1).

Notions de stratégie

Dans un tel jeu, on appelle *stratégie pure* le choix R, P ou C. Évidemment, aucun joueur sensé n’utiliserait une stratégie pure, car l’autre s’en apercevrait, et aurait gain de cause pour la suite des tours.

On appelle *stratégie mixte* une combinaison de stratégies pures, habituellement tirées au hasard, et la stratégie mixte se représente par les probabilités d’utiliser une stratégie pure. Dans notre exemple, une stratégie mixte est un vecteur (p_R, p_P, p_C) de probabilités de choisir R, P ou C, et donc $p_R + p_P + p_C = 1$.

Représentation matricielle

L’enjeu peut se résumer dans une matrice. Un joueur lit la matrice ligne par ligne, l’autre colonne par colonne. Pour le jeu RPC, la matrice A est la suivante :

$$\begin{array}{c|rrr} & R & P & C \\ \hline R & 0 & 1 & -1 \\ P & -1 & 0 & 1 \\ C & 1 & -1 & 0 \end{array} \tag{4.26}$$

Pour une stratégie pure donnée par joueur, par exemple $x = R = (1,0,0)^t$ et $y = C = (0,0,1)$, le résultat du jeu est $yAx = +1$, indiquant que le joueur x gagne car la roche émousse les ciseaux. Pour une stratégie mixte, $yAx = \sum_i \sum_j A_{ij} x_i y_j$ correspond à la moyenne attendue des gains.

Pour ce jeu simpliste, il semble évident qu’aucun des joueurs n’a d’avantage, qu’à long terme, les pertes et les gains vont s’équilibrer, et que cette uniformité assure que la stratégie mixte optimale est $(1/3, 1/3, 1/3)$ pour chacun des 2 joueurs.

Cependant, supposons simplement que nous modifions la matrice des gains comme suit :

$$\begin{array}{c|rrr} & R & P & C \\ \hline R & 0 & 1 & -2 \\ P & -3 & 0 & 4 \\ C & 5 & -6 & 0 \end{array} \tag{4.27}$$

Quelle est alors la stratégie optimale des 2 joueurs ? Le jeu est-il toujours équitable dans le sens que les gains et les pertes vont tendre à s'équilibrer à la longue pour les deux joueurs ? C'est à ces deux questions que nous pourrons répondre après avoir démontré le théorème fondamental de la théorie des jeux.

La matrice des gains s'interprète comme suit. Supposons que le joueur x utilise la stratégie pure R, $(1, 0, 0)^t$; $Ax = (0, -1, 1)^t$, ce qui fait que le joueur y devrait normalement utiliser la stratégie pure P $(0, 1, 0)$. Maintenant, intuitivement, pour le jeu original, la stratégie optimale pour le joueur x est $(1/3, 1/3, 1/3)^t$. $Ax = (0, 0, 0)^t$, ce qui signifie que le joueur y, étant donné le choix stratégique du joueur x, peut choisir n'importe quelle stratégie puisque son gain espéré est nul pour toute stratégie.

Examinons ce qui se passe avec la matrice (4.27) pour le choix stratégique $x = (1/3, 1/3, 1/3)^t$. $Ax = (-1/3, 1/3, -1/3)^t$ et le joueur y a à sa disposition deux stratégies pures avantageuses, et leur mélange, R, C.

En résumé, si le joueur x adopte sa stratégie, alors le joueur y obtient sa stratégie optimale par la solution du programme linéaire

$$\begin{array}{lll} \min_y & yAx & \\ \text{sujet à} & ye & = 1 \\ & y & \geqslant 0. \end{array} \tag{4.28}$$

Il est naturel ici de dénoter les stratégies du joueur x en colonne, et celles du joueur y en ligne. Ceci rappelle la dualité. On nomme *vecteur stochastique* un vecteur dont les composantes sont non-négatives, et somment à un, i.e. $ye = 1, y \geqslant 0$.

Avec la notation adoptée, le joueur y minimise ses pertes alors que le joueur x maximise ses gains. Puisque le joueur y définit sa stratégie par le programme linéaire (4.28), le joueur x devra maximiser la fonction

$$\begin{array}{lll} \max_x \min_y & yAx & \\ \text{sujet à} & ye & = 1 \\ & e^t x & = 1 \\ & x, y & \geqslant 0. \end{array} \tag{4.29}$$

Étant donnée une stratégie x, le joueur y peut toujours choisir une stratégie pure, correspondant à un point extrême, solution de base réalisable des contraintes $ye = 1, y \geqslant 0$. Soient donc e_i les stratégies pures, le joueur x doit

$$\begin{array}{lll} \max_x(\min_i & e_i^t Ax) & \\ \text{sujet à} & e^t x & = 1 \\ & x & \geqslant 0. \end{array} \tag{4.30}$$

Par économie de notation, e et e_i dénotent des vecteurs colonne de dimension appropriée.

On utilise alors le truc usuel pour les maxmin,

$$\begin{array}{lll} \max v & & \\ \text{sujet à} & v & \leqslant e_i^t A x \quad \forall i \\ & e^t x & = 1 \\ & x & \geqslant 0, \end{array} \tag{4.31}$$

ou encore

$$\begin{array}{lll} \max v & & \\ \text{sujet à} & ve & \leqslant Ax \\ & e^t x & = 1 \\ & x & \geqslant 0. \end{array} \tag{4.32}$$

Finalement ramenons le tout ainsi

$$\begin{array}{lll} \max & (0\ 1)\binom{x}{v} & \\ \text{sujet à} & \begin{pmatrix} -A & e \\ e^t & 0 \end{pmatrix}\binom{x}{v} & \begin{matrix}\leqslant\\=\end{matrix} \binom{0}{1} \\ & x & \geqslant 0, \\ & v \text{ libre.} & \end{array} \tag{4.33}$$

Évidemment, pour le joueur y, il faut résoudre

$$\begin{array}{lll} \min_y \max_x & yAx & \\ \text{sujet à} & ye & = 1 \\ & e^t x & = 1 \\ & x, y & \geqslant 0, \end{array} \tag{4.34}$$

ce qui peut se ramener à la forme :

$$\begin{array}{lll} \min & (y\ u)\binom{0}{1} & \\ \text{sujet à} & (y\ u)\begin{pmatrix} -A & e \\ e^t & 0 \end{pmatrix} & \{\ \geqslant \quad = \ \}(0\ 1) \\ & y & \geqslant 0, \\ & u \text{ libre.} & \end{array} \tag{4.35}$$

La notation par bloc avec des vecteurs ligne est un peu boiteuse, mais demeure cohérente.

Exercice 4.12.1 *[Joueur y]* Complétez les détails conduisant au problème (4.35) du joueur y.

Nous sommes maintenant en mesure d'énoncer le théorème fondamental de la théorie des jeux.

Théorème 4.12.1 *Il existe des vecteurs stochastiques x^* et y^* qui satisfont*

$$\max_x y^* Ax = \min_y yAx^*.$$

La preuve de ce théorème important découle de ce que (4.33) et (4.35) sont duaux.

Exercice 4.12.2 *[Preuve]* Fournissez les détails de la preuve

Dans notre exemple, avec la matrice (4.26), les stratégies optimales sont $x = (1/3, 1/3, 1/3)^t$ et $y = (1/3, 1/3, 1/3)$. La valeur des programmes est 0, on parle alors d'un jeu matriciel à somme nulle. Cela revient à dire qu'aucun des 2 joueurs n'est avantagé par le jeu.

Avec la matrice (4.27), les choses se compliquent un tout petit peu, évidemment. Les stratégies optimales ne sont plus les mêmes, mais plus important, la valeur du jeu devient $-16/102$, indiquant que le joueur y possède un avantage dans ce jeu.

Exercice 4.12.3 *[Jeux]* Résolvez le jeu avec la matrice (4.27), obtenez les stratégies, et vérifiez que la valeur du jeu est $-16/102$.

Exercice 4.12.4 *[Jeu simple]* Deux joueurs cachent une piece de monnaie, un 5 ou un 10 sous. Si elles sont pareilles, le joueur A empoche les 2, autrement le joueur B empoche les 2. Qui a l'avantage dans ce jeu ?

4.13 Implantation numérique de l'algorithme du simplexe

Jusqu'à maintenant, nous avons évité de discuter de l'implantation de l'opération de pivotage, permettant d'effectuer une itération de l'algorithme du simplexe de manière efficace. En général, le calcul de l'inverse d'une matrice B $m \times m$ non-singulière est une opération qui nécessite $\mathcal{O}(m^3)$ opérations. Comme nous allons le voir, il est possible d'implanter l'algorithme du simplexe en utilisant seulement $\mathcal{O}(mn)$ opérations par itération.

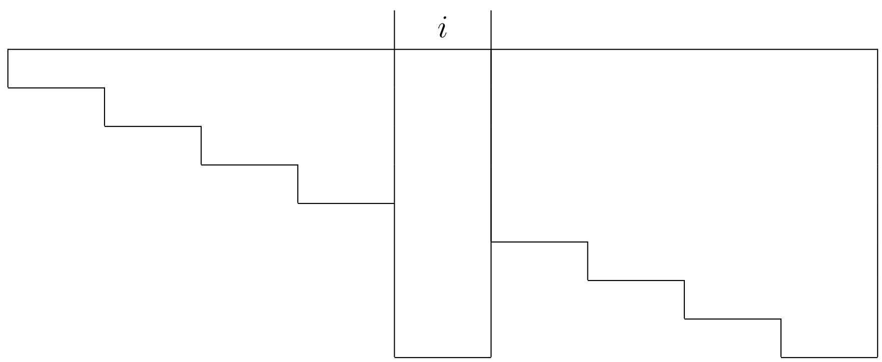
La matrice B^{-1} est requise dans deux sous-calculs de l'algorithme 1.1 : il s'agit de $\texttt{SOLUTION}(\pi B = c_B)$ et de $\texttt{SOLUTION}(B\delta x = N_{\bar{\jmath}})$. Nous avons choisi d'écrire ces opérations sous forme de solutions de systèmes linéaires pour atténuer le réflexe naturel de parler de

B^{-1}. En effet, l'utilisation de la matrice inverse est un bien mauvais choix pour obtenir la solution de systèmes linéaires, tant du point de vue de l'efficacité des calculs que du point de vue de la précision numérique de la solution. Dans les ouvrages de méthodes numériques, on recommande plutôt d'utiliser une *mise en facteurs*, aussi nommée *décomposition* de la matrice B ; les décompositions recommandées sont la décomposition LU, correspondant à la méthode de Gauss, et la décomposition QR, pouvant être calculée avec les algorithmes de Givens, Householder ou Gramm-Schmidt.

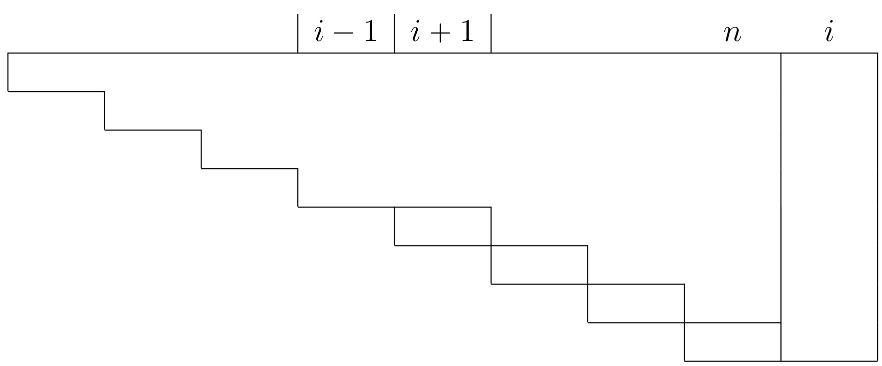
Dans la suite de cette section, nous présentons l'implantation de l'opération de pivotage de l'algorithme du simplexe en utilisant ces deux décompositions matricielles. Ensuite, nous discutons brièvement des difficultés numériques causées par le phénomène de dégénérescence, ou encore de dégénérescence numérique.

4.13.1 Utilisation de la décomposition LU

La décomposition LU d'une matrice carrée est obtenue par le procédé d'élimination de Gauss. L'appendice A résume les variantes de cette méthode. Supposons donc que la matrice B soit décomposée $B = LU$, où U est obtenue par le procédé d'élimination de Gauss, et L représente les opérations élémentaires utilisées lors de l'élimination. Supposons maintenant que la matrice B se fasse substituer une nouvelle colonne B'_i à la place de l'ancienne. Alors, de la relation $U = L^{-1}B$, on tire que $H = L^{-1}B'$ sera de la forme suivante :



Il suffit maintenant, à l'aide d'une permutation de colonne, de déplacer la colonne trouble à la dernière position en décalant les autres pour obtenir la forme suivante :



Cette dernière matrice est nommée une *matrice Hessenberg supérieure*, et est presque déjà triangularisée, à l'exception de quelques éléments sur la sous-diagonale. On voit donc qu'il est possible de triangulariser cette matrice Hessenberg supérieure en utilisant $n-i$ pivots de la méthode d'élimination de Gauss. De plus, on peut utiliser la stratégie de pivot partiel de manière très simple en examinant seulement l'élément diagonal, et l'élément sous-diagonal car les autres éléments d'une colonne sont tous nuls. Soit donc $M = M_n M_{m-1} \dots M_{i+1}$ la matrice qui représente les $n-i$ pivots. On a $ML^{-1}B' = U'$, de sorte que $L' = LM^{-1}$.

Une autre stratégie de choix de pivot permet de profiter de la structure de zéros de la matrice.

4.13.2 Utilisation de la décomposition QR

Il est possible de triangulariser une matrice non-singulière à l'aide d'une matrice orthogonale Q.

4.13.3 Dégénérescence numérique

Les problèmes causés par le phénomène de dégénérescence ne se limitent pas aux troubles de cyclage décrits plus haut. En fait, même en présence de problèmes dégénérés, lors des calculs, les variables x_B ne seront pas nulles, mais prendront des valeurs numériquement très petites.

Un autre problème numérique relié à la dégénérescence est le suivant. Considérons l'exemple où dans $\mathbb{R}^3$, trois contraintes se rencontrent selon une droite. Bien entendu, un sommet des contraintes impliquera au moins une quatrième contrainte. Cependant, il est fort possible que, *numériquement, les trois contraintes définissent un point.* Ce point n'est pas, en général, un point extrême des contraintes au sens mathématique, mais peut fort bien, dû aux erreurs numériques, être accepté comme solution optimale.

Il faut retenir que le phénomène de dégénérescence engendre des problèmes au niveau théorique, et en engendre d'autres, encore plus difficile à résoudre, au niveau numérique.

4.14 Résumé

Notre première incursion dans le domaine de l'optimisation sous contrainte concerne les fonctions objectif linéaires, les contraintes linéaires et des variables non-négatives. Les techniques d'algèbre linéaire sont suffisantes pour résoudre ce type de problème.

On analyse que s'il existe au moins une solution optimale, il suffit d'explorer les solutions dites de base pour en trouver une. Le nombre de solutions de base étant fini, si on s'assure de ne pas tourner en rond, on aura identifié la solution optimale en un nombre fini de calculs.

Une relation forte relie le problème sous étude (nommé primal) à un autre, nommé dual.

4.15 Extensions et références

Le matériel de ce chapitre est bien connu. Deux bonnes références quant aux aspects combinatoires de la programmation linéaire sont les livres de Chvátal et de Papadimitriou et Steiglitz [30]. Bien sûr, le livre classique de Dantzig [8], qui est à l'origine de toute cette théorie demeure un ouvrage de référence précieux.

Pour compléter, mentionnons qu'une grande partie de la popularité de la programmation linéaire provient d'interprétations économiques aussi instructives qu'utiles. En voici une résumée. Le problème de la diète consiste, pour un client à choisir des quantités d'aliments x_i (variables primales) qui minimisent le coût d'achat cx tout en satisfaisant des contraintes diététiques $a_i x \geqslant b^i$ (contenu suffisant de chacun de plusieurs nutriments). Le dual s'interprète du point de vue d'un vendeur de pilules de nutriments. Il veut fixer le prix π_j de ses pilules de telle sorte que, compte tenu de la demande en nutriments b, il maximise son profit πb tout en s'assurant que le coût des nutriments contenus dans chacun des l'aliment ne dépasse pas le coût de l'aliment $\pi A_j \leqslant c_j$.

Le traitement moderne de ces problèmes utilise des techniques de programmation non-linéaire, qui seront abordées dans des chapitres ultérieurs : les méthodes de point intérieur.

4.16 Tous les exercices du chapitres

Exercice (4.2.1, page 182) *[Systèmes d'équations linéaires]*

a) Montrez que si B est une matrice $m \times m$ singulière, et si le système $Bx = b$ possède une solution, alors l'ensemble des solutions constitue un ensemble affine.

b) Dans les conditions de a), montrez que si le système possède une solution telle que $x \geqslant 0$, alors il possède une solution avec une ou des composantes nulles.

Exercice (4.2.2, page 183) *[Équivalence de formulations]* Considérez le programme linéaire le plus général comportant des contraintes d'égalité et d'inégalité ainsi que des variables bornées et des variables libres.

$$\begin{aligned} \min\ & c_1 x_1 + c_2 x_2 \\ \text{sujet à}\ & A_{11} x_1 + A_{12} x_2 \leqslant b_1 \\ & A_{21} x_1 + A_{22} x_2 = b_2 \\ & x_1 \geqslant 0, \end{aligned}$$

où A est une matrice $(m_1 + m_2) \times (n_1 + n_2)$ et $c, x \in \mathbb{R}^{n_1+n_2}$ et $b \in \mathbb{R}^{m_1+m_2}$.

a) Démontrez que l'ensemble E des contraintes est un ensemble convexe.

b) Ramenez ce programme sous la forme standard du simplexe (4.1).

c) Ramenez ce programme sous la forme standard utilisant uniquement des contraintes d'inégalité (4.2)

Exercice (4.3.1, page 184) *[PL équivalent]* Considérez le problème $\min_x \|Ax - y\|_1$.

a) Reformulez-le sous la forme d'un programme linéaire.
b) Transformez le programme linéaire résultant sous la forme standard (4.1).
c) Transformez le programme linéaire résultant sous la forme standard (4.2).

Exercice (4.3.2, page 185) *[PL équivalent]* Considérez le problème $\min_{Ze=\tilde{y}} \|e\|_1$.

a) Reformulez-le sous la forme d'un programme linéaire.
b) Transformez le programme linéaire résultant sous la forme standard (4.1).
c) Transformez le programme linéaire résultant sous la forme standard (4.2).

Exercice (4.4.1, page 186) *[Programmes non bornés]* Est-il possible que le programme (4.1) ainsi que le programme max cx sous les mêmes contraintes soient tous deux non-bornés ? Justifiez.

Exercice (4.5.1, page 190) *[Illustration graphique]* Illustrez graphiquement l'itération de l'algorithme

Exercice (4.5.2, page 190) *[Pivots]* Pour s'exercer avec l'opération de pivot du simplexe, en commençant avec l'origine, effectuez 5 pivots pour passer successivement à tous les sommets de la figure 4.1.

Exercice (4.6.1, page 193) *[Programmes irréalisables]* Fournissez un exemple de programme pour lequel ni le primal, ni le dual ne possède de solution réalisable.

Exercice (4.6.2, page 193) *[Dualité]* Considérez le programme linéaire suivant

$$\begin{aligned} \min\ & cx \\ \text{sujet à}\ & Ax \leqslant b \\ & x \geqslant 0, \end{aligned}$$

et vérifiez que son dual est

$$\begin{aligned} \max\ & yb \\ \text{sujet à}\ & yA \leqslant c \\ & y \leqslant 0 \end{aligned}$$

en le ramenant à la forme standard, puis en simplifiant le dual ainsi obtenu.

Exercice (4.6.3, page 194) *[Dualité]* Considérez le programme linéaire le plus général envisageable

$$\begin{aligned} \min\ & c_1x_1 + c_2x_2 \\ \text{sujet à } & A_{11}x_1 + A_{12}x_2 \leqslant b_1 \\ & A_{21}x_1 + A_{22}x_2 = b_2 \\ & x_1 \geqslant 0, \end{aligned}$$

où A est une matrice $(m_1 + m_2) \times (n_1 + n_2)$ et $c, x \in \mathbb{R}^{n_1+n_2}$ et $b \in \mathbb{R}^{m_1+m_2}$. Caractérisez le dual de ce problème.

Exercice (4.7.1, page 197) *[Illustration graphique]* Illustrez graphiquement l'itération de l'algorithme. De plus, sur la figure 4.1, les points extrêmes primal-réalisables sont identifiés ; identifiez les points extrêmes dual-réalisables.

Exercice (4.7.2, page 197) *[Dual]* Nous allons refaire le tout directement sur le dual.

a) Écrivez le dual du problème (4.3).
b) Ramenez le dual sous la forme standard de la forme (4.1).
c) Faites un graphique du dual et établissez la correspondance entre les intersections des droites du primal (figure 4.1) et celles du dual.
d) Établissez la correspondance entre les solutions de base du primal et du dual, tous deux standartisés.
e) Appliquez l'algorithme primal du simplexe à la formulation du dual standardisée.
f) Constatez que l'algorithme primal appliqué au dual est équivalent l'algorithme dual appliqué au primal.

Exercice (4.9.1, page 200) *[Valeur de z]* Complétez le tableau pour avoir une expression de la valeur de $-z$ à chaque itération, et donc pour la solution optimale.

Exercice (4.9.2, page 200) *[Intervalle final]* Quel est l'intervalle de μ qui rend le tableau final réalisable et optimal ?

Exercice (4.9.3, page 201) *[Illustration graphique]* Illustrez graphiquement les itérations de l'algorithme

***Exercice (4.10.1, page 201)** *[Dégénérescence et unicité]*

a) Démontrez que si aucune solution optimale du programme (4.1) n'est dégénérée, alors le programme (4.11) possède une solution unique.
b) Est-il vrai que le programme (4.11) possède toujours plus d'une solution lorsque le primal (4.1) possède une solution dégénérée ?

Exercice (4.10.2, page 203) *[Dégénérescence]* Considérez les contraintes linéaires

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, \dots n$$
$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \dots n$$
$$x_{ij} \geqslant 0, i = 1, 2, \dots n, j = 1, 2, \dots n.$$

Trouvez un vecteur c pour lequel ce problème possède des bases dégénérée simultanément pour les programmes (4.1) et (4.11).

Exercice (4.10.3, page 205) *[Anti-cyclage]* Considérez le tableau initial de l'exemple de la section 4.10.1 ; avec chacune des deux règles d'anti-cyclage présentées à la section 4.10.2, effectuez suffisamment d'itérations de l'algorithme du simplexe pour changer de point réalisable.

Exercice (4.10.4, page 205) *[Valeur optimale]* Quelle est la valeur optimale du programme linéaire sur l'exemple de cyclage ?

Exercice (4.10.5, page 205) *[Deux phases]* Proposez une méthode, utilisant deux phases, permettant que pour chacune des deux phases, le tableau initial comporte des lignes lex-positives.

***Exercice (4.10.6, page 206)** *[Théorie]* Démontrez la formule utilisée dans la preuve du théorème 4.10.1 donnant la mise à jour de la ligne 0 après un pivot.

Exercice (4.12.1, page 213) *[Joueur y]* Complétez les détails conduisant au problème (4.35) du joueur y.

Exercice (4.12.2, page 214) *[Preuve]* Fournissez les détails de la preuve

Exercice (4.12.3, page 214) *[Jeux]* Résolvez le jeu avec la matrice (4.27), obtenez les stratégies, et vérifiez que la valeur du jeu est $-16/102$.

Exercice (4.12.4, page 214) *[Jeu simple]* Deux joueurs cachent une piece de monnaie, un 5 ou un 10 sous. Si elles sont pareilles, le joueur A empoche les 2, autrement le joueur B empoche les 2. Qui a l'avantage dans ce jeu ?

Chapitre 5

Optimisation différentiable avec contraintes linéaires

Sujets du chapitre

- Contraintes linéaires d'égalité et d'inégalité.
- Formulations diverses des conditions d'optimalité.
- Problèmes de complémentarité linéaire.
- Déduction de quelques algorithmes à partir des conditions d'optimalité.

Introduction

La programmation linéaire a connu un essor considérable dans les années 60. Il est donc naturel que des chercheurs aient généralisé de diverses manières les idées développées dans l'analyse de programmes linéaires. Nous présentons dans ce chapitre quelques généralisations. Nous abordons également ces problèmes non-linéaires du point de vue de la programmation non-linéaire, constituant ainsi un premier contact avec les sujets des chapitres à venir.

5.1 Énoncé du problème

Nous abordons maintenant la généralisation de problèmes de type (4.1) au cas où la fonction objectif est différentiable non-linéaire. Nous traitons le problème

$$\begin{aligned} \min\ & f(x) \\ \text{sujet à}\ & Ax = b \\ & x \geqslant 0, \end{aligned} \tag{5.1}$$

où l'ensemble $E \stackrel{\text{def}}{=} \{x : Ax = b, x \geqslant 0\}$ est nommé *ensemble réalisable*. Remarquons que l'ensemble réalisable est celui du programme (4.1). Il est parfois plus pratique d'utiliser l'ensemble réalisable du programme (4.2) $E = \{x : Ax \leqslant b\}$. Tel que discuté à la section 4.2, il est toujours possible de faire la conversion d'une formulation à l'autre. Dans cet esprit, nous utiliserons la formulation qui conduit aux développements les plus simples à exprimer. Rappelons que quelle que soit la formulation, l'ensemble E est toujours un ensemble convexe.

Nous allons d'abord déduire les conditions d'optimalité pour le programme (5.1) à partir de la théorie de la programmation linéaire présentée au chapitre 4. Ensuite, nous effectuerons d'autres démarches qui conduisent à des formulations équivalentes des conditions d'optimalité.

Comme dans les chapitres précédents, nous déduirons quelques algorithmes à partir de points qui ne satisfont pas aux conditions d'optimalité.

5.2 Conditions d'optimalité pour les points stationnaires

Définition 5.2.1 *Une direction d est dite réalisable pour un point $x \in E$ s'il existe un $\epsilon > 0$ tel que $x + td \in E$ $\forall t \in [0, \epsilon]$.*

Une direction réalisable en x est une direction qui nous maintient dans les contraintes à partir de x. Si x^* est un minimum local, aucune direction réalisable ne peut être une direction de descente.

Constatons que les directions réalisables d constituent un cône $K = \{d : Ad = 0, d_i \geqslant 0\ \forall i : x_i^* = 0$. Il s'agit bien d'un cône car si $d \in K$, on vérifie facilement que $\alpha d \in K \forall \alpha >= 0$. En fait, puisque E est un ensemble convexe, toute direction réalisable peut s'exprimer comme $x - x^*$ où $x \in E$.

Nous avons déjà vu que le DED est un demi espace. Une conséquence stipule simplement que $DED \cap K = \emptyset$.

Si x^* est une solution optimale du programme (5.1), alors la dérivée directionnelle de f en x^* dans toute direction réalisable $d = (x - x^*)$ doit être non-négative, ce qui s'exprime par l'*inégalité variationnelle*

$$\begin{aligned} \nabla f(x^*)(x - x^*) &\geqslant 0 \\ Ax &= b \\ x &\geqslant 0. \end{aligned} \tag{5.2}$$

Un tel point x^* est nommé *point stationnaire* pour le programme (5.1). Cette formulation constitue la base des analyses subséquentes.

5.2.1 Formulation inspirée de la programmation linéaire

Puisque x^* satisfait aux équations et inéquations (5.2), x^* est également solution optimale du programme linéaire

$$\begin{aligned} &\min \nabla f(x^*)x \\ &\text{sujet à } Ax = b \\ &x \geqslant 0. \end{aligned} \tag{5.3}$$

Cependant, ce n'est pas nécessairement une solution optimale de base, et nous devons généraliser quelque peu la théorie présentée au chap 4.

La partition Base—Hors-base revisitée

Pour le problème (5.1), rien ne permet de supposer que x^* est une solution de base. En fait, puisque l'objectif f n'est pas linéaire, il est fort possible que toutes les composantes de la solution x^* soient positives. Par exemple, le minimum de $f(x) = x^2$ sur l'intervalle $[-1, 1]$ est atteint *à l'intérieur* du domaine réalisable.

Heureusement, il est possible de tirer profit des cas ou elle comporte des composantes nulles. Ces composantes nulles sont toujours nommées composantes hors-base, et notées x_N. Par ailleurs, parmi les composantes non-nulles, nommons x_B les composantes associées à une sous-matrice inversible B, et baptisons les autres composantes *variables de super-base*, notées x_S. Nous obtenons donc la partition $[\![B \mid S \mid N]\!]$ et la matrice A, ainsi que les vecteurs c et

x sont partitionnés de telle sorte que nous puissions écrire $x = [x_B, x_S, x_N]$, $A = [B, S, N]$ ainsi que $\nabla f(x^*) = [\nabla f(x^*)_B, \nabla f(x^*)_S, \nabla f(x^*)_N]$.

Tout comme avant, il est possible de mettre x_B en fonction de x_S et de x_N :

$$x_B = B^{-1}(b - Sx_S - Nx_N).$$

Alors, en définissant encore (voir page 186) $\pi \stackrel{\text{def}}{=} \nabla f(x^*)_B B^{-1}$, on arrive à l'expression

$$\nabla f(x^*)x = (\nabla f(x^*)_N - \pi N)x_N + (\nabla f(x^*)_S - \pi S)x_S + Cst.$$

De même les contraintes se récrivent

$$\begin{aligned} B^{-1}(b - Nx_N - Sx_S) &\geqslant 0 \\ x_N, x_S &\geqslant 0. \end{aligned}$$

En adaptant le raisonnement qui nous a conduit aux conditions d'optimalité du programme (4.1), nous obtenons

$$\pi A = \begin{cases} \pi B = \nabla f(x^*)_B \\ \pi S = \nabla f(x^*)_S \\ \pi N \leqslant \nabla f(x^*)_N. \end{cases}$$

La première égalité (les composantes B) définit les multiplicateurs π ; la seconde égalité vient du fait que les composantes S sont strictement positives, et donc pourraient aussi bien augmenter que diminuer ; leurs coûts relatifs doivent donc s'annuler. L'inégalité sur les composantes hors-base est comme dans le cas linéaire.

En introduisant les variables d'écart $s \geqslant 0$, on peut reformuler les conditions précédentes comme

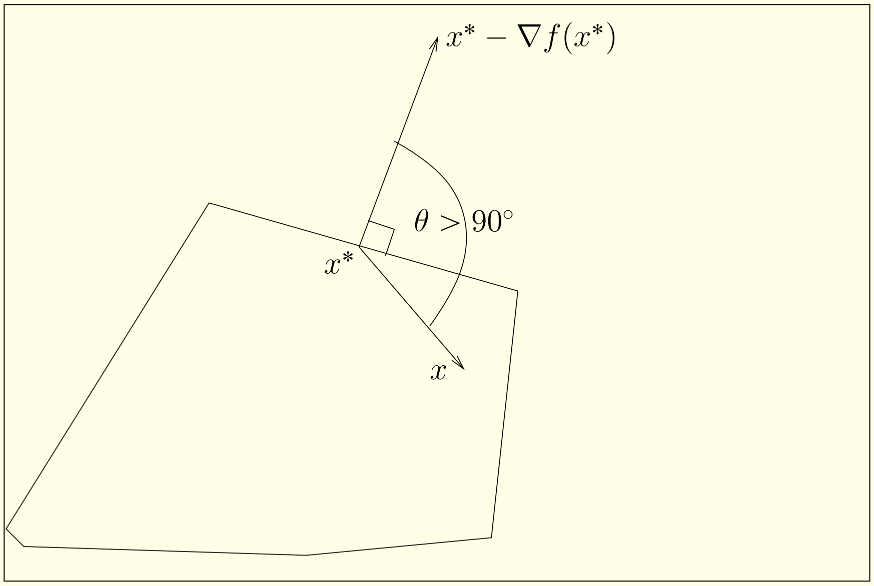
$$\begin{aligned} \pi A + s &= \nabla f(x^*) \\ sx &= 0, \end{aligned}$$

le dernier terme étant connu sous le nom de *terme de complémentarité.*

Dégénérescence

Les raisonnements précédents sont valables à condition qu'il soit possible de trouver des composantes positives de x^*, x^*_B associées à une sous-matrice inversible B. Cette limite n'est toutefois seulement apparente. En effet, si la matrice A possède des lignes linéairement indépendantes et si le problème possède au moins une solution réalisable, on peut toujours trouver des composantes $x^*_B \geqslant 0$ associés à une matrice inversible. À aucun moment nous avons utilisé le fait que les composantes de x^*_B devaient être strictement positives.

Nous verrons un peu plus loin que la dégénérescence de la solution optimale influence l'énoncé de conditions de second ordre.



On voit que si $-\nabla f(x^*)$ fait un angle obtus avec $x - x^*$ pour tout point réalisable x, alors le point réalisable le plus proche de $x^* - \nabla f(x^*)$ est le point x^* lui-même.

Figure 5.1 – Géométrie de la projection

5.2.2 Formulation sous forme de projection

Revenons aux inéquations variationnelles (5.2). On y fait état que l'angle entre $-\nabla f(x^*)$ et un vecteur allant de x^* à x, un point réalisable quelconque n'est pas un angle aigu car son cosinus est non-positif.

On constate donc que le point le plus proche de $x^* - \nabla f(x^*)^t$ à l'intérieur des contraintes est le point x^* lui-même. Ce point le plus proche se nomme *projection*.

Définition 5.2.2 *Le point le plus proche de x dans l'ensemble E est noté $Proj_E(x)$ et s'exprime comme :*

$$Proj_E(x) = \arg\min_{y \in E} \frac{1}{2}\|y - x\|^2.$$

On a donc que $x^* = Proj_{\{x:Ax=b, x\geqslant 0\}}(x^* - \nabla f(x^*)^t)$. On dit donc que x^* est un point fixe de la fonction $P(x) = Proj_E(x - \nabla f(x)^t)$, c'est-à-dire, $x^* = P(x^*)$.

5.3 Conditions de Karush-Kuhn-Tucker

Pour les développements qui suivent, nous allons nous éloigner de la formulation des contraintes sous la forme standard du simplexe. Nous allons développer ces conditions en deux étapes : d'abord, nous traiterons seulement les contraintes d'égalité, et ensuite, nous verrons comment adapter la théorie aux inégalités.

5.3.1 Conditions pour les contraintes d'égalité

Donc, momentanément, nous considérons les problèmes de la forme

$$\begin{aligned} &\min f(x) \\ &\text{sujet à } Ax = b. \end{aligned} \tag{5.4}$$

La notion d'orthogonalité est cruciale pour nos développements. Observons d'abord que si on a sous la main un point réalisable x, alors un autre point réalisable peut être obtenu à l'aide d'une direction $d : Ad = 0$, c'est-à-dire une direction appartenant au noyau de la matrice A. Nous savons que le noyau d'une matrice constitue un espace vectoriel. Soit Z une matrice dont les colonnes forment une base de $\ker A$. Alors, $AZ = 0$. De plus, en admettant que les m lignes de la matrice A soient linéairement indépendantes, la matrice Z comportera n lignes et $n - m$ colonnes. Une direction réalisable, donc appartenant à $\ker A$, peut donc s'exprimer comme une combinaison linéaire des colonnes de la matrice Z, qui, rappelons-le, constituent une base de $\ker A$. Notons les $(n - m)$ coefficients de la direction réalisable d dans la base en question $d_z : d = Zd_z$.

Maintenant, supposons que x^* soit un point réalisable et optimal (localement). Nous allons récrire $f(x)$ pour des x réalisables proches de x^* à l'aide de direction réalisables, $f(x) = f(x^* + Zd_z)$. Puisque x^* est un minimum local de f, alors $d_z = 0$ est un minimum local de $\phi(d_z) \stackrel{\text{def}}{=} f(x^* + Zd_z)$. En fait, l'étude de la fonction f dans l'intersection d'un voisinage de x^* avec le domaine réalisable est équivalent à l'étude de la fonction ϕ dans un voisinage de zéro. Les conditions d'optimalité s'exprimeront donc à l'aide des quantités $\nabla\phi(0)$ et $\nabla^2\phi(0)$. Or, $\nabla\phi(d_z) = \nabla f(x^* + Zd_z)Z$ et $\nabla^2\phi(d_z) = Z^t\nabla^2 f(x^* + Zd_z)Z$, et donc $\nabla\phi(0) = \nabla f(x^*)Z$ alors que $\nabla^2\phi(0) = Z^t\nabla^2 f(x^*)Z$. Ces dernières quantités portent les noms de *gradient réduit* et *hessien réduit* de f en x^* respectivement.

Maintenant, en appliquant les conditions d'optimalité du chapitre 3 à la fonction ϕ, on déduit que $\nabla f(x^*)Z = 0$. Examinons de plus près cette condition. Nous avons supposé que les lignes de la matrice A étaient linéairement indépendantes, et puisque les colonnes de la matrice Z forment une base, elles sont également linéairement indépendantes. En fait, les colonnes de la matrice Z constituent une base de $\ker A$ et les lignes de la matrice A constituent une base de $\ker Z^t$. Puisque $\nabla f(x^*)^t \in \ker Z^t$, on a donc que $\nabla f(x^*)$ peut s'écrire comme une combinaison linéaire des lignes de la matrice A, $\nabla f(x^*) = \pi A$.

L'optimisation avec contraintes d'égalité a été étudiée depuis fort longtemps. Lagrange avait formulé les conditions d'optimalité que nous venons de déduire à l'aide d'une fonction auxiliaire appelée *Lagrangien* $L(x, \lambda) \stackrel{\text{def}}{=} f(x) + \lambda(Ax - b)$. Les conditions d'optimalité de premier ordre s'écrivent simplement

$$\begin{aligned} L(x^*, \lambda^*) = 0 &= \nabla f(x^*) + \lambda^* A \\ \nabla_\lambda L(x^*, \lambda^*) = 0 &= Ax^* - b. \end{aligned}$$

On voit donc que la seconde condition assure que x est réalisable alors que la première est équivalente à celle que nous avons déduite avec $\lambda = -\pi$.

On peut résumer ces développements dans le théorème suivant.

Théorème 5.3.1 (Conditions nécessaires et suffisantes — contraintes d'égalité) - *Considérons le problème (5.4) et soit Z une matrice dont les colonnes constituent une base de* $\ker A$. *Soit $x^* \in E$.*

1. *Si x^* est un minimum local de f, alors il existe un vecteur λ^* tel que $\nabla f(x^*) + \lambda^* A = 0$, ou de façon équivalente, $\nabla f(x^*)Z = 0$. De plus, la matrice $Z^t \nabla^2 f(x^*) Z$ est semi-définie positive.*
2. *S'il existe un vecteur λ^* tel que $\nabla f(x^*) + \lambda^* A = 0$, ou de façon équivalente, $\nabla f(x^*)Z = 0$ et si de plus, la matrice $Z^t \nabla^2 f(x^*) Z$ est définie positive, alors x^* est un minimum local de f.*

Nous venons de voir qu'à partir d'un candidat minimum local x^*, on peut établir des conditions d'optimalité en exhibant l'équivalence entre $\min \phi(d_z)$ et le problème (5.4).

Voyons maintenant comment obtenir la solution d'un problème du type (5.4). Commençons par obtenir un point réalisable $x_0 : Ax_0 = b$. Il est bien connu que l'ensemble des solutions d'un système $Ax = b$ est constitué d'une solution particulièr x_0 auquel on additionne un élément quelconque de $\ker A$. Le problème (5.4) se ramène donc à résoudre $\min_{d_z} \phi_{x_0}(d_z)$ où la fonction $\phi_{x_0}(d_z) = f(x_0 + Zd_z)$. Obtenons $d_z^* \in \arg\min \phi_{x_0}(d_z)$ et nous en déduirons $x^* = x_0 + Zd_z^*$.

Exercice 5.3.1 *[Résoudre (5.4)]* Vérifiez que $x^* = x_0 + Z\alpha^*$ est bel et bien un minimum local de (5.4).

Exercice 5.3.2 *[Matrices Z]* Soient les contraintes linéaires suivantes.

$$A = \begin{pmatrix} 3 & -5 & 1 & 1 \\ 3 & 1 & -5 & 1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 3 \end{pmatrix}.$$

Soient également les quatre candidats matrices Z :

$$Z_1 = \begin{pmatrix} 3 & -3 \\ 3 & -3 \\ 3 & -3 \\ 3 & -3 \end{pmatrix} \quad Z_2 = \begin{pmatrix} -1 & 1 & 0 \\ -2 & 0 & -1 \\ -2 & 0 & -1 \\ -5 & -3 & -4 \end{pmatrix} \quad Z_3 = \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 3 \end{pmatrix} \quad Z_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Une seule de ces matrices Z a la propriété que ses colonnes constituent une base de $\ker A$. Identifiez laquelle et indiquez clairement pourquoi les trois autres n'ont pas la propriété voulue.

Exercice 5.3.3 *[Application simple]* Soit le programme

$$\begin{array}{rrcl} & \min f(x,y,z) & = & x^2+2y^2+3z^2-0.5xy+x-y+2z \\ \text{sujet à} & 2x+3y-z & = & 1. \end{array}$$

Identifiez un point stationnaire en utilisant trois techniques :

a) Calculez un multiplicateur de Lagrange λ^* et un point $(x^*, y^*, z^*)^t$ satisfaisant au système d'équations $\nabla f(x,y,z) + \lambda(2,3,-1) = 0$.

b) Éliminez une variable à l'aide de l'équation $2x+3y-z=1$ en posant $z = 2x+3y-1$ et traitez les deux variables qui restent sans contrainte.

c) Éliminez une variable en identifiant une base du noyau de la matrice $(2,3,-1)$, et obtenez la solution.

d) Vérifiez que les trois méthodes fournissent la même solution.

e) Quelle est la nature du point stationnaire ? (min, max, pt de selle, ?)

Exercice 5.3.4 *[Contraintes d'égalité]* Considérez le problème

$$\min_{x\in\mathbb{R}^3} \frac{1}{2}\|x\|^2$$
$$\text{sujet à } x_1 + 2x_2 + 3x_3 = 1.$$

a) Fournissez une expression pour la matrice Z.

b) Identifiez la solution x^* et λ^* du problème.

c) Vérifiez que $\nabla f(x^*)Z = 0$.

d) Vérifiez que $x^* = Proj_E(x^* - \nabla f(x^*))$.

e) Résolvez le problème

$$\min_{x\in\mathbb{R}^n} \|x\|^2$$
$$\text{sujet à } \sum_{i=1}^{n} i x_i = 1.$$

Exercice 5.3.5 *[Géométrie — condition d'optimalité]* Trouvez le rectangle de superficie maximale pour un périmètre donné ($2(h+l)=1$) en utilisant les conditions nécessaires. Vérifiez que votre solution satisfait aux conditions suffisantes de second ordre.

Exercice 5.3.6 *[Projection sur un sous-espace]* On sait que la projection sur un sous-espace vectoriel est une application linéaire, et donc peut se représenter par une matrice. Nous appliquerons la théorie pour obtenir cette matrice. Considérez le problème de projeter un vecteur $-g$, l'opposé du gradient sur un sous-espace défini par $x : Ax = 0$ où la matrice A est de plein rang, donc ses lignes sont linéairement indépendantes. Le problème s'écrit alors :

$$\begin{array}{rl} \min & \frac{1}{2}\|x+g^t\|^2 \\ \text{sujet à} & Ax = 0 \end{array}$$

a) Écrivez les conditions d'optimalité de ce problème.
b) Exprimez la solution $x(\lambda)$ explicitement en fonction du vecteur de multiplicateurs de Lagrange λ.
c) Injectez l'expression trouvée en b). dans les contraintes pour déterminer λ^*.
d) Obtenez alors l'expression de $x(\lambda^*) = x^* = -Mg^t$.
e) Vérifiez que la matrice de projection M est symétrique, et satisfait $M^2 = M$.

5.3.2 Conditions pour les contraintes d'inégalité

Encore ici, nous adoptons une formulation différente de la forme standard du simplexe, et considérons le problème

$$\begin{gathered} \min f(x) \\ \text{sujet à } Ax \leqslant b. \end{gathered} \tag{5.5}$$

Soit un point x^* candidat minimum local. Alors, x^* est d'abord un point réalisable. Parmi les contraintes, certaines seront saturées en x^*, c'est-à-dire certaines seront satisfaites avec égalité, et les autres seront satisfaites avec inégalité stricte. Séparons la matrice A ainsi que le vecteur b selon ce critère, de sorte que $A_= x^* = b_=$ et $A_< x^* < b_<$; les premières contraintes sont nommée *contraintes actives* et les autres contraintes inactives. Puisque nous étudions les optima locaux de f, nous limitons notre analyse à un voisinage de x^*, et donc, on peut choisir un voisinage assez petit pour que les contraintes inactives soient toujours satisfaites avec inégalité stricte. Par conséquent, on peut effectuer notre analyse avec les seules contraintes actives.

En particulier, on voit que les conditions présentées plus haut pour les contraintes d'égalités s'appliquent encore ici aux contraintes actives, et donc il devra exister des valeurs λ telles que $\nabla f(x^*) + \lambda A_= = 0$. Cependant, dans notre cas, les variables λ devront satisfaire à une condition additionnelle : il ne suffit plus que $\nabla f(x^*)^t \in \ker Z_=^t$ mais il faut également qu'il soit orienté *vers l'intérieur* du domaine réalisable. En effet, il faut que les directions réalisables fassent des angles aigus avec $\nabla f(x^*)^t$, selon l'inégalité variationnelle (5.2). Or, cette bonne orientation de $\nabla f(x^*)^t$ s'exprime simplement par $\lambda \geqslant 0$; l'expression est simple, et se déduit du lemme suivant, connu comme le *lemme de Farkas*.

Lemme 5.3.1 (Farkas) *Soient A une matrice $m \times n$ et $v \in \mathbb{R}^n$, vecteur ligne ; alors, un et un seul des deux systèmes suivants possède une solution :*

$$Ad \leqslant 0, vd < 0,$$

$$v + \lambda A = 0, \lambda \geqslant 0.$$

Preuve Si le second système possède une solution, multiplions sa première équation par d pour obtenir $vd + \lambda Ad = 0$, ce qui, supposant que $Ad \leqslant 0$, entraînerait que $\lambda Ad \leqslant 0$ et par conséquent $vd \geqslant 0$, et donc que le premier système ne possède pas de solution.

Si le premier système ne possède pas de solution, alors le progamme linéaire $\min_{Ad \leqslant 0} vd$ admet $d = 0$ comme solution optimale, et donc son dual possède une solution réalisable (théorème de dualité forte) ; son dual est $\max_{v+\lambda A=0, \lambda \geqslant 0} 0$. □

Ces observations nous donnent des conditions nécessaire d'optimalité, qui imitent celles du théorème 5.3.1.

Théorème 5.3.2 (Condition nécessaire d'ordre un) *Considérons le problème (5.5) et soit x^* un minimum local de f tel que $A_= x^* = b_=$ et $A_< x^* < b_<$; soit également $Z_=$ une matrice dont les colonnes constituent une base de $\ker A_=$. Alors, $\nabla f(x^*) Z_= = 0$ et il existe un vecteur $\lambda_=^* \geqslant 0$ tel que $\nabla f(x^*) + \lambda_=^* A_= = 0$. Ceci entraîne qu'il existe un vecteur $\lambda^* \geqslant 0$ tel que*

- $\nabla f(x^*) + \lambda^* A = 0$,
- $\lambda^*(Ax^* - b) = 0$.

Remarquons le terme $\lambda^*(Ax^* - b) = 0$, nommé terme de complémentarité, dans ces conditions. Ce terme peut se récrire

$$\sum_{i=1}^{m} \lambda_i^*(A_i x^* - b_i) = 0$$

et puisque pour chaque i $\lambda_i^* \geqslant 0$ et $A_i x^* - b_i \leqslant 0$, leur somme étant nulle, un seul de λ_i^* et $(A_i x^* - b_i)$ peut être non-nul pour chaque i. Une conséquence est que toute contrainte satisfaite avec inégalité stricte $A_i x^* - b_i$ se voit attribuer un $\lambda_i^* = 0$. Le terme de complémentarité entraîne donc que $\lambda_<^* = 0$.

Ces conditions d'optimalité constituent un cas particulier des conditions de Karush-Kuhn-Tucker (K–K–T), cas particulier aux contraintes linéaires. Nous verrons dans les chapitres ultérieurs des conditions plus générales.

Encore ici, nous pouvons utiliser le Lagrangien pour exprimer les conditions d'optimalité. Rappelons que $L(x, \lambda) = f(x) + \lambda(Ax - b)$. Les conditions d'optimalité de premier ordre s'écrivent :

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= 0 \\ Ax^* - b = \nabla_\lambda L(x^*, \lambda^*) &\leqslant 0 \\ \lambda^* &\geqslant 0 \\ \lambda^*(Ax^* - b) &= 0. \end{aligned}$$

Définition 5.3.1 *Une solution x^* satisfait à l'hypothèse de* stricte complémentarité *si et seulement si pour chaque contrainte active i (i.e. $A_i x^* = b_i$) on a $\lambda_i^* > 0$. Si les multiplicateurs ne sont pas uniques, on convient que l'hypothèse est satisfaite s'il existe un λ^* pour lequel pour chaque contrainte active i (i.e. $A_i x^* = b_i$) on a $\lambda_i^* > 0$.*

L'hypothèse de stricte complémentarité impose que $\lambda_i^* = 0 \iff A_i x^* < b_i$. Cette condition de stricte complémentarité est reliée au phénomène de dégénérescence dont nous avons parlé au chapitre 4.

Nous pouvons immédiatement énoncer une condition nécessaire d'ordre deux qui stipule que $Z_=^t \nabla^2 f(x^*) Z_= \succeq 0$, découlant naturellement des conditions nécessaires du théorème 5.3.1. Après avoir introduit une condition suffisante, nous présenterons une condition nécessaire d'ordre deux plus utile.

Si un problème présente une combinaison de contraintes d'égalité et d'inégalités non négatives et non positives, on peut déduire une condition nécessaire d'ordre un générale.

Corollaire 5.3.1 (Condition nécessaire générale) *Considérons le problème*

$$\begin{aligned} &\min f(x) \\ &\textit{sujet à } A_1 x \leqslant b_1, \\ &\qquad\quad A_2 x \geqslant b_2, \\ &\qquad\quad A_3 x = b_3. \end{aligned} \tag{5.6}$$

Si $x^* \in E$ est un minimum local, alors il existe trois vecteurs de multiplicateurs $\lambda_1^* \geqslant 0$, $\lambda_2^* \leqslant 0$ et λ_3^* libre tels que

- $\nabla f(x^*) + \lambda_1^* A_1 + \lambda_2^* A_2 + \lambda_3^* A_3 = 0,$

- $\lambda_1^*(A_1x^* - b_1) = 0$,
- $\lambda_2^*(A_2x^* - b_2) = 0$.

Exercice 5.3.7 *[Conditions générales]* Démontrez le corollaire 5.3.1 en convertissant le programme (5.6) avec uniquement des contraintes d'inégalités non-positives puis en appliquant le théorème 5.3.2.

Le cas de problèmes de maximisation se déduit naturellement du corollaire 5.3.1.

Corollaire 5.3.2 (Condition nécessaire générale—max) *Considérons le problème*

$$\begin{aligned} &\max f(x) \\ \textit{sujet à } &A_1x \leqslant b_1, \\ &A_2x \geqslant b_2, \\ &A_3x = b_3. \end{aligned} \tag{5.7}$$

Si $x^* \in E$ est un minimum local, alors il existe trois vecteurs de multiplicateurs $\lambda_1^* \leqslant 0$, $\lambda_2^* \geqslant 0$ et λ_3^* libre tels que

- $\nabla f(x^*) + \lambda_1^* A_1 + \lambda_2^* A_2 + \lambda_3^* A_3 = 0$,
- $\lambda_1^*(A_1x^* - b_1) = 0$,
- $\lambda_2^*(A_2x^* - b_2) = 0$.

Exercice 5.3.8 *[Conditions générales pour un max]* Démontrez le corollaire 5.3.2 en convertissant le programme (5.7) en un problème de minimisation avec uniquement des contraintes d'inégalités non-positives.

Définition 5.3.2 *Un point est dit* stationnaire *s'il satisfait aux conditions nécessaire d'ordre un pour un minimum local ou pour un maximum local. En particulier, en présence de seulement des contraintes d'inégalités de type non-positif, un point est stationnaire si tous ses multiplicateurs λ sont du même signe ; s'ils sont positifs, c'est un candidat minimum, autrement un candidat maximum.*

Pour confirmer qu'un point stationnaire candidat pour un minimum local est bel et bien un minimum, il faut développer des conditions suffisantes (d'ordre deux).

Il est commode d'introduire une fois pour toute une notation qui permet de regrouper les contraintes selon qu'elles sont actives (satisfaites avec égalité), inactives (inégalités strictes) et fortement actives (active et leur multiplicateur $\lambda_i > 0$). Remarquons en passant que les multiplicateurs associés aux contraintes inactives sont forcément nuls, c'est une conséquence de la condition de complémentarité.

Définition 5.3.3 *Soit $x^* \in E$ un point qui satisfait aux conditions nécessaires d'ordre un et λ^* un vecteur de multiplicateurs associé.*

$$I^* \stackrel{\text{def}}{=} \{i : A_i x^* = b_i\} \tag{5.8}$$

$$\hat{I} \stackrel{\text{def}}{=} \{i \in I^* : \lambda_i^* > 0\} \tag{5.9}$$

$$J \stackrel{\text{def}}{=} \{j : A_j x^* < b_j\}. \tag{5.10}$$

Évidemment, la définition s'étend aux combinaisons d'inégalités non-négatives et non-positives. Les contraintes d'égalités sont toujours dans $\hat{I}$, même si leur multiplicateur est nul.

Pour obtenir des conditions suffisantes, il faut observer que même si le vecteur λ^* est non-négatif, certaines de ses composantes pourraient être nulles. Dans l'énoncé qui suit, la sous-matrice $A_{\hat{I}}$ comporte les contraintes satisfaites avec égalité, mais pour lesquelles le multiplicateur associé est strictement positif. Les autres contraintes n'influencent pas la condition de premier ordre $\nabla f(x^*) + \lambda_{\hat{I}}^* A_{\hat{I}} = 0$ puisque leurs multiplicateurs sont nuls.

Théorème 5.3.3 (Conditions suffisantes) *Considérons le problème (5.5) et $x^* \in E$ un point réalisable ; s'il existe un vecteur $\lambda^* \geqslant 0$ tel que $\nabla f(x^*) + \lambda^* A = 0$ et $\lambda^*(Ax^* - b) = 0$, ou de façon équivalente, $\nabla f(x^*) Z_{\hat{I}} = 0$ et si de plus, l'expression $z^t \nabla^2 f(x^*) z$ est positive pour tout $z : A_{I^* \setminus \hat{I}} z \leqslant 0, A_{\hat{I}} z = 0, z \neq 0$, alors x^* est un minimum local de f.*

Preuve Nous voulons montrer qu'il existe $\epsilon > 0$ tel que $f(x) \geqslant f(x^*)$ pour tout $x \in E \cap V_\epsilon(x^*)$. Choisissons ϵ assez petit pour que $A_j x < b_j, j \in J$. L'analyse portera donc uniquement sur les contraintes de I^* (et $\hat{I}$). Considérons une direction réalisable quelconque $d : A_{I^*} d \leqslant 0$. Nous allons vérifier la condition suffisante du corollaire 3.2.2 que 0 satisfait aux conditions suffisantes pour $\min_{\theta \geqslant 0} h_{x^*,d}(\theta)$. Un développement de Taylor nous donne

$$h_{x^*,d}(\theta) = h_{x^*,d}(0) + h'_{x^*,d}(0) + \frac{1}{2} h''_{x^*,d}(\xi)$$

où $0 \leqslant \xi \leqslant \theta$. En exprimant h et ses dérivées, on obtient

$$f(x^* + \theta d) - f(x) = \theta \nabla f(x^*) d + \frac{1}{2} \theta^2 d^t \nabla^2 f(x^* + \xi d) d.$$

Or, $\nabla f(x^*) + \lambda^*_{I^*} A_{I^*} = 0$ donc pour les d réalisables, $\nabla f(x^*) d = 0$. Puisque pour les d réalisables $d^t \nabla^2 f(x^*) d > 0$, alors si ξ est suffisamment petit, $d^t \nabla^2 f(x^* + \xi d) d$ également, d'où $f(x^* + \theta d) - f(x) > 0$ lorsque $\theta > 0$. □

Ce dernier théorème appelle plusieurs commentaires.

Commentaires élémentaires

Nous utilisons les conditions nécessaires et les conditions suffisantes pour étudier des candidats de minima et–ou maxima de programmes d'optimisation. En présence de contraintes d'inégalités, les signes des multiplicateurs λ sont importants, tout comme le signe de l'expression $z^t \nabla^2 f(x^*) z$ pour des z appropriés. Considérons le problème (5.5). Si les multiplicateurs λ_i sont tous non-négatifs, alors le point étudié est un candidat pour un minimum local. Ceci sera confirmé si l'expression $z^t \nabla^2 f(x^*) z$ est positive pour tout z approprié. Si au contraire, les multiplicateurs λ_i sont tous non-positifs, alors le point étudié est candidat pour un maximum local, ce qui se confirmera par le fait que $z^t \nabla^2 f(x^*) z$ est négatif pour tout z approprié.

Si les multiplicateurs λ^*_i ne sont pas tous du bon signe pour un maximum ou un minimum local, le point étudié n'est pas un point stationnaire. Si les conditions de premier ordre sont satisfaites soit pour un min, soit pour un max, on dit que le point candidat est stationnaire. Il faut ensuite examiner le signe de $z^t \nabla^2 f(x^*) z$ pour confirmer (ou non) que le candidat, point stationnaire, est un minimum local ou un maximum local.

Cas particuliers Si jamais la matrice $\nabla^2 f(x^*)$ est définie positive, alors les conditions de second ordre pour un minimum local sont satisfaites automatiquement. En effet, $z^t \nabla^2 f(x^*) z \geqslant 0$ pour tout $z \neq 0$, et donc pour tout z non nul approprié. Pour un point x^* stationnaire candidat pour un minimum, c'est alors automatiquement confirmé qu'il s'agit d'un minimum local.

Si jamais il y a n contraintes dans $\hat{I}$, $ker A_{\hat{I}} = \vec{0}$ et donc *il n'y a pas de z non nul dans $ker A_{\hat{I}}$*. Dans ce cas, les conditions de second ordre sont satisfaites trivialement et le seul signe des multiplicateurs λ^* départage les minima des maxima locaux.

Subtilités impliquant I^* et $\hat{I}$

Remarque 5.3.1 *1. La portion de premier ordre ($\nabla f(x^*) + \lambda^*_{\hat{I}} A_{\hat{I}} = 0$) n'est pas influencée par les multiplicateurs nuls ;*

2. pour les conditions suffisantes, il faut distinguer, parmi les contraintes satisfaites avec égalité, celles qui ont un multiplicateur associé positif de celles qui en ont un nul ;

3. *pour les conditions suffisantes d'ordre 2, on s'écarte de la formulation avec contraintes d'égalités en explicitant des directions z pour lesquelles $z^t\nabla^2 f(x^*)z > 0$;*
4. *les z pour lesquels $z : A_{I^*\setminus\hat{I}}z \leqslant 0, A_{\hat{I}}z = 0, z \neq 0$ constituent un cône ;*
5. *on peut renforcer les conditions nécessaires en s'inspirant de la formulation des conditions suffisantes.*

Exemple 5.3.1 Pour mettre en relief les subtilités des conditions nécessaires et suffisantes en présence de contraintes d'inégalité, considérons le programme suivant :

$$\begin{aligned} \min\ & x_1^2 - x_2^2 \\ \text{sujet à}\ & 2x_2 - x_1 \leqslant 0 \\ & 2x_2 + x_1 \geqslant 0. \end{aligned} \tag{5.11}$$

L'origine en est un point stationnaire. Les deux contraintes y sont actives. Leurs multiplicateurs respectifs sont nuls. Donc, la matrice $A_{I^*\setminus\hat{I}}$ est vide. Cependant, le cône des directions réalisables comporte des vecteurs $v = (v_1, v_2)^t$ tels que $-v_1 \leqslant 2v_2 \leqslant v_1$, et pour de tels vecteurs, $v^t \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} v = v_1^2 - v_2^2 > \frac{3v_1^2}{4}$, ce qui confirme que l'origine est un minimum local pour ce programme. On voit aussi que les conditions nécessaires pourraient faire état que la matrice $\nabla^2 f(x^*)$ satisfait $z^t\nabla^2 f(x^*)z \geqslant 0$ pour les z appropriés. $\square$

Avant de présenter quelques exemples, abordons les conditions nécessaires d'ordre deux. Nous avons déjà mentionné qu'une condition nécessaire serait que $Z^t\nabla^2 f(x^*)Z \geq 0$ où les colonnes de Z engendrent $\ker A_{I^*}$. Si la condition de stricte complémentarité est satisfaite, c'est-à-dire si $\lambda^*_{I^*} > 0$, on ne peut pas affirmer d'avantage. Par contre, s'il existe des $i \in I^*$ tels que $\lambda^*_i = 0$, il est possible de tirer une conclusion plus forte. En plus d'affirmer que $z^t\nabla^2 f * x^*)z \geqslant 0$ pour tout $z : A_{I^*}z = 0$, nous pouvons affirmer que $z^t\nabla^2 f * x^*)z \geqslant 0$ pour tout $z : A_{I^*\setminus\hat{I}}z \leqslant 0, A_{\hat{I}}z = 0$.

Théorème 5.3.4 (Conditions nécessaire d'ordre deux) *Considérons le problème (5.5) et $x^* \in E$ un minimum local; alors il existe un vecteur $\lambda^* \geqslant 0$ tel que $\nabla f(x^*) + \lambda^* A = 0$ et $\lambda^*(Ax^* - b) = 0$; de plus, l'expression $z^t\nabla^2 f(x^*)z$ est positive pour tout $z : A_{I^*\setminus\hat{I}}z \leqslant 0, A_{\hat{I}}z = 0, z \neq 0$.*

Preuve L'existence de multiplicateurs λ^* découle du théorème 5.3.2. La démonstration se complète en constatant que pour toute direction réalisable d, $0 \geqslant f(x^* + \theta d) - f(x^*) = h_{x^*,d}(\theta) - h_{x^*,d}(0)$ entraîne que $\theta^* = 0$ est un minimum local de $h_{x^*,d}(\theta)$, et donc que $h''_{x^*,d}(0) \geqslant 0$. $\square$

5.4 Exemples de points stationnaires

Nous allons utiliser le problème (4.3) pour faire le tour des conditions d'ordre un, c'est-à-dire des conditions pour un point stationnaire pour un minimum local. En fait, nous allons conserver du problème ses contraintes, mais y traiter une fonction objectif non linéaire, $\frac{1}{2}\|x - \hat{x}\|^2$ où $\hat{x} = (2, 1)^t$.

Considérons le problème

$$\begin{array}{lrcrcl} \min f(x) = & \frac{1}{2}\|x-\hat{x}\|^2 & & & & \\ \text{sujet à} & 2x_1 & + & x_2 & \geqslant & 2 \\ & x_1 & + & 3x_2 & \leqslant & 3 \\ & x_1 & , & x_2 & \geqslant & 0 \end{array} \tag{5.12}$$

que l'on transforme sous la forme standard en

$$\begin{array}{lrcrcrcrcl} \min f(x) = & \frac{1}{2}\|x-\hat{x}\|^2 & & & & & & & & \\ \text{sujet à} & 2x_1 & + & x_2 & - & x_3 & & & = & 2 \\ & x_1 & + & 3x_2 & & & + & x_4 & = & 3 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0 \end{array} \tag{5.13}$$

On veut donc trouver le point le plus proche de $\hat{x}$ dans les contraintes. Par inspection, on voit bien que la solution optimale sera sur le côté haut du triangle, c'est-à-dire sur la droite d'équation $x_1 + 3x_2 = 3$. Calculons donc cette solution.

5.4.1 Deux dimensions

Commençons par profiter que nous savons que le problème avec une seule contrainte linéaire permet d'identifier la solution. Nous allons procéder de deux manières : par le gradient réduit et par les conditions de Lagrange.

Lagrange

Écrivons les conditions de Lagrange, x est réalisable et donc satisfait à la contrainte, et il existe un multiplicateur λ tel que $\nabla f(x) + \lambda A = 0$:

$$\begin{array}{rcl} (x_1 - 2, x_2 - 1) + \lambda(1, 3) & = & 0 \\ x_1 + 3x_2 & = & 3 \end{array}$$

ce qui nous donne trois équations, trois inconnues que l'on peut simplifier à :

$$\begin{array}{rcl} x_1 & = & 2 - \lambda \\ x_2 & = & 1 - 3\lambda \\ x_1 + 3x_2 & = & 3 \end{array}$$

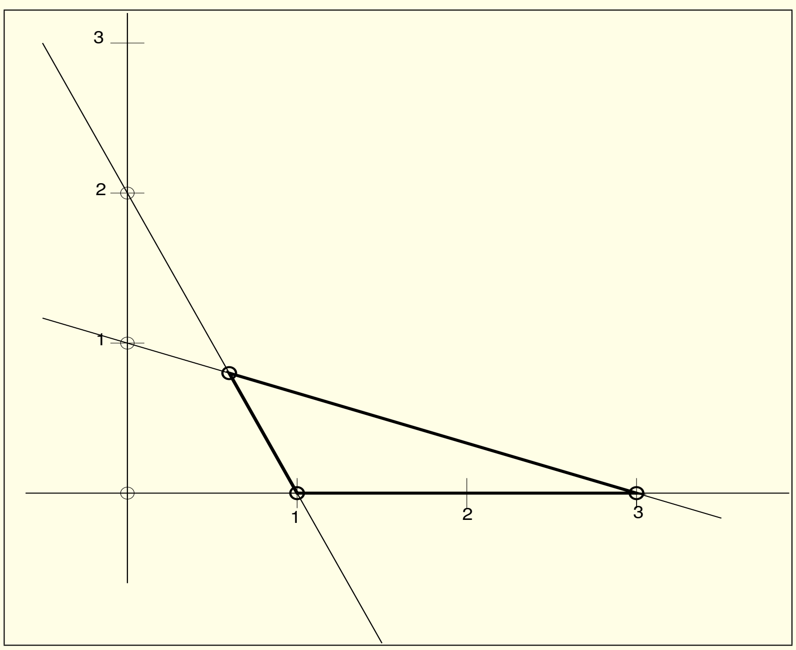
et en remplaçant x_1 et x_2 par leur expression en λ, on tire $5 - 10\lambda = 3$ d'où $\lambda = \frac{1}{5}$ et $x^* = (\frac{9}{5}, \frac{2}{5})^t$.

Gradient réduit

Maintenant, puisque $\nabla f(x^*)Z = 0$, prenons $Z = (3, -1)^t \perp (1, 3)^t$ pour écrire les équations à résoudre ; auparavant, remarquons que $x^* = (3, 0)^t + Z\alpha = (3 + 3\alpha, -\alpha)^t$. Ainsi, nous devons solutionner $\nabla f(3 + 3\alpha, -\alpha)Z = 0$, c'est-à-dire $3(3 + 3\alpha - 2) + \alpha + 1 = 0$ et donc $\alpha^* = -\frac{2}{5}$ et par conséquent $x^* = (\frac{9}{5}, \frac{2}{5})^t$.

Inégalités

Comme notre problème original comportait des inégalités, vérifions que notre intuition était bonne en vérifiant que les conditions de KKT pour le problème original sont satisfaites. Ces conditions stipulent qu'il existe un vecteur λ^* de dimension 4 car nous avons 4 contraintes, et que ses composantes sont négatives sauf la seconde associée à la contrainte $x_1 + 3x_2 \leqslant 2$. Les contraintes sont satisfaites. Sauf la contrainte $x_1 + 3x_2 \leqslant 2$, les contraintes sont satisfaites avec inégalité stricte et donc la condition de complémentarité stipule que les composantes correspondantes de λ^* sont nulles. Le tout se ramène donc aux conditions avec la seule contrainte d'égalité, et nous avons trouvé que $\lambda_2^* = \frac{1}{5} > 0$.



Les intersections des droites encerclées en gras sont les points extrêmes réalisables correspondant aux solutions de base réalisables de (5.14). Les intersections encerclées en mince ne sont pas réalisables, mais sont tout de même des solutions de base de (5.14).

Figure 5.2 – Exemple simple bi-dimensionnel

Projection

Le problème de départ étant la projection, on a évidemment la condition d'optimalité sous forme de projection qui est satisfaite.

Linéaire

On vérifie d'emblée que $c = \nabla f(x^*) = (-\frac{1}{5}, -\frac{3}{5})$ et en plus de x^* qui est solution optimale de $\min cx$ dans nos contraintes, on a que tout le côté du haut du triangle est solution optimale du programme linéaire, en particulier ses extrémités $P = (\frac{3}{5}, \frac{4}{5})^t$ et $Q = (3, 0)^t$:

$$cx^* = xP = xQ = -\frac{3}{5}.$$

5.4.2 Dimension 4

Examinons maintenant sous la forme dite standard du simplexe la solution $x^* = (\frac{9}{5}, \frac{2}{5})^t$. En examinant les contraintes, on trouve $x_3^* = 2$ et $x_4^* = 0$. Observons immédiatement que la variable d'écart x_3 n'est pas nulle pour la contrainte inactive. Puisque nous sommes en présence de $m = 2$ contraintes et $n = 4$ vartiables, une solution de base devrait avoir $n - m = 2$ composantes positives, et donc x^* n'est pas une solution de base. En fait, pour un problème général avec M contraintes d'inégalité dans $\mathbb{R}^N$, on se retrouve avec $n = N + M$ variables et $m = M$ contraintes une fois sous forme standard et on voit qu'une solution de base correspond à avoir exactement n contraintes actives dans le problème original.

Même si x^* n'est pas une solution de base, ceci n'empêche pas x^* d'être une solution optimale du problème linéaire dont la fonction objectif est $\nabla f(x^*)x = (-\frac{1}{5}, -\frac{3}{5}, 0, 0)x$

$$\begin{array}{lrcrcrcrcl} \min z = & -\frac{1}{5}x_1 & - & \frac{3}{5}x_2 & & & & & & \\ \text{sujet à} & 2x_1 & + & x_2 & - & x_3 & & & = & 2 \\ & x_1 & + & 3x_2 & & & + & x_4 & = & 3 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0 \end{array} \tag{5.14}$$

Comme trois composantes sont positives, il y en a une de trop car $n - m = 4 - 2 = 2$. On peut choisir deux d'entre elles comme base et l'autre comme "super"base.

Choix de base {1,2} Choisissons les composandes de base de B, x_1 et x_2 et la composante de super base x_3 alors que x_4 est la composante hors-base. La sous matrice $B = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$ et en multipliant par $B^{-1} = \begin{pmatrix} \frac{3}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{2}{5} \end{pmatrix}$ les contraintes $Ax = b$, on retrouve le système

$$\begin{pmatrix} 1 & 0 & -\frac{3}{5} & -\frac{1}{5} \\ 0 & 1 & \frac{1}{5} & \frac{2}{5} \end{pmatrix} x = \begin{pmatrix} \frac{3}{5} \\ \frac{4}{5} \end{pmatrix}$$

avec la colonne 3 correspondant à S et la colonne 4 à N. $c_B B^{-1} = (-\frac{1}{5}, -\frac{3}{5})\begin{pmatrix} \frac{3}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{2}{5} \end{pmatrix} = (0, -\frac{1}{5})$. On vérifie donc directement que $c_S - c_B B^{-1} S = 0$ et $c_N - c_B B^{-1} N = \frac{1}{5} > 0$.

Choix de base {1,3} Ce choix correspond au point $(3,0)^t$ sur la figure 5.2, donc un point réalisable. Alors, la variable de superbase est x_2 et en multipliant par $B^{-1} = \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix}$ les contraintes $Ax = b$, on retrouve le système

$$\begin{pmatrix} 1 & 3 & 0 & 1 \\ 0 & 5 & 1 & 2 \end{pmatrix} x = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

$c_B B^{-1} = (-\frac{1}{5}\ 0)\begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix} = (0\ -\frac{1}{5})$ et $c_S - c_B B^{-1} S = 0$, $c_N - c_B B^{-1} N = \frac{1}{5} > 0$.

Choix de base {2,3} Ce choix correspond au point $(0\ 1)^t$ sur la figure 5.2, et donc n'est pas une base réalisable. On peut cependant l'utiliser sans savoir que ce n'est pas une base réalisable. Alors, la variable de superbase est x_1 et en multipliant par $B^{-1} = \begin{pmatrix} 0 & \frac{1}{3} \\ -1 & \frac{1}{3} \end{pmatrix}$ les contraintes $Ax = b$, on retrouve le système

$$\begin{pmatrix} \frac{1}{3} & 1 & 0 & \frac{1}{3} \\ -\frac{5}{3} & 0 & 1 & \frac{1}{3} \end{pmatrix} x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

On constate que la base n'est pas réalisable puisque $B^{-1}b \ngeq 0$.

$c_B B^{-1} = (-\frac{3}{5}\ 0)\begin{pmatrix} 0 & \frac{1}{3} \\ -1 & \frac{1}{3} \end{pmatrix} = (0\ -\frac{1}{5})$ et $c_S - c_B B^{-1} S = 0$, $c_N - c_B B^{-1} N = \frac{1}{5} > 0$.

5.4.3 Exemples

Pour illustrer les conditions d'optimalité, voici une série d'exemples. Il s'agit d'abord d'un problème comportant deux contraintes d'égalité. Ensuite, nous analyserons le même problème, mais ne comportant qu'une seule contrainte d'égalité. Finalement, nous analyserons la situation si les deux contraintes du problème initial étaient des inégalités, soit non-négatives, non-positives, ou encore un mélange des deux.

Nous avons choisi des contraintes homogènes (du type $Ax = 0$) de sorte qu'un point particulier des contraintes est trivialement obtenu comme $x_0 = \vec{0}$. La fonction $\phi_{x_0}(d_z)$ est donc simplement $f(Zd_z)$. Pour des exemples où les contraintes ne seraient pas homogènes, $Ax = b$, il faudrait préalablement identifier un $x_0 : Ax_0 = b$ et constituer la fonction $\phi(d_z) = f(x_0 + Zd_z)$. Les colonnes de la matrice Z constitueraient toujours une base de $\ker A$.

Également, la fonction objectif est quadratique et $\nabla^2 f(x) \equiv Q$ est définie positive. Selon les commentaires de cas particuliers de la page 234, les conditions seront automatiquement satisfaites pour un minimum local. Les exemples sont de dimension $n = 3$ et aucun d'eux n'a un $\hat{I}$ qui contient 3 contraintes, donc aucun des exemples ne pourra être un maximum local.

Exemple 1

Considérons le programme

$$\begin{array}{rc} \min & x_1^2 - x_1x_2 + x_2^2 - x_2x_3 + x_3^2 + 2x_1 - x_2 \\ \text{sujet à} & 3x_1 - x_2 + x_3 = 0 \\ & 2x_1 - x_2 - x_3 = 0 \end{array}$$

qui peut également s'exprimer matriciellement :

$$\begin{array}{rc} \min & \frac{1}{2}x^tQx + cx \\ \text{sujet à} & Ax = 0 \end{array}$$

où

$$Q = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \quad c = \begin{pmatrix} 2 & -1 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 3 & -1 & 1 \\ 2 & -1 & -1 \end{pmatrix}.$$

En additionnant la première ligne a la seconde dans la matrice A, on obtient

$$A' = \begin{pmatrix} 3 & -1 & 1 \\ 5 & -2 & 0 \end{pmatrix},$$

d'où l'on déduit que $x_1 = \frac{2}{5}x_2$, et, utilisant la premiere ligne, $x_3 = \frac{-1}{5}x_2$, donc, une expression pour la base du noyau de A est donnée par $Z = \begin{pmatrix} 2 \\ 5 \\ -1 \end{pmatrix}$.

On a donc que tout x réalisable est un multiple de Z, $x = Zd_z$. Par conséquent, on peut solutionner le problème original avec contraintes en minimisant par rapport a d_z la fonction $\frac{1}{2}d_zZ^tQZd_z + c^tZd_z$. La matrice Z^tQZ est réduite au scalaire 50, le vecteur c^tZ est réduit au scalaire -1 et la valeur optimale de $25d_z^2 - d_z$ est $d_z^* = \frac{1}{50}$. Ceci entraîne que $x^* = \begin{pmatrix} \frac{2}{50} \\ \frac{5}{50} \\ \frac{-1}{50} \end{pmatrix}$.

Vérifions que $(x^*)^tQ + c + \lambda A$ possède bel et bien une solution λ^*. Remarquons que x^* est réalisable, car il satisfait aux contraintes. En général, pour un point réalisable quelconque $\bar{x}$, le système $\bar{x}^tQ + c + \lambda A$ ne possède pas de solution en λ. Seuls les points stationnaires partagent cette propriété.

$$(x^*)^t Q + c = \begin{pmatrix} \frac{99}{50} & \frac{-41}{50} & \frac{-7}{50} \end{pmatrix}.$$

La solution du système

$$\begin{pmatrix} \frac{99}{50} & \frac{-41}{50} & \frac{-7}{50} \end{pmatrix} + \begin{pmatrix} \lambda_1 & \lambda_2 \end{pmatrix} \begin{pmatrix} 3 & -1 & 1 \\ 2 & -1 & -1 \end{pmatrix} = 0$$

est

$$\lambda = \begin{pmatrix} \frac{-17}{50} & \frac{-24}{50} \end{pmatrix}.$$

Exemple 2

Considérons maintenant la même fonction objectif, mais à minimiser sous seulement la seconde contrainte, $2x_1 - x_2 - x_3 = 0$, donc $A = \begin{pmatrix} 2 & -1 & -1 \end{pmatrix}$. Une base du plan correspondant au domaine réalisable est

$$Z = \begin{pmatrix} 1 & 1 \\ 0 & 2 \\ 2 & 0 \end{pmatrix}$$

car les colonnes de cette matrice sont linéairement indépendantes, et satisfont toutes les deux à la contrainte. Cette fois-ci, la matrice

$$Z^t Q Z = \begin{pmatrix} 10 & -4 \\ -4 & 6 \end{pmatrix}$$

n'est pas réduite à un scalaire, et le vecteur

$$cZ = \begin{pmatrix} 2 & 0 \end{pmatrix}$$

non plus. Alors, d_z est ici un vecteur de dimension 2. La solution optimale de $\min_{d_z} \frac{1}{2} d_z^t Z^t Q Z d_z + cZd_z$ est donnée par $d_z^* = \begin{pmatrix} \frac{-3}{11} \\ \frac{-2}{11} \end{pmatrix}$. Par conséquent, $x^* = \begin{pmatrix} \frac{-5}{11} \\ \frac{-4}{11} \\ \frac{-6}{11} \end{pmatrix}$. Calculons

$$(x^*)^t Q + c = \begin{pmatrix} \frac{-6}{11} & \frac{3}{11} & \frac{-8}{11} \end{pmatrix} + \begin{pmatrix} 2 & -1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{16}{11} & \frac{-8}{11} & \frac{-8}{11} \end{pmatrix}.$$

On observe bien que $(x^*)^t Q + c + \frac{-8}{11} A = 0$, et donc $\lambda^* = \frac{-8}{11}$.

Exemple 3

Les deux contraintes de l'exemple 1 étaient des contraintes d'égalité. Si les deux contraintes avaient été

$$3x_1 - x_2 + x_3 \geqslant 0$$
$$2x_1 - x_2 - x_3 \geqslant 0,$$

alors le point x^* trouvé serait un point stationnaire pour le problème. En effet, dans la formulation Lagrangienne, $\nabla f(x^*) + \lambda^* A = 0$, et les multiplicateurs de Lagrange doivent etre du signe opposé aux contraintes $Ax - b$. Inégalités *plus grand ou égal* dans les contraintes entraînent λ *plus petit ou égal* à zéro, et vice-versa. Ici, les contraintes sont *plus grand ou égal* et les λ *plus petit ou égal* à zéro, $\lambda^* = \begin{pmatrix} \frac{-17}{50} & \frac{-24}{50} \end{pmatrix}$.

Exemple 4

Si les deux contraintes avaient été

$$3x_1 - x_2 + x_3 \leqslant 0$$
$$2x_1 - x_2 - x_3 \geqslant 0,$$

alors le point x^* trouvé à l'exemple 1 ne serait pas un point stationnaire pour le problème. En effet, la première contrainte étant non-positive, les multiplicateurs doivent être non-négatifs, et ce n'est pas le cas. Il faut supposer qu'au moins une des deux contraintes que nous avons implicitement supposées actives ne l'est pas à la solution. Puisque c'est la première qui permet d'établir la non-optimalité du point, supposons donc que seule la seconde contrainte est active à la solution. Ceci est la situation de l'exemple 2, et le multiplicateur que nous avons calculé est du bon signe, $\lambda^* = \frac{-8}{11}$.

Exemple 5

Les deux contraintes de l'exemple 1 étaient des contraintes d'égalité. Si les deux contraintes avaient été

$$3x_1 - x_2 + x_3 \leqslant 0$$
$$2x_1 - x_2 - x_3 \leqslant 0,$$

alors le point x^* trouvé serait un point stationnaire candidat maximum pour le problème. En effet, dans la formulation Lagrangienne, $\nabla f(x^*) + \lambda^* A = 0$ et les multiplicateurs de Lagrange doivent etre du même signe(pour un max) que les contraintes $Ax - b$. Inégalités *plus petit ou égal* dans les contraintes entraînent λ *plus petit ou égal* à zéro, et vice-versa. Ici, les contraintes sont *plus petit ou égal* et les λ *plus petit ou égal* à zéro, $\lambda^* = \begin{pmatrix} \frac{-17}{50} & \frac{-24}{50} \end{pmatrix}$. Cependant, en passant aux conditions de second ordre, puisque la matrice $\nabla^2 f(x^*) \equiv 0 \succ 0$, les conditions nécessaires d'ordre deux pour un maximum local ne peuvent pas être satisfaites et le candidat n'est ni un minimum local, ni un maximum local.

Exemple 6

Si les deux contraintes avaient été

$$3x_1 - x_2 + x_3 \leqslant 0$$
$$2x_1 - x_2 - x_3 \leqslant 0,$$

alors le point x^* trouvé à l'exemple 1 ne serait pas un point stationnaire candidat minimum pour le problème. En effet, aucune des deux contraintes se voit associer un multiplicateur du bon signe. Par ailleurs, l'ensemble des contraintes actives à la solution ne peut pas être uniquement la seconde contrainte car l'exemple 2 nous révèle que son multiplicateur n'est pas du bon signe. Il est donc possible que la solution optimale pour ce cinquième exemple ne sature aucune des deux contraintes, ou encore en sature une seule, la première. Les autres possibilités ont été éliminées dans les deux premiers exemples. Supposons donc qu'il n'y ait aucune contrainte saturée. Il faut alors résoudre $x^tQ + c = 0$. Or, la première ligne de la matrice Q est justement égale à c. La solution est donc $x^* = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$, et par bonheur, satisfait avec inégalité stricte aux deux contraintes. C'est donc la solution de notre problème.

Il aurait pu en être autrement, car même si les deux multiplicateurs sont du mauvais signe cela ne veut pas dire que les multiplicateurs calculés avec moins de contraintes seront toujours du mauvais signe. Il en est de même pour des multiplicateurs du bon signe. Sur de petits exemples, c'est habituellement le cas, mais, comme en programmation linéaire, au cours de l'exploration de l'algorithme du simplexe, qui correspond à la stratégie d'enlever des contraintes actives (variables hors-base) une contrainte associée à un multiplicateur du mauvais signe (coût relatif négatif), il est possible qu'une variable entre dans la base plus d'une fois, i.e. qu'une contrainte devienne présumément active à plusieurs reprises.

Lorsqu'un problème présente une combinaison de contraintes d'inégalité et d'égalité, les multiplicateurs associés aux contraintes d'inégalité doivent être de signe opposé aux inégalités, et les multiplicateurs associés aux contraintes d'égalité ne sont pas restreints en signe.

Exercice 5.4.1 *[Contraintes d'inégalités]*

a) Vérifiez que la solution obtenue à l'exercice 5.3.4 constitue la solution optimale du problème

$$\min_{x\in\mathbb{R}^3} \|x\|^2$$
$$\text{sujet à } x_1 + 2x_2 + 3x_3 = 1; \qquad x \geqslant 0.$$

b) Que peux-t-on dire de cette dernière solution par rapport au problème

$$\min_{x\in\mathbb{R}^3} \|x\|^2$$
$$\text{sujet à } x_1 + 2x_2 + 3x_3 \leqslant 1; \qquad x \geqslant 0.$$

c) Que peux-t-on dire de cette dernière solution par rapport au problème

$$\min_{x\in\mathbb{R}^3} \|x\|^2$$
$$\text{sujet à } x_1 + 2x_2 + 3x_3 \geqslant 1; \qquad x \geqslant 0.$$

Exercice 5.4.2 *[Application simple]* Pour les problèmes suivants, obtenez la solution.

a)

$$\begin{array}{rl} \min & -\left(x_1^2 - x_1x_2 + 2x_2^2 + x_1 + x_2\right) \\ \text{sujet à} & 2x_1 + x_2 \leqslant 1 \\ & x \geqslant 0 \end{array}$$

b)

$$\begin{array}{rl} \min & -\left(x_1^2 - 2x_1x_2 + 2x_2^2 + 2x_1 + 5x_2\right) \\ \text{sujet à} & 2x_1 + 3x_2 \leqslant 20 \\ & 3x_1 - 5x_2 \leqslant 4 \\ & x \geqslant 0 \end{array}$$

Exercice 5.4.3 *[Projection...]* Considérez le problème de trouver le point de norme minimale dans l'ensemble défini par les contraintes suivantes :

$$\begin{array}{rcl} x_1 + 2x_2 - x_3 & \geqslant & 4 \\ -x_1 + x_2 - x_3 & \leqslant & 2. \end{array}$$

a) Formulez ce problème comme problème de minimisation.

b) Supposons que les deux contraintes sont actives à la solution. En exprimant x_1 et x_2 en fonction de x_3, obtenez la solution du problème avec contraintes d'égalités.

c) Cette solution ne constitue pas le point de norme minimale dans l'ensemble réalisable original. Pourquoi ? Trouvez le point de norme minimale dans les contraintes d'inégalité.

***Exercice 5.4.4** *[PNL]* Obtenez la solution du problème

$$\min_{x,y,z} 4/3(x^2 - xy + y^2)^{3/4} - z$$
$$\text{sujet à } z \leqslant 2, \quad x, y, z \geqslant 0.$$

Indice : prolongez à $(0,0,0)^t$ le gradient de la fonction par continuité : le gradient n'existe pas à l'origine, mais la limite lorsque x s'approche de l'origine existe. Est-ce que la condition de stricte complémentarité est satisfaite en x^* ?

Exercice 5.4.5 *[Projection]* Soit $E = \{x : Ax - b \leqslant 0\}$ un ensemble réalisable. Vérifiez que $\bar{x}$ est un point fixe de l'opérateur de projection, $\bar{x} = Proj_E(\bar{x} - \nabla f(\bar{x})^t)$, si et seulement si $\bar{x}$ satisfait aux conditions nécessaires de premier ordre de K–K–T du problème (5.5).

Exercice 5.4.6 *[Caractérisation de la projection]* Si les contraintes sont de simples bornes sur les variables, i.e. $L \leqslant x \leqslant U$, vérifiez (en utilisant les conditions d'optimalité) que la solution de $\min_{L \leqslant x \leqslant U} \frac{1}{2}\|x - y\|^2$ est donnée par

$$x_i = \begin{cases} L_i & \text{si } L_i \geqslant y_i; \\ y_i & \text{si } L_i \leqslant y_i \leqslant U_i; \\ U_i & \text{si } y_i \geqslant U_i. \end{cases}$$

Exercice 5.4.7 *[Simplification de condition d'optimalité]* Prouvez, ou donnez un contre exemple :

Si x^* est un min local pour

$$\begin{array}{rl} \min & f(x) \\ \text{sujet à} & Ax \leqslant b \\ & x \geqslant 0 \end{array}$$

alors il existe $\lambda^* \geqslant 0$ tel que $\nabla f(x^*) + \lambda^* A \geqslant 0$, avec $\lambda^*(Ax^* - b) = 0$.

Exercice 5.4.8 *[Conditions d'optimalité]* Soient les trois fonctions suivantes :

- $f_1(x) = \frac{1}{2}((x_1 - 3)^2 + (x_2 + 1)^2)$;
- $f_2(x) = \frac{1}{2}((x_1 + 1)^2 + (x_2 - 1)^2)$;
- $f_3(x) = \frac{1}{2}((x_1 + 1)^2 + x_2^2)$.

Définissons un point stationnaire $\hat{x}$ comme étant un point réalisable pour lequel il existe des multiplicateurs de Lagrange, nuls pour les contraintes satisfaites avec inégalité stricte en $\hat{x}$, et tous du bon signe pour les contraintes actives en $\hat{x}$.

a) Considérez les (quatre) contraintes

$$0 \leqslant x_1 \leqslant 1,$$
$$0 \leqslant x_2 \leqslant 1,$$

définissant l'ensemble réalisable E, et le point candidat $x = (1, 0)^t \in E$.

i) Pour chacune des fonctions, vérifiez que le point candidat est un point stationnaire, et donnez les valeurs des multiplicateurs de Lagrange $\lambda_1, \lambda_2, \lambda_3$ et λ_4.

ii) Pour chacune des fonctions, déterminez si le point stationnaire est un minimum local, maximum local ou ni l'un ni l'autre.

5.5 Remarques générales sur les algorithmes

Comme pour les problèmes sans contrainte, nous allons déduire quelques algorithmes à partir de la violation des conditions d'optimalité. Cependant, un élément nouveau nous empêche de présenter une théorie de convergence globale aussi simple que celle du chapitre 3 : les contraintes d'inégalité introduisent un aspect combinatoire, à savoir l'identification des contraintes actives à la solution. Cet aspect se manifeste de deux manières principales : en présence de dégénérescence, les algorithmes inspirés de l'algorithme du simplexe pourront cycler, ou converger vers des points qui ne sont pas stationnaires. Pour d'autres types d'algorithmes, c'est le désir de demeurer dans le domaine réalisable, quitte à réduire le pas θ utilisé, qui provoque le désastre, les pas utilisés ne satisfaisant plus aux conditions d'admissibilité présentées au chapitre 3.

Cependant, tout comme pour l'algorithme du simplexe, ces difficultés théoriques n'ont pas empêché la production d'implantations efficaces qui se sont avérées robustes dans la pratique.

5.5.1 Algorithmes de directions réalisables

Les algorithmes que nous considérons pour les contraintes linéaires sont tous des algorithmes de *directions réalisables*. Ces algorithmes produisent une suite d'itérés x_k qui satisfont tous aux contraintes et les directions de descente utilisées sont réalisables, i.e. si on y effectue un pas de déplacement assez petit, $x_k + \theta d_k$ continue de satisfaire aux contraintes. En supposant que les contraintes définissent un ensemble réalisable E dont la forme algébrique n'est pas spécifiée autrement qu'en affirmant qu'il est décrit par un ensemble de contraintes linéaires d'inégalités et d'égalités, on écrit $x \in E$ pour signifier que x satisfait aux contraintes.

Exercice 5.5.1 *[Calcul de θ_{Max}]* Considérez deux formes pour les ensembles E,

a) $E = \{x : Ax = b, x \geqslant 0\}$;

b) $E = \{x : Ax \leqslant b\}$.

Itération de descente réalisable
{ Donnés : x_0, $\epsilon > 0$ et $0 < \tau_0 < \tau_1 < 1$ et $\tau_0 < \frac{1}{2}$ }
$x \leftarrow x_0$
tantque **(** $\neg$ (condition d'optimalité satisfaite) **)**
    $d \leftarrow$ direction_descente_réalisable(f, x)
    { $h'_{x,d} < 0$ et si θ est assez petit, alors $x + \theta d \in E$) }
    $\theta_{Max} \leftarrow \max(\theta : x + \theta d \in E)$
    $\theta^* \leftarrow$ pas_admissible(f, x, τ_0, τ_1)
    $\theta \leftarrow \min(\theta^*, \theta_{Max})$
    $x \leftarrow x + \theta d$

Algorithme 5.1: Descente réalisable.

Dans ces deux cas, soient un point réalisable $x \in E$ et une direction réalisable d. Proposez un calcul de θ_{Max} pour chaque cas.

5.6 Algorithmes inspirés de la programmation linéaire

Il est possible de définir un algorithme directement de l'analyse précédente. Nous en déduirons l'algorithme du gradient réduit. Par ailleurs, sans se préoccuper des partitions en variables de base et de super-base, nous pouvons quand même développer un algorithme basé sur le fait que x^* est solution de (5.3) ; c'est l'algorithme de Frank&Wolfe.

5.6.1 Algorithme de Frank&Wolfe

Revenons au problème (5.3). Supposons que l'on résolve ce problème en un point quelconque, qui n'est pas un minimum local du programme (5.1). Alors, le programme linéaire (5.3) possédera une solution y telle que $\nabla f(x)(y - x) < 0$, c'est-à-dire, la direction $(y - x)$ constitue une direction de descente réalisable pour le programme (5.1). Nommant d cette direction, nous allons choisir $x^+ \stackrel{\text{def}}{=} x + \theta d$ de sorte que θ minimise f sur la ligne joignant x et y, tout en demeurant réalisable. Alors, $f(x^+) < f(x)$, et nous pouvons recommencer le processus. Nous en déduisons la suite de quantités suivante :

$$\begin{aligned} y_k &\in \arg\min_y \nabla f(x_k)y : Ay = b, y \geqslant 0; \\ d_k &= y_k - x_k; \\ \theta_k &\in \arg\min_{0\leqslant\theta\leqslant 1} f(x_k + \theta d_k); \\ x_{k+1} &= x_k + \theta_k d_k. \end{aligned}$$

Les observations précédentes montrent que la suite $f(x_k)$ est décroissante. Pour justifier que les points d'accumulation $\bar{x}$ de la suite $\{x_k\}$ produite par l'algorithme sont des solutions satisfaisant $\nabla f(\bar{x})\bar{d} = 0$, nous appliquerons le théorème de Zangwill. Il est bien connu que $\theta = \arg\min_{0\leqslant\theta\leqslant 1} f(x + \theta(d))$ est une application fermée en fonction de x et d. De plus, l'exercice suivant montre que d_k est une application fermée en fonction de x_k. Par conséquent, le théorème de Zangwill s'applique.

L'algorithme de Frank&Wolfe utilise dans son calcul de direction la minimisation d'une fonction linéaire et donc résolue par un algorithme de programmation linéaire, i.e. $y \in \arg\min_{y\in E} \nabla f(x)y$ et $d = y - x$. Si on résout le programme linéaire avec l'algorithme du simplexe, la solution calculée sera toujours un point extrême de E et donc on peut toujours choisir $\theta_{Max} = 1$.

La condition d'optimalité est considérée satisfaite lorsque $\|\nabla f(x)(y-x)\|$ est assez petit.

***Exercice 5.6.1** *[Application fermée]* Une application A qui associe un ensemble $A(x)$ à un point x est dite *fermée* si pour toute suite convergente $x_k \to \bar{x}$, tout point d'accumulation $\bar{y}$ de la suite associée $y_k \in A(x_k)$ appartient à $A(\bar{x})$, $\bar{y} \in A(\bar{x})$. Démontrez que si $\nabla f(x)$ est une fonction continue, l'application A définie comme l'ensemble des solutions optimale du programme linéaire $\min_y \nabla f(x)y : Ay = b, y \geqslant 0$ est une application fermée.

Cet algorithme n'est plus beaucoup utilisé car sa vitesse de convergence est très lente. Cependant, il constitue un exemple d'extension de la théorie de la programmation linéaire. De plus, dans certaines applications particulières, notamment dans les problèmes de transport de grande taille, la simplicité et l'efficacité du calcul de la direction compensent sa lente convergence, et en font un algorithme acceptable, voire même l'algorithme de choix.

Exercice 5.6.2 *[Projections]* Considérons le problème de projeter l'origine sur l'enveloppe convexe de points de $\mathbb{R}^n$. On a un ensemble de points $P_1, P_2, \ldots, P_m$; l'enveloppe convexe est définie comme

$$\{x : x = \sum_{i=1}^{m} \alpha_i P_i, \sum_{i=1}^{m} \alpha_i = 1, \alpha \geqslant 0\}.$$

On peut formuler le problème comme suit :

$$\begin{aligned} &\min_{x\in\mathbb{R}^n,\alpha\in\mathbb{R}^m} \|x\|^2 \\ &\text{sujet à } x = \sum_{i=1}^m \alpha_i P_i = P\alpha \\ &\sum_{i=1}^m \alpha_i = 1 \\ &\alpha \geqslant 0, \end{aligned}$$

où les colonnes de la matrice P sont les points P_i.

a) En injectant l'expression pour x dans la fonction objectif, reformulez ce problème avec les seules variables α.

b) L'algorithme de Frank&Wolfe appliqué à cette dernière formulation devra calculer des solutions au problème

$$\begin{aligned} &\min_{y\in\mathbb{R}^m} \nabla f(\alpha_k) y \\ &\text{sujet à } \sum_{i=1}^m y_i = 1 \\ &y \geqslant 0; \end{aligned}$$

Obtenez une expression simple pour la solution de ce sous-problème.

c) Considérez le problème avec quatre points de $\mathbb{R}^2$ $P_1 = (0,5)^t$, $P_2 = (4,5)^t$, $P_3 = (4,1)^t$ et $P_4 = (5,-2)^t$. Obtenez la solution de ce problème graphiquement, et vérifiez-là avec les conditions d'optimalité.

d) Pour le problème en c), effectuez trois itérations de l'algorithme de Frank&Wolfe ; utilisez comme point de départ la valeur de α^0 correspondant à celui des quatre points le plus proche de l'origine : $P\alpha^0 = \arg\min_{i=1,2,3,4} \|P_i\|$.

5.6.2 Algorithme du gradient réduit

Cet algorithme utilise la partition Base—Super-Base—Hors-Base introduite ci haut. À l'aide de cette partition, récrivons (5.1) comme

$$\begin{aligned} \min\ & \bar{f}(x_S, x_N) = f([B^{-1}b - B^{-1}(Nx_N + Sx_S), x_S, x_N]) \\ \text{sujet à}\ & B^{-1}(b - Nx_N - Sx_S) \geqslant 0 \\ & x_S > 0 \\ & x_N = 0. \end{aligned} \tag{5.15}$$

Ce dernier programme se nomme *programme réduit*, et tant que ses contraintes demeurent satisfaites, on peut lui appliquer tout algorithme de minimisation d'une fonction sans contrainte. Lorsque l'algorithme de minimisation sans contrainte provoque la violation des contraintes $B^{-1}(b - Nx_N) \geqslant 0$, il est temps de changer de partition. L'algorithme du gradient réduit utilise comme algorithme sans contrainte l'algorithme du gradient, et formalise cette description intuitive.

On a pensé longtemps que l'algorithme du gradient réduit possédait la propriété de convergence globale. Puis, Wolfe a fourni ce contre-exemple :

$$\begin{aligned} \min_{x,y,z}\ & 4/3(x^2 - xy + y^2)^{3/4} - z \\ \text{sujet à}\ & z \leqslant 2, \quad x, y, z \geqslant 0. \end{aligned}$$

Pour cet exemple, on vérifie que l'algorithme, démarré en $(x_0, y_0, z_0)^t = (0, a, 0)^t$, où $0 \leqslant a \leqslant \sqrt{2}/4$, engendre une suite qui converge vers $(x_\infty, y_\infty, z_\infty)^t = (0, 0, (1 + \sqrt{2}/2)\sqrt{a})^t$, qui n'est ni la solution (exercice 5.4.4), ni un point stationnaire. C'est un exemple marquant des difficultés que causent la dégénérescence, lorsque la condition de stricte complémentarité n'est pas satisfaite.

***Exercice 5.6.3** *[Problème de convergence]* Vérifiez que l'exemple de Wolfe converge bel et bien vers le point décrit. Pour ce faire, vérifiez que l'algorithme engendre la suite

$$x_k = \begin{cases} (\alpha_k, 0, \beta_k)^t & \text{si } k \text{est impair} \\ (0, \alpha_k, \beta_k)^t & \text{si } k \text{est pair} \end{cases}$$

avec $\alpha_k = a(\frac{1}{2})^k$ et $\beta_k = \frac{1}{2}\sum_{j=0}^{k-1}\sqrt{\frac{a}{2^j}}$.

Il est possible de modifier l'algorithme pour assurer sa convergence globale, notamment en forçant $d_N = 0$ pour un certain nombre d'itérations consécutives. Cependant, une autre difficulté persiste, héritée de l'algorithme du simplexe : en présence de dégénérescence, l'algorithme peut rester emprisonné dans un cycle de partitions ne permettant aucun mouvement.

Gradient réduit

{ Données : une partition $[\![B \mid S \mid N]\!]$ de $\mathbb{R}^n$, une représentation }
{ de la matrice B^{-1} ainsi qu'une solution réalisable $[x_B, x_S, x_N]$. }
$d_N \leftarrow (-\nabla_N \bar{f}(x_S, x_N)^t)^+$
$d_S \leftarrow (-\nabla_S \bar{f}(x_S, x_N)^t)$
tantque ($\|d\| > 0$)
    $\theta_1 \leftarrow \max\{\theta : x_S + \theta d_S \geqslant 0\}$
    $\theta_2 \leftarrow \max\{\theta : B^{-1}(b - N(x_N + \theta d_N) - S(x_S + \theta d_S)) \geqslant 0\}$
    $\theta_3 \leftarrow \arg \min_{0 \leqslant \theta \leqslant \min\{\theta_1, \theta_2\}} \bar{f}(x_S + \theta d_S, x_N + \theta d_N)$
    $x_N \leftarrow x_N + \theta_3 d_N$
    $x_S \leftarrow x_S + \theta_3 d_S$
    $x_B \leftarrow B^{-1}(b - N x_N - S x_S)$
    { Mise-à-jour de la partition }
    si ($d_N \neq 0$) **alors** { certaines composantes passent de N à S }
    si ($\theta_3 = \theta_1$) **alors** { certaines composantes passent de S à N }
    si ($\theta_3 = \theta_2$) **alors**
        { UNE composante passe de B à N }
        { UNE composante passe de N ou S à B }
        { La représentation de la matrice B^{-1} est mise à jour. }
    $d_N \leftarrow (-\nabla_N \bar{f}(x_S, x_N)^t)^+$
    $d_S \leftarrow (-\nabla_S \bar{f}(x_S, x_N)^t)$

Voici l'algorithme classique du gradient réduit. L'opérateur de projection $(-\nabla \bar{f}(x_S, x_N)^t)^+_N$ pour le calcul de d_N consiste à remplacer par 0 les composantes négatives de $\nabla \bar{f}_N$. La mise-à-jour de la partition dans le cas où une composante quitte B s'effectue en utilisant la même mécanique algébrique que l'algorithme du simplexe révisé

Algorithme 5.2: Gradient réduit.

Exercice 5.6.4 *[GR]* Considérez à nouveau le problème

$$\min_{x\in\mathbb{R}^3} \frac{1}{2}\|x\|^2$$
$$\text{sujet à } x_1 + 2x_2 + 3x_3 = 1; \qquad x \geqslant 0.$$

Appliquez l'algorithme du gradient réduit à partir du point de départ $x^0 = (1,0,0)^t$.

5.7 Algorithmes de projection

Nous avons vu que les points stationnaires étaient également des points fixes de l'application $Proj_E(x-\nabla f(x)^t)$. Si on a sous la main un point x pour lequel $x \neq Proj_E(x-\nabla f(x)^t)$, il est possible de construire une direction descendante $d = Proj_E(x-\nabla f(x)^t) - x$. L'inconvénient de cette approche vient de la nécessité du calcul de la projection. Si les contraintes sont simples, par exemple des bornes sur les variables, alors il est possible de calculer la projection efficacement, et l'algorithme présente un certain intérêt.

L'algorithme de projection consiste à calculer $y = Proj_E(x-\nabla f(x)^t)$ et constituer la direction $d = y - x$. Dans ce cas, il est avantageux de calculer un θ_{Max} car il est fort possible que l'on puisse utiliser une valeur de θ beaucoup plus grande que un, ce qui permettra une convergence beaucoup plus rapide.

La condition d'optimalité est considérée satisfaite lorsque $\|Proj_E(x-\nabla f(x)) - x\|$ est assez petit.

***Exercice 5.7.1** *[Direction de projection]* Vérifiez que $d = Proj_E(x-\nabla f(x)^t) - x$ est une direction de descente, c'est-à-dire que $\nabla f(x)d < 0$.

5.7.1 Exemple de calculs de projection

Si l'ensemble E est constitué de simples bornes $x \geqslant 0$, alors $Proj_E(x) = \begin{cases} 0 & \text{si } x \leqslant 0 \\ x & \text{si } x > 0 \end{cases}$.
Nous développons maintenant une technique pour calculer la projection pour les contraintes

$$\sum_{i=1}^{n} x_i = 1 \quad x \geqslant 0.$$

Ceci constitue un exemple de résolution des conditions d'optimalité. Écrivons donc les conditions d'optimalité du programme qui définit la projection :

$$\begin{aligned} \min_x \ & \frac{1}{2}\|x-y\|^2 \\ \text{sujet à } & \sum_{i=1}^n x_i = 1 \\ & x \geqslant 0. \end{aligned} \tag{5.16}$$

Les conditions s'écrivent

$$\begin{aligned} (x-y)^t + e\lambda - \mu &= 0 \\ ex &= 1 \\ \mu x &= 0 \\ \mu &\geqslant 0 \end{aligned}$$

Maintenant, de la première équation, on tire que $x_i = y_i - \lambda + \mu_i$. Supposons que l'on connaisse λ. Alors, si $y_i - \lambda < 0$, $x_i = 0$ et $\mu_i = \lambda - y_i$. Autrement, $y_i - \lambda \geqslant 0$ et $x_i = y_i - \lambda$, $\mu_i = 0$. Nommons $x_i(\lambda) = \max(0, y_i - \lambda)$. Il suffit maintenant d'identifier une valeur de λ pour laquelle $ex(\lambda) = 1$. Or, chaque $x_i(\lambda)$ est une fonction linéaire par morceau, comportant deux morceaux qui se joignent au point $\lambda = y_i$, et non-croissante en fonction de λ. Donc, $e\lambda$ est une fonction non-croissante de λ linéaire par morceaux, dont les points de changement de morceaux sont les valeurs y_i. Il suffit donc de classer les points de changement de morceau y_i par ordre disons croissant $i_1, i_2, \ldots, i_n$, puis d'identifier le morceau tel que $ex(y_{i_j}) \geqslant 1 \geqslant ex(y_{i_{j+1}})$, et enfin d'effectuer une interpolation linéaire sur ce morceau.

Exercice 5.7.2 *[Projection]* Considérez un exemple dans $\mathbb{R}^4$. Soit $y = \frac{1}{4}(-1, -2, 2, 4)^t$. Calculez la projection décrite par le programme (5.16).

Exercice 5.7.3 *[Programmes séparables]* La solution de la projection présentée ci-haut peut se généraliser à des programmes quadratiques dont la matrice est une matrice diagonale Δ telle que $\Delta_{ij} = 0$ si $i \neq j$. Considérez le problème

$$\begin{aligned} \min_x \ & \frac{1}{2}x^t\Delta x + cx \\ \text{sujet à } & \sum_{i=1}^n a_i x_i = 1 \\ & l_i \leqslant x_i \leqslant u_i. \end{aligned}$$

On peut se limiter à des valeurs $a_i \geqslant 0$; en effet, si certains $a_i < 0$, proposez un problème équivalent avec $a_i > 0$. (Indice : on peut remplacer x_i par $-x_i$) Imitez la démarche pour proposer une technique de solution ; décrivez les conditions d'optimalité ; obtenez une expression $x_i(\lambda)$; indiquez quels sont les points de changement de morceau ; vérifiez que la fonction $s(\lambda) \stackrel{\text{def}}{=} \sum_{i=1}^n a_i x_i(\lambda)$ est monotone.

Exercice 5.7.4 *[Variante de contrainte simple]* Un certain modèle de gestion comporte un sous problème de la forme

$$\begin{aligned} \min \quad & \sum_{i=1}^{3} \left(h_i \frac{x_i}{2} + S_i \frac{d_i}{x_i} \right) \\ & \sum_{i=1}^{3} x_i = Q \\ & x \geqslant 0 \end{aligned}$$

où x représente le vecteur de production (des trois items), h le vecteur des coûts d'inventaire, S les coûts fixes de production, d la demande et enfin Q la production totale.

a) En définissant le Lagrangien *seulement par rapport à la contrainte d'égalité*,

$$L(x, \lambda) = \sum_{i=1}^{3} \left(h_i \frac{x_i}{2} + S_i \frac{d_i}{x_i} \right) + \lambda \left(\sum_{i=1}^{3} x_i - Q \right),$$

vérifiez que si l'on connaît λ^*, alors x^* minimise $L(x, \lambda^*)$ sujet à $x \geqslant 0$.

b) Soient $h = (1, 1, 2)^t$, $S = (100, 50, 400)^t$, $d = (20000, 40000, 40000)^t$ et $Q = 6000$. Obtenez une expression pour $x(\lambda)$.

c) Par bissection, estimez λ^*, et donc x^*.

d) À partir de $x_i = Q/3$, effectuez 3 itérations de l'algorithme de Frank&Wolfe.

5.8 Algorithmes de contraintes actives

Les conditions d'optimalité de K–K–T distinguent (ne serait-ce que par le terme de complémentarité) les contraintes actives des contraintes inactives. Si l'on connaissait l'ensemble des contraintes actives à la solution, on pourrait résoudre le problème à l'aide d'une méthode sans contrainte appliquée au problème $\phi(d_z) = f(x_0 + Zd_z)$.

L'idée de base de ces méthodes est la suivante. On démarre en un certain point x_0. Nommons W, l'ensemble des contraintes saturées en x_0, ensemble de travail. Nous espérons que l'ensemble de travail deviendra égal à l'ensemble des contraintes actives en x^*. À l'aide de W, nous définissons un sous-problème avec seulement des contraintes d'égalités. Nous appliquons à ce problème un algorithme d'optimisation sans contrainte. L'algorithme d'optimisation sans contrainte est interrompu dans deux circonstances :

- la progression de l'algorithme sans contrainte est empêchée par la rencontre d'une ou plusieurs contraintes ;
- une solution au sous-problème sans contrainte est atteinte.

Dans le premier cas, on doit ajouter des contraintes dans l'ensemble W (choisies parmi celles rencontrées par l'algorithme) alors que dans le second cas, on doit retirer des contraintes de l'ensemble W (choisies parmi celles dont les variables λ_i associées sont négatives).

Plus précisément, on distingue deux types d'itérations : des itérations de *restriction* et de *relaxation*. On parle de restriction lorsque l'on applique un algorithme avec seulement les contraintes d'égalité de W. On parle de relaxation lorsque l'on enlève des éléments de W.

```
Contraintes actives
  { Données : un point réalisable x un ensemble de travail W, }
  { une matrice Z telle que A_W Z = 0 ; }
  { deux constantes τ_0 et τ_1 ; }
  tantque ( ¬ ≈ Optimal(x) )
    { Étape de restriction }
        y ← x
        tantque (condition de relaxation NON satisfaite )
          d ← Direction de descente dans ker A_W
          θ_max ← max_θ{A(y + θd) ⩽ b}
          θ_adm ← pas admissible (définition 3.3.2)
          si (θ_max < θ_adm ) alors W ← W ∪ {i : [A_i y < b_i] ∧ [A_i(y + θ_max d) = b_i]}
          y ← y + min{θ_max, θ_adm}d
        x ← y
    { Étape de relaxation }
        W ← W réduit
```

Algorithme 5.3: Contraintes actives.

Remarquons que ce schéma d'algorithme peut très bien être implanté à l'aide du schéma de l'algorithme 5.2. Dans ce dernier algorithme, on implante une étape de restriction en forçant $d_N = 0$; on rencontre une nouvelle contrainte et ajoute un ou des éléments à W

```
Contraintes actives—programme quadratique convexe
{ Données : c et Q ≻ 0, }
{ Données : un point réalisable x un ensemble de travail W, }
{ une matrice Z telle que A_W Z = 0 ; }
Optimal ← faux
tantque ( ¬ Optimal )
    { Étape de restriction }
        y ← x
        tantque (‖(Z^t(Qx + c^t))‖ > ϵ )
            d ← Z(Z^t Q Z)^{-1}(Z^t(Qx + c^t))
            θ_max ← max_θ{A(y + θd) ⩽ b}
            si (θ_max < 1 ) alors
                W ← W ∪ {i : [A_i y < b_i] ∧ [A_i(y + θ_max d) = b_i]}
                Z ← ker A_W
            y ← y + min{θ_max, 1}d
        x ← y
    { Étape de relaxation }
        λ ← arg min_λ ‖xQ + c + λA_W‖^2
        i* ← arg min_j{λ_j}
        si (λ_i* ⩾ ϵ ) alors Optimal ← vrai
        sinon W ← W\i*
```

Algorithme 5.4: Contraintes actives pour programmes quadratiques.

lorsque $\theta_3 = \min(\theta_1, \theta_2)$ et on atteint une solution au sous-problème sans contrainte lorsque $\|d_S\| \approx 0$ et $\|d_B\| \approx 0$, dans quel cas on enlève une contrainte de W en enlevant un indice de N. On voit donc que les contraintes de travail correspondent en fait aux composante nulles du vecteur x. Une particularité de cette implantation vient du fait que lorsqu'une ou plusieurs composantes de B devient nulle, la mécanique de l'algorithme du simplexe nous impose d'en introduire une seule dans l'ensemble N. Une autre particularité est que le choix de l'algorithme restreint est celui de la pente la plus forte alors qu'il serait avantageux de calculer une direction de Newton modifiée (section 3.4) plutôt que d'utiliser $d_S = -\nabla \bar{f}_S$.

Le parallèle entre l'algorithme du gradient réduit et l'algorithme de contraintes actives soulève le problème de la convergence globale de ce type d'algorithme. En effet, si la condition de relaxation est trop permissive et la stratégie de réduction de l'ensemble W trop libérale, on peut retrouver un algorithme *équivalent* à l'algorithme 5.2, et donc permettre la possibilité de convergence vers un point non-stationnaire. Néanmoins, la possibilité d'utiliser la mécanique algébrique de l'algorithme du simplexe rends ce type d'algorithmes attrayant.

5.8.1 Quelques aspects pratiques

Enfin, l'algorithme de contraintes actives enrichit le schéma simple 5.1 en maintenant un ensemble d'indice W des contraintes supposées actives. Dans ce cas, on distingue deux types d'itérations :

1. les itérations de restriction dans lesquelles on calcule une direction à l'aide d'un programme avec contraintes d'égalités seulement, $A_W x = b_W$; on dénote alors par Z_W une matrice dont les colonnes engendrent $\ker A_W$.
2. les itérations de relaxation, que l'on se permet sous certaines conditions, et qui permettent de simplement enlever des indices de l'ensemble W.

Au point courant x, en constituant la fonction $\phi(\alpha) = f(x + Z_W\alpha)$, n'importe quelle direction de descente pour ϕ en $\alpha = 0$ permet d'obtenir une direction de descente réalisable pour f. Par exemple,

- la direction opposée au gradient $d_\alpha = -\nabla\phi(0)^t = -Z_W^t \nabla f(x)^t$ fournit la direction de descente $d = Z_W d_\alpha = -Z_W Z_W^t \nabla f(x)^t$;
- la direction de Newton $d_{N\alpha} = -\nabla^2\phi(0)\nabla\phi(0)^t = -(Z_W^t \nabla^2 f(x) Z_W)^{-1} Z_W^t \nabla f(x)^t$ fournit la direction de Newton $d_N = Z_W d_{N\alpha}$.

En fait, pour minimiser $\phi(\alpha)$, on dispose de toutes les méthodes du chapitre 3.

Si une condition de relaxation devient satisfaite, on se permet d'enlever un indice de W. Dans cet algorithme de contraintes actives, si jamais $\theta = \theta_{Max}$, c'est le signe que le pas de déplacement θ a été raccourci pour éviter de violer une contrainte, et donc l'indice de cette contrainte est ajouté dans W et on continue avec des itérations de restrictions avec ce nouvel ensemble W.

La condition d'optimalité est considérée satisfaite dès que $\|\nabla f(x) Z_W\|$ est assez petit, ce qui est équivalent à ce que le minimum par rapport à λ_W de $\|\nabla f(x) + \lambda_W A_W\|$ soit assez petit <u>et que</u> les λ_W qui minimisent $\|\nabla f(x) + \lambda_W A_W\|$ sont du bon signe.

Fonctions objectifs quadratiques Remarquons que lorsque la fonction objectif est quadratique et convexe, la direction de Newton conduit directement au minimum sous la contrainte $A_W x = b_W$. S'il faut raccourcir la direction, on ajoutera un indice dans W. Autrement, la condition de relaxation est satisfaite au point $x + d_N$ par construction de la direction de Newton. L'algorithme 5.4 exploite cette observation.

Dans les cas des directions de Frank & Wolfe ou de projection, on peut obtenir le minimum de la fonction $f(x + \theta d)$ par une formule explicite et ce minimum est un pas admissible.

Exercice 5.8.1 *[Newton pour f quadratique]*

a) Pour une fonction f quadratique et convexe (Q définie positive), démontrez que si le point $x^+ = x + d_N$ est réalisable, alors il existe des multiplicateurs de Lagrange λ_W tels que $\nabla f(x^+) + \lambda_W A_W = 0$;

b) que peut-on dire si la fonction quadratique n'est pas convexe ?

Exercice 5.8.2 *[Pas admissible pour f quadratique]*

a) obtenez une formule pour le minimum selon θ de $f(x + \theta d)$ lorsque f est une fonction quadratique convexe, i.e. lorsque la matrice Q définissant la fonction quadratique est définie positive ;

b) vérifiez que ce minimum est un pas admissible ;

c) que peut-on dire si la fonction quadratique n'est pas convexe et que d est une direction de descente réalisable ?

Exercice 5.8.3 *[Application simple]* Considérez le problème

$$\begin{array}{rc} \min & x_1^2 + 2x_2^2 + x_3^2 - 2x_1 - 4x_2 + 2x_3 \\ \text{s.à} & x_1 + x_2 + x_3 \leqslant 1 \\ & x \geqslant 0 \end{array}$$

a) En supposant que la contrainte $ex \leqslant 1$ est satisfaite avec égalité, obtenez la solution avec la technique de la section 5.7.

b) À partir de l'origine, appliquez l'algorithme des contraintes actives pour obtenir la solution.

Exercice 5.8.4 *[Application simple]* Considérez le programme

$$\begin{array}{rcl} \min f(x) & = & \dfrac{x_1^2 + x_2^2}{2} + x_3^2 - 10x_1 - \dfrac{3x_2}{4} - \dfrac{x_3}{2} \\ \text{sujet à} & & 0 \leqslant x_1 \leqslant 1, \\ & & 0 \leqslant x_2 \leqslant 1, \\ & & 0 \leqslant x_3 \leqslant 1. \end{array}$$

À partir de l'origine,

a) en utilisant l'algorithme des contraintes actives, obtenez la solution ; puisque la fonction objectif est quadratique, pour un ensemble W donné, on peut calculer une solution exacte, et le vecteur allant de la solution courante à cette solution exacte peut être utilisé comme direction de descente dans $\ker A_w$;

b) effectuez 2 itérations de l'algorithme de Frank & Wolfe ;

c) effectuez 2 itérations de l'algorithme de projection.

5.8.2 Un exemple d'algorithme globalement convergent

L'algorithme 5.5 présente une variante possible qui précise une condition de relaxation, un critère d'arrêt basé sur l'optimalité, une direction de Newton sur les contraintes actives et finalement une stratégie de relaxation. Si le problème de cyclage ne se présente pas, alors les points d'accumulations de cette variante sont des points stationnaires.

Condition de relaxation

Parmi les différentes conditions de relaxation proposées dans la littérature, nous avons retenu la plus simple : aussitôt qu'un pas admissible (au sens de la définition 3.3.2) a pu être utilisé, on se permet d'enlever des composantes de W, c'est-à-dire de permettre à des composantes nulles de devenir positives.

Critère d'arrêt

La norme du gradient réduit sert de critère d'arrêt.

Direction de descente

Puisque la réduction rendue possible par la base B transforme le problème original en un problème sans contrainte, et puisque pour ce nouveau problème, les variables S sont libres alors que les variables N sont fixées à zéro dans les étapes de restriction, nous utilisons la direction de Newton dans les composantes S. En fait, il faut utiliser une direction de Newton modifiée (voir la section 3.4) pour s'assurer d'avoir une direction de descente.

Stratégie de relaxation

Comme l'exemple de Wolfe l'illustre, il faut se montrer prudent quant à la stratégie de relaxation. Nous avons retenu la plus simple et la plus usuelle, c'est-à-dire celle de permettre à la variable ayant la composante de $\nabla_N \bar{f}(x_S, x_N)$ la plus négative de devenir positive.

Variantes

En fait, les variantes spécifiques à cet algorithme concernent surtout les aspects reliés à la relaxation : les variantes de la méthode de Newton et les tests d'arrêt ayant déjà été discutées au chapitre 3.

Parmi d'autres conditions de relaxation, mentionnons l'utilisation d'une suite η_k convergeant vers zéro qui sert de critère d'arrêt pour la boucle de restrictions. Avec cette stratégie, on tente de réduire $\|d_S\|$ en deçà du seuil η_k à l'itération k.

Pour ce qui est des stratégies, il est possible de permettre à plusieurs composantes de l'ensemble N de devenir positives. Une possibilité permet aux indices atteignant un certain pourcentage fixé du minimum de $\nabla_N \bar{f}(x_S, x_N)$ d'être considérés pour devenir positifs. Une autre stratégie consiste à utiliser la direction de projection (voir section 5.7) pour déterminer les composantes pouvant être augmentées.

5.9 Résumé

Les problèmes avec contraintes linéaires d'égalité se réduisent aisément à des problèmes sans contrainte en utilisant les propriétés algébriques des contraintes linéaires. Les algorithmes d'optimisation sans contrainte s'adaptent aisément à ce cas.

En présence de contraintes d'inégalité, un phénomène combinatoire se manifeste : identifier lesquelles des inégalités seront saturées à la solution (ou en un point candidat à la solution). Si l'on connaît les contraintes saturées, aussi nommées contraintes actives, le problème se ramène à un problème avec contraintes linéaires d'égalité. Certaines analyses manipulent cet aspect combinatoire de manière explicite (généralisations de la programmation linéaire, conditions de K–K–T, algorithmes de contraintes actives) alors que d'autres dissimulent la difficultés (condition 5.3 et algorithme de Frank&Wolfe, conditions d'optimalité et algorithmes de projection).

Les algorithmes les plus répandus utilisent la mécanique algébrique du simplexe pour implanter des variantes de la méthode de Newton se restreignant sur un ensemble de contraintes

Gradient réduit amélioré

{ Données : une partition $[\![B \mid S \mid N]\!]$ de $\mathbb{R}^n$, une représentation }
{ de la matrice B^{-1} ainsi qu'une solution réalisable $[x_B, x_S, x_N]$. }
$d_N \leftarrow (-\nabla_N \bar{f}(x_S, x_N)^t)^+$
$d_S \leftarrow -(\nabla^2_{SS}\bar{f}(x_S, x_N))^{-1}\nabla_S \bar{f}(x_S, x_N)^t$
$Pas\_admissible \leftarrow$ **faux**
tantque ($\|d\| > 0$)
    { Restriction et/ou relaxation... }
        si ($\neg$ $Pas\_admissible$) **alors** $d_N \leftarrow 0$
        sinon
            { Permettre à la composante la plus grande de d_N de quitter N. }
            $i \leftarrow \arg\min_j\{d_{Nj}\}$
            $d_N \leftarrow d_{Ni}e_i$
            $Pas\_admissible \leftarrow$ **faux**
    { Calcul du pas }
        $\theta_1 \leftarrow \max\{\theta : x_S + \theta d_S \geqslant 0\}$
        $\theta_2 \leftarrow \max\{\theta : B^{-1}(b - N(x_N + \theta d_N) - S(x_S + \theta d_S)) \geqslant 0\}$
        $\theta_{adm} \leftarrow$ pas admissible (définition 3.3.2)
        $\theta_3 \leftarrow \min\{\theta_1, \theta_2, \theta_{adm}\}$
        si ($\theta_{adm} = \theta_3$) **alors** $Pas\_admissible \leftarrow$ **vrai**
    { Mise-à-jour de x }
        $x_N \leftarrow x_N + \theta_3 d_N$
        $x_S \leftarrow x_S + \theta_3 d_S$
        $x_B \leftarrow B^{-1}(b - Nx_N - Sx_S)$
    { Mise-à-jour de la partition }
        si ($d_N \neq 0$) **alors** { certaines composantes passent de N à S }
        si ($\theta_3 = \theta_1$) **alors** { certaines composantes passent de S à N }
        si ($\theta_3 = \theta_2$) **alors**
            { UNE composante passe de B à N }
            { UNE composante passe de N ou S à B }
            { La représentation de la matrice B^{-1} est mise à jour. }
    $d_N \leftarrow (-\nabla_N \bar{f}(x_S, x_N)^t)^+$
    $d_S \leftarrow -(\nabla^2_{SS}\bar{f}(x_S, x_N))^{-1}\nabla_S \bar{f}(x_S, x_N)^t$

Algorithme 5.5: Gradient réduit amélioré.

de travail qui, on l'espère, se confondra avec l'ensemble des contraintes actives. La convergence globale de tels algorithmes est délicate, mais des implantations soignées donnent depuis longtemps d'excellents résultats.

5.10 Extensions et références

5.11 Tous les exercices du chapitres

Exercice (5.3.1, page 227) *[Résoudre (5.4)]* Vérifiez que $x^* = x_0 + Z\alpha^*$ est bel et bien un minimum local de (5.4).

Exercice (5.3.2, page 227) *[Matrices Z]* Soient les contraintes linéaires suivantes.

$$A = \begin{pmatrix} 3 & -5 & 1 & 1 \\ 3 & 1 & -5 & 1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 3 \end{pmatrix}.$$

Soient également les quatre candidats matrices Z :

$$Z_1 = \begin{pmatrix} 3 & -3 \\ 3 & -3 \\ 3 & -3 \\ 3 & -3 \end{pmatrix} \quad Z_2 = \begin{pmatrix} -1 & 1 & 0 \\ -2 & 0 & -1 \\ -2 & 0 & -1 \\ -5 & -3 & -4 \end{pmatrix} \quad Z_3 = \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 3 \end{pmatrix} \quad Z_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Une seule de ces matrices Z a la propriété que ses colonnes constituent une base de $\ker A$. Identifiez laquelle et indiquez clairement pourquoi les trois autres n'ont pas la propriété voulue.

Exercice (5.3.3, page 228) *[Application simple]* Soit le programme

$$\begin{array}{rrcl} & \min f(x,y,z) & = & x^2 + 2y^2 + 3z^2 - 0.5xy + x - y + 2z \\ \text{sujet à} & 2x + 3y - z & = & 1. \end{array}$$

Identifiez un point stationnaire en utilisant trois techniques :

a) Calculez un multiplicateur de Lagrange λ^* et un point $(x^*, y^*, z^*)^t$ satisfaisant au système d'équations $\nabla f(x,y,z) + \lambda(2,3,-1) = 0$.
b) Éliminez une variable à l'aide de l'équation $2x + 3y - z = 1$ en posant $z = 2x + 3y - 1$ et traitez les deux variables qui restent sans contrainte.
c) Éliminez une variable en identifiant une base du noyau de la matrice $(2, 3, -1)$, et obtenez la solution.
d) Vérifiez que les trois méthodes fournissent la même solution.

e) Quelle est la nature du point stationnaire ? (min, max, pt de selle, ?)

Exercice (5.3.4, page 228) *[Contraintes d'égalité]* Considérez le problème

$$\min_{x\in\mathbb{R}^3} \frac{1}{2}\|x\|^2$$
$$\text{sujet à } x_1 + 2x_2 + 3x_3 = 1.$$

a) Fournissez une expression pour la matrice Z.

b) Identifiez la solution x^* et λ^* du problème.

c) Vérifiez que $\nabla f(x^*)Z = 0$.

d) Vérifiez que $x^* = Proj_E(x^* - \nabla f(x^*))$.

e) Résolvez le problème

$$\min_{x\in\mathbb{R}^n} \|x\|^2$$
$$\text{sujet à } \sum_{i=1}^{n} i x_i = 1.$$

Exercice (5.3.5, page 229) *[Géométrie — condition d'optimalité]* Trouvez le rectangle de superficie maximale pour un périmètre donné ($2(h + l) = 1$) en utilisant les conditions nécessaires. Vérifiez que votre solution satisfait aux conditions suffisantes de second ordre.

Exercice (5.3.6, page 229) *[Projection sur un sous-espace]* On sait que la projection sur un sous-espace vectoriel est une application linéaire, et donc peut se représenter par une matrice. Nous appliquerons la théorie pour obtenir cette matrice. Considérez le problème de projeter un vecteur $-g$, l'opposé du gradient sur un sous-espace défini par $x : Ax = 0$ où la matrice A est de plein rang, donc ses lignes sont linéairement indépendantes. Le problème s'écrit alors :

$$\begin{array}{rl} \min & \frac{1}{2}\|x + g^t\|^2 \\ \text{sujet à} & Ax = 0 \end{array}$$

a) Écrivez les conditions d'optimalité de ce problème.

b) Exprimez la solution $x(\lambda)$ explicitement en fonction du vecteur de multiplicateurs de Lagrange λ.

c) Injectez l'expression trouvée en b). dans les contraintes pour déterminer λ^*.

d) Obtenez alors l'expression de $x(\lambda^*) = x^* = -Mg^t$.

e) Vérifiez que la matrice de projection M est symétrique, et satisfait $M^2 = M$.

Exercice (5.3.7, page 232) *[Conditions générales]* Démontrez le corollaire 5.3.1 en convertissant le programme (5.6) avec uniquement des contraintes d'inégalités non-positives puis en appliquant le théorème 5.3.2.

Exercice (5.3.8, page 232) *[Conditions générales pour un max]* Démontrez le corollaire 5.3.2 en convertissant le programme (5.7) en un problème de minimisation avec uniquement des contraintes d'inégalités non-positives.

Exercice (5.4.1, page 243) *[Contraintes d'inégalités]*

a) Vérifiez que la solution obtenue à l'exercice 5.3.4 constitue la solution optimale du problème

$$\begin{gathered}\min_{x\in\mathbb{R}^3} \|x\|^2 \\ \text{sujet à } x_1 + 2x_2 + 3x_3 = 1; \qquad x \geqslant 0.\end{gathered}$$

b) Que peux-t-on dire de cette dernière solution par rapport au problème

$$\begin{gathered}\min_{x\in\mathbb{R}^3} \|x\|^2 \\ \text{sujet à } x_1 + 2x_2 + 3x_3 \leqslant 1; \qquad x \geqslant 0.\end{gathered}$$

c) Que peux-t-on dire de cette dernière solution par rapport au problème

$$\begin{gathered}\min_{x\in\mathbb{R}^3} \|x\|^2 \\ \text{sujet à } x_1 + 2x_2 + 3x_3 \geqslant 1; \qquad x \geqslant 0.\end{gathered}$$

Exercice (5.4.2, page 244) *[Application simple]* Pour les problèmes suivants, obtenez la solution.

a)

$$\begin{array}{rl} \min & -\left(x_1^2 - x_1x_2 + 2x_2^2 + x_1 + x_2\right) \\ \text{sujet à} & 2x_1 + x_2 \leqslant 1 \\ & x \geqslant 0 \end{array}$$

b)

$$\begin{array}{rl} \min & -\left(x_1^2 - 2x_1x_2 + 2x_2^2 + 2x_1 + 5x_2\right) \\ \text{sujet à} & 2x_1 + 3x_2 \leqslant 20 \\ & 3x_1 - 5x_2 \leqslant 4 \\ & x \geqslant 0 \end{array}$$

Exercice (5.4.3, page 244) *[Projection...]* Considérez le problème de trouver le point de norme minimale dans l'ensemble défini par les contraintes suivantes :

$$\begin{aligned} x_1 + 2x_2 - x_3 &\geqslant 4 \\ -x_1 + x_2 - x_3 &\leqslant 2. \end{aligned}$$

a) Formulez ce problème comme problème de minimisation.
b) Supposons que les deux contraintes sont actives à la solution. En exprimant x_1 et x_2 en fonction de x_3, obtenez la solution du problème avec contraintes d'égalités.
c) Cette solution ne constitue pas le point de norme minimale dans l'ensemble réalisable original. Pourquoi ? Trouvez le point de norme minimale dans les contraintes d'inégalité.

***Exercice (5.4.4, page 244)** *[PNL]* Obtenez la solution du problème

$$\begin{gathered} \min_{x,y,z} 4/3(x^2 - xy + y^2)^{3/4} - z \\ \text{sujet à } z \leqslant 2, \quad x, y, z \geqslant 0. \end{gathered}$$

Indice : prolongez à $(0, 0, 0)^t$ le gradient de la fonction par continuité : le gradient n'existe pas à l'origine, mais la limite lorsque x s'approche de l'origine existe. Est-ce que la condition de stricte complémentarité est satisfaite en x^* ?

Exercice (5.4.5, page 245) *[Projection]* Soit $E = \{x : Ax - b \leqslant 0\}$ un ensemble réalisable. Vérifiez que $\bar{x}$ est un point fixe de l'opérateur de projection, $\bar{x} = Proj_E(\bar{x} - \nabla f(\bar{x})^t)$, si et seulement si $\bar{x}$ satisfait aux conditions nécessaires de premier ordre de K–K–T du problème (5.5).

Exercice (5.4.6, page 245) *[Caractérisation de la projection]* Si les contraintes sont de simples bornes sur les variables, i.e. $L \leqslant x \leqslant U$, vérifiez (en utilisant les conditions d'optimalité) que la solution de $\min_{L \leqslant x \leqslant U} \frac{1}{2}\|x - y\|^2$ est donnée par

$$x_i = \begin{cases} L_i & \text{si } L_i \geqslant y_i; \\ y_i & \text{si } L_i \leqslant y_i \leqslant U_i; \\ U_i & \text{si } y_i \geqslant U_i. \end{cases}$$

Exercice (5.4.7, page 245) *[Simplification de condition d'optimalité]* Prouvez, ou donnez un contre exemple :

Si x^* est un min local pour

$$\begin{array}{ll} \min & f(x) \\ \text{sujet à} & Ax \leqslant b \\ & x \geqslant 0 \end{array}$$

alors il existe $\lambda^* \geqslant 0$ tel que $\nabla f(x^*) + \lambda^* A \geqslant 0$, avec $\lambda^*(Ax^* - b) = 0$.

Exercice (5.4.8, page 245) *[Conditions d'optimalité]* Soient les trois fonctions suivantes :

- $f_1(x) = \frac{1}{2}((x_1 - 3)^2 + (x_2 + 1)^2)$;
- $f_2(x) = \frac{1}{2}((x_1 + 1)^2 + (x_2 - 1)^2)$;
- $f_3(x) = \frac{1}{2}((x_1 + 1)^2 + x_2^2)$.

Définissons un point stationnaire $\hat{x}$ comme étant un point réalisable pour lequel il existe des multiplicateurs de Lagrange, nuls pour les contraintes satisfaites avec inégalité stricte en $\hat{x}$, et tous du bon signe pour les contraintes actives en $\hat{x}$.

a) Considérez les (quatre) contraintes

$$0 \leqslant x_1 \leqslant 1,$$
$$0 \leqslant x_2 \leqslant 1,$$

définissant l'ensemble réalisable E, et le point candidat $x = (1,0)^t \in E$.

i) Pour chacune des fonctions, vérifiez que le point candidat est un point stationnaire, et donnez les valeurs des multiplicateurs de Lagrange $\lambda_1, \lambda_2, \lambda_3$ et λ_4.

ii) Pour chacune des fonctions, déterminez si le point stationnaire est un minimum local, maximum local ou ni l'un ni l'autre.

Exercice (5.5.1, page 246) *[Calcul de θ_{Max}]* Considérez deux formes pour les ensembles E,

a) $E = \{x : Ax = b, x \geqslant 0\}$;

b) $E = \{x : Ax \leqslant b\}$.

Dans ces deux cas, soient un point réalisable $x \in E$ et une direction réalisable d. Proposez un calcul de θ_{Max} pour chaque cas.

***Exercice (5.6.1, page 248)** *[Application fermée]* Une application A qui associe un ensemble $A(x)$ à un point x est dite *fermée* si pour toute suite convergente $x_k \to \bar{x}$, tout point d'accumulation $\bar{y}$ de la suite associée $y_k \in A(x_k)$ appartient à $A(\bar{x})$, $\bar{y} \in A(\bar{x})$. Démontrez que si $\nabla f(x)$ est une fonction continue, l'application A définie comme l'ensemble des solutions optimale du programme linéaire $\min_y \nabla f(x) y : Ay = b, y \geqslant 0$ est une application fermée.

Exercice (5.6.2, page 248) *[Projections]* Considérons le problème de projeter l'origine sur l'enveloppe convexe de points de $\mathbb{R}^n$. On a un ensemble de points $P_1, P_2, \ldots, P_m$; l'enveloppe convexe est définie comme

$$\{x : x = \sum_{i=1}^{m} \alpha_i P_i, \sum_{i=1}^{m} \alpha_i = 1, \alpha \geqslant 0\}.$$

On peut formuler le problème comme suit :

$$\begin{aligned}\min_{x\in\mathbb{R}^n,\alpha\in\mathbb{R}^m} &\|x\|^2\\ \text{sujet à } x = &\sum_{i=1}^m \alpha_i P_i = P\alpha\\ &\sum_{i=1}^m \alpha_i = 1\\ &\alpha \geqslant 0,\end{aligned}$$

où les colonnes de la matrice P sont les points P_i.

a) En injectant l'expression pour x dans la fonction objectif, reformulez ce problème avec les seules variables α.

b) L'algorithme de Frank&Wolfe appliqué à cette dernière formulation devra calculer des solutions au problème

$$\begin{aligned}\min_{y\in\mathbb{R}^m} &\nabla f(\alpha_k) y\\ \text{sujet à } &\sum_{i=1}^m y_i = 1\\ &y \geqslant 0;\end{aligned}$$

Obtenez une expression simple pour la solution de ce sous-problème.

c) Considérez le problème avec quatre points de $\mathbb{R}^2$ $P_1 = (0, 5)^t$, $P_2 = (4, 5)^t$, $P_3 = (4, 1)^t$ et $P_4 = (5, -2)^t$. Obtenez la solution de ce problème graphiquement, et vérifiez-là avec les conditions d'optimalité.

d) Pour le problème en c), effectuez trois itérations de l'algorithme de Frank&Wolfe ; utilisez comme point de départ la valeur de α^0 correspondant à celui des quatre points le plus proche de l'origine : $P\alpha^0 = \arg\min_{i=1,2,3,4} \|P_i\|$.

***Exercice (5.6.3, page 250)** *[Problème de convergence]* Vérifiez que l'exemple de Wolfe converge bel et bien vers le point décrit. Pour ce faire, vérifiez que l'algorithme engendre la suite

$$x_k = \begin{cases} (\alpha_k, 0, \beta_k)^t & \text{si } k \text{est impair}\\ (0, \alpha_k, \beta_k)^t & \text{si } k \text{est pair}\end{cases}$$

avec $\alpha_k = a(\frac{1}{2})^k$ et $\beta_k = \frac{1}{2}\sum_{j=0}^{k-1}\sqrt{\frac{a}{2^j}}$.

Exercice (5.6.4, page 252) *[GR]* Considérez à nouveau le problème

$$\begin{aligned} &\min_{x \in \mathbb{R}^3} \frac{1}{2}\|x\|^2 \\ &\text{sujet à } x_1 + 2x_2 + 3x_3 = 1; \qquad x \geqslant 0. \end{aligned}$$

Appliquez l'algorithme du gradient réduit à partir du point de départ $x^0 = (1, 0, 0)^t$.

***Exercice (5.7.1, page 252)** *[Direction de projection]* Vérifiez que $d = Proj_E(x - \nabla f(x)^t) - x$ est une direction de descente, c'est-à-dire que $\nabla f(x) d < 0$.

Exercice (5.7.2, page 253) *[Projection]* Considérez un exemple dans $\mathbb{R}^4$. Soit $y = \frac{1}{4}(-1, -2, 2, 4)^t$. Calculez la projection décrite par le programme (5.16).

Exercice (5.7.3, page 253) *[Programmes séparables]* La solution de la projection présentée ci-haut peut se généraliser à des programmes quadratiques dont la matrice est une matrice diagonale Δ telle que $\Delta_{ij} = 0$ si $i \neq j$. Considérez le problème

$$\begin{aligned} \min_x \; & \frac{1}{2} x^t \Delta x + cx \\ \text{sujet à } & \sum_{i=1}^n a_i x_i = 1 \\ & l_i \leqslant x_i \leqslant u_i. \end{aligned}$$

On peut se limiter à des valeurs $a_i \geqslant 0$; en effet, si certains $a_i < 0$, proposez un problème équivalent avec $a_i > 0$. (Indice : on peut remplacer x_i par $-x_i$) Imitez la démarche pour proposer une technique de solution ; décrivez les conditions d'optimalité ; obtenez une expression $x_i(\lambda)$; indiquez quels sont les points de changement de morceau ; vérifiez que la fonction $s(\lambda) \stackrel{\text{def}}{=} \sum_{i=1}^n a_i x_i(\lambda)$ est monotone.

Exercice (5.7.4, page 254) *[Variante de contrainte simple]* Un certain modèle de gestion comporte un sous problème de la forme

$$\begin{aligned} \min \quad & \sum_{i=1}^3 \left(h_i \frac{x_i}{2} + S_i \frac{d_i}{x_i} \right) \\ & \sum_{i=1}^3 x_i = Q \\ & x \geqslant 0 \end{aligned}$$

où x représente le vecteur de production (des trois items), h le vecteur des coûts d'inventaire, S les coûts fixes de production, d la demande et enfin Q la production totale.

a) En définissant le Lagrangien *seulement par rapport à la contrainte d'égalité*,

$$L(x, \lambda) = \sum_{i=1}^{3} \left(h_i \frac{x_i}{2} + S_i \frac{d_i}{x_i} \right) + \lambda \left(\sum_{i=1}^{3} x_i - Q \right),$$

vérifiez que si l'on connaît λ^*, alors x^* minimise $L(x, \lambda^*)$ sujet à $x \geqslant 0$.

b) Soient $h = (1, 1, 2)^t$, $S = (100, 50, 400)^t$, $d = (20000, 40000, 40000)^t$ et $Q = 6000$. Obtenez une expression pour $x(\lambda)$.

c) Par bissection, estimez λ^*, et donc x^*.

d) À partir de $x_i = Q/3$, effectuez 3 itérations de l'algorithme de Frank&Wolfe.

Exercice (5.8.1, page 258) *[Newton pour f quadratique]*

a) Pour une fonction f quadratique et convexe (Q définie positive), démontrez que si le point $x^+ = x + d_N$ est réalisable, alors il existe des multiplicateurs de Lagrange λ_W tels que $\nabla f(x^+) + \lambda_W A_W = 0$;

b) que peut-on dire si la fonction quadratique n'est pas convexe ?

Exercice (5.8.2, page 258) *[Pas admissible pour f quadratique]*

a) obtenez une formule pour le minimum selon θ de $f(x + \theta d)$ lorsque f est une fonction quadratique convexe, i.e. lorsque la matrice Q définissant la fonction quadratique est définie positive ;

b) vérifiez que ce minimum est un pas admissible ;

c) que peut-on dire si la fonction quadratique n'est pas convexe et que d est une direction de descente réalisable ?

Exercice (5.8.3, page 258) *[Application simple]* Considérez le problème

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 + x_3^2 - 2x_1 - 4x_2 + 2x_3 \\ \text{s.à} \quad & x_1 + x_2 + x_3 \leqslant 1 \\ & x \geqslant 0 \end{aligned}$$

a) En supposant que la contrainte $ex \leqslant 1$ est satisfaite avec égalité, obtenez la solution avec la technique de la section 5.7.

b) À partir de l'origine, appliquez l'algorithme des contraintes actives pour obtenir la solution.

Exercice (5.8.4, page 259) *[Application simple]* Considérez le programme

$$\begin{aligned} \min f(x) &= \frac{x_1^2 + x_2^2}{2} + x_3^2 - 10x_1 - \frac{3x_2}{4} - \frac{x_3}{2} \\ \text{sujet à} \quad & 0 \leqslant x_1 \leqslant 1, \\ & 0 \leqslant x_2 \leqslant 1, \\ & 0 \leqslant x_3 \leqslant 1. \end{aligned}$$

À partir de l'origine,

a) en utilisant l'algorithme des contraintes actives, obtenez la solution ; puisque la fonction objectif est quadratique, pour un ensemble W donné, on peut calculer une solution exacte, et le vecteur allant de la solution courante à cette solution exacte peut être utilisé comme direction de descente dans $\ker A_w$;

b) effectuez 2 itérations de l'algorithme de Frank & Wolfe ;

c) effectuez 2 itérations de l'algorithme de projection.

Partie IV

Optimisation différentiable avec contraintes non-linéaires