TROIS ALGORITHMES DE POINTS INTÉRIEURS POUR LA PROGRAMMATION LINÉAIRE (version très préliminaire)

Patrice Marcotte

17 avril 2003

Ce court document, appendice aux notes du cours IFT6511 (Optimisation IV), constitue une brève introduction à des algorithmes polynomiaux pour la programmation linéaire. La première section est basée sur la présentation de Bazaara, Sherali et Shetty (Nonlinear Programming: Theory and Algorithms, 2nd edition, John Wiley and Sons, 1993). Les deux dernières sections sont inspirées de Fang et Puthenpara (Linear Optimization and Extensions. Theory and Algorithms, Prentice Hall, 1993).

À moins que cela ne soit strictement nécessaire, on n'utilisera pas le symbole de transposition pour les vecteurs.

1 Méthode primale-duale à petits pas

On considère le couple primal-dual de programmes linéaires:

$$\text{PRIMAL:} \quad \begin{aligned} \min_{x \in R^n} \quad & cx \\ & Ax = b \\ & x \geq 0 \end{aligned} \qquad\qquad \text{DUAL:} \quad \begin{aligned} \max_{y \in R^m} \quad & yb \\ & yA + u = c \\ & u \geq 0 \end{aligned}$$

On associe au programme primal une **barrière logarithmique** pour obtenir le problème pénalisé

$$\text{BARLOG:} \quad \begin{aligned} \min_{x > 0} \quad & cx - \mu \sum_j \ln x_j \\ & Ax = b, \end{aligned}$$

où μ est un scalaire positif.

Les conditions d'optimalité de ce programme mathématique *strictement* convexe sont

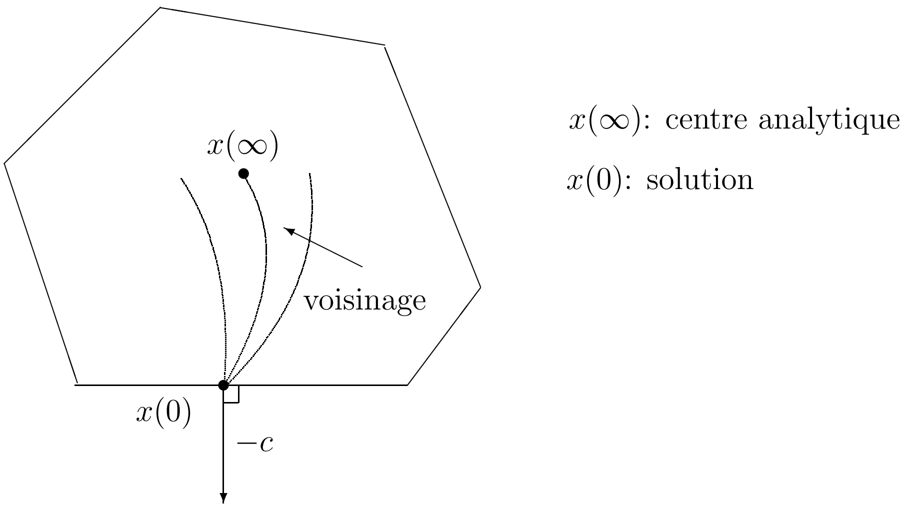

Figure 1: Géométrie associée à la méthode primale-duale

$$\text{KKT:} \qquad \begin{array}{rcll} Ax & = & b & \\ yA+u & = & c & \\ u_j x_j & = & \mu & \forall j \\ x,u & \geq & 0. & \end{array}$$

Lorsque $\mu = 0$ on retrouve les conditions d'optimalité primales-duales du programme linéaire originel. Noter que toute solution de KKT satisfait: $x, u > 0$.

Les solutions $x(\mu)$ du programme paramétré BARLOG décrivent le **chemin central**. Le point $x(\infty)$ est le **centre analytique** du polyèdre primal. Lorsque μ tend vers 0, le chemin central converge, intuitivement, vers le centre analytique de l'ensemble des solutions optimales du programme linéaire.

L'algorithme primal-dual, idéalement, souhaite parcourir le chemin central en faisant diminuer μ progressivement de ∞ à 0, par incréments finis. Ceci est malheureusement irréalisable en pratique car il est impossible d'obtenir une solution exacte de BARLOG. Nous allons donc nous contenter d'une solution primale-duale (x, y, u) approximative dont la qualité est jugée satisfaisante lorsqu'elle se trouve dans le **voisinage du chemin central**. Ce voisinage correspond aux conditions de KKT où l'on remplace les contraintes non linéaires par la contrainte agrégée plus faible

$$[\sum_j (u_j x_j - \mu)^2]^{\frac{1}{2}} \leq \theta\mu$$

avec $0 < \theta < 1/2$.

Nous sommes maintenant en mesure de décrire une itération de l'algorithme.

- Soit $\theta \in (0, 1/2)$ et $\beta \in (0, 1)$.

- Soit (x, y, u) une solution du système

$$
\begin{aligned}
Ax &= b && (1)\\
yA + u &= c && (2)\\
ux &= n\mu && (3)\\
(x, y, u) &\geq 0 && (4)
\end{aligned}
$$

  et se trouvant dans le voisinage du chemin central, c'est-à-dire:

$$[\sum_j (u_j x_j - \mu)^2]^{\frac{1}{2}} \leq \theta\mu.$$

- $\mu^+ = \beta\mu$
- Soit (x^+, y^+, u^+) le vecteur obtenu en effectuant une itération de la méthode de Newton sur le système (1)–(4).

En ignorant les contraintes de non négativité sur les variables x et u (on verra plus tard qu'elles sont automatiquement satisfaites), on a

$$(x^+, y^+, u^+) = (x + dx, y + dy, u + du),$$

où le vecteur de direction (dx, dy, du) est solution du système newtonien (approximation linéaire du système KKT au point courant)

$$
\begin{aligned}
A dx &= 0\\
dy\, A + du &= 0\\
x_j\, du_j + u_j\, dx_j &= \mu^+ - x_j u_j.
\end{aligned}
$$

En divisant la dernière égalité par $\sqrt{x_j u_j}$ on obtient:

$$\sqrt{\frac{x_j}{u_j}}\, du_j + \sqrt{\frac{u_j}{x_j}}\, dx_j = \frac{\mu^+ - x_j u_j}{\sqrt{x_j u_j}}.$$

On pose $\beta_j = \sqrt{x_j/u_j}\, du_j$ et $\alpha_j = \sqrt{u_j/x_j}\, dx_j$. On a:

$$
\begin{aligned}
u_j^+ x_j^+ - \mu^+ &= (u_j + du_j)(x_j + dx_j) - \mu^+\\
&= du_j\, dx_j + x_j\, du_j + u_j\, dx_j + u_j x_j - \mu^+\\
&= du_j\, dx_j, && (5)
\end{aligned}
$$

d'où l'on tire:

$$[\sum_j (u_j^+ x_j^+ - \mu^+)^2]^{\frac{1}{2}} = [\sum_j (du_j\, dx_j)^2]^{\frac{1}{2}}$$

$$
\begin{aligned}
&= [\sum_j (\alpha_j \beta_j)^2]^{\frac{1}{2}} \\
&\leq \sum_j \alpha_j \beta_j \\
&\leq \frac{1}{2} \sum_j (\alpha_j + \beta_j)^2 \\
&= \frac{1}{2} \sum_j \left(\frac{\mu^+ - u_j x_j}{\sqrt{u_j x_j}} \right)^2 \\
&\leq \frac{1}{2\xi} \sum_j (\mu^+ - u_j x_j)^2 \qquad \text{où } \xi = \min_j \{u_j x_j\} > 0 \\
&= \frac{1}{2\xi} \sum_j (u_j x_j - \mu + \mu - \mu^+)^2 \\
&= \frac{1}{2\xi} [\sum_j (u_j x_j - \mu)^2 + n(\mu - \mu^+)^2] \qquad \text{puisque } \sum_j (u_j x_j - \mu) = 0 \\
&\leq \frac{1}{2\xi} [\theta^2 \mu^2 + n\mu^2 (1-\eta)^2] \\
&= \frac{\mu^2 [\theta^2 + n(1-\beta)^2]}{2\xi}.
\end{aligned}
$$

Or: $|u_j x_j - \mu| \leq \theta\mu \quad \Rightarrow \quad u_j x_j \geq \mu(1-\theta) \quad \Rightarrow \quad \xi \geq \mu(1-\theta)$, d'où l'on tire:

$$
\begin{aligned}
du_j \, dx_j &= u_j^+ \, x_j^+ - \mu^+ \\
&\leq [\sum_j (u_j^+ x_j^+ - \mu^+)^2]^{\frac{1}{2}} \\
&\leq \frac{\mu^2 [\theta^2 + n(1-\beta)^2]}{2\mu(1-\theta)} \\
&= \frac{\mu [\theta^2 + n(1-\beta)^2]}{2(1-\theta)}.
\end{aligned}
$$

On choisit maintenant des paramètres β, δ et θ tels que les conditions suivantes soient réalisées:

$$
\begin{aligned}
\beta = 1 - \frac{\delta}{\sqrt{n}} & \qquad 0 < \delta < \sqrt{n} \\
\frac{\theta^2 + \delta^2}{2(1-\delta)} \leq \theta \left(1 - \frac{\delta}{\sqrt{n}} \right) & \qquad 0 \leq \theta \quad .
\end{aligned}
$$

Par exemple, on peut poser $\theta = \delta = 1/4$. On obtient alors

$$
\begin{aligned}
[\sum_j (u_j^+ x_j^+ - \mu^+)^2]^{\frac{1}{2}} &\leq \frac{\mu[\theta^2 + n(1-\beta)^2]}{2(1-\theta)} = \frac{\mu[\theta^2 + \delta^2]}{2(1-\theta)} && (6) \\
&\leq \mu\theta \left(1 - \frac{1}{\sqrt{n}} \right) = \mu\theta\beta = \theta\mu^+, && (7)
\end{aligned}
$$

ce qui démontre que l'on reste dans le voisinage du chemin central.

Pour s'assurer de la validité de l'algorithme, il reste à vérifier que la condition $ux = n\mu$ ainsi que la non négativité des variables sont bien satisfaites. On a:

$$u_j^+ x_j^+ = (u_j + du_j)(x_j + dx_j) = \mu^+ + du_j\, dx_j.$$

On en déduit:

$$\sum_j u_j^+ x_j^+ = n\mu^+ + du\, dx = n\mu^+ - dy\, Adx = n\mu^+.$$

Maintenant on déduit de (7) que

$$u_j^+ x_j^+ \geq (1-\theta)\mu^+ > 0$$

et donc que les deux variables sont de même signe. Se pourrait-il que u_j^+ et x_j^+ soient toutes les deux négatives? Pas vraiment, car alors

$$\begin{array}{rrcl} & u_j^+ & = & u_j + du_j \\ \Rightarrow & |du_j| = -du_j & = & u_j - u_j^+ \\ & & > & -u_j^+ = |-u_j^+|. \end{array}$$

Similairement: $|dx_j| > |x_j^+|$, d'où l'on tire

$$du_j\, dx_j = |du_j|\,|dx_j| > |u_j^+|\,|x_j^+| = u_j^+ x_j^+,$$

en contradiction avec $du_j\, dx_j = u_j^+ x_j^+ - \mu^+$.

Soit $\{x^k, y^k, u^k\}$ une suite d'itérés produite par l'algorithme. Il est possible, dans une première phase, d'obtenir une solution satisfaisant les conditions de l'algorithme, avec $n\mu_0 \leq 2^{2L}$, où L représente la taille du problème, exprimée en nombre de bits requis pour exprimer les paramètres du programme linéaire. On peut également démontrer, que si le saut de dualité est inférieur à 2^{-2L}, une procédure de 'purification' permet d'obtenir en temps polynomial une solution optimale du programme linéaire originel. Il nous reste donc à déterminer le nombre d'itérations requis pour obtenir un saut de dualité aussi faible. À l'itération k, on a:

$$u^k x = n\mu_k = n\beta^k \mu_0.$$

Pour que le saut de dualité soit inférieur à 2^{-2L}, il suffit de poser $\beta^k \leq 2^{-4L}$, c'est-à-dire:

$$k \geq \left(\frac{4\ln 2}{-\ln\beta}\right) L.$$

Or:

$$-\ln\beta = -\ln\left(1 - \frac{\delta}{\sqrt{n}}\right) \geq \frac{\delta}{\sqrt{n}}$$

et $k \geq ((4 \ln 2)L/\delta)\sqrt{n}$ fait l'affaire. Puisque chaque itération requiert la résolution d'un système linéaire, on en déduit la complexité $O(n^3\sqrt{n}L)$.

Remarque: Le taux de diminution β du saut de dualité est très faible. Il est possible d'utiliser un pas plus 'volontariste'. Malheureusement, il est alors possible que $x(\mu)$ se retrouve à l'extérieur du voisinage du chemin central, suite à une itération de la méthode de Newton. On peut cependant démontrer qu'un nombre *constant* de pas de Newton permet de réintégrer le voisinage. Les démonstrations sont malheureusement plus élaborées.

Initialisation

Considérons le couple primal-dual de programmes linéaires perturbés (e désigne le vecteur dont tous les coefficients sont égaux à 1)

PRIMAL

$$\begin{array}{ll} \min\limits_{x \in R^n} & cx + M_1 x_0 \\ & Ax + (b - \lambda A e)x_0 = b \\ & (c - \gamma e)x - \gamma x_{n+1} = -M_2 \\ & \\ & (x, x_0, x_{n+1}) \geq 0 \end{array}$$

DUAL

$$\begin{array}{ll} \max\limits_{y \in R^m} & yb - M_2 y_0 \\ & yA + (c - \gamma e)y_0 + u = c \\ & (b - \lambda A e)y + u_0 = M_1 \\ & -\gamma u_0 + u_{n+1} = 0 \\ & (u, u_0, u_{n+1}) \geq 0 \end{array}$$

où λ et γ sont deux 'grands' nombres. On pose:

$$\begin{array}{rcl} M_1 & = & \lambda\gamma \\ M_2 & = & (n+1)\lambda\gamma - \lambda c e. \end{array}$$

On pose $(x, x_0, x_{n+1}) = (\lambda e, 1, \lambda)$ et $(y, y_0, u, u_0, u_{n+1}) = (0, 1, \gamma, \lambda\gamma, \gamma)$. On vérifie aisément que que ces deux vecteurs sont réalisables pour le primal et le dual, respectivement. De plus, le couple se trouve sur le chemin central avec $\mu = \lambda\gamma$. Si M_1 et M_2 sont assez grands, on aura $x_0 = 0$ et $y_0 = 0$ à l'optimum. Cette technique est réminiscente de la méthode dite 'du grand M' en programmation linéaire.

2 Algorithme de Karmarkar

Considérons un couple primal-dual de programmes linéaires d'une structure un peu particulière:

PRIMAL:
$$\begin{array}{ll} \min\limits_{x \in R^n} & cx \\ & Ax = 0 \\ & x \in S \end{array}$$

DUAL:
$$\begin{array}{ll} \max\limits_{z \in R, y \in R^m} & z \\ & yA + ze \leq c \end{array}$$

où $S = \{x \in R^n_+ : \sum_j x_j = 1\}$ représente le **simplexe unitaire**. On fait les trois hypothèses suivantes:

- la matrice A est de plein rang (cette hypothèse n'est pas restrictive);
- $Ae = 0$, où $e = (1, 1, \ldots, 1)$;
- la valeur optimale du programme linéaire est 0.

Nous verrons à la fin de la section comment relaxer les deux dernières hypothèses, qui paraissent fortes. Remarquer que les contraintes du primal forment un ensemble géométriquement simple: l'intersection d'un hyperplan (espace linéaire) et du simplexe unitaire (hypertriangle équilatéral).

En se basant sur un vecteur de composantes strictement positives, on définit une **transformation projective** T_d:

$$x'_j = (T_d(x))_j = (x_j/d_j)/\sum_k (x_k/d_k).$$

Si l'on définit D comme la matrice diagonale d'éléments d_j, on a:

$$T_d(x) = \frac{D^{-1}x}{eD^{-1}x} \qquad T_d^{-1}(x') = \frac{Dx'}{eDx'}.$$

Cette transformation possède quelques propriétés intéressantes:

- $T_d(d) = e/n$ (l'image de d est le centre du simplexe);
- $T_d(e^j) = e^j$ (les sommets de S sont des points fixes de T_d);
- $T_d(S) = S$;
- T_d préserve les segments.

Soit x le point courant. En appliquant la transformation T_x, on transforme le programme primal originel en un programme non linéaire:

$$\begin{aligned} \min_{x'} \quad & (cXx')/(eXx') \\ & AXx' = 0 \\ & x' \in S. \end{aligned}$$

Puisque la valeur optimale de ce programme est 0, celui-ci est équivalent au programme *linéaire*

$$\begin{aligned} \min_{x'} \quad & c'x' \\ & A'x' = 0 \\ & x' \in S, \end{aligned}$$

où l'on a posé $c' = cX$ et $A' = AX$. Dans le simplexe transformé, le centre correspond au point x dans le simplexe de départ. Soit $B(e/n, r)$ la boule *inscrite* dans S et $B(e/n, R)$ la boule *circonscrite* à S. Il n'est pas difficile de vérifier que $r = 1/\sqrt{n(n-1)}$ et $R = \sqrt{(n-1)/n}$.

Maintenant on remplace, dans le programme linéaire précédent, la contrainte $x' \in S$ par la contrainte $x' \in B(e/n, r)$. La solution de ce programme est obtenue en effectuant un pas de -1 dans la direction opposée au gradient projeté (et normalisé) sur la variété linéaire $\{A'x' = 0\ \sum_j x'_j = 0\}$, c'est-à-dire:

$$
\begin{aligned}
&Q = \begin{pmatrix} A' \\ e^t \end{pmatrix} \qquad d' = -(I - Q^t(QQ^t)^{-1}Q)c' \\
&x' \leftarrow e/n + d'/\|d'\|.
\end{aligned}
$$

Le problème est que la solution se trouve maintenant sur la frontière du simplexe et non pas dans l'intérieur (relatif) du domaine admissible. Afin de pallier à ce problème, on effectue un pas α/n inférieur à 1. Nous sommes maintenant en mesure d'énoncer l'algorithme, qui est, vous l'avouerez, d'une simplicité désarmante!

- Soit $\alpha \in (0, 1)$ et x un point dans l'intérieur relatif du domaine admissible.
- $x' \leftarrow e/n + (\alpha/n)d'/\|d'\|$.
- $x^+ = (Xx')/(eXx')$.

La preuve de convergence de la méthode est basée sur l'analyse de la **fonction potentielle**

$$
\begin{aligned}
f_c(x) &= n \ln cx - \sum_j \ln x_j \\
&= \sum_j \ln \frac{cx}{x_j},
\end{aligned}
$$

qui n'est autre que l'objectif linéaire pénalisé par une barrière logarithmique, à une constante multiplicative près. Une propriété importante de cette fonction est qu'elle est invariante par rapport à la transformation projective T_d, à un facteur additif près.

Lemme 1 *Soit $x' = T_d(x)$. On a:*

$$
f_{c'}(x') = f_c(x) + \sum_j \ln d_j.
$$

Preuve: Posons $c'_j = d_j c_j$ et $x'_j = (x_j/d_j)/\sigma$, avec $\sigma = \sum_k (x_k/d_k)$. On écrit:

$$
\begin{aligned}
f_{c'}(x') &= n \ln \frac{\sum_j c_j x_j}{\sigma} - \sum_j \ln \frac{x_j/d_j}{\sigma} \\
&= f_c(x) + \sum_j \ln d_j \\
&= f_c(x) + \ln \det(D). \qquad \square
\end{aligned}
$$

Lemme 2 *Soit $x' = (e/n) + (\alpha/n)(d'/\|d'\|)$. On a:*

$$n \ln c'x' \le n \ln c' \frac{e}{n} - \alpha.$$

Preuve: Par construction de la direction projetée d', on a: $c'd' = -\|d'\|^2$. On en déduit

$$c'x' = c'\frac{e}{n} - \frac{\alpha}{n}\|d'\|.$$

Soit $x'(R) = c'e/n + Rd'/\|d'\|$ la solution du programme linéaire où l'on a substitué à S la boule circonscrite $B(e/n, R)$. La solution de ce problème relaxé est évidemment meilleure que la solution du problème originel, c'est-à-dire: $c'x'(R) \le 0$. On a donc:

$$\begin{array}{lrcll}
 & c'x'(R) & = & c'\frac{e}{n} - R\|d'\| \le 0 & \\
\Rightarrow & \|d'\| & \ge & \frac{1}{R}c'\frac{e}{n} & \\
\Rightarrow & c'x' & = & c'\frac{e}{n} - \frac{\alpha}{n}\|d'\| & \\
 & & \le & \left(1 - \frac{\alpha}{nR}\right) c'\frac{e}{n} & \\
 & & < & \left(1 - \frac{\alpha}{n}\right) c'\frac{e}{n} & \text{car } R < 1 \\
\Rightarrow & \ln c'x' & \le & \ln\left(1 - \frac{\alpha}{n}\right) + \ln\frac{c'e}{n} & \\
 & & \le & -\frac{\alpha}{n} + \ln\frac{c'e}{n} \quad . & \square
\end{array}$$

Lemme 3

$$x \in B\left(\frac{e}{n}, \frac{\alpha}{n}\right) \quad \Rightarrow \quad -\sum_j \ln x_j \le n \ln n + \frac{\alpha^2}{2(1-\alpha)^2}.$$

Preuve: L'expansion de Taylor de la fonction $\ln(1 + (nx_j - 1))$ autour du point 1 donne assure l'existence d'un nombre $\xi_j \in [1, nx_j]$ tel que

$$\ln(nx_j) = \ln 1 + (nx_j - 1) - \frac{1}{2\xi_j^2}(nx_j - 1)^2.$$

(Noter que qu'il n'est pas exclu que nx_j soint inférieurs à 1.) D'où:

$$\begin{array}{rcl}
x \in B\left(\frac{e}{n}, \frac{\alpha}{n}\right) & \Rightarrow & x_j \ge \frac{1-\alpha}{n} \\
 & \Rightarrow & \xi_j \ge \min\{1, nx_j\} \ge 1 - \alpha \\
 & \Rightarrow & \ln(nx_j) \ge nx_j - 1 - \frac{1}{2(1-\alpha)^2}(nx_j - 1)^2
\end{array}$$

$$
\begin{aligned}
&\Rightarrow \sum_j \ln(nx_j) \geq 0 - \frac{n^2}{2(1-\alpha)^2}\|x - \frac{e}{n}\|^2 \\
&\Rightarrow \sum_j \ln x_j \geq -n\ln n - \frac{n^2}{2(1-\alpha)^2}\frac{\alpha^2}{n^2}. \qquad \square
\end{aligned}
$$

Théorème 1 *Pour une valeur de c arbitraire, on a:* $f_c(x^k) \leq f_c(x^0) - k/5$.

Preuve: D'après les lemmes précédents, on a:

$$
\begin{aligned}
f_{c'}(x') &\leq f_{c'}\left(\frac{e}{n}\right) - \alpha \quad \text{(lemme 2)} \qquad + \frac{\alpha^2}{2(1-\alpha)^2} \quad \text{(lemme 3)} \\
&\leq f_{c'}\left(\frac{e}{n}\right) - \frac{1}{5} \qquad \text{si } \alpha = \frac{1}{3}.
\end{aligned}
$$

Comme les fonctions f_c ne diffèrent que d'une constante additive, le résultat s'ensuit. $\square$

Corollaire 1

$$cx^k \leq \exp(-k/5n)cx^0.$$

Preuve: Par le théorème précédent:

$$
\begin{aligned}
n\ln cx^k - \sum_j \ln x_j^k &\leq n\ln cx^0 - \sum_j \ln x_j^0 - \frac{k}{5} \\
\Rightarrow \qquad n\ln cx^k &\leq n\ln cx^0 - \frac{k}{5} && (8) \\
& && (9)
\end{aligned}
$$

puisque, sur le simplexe unitaire:

$$\sum_j \ln x_j^k \leq \sum_j \ln\frac{1}{n} = \sum_j \ln x_j^0.$$

En divisant (8) par n et en prenant l'exponentielle, on obtient le résultat. $\square$

On déduit du corollaire que le nombre d'itérations requis est $O(nL)$ et que la complexité de l'algorithme de Karmarkar est $O(n^4L^2)$.

Remarque: En profitant astucieusement de la similarité entre deux projections (la matrice Q ne change pas 'beaucoup'), il est possible de réduire la complexité par un facteur de $\sqrt{n}$.

Initialisation

Tout problème de programmation linéaire peut se ramener à un problème de réalisabilité:

$$Mz \le u \qquad z \ge 0.$$

Considérons le programme linéaire en 'forme de Karmarkar':

$$\begin{array}{ll} \min\limits_{z,s,t,\lambda} & \lambda \\ & Mz + s - tu + (u - Me - e)\lambda = 0 \\ & ez + es + t + \lambda = 1 \\ & z, s, t, \lambda \ge 0. \end{array}$$

Les deux problèmes sont équivalents. En effet, s'il existe une solution z au problème de réalisabilité, alors le vecteur $(z, u - Mz, 1, 0)$, à un facteur multiplicatif près, est solution optimale du programme linéaire. Réciproquement, si $(z, s, t, 0)$ est solution optimale du programme linéaire, on a:

$$Mz = tu - s.$$

Si $t \ne 0$, z/t est solution du problème de réalisabilité. Or $t \ne 0$, sinon $Mz = -s$ impliquerait que le système primal-dual est non borné, ce qui est impossible.

3 Méthode primale affine

Pourquoi se compliquer la vie avec une transformation projective alors qu'on peut très simplement éloigner un point de la frontière du domaine admissible par une transformation affine (linéaire)? Considérons le programme linéaire

$$\begin{array}{ll} \min & cx \\ & Ax = b \\ & x \ge 0. \end{array}$$

On fait l'hypothèse que la matrice A est de plein rang et que l'intérieur relatif $\{Ax = b, x > 0\}$ n'est pas vide et on définit la transformation linéaire

$$x' = T(x) = X^{-1}x,$$

où X, comme auparavant, est une matrice diagonale dont les éléments correspondent à ceux du vecteur x. On a $T(x) = e$, et la transformation inverse prend la forme

$$x = T^{-1}(x') = Xx'.$$

En effectuant le changement de variable $y = T(x)$, le programme linéaire originel devient

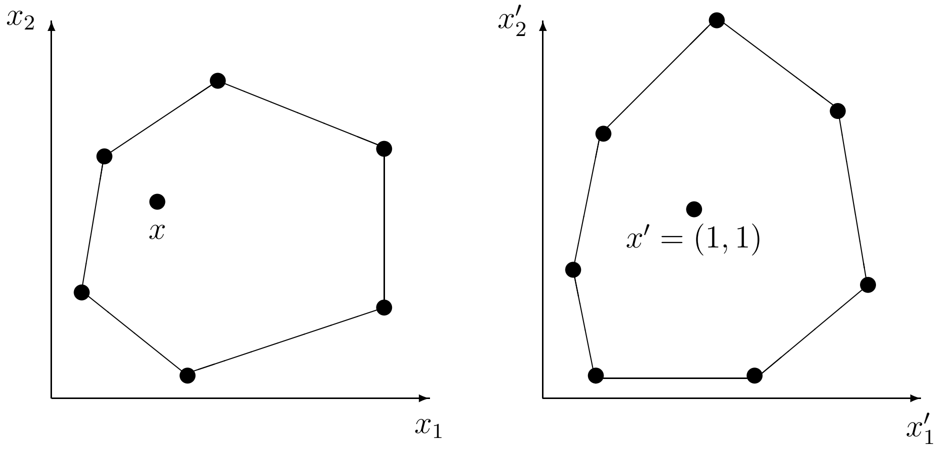

Figure 2: Transformation affine. Les contraintes de non négativité s'appliquent sur les variables d'écart.

$$\begin{array}{ll} \min & cXx' \\ & AXx' = b \\ & x' \geq 0. \end{array}$$

En posant $c' = Xc$ et $A' = AX$, on réécrit celui-ci sous la forme

$$\begin{array}{ll} \min & c'x' \\ & A'x' = b \\ & x' \geq 0. \end{array}$$

On calcule maintenant la direction opposée du gradient projeté sur les contraintes actives, dans le domaine transformé.

$$\begin{array}{rcll} d' & = & \mathrm{proj}_{\{y:A'y=0\}} - c' = P'(-c') & \\ [P' & = & I - A'(AA'^t)^{-1}A' & \text{est la matrice de projection}] \\ x'^+ & = & x' + \alpha' d'. & \end{array}$$

Comme dans l'algorithme du simplexe, soit

$$\begin{array}{rcll} t_{\max} & = & \min\{x'_j/(-d'_j) : d'_j < 0\} & \\ & = & \min\{1/(-d'_j) : d'_j < 0\} & \text{puisque } x' = (1,1,\ldots,1) \\ & \geq & 1/\|d'\| & \end{array} \tag{10}$$

le pas maximal que l'on puisse effectuer dans la direction d'. On choisit comme pas $\alpha' = \alpha t_{\max}$, avec $\alpha \in (0,1)$. On peut également écrire, puisque $x = Xx'$:

$$d' = -[I - A'^t (A'A'^t)^{-1} A']Xc.$$

Dans l'espace originel, la direction est

$$d = Xd' = -X[I - XA^t(AX^2A^t)^{-1}AX]Xc.$$

Donnons maintenant le pseudo-code d'une itération de l'algorithme.

- estimé du vecteur dual: $y = (AX^2A^t)^{-1}AX^2c$
- estimé des coûts réduits: $r = c - yA$
- direction d'amélioration: $d' = -Xr$, $d = Xd'$.
- nouvel itéré: $x^+ = x + \alpha' d$.

Remarque: Si x est un point extrémal associé à une base B, il est facile de vérifier que $y = (c_B B^{-1}, 0)$ correspond à un vecteur dual (pas nécessairement réalisable). De plus, $r = c - yA$ correspond au vecteur des coûts réduits.

Lors d'une itération de l'algorithme, l'objectif devient

$$\begin{aligned} cx^+ &= cx + \alpha' cXd' \\ &= cx + \alpha' c'd' \\ &= cx - \alpha' \|d'\|^2 \\ &< cx \qquad\qquad \text{si } x \notin X^*. \end{aligned} \tag{11}$$

Dans l'étude de convergence on fera l'hypothèse que le domaine admissible est borné (compact, donc), que sont intérieur est non vide et que les problèmes primal et dual sont non dégénérés. Ces hypothèses impliquent que la matrice AX est de plein rang et que le vecteur r possède au plus m composantes nulles (A est une matrice $m \times n$).

Lemme 4 $\lim_{k\to\infty} X_k r^k = 0$.

Preuve: D'après (11) et (10) on obtient

$$cx - cx' = \alpha' \|d'\|^2 \geq \frac{\alpha}{\|d'\|}\|d'\|^2 = \alpha \|d'\|. \tag{12}$$

Puisque la suite $\{cx^k\}$ engendrée par l'algorithme est décroissante et bornée inférieurement, elle converge. Donc, en utilisant (12):

$$0 = \lim_k (cx^k - cx^{k+1}) \geq \alpha \|d'^k\|,$$

ce qui implique:

$$\lim_k \|d'^k\| = \lim_k \|X_k r^k\| = 0. \qquad \square$$

On définit maintenant les deux ensembles:

$$\begin{aligned} C &= \{x : Ax = b, x \geq 0, Xr = 0\} \\ D &= \{x : Ax = b, x \geq 0, r \geq 0\}, \end{aligned}$$

Il est facile de voir que $C \cap D \subseteq X^*$, car alors le vecteur y est dual-réalisable et la complémentarité est satisfaite.

Théorème 2 *Si la suite* $\{x^k\}$ *converge vers* $\overline{x}$*, alors* $\overline{x} \in X^*$*.*

Preuve: Supposons, par contradiction, que $\overline{x} \notin X^*$, et soit $\overline{r} = \lim_k r^k$. Par le lemme 4 et un argument de continuité on a $\overline{X}\overline{r} = 0$, c'est-à-dire: $\overline{x} \in C$. Par contre, $\overline{x} \notin D$, et il existe un indice j tel que $\overline{r}_j < 0$. On en déduit $\overline{x}_j = 0$.

Maintenant, soit K un indice tel que $r_j^k < 0$ pour tout $k \geq K$. On a:

$$\begin{aligned} x_j^{k+1} &= x_j^k + \alpha'_k x_j^k(-x_j^k r_j^k) \\ &= x_j^k - \alpha'_k (x_j^k)^2 r_j^k \\ &> x_j^k \\ &> 0 \qquad\qquad \forall k \in K, \end{aligned}$$

en contradiction avec $\overline{x}_j = 0$. $\square$

Théorème 3 *La suite* $\{x^k\}$ *converge.*

Preuve: Soit x^* un point d'accumulation de la suite. Le lemme 4 nous assure que $x^* \in C$. L'hypothèse de non dégénérescence nous permet d'affirmer que l'ensemble C ne contient que des sommets du domaine admissible (exercice). Soit $x^* = (x_B^*, x_N^*)$ la décomposition de x^* en variables 'de base' et variables 'hors base' et

$$\epsilon \in (0, \min_{j \in N} |r_j^*|).$$

On choisit un nombre positif δ tel que (on choisit la norme du maximum pour les boules)

- $B(x^*, 2\delta) \cap C = \{x^*\}$;
- $x^k \in B(x^*, \delta) \Rightarrow \min_{j \in N} |r_j^k| > \epsilon$ (non dégénéré)

et on définit l'ensemble

$$S_{\epsilon,\delta} = \{x \in B(x^*, 2\delta) : \alpha'_k (x_j^k r_j^k)^2 < \epsilon\delta\}.$$

Notons que, puisque le domaine est borné, le pas est aussi borné. On a:

$$\begin{array}{rcll} x^k \in S_{\epsilon,\delta} & \Rightarrow & \alpha'_k {(x_j^k)}^2 |r_j^k| < \delta & \\ & \Rightarrow & |x_j^{k+1} - x_j^k| = \alpha'_k {(x_j^k)}^2 r_j^k < \delta & \\ & \Rightarrow & |x_j^{k+1} - x^*| \leq 2\delta & \forall j \in N \\ & \Rightarrow & x^{k+1} \in B(x^*, 2\delta). & \end{array}$$

Si la suite $\{x^k\}$ ne converge pas vers x^*, elle doit posséder au moins un autre point d'accumulation, qui doit aussi ête un sommet de P. La suite doit donc pénétrer et quitter $S_{\epsilon,\delta}$ une infinité de fois. Comme, en sortant de $S_{\epsilon,\delta}$, elle pénètre dans $B(x^*, 2\delta)$, il doit donc exister un point d'accumulation $\tilde{x}$ qui se trouve dans $B(x^*, 2\delta)$ mais pas dans $S_{\epsilon,\delta}$. Or $\tilde{x} \in C$ implique que $\tilde{x}$ est un point extrémal du domaine, en contradiction avec la définition de l'ensemble $B(x^*, 2\delta)$. $\square$