Introduction aux méthodes de points intérieurs

Extrait de notes de cours

par
Gilles Savard
GERAD et Département de mathématiques et génie industriel
École Polytechnique de Montréal

Février 2001

1 Préface

Ce document présente une introduction aux méthodes de points intérieurs. Il correspond essentiellement à la deuxième partie du cours MTH 6413 (Programmation mathématique II) de l'École Polytechnique de Montréal, la première partie étant consacrée aux méthodes algorithmiques classiques de la programmation mathématique non linéaire, la troisième aux méthodes de décomposition et la quatrième à l'optimisation globale.

Plusieurs raisons nous ont amené à inclure les méthodes de points intérieurs dans le cursus des étudiants inscrits au programme de recherche opérationnelle du département de mathématique et génie industriel. D'abord l'impact considérable qu'ont eu ces méthodes sur la communauté scientifique de la programmation mathématique, dû à leurs bonnes propriétés théoriques et algorithmiques, se devait d'être compris. Puis le potentiel de certains concepts de ces méthodes pour deux domaines actifs de recherche au département, les méthodes de décomposition de problème de grande taille et l'optimisation globale, était à considérer. Enfin, le rôle que joue la théorie de la complexité dans les méthodes de points intérieurs nous permet de revoir certains concepts de cette théorie dans un cadre continu plutôt que combinatoire.

Ce document contient seulement un extrait des notes de cours. En particulier, les figures qui aident à donner une interprétation géométrique aux concepts algébriques seront données en classe. La section 2 présente quelques notations et rappelle les principales propriétés de la programmation linéaire et non linéaire. Après un bref survol de l'historique des méthodes de points intérieurs, la section 3 expose quelques notions de complexité. La section 4 présente la méthode des ellipses (une méthode de points extérieurs!) appliquées à la programmation linéaire et la section 5 expose la méthode affine primale, dont la simplicité permet une introduction en douceur aux principales idées sous-jacentes aux méthodes de points intérieurs. La méthode duale de chemin central fait l'objet de la section 6 alors que la méthode primale-duale est discutée à la section 7. Quelques détails sur les implantations pratiques sont données à la section 8. La méthode projective de Karmarkar est présentée à la section 9. Enfin, nous terminons le document par une courte section sur les extensions aux cas non linéaires.

2 Éléments et notions de base

2.1 Notations

Un programme mathématique (PM) est un problème d'optimisation de la forme:

$$\begin{aligned} \min\ & f(x) & \\ g_i(x) &\leq 0, & i = 1,\ldots,k \\ h_j(x) &= 0, & j = 1,\ldots,m \\ x &\in C \subseteq R^n & \end{aligned}$$

où $f,g,h : \Re^n \to \Re$. Parmi les cas particuliers les plus étudiés, on note:

- la programmation linéaire ($f(x)$ linéaire, $g(x)$,$h(x)$ affines, C octant positif)
- la programmation convexe ($f(x)$,$g(x)$ convexes, $h(x)$ affines, C convexe)
- la programmation en nombres entiers (C discret)

Définition 2.1.1 *L'ensemble* $S = \{x \in C \ : \ g_i(x) \leq 0, h_j(x) = 0\}$ *est appelé* **ensemble des solutions réalisables**.

Définition 2.1.2 *Soit* $x^* \in S$. *On dit que* x^* *est un* **minimum local** *s'il existe* $\epsilon > 0$ *tel que*

$$f(x^*) \leq f(x), \qquad \forall x \in B_\epsilon(x^*)$$

où $B_\epsilon(x^*) = \{x \in \Re^n \ : \ \|x - x^*\| < \epsilon\}$.

Définition 2.1.3 *Soit* $x^* \in S$. *On dit que* $y \in R^n$ *est une* **direction admissible** *s'il existe* $\alpha > 0$ *tel que*

$$x^* + ty \in S, \qquad \forall t \in [0,\alpha].$$

2.2 Conditions d'optimalité (sans contrainte)

Soit le programme non linéaire et non contraint (PNC)

$$\min_{x \in \Re^n} \ f(x) \tag{1}$$

où la fonction $f(x)$ est au moins deux fois continûment différentiables.

Soit x^* un minimum local de $f(x)$, on a alors nécessairement pour tout $t > 0$ assez petit

$$f(x^*) \leq f(x^* + ty), \qquad \forall y \text{ admissible.}$$

Ceci implique:

$$\lim_{t \downarrow 0} \frac{f(x^* + ty) - f(x^*)}{t} = \nabla f(x^*) y \geq 0.$$

En particulier, la direction $y = -\nabla f(x^*)^T$ est admissible et on déduit:

$$\nabla f(x^*) \nabla f(x^*)^T \leq 0 \quad \Leftrightarrow \quad \nabla f(x^*) = 0.$$

Théorème 2.2.1 *(Condition nécessaire du 1er ordre)*
Soit x^* un minimum local pour (1). On a alors

$$\nabla f(x^*) = 0.$$

Un point x satisfaisant cette condition est appelé un point stationnaire.
Au deuxième ordre, on obtient:

$$\begin{aligned} f(x^*) &\leq f(x^* + ty) \\ &= f(x^*) + t\nabla f(x^*) y + \frac{t^2}{2} y^T \nabla^2 f(x^*) y + o(t^2) \end{aligned}$$

avec $o(t^2)$ dénotant une fonction qui est petit ordre de t^2, c'est-à-dire:

$$\lim_{t \downarrow 0} \frac{o(t^2)}{t^2} = 0.$$

Comme x^* est un minimum local, on a:

$$0 \leq \frac{t^2}{2} y^T \nabla^2 f(x^*) y + o(t^2).$$

En divisant par t^2, en prenant la limite et par continuité du hessien, on obtient:

Théorème 2.2.2 *(Conditions nécessaires du 2ième ordre)*
Soit x^* un minimum local pour (1). On a alors

$$\begin{aligned} \nabla f(x^*) &= 0 \\ y^T \nabla^2 f(x^*) y &\geq 0, \quad \forall y \in R^n. \end{aligned}$$

La matrice hessienne au point x^* est donc semi-définie positive. Ces conditions ne sont pas suffisantes comme le montre, par exemple, la fonction x^3.

Théorème 2.2.3 *(Conditions suffisantes du 2ième ordre)*
Soit x^* un point tel que:

$$\begin{aligned} \nabla f(x^*) &= 0 \\ y^T \nabla^2 f(x^*) y &> 0, \quad \forall y \in R^n, \end{aligned}$$

alors x^* est un minimum local (strict).

Remarquons que cette fois, ces conditions ne sont pas nécessaires (considérer la fonction $f(x) = x^4$).

2.3 Conditions d'optimalité (avec contraintes)

Soit le programme mathématique non linéaire (PM):

$$\begin{aligned} \min_{x \in \Re^n} \quad & f(x) \\ \text{s.c.} \quad g_i(x) &\leq 0, \quad i = 1,\ldots,k \\ h_j(x) &= 0, \quad j = 1,\ldots,m \end{aligned} \tag{2}$$

où les fonctions f,g_i et h_j sont au moins deux fois continûment différentiables.

Définition 2.3.1 *Le* **lagrangien** *du programme mathématique (2) est défini par*

$$L(x; \lambda,\mu) = f(x) + \sum_{j=1}^{m} \lambda_j h_j(x) + \sum_{i=1}^{k} \mu_i g_i(x).$$

Les conditions d'optimalité requièrent des hypothèses sur la *régularité* de la solution. Essentiellement, un point x^* est régulier dans les cas où la géométrie de l'ensemble réalisable au voisinage de x^* est adéquatement représentée par la linéarisation des contraintes. Ils existent une panoplie de conditions de régularité dont l'indépendance linéaire des gradients des contraintes actives, les conditions de Mangasarian-Fromovitz et les conditions de Slater (dans le cas convexe). Les démonstrations des conditions d'optimalité qui suivent, ainsi que diverses notions de régularité, se trouvent dans le livre de Bazaara, Sherali et Shetty (1993).

Théorème 2.3.1 *(Karush-Kuhn-Tucker)*
Soit x^* un minimum local régulier de (2). Alors, il existe les multiplicateurs $\lambda^* \in \Re^m$ et $\mu^* \in \Re^k$ tels que:

$$\begin{aligned} \nabla f(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla h_j(x^*) + \sum_{i=1}^{k} \mu_i^* \nabla g_i(x^*) &= 0 \\ (\nabla_x L(x^*; \lambda^*,\mu^*) = 0) \\ \mu_i^* &\geq 0, \quad i = 1,\ldots,k \qquad (KKT) \\ u_i^* g_i(x^*) &= 0, \quad i = 1,\ldots,k \\ g_i(x^*) &\leq 0, \quad i = 1,\ldots,k \\ h_j(x^*) &= 0, \quad j = 1,\ldots,m. \end{aligned}$$

Les conditions nécessaires du premier ordre (KKT) sont souvent appelées les conditions de Karush-Kuhn-Tucker (ou Kuhn-Tucker). Dans le cas convexe (c'est-à-dire lorsque $f(x)$ et $g_i(x)$ sont des fonctions convexes et $h_j(x)$ des fonctions affines), un point x^* régulier est un minimum global pour le programme (2) si et seulement si il satisfait les conditions de Karush-Kuhn-Tucker.
Les conditions du deuxième ordre utilisent les notations suivantes:

$$I(x) = \{i = 1,\ldots,k \ : \ g_i(x) = 0\},$$

$$
\begin{aligned}
I'(x) &= \{i=1,\dots,k \ : \ g_i(x)=0, \mu_i>0\},\\
T(x) &= \{y\in\Re^n \ : \ \nabla g_i(x)y\le 0, i\in I(x)\\
&\qquad \text{et } \nabla h_j(x)y=0, j=1,\dots,m\},\\
T'(x) &= \{y\in\Re^n \ : \ \nabla g_i(x)y\le 0, i\in I'(x)\\
&\qquad \text{et } \nabla h_j(x)y=0, j=1,\dots,m\}.
\end{aligned}
$$

L'ensemble $I(x)$ correspond à l'ensemble des indices des contraintes d'inégalités actives au point x. L'ensemble $I'(x)$ ne contient que les indices des contraintes d'inégalités actives qui vérifient la complémentarité stricte. $T(x)$ est le cône des directions admissibles au point x qui est inclus dans le cône élargi $T'(x)$. Ces ensembles permettent la caractérisation de la matrice hessienne réduite du lagrangien.

Théorème 2.3.2 *(Conditions nécessaires du 2ième ordre) Soit x^* un minimum local régulier de (2). Alors, il existe des multiplicateurs $\lambda^*\in\Re^m$ et $\mu^*\in\Re^k$ tels que:*

- *les conditions de KKT sont satisfaites;*
- $y^T\nabla^2 L(x^*;\lambda^*,\mu^*)y\ge 0, \quad \forall y\in T(x^*)$.

Théorème 2.3.3 *(Conditions suffisantes du 2ième ordre) Le point x^* est un minimum local strict de (2) si il existe les multiplicateurs $\lambda^*\in\Re^m$ et $\mu^*\in\Re^k$ tels que:*

- *les conditions de KKT sont satisfaites;*
- $y^T\nabla^2 L(x^*;\lambda^*,\mu^*)y> 0, \quad \forall y\in T'(x^*)$.

2.4 Programmation linéaire

On considère le programme linéaire primal (P):

$$
\begin{aligned}
\min_x \ & c^T x\\
\text{s.c.} \ \ Ax &= b\\
x &\ge 0.
\end{aligned}
$$

Ainsi formulé, on dira qu'un programme linéaire est sous forme standard. Notons que tout programme linéaire, c'est-à-dire tout problème d'optimisation où la fonction objectif et les fonctions définissant les contraintes sont affines, peut se mettre sous la forme standard par des manipulations algébriques simples. La matrice A est de dimension $m\times n$ avec $m\le n$. On pose l'hypothèse que la matrice A est de plein rang. Sur des modélisations pratiques, cette hypothèse est rarement satisfaite. La matrice A peut toutefois être réduite à une matrice de plein rang par une décomposition QR ou une élimination de Gauss. L'ensemble des solutions réalisables

$$S=\{x\in\Re^n \ : \ Ax=b,\ x\ge 0\}$$

forme un polyèdre convexe, c'est-à-dire l'intersection d'un nombre fini de demi-espaces. Un programme linéaire est réalisable si l'ensemble S est non vide. Un programme réalisable est borné si l'objectif est borné sur S. On dit que x est un point extrême de S, s'il ne peut pas s'écrire comme une combinaison convexe stricte de deux points de S.

On présente ici, sans preuve, les principaux résultats de la programmation linéaire. Le lecteur les retrouvera dans l'excellente monographie de Chvátal (1983).

Remarque: Un programme linéaire sans contrainte d'inégalité n'est d'aucun intérêt.

Proposition 2.4.1 *Un point x est un point extrême de S si et seulement si les colonnes $\{A^j \ : \ x_j>0\}$ de A sont linéairement indépendantes.*

On en déduit que si S est non vide, alors il existe au moins un point extrême.

Proposition 2.4.2 *Un programme linéaire réalisable et borné possède une solution optimale à un point extrême du polyèdre de ses solutions réalisables.*

Définition 2.4.1 *Une* **base** *de A est toute sous-matrice B formée de m colonnes linéairement indépendantes de A.*

Définition 2.4.2 *Soit B une base de A. La* **solution de base** *associée à B est le point $x\in R^n$ tel que*

$$
\begin{aligned}
x_B &= B^{-1}b\\
x_N &= 0.
\end{aligned}
$$

Définition 2.4.3 *Une solution de base telle que $x_B\ge 0$ est dite* **solution de base réalisable**.

À une solution de base réalisable du système:

$$Ax=b$$

correspond un point extrême du polyèdre convexe. À tout point extrême, correspond au moins une base réalisable.

2.4.1 La méthode du simplexe (d'un sommet l'autre)

La méthode du simplexe itère d'une base réalisable à une autre base réalisable adjacente (seuls deux indices de la base sont modifiés) de valeur inférieure ou égale. Le nombre de base réalisable est clairement fini. Dans le cas non dégénéré ($x_B>0$), ces deux bases correspondent à deux points extrêmes voisins de valeurs différentes. Dans le cas dégénéré (au moins une composante du vecteur x_B est nulle), les deux bases peuvent corresponde au même point extrême, et donc de même valeur. Comme la suite des valeurs des bases visitées n'est pas nécessairement strictement monotone croissante dans le cas dégénéré, une règle d'anti-cyclage est requise afin de garantir la convergence (règle lexicographique, règle de Bland (1977)).

1. Soit une solution de base x réalisable:

$$
\begin{aligned}
Ax &= b\\
\Leftrightarrow \quad Bx_B+Nx_n &= b\\
\Leftrightarrow \quad x_B &= B^{-1}b-B^{-1}Nx_N
\end{aligned}
$$

et de valeur

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

De façon compacte, on associe à chaque base un *dictionnaire* (ou encore tableau):

$$\begin{array}{rcll} x_B & = & B^{-1}b & -B^{-1}Nx_N \\ \hline z & = & c_B^T B^{-1}b & + (c_N^T - c_B^T B^{-1}N)x_N \end{array}$$

On note

$$\bar{c}_N = c_N^T - c_B^T B^{-1}N$$

le vecteur des coûts réduits des variables hors-base et on pose

$$y = c_B^T B^{-1}.$$

(On devrait plutôt dire y solution du système: $By = c_B^T$.)

2. Changement de base profitable?
   (a) Choix de la variable d'entrée:
   choisir $i \in N$ tel que $\bar{c}_i < 0$;
   s'il n'existe pas un tel i, la solution est optimale.
   (b) Choix de la variable de sortie:
   on cherche la première composante j du vecteur

   $$x_B = B^{-1}b - B^{-1}A^i t$$

   qui s'annule lorsque t augmente, c'est-à-dire

   $$j \in \arg \min_{j \in B \,:\, (B^{-1}A^i)_j > 0} \left\{ \frac{(B^{-1}b)_j}{(B^{-1}A^i)_j} \right\}.$$

   S'il n'existe pas un tel j, le programme linéaire est non borné.
3. Pivotage
   (a) Mettre à jour la base: la variable i entre dans la base et la variable j sort de la base.
   (b) Obtenir le nouveau dictionnaire par pivotage. Aller en 2.

Remarques:

1. L'ajout de variables artificielles est requis (presque toujours) pour obtenir le premier dictionnaire initial. En pratique, les procédures déterminent une solution initiale en utilisant un nombre restreint de telles variables (*crash procedure*).
2. L'inverse de la base n'est pas calculée explicitement; les différents systèmes sont résolus par décomposition de type LU adaptée (deux colonnes seulement sont interchangées).
3. Le choix de la variable d'entrée est coûteux: en pratique, on ne calcule qu'un sous-ensemble des coûts réduits (partial pricing), ce sous-ensemble évoluant dynamiquement au cours de l'algorithme.
4. Différentes règles existent pour le choix de la variable d'entrée dont: *steepest descent* (coût réduit le plus grand), *steepest edge* (plus grande amélioration par rapport à une unité de changement le long de l'arête) et *Devex* (heuristique qui approxime le calcul requis et coûteux pour le steepest edge). En pratique, la méthode *Devex* est utilisée pour la méthode primale du simplexe et le *steepest edge* pour la méthode duale du simplexe (calculs moins coûteux).

2.4.2 Dualité

À chaque programme linéaire est associé un autre programme linéaire appelé dual. Le dual (D) du programme linéaire standard (P) est:

$$\begin{aligned} &\max_y \; b^T y \\ &\text{s.c.} \quad A^T y \leq c. \end{aligned}$$

À partir de la relation primale-duale définie pour la forme standard, il est possible de retrouver le programme dual de tous les programmes linéaires non formulés sous cette forme. Par exemple, le dual du programme suivant:

$$\begin{aligned} &\min_x \; c^T x \\ &\text{s.c.} \quad Ax \geq b \\ &\qquad\quad x \geq 0 \end{aligned}$$

est:

$$\begin{aligned} &\max_y \; b^T y \\ &\text{s.c.} \quad A^T y \leq c \\ &\qquad\quad y \geq 0. \end{aligned}$$

Les solutions des programmes primal et dual sont liées par les théorèmes de la dualité faible et forte.

Proposition 2.4.3 Dualité faible.
Si x est primal réalisable et y est dual réalisable, alors

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

La preuve de ce théorème est immédiate.

Proposition 2.4.4 Dualité forte.
Si x est optimal pour le programme primal, alors le programme dual a une solution optimale telle que $y^T b = c^T x$.

Ici, la preuve est non immédiate. Elle s'élabore soit en conséquence des théorèmes d'alternative, ou bien en utilisant une preuve constructive à l'aide de l'algorithme du simplexe (dont la convergence nécessite une règle d'anti-cyclage...), ou encore en considérant les conditions d'optimalité de KKT d'un programme convexe...

Enfin, un résultat qui sera régulièrement utilisé par la suite:

Proposition 2.4.5 *Soit x primal réalisable et y dual réalisable avec $s = c - A^T y$. Alors, on a*

$$c^T x - b^T y = s^T x.$$

2.4.3 Condition d'optimalité et complémentarité

À partir des résultats précédents, on obtient immédiatement les conditions nécessaires et suffisantes d'optimalité. Supposons le programme primal (P) réalisable et son dual (D) réalisable. On a:

Condition d'optimalité: le point x est optimal pour le programme primal si et seulement si il existe y et s tel que les conditions suivantes sont satisfaites:

$$\begin{aligned} Ax &= b && (3)\\ A^T y + s &= c && (4)\\ x_i s_i &= 0, \quad i = 1,\ldots,n && (5)\\ x,s &\geq 0. && (6) \end{aligned}$$

Ces conditions correspondent aux conditions d'optimalité du premier ordre de Karush-Kuhn-Tucker (les conditions de régularité sont toujours satisfaites). La contrainte (5) met en évidence le caractère complémentaire des solutions optimales du primal et du dual. On dit que la complémentarité est stricte si

$$x_i + s_i > 0, \quad \forall i = 1,\ldots,n.$$

Aux points extrêmes optimaux, cette stricte complémentarité n'est généralement pas satisfaite. La stricte complémentarité est souvent requise pour assurer la convergence des méthodes de points intérieurs. Le théorème suivant, dû à Goldman-Tucker, indique qu'il existe toujours au moins une solution primale-duale qui vérifie la stricte complémentarité.

Si z^* correspond à la valeur optimale du programme linéaire, on définit l'ensemble des solutions du programme primal et du programme dual comme:

$$\begin{aligned} M_P &= \{x \in S_P \;:\; c^T x = z^*\}\\ M_D &= \{(y,s) \in S_D \;:\; y^T b = z^*\} \end{aligned}$$

où S_P et S_D sont respectivement les ensembles réalisables des programmes linéaires primal et dual.

On définit les deux ensembles d'indices suivants:

$$\begin{aligned} \mathcal{B} &= \{j \in \{1,\ldots,n\} \;:\; x_j^* \neq 0, \text{ pour } x^* \in M_P\}\\ \mathcal{N} &= \{j \in \{1,\ldots,n\} \;:\; s_j^* \neq 0, \text{ pour } (y^*,s^*) \in M_D\}. \end{aligned}$$

La proposition suivante indique qu'il existe toujours une partition stricte entre ces deux ensembles.

Proposition 2.4.6 *(Goldman-Tucker, 1956)*

$$\begin{aligned} \mathcal{B} \cup \mathcal{N} &= \{1,\ldots,n\}\\ \textit{et} \quad \mathcal{B} \cap \mathcal{N} &= \emptyset. \end{aligned}$$

Enfin, les résultats précédents nous permettent d'obtenir le tableau des compatibilités primal-dual.

Primal/dual	borné	Non borné	Non réalisable
borné	possible	impossible	impossible
non borné	impossible	impossible	possible
non réalisable	impossible	possible	possible

La seule occurence qui ne se montre pas à l'aide des résultats de dualité faible et forte est le cas non réalisable–non réalisable. On montre ce dernier à l'aide d'un exemple.

3 Méthodes de points intérieurs

3.1 Historique et généralités

La programmation linéaire est certainement l'un des plus beaux succès de la recherche opérationnelle. Il provient, d'une part, de la puissance de modélisation qu'elle offre et ce malgré la limite inhérente qu'impose la linéarité des fonctions impliquées, et d'autre part, de la richesse de la théorie qu'elle a initiée et qui a permis le développement d'algorithmes extrêmement efficaces pour sa résolution. Depuis sa formulation et le développement de la méthode du simplexe pour sa résolution vers la fin des années 40, la programmation linéaire demeure le modèle d'optimisation le plus utilisé par les décideurs ce qui est certainement dû à la robustesse des algorithmes disponibles.

La nature combinatoire de l'algorithme du simplexe a probablement fait en sorte que la programmation linéaire a été longtemps considérée comme une classe de problème distincte de la programmation non linéaire. Lorsque Klee et Minty (1972) ont trouvé un exemple qui montrait pour la première fois que la méthode du simplexe (sous la règle de steepest descent) pouvait prendre un nombre exponentiel d'itérations (une question demeure toutefois ouverte: existe-t-il un algorithme de type pivot et de complexité polynomiale pour résoudre un programme linéaire?), la recherche d'un algorithme polynomial pour la programmation linéaire était lancée. La réponse est venue de Khashiyan (1979) lorsqu'il a montré que la méthode des ellipses (un algorithme de points extérieurs développé au début des années 70 pour la programmation non linéaire) était de complexité globale $O(n^4 L)$ lorsqu'appliqué à la programmation linéaire, où L est le nombre de bits requis pour stocker (et traiter) les données. Cependant, et tandis que la méthode du simplexe, de complexité exponentielle dans le pire des cas, fonctionnait bien en pratique (montrant un très bon comportement polynomial près de $O(3n)$), la méthode des ellipses s'est avérée nettement moins efficace que la méthode du simplexe, le nombre d'itérations requis s'approchant du pire cas prévu par la théorie. Lorsque Karmarkar (1984) a proposé un algorithme polynomial de complexité $O(n^{3.5}L)$ *et* efficace en pratique, basé sur une méthode de points intérieurs, l'intérêt pour ces méthodes principalement développées depuis les années 60 (voir Dikin (1967) et Fiacco et McCormick (1968)), a connu un renouveau pour les problèmes non linéaires et a ouvert un nouveau domaine pour les problèmes linéaires. La distinction entre la programmation linéaire et non linéaire n'était plus aussi nette.

Den Hertog (1994) a classé les méthodes de points intérieurs en quatre catégories:

- Méthodes de chemin central
- Méthodes affines et mise à l'échelle
- Méthodes projectives avec potentiel
- Méthodes affines avec potentiel

L'algorithme de Karmarkar entre dans la classe des algorithmes projectifs avec potentiel et résout un programme linéaire en $O(nL)$ itérations majeures, chacune impliquant la résolution d'un système linéaire dans $\Re^n$ de complexité typique $O(n^3)$, mais réduite à $O(n^{2.5})$ en utilisant une technique de réinversion adaptée, pour obtenir une complexité globale de $O(n^{3.5}L)$, améliorant la complexité $O(n^4L)$ de l'algorithme de Khachiyan (ouf!). En conservant toutefois la même complexité globale que l'algorithme de Karmarkar, Renegar (1988) a montré que la méthode des centres (qui entre dans la catégorie des méthodes de chemin central) requiert $O(\sqrt{n}L)$ itérations majeures. La meilleure complexité $O(n^3L)$ connue à ce jour a initialement été obtenue par Gonzaga (1989) (barrière logarithmique) et Vaidya (1990)(méthode des centres). Aujourd'hui, plusieurs algorithmes partagent cette complexité, en particulier les algorithmes de type primal-dual discutés à la section 7.

La méthode affine primale a été proposée par Dikin (1967) et la convergence prouvée sous des hypothèses de non dégénérescence primale. Les approches affines primales (ou duales) ne sont pas actuellement pas prouvées polynomiales. Par contre, Monteiro et al (1990) ont démontré que l'approche affine primale-duale est de complexité polynomiale ($O(nL^2)$). Une approche primale avec centrage et utilisation d'une fonction potentiel a été montrée polynomiale par Barnes et al (1988). Une variante de ce dernier algorithme a permis à Mizuno et Nagasawa (1993) de montrer une complexité de $O(\sqrt{n}L)$ pour l'approche primale-duale.

Les méthodes de points intérieurs ont fait récemment l'objet de plusieurs monographies dont Wright (1997), den Hertog (1994), Ye (1997), Roos, Terlaky et Vial (1997) et Nesterov et Nemirovskii (1994). Un excellent survol des méthodes de points intérieurs pour la programmation linéaire est celui de Gonzaga (1992).

3.2 Complexité

La théorie de la complexité s'intéresse à l'ordre de grandeur du nombre requis d'opérations (en terme d'itérations ou d'opérations arithmétiques ou encore d'opérations sur les bits) pour déterminer, dans le pire des cas, une solution exacte d'une instance donnée d'une classe de problème. Un algorithme polynomial garantit l'obtention d'une solution optimale pour chaque instance d'un problème en un temps donné par un polynôme d'une certaine mesure de la taille du problème.

Différents modèles existent selon les opérations considérées. La complexité algorithmique des méthodes de points intérieurs réfère presque exclusivement au modèle rationnel (rational-number model) où les opérations arithmétiques sur des nombres rationnels peuvent se ramener à des opérations sur des entiers. Sous ce modèle, il est nécessaire de conserver la rationalité des nombres traités par l'algorithme. C'est en particulier le cas si l'instance considérée est constituée au départ de nombres rationnels et que seules les opérations élémentaires ($+, -, \times, \div$) sont appliquées au cours de l'algorithme. Dans plusieurs algorithmes de points intérieurs, la résolution d'une équation quadratique est notamment requise nécessitant l'utilisation de l'opérateur racine carrée $\sqrt{\cdot}$ qui ne conserve pas la rationnalité. Il est alors nécessaire de montrer qu'une solution rationnelle approchée est suffisante pour la convergence de l'algorithme.

Une complexité polynomiale sur un tel modèle requiert également que la taille de tout nombre rationnel produit par l'algorithme soit bornée supérieurement par un polynôme en la taille du problème. Tous les algorithmes présentés dans ce cours et appliqués à des instances avec données rationnelles (ou entières) produiront des itérés rationnels dont la taille est bornée par un polynôme en la taille de l'instance (voir Schrijver (1986)).

Nous utiliserons la définition suivante:

Définition 3.2.1 *Une méthode est dite polynomiale si le nombre d'opérations élémentaires pour déterminer une solution optimale est borné par un polynôme en la taille L du problème.*

Une méthode de points intérieurs itère sur des points strictement intérieurs et converge à la limite vers une solution du problème. Comme les itérés (en x et s) sont strictement positifs (violant la condition de complémentarité d'un programme linéaire), aucun itéré ne peut être la solution du problème. Pour conserver une complexité polynomiale, il est donc nécessaire dans une première étape de calculer une solution dont la valeur est *proche* de la valeur optimale, et dans une seconde étape, d'*arrondir* cette solution par une procédure finie ad hoc pour obtenir la solution optimale.

Considérons le programme linéaire:

$$\begin{array}{lrcl} \min\limits_x & c^T x & & \\ \text{s.c.} & Ax & = & b \\ & x & \geq & 0. \end{array}$$

On suppose que A est de plein rang et que tous les coefficients du programme sont entiers. On définit $l = l_1 + l_c$ comme étant égal au nombre total de bits requis pour stocker A,b,c (l_1 pour stocker A et b et l_c pour stocker c). On peut poser:

$$l = \sum_{i=0}^{m}\sum_{j=0}^{n}\lceil \lg_2(|a_{ij}|+1)+1\rceil$$

avec $a_{i0} \equiv b_i$ et $a_{0j} \equiv c_j$.
Remarquons qu'aucune opération élémentaire sur les données (sans répétition) ne peut donner un résultat supérieur à 2^l. Ceci implique que $|\det(M)| < 2^l$ où M est une matrice carrée issue des éléments de A,b,c (ce nombre étant dominé par le produit des normes des colonnes).
Posons $L = 2l + n + 1$.

Lemme 3.2.1 *(Gonzaga, 1995)*
Soit v la valeur optimale d'un programme linéaire réalisable et borné. Si x est un point extrême, alors on a:
i) $\forall i = 1,\ldots,n, x_i = 0$ ou $x_i > 2^{-L}$;
ii) Soit $c^T x = v$ ou bien $c^T x > v + 2^{-L}$; et
iii) Si le domaine réalisable est borné, alors $c^t x \leq 2^L$.

Preuve: i) Soit x un point extrême. On a

$$Ax = b \Rightarrow A_B x_B + A_N x_N = b$$

avec $x_i = 0, \forall i \in N$ et $x_i > 0, \forall i \in B$ (cette partition ne correspond pas nécessairement à une base..., mais les colonnes de A_B sont linéairement indépendantes). Si A_B n'est pas carrée, on élimine les contraintes redondantes. On a par la règle de Cramer:

$$x_i = \frac{\det(A_B^i|b)}{\det(A_B)} = \frac{\Delta_i}{\Delta}.$$

On a de plus:

$$\begin{aligned}\frac{|\Delta_i|}{|\Delta|} &\geq \frac{1}{2^{l_1}}\\ &> 2^{-L}\end{aligned}$$

puisque Δ_i est un entier.
ii) Soit x^* un point extrême optimal. On a alors

$$x_i^* = 0 \quad \text{ou} \quad x_i^* = \frac{\Delta_i^*}{\Delta^*}$$

avec $\Delta^* x_i^*$ entier, $i \in B^*$. On en déduit que $\Delta\Delta^* c^T(x - x^*)$ est également un entier. Alors soit:

$$c^T(x - x^*) = 0,$$

soit

$$c^T(x - x^*) \geq \frac{1}{\Delta\Delta^*} \geq 2^{-2l_1} > 2^{-L}.$$

iii) Soit x un point extrême. Pour $i \in B$, on a:

$$x_i = \frac{\Delta_i}{\Delta} \leq 2^{l_1}$$

et donc

$$e^T x \leq n * 2^{l_1} \leq 2^L.$$

Comme

$$c_i x_i \leq 2^{l_1 + l_c} = 2^l,$$

on a

$$c^T x \leq n 2^l \leq 2^L.$$

Par le théorème de Minkowski et puisque l'ensemble réalisable est borné, on peut écrire tout point réalisable comme combinaison convexe des points extrêmes:

$$\begin{aligned} x &= \sum_{j\in J}\lambda_j x^j\\ \sum_{j\in J}\lambda_j &= 1\\ \lambda_j &\geq 0.\end{aligned}$$

On obtient:

$$c^T x = \sum_{j\in J}\lambda_j c^T x^j \leq 2^L \sum_{j\in J}\lambda_j = 2^L.$$

□

La prochaine proposition indique qu'à partir d'un point réalisable, il est possible de déterminer en temps polynomial un point extrême de valeur inférieure ou égale.

Proposition 3.2.1 *(Kortanek et Zhu, 1988)*
Pour tout point x réalisable, il existe une procédure (dite de purification) qui trouve en $O(n^3)$ un point extrême $\tilde{x}$ tel que $c^T\tilde{x} \leq c^T x$.

Cette procédure est de type pivot et requiert $O(n)$ pivots. Soit maintenant x tel que

$$c^T x \leq v + 2^{-L}$$

avec $v = c^T x^*$. L'algorithme de purification trouve un point extrême $\tilde{x}$ tel que

$$c^T\tilde{x} \leq c^T x \leq v + 2^{-L}.$$

En appliquant le lemme 3.2.1, on déduit immédiatement que le point extrême $\tilde{x}$ est optimal.

4 Méthode des ellipses

La méthode des ellipses résout un problème de réalisabilité stricte du type:

Déterminer un point x tel que $Ax < b$, ou montrer qu'un tel point n'existe pas.

Il est facile de ramener un programme linéaire sous la forme d'un problème de réalisabilité avec inégalités non strictes.

Proposition 4.0.2 *Tout programme linéaire peut se ramener à un problème de réalisabilité avec inégalités non strictes.*

Preuve: Exercice.

On note par e le vecteur unitaire (dont toutes les composantes sont égales à 1). La proposition suivante, dont la preuve utilise des arguments similaires au lemme 3.2.1 de la section précédente, relie le problème de réalisabilité avec inégalités non strictes à celui avec inégalités strictes.

Proposition 4.0.3 *(Kaschiyan, 1979)*
Soit une matrice A et un vecteur b dont les composantes sont entières. Si le système avec inégalités strictes $Ax < b + 2^{-L}e$ possède une solution, alors:
i) le système $Ax \leq b$ possède également une solution; et
ii) le volume de l'ensemble réalisable est au moins $2^{-(n+1)2L}$.

La méthode des ellipses est une méthode de points extérieurs et débute par une ellipse E_0 dont le volume est suffisamment grand pour contenir une partie S_0 de la région réalisable, si cette dernière est non vide. À chaque itération, le centre x_k de l'ellipse E_k est calculé et sa réalisabilité vérifiée. Si le centre est réalisable, l'algorithme termine avec x_k comme solution.

Sinon on définit une nouvelle ellipse de volume minimal E_{k+1} à partir de E_k et d'une inégalité non satisfaite ($a_i^T x_k \geq b_i$) telle que

$$\begin{array}{rcl} E_k \cap \{x : a_i^t x \leq a_i^t x_k\} & \subset & E_{k+1} \\ \text{et } E_{k+1} \cap \{x : a_i^t x = a_i^t x_k\} & = & E_k \cap \{x : a_i^t x = a_i^t x_k\}. \end{array}$$

À chaque itération, le volume de l'ellipse est réduit d'un facteur constant. Ainsi, ou bien une solution est trouvée, ou bien le volume de l'ellipse devient plus petit que la borne inférieure sur le volume de S_0, ce qui implique que l'ensemble réalisable est vide.

Méthodes des ellipses
i) Poser l'ellipse initiale

$$E_0 = \{x : (x - x_0)^t M_0^{-1}(x - x_0) \leq 1\},$$

où M_0 est une matrice symétrique définie positive et telle que l'ellipse E_0 contient une partie de $\{x : Ax < b\}$ si ce dernier ensemble est non vide. Un exemple d'une telle matrice est $M_0 = 2^L I$.

ii) (Itération k) Si x_k est le centre de E_k et est tel que $Ax_k < b$, alors on arrête.

Si $k > 6(n+1)^2 L$, on arrête: l'ensemble réalisable est vide.

Sinon à partir d'une inégalité non satisfaite ($a_i^T x_k \geq b_i$), on pose

$$\begin{array}{rcl} x_{k+1} & = & x_k - \dfrac{M_k a_i}{(n+1)\sqrt{a_i^T M_k a_i}} \\ M_{k+1} & = & \dfrac{n^2}{n^2-1}(M_k - \dfrac{2(M_k a_i)(M_k a_i)^T}{(n+1)a_i^T M_k a_i}. \end{array}$$

La proposition suivante donne le taux de réduction du volume de chaque ellipse.

Proposition 4.0.4
On a

$$vol(E_{k+1}) = c(n) vol(E_k)$$

avec

$$c(n) = (\frac{n^2}{n^2-1})^{\frac{n-1}{2}} \frac{n}{n+1}.$$

Comme $c(n) < e^{\frac{-1}{2(n+1)}}$ et que le volume de l'ellipse initiale est borné supérieurement par $2^{(n+1)L}$, il est facile de montrer qu'après $6L(n+1)^2$ itérations, si l'algorithme n'a pas trouvé de solutions, le volume de l'ellipse sera inférieur à $2^{-(n+1)2L}$.

Chaque itération de cet algorithme requiert $O(mn + n^2)$ opérations. Au total, la méthode des ellipses requiert $O((mn^3 + n^4)L)$ opérations.

5 Méthode affine primale

On présente la méthode affine primale comme introduction aux méthodes de points intérieurs. Cette méthode se caractérise par sa simplicité et sa relative efficacité pratique. Par contre, la complexité polynomiale n'est pas connue pour l'approche primale (ou duale), alors qu'elle est établie pour l'approche primale-duale. De plus l'analyse de convergence est davantage complexe que pour les algorithmes de chemin central.

L'idée consiste à utiliser à partir d'un point intérieur la direction projetée de plus forte descente dans un espace transformé. La direction de plus forte descente est parmi toutes les directions réalisables celle qui donne le plus grand taux de décroissance. Cette propriété est utile si le point courant se trouve éloigné des frontières définies par les bornes de non négativité. Dans le cas contraire, le pas effectué dans cette direction sera petit donnant une décroissance nette petite de l'objectif. Ainsi, l'approche consiste à calculer une direction de descente qui n'approche pas trop *rapidement* de la frontière. Cette direction de plus forte descente est calculée à partir d'un problème mis à l'échelle qui *centre* la solution courante et qui est ensuite transformée dans l'espace original. L'algorithme comprend les trois étapes suivantes: calcul de la direction de descente, calcul du pas et calcul de la transformation affine. (Discussion en classe sur l'équivalence entre cette approche et la minimisation sur une ellipse inscrite dans S.)

On considère le programme linéaire:

$$\begin{array}{lrcl} \min_x & c^T x & & \\ \text{s.c.} & Ax & = & b \\ & x & \geq & 0. \end{array}$$

avec la matrice A de plein rang. On note S^0 l'intérieur de S, c'est-à-dire:

$$S^0 = \{x \in \Re^n \ : \ Ax = b,\, x > 0\}.$$

On suppose que l'ensemble S^0 n'est pas vide.

Calcul de la direction de descente

Soit x_c le point courant tel que:

$$\begin{array}{lrcl} & Ax_c & = & b \\ \text{et} & x_c & > & 0. \end{array}$$

On cherche

$$x^+ = x_c + \alpha\Delta x$$

tel que

$$\begin{array}{lrcl} & c^T x^+ & \leq & c^T x_c \\ \text{et} & Ax^+ & = & b. \end{array}$$

Le déplacement doit donc vérifier:

$$\begin{array}{lrcl} & c^t\Delta x & \leq & 0 \\ \text{et} & Ax^+ & = & A(x_c + \alpha\Delta x) = b. \end{array}$$

Sous l'hypothèse que $\alpha > 0$, Δx doit être dans le noyau de A, c'est-à-dire

$$\Delta x \in \mathcal{N}\ (A) \quad = \quad \{x \in \Re^n | Ax = 0\}.$$

La direction de plus forte descente est donnée par

$$\begin{array}{lrcl} \min_{\Delta x} & c^T\Delta x & & \\ \text{s.c.} & A\Delta x & = & 0 \\ & \|\Delta\| & = & 1 \end{array}$$

dont la solution est:

$$\frac{\text{proj}_A(-c)}{\|\text{proj}_A(-c)\|} = \Delta x,$$

$\text{proj}_A()$ étant la matrice orthogonale de projection sur le noyau de A. Comme A est de plein rang et en oubliant la normalisation, on a:

$$\begin{array}{rcl} \Delta x & = & \text{proj}_A(-c) \\ & = & -(I - A^T(AA^T)^{-1}A)c. \end{array}$$

Rappel: Une matrice orthogonale de projection P sur le noyau de A vérifie les propriétés suivantes:

$$\begin{array}{rcl} AP & = & 0 \\ P & = & P^T \\ P^2 & = & P. \end{array}$$

Il est alors facile de vérifier que:

$$\begin{array}{rcl} c^T\Delta x & = & -c^T\text{proj}_A(c) \\ & = & -c^T\text{proj}_A^2(c) \\ & = & -\|\text{proj}_A(c)\|^2 \leq 0. \end{array}$$

Longueur de pas

Puisque nous sommes dans le cas linéaire, le taux de décroissance est constant dans la direction Δx. Le pas maximal est limité par les contraintes de non négativité. De plus, la transformation affine qui nous permettra de centrer le point n'est pas définie sur la frontière. Donc, il suffit de choisir:

$$\alpha = \gamma\alpha_{\max} \quad \text{avec} \quad 0 < \gamma < 1,$$

et

$$\alpha_{\max} = \min_{\Delta x_i < 0} \frac{-(x_c)_i}{\Delta x_i}.$$

En pratique, on choisit $\gamma = 0.995$. Si $\Delta x \geq 0$ et $\Delta x \neq 0$, le problème est alors non borné.

Transformation affine

On fait une mise à l'échelle afin que le nouveau point x^+ soit loin de la frontière définie par $x \geq 0$. Un point *idéal* serait le vecteur unitaire e. Ainsi, on cherche une transformation affine qui transforme x^+ en e. La matrice de transformation est simplement l'inverse de la matrice diagonale dont les composantes sont les mêmes que celles de x^+. Cette matrice est inversible puisque $x^+ > 0$. Notons cette matrice par X. On a alors

$$\begin{array}{lrcl} & X^{-1}x^+ & = & e \\ \text{et notons} & X^{-1}x & = & \overline{x}. \end{array}$$

Le programme linéaire devient (dans l'espace transformé)

$$\begin{array}{lrcl} \min_{\overline{x}} & c^T X\overline{x} & = & \overline{c}^T\overline{x} \\ \text{s.c.} & AX\overline{x} & = & \overline{A}\overline{x} = b \\ & \overline{x} & \geq & 0. \end{array}$$

Dans l'espace-$\overline{x}$ et étant donné le nouveau point courant $\overline{x}_c = e$, on obtient:

$$\begin{array}{rcl} \Delta\overline{x} & = & \text{proj}_{\overline{A}}(-\overline{c}) \\ & = & -(I - \overline{A}^t(\overline{AA}^t)^{-1}\overline{A})\overline{c} \\ & = & -(I - XA^t(AX^2A^t)^{-1}AX)Xc. \end{array}$$

et

$$\begin{aligned}
\overline{x}^+ &= \overline{x}_c + \alpha\Delta\overline{x} \\
\Rightarrow \quad x^+ &= X\overline{x}^+ \\
&= x_c + \alpha X\Delta\overline{x} \\
&= x_c - \alpha X(I - XA^t(AX^2A^t)^{-1}AX)Xc.
\end{aligned}$$

Le passage à l'espace-$\overline{x}$ n'est évidemment pas nécessaire.

Convergence
On a le résultat suivant:

Théorème 5.0.1 *Sous les hypothèses que A est une matrice de plein rang, qu'il existe un point strictement intérieur et que la fonction objectif est non constante sur le domaine réalisable, alors:*

1. *Si le problème primal et le problème dual sont non dégénérés, alors pour tout $\gamma < 1$, la suite générée par l'algorithme converge vers une solution optimale.*
2. *Pour $\gamma \leq 2/3$, la suite générée par l'algorithme converge vers une solution optimale.*

La preuve (relativement complexe) de ce résultat peut être trouvée dans Saigal (1995). Notons qu'il existe un seul exemple connu (Vanderbei, 1996) où l'algorithme ne converge pas ($\forall\gamma > 0.995$). Rappelons qu'aucun résultat de complexité polynomiale n'est connu pour la méthode affine primale (et méthode affine duale). La méthode affine primale-duale, où la direction de descente est calculée dans l'espace primal et dual et où la mise à l'échelle transforme le vecteur de complémentarité XSe vers le point e, a une complexité polynomiale de $O(nL^2)$ itérations majeures.

Critère d'arrêt
Le critère d'arrêt devrait être relié à la satisfaction des conditions d'optimalité. Considérons d'abord le cas non dégénéré (primal et dual) et soit le programme dual (D):

$$\begin{aligned}
&\max_{y,s} \; b^Ty \\
\text{s.c.} \quad & A^Ty + s = c \\
& s \geq 0.
\end{aligned}$$

Soit x_k la solution courante et considérons le programme

$$\begin{aligned}
&\min_{y,s} \; ||X_ks|| \\
\text{s.c.} \quad & A^Ty + s = c \\
& s \geq 0.
\end{aligned}$$

La solution de ce programme est donnée par

$$\begin{aligned}
y_k &= (AX_k^2A^T)^{-1}AX_k^2c \\
s_k &= c - A^Ty_k.
\end{aligned}$$

On en déduit:

$$\Delta x_k = -X_k^2 s_k,$$

et le vecteur dual est obtenu comme sous-produit du calcul de la direction de descente.

Théorème 5.0.2 *Sous l'hypothèse de non dégénérescence primale et duale, la solution(y_k,s_k) converge vers la solution optimale duale lorsque x_k converge vers la solution optimale primale.*

Dans ce cas, un critère d'arrêt peut être défini sur la norme du vecteur de complémentarité.
Dans le cas dégénéré, la convergence de (y_k,s_k) n'est pas assurée, et un critère d'arrêt classique sur l'amélioration successive de deux itérés est utilisé. Le détail de l'algorithme est donné au tableau 1.

```
Initialisation:
  γ (0 < γ < 1);
  x_0 le point initial;
  Δc la décroissance de la fonction objective;
  ϵ le critère d'arrêt;
début
  x_c := x_0;
  Δc := ϵ(c^T x_c) + 1;
  tant que Δc / (c^T x_c) > ϵ faire
  début
    X := X_c;
    Δx := −X(I − XA^t(AX^2A^t)^{−1}AX)Xc;
    α = γ min_{Δx_i<0} −(x_c)_i / Δx_i
    x^+ := x_c + αΔx
    Δx := c^T x_c − c^T x^+;
    x_c := x^+;
  fin
fin
```

TAB. 1 – *Méthode affine primale pour la programmation linéaire*

6 Méthode duale de chemin central (barrière logarithmique)

Dans cette section, nous développons la méthode duale de chemin central. Cette méthode s'inscrit dans le cadre des méthodes de minimisation séquentielle de programmes non contraints (SUMT) avec ici, l'utilisation d'une barrière logarithmique. Nous y définissons la notion de chemin central qui est fondamentale à l'analyse de complexité des méthodes de points intérieurs. Il sera en effet possible de montrer que la méthode de Newton (voir Dennis et Schnabel, 1983), dans un certain voisinage du chemin central, possède une convergence globale et quadratique vers un point du chemin central.

Nous définissons dans un premier temps la barrière logarithmique en considérant un programme convexe et établissons le lien entre le chemin central et les conditions d'optimalité des programmes non contraints. Un algorithme générique de barrière algorithmique est ensuite présenté. Nous allons nous concentrer sur la programmation linéaire pour laquelle l'analyse de convergence sera faite pour la méthode à petit pas. Cette section suit essentiellement le développement de la méthode tel que présentée dans den Hertog (1994).

6.1 Barrière logarithmique

Considérons le programme convexe suivant:

$$\begin{array}{lccl} \max\limits_{y} & f_0(y) & & \\ \text{s.c.} & f_i(y) & \leq & 0, \quad i = 1, \ldots, n. \end{array}$$

où $-f_0(y)$ et $f_i(y)$,$i = 1, \ldots, n$ sont des fonctions convexes au moins deux fois continûment différentiables de $\Re^m \to \Re$. On pose les hypothèses suivantes:

$$\begin{array}{l} A_1 : \operatorname{int}(S) = S^0 \neq \emptyset \ \ \text{(condition de Slater)}, \\ A_2 : S \text{ est borné}. \end{array}$$

Le conditions d'optimalité du premier ordre (KKT), nécessaires et suffisantes sous les hypothèses posées, sont l'existence d'un point y et d'un vecteur de multiplicateurs x qui satisfont:

$$\nabla f_0(y) - \sum_{i=1}^{n} x_i \nabla f_i(y) = 0 \tag{7}$$

$$f_i(y) \leq 0, \qquad i = 1, \ldots, n \tag{8}$$

$$x_i f_i(y) = 0, \qquad i = 1, \ldots, n \tag{9}$$

$$x_i \geq 0, \qquad i = 1, \ldots, n. \tag{10}$$

Considérons maintenant la fonction barrière suivante:

$$B(y) = -\sum_{i=1}^{n} \ln(-f_i(y))$$

et le programme non contraint:

$$\begin{aligned} \min_{y} \ \phi_B(y, \mu) &= -\frac{f_0(y)}{\mu} + B(y) && (11) \\ &= -\frac{f_0(y)}{\mu} - \sum_{i=1}^{n} \ln(-f_i(y)). \end{aligned}$$

où la constante $\mu > 0$ est le paramètre de pénalité.

Rappel: Une fonction barrière $B(x)$ pour le programme mathématique

$$\begin{array}{ll} \min & f(x) \\ \text{s.c.} & g_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

est une fonction non négative et continue sur le domaine réalisable qui tend vers ∞ lorsque x tend vers la frontière à partir de l'intérieur du domaine. On définit souvent une fonction barrière sous la forme:

$$B(x) = \sum_{i=1}^{m} \theta[g_i(x)],$$

où la fonction θ vérifie:

1. $\theta(y) \geq 0$ si $y < 0$
2. $\lim_{y \to 0^-} \theta(y) = \infty$.

Remarque: La barrière logarithmique utilisée ne vérifie ces conditions que dans un voisinage de 0.

Le gradient de la fonction ϕ_B est:

$$\nabla \phi_B(y, \mu) = g(y, \mu) = -\frac{\nabla f_0(y)}{\mu} + \sum_{i=1}^{n} \frac{\nabla f_i(y)}{-f_i(y)}$$

et la matrice hessienne:

$$\begin{aligned} \nabla^2 \phi_B(y, \mu) &= H(y, \mu) \\ &= -\frac{\nabla^2 f_0(y)}{\mu} + \sum_{i=1}^{n} \left(\frac{\nabla^2 f_i(y)}{-f_i(y)} + \frac{\nabla f_i(y)^T \nabla f_i(y)}{(f_i(y))^2} \right) \end{aligned}$$

Il est facile de voir que la matrice hessienne est semi-définie positive pour $y \in S^0$ sous les hypothèses A_1 et A_2 posées. Elle est définie positive sous des conditions un peu plus fortes.

Proposition 6.1.1 *Sous les hypothèses A_1 et A_2, la matrice hessienne $H(y, \mu)$, $y \in S^0$ et $\mu > 0$, est définie positive dans les deux cas suivants:*

1. *les fonctions $f_i(y)$,$i = 0, \ldots, n$ sont linéaires (affines);*
2. *la fonction $-f_0(y)$ est convexe et les fonctions $f_i(y)$,$i = 1, \ldots, n$ sont quadratiques convexes.*

Preuve: Exercice.

L'exemple suivant (den Hertog, 1994) montre que les hypothèses ne sont pas suffisantes dans le cas convexe général pour assurer la positivité de la matrice hessienne.

Exemple:

$$\begin{aligned} f_0(y) &= 0 \\ f_1(y) &= \begin{cases} y^4 - 1 & y \leq 0 \\ -1 & 0 < y < 1 \\ (y-1)^4 - 1 & y \geq 1. \end{cases} \end{aligned}$$

Dans ce cas, $y^T H y = 0$ pour $0 < y < 1$.

Ainsi, la fonction barrière logarithmique $\phi_B(y,\mu)$ est strictement convexe pour le cas linéaire et convexe/quadratique convexe. Dans le cas général convexe, certaines conditions supplémentaires (*self-concordance conditions*, voir Nesterov et Nemirovski, 1994) doivent être imposées pour assurer la stricte convexité de la fonction $\phi_B(y,\mu)$. On suppose ici que ces conditions sont vérifiées.

La fonction barrière $\phi_B(y,\mu)$ est donc strictement convexe sur le domaine borné $\mathrm{int}(S)$ et prend des valeurs infinies sur la frontière de S. Elle atteint alors son minimum, pour $\mu > 0$ fixé, en un point unique de $\mathrm{int}(S)$ noté $y(\mu)$.

Les conditions nécessaires et suffisantes d'optimalité d'un point y du problème non contraint (11) sont données par les équations suivantes (voir la section 2.2):

$$\begin{aligned} \nabla\phi_B(y,\mu) &= 0 \\ f_i(y) &\le 0, \quad i = 1,\ldots,n. \end{aligned}$$

Ces conditions sont satisfaites si et seulement si:

$$\begin{aligned} \nabla f_0(y) + \mu \sum_{i=1}^{n} \frac{\nabla f_i(y)}{f_i(y)} &= 0 \\ f_i(y) &\le 0, \quad i = 1,\ldots,n \end{aligned}$$

ou par un changement de variables:

$$\begin{aligned} \nabla f_0(y) - \sum_{i=1}^{n} x_i \nabla f_i(y) &= 0 && (12) \\ f_i(y) &\le 0, \quad i = 1,\ldots,n && (13) \\ -x_i f_i(y) &= \mu, \quad i = 1,\ldots,n && (14) \\ x_i &\ge 0, \quad i = 1,\ldots,n. && (15) \end{aligned}$$

Ces conditions sont semblables à celles présentées plus haut dans cette section (7)-(10) pour le programme convexe. En fait, seule la contrainte de complémentarité est perturbée.

6.2 Chemin central et centre analytique

On peut maintenant donner la définition du chemin central.

Définition 6.2.1 *On définit le* **chemin central** *comme le chemin paramétré formé de la suite de points $y(\mu)$ lorsque l'on fait varier μ de $+\infty$ à 0^+.*

Ce chemin possède des propriétés intéressantes:

Théorème 6.2.1 *(Fiacco-McCormick (1968)) La fonction $y(\mu)$ est continûment différentiable. De plus, on a que*

$$\lim_{\mu \downarrow 0} y(\mu) = y^*.$$

où y^* est une solution optimale.

Il est bon de noter que si la solution du système perturbé (12)-(15) est unique pour tout $\mu > 0$, la solution du système avec $\mu = 0$ (qui coïncide alors avec le système (7)-(10) n'est pas nécessairement unique.

Définition 6.2.2 *Soit $S = \{y : f_i(y) \le 0, i = 1,\ldots,n\}$ un ensemble borné et tel que $int(S) \ne \emptyset$. Le* **centre analytique** *de S est le point qui est solution du programme:*

$$\max \sum_{i=1}^{n} \ln(-f_i(y)).$$

Le centre analytique est en fait le point de départ du chemin central et est solution du programme

$$\min \ \phi_B(y,\mu)$$

lorsque $\mu \to \infty$. Il est facile de voir que le centre analytique est le point intérieur qui maximise le produit des variables d'écart. Ce point est donc dépendant de la représentation algébrique de l'ensemble réalisable. Dédoubler une contrainte éloignera le centre analytique de la frontière définie par cette contrainte, sans que la géométrie de l'ensemble soit modifiée.

Exemple: Trouver le chemin central de

$$\begin{gathered} \max \ y_1 \\ -y_1 \le 0 \\ -y_2 \le 0 \\ y_1 + y_2 \le 1. \end{gathered}$$

On a

$$\phi_B(y,\mu) = \frac{-y_1}{\mu} - \ln(y_1) - \ln(y_2) - \ln(1 - y_1 - y_2)$$

et

$$\nabla\phi_B(y,\mu) = [-\frac{1}{\mu} - \frac{1}{y_1} + \frac{1}{1 - y_1 - y_2}, -\frac{1}{y_2} + \frac{1}{1 - y_1 - y_2}].$$

Les conditions nécessaires et suffisantes d'optimalité sont:

$$\begin{aligned} -\frac{1}{\mu} - \frac{1}{y_1} + \frac{1}{1 - y_1 - y_2} &= 0 \\ -\frac{1}{y_2} + \frac{1}{1 - y_1 - y_2} &= 0 \\ y_1 &\ge 0 \\ y_2 &\ge 0 \\ y_1 + y_2 &\le 1. \end{aligned}$$

Après quelques calculs, on obtient le vecteur solution $y(\mu)$:

$$\begin{aligned} y_2(\mu) &= \frac{(1 + 3\mu) - \sqrt{9\mu^2 - 2\mu + 1}}{4} \\ y_1(\mu) &= -2y_2 + 1. \end{aligned}$$

La solution optimale est:

$$\lim_{\mu \downarrow 0}(y_1(\mu),y_2(\mu)) = (1,0)$$

et le centre analytique:

$$\lim_{\mu \to \infty}(y_1(\mu),y_2(\mu)) = (1/4,1/2).$$

6.2.1 Algorithme de barrière logarithmique

L'algorithme de barrière logarithmique est une méthode de points intérieurs qui, à partir d'un point initial situé près du chemin central, suit approximativement ce dernier jusqu'à la solution optimale.

Algorithme dual de chemin central:
Soit $\mu > 0$ et un critère d'arrêt ϵ donné.
Faire (itération majeure)
  résoudre (approximativement) min $\phi_B(y,\mu)$ et obtenir $\approx y(\mu)$
  mettre à jour $\mu = (1-\theta)\mu$,avec $0 < \theta < 1$
tant que $\mu > \epsilon$.

L'analyse de convergence et de complexité requiert évidemment plus de détails comme le choix de l'estimé initial, le détail du calcul de la solution approchée de $y(\mu)$, la valeur de θ et de ϵ, etc. L'algorithme que nous allons analyser pour le cas linéaire est celui présenté dans den Hertog (1994). Pour μ fixé, le problème d'optimisation est résolu par la méthode de Newton et la notion de proximité (pour évaluer la qualité de l'approximation) est basée sur la norme induite par la matrice hessienne (définie positive) du pas de Newton.

Le tableau 2 présente l'algorithme général de barrière logarithmique.

Selon la valeur de θ (facteur de réduction), on parlera de méthode à:

1. long pas: $0 < \theta < 1$, indépendant de n et ϵ;
2. pas moyen: $\theta = \nu/\sqrt{n}$, où $\nu > 0$ est indépendant de n et ϵ; et
3. petit pas: $\theta = \nu/\sqrt{n}$, où $\nu > 0$ est suffisamment petit de façon à ce qu'un seul pas de Newton soit suffisant pour atteindre le voisinage du nouveau centre.

Nous ferons l'analyse de convergence et de complexité pour la méthode à petit pas appliquée à un programme linéaire.

6.3 Cas linéaire

On considère le programme linéaire:

$$\begin{aligned} \max_y \ & b^T y \\ \text{s.c.} \ & A^T y \leq c, \end{aligned}$$

Initialisation:
  ϵ est le paramètre de précision;
  τ est le paramètre de proximité;
  θ est le paramètre de réduction, $(0 < \theta < 1)$;
  μ_0 est la valeur initiale de la pénalité;
  y^0 est un point intérieur tel que
    $||p(y^0,\mu_0)||_{H(y^0,\mu_0)} \leq \tau$.

début
  $y := y^0; \mu := \mu_0$;
  tant que $\mu > \frac{\epsilon}{4n}$ faire
  début (itération majeure)
    $\mu := (1-\theta)\mu$;
    tant que $||p||_H \geq \tau$ faire
    début (itération mineure)
      $\tilde{\alpha} = \arg\min_{\alpha > 0}\{\phi_B(y+\alpha p,\mu) : y + \alpha p \in S^0\}$
      $y := y + \tilde{\alpha} p$
    fin (itération mineure)
  fin (itération majeure)
fin

TAB. 2 – *Algorithme général de barrière logarithmique (den Hertog)*

qui est le dual de:

$$\begin{aligned} \min_x \ & c^T x \\ \text{s.c.} \ & Ax = b \\ & x \geq 0. \end{aligned}$$

On suppose toujours que $A_{m\times n}$ est de plein rang, que $\text{int}(S) \neq \emptyset$ et que S est borné. La fonction barrière (en posant $s = c - A^T y$) est alors:

$$\phi_B(y,\mu) = -\frac{b^T y}{\mu} - \sum_{i=1}^{n}(\ln(s_i))$$

et le gradient:

$$\begin{aligned} \nabla\phi_B(y,\mu) &= -\frac{b}{\mu} - \nabla(\sum_{i=1}^{n}\ln((c - A^T y)_i)) \\ &= -\frac{b}{\mu} + AS^{-1}e, \end{aligned}$$

où S est la matrice diagonale formée à partir du vecteur s. La matrice hessienne est (exercice):

$$\begin{aligned} H(y,\mu) &= AS^{-1}S^{-1}A^T \\ &= AS^{-2}A^T. \end{aligned}$$

Il est facile de voir que la matrice $H(y,\mu)$ est définie positive puisque $s > 0$ et A est de plein rang.

Les conditions d'optimalités du premier ordre donnent:

$$\begin{array}{rrcl} & \nabla\phi_B(y,\mu) & = & 0 \\ \text{d'où} & \mu AS^{-1}e & = & b. \end{array}$$

Posons $x = \mu S^{-1}e$. Ainsi, on doit résoudre (approximativement) à chaque itération majeure (pour $\mu > 0$ fixé)

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ Sx &= \mu e \\ s &> 0 \\ x &> 0 \end{aligned}$$

afin d'approcher la solution unique $y(\mu)$ du chemin central. Notons que le gap d'optimalité d'un point sur le chemin central est simplement $x^T(\mu)s(\mu) = n\mu$.

Une notion de voisinage a été proposée par Roos et Vial (1992).

Définition 6.3.1 *On définit la distance de y à $y(\mu)$ par*

$$\delta(y,\mu) \quad = \quad \min_{\{x:Ax=b\}} ||\frac{Sx}{\mu} - e||.$$

La distance $\delta(y,\mu)$ est une mesure de la violation de la contrainte $Sx = \mu e$.

Proposition 6.3.1 *La solution de*

$$\begin{array}{ll} \min\limits_x & ||\frac{Sx}{\mu} - e|| \\ s.c. & Ax = b \end{array}$$

est unique et est égale à

$$x(y,\mu) = \mu S^{-1}e - S^{-2}A^T(AS^{-2}A^T)^{-1}(\mu AS^{-1}e - b).$$

Preuve: Exercice. □

On a:

$$\delta(y,\mu) = 0 \Leftrightarrow y = y(\mu).$$

Notons par $p_y = \tilde{\alpha}p$ le déplacement en y dans l'ensemble S^0 lors d'une itération de la boucle intérieure. Ce déplacement induit un déplacement:

$$p_s = -A^T p_y$$

dans l'espace des valeurs que peut prendre le vecteur

$$s = c - A^T y.$$

Posons:

$$\begin{aligned} y^+ &= y + p_y \\ s^+ &= s + p_s \\ &= c - A^T y - A^T p_y. \end{aligned}$$

Le déplacement p_y est donné par la méthode de Newton (avec pas unitaire). Ainsi, on a

$$\begin{aligned} p_y &= -(H(y,\mu))^{-1}\nabla\phi_B(y,\mu) \\ &= -(AS^{-2}A^T)^{-1}(-\frac{b}{\mu} + AS^{-1}e) \end{aligned}$$

et donc

$$p_s = A^T(AS^{-2}A^T)^{-1}(-\frac{b}{\mu} + AS^{-1}e). \qquad (16)$$

Convergence

Nous avons tous les éléments pour démontrer la convergence polynomiale de l'algorithme à petit pas donné au tableau 6.3. Rappelons que dans le cas de la méthode à petits pas, il n'y a qu'une seule itération mineure par itération majeure.

```
Initialisation:
  ϵ est le paramètre de précision;
  θ est le paramètre de réduction (0 < θ < 1);
  μ_0 est la valeur initiale de la pénalité;
  y^0 tel que δ(y^0,μ_o) ≤ 1/2;
début
  y := y^0; μ := μ_0;
  tant que μ > ϵ/(4n) faire
  début
    μ := (1 − θ)μ;
    p_y := −(H(y,μ))^{-1} ∇φ_B(y,μ);
    y := y + p_y;
  fin
fin
```

TAB. 3 – *Algorithme de barrière logarithmique à petits pas*

Nous devons montrer dans un premier temps que la suite des itérés est strictement réalisable (lemme 6.3.2) et que μ_k représente une bonne approximation du saut de dualité (lemme 6.3.3). Le nombre d'itérations majeures est alors facile à montrer (théorème 6.3.1). Ces résultats sont valides si l'itéré courant se trouve dans un voisinage du chemin central; ce résultat est montré au lemme 6.3.5.

Le prochain lemme technique relie le pas de Newton à la notion de proximité.

Lemme 6.3.1 *Pour y et μ donnés, on a:*

$$\begin{aligned} \delta(y,\mu) &= ||p_y||_H \\ &= ||S^{-1}A^T p_y||_I \\ &= ||S^{-1}p_s||_I. \end{aligned}$$

Preuve: On a: $||S^{-1}A^Tp_y||_I = ||S^{-1}p_s||_I$. Par la proposition 6.3.1 et l'équation (16), on a (exercice):

$$\delta(y,\mu) = ||S^{-1}A^Tp_y||_I.$$

Finalement, on obtient

$$\begin{aligned} ||S^{-1}A^Tp_y||_I &= \sqrt{p_y^TAS^{-1}S^{-1}A^Tp_y} && (17)\\ &= \sqrt{p_y^TH(y,\mu)p_y} && (18)\\ &= ||p_y||_H. && (19)\end{aligned}$$

□

Le prochain lemme assure la réalisabilité stricte du pas de Newton si le point courant est situé dans le voisinage du chemin central. De plus, la convergence quadratique vers le chemin central est montrée. Notons par S^+ la matrice diagonale formée à partir du vecteur s^+.

Lemme 6.3.2 *Soit $y \in S^0$ dual réalisable et tel que $\delta(y,\mu) < 1$. On a:*

$$\begin{aligned} & y^+ &\in& S^0\\ \textit{et}\quad & \delta(y^+,\mu) &\leq& \delta(y,\mu)^2.\end{aligned}$$

Preuve: On a $S^+ = S(e+S^{-1}p_s)$. Comme $\delta(y,\mu) < 1$, on a par le lemme 6.3.1 que les valeurs absolues des composantes de $S^{-1}p_s$ sont toutes < 1 et par conséquent on a

$$s^+ > 0 \Rightarrow y^+ \in S^0.$$

De plus, on a:

$$\begin{aligned} \delta(y^+,\mu) &= ||\frac{S^+x(y^+,\mu)}{\mu} - e||\\ &\leq ||\frac{S^+x(y,\mu)}{\mu} - e||\\ &= ||\frac{(S+P_s)x(y,\mu)}{\mu} - e||\\ &= ||\frac{(S+P_s)(\mu S^{-1}e - \mu S^{-2}p_s)}{\mu} - e||\\ &= ||P_sS^{-2}p_s||\\ &\leq ||S^{-1}p_s||^2\\ &= \delta(y,\mu)^2.\end{aligned}$$

□

Le lemme suivant borne la différence entre la valeur de l'objectif dual évalué aux points y et $y(u)$. Ce lemme est requis pour montrer que si ϵ est petit et que $\delta(y,\mu) < 1$, alors la valeur b^Ty sera près de la valeur optimale z^*.

Lemme 6.3.3 *Soit $\delta = \delta(y,\mu)$. Si $\delta < 1$, alors*

$$|b^Ty - b^Ty(\mu)| \leq \frac{\delta(1+\delta)}{1-\delta}\mu\sqrt{n}$$

Preuve: On a:

$$\begin{aligned} -p^Tg &= -\delta^2\\ &= p^T(\frac{b}{\mu} - AS^{-1}e)\\ &= \frac{b^Tp}{\mu} - e^TS^{-1}A^Tp,\end{aligned}$$

ce qui implique:

$$\frac{b^Tp}{\mu} = \delta^2 + e^TS^{-1}A^Tp.$$

En utilisant l'inégalité de Cauchy-Schwartz, on a:

$$|e^TS^{-1}A^Tp| \leq \|S^{-1}A^Tp\|\|e\| = \delta\sqrt{n},$$

et donc:

$$|b^Tp_y| \leq \mu(\delta^2 + \delta\sqrt{n})\delta(1+\delta)\mu\sqrt{n}.$$

Soit la séquence $\{y_i\}$,$i = 1,\ldots,\infty$, donnée par $y_0 = y$ et $y_{i+1} = y_i + p_{y_i}$. On a:

$$\begin{aligned} |b^Ty - b^Ty(\mu)| &= |\sum_{i=0}^{\infty}(b^Ty_i - b^Ty_{i+1})|\\ &\leq \sum_{i=0}^{\infty}|b^Tp_{y_i}|\\ &\leq \sum_{i=0}^{\infty}\delta_i(1+\delta_i)\mu\sqrt{n}.\end{aligned}$$

Par le lemme 6.3.2, on a que $\delta_i \leq \delta_0^{2^i} = \delta^{2^i}$. Ainsi, on a:

$$\begin{aligned} \sum_{i=0}^{\infty}\delta_i(1+\delta_i)\mu\sqrt{n} &\leq \sum_{i=0}^{\infty}\delta^{2^i}(1+\delta^{2^i})\mu\sqrt{n}\\ &\leq (1+\delta)\mu\sqrt{n}\sum_{i=0}^{\infty}\delta^{2^i}\\ &\leq (1+\delta)\mu\sqrt{n}\frac{\delta}{1-\delta}.\end{aligned}$$

□

Le prochain lemme borne la distance de l'itéré courant y au nouveau point $y(\mu^+)$ sur le chemin central. Ce lemme permet de déterminer le facteur de réduction qui fera en sorte que l'itéré courant restera dans le voisinage de convergence quadratique de la méthode de Newton.

Lemme 6.3.4 *Soit $\mu^+ = (1-\theta)\mu$. On a:*

$$\delta(y,\mu^+) \leq \frac{1}{1-\theta}(\delta(y,\mu) + \theta\sqrt{n}).$$

Preuve: On a:

$$\begin{aligned} \delta(y,\mu^+) &= \|\frac{Sx(y,\mu^+)}{\mu^+} - e\| \\ &\leq \|\frac{Sx(y,\mu)}{\mu^+} - e\| \\ &= \frac{1}{1-\theta}\|\frac{Sx(y,\mu)}{\mu} - e + \theta e\|. \end{aligned}$$

On a finalement:

$$\begin{aligned} \frac{1}{1-\theta}\|\frac{Sx(y,\mu)}{\mu} - e + \theta e\| &\leq \frac{1}{1-\theta}(\|\frac{Sx(y,\mu)}{\mu} - e\| + \theta\|e\|) \\ &= \frac{1}{1-\theta}(\delta(y,\mu) + \theta\sqrt{n}). \qquad \square \end{aligned}$$

Enfin, le lemme suivant montre que les paramètres de l'algorithme garantissent qu'un pas de Newton amène l'itéré au voisinage du chemin central avec $\delta(y^+,\mu^+) < 1/2$, et donc qu'une seule itération mineure est requise.

Lemme 6.3.5 *Soit $\mu^+ = (1-\theta)\mu$ avec $\theta = \frac{1}{9\sqrt{n}}$. Si $\delta(y,\mu) \leq \frac{1}{2}$, alors $\delta(y^+,\mu^+) \leq \frac{1}{2}$.*

Preuve: On a par le lemme 6.3.4:

$$\delta(y,\mu^+) \leq \frac{1}{1-\frac{1}{9\sqrt{n}}}(\frac{1}{2}+\frac{1}{9}) \leq \frac{1}{1-\frac{1}{9}}(\frac{1}{2}+\frac{1}{9}) = \frac{11}{16} < 1.$$

Ainsi, par le lemme 6.3.2, on a:

$$\begin{aligned} \delta(y^+,\mu^+) &\leq \delta(y,\mu^+)^2 \\ &\leq (\frac{11}{16})^2 < \frac{1}{2}. \qquad \square \end{aligned}$$

On peut maintenant énoncer le théorème de complexité de l'algorithme de barrière logarithmique dual à petits pas.

Théorème 6.3.1 *Après au plus $\frac{1}{\theta}\ln(\frac{4n\mu_0}{\epsilon})$ itérations majeures, l'algorithme arrête avec une solution duale tel que $z^* - b^Ty \leq \epsilon$.*

Preuve: L'algorithme arrête lorsque:

$$\mu_k = (1-\theta)^k\mu_0 \leq \frac{\epsilon}{4n}.$$

Il suffit de prendre:

$$k \geq \frac{1}{\theta}\ln(\frac{4n\mu_0}{\epsilon}),$$

puisque:

$$\frac{1}{\theta} \geq \frac{1}{-\ln(1-\theta)}.$$

De plus, on a:

$$z^* - b^Ty = z^* - b^Ty(\mu_k) + b^Ty(\mu_k) - b^Ty.$$

Par le lemme 6.3.3, on a

$$b^Ty(\mu_k) - b^Ty \leq \frac{\delta(1+\delta)}{1-\delta}\mu_k\sqrt{n}.$$

D'autre part, on a par la proposition 2.4.5:

$$z^* - b^Ty(\mu_k) \leq s(\mu_k)^Tx(\mu_k) = \mu_k n.$$

Ainsi:

$$\begin{aligned} z^* - b^Ty &\leq \mu_k n + \frac{\delta(1+\delta)}{1-\delta}\mu_k\sqrt{n} \\ &= \mu_k(n + \frac{\delta(1+\delta)}{1-\delta}\sqrt{n}). \end{aligned}$$

Puisque:

$$\frac{\delta(1+\delta)}{1-\delta} \leq \frac{3}{2}, \quad \forall\delta \in [0,\frac{1}{2}],$$

on a:

$$z^* - b^Ty \leq \mu_k(n + \frac{3}{2}\sqrt{n}).$$

Finalement, puisque $\mu_k \leq \frac{\epsilon}{4n}$, on obtient:

$$z^* - b^Ty \leq \frac{\epsilon}{4n}(n + \frac{3}{2}\sqrt{n}) \leq \epsilon.$$

$\square$

En choisissant $\epsilon = 2^{-L}$ et $\theta = \frac{1}{9\sqrt{n}}$, l'algorithme effectue au plus un nombre d'itérations majeures égal à

$$\begin{aligned} 9\sqrt{n}\ln(\frac{4n\mu_0}{2^{-L}}) &= 9\sqrt{n}\ln(4n\mu_0 2^L) \\ &= 9\sqrt{n}(\ln(4n\mu_0) + L\ln(2)). \end{aligned}$$

À la fin de l'algorithme, la valeur choisie pour ϵ garantit que l'application de la procédure de purification trouvera un point extrême optimal. De plus, la valeur choisie pour θ permet d'effectuer une seule itération mineure pour chaque itération majeure. Ces observations permettent de conclure que l'algorithme présenté dans cette section est un algorithme polynomial.

Deux autres fonctions barrières (Gonzaga, 1992)

La fonction barrière centre est donnée par

$$\phi_D(y,z) = -g(\ln(b^Ty - z)) - \sum_{i=1}^{n}(\ln(s_i)),$$

où $g \leq n$ est une constante et z est une borne inférieure sur la valeur optimale.

Une autre fonction barrière est donnée par:

$$\phi_P(y,v) = q(\ln(v - b^Ty)) - \sum_{i=1}^{n}(\ln(s_i)).$$

Ici, $q \leq n$ est une constante et v est une borne supérieure sur la valeur optimale.

Il y a une relation entre les valeurs optimales de ces deux fonctions et celles de la barrière logarithmique duale que nous venons de voir. Le chemin central défini par la solution optimale de la fonction $\phi_D(y,z)$ (resp. $\phi_P(y,v)$) paramétrée par la valeur z (resp. v) est le même que celui déterminé par la barrière logarithmique duale (exercice).

Développement par l'approche primale

La méthode primale de chemin central s'obtient en considérant la barrière logarithmique appliquée au problème primal. Suivant le même développement que pour l'approche duale, on aurait obtenu:

$$\begin{aligned}
\phi_B(x,\mu) &= \frac{c^T x}{\mu} - \sum_{i=1}^{n}(\ln(x_i)),\\
\nabla\phi_B(x,\mu) &= \frac{c}{\mu} - X^{-1}e,\\
\nabla^2\phi_B(x,\mu) &= X^{-2}.
\end{aligned}$$

Les conditions nécessaires et suffisantes d'optimalité pour un minimun de $\phi_B(x,\mu)$ sous les contraintes $Ax = b$ sont:

$$\begin{aligned}
Ax = b\\
x > 0\\
\frac{c}{\mu} - X^{-1}e - A^T y = 0.
\end{aligned}$$

En posant $s = \mu X^{-1}e$, on retrouve le système déduit pour le dual.

7 Approche primale-duale

L'approche primale-duale de chemin central considère directement le système perturbé décrivant le chemin central. Étant donné un point (x,y,s), cette fois dans l'espace primal-dual et situé dans un voisinage du chemin central, le pas est donné par la direction de Newton associé au système.

Soit le programme primal:

$$\begin{aligned}
\min_x\ & c^T x\\
\text{s.c.}\quad Ax &= b\\
x &\geq 0
\end{aligned}$$

et le programme dual:

$$\begin{aligned}
\max_y\ & b^T y\\
\text{s.c.}\quad A^T y + s &= b\\
s &\geq 0.
\end{aligned}$$

Les conditions d'optimalité sont:

$$\begin{aligned}
Ax &= b\\
A^T y + s &= c\\
x^T s &= 0\\
x,s &\geq 0.
\end{aligned}$$

Le chemin central est basé sur une perturbation de ces dernières:

$$\begin{aligned}
Ax &= b\\
A^T y + s &= c\\
Xs &= \mu e\\
x,s &> 0.
\end{aligned}$$

Rappelons que ce système purturbé a une solution unique (sous les hypothèses de plein rang et d'intérieur non vide) $(x(\mu),y(\mu),s(\mu))$ pour tout $\mu > 0$ et que le chemin central dans l'espace primal-dual est défini par:

$$\mu \to (x(\mu),y(\mu),s(\mu)).$$

On peut réécrire l'algorithme dual de chemin central sous une forme légèrement différente:

<u>Algorithme primal-dual de chemin central:</u>
Soit $\mu > 0$ et un critère d'arrêt ϵ donné.
Faire (itération majeure)
trouver (approximativement) $(x(\mu),y(\mu),s(\mu))$
mettre à jour $\mu = (1-\theta)\mu$,avec $0 < \theta < 1$
tant que $\mu > \epsilon$.

Ici, l'étape d'optimisation dans l'espace dual (minimisation de la fonction barrière logarithmique), est remplacé par un problème de réalisabilité (satisfaction du système non linéaire) dans l'espace primal-dual.

L'approche primal-dual détermine la solution approximative du système non linéaire par la méthode de Newton. Les trois premiers ensembles d'équations forment un système carré. Réécrivons ce système comme:

$$F(x,y,s) = \begin{bmatrix} Ax - b\\ A^T y + s - c\\ Sx - \mu e \end{bmatrix} \quad = \quad 0$$

avec $x,s > 0$.

La méthode de Newton forme une approximation linéaire du système non linéaire (ici, seul le dernier bloc de contraintes est non linéaire) et itère selon:

$$(x^{k+1},y^{k+1},s^{k+1}) = (x^k,y^k,s^k) + (\Delta x,\Delta y,\Delta s)$$

où $(\Delta x,\Delta y,\Delta s)$ est solution du système linéaire:

$$J(x_k,y_k,s_k)\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta s\end{bmatrix} = -F(x^k,y^k,s^k).$$

J est la matrice Jacobienne du système. On obtient:

$$\begin{bmatrix} A & 0 & 0\\ 0 & A^T & I\\ S & 0 & X \end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta s\end{bmatrix} = \begin{bmatrix} 0\\ 0\\ \mu e - XSe \end{bmatrix}. \qquad (20)$$

De la deuxième équation, on a:

$$\Delta s = -A^T \Delta y.$$

En substituant dans la troisième, on obtient:

$$S\Delta x - XA^T \Delta y = \mu e - XSe.$$

En multipliant par AS^{-1} et sachant que $A\Delta x = 0$, on a:

$$-AS^{-1}XA^T \Delta y = AS^{-1}(\mu e - XSe).$$

Comme A est de plein rang, on a la solution cherchée:

$$\begin{aligned} \Delta y &= -(AS^{-1}XA^T)^{-1}AS^{-1}(\mu e - XSe) && (21) \\ \Delta s &= -A^T \Delta y && (22) \\ \Delta x &= S^{-1}(\mu e - XSe) - S^{-1}X\Delta s. && (23) \end{aligned}$$

Remarques:

1. Comme la matrice A est de plein rang, y est défini de façon unique pour s fixé. Puisque
$$A^T y + s = c$$
on a:
$$y = (AA^T)^{-1}A(c - s),$$
ou encore, $c - s \in \mathcal{R}(A^T)$. Il est donc possible de travailler uniquement dans l'espace (x,s). De même, notons que les directions de Newton en x et en s sont orthogonales. D'une part, on a $\Delta x \in \mathcal{N}(A)$ (premier bloc de contraintes) et d'autre part, $\Delta s \in \mathcal{R}(A^T)$ que l'on peut déduire du deuxième bloc de contraintes.
2. En pratique, on ne calcul pas explicitement l'inverse de la matrice $AS^{-1}XA^T$ pour déterminer Δy. On utilise plutôt une décomposition de Cholesky, la matrice $AS^{-1}XA^T$ étant définie positive et symétrique.

Dans l'espace primal-dual, la notion de proximité au chemin central est explicitement donnée par:

$$\delta(x,s,\mu) = \|\frac{Xs}{\mu} - e\|.$$

Convergence

Nous présentons les résultats de complexité pour l'algorithme primal-dual à petit pas donné au tableau 4 (Gonzaga, 1995).

Le théorème suivant montre que dans le voisinage du chemin central, la méthode de Newton réduit de façon quadratique la mesure de proximité. Il est l'équivalent du lemme 6.3.2.

Théorème 7.0.2 *(Gonzaga, 1995)*
Soit (x,s) primal-dual réalisable et $\mu > 0$ tel que $\delta(x,s,\mu) = \delta < 1$ et (x^+,s^+) l'itéré obtenu par le pas de Newton à partir de (x,s). Alors,

1. $$\delta(x^+,s^+,\mu) \leq \frac{\delta^2}{\sqrt{8}(1-\delta)}$$
2. *Si $\delta \leq 0.7$, alors pour tout $\lambda \in [0,1]$, le point $((x + \lambda\Delta x),(s + \lambda\Delta s))$ est un point intérieur.*

Initialisation:
  ϵ, paramètre de précision;
  θ, paramètre de réduction $(0 < \theta < 1)$;
  μ_0, valeur initiale de la barrière;
  (x^0,s^0) tel que $\delta(x^0,s^0,\mu_0) \leq \frac{1}{2}$;
début
  $x := x^0; s := s^0; \mu := \mu_0;$
  tant que $\mu \geq \epsilon$ faire
  début
    $\mu := (1 - \frac{0.2}{\sqrt{n}})\mu;$
    Calculer Δx etΔs selon (21)-(23)
    $x := x + \Delta x;$
    $s := s + \Delta s;$
  fin
fin

TAB. 4 – *Algorithme primal-dual à petits pas*

Le lemme suivant assure que le nouveau point est dans le voisinage du chemin central.

Lemme 7.0.6
L'algorithme primal-dual génère une séquence de solutions réalisables telle que $\delta(x^k,s^k,\mu_k) \leq \frac{1}{2}$, pour $k = 0,1,\ldots$.

Théorème 7.0.3
L'algorithme primal-dual avec $\epsilon = 2^{-2L}$ se termine en $O(\sqrt{n}L)$ itérations avec une solution $(\tilde{x},\tilde{s})$. Un algorithme de purification appliqué à la solution primale $\tilde{x}$ donne un point extrême optimal.

8 Méthodes prédicteur-correcteur

Plusieurs facteurs font en sorte que les deux algorithmes de chemin central que nous avons décrits (dual et primal-dual) ne sont pas utilisés en pratique sous la forme donnée. Les facteurs les plus significatifs sont:

1. Une solution initiale réalisable se situant dans le voisinage du chemin central peut être difficile à obtenir, ou encore, le programme linéaire peut simplement ne pas être réalisable;
2. Une approche à petit pas ne permet pas une décroissance rapide de l'objectif;
3. La convergence quadratique de la méthode de Newton dans un voisinage du chemin central est l'élément important qui permet d'obtenir une complexité polynomiale. Cependant, la direction de Newton peut être trop biaisée vers sa composante *centrage* par rapport à sa composante *descente*.

Aujourd'hui, la majorité des implantations sont basées sur des variantes (souvent techniques) de la méthode prédicteur-correcteur de chemin central à itérés non-réalisables de Mehrotra (1992).

Non réalisabilité

Les algorithmes que nous avons vu jusqu'à maintenant requièrent une solution initiale qui soit:

1. réalisable (duale ou primal-dual selon le cas);
2. strictement intérieur; et
3. dans un certain voisinage du chemin central.

Il n'est pas toujours facile de trouver un tel point. En fait, certains (même plusieurs) programmes linéaires réalisables ne possèdent pas de point strictement intérieur. Deux approches sont possibles. Une première (aujourd'hui plus théorique que pratique) transforme le programme linéaire en un programme auxiliaire pour lequel un point initial est facilement identifiable et dont la solution optimale fournit la solution au programme original ou montre que le programme est non réalisable. La deuxième approche est *simplement* le développement d'algorithmes qui ne requièrent pas un point réalisable initial (mais seulement un point avec composantes strictement positives).

Reformulation

Soit le programme linéaire

$$\begin{array}{rcl} \min_x c^T x & & \\ \text{s.c. } Ax & = & b \\ x & \geq & 0. \end{array}$$

Considérons le programme auxiliaire:

$$\begin{array}{rcl} \min c^T x + Mx_{n+1} & & \\ Ax + (\frac{b}{M} - Ae)x_{n+1} & = & b \\ -(e - \frac{c}{M})^T x - x_{n+2} & = & -(n+1)M + e^T c \\ x \geq 0, x_{n+1} \geq 0, x_{n+2} \geq 0, & & \end{array}$$

et son dual:

$$\begin{array}{rcl} \max b^T y - [(n+1)M + e^T c]s_{n+2} & & \\ A^T y + s - (e - \frac{c}{M})s_{n+2} & = & c \\ (\frac{b}{M} - Ae)^T y + s_{n+1} & = & M \\ s \geq 0, s_{n+1} \geq 0, s_{n+2} \geq 0, & & \end{array}$$

où $M = O(2^L)$. En posant:

$$\begin{array}{l} x^0 = Me, x^0_{n+1} = x^0_{n+2} = M \\ y^0 = 0, s^0 = Me, s^0_{n+1} = s^0_{n+2} = M \end{array}$$

on a une solution strictement intérieure $x^0 > 0, s^0 > 0$, et primal-dual réalisable. De plus, comme

$$S^0 x^0 = M^2 e$$

cette solution est située sur le chemin central.

On peut montrer que si la solution optimale vérifie $x_{n+1} = s_{n+2} = 0$, alors la solution correspondant à x et y résout respectivement le programme primal et dual. Par contre, si $x_{n+1} = 0$ et $s_{n+2} > 0$, le programme primal est non borné; si $x_{n+1} > 0$ et $s_{n+2} = 0$, le programme primal est non réalisable; enfin si $x_{n+1} > 0$ et $s_{n+2} > 0$, le programme primal est soit non réalisable, soit non borné.

Il existe une autre reformulation due à Ye, Todd et Mizuno (1994) et qui exploite la notion de programme linéaire auto-dual (self-dual) ne faisant pas intervenir de grand M. Un programme linéaire est dit auto-dual si les programme primal et dual sont les mêmes.

Point initial non réalisable

Supposons un point initial positif $x > 0$ et $s > 0$ mais non primal-dual réalisable. Les résidus des systèmes linéaires sont définis par:

$$\begin{array}{rrcl} & r_P & = & Ax - b \\ \text{et} & r_D & = & A^T y + s - c. \end{array}$$

Le système linéarisé en $\Delta x, \Delta s$ et Δy que l'on cherche à résoudre est alors:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b - AX \\ c - A^T y - s \\ \mu e - XSe \end{bmatrix} = \begin{bmatrix} -r_P \\ -r_D \\ \mu e - XSe \end{bmatrix},$$

dont la solution (exercice) est:

$$\begin{array}{rcl} \Delta y & = & -(AS^{-1}XA^T)^{-1} \\ & & [AS^{-1}(\mu e - XSe) - AS^{-1}Xr_D - r_P] \\ \Delta s & = & -A^T \Delta y + r_D \\ \Delta x & = & S^{-1}(\mu e - XSe) - S^{-1}X\Delta s. \end{array}$$

Ce système ne diffère que légèrement du système linéaire (20) obtenu pour le cas réalisable. Si un pas complet de Newton est pris, la réalisabilité est obtenue en une itération et tous les itérés subséquents seront primal-dual réalisables.

Ceci n'est pas le cas dans les méthodes à long pas dont, en particulier, l'algorithme prédicteur-correcteur. L'analyse de convergence et de complexité (toujours polynomiale) de tels algorithmes requiert alors une nouvelle définition de voisinage faisant intervenir, en plus de la déviation par rapport au chemin central, une borne supérieure sur la norme du résidu. Comme cette borne est proportionnelle au paramètre μ, et que $\mu \to 0$, la norme du résidu tend vers 0 et alors $r_P^k \to 0$ et $r_D^k \to 0$ lorsque $k \to \infty$.

Algorithme prédicteur-correcteur

Considérons de nouveau le système linéarisé qui donne la direction primal-dual de Newton (on se place dans la situation où l'on dispose d'un point strictement réalisable):

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e - XSe \end{bmatrix}.$$

Rajoutons un paramètre σ dans le membre de droit:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - XSe \end{bmatrix}. \tag{24}$$

On obtient évidemment le même système si $\sigma = 1$, et le pas de Newton fournit donc la direction vers le chemin central que l'on appelle direction de centrage. Cette direction nous a permis d'établir la convergence polynomiale de l'algorithme primal-dual. La décroissance du gap étant assurée par la décroissance du paramètre de pénalité (généralement faible).

Par contre, si $\sigma = 0$, la direction de Newton obtenue est celle qui vise à diminuer le gap en une seule itération, c'est-à-dire, la solution du système linéarisé des conditions nécessaires et suffisantes d'optimalité du programme linéaire. Appelons cette direction la direction affine. Elle permet une décroissance rapide de l'objectif, mais tend vers la frontière en s'éloignant du chemin central (et des propriétés de convergence quadratique).

Les méthodes prédicteur-correcteur tirent profit de ces deux directions en alternant entre elles:

- Pas prédicteur avec $\sigma = 0$,
- Pas correcteur avec $\sigma = 1$.

L'algorithme donné au tableau 5 présente une méthode prédicteur-correcteur.

Initialisation:
  ϵ, paramètre de précision;
  (x^0, s^0) tel que $\delta(x^0, s^0, \mu_0) \leq \frac{1}{4}$;
début
  tant que $\mu_k \geq \epsilon$ faire
  début
    Si k est pair (pas prédicteur):
      Calculer Δx, Δs et Δy solution du système (24) avec $\sigma = 0$;
      Déterminer le pas maximum $\alpha \in [0,1]$ tel que $\delta \leq 1/2$;
      $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k(\alpha), y^k(\alpha), s^k(\alpha))$;
      $\mu_{k+1} = (x^{k+1})^T s^{k+1}/n$;
    Sinon (pas correcteur)
      Calculer Δx, Δs et Δy solution du système (24) avec $\sigma = 1$;
      $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\Delta x, \Delta y, \Delta s)$
      $\mu_{k+1} = \mu_k$
    fin
  fin
fin

TAB. 5 – *Algorithme prédicteur-correcteur*

Algorithme prédicteur-correcteur à itérés non réalisables

Les principaux algorithmes de points intérieurs (CPLEX/-Barrier, LOQO, HOPDM, LIPSOL) sont basés sur des variantes de l'algorithme prédicteur-correcteur de Mehrotra avec itérés non réalisables. Essentiellement, il consiste en l'algorithme prédicteur-correcteur présenté au tableau 5 où les directions tiennent compte des non réalisabilités primales et duales. De plus, la valeur de σ dans l'étape de centrage est adaptée au besoin (ou non) de recentrer l'itéré obtenu après l'étape prédiction. Enfin, une certaine non linéarité est prise en compte lors du calcul de la direction de centrage. Nous suivons ici la présentation faite par Wright (1997).

L'étape prédiction calcule la direction affine donnée par la solution du système:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta\hat{x} \\ \Delta\hat{y} \\ \Delta\hat{s} \end{bmatrix} = \begin{bmatrix} -r_P \\ -r_D \\ -XSe \end{bmatrix}.$$

L'itéré est donné par:

$$\begin{aligned} \hat{x} &= x + \hat{\alpha}_P \Delta\hat{x} \\ \hat{s} &= s + \hat{\alpha}_D \Delta\hat{s} \end{aligned}$$

où:

$$\begin{aligned} \hat{\alpha}_P &= \arg\max\{\alpha \in [0,1] \ : \ x + \alpha\Delta\hat{x} \geq 0\} \\ \hat{\alpha}_D &= \arg\max\{\alpha \in [0,1] \ : \ s + \alpha\Delta\hat{s} \geq 0\}. \end{aligned}$$

La direction affine s'est avérée une bonne direction si le saut de dualité au nouvel itéré donné par:

$$\hat{\mu} = (\hat{x})^T \hat{s}/n$$

est beaucoup plus petit que μ. Ceci indique qu'il est peu nécessaire de centrer l'itéré et la valeur de σ est fixée proche de 0. Au contraire, si la diminution du gap a été petite, il y a lieu de recentrer le point afin de permettre une descente affine plus considérable à l'itération suivante. C'est dans cet esprit que l'ajustement suivant est proposé:

$$\sigma = (\frac{\hat{\mu}}{\mu})^3.$$

Étant donnée la valeur de σ, on peut utiliser l'information obtenue lors de l'étape affine dans le calcul de la direction de centrage. Rappelons le système non linéaire associé à un point sur le chemin central:

$$\begin{array}{rcl} S\Delta x + X\Delta s &=& \mu e - XSe - \Delta X \Delta S e \\ A\Delta x &=& -r_P \\ A^T\Delta y + \Delta s &=& -r_D. \end{array}$$

La direction primale-duale de Newton est donnée en négligeant la partie non linéaire de ce système, soit le terme:

$$-\Delta X \Delta S e.$$

Le pas de Newton obtenue à l'étape prédiction (direction affine) est donc tel que:

$$(X + \delta\hat{X})(s + \delta\hat{s}) = \Delta\hat{X}\Delta\hat{s},$$

au lieu de la valeur 0. Cette erreur est prise en compte et le pas de centrage est alors solution du système linéaire:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta\tilde{x} \\ \Delta\tilde{y} \\ \Delta\tilde{s} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - \Delta\hat{X}\Delta\hat{S} \end{bmatrix}.$$

La direction combinée est donnée par:

$$(\Delta x, \Delta y, \Delta s) = (\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{s}) + (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{s}).$$

et le nouvel itéré:

$$\begin{array}{rcl} x^{k+1} &=& x^k + \alpha_P \Delta x \\ s^{k+1} &=& s^k + \alpha_D \Delta s \end{array}$$

où: α_P (resp. α_D) est le minimum entre 1 (pas de Newton) et $\tilde{\alpha}_P$ (resp. $\tilde{\alpha}_D$) avec:

$$\begin{array}{rcl} \tilde{\alpha}_P &=& 0.99 \times \arg\max\{\alpha \in [0,1] \ : \ x^k + \alpha\Delta x \geq 0\} \\ \tilde{\alpha}_D &=& 0.99 \times \arg\max\{\alpha \in [0,1] \ : \ s^k + \alpha\Delta s \geq 0\}. \end{array}$$

9 Algorithme de Karmarkar

L'algorithme de Karmarkar (1984) est une méthode projective avec potentiel (ou fonction mérite). Cet algorithme est semblable à la méthode affine primale. Elle diffère par la transformation utilisée, projective plutôt qu'affine, ainsi que par l'utilisation d'une fonction potentiel qui permet de montrer la convergence polynomiale.

On considère le programme linéaire suivant:

$$\begin{array}{rrcl} \min\limits_x & c^T x & & \\ \text{s.c.} & Ax &=& 0 \\ & e^T x &=& 1 \\ & x &\geq& 0. \end{array}$$

Cette forme n'est pas restrictive puisque tout programme linéaire peut se ramener à cette forme (en classe).

On pose les trois hypothèses suivantes:

A_1 : rang$(A) = m$

A_2 : $int(S) \neq \emptyset$ et l'ensemble des optimal est borné

A_3 : $\min\limits_{x\in S} \ c^T x = 0.$

La troisième hypothèse n'est que technique. Il est possible d'obtenir une convergence polynomiale sans cette hypothèse par l'utilisation d'une borne inférieure de la valeur optimale (obtenue par une solution duale).

Soit Δ le simplexe défini par:

$$\Delta = \{x \in \Re^n \ : \ e^T x = 1, x \geq 0\}$$

L'algorithme de Karmarkar suit les étapes suivantes:

- Etant donné le point courant x_k, on effectue une transformation projective (TP) afin de centrer le point courant $x_k \to \frac{1}{n}e$ et conserver le problème dans le simplexe unité Δ.
- On se déplace, dans le nouvel espace, dans la direction opposée au gradient projeté sur le noyau de $\begin{bmatrix} \hat{A} \\ e^T \end{bmatrix}$ avec un pas relié au rayon de la plus grande sphère inscrite dans le simplexe Δ.

Transformation projective:

Soit $\bar{X} \equiv X_k$ la solution courante. La transformation est définie par:

$$x \to \hat{x} = \frac{\bar{X}^{-1}x}{e^T\bar{X}^{-1}x}$$

et son inverse par:

$$\hat{x} \to x = \frac{\bar{X}\hat{x}}{e^T\bar{X}\hat{x}}.$$

Sous cette transformation, on peut vérifier que le point courant $\bar{x}$ devient $\frac{1}{n}e$ et que le simplexe unité Δ demeure inchangé. En fait, la transformation projective est la composition d'une transformation affine:

$$\hat{x} \to \tilde{x} = \frac{1}{n}\bar{X}^{-1}x$$

suivie d'une projection radiale:

$$\tilde{x} \to \hat{x} = \frac{\tilde{x}}{e^T \tilde{x}}.$$

Sous la transformation projective, le programme linéaire devient:

$$\begin{array}{lrcl} \min & \frac{c^T \bar{X} \hat{x}}{e^T \bar{X} \hat{x}} & & \\ \text{s.c.} & A\bar{X}\hat{x} & = & 0 \\ & \hat{x} & \in & \Delta. \end{array}$$

Les hypothèses A_2 et A_3 permettent d'écrire ce programme non linéaire comme le programme linéaire:

$$\begin{array}{lrcl} \min & \hat{c}^T \hat{x} & & \\ \text{s.c.} & \hat{A}\hat{x} & = & 0 \\ & \hat{x} & \in & \Delta. \end{array}$$

Calcul de la direction et du pas:

Considérons le programme non linéaire suivant:

$$\begin{array}{lrcl} \min & \hat{c}^T \hat{x} & & \\ \text{s.c.} & \hat{A}\hat{x} & = & 0 \\ & \hat{x} & \in & S(\frac{e}{n},\rho) \end{array} \tag{25}$$

où:

$$S(\frac{e}{n},\rho) = \{x \in R^n : e^T x = 1, ||x - \frac{e}{n}|| \le \rho\}$$

est la sphère de rayon ρ centrée au centre du simplexe.
La direction de plus forte pente à partir du centre de $S(\frac{e}{n},\rho)$ est donnée par (exercice):

$$d = \text{proj}_B(-\hat{c}) = -(I - B^T (BB^T)^{-1} B)\hat{c}$$

où

$$B = \begin{bmatrix} \hat{A} \\ e^T \end{bmatrix}.$$

La proposition suivante donne les rayons de la plus grande (resp. plus petite) sphère incluse dans (resp. contenant) le simplexe.

Proposition 9.0.2 *Soit* $r = (n(n-1))^{-\frac{1}{2}}$ *et* $R = \sqrt{\frac{n-1}{n}}$. *On a*

$$S(\frac{e}{n},r) \subseteq \Delta \subseteq S(\frac{e}{n},R).$$

Le nouvel itéré est donné par:

$$\hat{x}_{k+1} = \hat{x}_k + \frac{\rho}{||d||} d.$$

En pratique, on choisit $\rho = \theta\, r$ avec θ proche de 0.995. Pour assurer la complexité polynomiale, θ est posé à $\frac{1}{3}$.
Le tableau 6 présente l'algorithme de Karmarkar.

```
Initialisation:
  ρ (typiquement ρ = 0.995((n(n − 1))^(−1/2));
  x_0 = e/n point initial intérieur;
début
  x_c := x_0;
  tant que critère d'arrêt pas satisfait faire
  début
    X := X_c;
    ĉ := Xc;
    Â := AX;
    B := [Â ; e^T];
    d̂ := −(I − B^T(BB^T)^(−1)B)ĉ;
    x̂ := e/n + (ρ/||d||) d
    évaluer critère d'arrêt;
    x_c := Xx̂ / (e^T X x̂);
  fin
fin
```

TAB. 6 – *Algorithme de Karmarkar*

Calcul de la direction et du pas:

Si l'on prend

$$\rho = \frac{1}{3}((n(n-1))^{-\frac{1}{2}}),$$

alors l'algorithme de Karmarkar génère une suite $\{x^k\} > 0$ qui satisfait:

$$c^T x^k \le \exp(-k/5n) c^T x^0.$$

Il suffit de prendre $k \ge 10nL \ln 2$, pour obtenir une solution qui satisfait

$$c^T x^k \le 2^{-2L} c^T x^0.$$

10 Autres programmes mathématiques

Nous voyons dans cette section comment les méthodes primales-duales s'appliquent à des programmes mathématiques qui partagent des structures semblables. Ces problèmes sont présentés avec plus de détails dans Wright (1997).

Problème de complémentarité linéaire monotone

Le problème de complémentarité monotone s'énonce comme suit:
Déterminer les points x et s dans $\Re^n$ tels que:

$$\begin{array}{rcl} s & = & Mx + q \\ x^T s & = & 0 \\ x,s & \ge & 0, \end{array}$$

où M est une matrice $n \times n$ semi-définie positive. Ce système non linéaire est très semblable au système KKT donné par les conditions nécessaires et suffisantes d'optimalité d'un programme linéaire. On note:

$$F(x,s) = \begin{bmatrix} Mx + q - s \\ Sxe \end{bmatrix} = 0$$

avec $x,s \geq 0$. Le chemin central, paramétré par $\mu \geq 0$, d'un problème de complémentarité linéaire monotone est donné par la solution du système:

$$\begin{aligned} Mx + q - s &= 0 \\ Sx &= \mu e \end{aligned}$$

avec $(x,s) > 0$. L'algorithme primal-dual (et ses extensions) s'applique directement sur ce problème, avec la même complexité polynomiale.

Complémentarité linéaire monotone mixte

Le problème de complémentarité monotone mixte s'énonce comme suit: déterminer les points x et s dans $\Re^n$ et z dans $\Re^m$ tels que:

$$\begin{aligned} \begin{bmatrix} s \\ 0 \end{bmatrix} &= \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \\ x^T s &= 0 \\ x,s &\geq 0 \end{aligned}$$

où M est une matrice $(n+m)\times(n+m)$ semi-définie positive.

Clairement, le problème de complémentarité linéaire monotone est un cas particulier de celui-ci (on peut montrer également l'inverse, les deux formulations étant en fait équivalentes). Il est possible (exercice) de montrer que la programmation linéaire se ramène à un problème de complémentarité linéaire mixte monotone.

Selon les mêmes lignes que nous avons vues dans ce cours, il est possible de développer un algorithme primal-dual (et ses extensions) pour ce problème où la direction de Newton, à un itéré strictement positif, est donnée par:

$$\begin{bmatrix} M_{11} & M_{12} & -I \\ M_{21} & M_{22} & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \\ \Delta s \end{bmatrix} = - \begin{bmatrix} M_{11}x + M_{12}z + q_1 - s \\ M_{21}x + M_{22}z + q_2 \\ -\mu e + XSe \end{bmatrix}$$

Les résultats de convergence et de complexité des algorithmes primal-dual se généralisent pour ce problème.

Problème quadratique convexe

Le programme quadratique convexe (forme standard) se formule comme:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Q x + c^T x \\ \text{s.à} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

où Q est une matrice $(n \times n)$ semi-définie positive.

Les conditions nécessaires et suffisantes d'optimalité pour le problème quadratique convexe sont (exercice):

$$\begin{aligned} \begin{bmatrix} s \\ 0 \end{bmatrix} &= \begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} \\ x^T s &= 0 \\ x,s &\geq 0 \end{aligned}$$

Un problème quadratique convexe est donc un cas particulier du problème de complémentarité linéaire mixte. Un algorithme primal-dual de complexité polynomiale pour ce dernier problème résout donc le problème quadratique convexe en temps polynomial.

Problème convexe

Considérons le programme convexe suivant:

$$\begin{aligned} \max_y \quad & f_0(y) \\ \text{s.c.} \quad & f_i(y) \leq 0, \quad i = 1, \ldots, n. \end{aligned}$$

où $-f_0(y)$ et $f_i(y), i = 1, \ldots, n$ sont des fonctions convexes au moins trois fois continûment différentiables de $\Re^m \to \Re$. On pose les hypothèses suivantes:

$$\begin{aligned} & A_1 : \text{int}(S) = S^0 \neq \emptyset \quad \text{(condition de Slater)}, \\ & A_2 : S \text{ est borné}. \end{aligned}$$

Les conditions d'optimalité du premier ordre (KKT), nécessaires et suffisantes sous les hypothèses posées, sont l'existence d'un point y et d'un vecteur de multiplicateurs x qui satisfont:

$$\begin{aligned} \nabla f_0(y) - \sum_{i=1}^n x_i \nabla f_i(y) = 0 & \\ f_i(y) + s_i = 0, & \quad i = 1, \ldots, n \\ s_i x_i = 0, & \quad i = 1, \ldots, n \\ x,s \geq 0, & \quad i = 1, \ldots, n. \end{aligned}$$

Il est possible de généraliser l'algorithme primal-dual pour résoudre le problème convexe. L'itéré de Newton est alors solution de:

$$\begin{bmatrix} \nabla^2_{yy} L(y,x) & F(y)^T & 0 \\ F(y)^T & & I \\ 0 & S & X \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} =$$

$$- \begin{bmatrix} -\nabla f_0(y) + F(y)^T x \\ f(y) + s \\ -\mu e + XSe \end{bmatrix}.$$

La non linéarité du problème requiert l'utilisation de l'algorithme à itérés non réalisables, controlés par l'utilisation d'un voisinage qui limite la non réalisabilité en fonction de la valeur de μ. L'analyse de complexité requiert que la fonction barrière utilisée vérifie la condition de self-concordance (Nesterov et Nemirovski, 1994). Cette condition assure que, pour tout point strictement intérieur, la troisième dérivée de la fonction barrière pour toutes directions, est bornée en terme de la deuxième dérivée.

11 Références

1. E.R. Barnes and S. Chopra and D.J. Jensen, *A polynomial–time version of the affine–scaling algorithm*, Working Paper Series, Graduate School of Business and Administration, New York University, Number 88-101, 1988.
2. M.S. Bazaraa, H.D. Sherali et C.M. Shetty, Nonlinear Programming : Theory and Algorithms, Wiley, New York, 1993.
3. R.G. Bland, *New finite pivoting rules for the simplex method*, Mathematics of Operations Research, Vol. 2, pp. 103-107, 1977.
4. V. Chvatal, Linear Programming, W.H. Freeman and Company, New York, 1983.
5. D. den Hertog, Interior Point Approach to Linear, Quadratic and Convex Programming, Kluwer Academic Publishers, Dordrecht, 1994.
6. J.E. Dennis and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Englewood Cliffs, NJ, 1983.
7. I.I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Doklady Akademii Nauk SSSR, Vol. 174, pp. 747-748, 1967.
8. A.V. Fiacco and G.P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, SIAM Classics in Applied Mathematics, Philadelphia, 1990 (1968).
9. D. Goldfard and M.J. Todd, *Linear programming*, in Optimization, G.L. Nemhauser, A.H.G. Rinnooy Kan et M.J. Todd, eds., North-Holland, Amsterdam, pp. 141-170, 1989.
10. G.H. Goldman and A.W. Tucker, *Theory of linear programming*, in Linear Equalities and Related Systems, H.W. Kuhn and A.W. Tucker, eds., Princeton University Press, Princeton, pp. 53-97, 1956.
11. C.C. Gonzaga, *An algorithm for solving linear programming problems in $O(n^3L)$ operations*, Progress in Mathematical Programming: Interior Point and Related Methods, pp. 1-28, Springer Verlag, 1989.
12. C.C. Gonzaga, *Path-following methods for linear programming*, SIAM Review, Vol. 34, pp.167-224, 1992.
13. C.C. Gonzaga, *On the complexity of linear programming*, Resenhas do Instituto de Matemática e Estatístíca da Universidade de Sao Paulo, Vol. 2, pp. 197-207, 1995.
14. N. K. Karmarkar *A new polynomial–time algorithm for linear programming*, Combinatorica, Vol. 4, pp. 373-395, 1984
15. V.Klee and G.J. Minty, *How good is the simplex algorithm?*, in Inequalities, III, O. Shisha (editor), Academic Press, New York, pp. 159-175, 1972.
16. L.G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Mathematics Doklady, Vol. 20, pp. 191-194, 1979.
17. K.O. Kortanek and J. Zhu *New purification algorithms for linear programming*, Naval Research Logistics Quarterly, Vol. 35, pp. 571-583, 1988.
18. S. Mehrotra, *On the implementation of a primal-dual interior point method*, Siam Journal on Optimization, Vol. 2, pp. 575-601, 1992.
19. S. Mizuno and A. Nagasawa, *A primal–dual affine scaling potential reduction algorithm for linear programming*, Mathematical Programming, Vol. 62, pp. 119-131, 1993.
20. R.D.C. Monteiro, I. Adler and M.G.C. Resende, *A polynomial–time primal–dual affine scaling algorithm for linear and convex quadratic programming and its power series extension*, Mathematics of Operations Research, Vol. 15, pp. 191-214, 1990.
21. S.G. Nash and A. Sofer, Linear and Nonlinear Programming, McGray-Hill, New York, 1996.
22. Y.E. Nesterov and A. Nemirovskii, Interior-Point Polynomial Algorithms in Convex Programming, SIAM, 1994.
23. J. Renegar, *A polynomial–time algorithm, based on Newton's method, for linear programming*, Mathematical Programming, Vol. 40, pp. 59-93, 1988.
24. C. Roos and J.-Ph. Vial, *A polynomial method of approximate centers for linear programming*, Mathematical Programming, Vol. 54, pp. 295-305, 1992.
25. C. Roos, T. Terlaky and J.-Ph. Vial, Theory and Algorithms for Linear Optimization: An Interior Point Approach, John Wiley & Sons, 1997
26. R. Saigal, Linear Programming; A Modern Integrated Analysis, International Series in Operations Research and Management Science, Vol. 1, Kluwer Academic Publishers, November 1995.
27. A. Schrijver, Theory of Linear and Integer Programming, John Wiley & Sons, 1986.
28. P.M. Vaidya, *An algorithm for linear programming which requires $O((m+n)n^2+(m+n)^1.5nL)$ arithmetic operations*, Mathematical Programming, Vol. 47, pp. 175-201, 1990.

29. R.J. Vanderbei, Linear Programming; Foundations and Extensions, Kluwer Academic Publishers, 1996.
30. M.H. Wright, *Interior Methods for Constrained Optimization*, Acta Numerica, Cambridge University Press, New York, 1992, pp. 341-407.
31. S. Wright, Primal-dual interior-point methods, SIAM, Philadelphia, 289 pp., 1997.
32. Y. Ye, Interior Point Algorithms: Theory and Analysis. 1997, John Wiley and Sons, New York Wiley-Interscience Series in Discrete Mathematics Optimization, 1997.
33. Y. Ye, M.J. Todd and S. Mizuno, *An* $O(\sqrt{n}L)$*-iteration homogeneous and self-dual linear programming algorithm*, Mathematics of Operations Research, Vol. 19, pp. 53-67, 1994.